British Library Research and Innovation Report 146

WebWatching UK Web Communities:
Final Report For The WebWatch Project

Brian Kelly and Ian Peacock

British Library Research and Innovation Centre

1999

Abstract

This document is the final report for the WebWatch project. The aim of the project was to develop and use robot software for analysing and profiling Web sites within various UK communities and to report on the findings.

This document reviews the original bid, gives a background to robot software, describes the robot software used by the WebWatch project, and summaries the conclusions gained from the WebWatch trawls. A list of recommendations for further work in this area is given. The appendices include a number of the reports which have been produced which describe the main trawls carried out by the project.

Author(s)

Brian Kelly and Ian Peacock, UKOLN (UK Office for Library and Information Networking).

Brian Kelly is UK Web Focus at UKOLN, the University of Bath. Brian managed the WebWatch project.

Ian Peacock is WebWatch Computer Officer at UKOLN, the University of Bath. Ian was responsible for software development and analysis and producing reports.

Disclaimer

The opinions expressed in this report are those of the authors and not necessarily those of the British Library.

Grant Number

RIC/G/383

Availability of this Report

British Library Research and Innovation Reports are published by the British Library Research and Innovation Centre and may be purchased as photocopies or microfiche from the British Thesis Service, British Library Document Supply Centre, Boston Spa, Wetherby, West Yorkshire LS23 7BQ, UK.

In addition this report is freely available on the World Wide Web at the address <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/reports/final/`>.

ISBN and ISSN Details

ISBN 0 7123 97361

ISSN 1366-8218

Table of Contents

1 Introduction

This document is the final report for the WebWatch project. The WebWatch project has been funded by the BLRIC (the British Library Research and Innovation Centre).

The aim of this document is to describe why a project such as WebWatch was needed, to give a brief overview of robot software on the World Wide Web, to review the development of the WebWatch robot software, to report on the WebWatch trawls which were carried out and to summarise the conclusions drawn from the trawl.

This document includes the individual reports made following each of the trawls, in order to provide easy access to the reports.

The structure of this document is summarised below:

Section 2 gives an outline of the WeWatch project.

Section 3 gives a background to robot software, describes the robot software used by the WebWatch project.

Section 4 provides background reading on robot technologies.

Section 5 reviews the trawls carried out by WebWatch.

Section 6 summaries the conclusions gained from the WebWatch trawls.

Section 7 describes Web-based services which have been produced to accompany the WebWatch robot software development.

Section 8 describes possible future developments for automated analyses of Web resources.

The appendices include a number of the reports which have been produced which describe the main trawls carried out by the project and reviews WebWatch dissemination.

2 Outline of the WebWatch Project

UKOLN submitted a proposal to the British Library Research and Innovation Centre to provide a small amount of support to a WebWatch initiative at UKOLN. The aim of the project was to develop a set of tools to audit and monitor design practice and use of technologies on the Web and to produce reports outlining the results obtained from applying the tools. The reports provided data about which servers are in use, about the deployment of applications based on ActiveX or Java, about the characteristics of Web servers, and so on. This information should be useful for those responsible for the management of Web-based information services, for those responsible for making strategic technology choices and for vendors, educators and developers.

Aims and Objectives

WebWatch aims to improve Web information management by providing a systematic approach to the collection, analysis and reporting of data about Web practices and technologies.

Specific objectives include:

- Develop tools and techniques to assist in the auditing of Web practice and technologies
- Assist the UK Web Focus to:
  - develop a body of experience and example which will guide best practice in Web information management
  - advise the UK library and information communities
- Set up and maintain a Web-based information resource which reports findings
- Document its findings in a research report.

Timeliness

After a first phase which has seen the Web become a ubiquitous and simple medium for information searching and dissemination we are beginning to see the emergence of a range of tools, services and formats required to support a more mature information space. These include site and document management software, document formats such as style sheets, SGML and Adobe PDF, validation tools and services, metadata for richer indexing, searching and document management, mobile code and so on.

Although this rapid growth is enhancing the functionality of the Web, the variety of tools, services and architectures available to providers of Web services is increasing the costs of providing the service. In addition the subsequent diversity may result in additional complications and expenses in the future. This complication makes it critical to have better information available about what is in use, what the trends are, and what commonality exists.

Benefits

Benefits include better information for information providers, network information managers, user support staff and others about practice and technologies.

Methodology

Work proceeded as follows:

- *Development of a robot to retrieve institutional Web pages, from a database of institutional URLs*
  An automated robot which retrieves Web resources from an input file was developed. The robot was written in Perl, and makes use of the libwww Perl library. Originally use was made of a tailored version of the Harvest Gatherer. As limitations in this robot became apparent modifications were made to a number of modules, until we finally were using a locally-developed robot suite.
- *Development of a suite of programs to analyse results retrieved by robot*
  Software to analyse the results from the robot was used. This included locally developed Unix utilities to process the data into a format suitable for use with other applications, and desktop applications including Microsoft Excel and SPSS.
- *Production of reports*
  Examples of areas covered by the reports included:

    - Report on the quality and type of HTML used in institutional home pages (e.g. conformance to HTML standard, use of HTML extensions such as frames, use of mobile code such as Java) and size of the institutional home page (e.g. number and size of images).
    - Report on the use of metadata.
    - Report on the numbers of hypertext links in pages.
- *Liaison with specific communities:*
  The robot was used to retrieve resources from particular domains and subject areas, including public libraries and library services within Higher Education. Reports on these trawls enabled:
    - Analyses on the uptake of new Web facilities to be monitored within a subject area. For example, use of metadata tags within the Library community.
    - Periodic surveys to be carried out to observe trends within the communities.

3 Background to Robot Technologies

Introduction

Robots are automated software agents designed to carry out work that is repetitive or, because of sheer scale, not possible as an individual human undertaking.

The definition of a software agent is subject to debate, but one relevant notion of an autonomous agent is "a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future".

Autonomous agents are not really a new phenomena on the Internet. Unix, the operating system over which the Internet was borne utilizes network *daemons*, which are essentially simple automata for dealing with network transactions. Newsbots, IRCbots and other application-specific autonomous agents are now common-place on the Internet.

Agents and the Web

The WebWatch project is concerned with automated Web agents. These are robots that traverse the Web, navigating through hyperlinks and performing a job, such as recognising and indexing resources. Such Web crawling robots are known by a variety of aliases, including crawlers, trawlers, spiders and worms.

The current generation of Web robots is, perhaps surprisingly, large. A glance at the Web Robots Pages [1] reveals a list of over 160 well-known robots. These robots are written in a variety of different languages. Some may use libraries that are written to aid Web agent design. Two popular libraries used for this purpose are libwww for the 'C' language and LWP for Perl.

A majority of robots navigating the Web at this time are used by search-engines to index keywords and other metadata to build indices for resource retrieval. Other robots are available to maintain hypertext structures or to summarise resources in other ways (e.g. monitoring technologies rather than indexing).

Web robot behaviour could be broadly classified into:

- Exploratory Web crawling (e.g. resource discovery or broken link discovery)
- Directed Web crawling (e.g. of URLs submitted to a search engine)
- Restricted Web crawling (e.g. all URLs within a DNS domain).

The Role of Robots in the Web

Maintaining the Web

The dynamic nature of the Web, and the fact that it is composed of administratively independent "Web sites" leads to the danger that states of disrepair can arise because extensive checking can be too big and difficult an undertaking. This maintenance is necessary to keep up with changes in the structure of Web space and changes in technologies and standards. For example:

- When pages change, come into existence or cease to exist, references to those pages must be correspondingly changed, created or removed.
- In order that pages are displayed as required, there must be conformance to standards such as HTML.

Understanding the Web

As a widely used and strategically important communications medium, the Web warrants analysis of its structure to provide insight into its nature and to monitor its development. The results of these analyses can help to solve the problems that are associated with the Web. Monitoring change and growth may help predict future trends and development and to recognise the potential or necessity for enhancements and developments. This may be true on the small scale of Web areas under the same administrative control.

This is an area in which the World Wide Web Consortium is actively involved through the HTTP/NG's Web Characterisation working group [2].

Benefits

Spiders offer an effective solution to obtaining a user view of remote Web space. Simulating a human user, they have the advantage of being able to repetitively cover large subspaces of Web and keep accurate logs and summaries. Additionally, they can be run for long periods or at regular periods without fatiguing and can monitor changes that may not be apparent to a normal user.

General benefits include:

- User satisfaction from search directed access to resources and easier browsability (via maintenance and advancements of the Web resulting from analyses).
- Reduced network traffic in document space resulting from search-directed access.
- Effecting archiving/mirroring, and populating caches (to produce associated benefits).
- Monitoring and informing users of changes to relevant areas of Web space.
- "Schooling" network traffic into localised neighbourhoods through having effected mirroring, archiving or caching.
- Multi-functional robots can perform a number of the above tasks, perhaps simultaneously.

Robot Ethics

All ethical spider users should balance the advantages gained against the disadvantages that may arise from the use of a Web crawler. There are some generally accepted guidelines for producing an ethical crawler [3]. These guidelines, known as the Robot Exclusion protocol (REP) are designed to minimise problems for users, servers and networks. It is also considered ethical to use a spider that will parse and obey robot exclusion protocols. Robot exclusion refers to methods that can be implemented server-side to direct robot accesses.

Disadvantages of Web Robots

Network Performance

Robots traditionally have a bad press in discussions on bandwidth, even though the functions of some well-written and ethical robots are ultimately to conserve bandwidth.

There are points to consider on the bandwidth front, since robots can span relatively large portions of Web-space over short periods. Bottlenecks can arise locally though high bandwidth consumption, particularly if the robot is in frequent or permanent use, or if it is used during network peak times. The problem is exacerbated if the frequency of requests for resources is unregulated.

Server-side Concerns

So-called "rapid-fire" requests (successive HTTP requests to a single server without delays) have been shown to be very resource consuming for a server under current HTTP implementations (in fact, this is the basis of several "denial of service" attacks). Here again, an unregulated robot can cause problems. Suitable delays and an ethical traversal algorithm can help resolve this.

The skewing of server logs is another issue that causes concern. A robot that indexes an entire site will distort logs if a recognised "user-agent" is not supplied. These may be hard to distinguish from regular users.

Unethical Robots

A small number of rogue robots are in use. The tasks of these robots are such that they are particularly unwelcome by servers. Such tasks include email culling, for the production of large lists of email addresses that can be sold to advertisers and copyright violation through copying entire sites.

Additionally robots can contribute to a site "hit quota" and consume bandwidth which the site may pay for.

4 The WebWatch Robot

Robot Software

The primary software concern was for a suitable robot that could satisfy the WebWatch robot requirements. Since the majority of available robots were optimised for indexing, most did not satisfy our criteria.

Initially, the Harvest gatherer [4] was used as the WebWatch robot. The gatherer is the robot component of the Harvest indexing suite of programs. The gatherer identifies resources based on configurable file naming heuristics (generally file suffixes) and on the Unix file type determining utility. When a resource is recognised, it may be summarised by a type-specific summariser. Such a summariser will take, as input, the appropriate resource (e.g. an HTML document) and output a summary in SOIF (Summary Object Interchange Format). Thus when a robot crawl is completed, the end product will be a file of SOIF summaries for the encountered resources.

The gatherer may be configured to crawl depth-first or breadth-first and can ignore resources based upon type or upon a regular expression matched against each URL. Generally, the gatherer crawls through the initial server (although it can be configured to crawl any number of additional servers found from links). Default constraints on robot navigation (such as maximum depth and maximum URLs) can also be altered.

After the initial WebWatch crawl, we observed that the gatherer had a number of deficiencies with respect to our evolving robot criteria, they were:

- No resource identification by HTTP MIME types
- Non-configurable link extraction from HTML documents
- No differentiation between inline objects and hyperlinked objects
- The long-winded nature of altering the gatherer 'C' source code

Although we attempted to resolve some of the problems by altering the gatherer code, this quickly proved to be a long-winded process. Consequently, the next version of the robot was custom-written in Perl. The Perl based robot has gone through three different versions as it became optimised for our purposes. The Perl robots operate in a similar fashion to the gatherer described above, but with the identified deficiencies corrected. The output is still SOIF.

Recently, a Perl implementation of the Harvest gatherer has been released called Harvest-NG. Based on informal communications with the Harvest-NG developers and reading Harvest-NG documentation, use of Harvest-NG should be considered for future WebWatch-type work.

Processing Software

From the SOIF summary, each SOIF template represents a single resource. It contains resource-specific information, for example, an HTML summary contains details on all the HTML elements and attributes used. Some information is common to all resources, such as collection time and the HTTP headers associated with it.

Each SOIF template is mapped onto a command separated variable (CSV) record. All such records can then be loaded into a spreadsheet (we have been using a mixture of Microsoft Excel and SPSS for Windows) for statistical analysis.

Availability of the WebWatch Software

The WebWatch robot software will be made freely available for non-commercial use. It should be noted, however, that installation and use of the software will require Unix expertise. In addition we are unable to provide any support in use of the software.

Further information about the availability of the software is available at
<URL: `http://www.ukoln.ac.uk/web-focus/webwatch/software/`>

5 WebWatch Trawls

A summary of the main trawls carried out by WebWatch is given below.

The first trawl took place on 15 October 1997. The trawl covered **UK Public Library Web Sites**. This trawl made use of the first version of the WebWatch robot software. A report of the trawl was published in the *LA Record* [5]. For further information see <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/articles/la-record-dec1997/`>.

The second trawl took place on 24 October 1997. The trawl covered **UK University Entry Points**. This trawl also made use of the first version of the WebWatch robot software. A report of the trawl was published in the Web version of *Ariadne*, issue 12 [6].

A report on the use of the robot exclusion protocol was also produced based on this trawl in the Web version of *Ariadne*, issue 15 [7].

The second trawl took place in November 1997. The trawl covered **eLib Project Web Sites**. This trawl also made use of the first version of the WebWatch robot software. A report of the trawl was published on the UKOLN Web site [8].

During the analysis of the results obtained during the first two trawls it became obvious that the Harvest robot software had limitations in its use as an auditing robot. For example, since Harvest was originally developed for indexing purposes, it did not retrieve binary resources, such as images. Also it did not store the MIME type of the resources it encountered. Both of these limitations made it difficult to extend the range of analyses which WebWatch intended to carry out.

After the date of the second trawl, the WebWatch project began to develop its own robot, which overcome these limitations.

The third major trawl took place in May 1998. The trawl covered **UK Academic Library Web Sites**. A report of the trawl was published on the UKOLN Web site [9].

The fourth major trawl took place in July 1998. The trawl covered **UK University Entry Points**. A report of the trawl, together with a comparison of the first trawl of UK University entry points will published in the *Journal of Documentation* [10].

The fifth major trawl took place in December 1998. The trawl covered **UK University Entry Points**. A report of the trawl, together with a comparison of the first and second trawls of UK University entry points was published on the UKOLN Web site [11].

In addition to these major trawls, a number of additional smaller trawls were carried out, including:

> **Analysis of University of Wolverhampton**: A trawl was carried out of the University of Wolverhampton Web site. The trawl was carried out in February 1998. Unfortunately due to software problems the trawl was incomplete and a report was not produced.
>
> **Analysis of Napier University**: A trawl was carried out of the Napier University Web site. The trawl was carried out in February 1998. A report on the trawl was made available to Napier University.
>
> **Link analysis for De Montfort University**: A trawl was carried out of the De Montfort University Web site and the hyperlinks within the Web site were analysed. The trawl was carried out in July 1998. The aim of the analysis was to investigate the possibility of using WebWatch as a tool for institutions to analyse links from their Web site, especially to US resources.
>
> **Analysis of University entry points in the south west**: A trawl was carried out of University entry points in the south west of England (including Bath, Bristol, UWE) in September 1998. A report was given at a meeting of South West Web Managers.

6 Observations

During the analysis of the results of the trawls, the WebWatch project made a number of observations which will be of interest to users of the Web. These are listed below.

For Information Providers

Directory Structure

During the analysis of UK Public Library Web sites in particular, it was noted that a number of Web sites used a very flat directory structure, with the library departmental menu page located at the root directory of the Web, along with other departmental menu pages. This can make it difficult for an automated tool to limit itself to the relevant area. This will make it difficult to restrict the coverage of robot software, including indexing and auditing tools.

Recommendation: The menu page for departments should be located beneath the departmental directory. For example, `http://www.foo.ac.uk/library/library.html` and not `http://www.foo.ac.uk/library.html`

Broken Links

During all of the trawls broken links were found.

Recommendation: Institutions should systematically run link checkers on their Web sites, and define procedures for fixing broken links.

Metadata

A relatively low proportion of metadata was found, even on key Web pages, such as institutional and departmental entry points. Although there may be problems with widespread deployment of metadata (difficulties of maintaining the metadata, lack of tools to create, edit and validate the metadata, volatility of metadata standards, etc.) the deployment of a small amount of metadata using stable conventions (primarily "AltaVista" metadata) will improve the quality of resource discovery.

Recommendation: Institutions should deploy "AltaVista" type metadata on a small number of key entry points.

HTML Conformance

Web pages were found which failed to validate against published HTML DTDs (Document Type Definitions). Although browsers tend to be tolerant of HTML errors, other user agents, such as indexing and auditing robots, may have problems in processing such pages.

Recommendation: Institutions should develop and implement procedures for ensuring that HTML pages conform to a published DTD.

Use of Frames

The initial version of the WebWatch robot, which used the Harvest software, could not process Web sites which made use of frames. Although the latest version of the WebWatch robot can handle Web sites which make use of frames, many indexing robots cannot. The use of frames may mean that indexing robots will not index such Web sites.

Recommendation: Information providers should be aware of the problems that frames have on robot software, and should either avoid use of frames, or make use of techniques to overcome this problem.

Size of Institutional Entry Points

During the trawl of UK University entry points the largest entry points were found to contain animated GIFs. The use of animated GIFs not only results in large file sizes, they can also cause accessibility problems.

Recommendation: Institutions should avoid be aware of the dangers in use of animated GIFs.

Use of "Splash Screens"

During the trawl of UK University entry points a small number of pages were found to contain "splash screens". Splash screens may cause accessibility problems. In addition, although modern browsers support them, older browsers and other user agents may not.

Recommendation: Institutions should avoid be aware of the dangers in use of "splash screens".

Use of Java and JavaScript to Provide Scrolling Text

During the trawl of UK University entry points a small number of pages were found to make use of Java and JavaScript. Subsequent manual analysis showed that in some cases Java and JavaScript were used to provide scrolling text. Although this may be of use in certain circumstances, scrolling text can cause accessibility problems.

Recommendation: Institutions should be aware of the dangers in use of Java or JavaScript to provide scrolling text.

Use of Hyperlinks

The trawl of UK University entry points indicates differences in approaches to the provision of hyperlinks on the main institutional entry point. Increasingly it appears that organisations are reducing the number of hyperlinks on key pages.

Recommendation: Institutions should be aware of the dangers in providing too many hyperlinks.

Use of Hyperlinks to Remote Resources

With the introduction of charging for network traffic from the US, increasing numbers of Universities are reducing the numbers of links to US resources.

Recommendation: Institutions should provide hyperlinks to UK resources where possible.

For Webmasters

System Configuration

Web sites were found with misconfigured server configuration files. For example, on several Web sites image files were configured with a MIME type of `text/html`. Although this has no noticeable affect on commonly used browsers, it can cause problems with robot software (for example, indexing software may index the contents of the binary file).

Recommendation: Webmasters should ensure that server configuration files are configured correctly.

Server Software

In the initial survey of UK University entry points, several unusual Web server software packages were found, such as Microsoft's Personal Web Server. In the second trawl the use of such software had declined.

Recommendation: Webmasters should take note of the profile of server software used within their community, and be aware of the implications of running unusual software (such as possible lack of support and expertise).

HTTP/1.1

Analysis of the trawls of UK University entry points indicate that about 50% use servers which support HTTP/1.1 HTTP/1.1 has many performance benefits for the server and the network over HTTP/1.0.

Recommendation: Webmasters should consider deploying server software which implements HTTP/1.1

The `robots.txt` File

Many Web sites do not have a `robots.txt` file. A `robots.txt` file can improve the server performance by ensuring that certain areas of the Web site are not indexed. It can improve the quality of the contents of search engines by ensuring that areas of the Web site containing poor quality information are not indexed.

Recommendation: Webmasters should create a `robots.txt` file and configure it so that robots do not access resources unnecessarily.

Monitoring Robots

The existence of unethical robots may present a problem to the administrator of a Web site. Webmasters should monitor their server logs in order to identify unethical robots.

Recommendation: Webmasters should consider installing software to report on visits by robots.

URL Conventions

The trawl of eLib project Web sites indicated that several projects used URLs for their main entry point which were either long or made use of personal home directories (e.g. the ~*name* convention). There is a danger that such projects will be renamed with a more persistent URL, and that access to the original URL could be discontinued.

Recommendation: URLs for projects which are likely to have a long life-span should aim to be persistent.

For Robot Developers

Writing Robot Software

Implementers of robots should follow the guidelines on robot ethics.

Recommendation: Robot developers should reuse existing spiders rather than creating new ones.
Robots should identify themselves to servers.
Robots should be thoroughly tested locally before being unleashed on the Web.
Robots should retrieve only resources that they are interested in.
The results of using a robot should be disseminated

Memory Leakage

Implementers of robots should be aware of memory leakage problems in versions of the LWP module and Perl itself.

Recommendation: Robot developers should be aware of the possible need for a controller process to kill and re-start the robot on large trawls.

Restarting Trawls

There may be a need to restart trawls if memory leakages cause a trawl to be abandoned.

Recommendation: A history file should be built up for the robot's use, that can allow the robot to stop and to later re-start at that point. Such a file should include the canonicalised URL for a resource and a content-based checksum (such as MD5) and also some resource metadata, such as an expiry date. An error log file is also very useful.

Trawling Guidelines

Implementers of robots should be aware of memory leakage problems in versions of the LWP module and Perl itself.

Recommendation: Hierarchical crawling can be useful where collections within Web sites are grouped by directory. After directory and DNS canonicalisation, each URL should be checked against that in the history file and HTTP headers such as "`Expires`" should be looked for. There should also be default limits on total number of URLs encountered and on depth. These are used to prevent useless crawling of a "black hole". Stop-lists based on resource type and URL regular expressions can also be used to prevent unnecessary trawling.

Registration of Robot

Implementers of robots should register their robot.

Recommendation: Robot developers should use the form at <URL: `http://www.botwatch.com/addbots/`> to register their robot.

User-Agent Negotiation

Web servers which make use of server-side scripting may choose to serve different content to "unusual" user-agents (i.e. user-agents other than Netscape and Internet Explorer).

Recommendation: Robot developers should be aware that Web servers may choose to serve different content to "unusual" user-agents.

For Protocol Developers

Web Collections

Robot indexing and auditing software developers require more sophisticated mechanisms for defining collections of related resources than is currently provided.

Recommendation: Protocol developers should develop site mapping protocols which will enable information providers and Webmasters to define areas which robot software should access.

Robot Exclusion Protocol

A more sophisticated robot exclusion protocol is required which (a) can be managed by information providers, without requiring the intervention of a Webmaster and (b) provide greater control.

Recommendation: Protocol developers should develop a more sophisticated robot exclusion protocol.

7 WebWatch Web Services

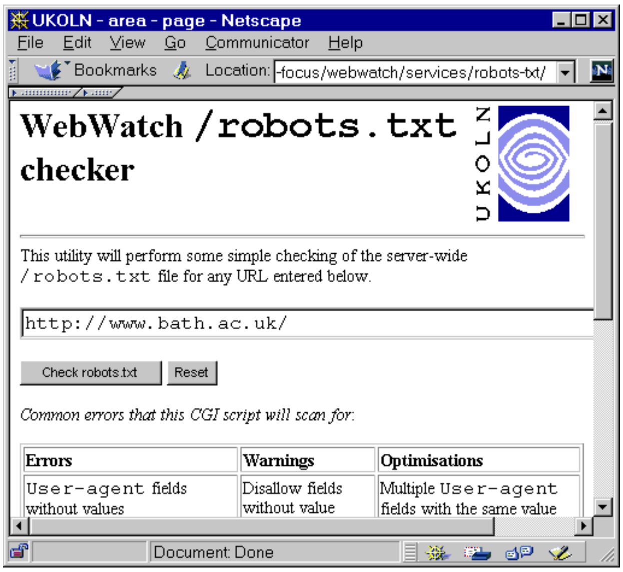
The WebWatch project developed a set of Unix-based tools to support its work. In order to provide easy access to these tools, a Web interface was developed. The Web interface has been enhanced and these Web services are now freely available. A summary of the services is given below.

The `robots.txt` Checker

The `robots.txt` checker is a Web-based service which tests for the existence of a `robots.txt` file on a web server, and runs some simple checks for common errors in configuring the file.

The `robots.txt` checker is available at <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/services/robots-txt/`>.

The `robots.txt` checker is illustrated in Figure 7-1.



WebWatch /robots.txt checker

This utility will perform some simple checking of the server-wide `/robots.txt` file for any URL entered below.

`http://www.bath.ac.uk/`

Common errors that this CGI script will scan for:

Errors	Warnings	Optimisations
`User-agent` fields without values	Disallow fields without value	Multiple `User-agent` fields with the same value

Figure 7-1 The `/robots.txt` Checker

A typical `robots.txt` file is illustrated in Figure 7-2.

```
# robots.txt file

User-agent: CharlieSpider
Disallow: /

User-agent: *
Disallow: /UCS/mail-archives/
Disallow: /cgi-bin/
```

Figure 7-2 A Typical `robots.txt` File

In this example the `CharlieSpider` robot (referred to as a *user-agent* in the file) is prohibited from trawling the web site. All robots are prohibited from accessing files storing in and beneath the directories `/cgi-bin` and `/ucs/mail-archives`.

The `CharlieSpider` robot is probably banned because it is known to be *unethical*. For example, it could download email addresses for spamming purposes, or it could simply place unnecessary load on the web server due to inefficiencies in the design of the program.

Robots are expected not to access the `/cgi-bin` directory as this directory contains software and not documents, and so it would not be sensible for the contents of the directory to be indexed.

Robots are expected not to access the `/ucs/mail-archives` as this directory contains archives of mail messages which the institution does not want to be indexed by search engines, such as AltaVista.

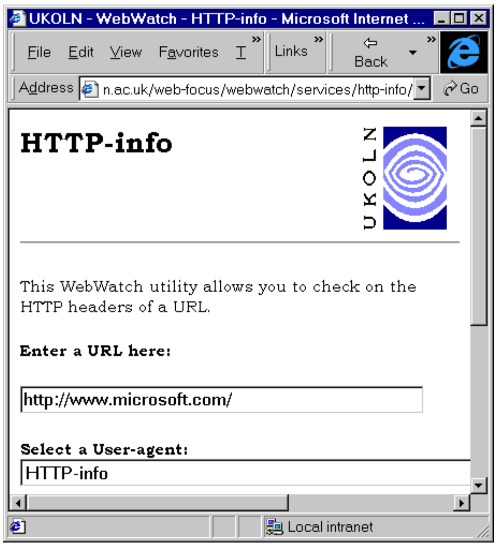
The `robots.txt` checker service is a useful tool for information providers who wish to check if their web server contains a `robots.txt` file, and for webmasters who wish to check their `robots.txt` file and compare it with others.

The HTTP-info Service

HTTP-info is a Web-based service which displays the HTTP headers associated with a Web resource.

HTTP-info is available at
<URL: http://www.ukoln.ac.uk/web-focus/webwatch/services/http-info/>.

The HTTP-info interface is illustrated in Figure 7-3a. Figure 7-3b illustrates the output from the service.



Address: n.ac.uk/web-focus/webwatch/services/http-info/

HTTP-info

This WebWatch utility allows you to check on the HTTP headers of a URL.

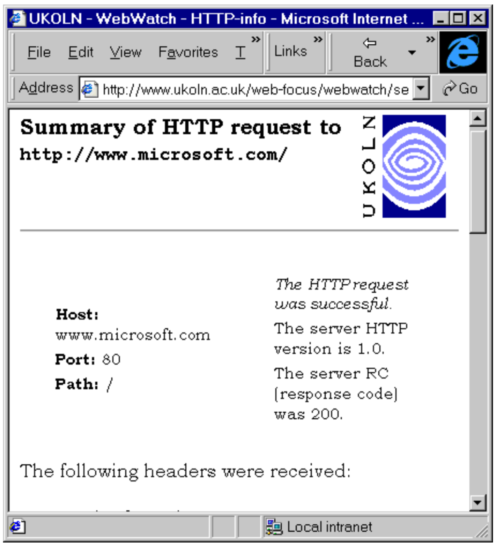
Enter a URL here:

http://www.microsoft.com/

Select a User-agent:

HTTP-info

Figure 7-3a HTTP-info Interface



Address: http://www.ukoln.ac.uk/web-focus/webwatch/se

Summary of HTTP request to http://www.microsoft.com/

Host: www.microsoft.com
Port: 80
Path: /

The HTTP request was successful.
The server HTTP version is 1.0.
The server RC (response code) was 200.

The following headers were received:

Figure 7-3b Output From HTTP-info

In Figure 7-3b the headers for the resource at <URL: http://www.microsoft.com/> were displayed using HTTP-info. The output shows the host name, port and the path. The server is using HTTP/1.0. The return code (RC) of 200 shows that the document was successfully retrieved. The HTTP headers which were retrieved (which are not all shown in the diagram) show that, for example:

- The server using the ETag method for cache control.
- The document was last modified a day ago.
- The server software is, not surprisingly, Microsoft-IIS/4.0.

HTTP-Info is a useful tool for information providers who wish to check the HTTP headers for their web resources, for end users who wish to check the HTTP headers of services they are accessing and for webmasters who may wish to check server configuration options.

The functionality provided by HTTP-Info can be obtained by simple use of the telnet command. However HTTP-Info provides a simplified interface, for those who aren't familiar with or have access to telnet.

The Doc-info Service

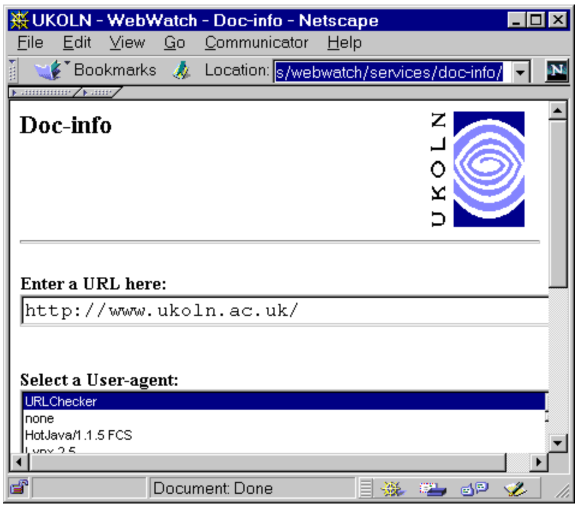
Doc-info is a Web-based service which displays various information about an HTML resource.

Doc-info is available at
<URL: http://www.ukoln.ac.uk/web-focus/webwatch/services/doc-info/>.

The Doc-info interface is illustrated in Figure 7-4a. Figure 7-4b illustrates the output from the service.

Information provided by Doc-Info about HTML resources includes:

- The names of embedded resources (e.g. images).
- The size of embedded resources.
- The total size of the resource.
- Details of links from the resource.
- A summary of the HTML elements in the resource.
- Details of the web server software.
- Details of cache headers.


UKOLN - WebWatch - Doc-info - Netscape
File Edit View Go Communicator Help
Bookmarks Location: s/webwatch/services/doc-info/
Document: Done

Doc-info

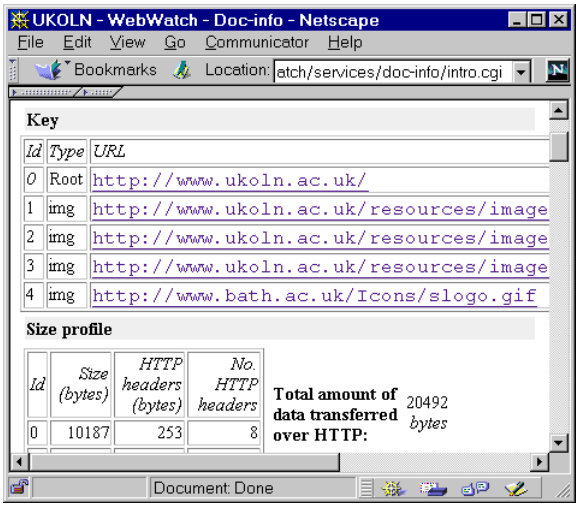
Enter a URL here:

http://www.ukoln.ac.uk/

Select a User-agent:

URLChecker
none
HotJava/1.1.5 FCS

Figure 7-4a The Doc-Info Interface


UKOLN - WebWatch - Doc-info - Netscape
File Edit View Go Communicator Help
Bookmarks Location: atch/services/doc-info/intro.cgi
Document: Done

Key

Id	*Type*	*URL*
0	Root	http://www.ukoln.ac.uk/
1	img	http://www.ukoln.ac.uk/resources/image
2	img	http://www.ukoln.ac.uk/resources/image
3	img	http://www.ukoln.ac.uk/resources/image
4	img	http://www.bath.ac.uk/Icons/slogo.gif

Size profile

Id	*Size (bytes)*	*HTTP headers (bytes)*	*No. HTTP headers*
0	10187	253	8

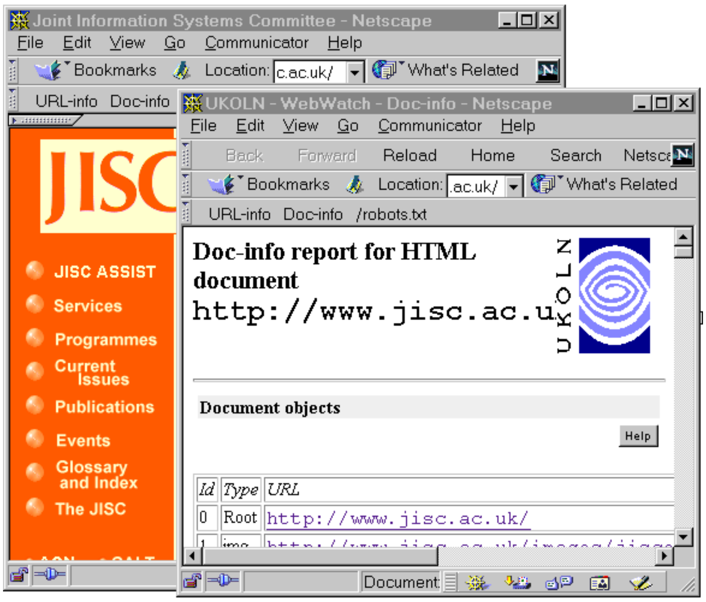
Total amount of data transferred over HTTP: 20492 *bytes*

Figure 7-4b Output From Doc-Info

Using WebWatch Services From A Browser Toolbar

Although the WebWatch services provide a simple user interface for getting information about web resources, the need to go to the page, either by entering the URL or using a bookmark, and then enter the URL of the resource to be analysed can be a barrier to use of the services. It is desirable to access the services directly when viewing a page.

Use of the Netscape personal toolbar enables the services to be accessed directly. A link to the underlying CGI script for the WebWatch services can be dragged onto the toolbar. Then when an arbitrary page is being viewed, the option in the personal toolbar can be selected. The WebWatch service will analyse the page being viewed. The output from the WebWatch service will be displayed in a separate browser window, as shown in Figure 7-5.


Joint Information Systems Committee - Netscape
File Edit View Go Communicator Help
Bookmarks Location: c.ac.uk/ What's Related
URL-info Doc-info
JISC
JISC ASSIST
Services
Programmes
Current Issues
Publications
Events
Glossary and Index
The JISC
UKOLN - WebWatch - Doc-info - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netsc
Bookmarks Location: ac.uk/ What's Related
URL-info Doc-info /robots.txt
Help
Document

Doc-info report for HTML document http://www.jisc.ac.u

Document objects

Id	*Type*	*URL*
0	Root	http://www.jisc.ac.uk/

Figure 7-5 Accessing Doc-Info from the Netscape Personal Toolbar

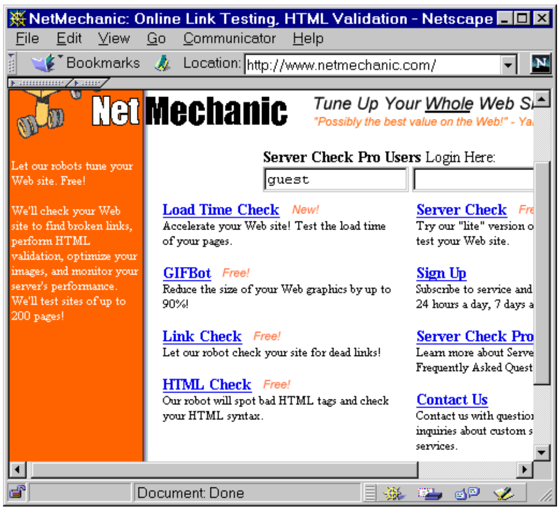
The list of WebWatch services at <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/services/`> contains details on how to include the services in the Netscape personal toolbar.

8 Future Developments

This section describes options for future developments of a community-based Web-monitoring and auditing system such as WebWatch.

The Business Case

The WebWatch project has provided useful information on the development of Web sites within a number of UK communities. An indication of the usefulness of a project such as WebWatch can be gauged by observing the development of commercial services which have a similar function to Web Watch.


NetMechanic: Online Link Testing, HTML Validation - Netscape
File Edit View Go Communicator Help
Bookmarks
Location: http://www.netmechanic.com/
Document: Done

NetMechanic

Tune Up Your Whole Web Si

"Possibly the best value on the Web!" - Ya

Let our robots tune your Web site. Free!

We'll check your Web site to find broken links, perform HTML validation, optimize your images, and monitor your server's performance. We'll test sites of up to 200 pages!

Server Check Pro Users Login Here:

guest

Load Time Check New!
Accelerate your Web site! Test the load time of your pages.

GIFBot Free!
Reduce the size of your Web graphics by up to 90%!

Link Check Free!
Let our robot check your site for dead links!

HTML Check Free!
Our robot will spot bad HTML tags and check your HTML syntax.

Server Check Fre
Try our "lite" version o
test your Web site.

Sign Up
Subscribe to service and
24 hours a day, 7 days a

Server Check Pro
Learn more about Serve
Frequently Asked Quest

Contact Us
Contact us with questio
inquiries about custom s
services.

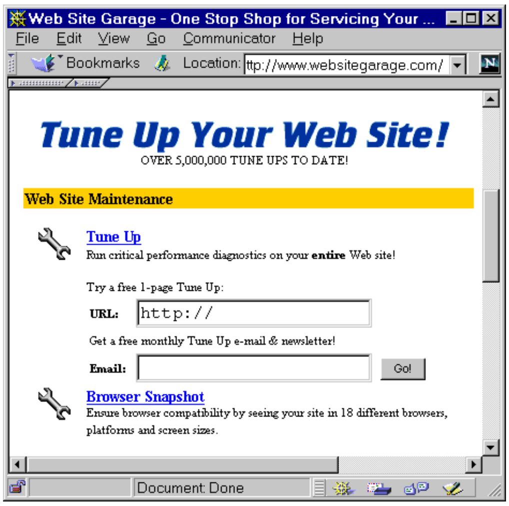
Figure 8-1 NetMechanic Home Page

NetMechanic (their home page is shown in Figure 8-1) provide a free service for reporting on single Web pages. A number of commercial packages are also available which currently cost between $24 per month and $399 per year for reporting on a single Web page.

NetMechanic is available at the address <URL: `http://www.netmechanic.com/`>

WebSiteGarage (their home page is shown in Figure 8-2) also provide a free service for reporting on single Web pages. A number of commercial packages are also available which currently cost between $24 per month and $399 per year for reporting on a single Web page.

WebSiteGarage is available at the address <URL: `http://www.websitegarage.com/`>


Web Site Garage - One Stop Shop for Servicing Your ...
File Edit View Go Communicator Help
Bookmarks
Location: ttp://www.websitegarage.com/
Go!
Document: Done

Tune Up Your Web Site!

OVER 5,000,000 TUNE UPS TO DATE!

Web Site Maintenance

Tune Up
Run critical performance diagnostics on your **entire** Web site!

Try a free 1-page Tune Up:

URL: http://

Get a free monthly Tune Up e-mail & newsletter!

Email:

Browser Snapshot
Ensure browser compatibility by seeing your site in 18 different browsers, platforms and screen sizes.

Figure 8-2 WebSiteGarageHome Page

Technologies

Possible technological developments for a WebWatch service are given below.

Database Access to WebWatch Data

One of the main limitations to development of the WebWatch project is the need to develop *ad hoc* Unix scripts in order to process the data. In addition, analysis of the data can only be carried out by staff involved with the WebWatch project. It would be useful to store the WebWatch data in a backend database and to provide access to the data using a Web browser.

Evaluation of Web Site Accessibility

The WebWatch software has a possible role in the evaluation of the accessibility of a Web site. For example, WebWatch could report on the occurrence of HTML features which cause accessibility problems (e.g. missing `ALT` attributes on `<IMG>` elements). The WebWatch project could work in conjunction with projects such as

DISinHE, who have a remit to support the use of accessible Web sites and the W3C WAI (Web Accessibility Initiative).

Note that the WebWatch software is included in the Web Accessibility Initiative list of tools at <URL: `http://www.w3.org/WAI/ER/existingtools.html`>.

Monitoring SSL Deployment

WebWatch could be used to monitor deployment of SSL. SSL (Secure Sockets Layer) is used to provide access to secure Web services. Note that a commercial SSL server survey is also available from Netcraft, at a price of £1,200 for a monthly updated analysis reflecting the topology of encrypted transactions electronic commerce on the Internet - see <URL: `http://www.netcraft.com/ssl/` >.

Use of Harvest/NG

During the development of the WebWatch robot links were made with the developers of the Harvest software. The WebWatch project team have provided input into the future development of Harvest, which is known as Harvest/NG. It is now recommended that future WebWatch work should make use of the Harvest/NG software.

Development of a Cachability Checking Service

The WebWatch project has developed a Web-based HTTP analysis service. In order to enable information providers to check whether a Web resource can be cached, it would be useful to develop a Cachability checking service. Although this can be done very easily for HTTP/1.0 servers, for HTTP/1.1 servers it would be necessary to develop a more sophisticated service, it order to check the status of the `Etag` field.

Configurable User-Agent Field

Websites are beginning to provide different content depending on the user-agent (browser) which accesses them. It would be useful if it was possible to configure the WebWatch robot software so that it could provide reports on services which make use of user-agent negotiation.

Communities

Regular WebWatch trawls of communities such as the main entry points for higher education and other public sector services would be valuable. This would enable trends such as the use of server software (as monitored globally by Netcraft which is shown at <URL: `http://www.netcraft.com/Survey/`>) to be carried out.

A comparison of UK communities with other communities (e.g. UK University entry points with US University entry points, UK University Web sites with commercial Web sites) would also be useful.

9 References

1 *Web Robots Page*,
<URL: `http://info.webcrawler.com/mak/projects/robots/robots.html`>

2 *HTTP/NG Web Characterisation Group*, W3C <URL: `http://www.w3.org/WCA/`>

3 *Robots Exclusion*, <URL:
`http://info.webcrawler.com/mak/projects/robots/exclusion.html`>

4 *Harvest*, <URL: `http://harvest.transarc.com/` >

5 *Robot Seeks Public Library Websites*, LA Record, December 1997 Vol. 99(12). Also at <URL:
`http://www.ukoln.ac.uk/web-focus/webwatch/articles/la-record-dec1997/`>.

6 *WebWatching UK Universities and Colleges*, Ariadne issue 12 (Web version),
<URL: `http://www.ariadne.ac.uk/issue12/web-focus/`>.

7 *Showing Robots the Door*, Ariadne issue 15 (Web version),
<URL: `http://www.ariadne.ac.uk/issue15/robots/`>.

8 *Report of WebWatch Crawl of eLib Web Sites*, UKOLN,
<URL: `http://www.ukoln.ac.uk/web-focus/webwatch/reports/elib-nov1997/`>.

9 *A Survey of UK Academic Library Web Sites, UKOLN,*
<URL: `http://www.ukoln.ac.uk/web-focus/webwatch/reports/hei-lib-may1998/` >

10 *How Is My Web Community Doing? Monitoring Trends in Web Service Provision,* Journal of Documentation, Vol. 55 No. 1, January 1999.

11 *Third Trawl of UK Academic Entry Points*, UKOLN,
<URL: `http://www.ukoln.ac.uk/web-focus/webwatch/reports/hei-nov1998/`>.

Appendix 1 Trawl of UK Public Libraries

This appendix is based on an article originally published in the December 1997 Vol. 99 (12) edition of *LA Record*. See <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/articles/la-record-dec1997/`>. We are grateful to the *LA Record* for granting permission to republish this article

This trawl took place on 15 October 1997.

Robot Seeks Public Library Websites

UKOLN has been active recently analysing UK Public Library websites. **Brian Kelly**, **Sarah Ormes** and **Ian Peacock** report on their findings.

Introduction

If you have eaten a packet of crisps, read a newspaper or watched a TV commercial over the last week you have probably, at some point, come across the increasingly ubiquitous `http://`... of a World Wide Web address. At times it seems that having a website is as essential as having a phone number.

Public libraries in the UK have also been following this trend. In late 1995 the '*Library and Information Commission public library Internet survey*' [1] showed that already 25% of library authorities were providing some sort of information over the Internet. By the time of the publication of '*New Library: The People's Network*' [2] Sheila and Robert Harden's '*UK Public Libraries*' web page [3] had links to about 90 public library websites.

Whereas many company websites are little more than hi-tech adverts for their products a website offers a public library a real opportunity for enhancing current services and developing new ones. Libraries in Denmark, the UK and America, for example, are already using the Web to provide networked services to their readers - whether this means logging in to the library OPAC at home, emailing a reference query, viewing digitised images from the local history collection or even finding out where the local allotments are.

These type of networked developments were highlighted in the '*New Library: The People's Network*' report as being essential components of the public library of the future. Public library authority websites will be the gateways though which an increasing number of people will use their library services. Considering the importance these websites could play we know very little about public library authority websites in the UK. We know roughly how many there are but other statistics are difficult if not impossible to find.

Although we are all familiar with the quotation 'there are lies, damned lies and statistics' statistics can be useful, enlightening and essential. We now need statistics about public library authority websites which will give an indication of the strengths, preparedness and shortfalls of current sites so we can begin to develop the networked library services of the future.

The *WebWatch* project is funded by the British Library Research and Innovation Centre (BLRIC). The main aim of WebWatch, which is based at UKOLN (UK Office for Library and Information Networking), is to develop and use robot software for analysing various aspects of the World Wide Web within a number of UK communities, such as academic institutions and public libraries.

The work plan for WebWatch includes:

- Evaluation of robot technologies and recommendations on the technologies to be used for the project.
- Identification of relevant communities.
- Running several WebWatch trawls of web resources within the communities.
- Analysis of the results obtained, and liaising with the relevant communities in interpreting the analyses and making recommendations.
- Working with the international web robot communities.
- Analysing other related resources, such as server log files.

Analysis of UK Public Library Websites

Background

The publication of the '*New Library: The People's Network*' report and preparatory work for the National Libraries Week coincided with the plans for the initial run of the WebWatch robot. Following discussions within UKOLN and with representatives of the UK public library community it was agreed that the initial launch of the WebWatch robot would cover websites for UK public libraries.

Following some initial test runs, the WebWatch robot trawled UK public libraries' websites on the evening of Wednesday, 15th October 1997 – the day of the launch of the *New Library* report. The robot took its list of public library websites from the Harden's *UK Public Libraries* web page.

Methodology and Analysis

The WebWatch robot is based on the Harvest software [4]. Various modifications to the software were made in order to tailor the software for auditing and monitoring purposes.

Developments are still being made as we gain experience in using the software.

The Harden list of UK Public Library websites has 90 entries. The WebWatch robot successfully trawled 64 sites. Eight of these sites which could not be accessed were hosted by the NIAA (the Northern Informatics Applications Agency) which was in the process of upgrading its web server while the WebWatch robot was in use.

As can be seen from Figure A1-1, the majority of sites contain a small number of pages, with a median of 24 pages. Thirteen sites contained over 100 pages, with only four sites containing over 300 pages.

A manual analysis of some of the large websites indicated that the robot had analysed non-library pages, such as information on museums, leisure services, etc.

Figure A1-2 shows the number of (inline) images on public library websites. Again we can see that most websites used a small number of images, and that the websites containing large numbers are probably distorted by the analysis of non-library resources.

Figure A1-3 shows the total size of public library websites. Again we can see that most of the websites are small, with a median value of 190 Kbytes. The mean value of 730 Kbytes is again likely to be skewed by the analysis of whole council websites.

In addition to analysing the numbers and sizes of the websites, we also analysed the domain names. We were interested in whether public libraries used their own domain name (such as `www.ambridge.org.uk`) or if they simply rented space from an Internet Service Provider.

Issues

What Is A Public Library?

The WebWatch robot took its input data from a list of Public Library websites. The total number of websites is taken from this list. However in some cases these may refer to Public Library Authorities.

Defining A Public Library Website

The WebWatch robot analysed resources located beneath the directory defined in the input data. In one case the library website had its own domain name (e.g. `http://www.lib.ambridge.org.uk/`). In most other cases the library stored its resources under its own directory (e.g. `http://www.ambridge.org.uk/library/`) with other services on the domain having their own directory name (e.g. `http://www.ambridge.org.uk/leisure-services/`). In both of these cases the robot has knowledge of the extent of the library website service and can audit the resources correctly.

In some cases, however, the main library entry point was located in the same directory as other resources (e.g. `http://www.ambridge.org.uk/library.html` and `http://www.ambridge.org.uk/leisure.html`). This case is more difficult to process automatically.

In the analysis the figures for the sizes of a number of public library websites are likely to be inflated by the robot indexing non-library pages.

The Way Forward

WebWatch Developments

The initial run of the WebWatch robot was timely as it coincided with the launch of the '*New Libraries*' report. However the robot is still in its infancy, and we intend to implement a number of new facilities. Our plans include:

Additional trawls of public libraries: To provide greater and more detailed coverage of public library websites.

Analysis of header information: This will enable us to determine when HTML pages were last updated.

Analysis of HTML elements: This will enable us to monitor usage of HTML elements (such as tables, frames, etc.) and technologies such as Java. This can be important in ensuring that web resources are accessible to large numbers of people, and not just to those running the latest versions of web browsers.

Analysis of quality and conformance: This will enable us to monitor conformance to HTML standards, HTTP errors (which can indicate broken links, misconfigured servers, etc).

More refined classification of resources: This will address the issue of the robot accessing non-library resources.

In addition to developments to the robot software, we will be looking to analyse server log files. Server log files provide useful statistics, including details of the browser (such as browser name, version number and the platform and operating system on which the browser is being used) used to access web resources.

Conclusions

The WebWatch analysis of the UK public library websites took place at a timely moment for the library community. It provided a snapshot of the community on the day of the launch of the *New Libraries* report.

The analysis indicated that, with a small number of exceptions, most public library websites consist of a small number of pages – it seems likely that the majority of public library websites would fit comfortably on a floppy disk! Although this is still early days for public libraries on the web, it is pleasing to note that our involvement with the public library community shows that a number of public libraries are developing comprehensive websites.

The analysis also indicated that refinements were needed to the robot, in particular to the definition of sites to be analysed.

UKOLN looks forward to working with the public library community in further surveys.

References

[1] 'Library and Information Commission public library Internet survey', see <URL: http://www.ukoln.ac.uk/publib/lic.html>

[2] 'New Library: The People's Network', see <URL: http://www.ukoln.ac.uk/services/lic/newlibrary/>

[3] 'UK Public Libraries', see <URL: http://dspace.dial.pipex.com/town/square/ac940/ukpublib.html>

[4] 'Harvest', see <URL: http://harvest.transarc.com/>

***Brian Kelly** is UK Web Focus, a national Web coordination post funded by JISC.*

***Sarah Ormes** is UKOLN's public library networking researcher.*

***Ian Peacock** is WebWatch, and is responsible for software development and running the WebWatch robot software.*

UKOLN is funded by the British Library Research and Innovation Centre and the JISC of the Higher Education Funding Council.

Diagrams

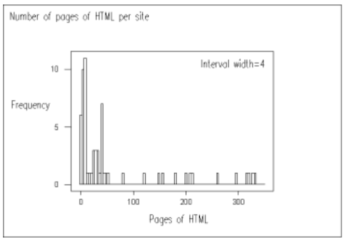


Figure A1-1 Size of Website (by number of HTML pages) Versus Frequency

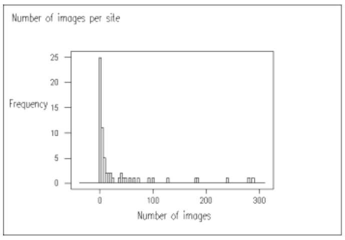


Figure A1-2 No. of Images Versus Frequency

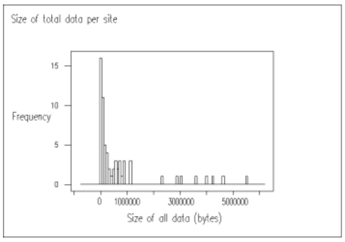


Figure A1-3 Size of Website (file size, including HTML pages, images, etc.) versus frequency

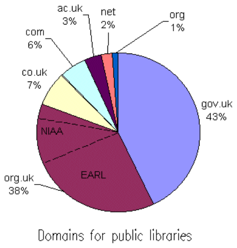


Figure A1-4 Domain Name Usage

What is a Web Robot?

A ***web robot*** is an automated software tool, which retrieves web resources.

Web robots are often used for indexing the World Wide Web. For example the large search engines, such as Alta Vista and Lycos, use a web robot for retrieving and indexing web resources.

The WebWatch robot is based on the Harvest software, which is widely used, especially in the academic community, for indexing web services.

The WebWatch robot conforms to the Robot Exclusion protocol. It will not analyse resources which have been appropriately flagged. In addition we aim to run the robot overnight and at weekends when network and server load is likely to be at a minimum.

Appendix 2 First Trawl of UK University Entry Points

This appendix is based on an article originally published in the web version of Ariadne, issue 12. See <URL: `http://www.ariadne.ac.uk/issue12/web-focus/`>.

This trawl took place on 24 October 1997.

WebWatching UK Universities and Colleges

Java, ActiveX, JavaScript, style sheets, PICS, metadata - examples of web technologies which have emerged over recent years. But how widely used are such technologies? Technologies, such as hypertext linking, images and active maps, are more widely deployed, but how are these technologies used?* Brian Kelly *describes the WebWatch project which attempts to answer the questions.

About WebWatch

WebWatch is a one year project funded by the British Library Research and Innovation Centre (BLRIC) [1]. The main aim of WebWatch is to develop and use robot software to analyse the use of web technologies with various UK communities and to report on the findings to various interested communities. Other aims of WebWatch include:

- Evaluation of robot technologies and making recommendations on appropriate technologies.
- Analysis of the results obtained, and liaising with the relevant communities in interpreting the analyses and making recommendations.
- Working with the international web robot communities.
- Analysing other related resources, such as server log files.

WebWatch Trawls

UK Public Libraries

The WebWatch robot was launched on the evening of Wednesday, 15th October 1997 - the day of the launch of the LIC's *'New Library: The People's Network*' report [2]. The robot trawled UK Public Library websites, as defined in the Harden's list [3]. The aim of this initial trawl was to audit the size of public library websites on the day of the launch of the New Library report.

UK Universities and Colleges Home Pages

The second WebWatch trawl took place on the evening of Friday 24th October. This time the robot analysed UK Higher Education Universities and Colleges home pages (i.e. the institutional entry point), as defined by the HESA list [4].

The WebWatch robot stored the following information for subsequent analysis:

HTML Elements

A count of all HTML elements used in the institutional entry page and details of element attributes. This includes:

Metadata Details
Use of the `<META>` element and of the type of metadata used.

Link Details
A count of the numbers of links and details of the destinations of the links.

Image Details
A count of the numbers of inline images and details of the `<IMG>` attribute values (e.g. `WIDTH`, `HREF`, etc).

Script Details
A count of the number of client-side scripts.

Header Information

HTTP header information, including:

Server Software
The name and version of the server software used.

File Size
The size of the institutional HTML entry page.

Modification Date
The modification date of the institutional entry page.

Figure A2-1 illustrates the raw data file.

```
Gatherer-Time{24}: Fri Oct 24 19:21:00 1997
File-Size{4}:      2323
CRC{9}:             200 (OK)
Message{3}:        OKD
Date{20}:          Fri, 24 Oct 1997 18
Server{13}:        Apache/1.1.3
...
Type{4}:           HTML
total-count{2}:    69
p-count{1}:        3
a-count{2}:        15
center-count{1}:   1
b-count{1}:        5
title-count{1}:    1
head-count{1}:     1
br-count{2}:       17
..
img-60-attrib{61}: width=13|src=../gifs/redgem.gif|height=13|alt=*|nosave=nosave
a-48-attrib{30}:   href=/www/schools/schools.html
...
```

Figure A2-1 - Portion of the Raw Data Collected by the WebWatch Robot

The first part of the data file contains the HTTP header information. The second part contains a count of all HTML elements found in the home page. The final part contains the attribute values for all HTML elements.

Analysis of UK Universities and Colleges Home Pages

A total of 164 institutions were included in the input file. The WebWatch robot successfully trawled 158 institutions. Six institutional home pages could not be accessed, due to server problems, network problems or errors in the input data file.

Page Size

The average size of the HTML page is 3.67 Kb. Figure A2-2 gives a histogram of file sizes.

It should be noted that the file sizes do not include the sizes of inline or background images. This histogram therefore does not indicate the total size of the files to be downloaded.

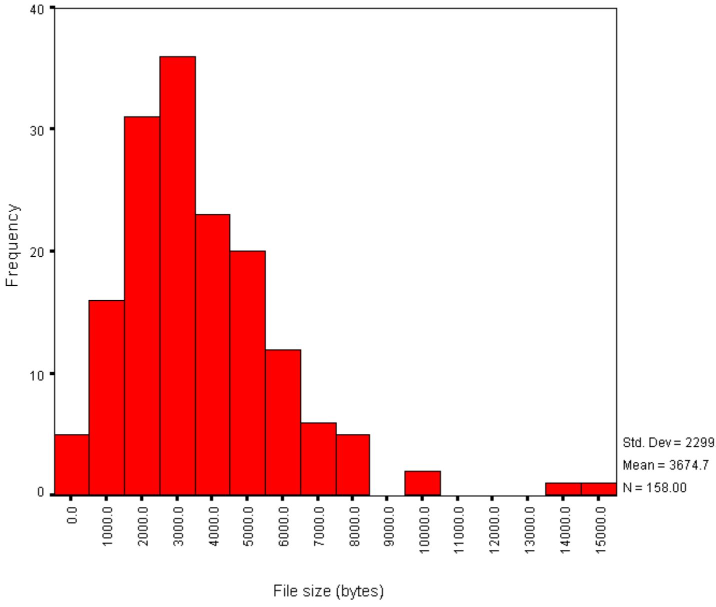


Figure A2-2 - Histogram of HTML File Sizes versus Frequency

HTML Element Usage

The average number of HTML elements on institutional HTML pages is 80. Figure A2-3 gives a histogram of the numbers of HTML elements.

Note that this data is based on counts of HTML start tags. It will omit implied element usage (such as text following a head element which have an implied paragraph start tag).

Also note that in a web document consisting of several frames the numbers of HTML start tags will only cover the tags included in the page containing the information about the frames, and not the documents included in the frames.

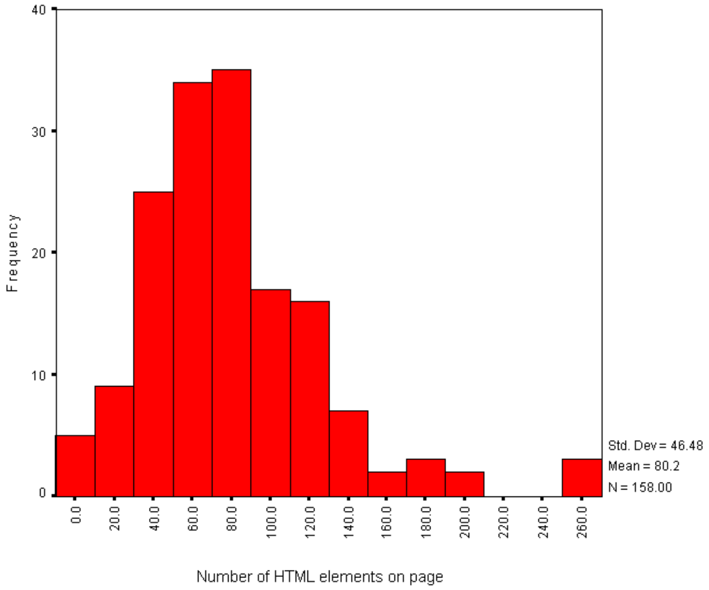


Figure A2-3 - Histogram of Numbers of HTML Elements versus Frequency

The most frequently used HTML element in the sample was the <A> element. Figure A2-4 gives a summary of the five most popular HTML elements.

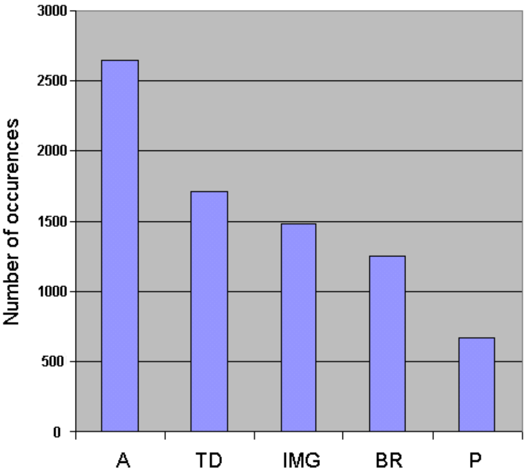


Figure A2-4 - The Five Most Widely Used HTML Elements

Examination of Particular HTML Elements

Usage of a number of particular HTML elements was examined in greater detail.

- A total of 104 out of the 158 institutions surveyed (66%) made use of the <TABLE> element.
- A total of 12 out of the 158 institutions surveyed (7.6%) made use of the <FRAME> element.
- A total of 11 out of the 158 institutions surveyed (7%) made use of the <SCRIPT> element. Note that does not include use of JavaScript event handlers.
- A total of 16 out of the 158 institutions surveyed (10.1%) made use of client-side maps.
- One institution made a single use of an inline style defined in the HTML BODY and one institution made a single use of an inline style defined in the HTML HEAD.

In addition it was observed that there were no occurrences of Java in institutional home pages. There was one occurrence of a page with background sound.

A number of metadata attributes were analysed, including:

- The GENERATOR attribute which defines the tool used to create the HTML page. This attribute is created by software such as
- Microsoft FrontPage and Netscape Gold.
- The NAME="Description" and NAME="Keywords" attributes which are used by the AltaVista search engine.
- PICS metadata.
- Dublin Core metadata.
- The REFRESH attribute, used to refresh pages and to automatically load other pages.

A histogram of use of these <META> element attributes is shown in Figure A2-5.

Software used to create the home page included various Netscape authoring software (15 occurrences, 9.5%), Microsoft Front Page (12 occurrences, 7.6%), Internet Assistant for Word (3 occurrences, 1.9%), Claris HomePage (3 occurrences, 1.9%) and PageMill (1 occurrence, 0.6%).

The "REFRESH" attribute was used to refresh the page (or send the user to another page) in 5 institution home pages. Of these, two used a refresh time of 0 seconds, one of 8 seconds, one of 10 seconds and one of 600 seconds.

Dublin Core metadata was used in two institutions. PICS content filtering metadata was used in two institutions.

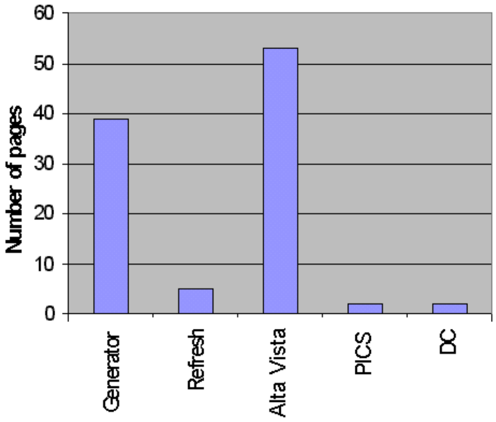


Figure A2-5 - Histogram of META Attributes versus Frequency

Numbers of Links

The average number of links on institutional HTML pages is 17. Figure A2-6 gives a histogram of the number of links.

This histogram includes links contained in the following HTML elements: <A>, <APPLET>, <AREA>, <EMBED>, <FRAME>, <BASE>, <IFRAME>, <IMAGE>, <LINK>, <MAP> and <OBJECT>. It does not, however, include links used in server-side active maps.

Note that the histogram shows the total number of links - in some cases links may be duplicated, such as links provided by client side maps and repeated as simple hypertext links.

Also note that the WebWatch robot does not obey the HTTP REFRESH method, and so the numbers of links for the small numbers of institutions which make use of REFRESH will be underestimated.

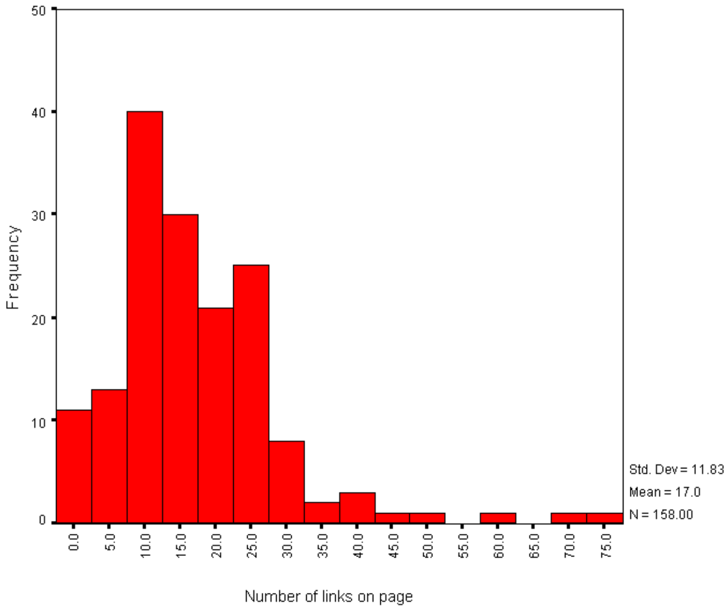


Figure A2-6 - Histogram of Numbers of Links versus Frequency

The WebWatch robot retrieves the initial HTML file specified in the input file. If this file contains a FRAMESET element the robot will only analyse the data contained in the original file, and will not retrieve the files included in the frames. This means that the numbers of links for the 12 institutions which uses frames will be underestimated.

Server Usage

The most popular server software was Apache, used by 49 institutions (31%). Figure A2-7 gives a chart of HTTP server software usage.

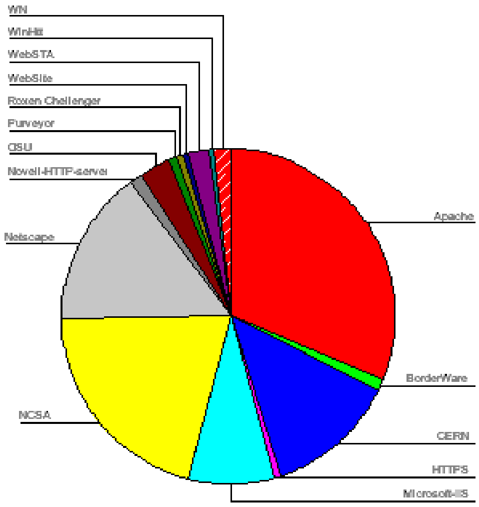


Figure A2-7 - HTTP Server Software Usage

Interpretation of Results

The results summarised in this article should be of interest to institutional web teams, as they will help institutions to compare their web site with others in the community. Institutions face the conflicting pressures of ensuring that the resources can be accessed by a wide community, using a variety of different browsers on a variety of platforms, and making their institutional entry point attractive and distinctive from other institutions. The analysis provides useful information on how the community is facing up to these conflicting pressures.

Institutional Home Pages

The analysis of institutional web pages shows a normal distribution for the size of the HTML page, with two significant outliers (Figure A2-2). On examination of these two pages, it is the use of Dublin Core metadata in one case, and extensive use of JavaScript in the other, which appear to add to the size of the HTML file. The size of the HTML file is not likely, however, to be indicative of the time needed to download the page, since this is likely to be dominated by the size of images, which were not analysed in this survey.

The analysis of the number of HTML elements also shows a normal distribution with three outliers. In each of these cases tables were used extensively to provide control over the appearance of the page.

The chart of the most popular HTML elements (Figure A2-4) shows the `<A>` (anchor) element to be most widely used, with 2,641 occurrences (an average of over 16 hypertext links per institutional home page). The next most widely used element was `<TD>` (table data), which is indicative of the popularity of tables. The third most widely used element was `<IMG>`, with almost 1,500 occurrences (an average of 9.4 images per institutional home page).

Examination of use of the `<META NAME="GENERATOR">` element attribute shows that Netscape and Microsoft are battling for the most widely used authoring tool. However it should be noted that the `GENERATOR` attribute is only used in 23% of the home pages, perhaps indicating that the majority of home pages are produced by other software packages or by hand.

The `REFRESH` attribute is used in 5 institutions to refresh a page after a period, or to send the user to another page. It is used to display an eye-catching page, and then take the user to the main institutional menu page. It should be noted that since the WebWatch robot does not make use of this attribute, the data collected by the robot will reflect the HTML page containing the `REFRESH` attribute and not the final page viewed by the end user.

Over 50 institutions make use of the metadata popularised by the AltaVista search engine. However it is perhaps surprising that more institutions do not provide such information.

Clearly both PICS and Dublin Core metadata have not yet taken off within the community, with only two institutions providing PICS information and two providing Dublin Core metadata.

The histogram of numbers of links (FigureA2- 6) shows a normal distribution, with a number of outliers. Examination of the outliers shows that a small number of institutions provide large numbers of links to their resources, whereas most institutions have a more minimalist set of links.

Almost two thirds of the sites surveyed made use of tables, indicating that table support is taken as standard by the majority of sites.

Only 7.6% of the sites made use of frames, indicating, perhaps, that institutions felt that the level of browser support of frames was too low.

Little use is made of client-side scripting languages, with only 7% of the sites made use of JavaScript in their institutional entry page. No sites made use of ActiveX. Only 10% of the sites made use of client side maps in their institutional entry page.

Only two institutions have made use of style sheets, and even this use is minimal.

Institutional Server Software

The analysis of server software shows that, as may have been expected, the Apache software is the most popular. This is followed by the NCSA and CERN software - which were the original HTTP servers used by most institutions. It is perhaps surprising that these servers are still so popular, as

NCSA and CERN are no longer significant players in the web software development circles and the CERN server, in particular, suffers from performance problems.

Netscape servers are popular, with an even split of 10 apiece between the Netscape Communications and Enterprise servers, and 3 occurrences of the FastTrack server.

Microsoft lags behind Netscape, with 12 institutions using the Internet-Information-Server software, and, surprisingly, one using the MS Windows 95 Personal Web Server.

Other server software products are used by one or two institutions.

WebWatch Futures

Further Analyses Of UK HEIs

The initial analysis of the data has provided some interesting statistics, and also indicated areas in which additional information is required.

It is planned to modified the WebWatch robot slightly in order to enable inline images and background images to be analysed.

Additional analyses will be carried out including:

- Detailed analysis of hypertext links to build a profile of hypertext linking from institutional home pages.
- Analysis of HTML conformance.
- Analysis of broken links on institutional home pages.
- Analysis of modification dates of institutional home pages.
- Analysis of client-side scripts.
- Analysis of documents using frames.

Working With Other Communities

An important aspect of the WebWatch project is liaison with various communities. We intend to give presentations of our findings at a number of conferences, workshops and seminars. In addition, we would like to work closely with particular communities, in identifying resources to monitor, interpreting the results and making recommendations to relevant bodies. If you would be interested in working with the WebWatch project, please contact Brian Kelly (email B.Kelly@ukoln.ac.uk or phone 01225 323943).

References

[1] BLRIC, <URL: http://www.bl.uk/services/ric/>

[2] New Library: The People's Network, <URL: http://www.ukoln.ac.uk/services/lic/newlibrary >

[3] UK Public Libraries, <URL: http://dspace.dial.pipex.com/town/square/ac940/ukpublib.html>

[4] HESA List of Higher Education Universities and Colleges, <URL: http://www.hesa.ac.uk/hesect/he_inst.htm>

Appendix 3 Analysis of `/robots.txt` Files

This appendix is based on an article originally published in the web version of Ariadne, issue 12. See <URL: `http://www.ariadne.ac.uk/issue15/robots/` >.

The report is based on the trawl of UK academic entry points which took place on 24 October 1997.

Showing Robots the Door

Do you have concerns about robots crawling your site? Are they overloading your web server? Do you know what they are indexing? Can you control them?* Ian Peacock *describes the Robots Exclusion Protocol and reports on a analysis of the use of this protocol by UK Universities and Colleges.

What is Robots Exclusion Protocol?

The robot exclusion protocol (REP) is a method implemented on web servers to control access to server resources for robots that crawl the web. Ultimately, it is up to the designer or user of robot-software to decide whether or not these protocols will be respected. However, the criteria defining an ethical robot includes stipulation that a robot should support REP.

This article refers to the established REP [1] accredited to Martijn Koster [2]. This involves creating a server-wide set of directives contained within the top-`level /robots.txt` plain-text file (e.g. corresponding to `http://my.server.foo-domain/robots.txt`). The currently deployed protocol allows multiple `Disallow` fields, one per-line, to be followed by a directory path. Robots parsing a `/robots.txt` file will not retrieve any resource with a URL path below the path specified by the Disallow directive. A `Disallow` field without a value is interpreted to mean no restrictions. Groups of Disallow directives must be associated with a particular `User-agent` (corresponding to the HTTP User-agent request header, which a robot should use to identify itself). This is done by inserting a `User-agent` field above the directives associated with it. The values for the `User-agent` field are allowed to be a particular user-agent (e.g. RogueRobot/v3.0), a list of user-agents or '*' which specifies all robots. Figure A3-1 gives an example of a `/robots.txt` file.

```
# This is an example robots.txt file for the site
# http://my.site.ac.uk/
# Note that you can insert comments by preceding them with a hash

# and that blank lines are also valid.

User-agent: *                       # All user-agents (except others specified in this
file)
Disallow: /cgi-bin/                 # Don't look at stuff in http://my.site.ac.uk/cgi-bin/

User-agent: WebWatch/v3.0
Disallow:                           # The WebWatch robot is allowed to look at everything

User-agent: BadRobot1, BadRobot2
Disallow: /                         # These BadRobots are denied access to everything

User-agent: IndexingRobot
Disallow: /cgi-bin/                 # Binaries are no good for indexing
Disallow: /images/                  # Images are no good for indexing
Disallow: /home-pages/              # Privacy issues with home-pages??
Disallow: /site/admin/stats/webstats    # Web stats are no good for indexing (they may also be
                                    # sensitive)
```

Figure A3-1 - An Example `/robots.txt` File

UK Universities and Colleges `/robots.txt` files

The WebWatch project has recently undertaken an analysis of the `/robots.txt` files of UK Higher Education Institutions (HEIs) main web sites, as defined by the list of institutional web services [3] maintained by NISS. From a list of 163 institutional web servers, 53 `/robots.txt` files were retrieved by a WebWatch robot. This is around 33% of the servers we looked at; the remaining servers did not have a `/robots.txt` (i.e. the server returned a 404 response code) or the connection timed-out.

The robot wrote each robots.txt file to disk for subsequent analysis. This was achieved with two Perl scripts, one to produce analysis information on the file, and another to perform basic error-checking.

The first script output records containing the following information:

- File-size in bytes
- File-size as total number of lines
- Number of lines starting with a comment
- Number of lines containing a comment
- Number of Disallow directives corresponding to each `User-agent`
- Number of Allow directives corresponding to each `User-agent`
- Total number of `User-agent` fields

where a line break was defined as NL. The `Allow` directive is not strictly part of the currently deployed protocol but was checked for.

The error-checking script scans for common errors. This has been subsequently been made into a web-based service for server administrators to check their own `/robots.txt` files.

Analysis of UK Universities and Colleges `/robots.txt` Files

Size of File

The mean size of a `/robots.txt` files is around 427 bytes. This corresponds to a mean total number of lines of about 15. Figure A3-2 shows the distribution of the total number of lines of text in a `/robots.txt` file.

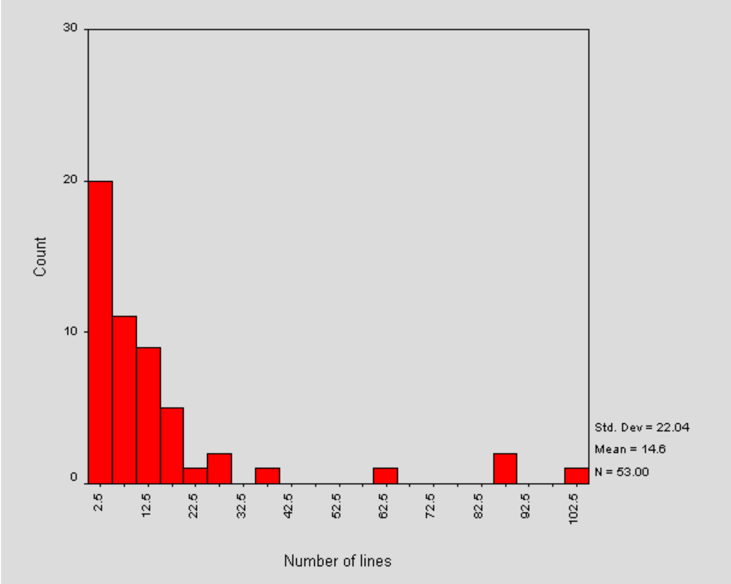


Figure A3-2 - Distribution of Total Number of Lines in our Sample of `/robots.txt` Files

The distribution of size in bytes is roughly the same shape as Figure A3-2. The two measurements of size are approximately proportional, the average number of bytes per line being about 28.

Figure 2 indicates that the majority of files contain less than 22 lines, with the outliers each representing one or two sites containing more. The large intervals between these corresponds to a visual inspection that the "average" `robots.txt` contains "typical" lines such as `Disallow: /cgi-bin` and `Disallow: /images` and a small number of sites list large (site-specific) lists of directories. It will be interesting to monitor the shape of this distribution as sites tailor `robots.txt` files to reflect their own web site.

The range of total number of lines is from 0 lines (an empty /robots.txt, of which there were 2 cases) to 101 lines.

Stripping the /robots.txt file of comments and blank lines and comparing this to the original, indicated that a large number were similar - i.e. many files contained few comments or blank lines. For those files that contained no comments and no blank lines, over 80% contained less than 6 lines in total. There were no cases of files containing only comments.

On average, 21% of a /robots.txt file is composed of non-parsed lines. Further analysis indicates that this corresponds to an average of approximately 1 blank line and 2 comment lines per file. The distribution of the total number of non-parsed lines is of roughly the same shape as the size distribution of the /robots.txt file. This suggests that comments and blank lines grow in rough proportion to the total number of lines contained in a file.

Use of User-agent

The mean number of User-agent fields included in a /robots.txt file is just over 1. There were no cases of multiple user-agents referred to by a single User-agent field. The distribution of number of User-agent fields per file is spread over 0, 1, 2, 3 and 7 occurrences. Those /robots.txt files with 0 occurrences are syntactically incorrect since each file must contain at least one reference to a User-agent.

Use of Directives

The mean number of directives per line is around nine. These were all Disallow directives - no Allow directives were found. Figure A3-3 shows a frequency distribution of the number of directives found per file.

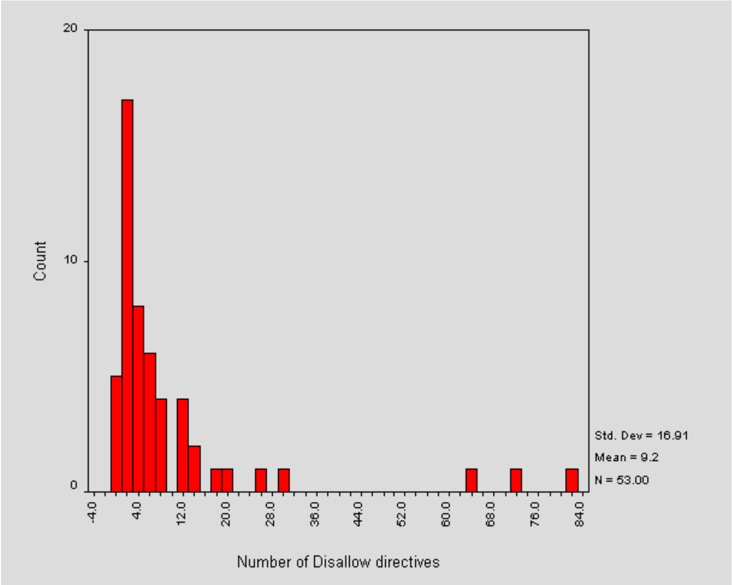


Figure A3-3 - Distribution of Number of Directives per /robots.txt File

Figure A3-3 shows that most sites are using less than 12 directives per file and that the most frequent number of directives is actually two. This is due to the large number of "standard" /robots.txt files which Disallow a couple of "standard" locations (e.g. /cgi-bin and /images). Note the logical correlation between the outliers in Figure 3 and in Figure 2 - the larger files contain more directives. There were 4 cases of 0 directives. These correspond to the zero-length files, and two invalid /robots.txt files.

Calculated from the above approximate means, the number of directives per User-agent is approximately 9. Further analysis shows this is closer to 8 and the distribution is shown in Figure A3-4.

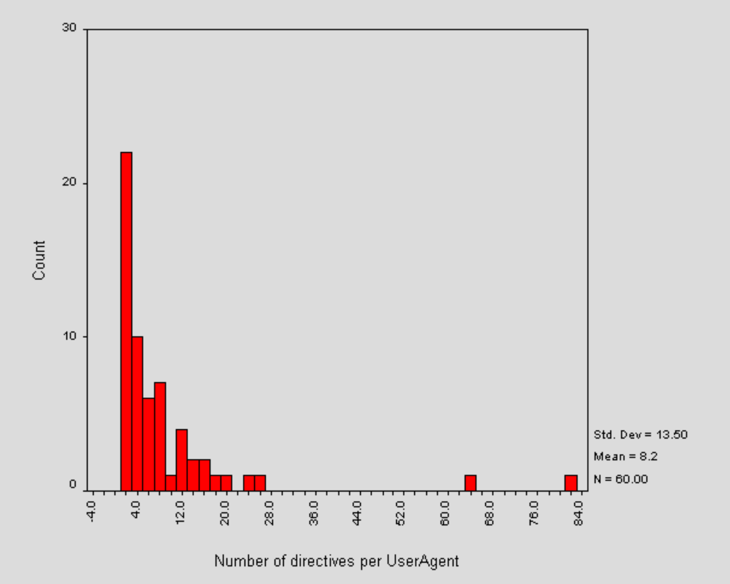


Figure A3-4 - Distribution of Number of Directives per User-agent Field

Note that in Figure A3-4 compared to Figure A3-3, some outliers have shifted left. This implies that the shifted outlier sites had their `/robots.txt` file organised into more than one User-agent field. The static outliers contain only one User-agent field with many directives, showing that the site administrators don't recognise individual robots. This is wise unless the administrator can keep up with additions and changes to the current pool of robots.

Error Checking

The second script performed some simple error checking on each of the retrieved `/robots.txt` files. The common errors are shown in Figure A3-5.

Errors	No. of Occurrences
User-agent field without value	5
No User-agent value corresponding to Disallow	5
No Disallow fields corresponding to User-agent	2
Warnings	
Unknown fields	4
CR DOS-style end of line	4
Empty file	2
Optimise	
Multiple User-agent fields referring to the same value	1

Figure A3-5 - Errors Encountered in `/robots.txt` Files

The three errors shown here are strictly illegal, each non-zero-length file must contain at least one `User-agent` field with a corresponding value and at least one `Disallow` directive. The latter error was triggered by the files that also triggered the "Unknown field" warning mentioned below. Interestingly, there were no cases of a file without an attempt at inserting a `Disallow` directive (apart from those of zero-length, which is valid).

The unknown fields warnings refer to a field that is not one of `User-agent`, `Disallow` or `Allow`. Closer examination of these warnings reveals that two sites put spaces or tab stops at the start of a line, before otherwise valid fields. The remaining cases failed to postfix valid fieldnames with a colon. DOS end of lines are valid, but mentioned because some Unix robots may have problems with this.

The optimisation remark refers to a file which uses multiple `User-agent` fields referring to the same user-agent. All directives referring to the same user-agent can be inserted under the one field.

Conclusions

Almost all of the 'mainstream' robots (i.e. those run by the major search-engines) and many other ethical robots respect REP. This means that having a site `/robots.txt` file will control access to your server for many robotic visitors. It is recommended that sites implement REP, if possible, in order to exercise some degree of control over server accesses and corresponding server load. Implementation will also aid the production of useful index-spaces on the web and cut-down on the proportion of 'spam' that is indexed. There are also benefits for the users of robots. Site administrators may direct indexing robots away from irrelevant material and point out 'black-holes' and other stumbling blocks for robots. As REP becomes ever-more widespread, the number of robots implementing the standard will probably increase.

It should be borne in mind that REP relies in the cooperation of a (possibly unethical) robot user. More reliable exclusion can be enforced via HTTP authentication or lower-level access controls.

The open nature of the `/robots.txt` file means that it should not contain confidential information (directly or indirectly). Disallowing a robot to index material stored in certain directories should not be an indication that the material contained within is 'secret' or sensitive. The protocol is not a security mechanism. In cases where material must be protected from indexing, password-protection should be used. For information not-requiring particularly fascist protection, it is worth remembering that a URL not-linked anywhere on the site (or other sites) will not be stumbled upon by a robot. Also within HTML forms, Submit buttons are rarely followed by robots.

The design of a `/robots.txt` file should direct task-specific robots away from areas that are irrelevant to their work. For example, a hyperlink maintenance robot needs access to the whole site, except perhaps 'black-holes' and CGI scripts (most of the time you should `Disallow` indexing of scripts). An indexing robot, on the other hand, needs access only to relevant indexable documents (i.e. you should also `Disallow` resources like images). Our observations show that there tends to be a 'standard' `/robots.txt` file similar to that shown in Figure A3-6.

```
User-agent: *
Disallow: /cgi-bin/
Disallow: /images/
```

Figure A3-6 - A Typical `/robots.txt` File

This file is fine, though does not address the characteristics of the web-space served. There is almost certainly other material which is unsuitable for indexing, for example, collections of web-logs.

There have been some reports of very large /robots.txt files causing errors when parsed by some robots.

One disadvantage of the `/robots.txt` method is that it is server-wide and should be maintained by an individual on behalf of the servers information-providers. Note that it is not valid in terms of the protocol to have a `/robots.txt` file in a subdirectory of the root ('/') directory, although employing this technique may be a useful strategy in maintaining a cross-departmental (or similar) exclusion file, perhaps with a script collecting all of these and forming the top level file.

A recent, less widely supported exclusion protocol [4] overcomes the problem mentioned above, but is restricted in other ways. The method involves directives embedded within HTML documents and allows the page author to specify whether the page should be indexed and/or followed (parsed for hyperlinks or links to inline objects).

This method is implemented with the HTML META element using the `NAME="ROBOTS"` attribute-value pair. The `CONTENT` attribute of the `META` element then includes a list of non-conflicting directives that should be implemented by a robot. The possibilities are `INDEX` or `NOINDEX` and `FOLLOW` or `NOFOLLOW`. Alternatively, the convenience keywords `ALL=` or `NONE=` may be used to precede a list of directives that should all be set on or off respectively.

Example 1

<META NAME="ROBOTS" CONTENT="NOINDEX,FOLLOW">

This document should not be indexed, but should be parsed for links.

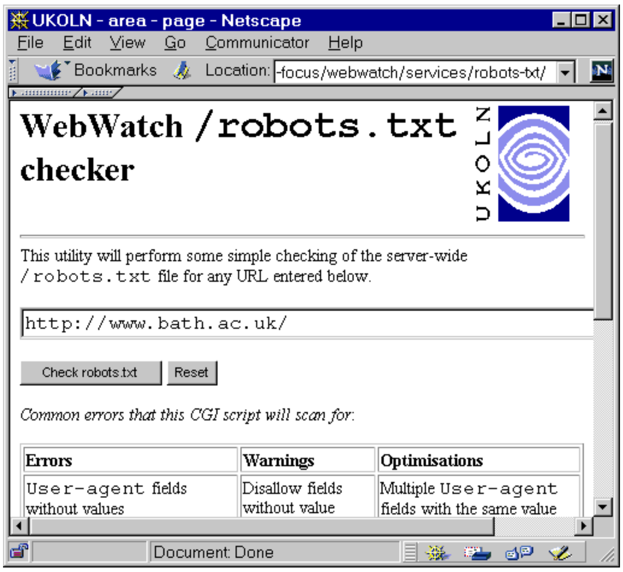
Example 2

> <META NAME="ROBOTS" CONTENT="ALL=INDEX,FOLLOW">
>
> This document should be indexed and parsed for links.

The current `/robots.txt` exclusion protocol is currently being revised and written as an Internet draft [5].

This draft clarifies a number of points originating from the previous defining document and gives a more detailed description of the relationship between robot and `/robots.txt` file. From the server administrators point of view, the new directive Allow is added. Our above analysis would indicate the lack of Allow directives to imply that this revision has not yet been widely adopted. It is not a recommendation to do - the draft is, at present, uncompleted.

The error-checking script used in the above analysis has been turned into a WebWatch service so that site-administrators can check for common errors in their own /robots.txt files. The service runs as a CGI script at <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/services/robots-txt/` >. An example session is shown in Figure A3-7.


UKOLN - area - page - Netscape
File Edit View Go Communicator Help
Bookmarks
Location: -focus/webwatch/services/robots-txt/
UKOLN
Check robots.txt
Reset
Document: Done

WebWatch /robots.txt checker

This utility will perform some simple checking of the server-wide /robots.txt file for any URL entered below.

http://www.bath.ac.uk/

Common errors that this CGI script will scan for:

Errors	Warnings	Optimisations
User-agent fields without values	Disallow fields without value	Multiple User-agent fields with the same value

Figure A3-7 - The WebWatch /robots.txt Checking Service

We intend to continue monitoring the use of `/robots.txt` files as part of the WebWatch project.

References

1 Koster, M:A ***Standard for Robot Exclusion***
<URL: `http://info.webcrawler.com/mak/projects/robots/norobots.html` >

2 Koster, M: m.koster@webcrawler.com

3 ***NISS-maintained List of UK HE Campus Information Services***
<URL: `http://www.niss.ac.uk/education/hesites/cwis.html` >

4 ***HTML Author's Guide to the Robots META tag***
<URL: `http://info.webcrawler.com/mak/projects/robots/meta-user.html` >

5 Koster, M: ***Internet Draft specification of robots.txt***
<URL: `http://info.webcrawler.com/mak/projects/robots/norobots-rfc.html` >

Appendix 4 Trawl Of UK University Entry Points - July 1998

The Trawl

A trawl of UK University entry points was initiated on the evening of Friday 31 July 1998. The results of this trawl are compared with a trawl of the same community which was carried out on 24 October 1997.

The NISS list of Higher Education Universities and Colleges [1] was used for the trawl. This file contains 170 institutions. The WebWatch robot successfully trawled 149 institutions. Twenty-one institutional home pages could not be accessed, due to server problems, network problems, restrictions imposed by the robot exclusion protocols or errors in the input data file.

A total of 59 sites had `robots.txt` files. Of these, two sites (Edinburgh and Liverpool universities) prohibited access to most robots. As these sites were not trawled they are excluded from most of the summaries. However details about the server configuration is included in the summaries.

Note that when manually analysing outliers in the data it was sometimes found that information could be obtained which was not available in the data collected by the robot.

The Findings

A brief summary of the findings is given below. More detailed commentary is given later in this article.

Server	**Usage (No. / %) Oct 1997**	**Usage (No. / %) July 98**	**Comments**
Apache	48 / 31%	62 / 42%	Mostly Unix platform (possibly also Windows NT)
Netscape	24 / 15%	25 / 17%	Unix and Windows NT platforms
Microsoft	13 / 8%	20 / 13%	Windows NT platform
NCSA	33 21%	14 / 9%	Unix platform
CERN	20 / 13%	13 / 9%	Unix platform
Webstar	3 / 2%	4 / 2%	Macintosh platform
Novell	3 / 2%	3 / 2%	PC
OSU	5 / 3%	2 / 1%	Dec VMS platform. Used at http://www.mdx.ac.uk/ and http://www.rhbnc.ac.uk/
Lotus Domino	0 / 0%	1 / 1%	Windows NT platform. Used at http://www.henleymc.ac.uk/
BorderWare	2 / 1%	1 / 1%	Used at http://www.marjon.ac.uk/
SWS	0 / 0%	1 / 1%	Sun (Unix) platform. Used at http://www.norcol.ac.uk/
HTTPS	1 / 1%	1 / 1%	Used at http://www.rgu.ac.uk/
WinHTTPD	1 / 1%	1 / 1%	Used at http://www.ssees.ac.uk/
WN	0 / 0%	1 / 1%	Used at http://www.haac.ac.uk/
Microsoft PWS	1 / 1%	0 / 0%	Was used at http://www.rave.ac.uk/ Now upgraded to Microsoft-IIS.
Purveyor	1 / 1%	0 / 0%	Was used at http://www.uwic.ac.uk/ Now upgraded to Microsoft-IIS
Roxen Challenger	1 / 1%	0 / 0%	Used at http://www.uel.ac.uk/ Server down at time of second trawl.
WebSite	1 / 1%	0 / 0%	Used at http://www.york.biosis.org/ Site not in input file of second trawl.
TOTAL	**157**	**149**	

Table A4-1 Table of Server Usage

As can be seen from Table A4-1 the Apache server has grown in popularity. This has been mainly at the expense of the NCSA and CERN servers, which are now very dated and no longer being developed.

In addition a number of servers appear to be no longer in use within the community (e.g. Purveyor and WebSite). Microsoft's server has also grown in popularity.

The popularity of Apache is also shown in the August 1998 Netcraft Web Server Survey [2], which finds Apache to be the most widely used server followed by Microsoft-IIS and Netscape-Enterprise. The Netcraft surveys are taken over a wider community than the academic sites looked at in this paper. The community surveyed by Netcraft is likely to consist of more diverse platforms (such as PCs) whereas academic sites show a bias towards Unix systems. This may explain the differences in the results of the next most popular servers.

Table A4-2 shows a profile of HTTP headers.

HTTP/1.0	50%
HTTP/1.1	50%
Cachable resources	54% of HTML pages and 60% of images
Non-cachable resources	1% of HTML pages and 0% of images
Cachability not determined	36% of HTML pages and 40% of images

Table A4-2 HTTP Headers

Note that this information was not collected for the first trawl due to limitations in the robot software.

In Table A4-2 a resource is defined as cachable if:

- It contains an `Expires` header showing that the resource has not expired
- It contains a `Last-Modified` header with a modification date greater than 1 day prior to the robot trawl.
- It contains the `Cache-control: public` header

A resource is defined as **not** cachable if:

- It contains an `Expires` header showing that the resource has expired
- It contains a `Last-Modified` header with a modification date coinciding with the day of the robot trawl
- It contains the `Cache-control: no-cache` or `Cache-control: no-store` headers
- It contains the `Pragma: nocache` header

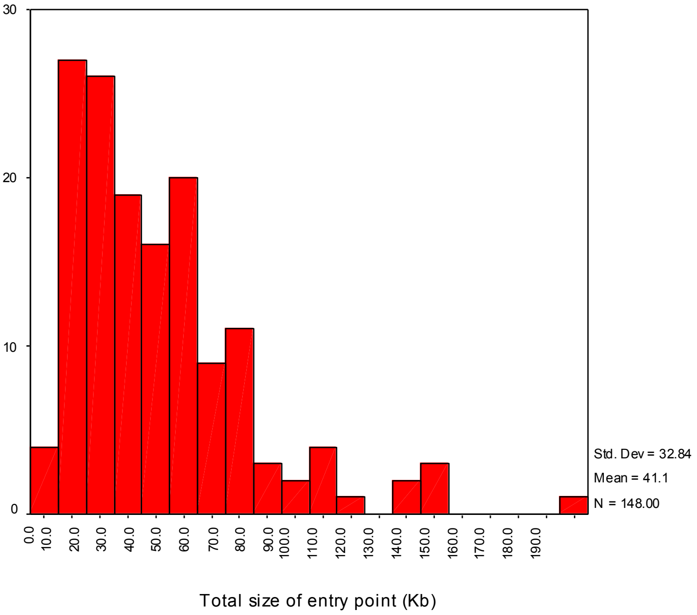


Figure A4-1 Size of Entry Point

The cachability of resources was not determined if the resource used the `Etag` HTTP/1.1 header, since this would require additional testing at the time of the trawl which was not carried out.

Figure A4-1 gives a histogram of the total size of the institutional entry point.

As shown in Figure A4-1, four institutions appear to have an institutional web page which is less than 5Kbytes. The mean size is 41 Kb, with a mode of 10-20 Kb. The largest entry point is 193 Kbytes.

Note that this information is based on the size of the HTML file, any framed or refresh HTML pages, inline images and embedded Java applets.

It does not include any background images, since the current version of the robot does not parse the `<BODY>` element for the `BACKGROUND` attribute. Subsequent analysis showed that 56 institutions used the `BACKGROUND` attribute in the `<BODY>` element. Although this would increase the file size, it is unlikely to do so significantly as background elements are typically small files.

The histogram also does not include any linked style sheet files. The WebWatch robot does not parse the HTML document for linked style sheets. In this the robot can be regarded as emulating a Netscape 3 browser.

Figure A4-2 gives a histogram for the number of images on the institutional entry point. As mentioned previously this does not include any background images.

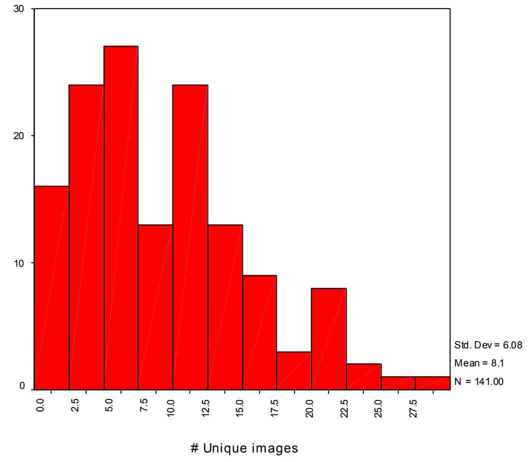


Figure A4-2 Numbers of Images

Figure A4-3 gives a histogram for the number of hypertext links from institutional entry points.

Note that Figure A4-3 gives the total number of links which were found. This includes <A> elements and client-side image maps. Note that typically links in client-side maps are duplicated using the <A> element. No attempt has been made in this report to count the number of unique links.

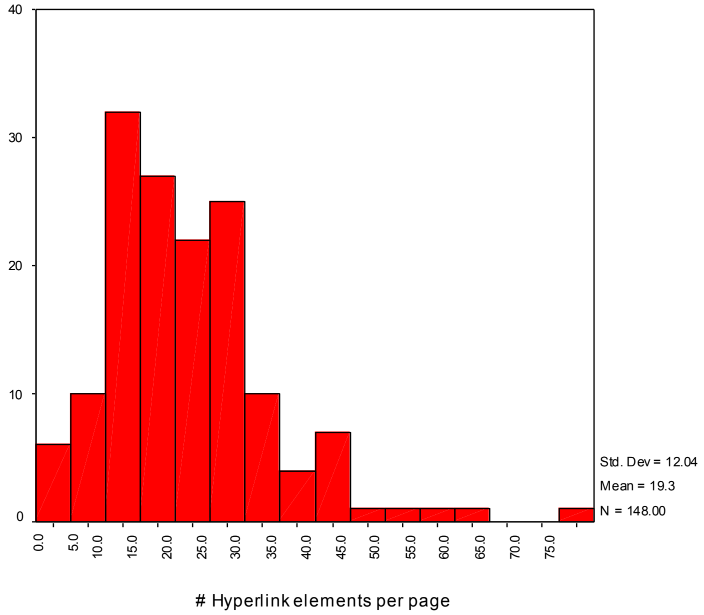


Figure A4-3 Link Profiles

Discussion of Findings

In this section we discuss the findings of the trawls.

The discussion covers the accessibility of the pages and the technologies used. In the accessibility discussion we consider factors relevant to users accessing the pages, including the files sizes (which affects download times), whether the pages can be cached (which also affects download times) and the usage of hyperlinks (which can affect the usability). In the technology discussion we consider the technologies used, such as server hardware and software, and web technologies such as use of JavaScript and Java, metadata and style sheets.

The results of the WebWatch trawl are intended to correspond closely with those that would be observed by a user using a web browsers. This is unlike, for example, many indexing robots which are not capable of processing frames. Robot software can also have problems in downloading linked resources, such as style sheet files, parsing HTML elements which may link to external resources, such as images, or processing HTTP headers, such as redirects. Robots developers often have a conservative approach to implementing new features in order to minimise the dangers of robots recursively requesting resources or causing other network or server problems.

The WebWatch has a similar conservative design. In a number of cases the automated analyses were modified by subsequent manual investigation in order to provide results which are applicable to a human view of a website (for example the size of a framed resource is the sum of the framed elements and not the hosting frameset). Where it has not been possible to do this, commentary is provided.

Size of Institutional Entry Point

The majority of institutional entry points appear to be between 10 Kb and 100 Kb (excluding background images which, as stated previously, were not included in the analysis).

Details of the largest and smallest institutional entry points are given in Table A4-3.

Institution	Size	Comments
South Devon College http://www.torbay.gov.uk/sdc/	0.5 Kb	Error in input data file. Points to directory listing, not to resource
Royal College of Music http://www.rcm.ac.uk/	2.9 Kb	
Westminster College http://www.ox-west.ac.uk/	3.9 Kb	Temporary interface while website being redesigned
University of Plymouth http://www.plym.ac.uk/	4.2 Kb	Contains background image (size not included in analysis)
Kent Institute of Art and Design http://www.kiad.ac.uk/	192 Kb	Contains animated GIF
University of Greenwich http://www.gre.ac.uk/	145 Kb	Contains animated GIF
Regent's College http://www.regents.ac.uk/	143 Kb	(Not available for manual analysis at time of writing)
University of Central England http://www.uce.ac.uk/	137 Kb	Contains animated GIF
King Alfred's http://www.wkac.ac.uk/	134 Kb	Contains animated GIF

Table A4-3 Summary Details of Largest and Smallest Sites in Current Trawl

Although perhaps not noticeable when accessing the page locally or across the SuperJANET network the large differences in sizes between, for example, the entry points for the University of Plymouth University and the Kent Institute of Art and Design are likely to cause noticeable differences in the download time for overseas users or accesses using modems.

It was also noted that all of the large sites which were available for manual inspection contained animated images.

Cachability of Institutional Entry Point

Interest in caching has grown in the UK Higher Education community since the advent of institutional changing for international bandwidth. In addition to interest in the cachability of resources from overseas websites, institutions are interest in the cachability of their own pages, especially key pages such the main entry point. Speedy access to such pages from local caches can be important when attempting to provide information to remote users, such as potential students. Unfortunately the need to provide cache-friendly pages may conflict with the need to provide attractive customised pages.

A study of the cachability of institutional entry points was carried out in order to observe the priorities given by institutions.

Over half of the institutional entry points have been found to be cachable, and only 1% not-cachable. 40% of the HTML resources used the `Etag` HTTP/1.1 header which is the current recommended method of establishing cachability. Unfortunately in order to identify if a resource can be cached the `Etag` value needs to be rechecked on a subsequent trawl and this was not carried out during this survey.

Links from Institutional Entry Point

The histogram of the numbers of hyperlinks from institutional entry points shows an approximately normal distribution, with a number of outliers indicating a small number of institutions with a large number of links. The institutional with the largest number of links on its entry point was Royal Holloway at <URL: `http://www.rhbnc.ac.uk/`>. The entry point contained 76 hyperlinks.

Providing a simple, uncluttered interface, especially to users accessing an institutional entry point for the first time, is arguably preferable to providing a comprehensive set of links to resources, although it could be argued that the a comprehensive set of links can minimise the navigation though series of sub-menus.

Future WebWatch trawls of institutional entry points will monitor the profile of hyperlink usage in order to determine any interesting trends.

"Splash Screens"

"Splash screens" are pages which are displayed for a short period before an alternative page is displayed. In the commercial world splash screens are used to typically used to display some form of advertisement before the main entry page, containing access to the main website , is displayed. Splash screens are normally implemented using the `<META REFRESH="value">` element. Typically values of about 5 seconds are used. After this period the second page is displayed.

In the initial WebWatch trawl, a total of five occurrences of `the <META REFRESH="value">` element were found. Of these, two had a value of 0. This provides a "redirect" to another page rather than displaying a splash screen.

In the second WebWatch trawl, a total of four occurrences were found (at the universities of Glamorgan, Greenwich, Sheffield and Staffordshire). Further investigation revealed that a number of additional sites use this feature which weren't detected in the robot trawl, due to the site being unavailable at the time of the trawl. Further details are given in Table A4-4.

Institution	Trawl Oct 97	Trawl July 98
De Montford University	Refreshes after 8 seconds	Refreshes after 8 seconds
Glasgow School of Art	Redirects after 10 seconds	Redirects after 10 seconds (Note site not trawled due to omission in input file)
Glamorgan	Redirects to static page	Redirects to static page
Greenwich	Redirect to static page containing server-side include	Redirect to static page containing server-side include
Queen's University Belfast	Refreshes after 10 minutes	No refresh
Ravensbourne College of Art and Design	No refresh	Redirect (Note site not trawled due to omission in input file)
Sheffield	No refresh	Refresh after 10 minutes
Staffordshire	No refresh	Redirect to CGI script

Table A4-4 Comparison of Client-Side Refreshes

Metadata

Metadata can aid the accessibility of a web resource by making the resource more easy to find. Although the management of metadata may be difficult for large websites, management of metadata for a single, key page such as the institutional entry point should not provide significant maintenance problems.

The main HTML elements which have been widely used for resource discovery metadata are the `<META NAME="keywords" VALUE="…">` and `<META NAME="description" VALUE="…">`. These elements are supported by popular search engines such as Alta Vista.

The resource discovery community has invested much time and energy into the development of the Dublin Core attributes for resource discovery. However as yet no major search engine is making use of Dublin Core metadata.

Metadata Type	Oct 1997	Jul 1998
Alta Vista metadata	54	74
Dublin Core	2	2

Table A4-5 Use of Metadata

As can be seen from Table A4-5, the metadata popularised by Alta Vista is widely used, although perhaps not as widely used as might have been expected, given the ease of creating this information on a single page and the importance it has in ensuring the page can be found using the most widely used search engines.

Dublin Core metadata, however, is only used on two institutional entry points: the University of Napier and St George's Hospital Medical School. Although this may be felt to be surprising given the widespread awareness of Dublin Core within the UK Higher Education community, the very limited use appears to be indicative that web technologies are not used unless applications are available which make use of the technologies.

Server Profiles

Since the initial trawl the server profile has changed somewhat. A number of server which were in use in October 1997 (Purveyor, BorderWare, WebSite, Roxen Challenger, Windows PWS) have disappeared. The major growth has been in usage of Apache, which has grown in usage from 31% to 42%.

Unfortunately it is not possible to obtain the hardware platform on which the server is running. Certain assumptions can be made. For example, Apache probably runs on Unix platforms since the Windows NT version is relatively new and reports indicate that the Windows NT version is not particularly fast. The Microsoft IIS server probably runs on a Windows NT platform. The CERN and NCSA server probably run on Unix. Unfortunately it is difficult to make realistic assumptions about the Netscape servers since these have been available for Unix and Windows NT platforms for some time.

Based on these assumptions Table A4-6 gives estimates for platform usage, based on the Netscape server being used solely on Unix or Windows NT.

Platform	Estimated Min.	Estimated Max.
Unix	89	115
Windows NT	21	46
Other PC platform	6	6
Macintosh	4	4
DEC	2	2

Table A4-6 Estimated Platform Usage

As may be expected the Unix platform is almost certainly the most popular platform. (This cannot be guaranteed, since the Apache server is now available for Windows NT. However as it has only been available on Windows NT for a short period and the Windows NT version is believed to be less powerful than Microsoft's IIS server, which is bundled free with Windows NT, it appears unlikely that Apache has made much inroads in the Windows NT world).

It will be interesting to analyse these results in a year's time, to see, for example, if Windows NT gains in popularity.

Java

None of the sites which were trawled contained any <APPLET>, <OBJECT> or <EMBED> elements, which are used to define Java applets. However it had been previously noted that the Liverpool University entry point contained a Java applet. Inspection of the `robots.txt` file for this site showed that all robots except the Harvest robot were excluded from this site.

The little use of Java could be indicative that Java does not have a role to play in institutional entry points or that institutions do not feel that sufficient number of their end users have browsers which support Java. The latter argument does, however, appear to contradict the growing use of technologies such as Frames and JavaScript which do require modern browsers.

JavaScript

In the initial trawl 22 of the 158 sites (14%) contained a client-side scripting language, such as JavaScript. In the second trawl 38 of the 149 sites (26%) contained a client-side scripting language, such as JavaScript.

The increasing uptake would appear to indicate confidence in the use of JavaScript as a mainstream language and that incompatibility problems between different browsers, or different versions of the same browser are no longer of concern.

With the increasing importance of client-side scripting languages in providing responsive navigational aids we can expect to see even more usage in the future. Future WebWatch trawls will help to identify if this supposition is true.

Frames

There has been a small increase in the number of sites using frames. In the original trawl 12 sites (10%) used frames. In the second trawl a total of 19 (12%) sites used frames.

HTML Validation

In the second trawl only three sites contained a page of HTML that validated without errors against the HTML3.2 DTD. Since it is reasonable to assume that most institutional webmasters are aware of the importance of HTML validity and have ready access to HTML validators (such as the HTML validation service which is mirrored at HENSA [3]) we might recommend a greater adoption of validated HTML pages.

Future Work

The WebWatch project has developed and used robot software for auditing particular web communities. Future work which the authors would like to carry out include:

- Running regular trawls across consistent samples in order to provide better evidence of trends.
- Making the data accessible for analysis by others across the Web. This would probably involve the development of a backend database which is integrated with the Web, enabling both standard and *ad hoc* queries to be initiated.
- Developing a number of standardised analyses. For example the development of an analysis system for analysing the accessibility of a website for the visually impaired, or the cachability of a website.
- Providing a web-based front-end for initiating "mini-WebWatch" analyses. Work on this has already begun, with the release of a web form for analysing HTTP headers [4].

References

[1] NISS, *Higher Education Universities and Colleges*. <URL: `http://www.niss.ac.uk/education/hesites/cwis.html` >

[2] Netcraft, <URL: `http://www.netcraft.co.uk/` >

[3] WebTechs, *HTML Validation Service*. <URL: `http://www.hensa.ac.uk/html-val-svc/` >

[4] UKOLN, *URL-info*. <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/services/http-info/` >

Appendix 5 Trawl of eLib Projects

This appendix is based on an article originally published on the web. See <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/reports/elib-nov1997/` >.

This trawl took place in November 1997.

Report of WebWatch Trawl of eLib Web Sites

The third major WebWatch crawl took place in November, 1997. The WebWatch robot software analysed eLib project web sites. A report on the analysis follows.

Background

The WebWatch project analyzed eLib project web sites as defined at <URL: `http://www.ukoln.ac.uk/services/elib/projects/` > following a trawl that took place on 3 occasions in November 1997. This report gives a summary of the findings. The report is intended primarily for eLib project webmasters, but eLib project managers may also find it of interest.

The Trawl

The trawl took place on 14/15, 21/22 and 25 of November. Although it was initially intended to carry out the trawl in one run, the size of eLib project websites revealed a number of problems with the WebWatch robot and so a number of runs were needed.

Since the runs provided different sets of data, this report is based on a combination of data. Note that eLib project websites were not completely covered.

Two files of data from separate trawls have been analyzed which contain samples of HTML data, images and other resources.

Initialization

Fifty-five eLib project sites were cited for crawling. Some of these were not indexed as fully as intended as a result of problems including time-outs and various interpretation difficulties (see later).

Where relevant, we refer to % of sites rather than actual numbers to avoid misinterpretations over our analysis of two differing summary files.

Of the 55 sites considered:

- 1 no longer existed
- 35 had no robots.txt at the top level (http://foo.ac.uk/robots.txt)
- 11 had their own domain name (in which we include omni.ac.uk, sosig.ac.uk etc.)
- 13 had a significant machine name (e.g. cain.ulst.ac.uk. Not including naming schemes like
- omni.ac.uk).
- 3 contained a tilde (~) in their path
- 18 had entry points at the top level (/) (e.g. http://intarch.ac.uk/).

Initial observations

The following are expressed as percentages of all HTML-typed files encountered.

- 19% contained a question mark, '?', in the path.
- 12% contained meta elements compliant with search engine recommendations (such as Alta-Vista).
- 5.5% contained meta elements based on Dublin Core metadata specifications.
- 1% contained meta elements based on PICS metadata specifications.
- 0.1% contained meta elements with the http-equiv='refresh' attribute.

Analyses

Web Server Software

Figure A5-1 shows a pie chart of the Web servers encountered.

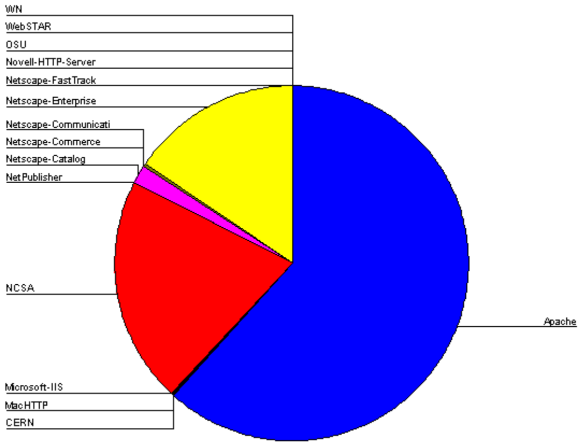


Figure A5-1 - HTTP Server Software Usage

The top three web servers are Apache (~62%), NCSA (~22%) and Netscape-Enterprise (~17%).

File Sizes

Figure A5-2 shows the frequency of HTML file sizes.

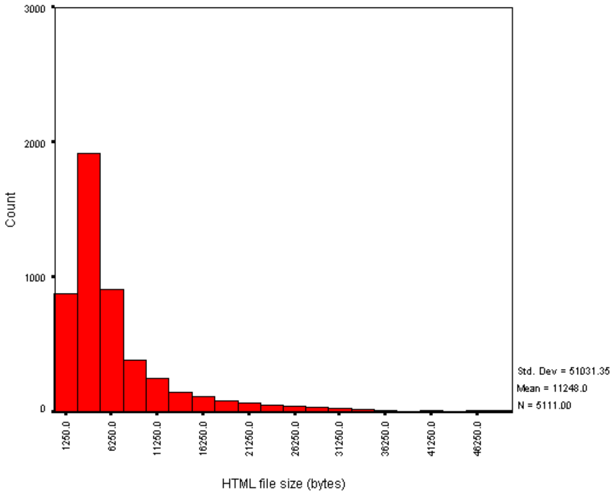


Figure A5-2 - HTML File Size

The majority of HTML documents were under 25 Kb, the mean being about 10 Kb. There were a number of extreme values for HTML file-size (the standard deviation is roughly 49 Kb), on closer inspection these are usually large indices or server logs. The smallest document encountered was 49 bytes - a CGI generated error message and the largest document was about 1 Mb - site server statistics.

Figure A5-3 shows the frequency of image file sizes encountered.

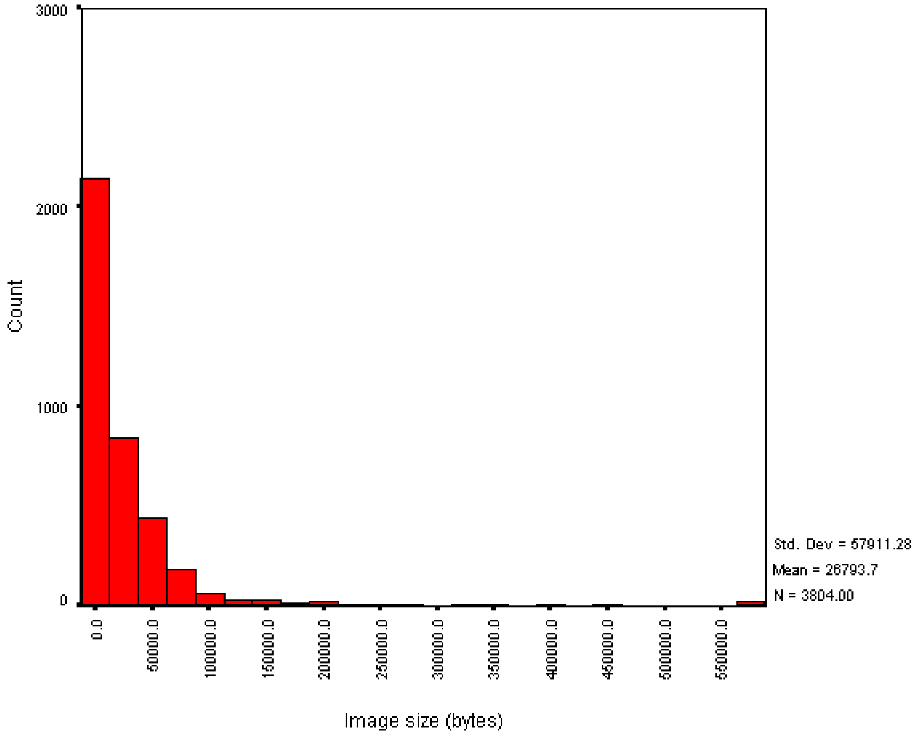


Figure A5-3 - Image File Size

The rough attenuation of the tail compared to HTML file sizes is not really an effect of the different interval-sizes. This might be due to our incomplete trawling of some sites.

We encountered mostly JPEG and GIF formats, the later being roughly 9% more prevalent than the former. The GIF size distribution is of a similar shape to Figure 3 (but with mean 18,732 bytes). The JPEG size distribution is slightly different and had mean 34,865 bytes.

Figure A5-4 shows the frequency distribution of the size of the entry point to the site. This is defined as the sum of the file sizes of the HTML page and all inline components (mostly inline images).

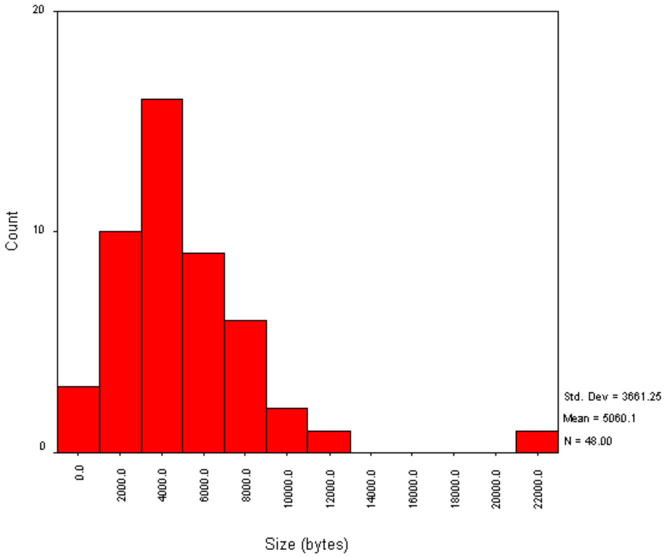


Figure A5-4 - Size of Entry Point to Site

The mean value is 5 Kb. Over a 28.8 Kbps modem connection, this would take about 1.4 seconds to download.

General HTML element usage

Figure A5-5 shows the number of unique HTML elements found on each page.

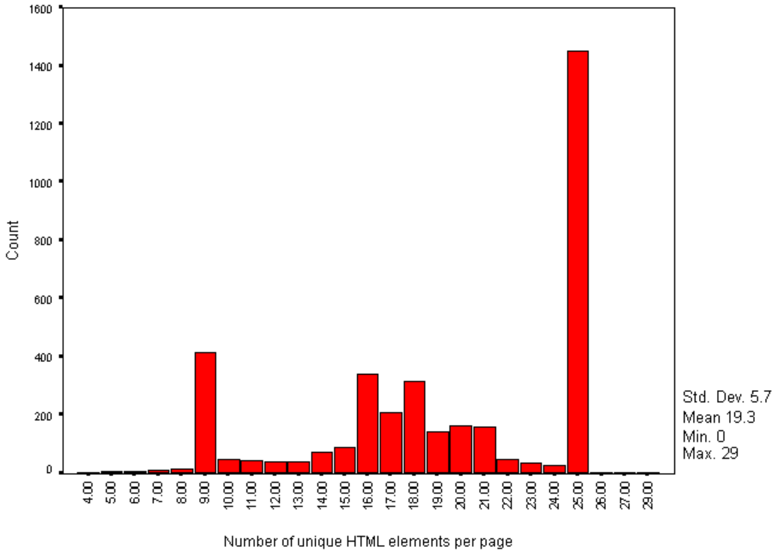


Figure A5-5 - Number of Unique HTML Elements per Page

The number of unique HTML elements on each page of HTML trawled peaks at 25. However we suspect that the distribution is approximately normal and that the peaks at 9 and 25 are due to 'in-house style' of a number of sites that were trawled completely, dwarfing others that were incompletely trawled.

In contrast, Figure A5-6 shows the count of total number of HTML elements per page.

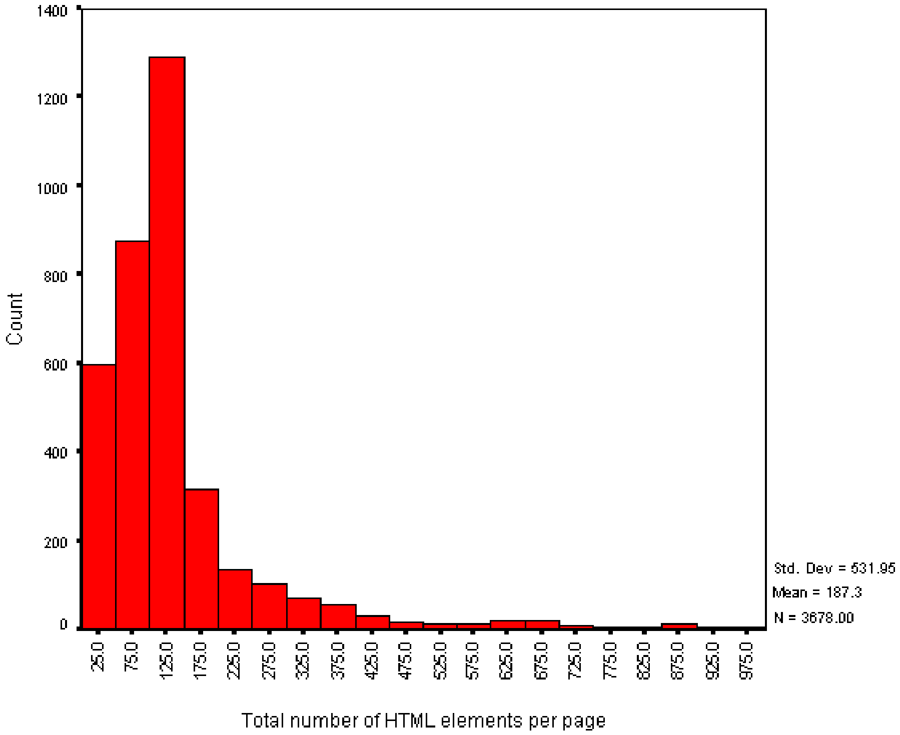


Figure A5-6 - Total number of HTML Elements per Page

The mean number of elements for Figure 5 is 187, with standard deviation 532.

The ten most popular elements used per page are shown in Figure A5-7.

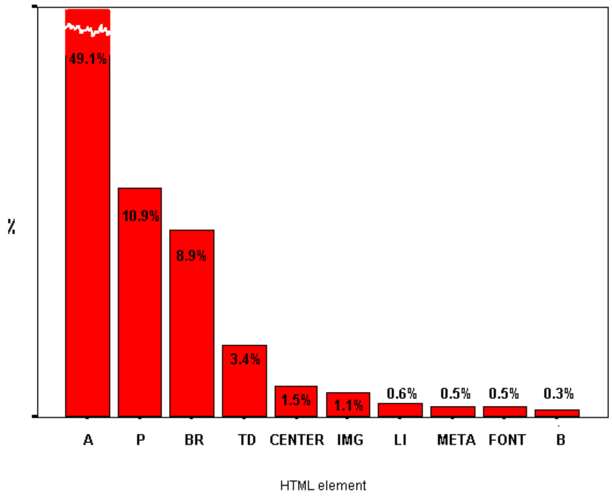


Figure A5-7 - Top ten HTML Elements (per page)

Figure A5-7 shows that the A element is by far the most widely used HTML element in the eLib project pages which were analyzed.

This could be due to extensive linking to external resources, or extensive cross-linking within the website. The Access to Network Resources (ANR) projects (such as OMNI, which was completely indexed,) are likely to contain large numbers of hyperlinks. It is perhaps surprising that eLib projects generally contain such a high proportion of hyperlinks.

Use of the META Element

We looked at specific uses of the META tag, namely for HTTP-EQUIV=refresh, search-engine metadata specifications (e.g. as recommended by Alta-Vista), Dublin Core metadata specifications and PICS metadata specifications. Around 19% of all trawled pages of HTML contained such instances of META usage. See Figure A5-7 (HTTP-EQUIVs are not shown).

Figure A5-8 shows usage of the META element.

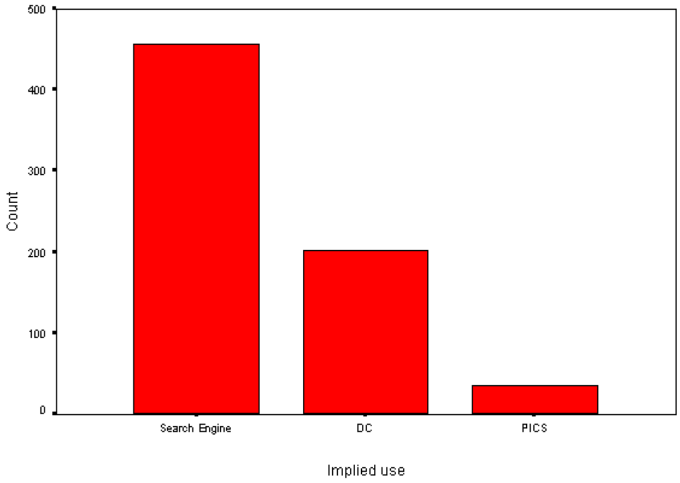


Figure A5-8 - Use of the <META> Element

A more in-depth look at the use of Dublin Core metadata is presented in Figure A5-9.

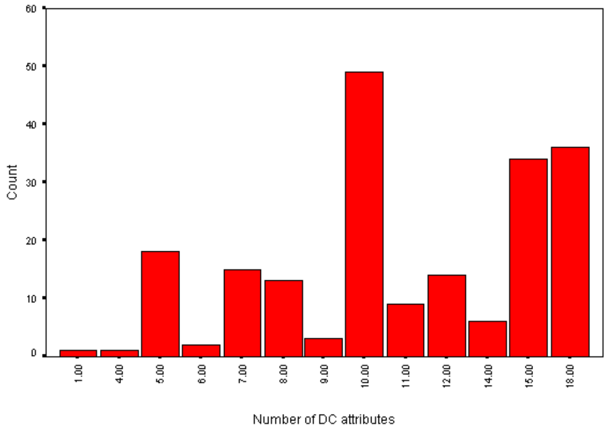


Figure A5-9 - Number of DC Attribute Values per Page

As can be seen from Figure A5-9, up to 18 DC metadata attributes per page were used. eLib pages containing DC metadata tended to make extensive use of the DC attributes, with only a small number using a handful of values.

Use of Inline Scripts

We monitored usage of the SCRIPT element. Event handlers within tags were not analyzed on this trawl.

A brief summary of the SCRIPT element is shown in Figure A5-10.

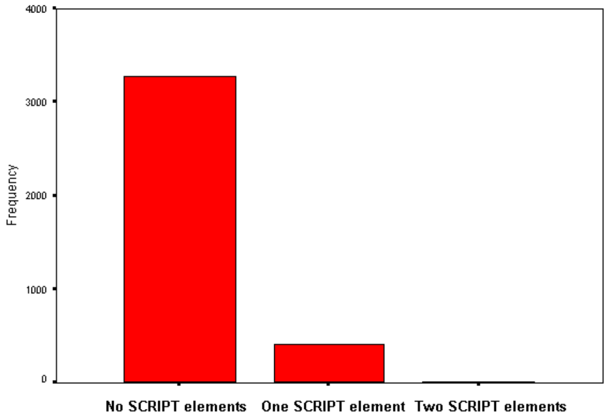


Figure A5-10 - Use of the <`SCRIPT`> Element

Analysis of Links

An analysis of absolute URL references (i.e. `http://foo.com/blah.html`) within the usual hyperlink elements (`A`, `AREA`, `LINK`, `MAP`) provides information on the top-level domains linked to.

Figure A5-11 shows the ten most popular linked-to domains. Every link in each document was considered and the top-ten calculated. Note that the y-axis is logarithmic.

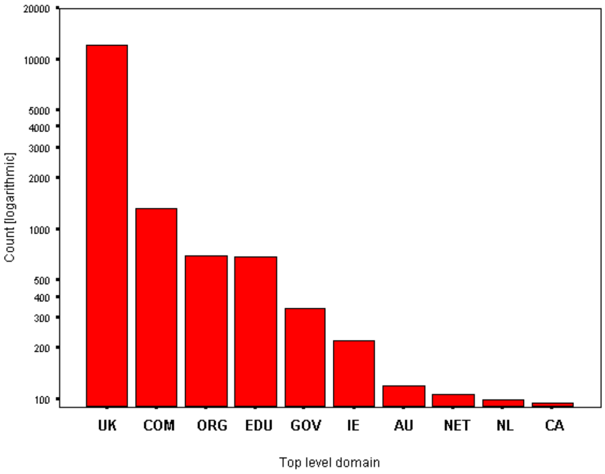


Figure A5-11 - Top Ten Linked-to Top Level Domains (Evaluated Overall)

The Robot

We have analyzed two files of data from separate trawls and have a large sample of HTML, images and other resources. When all bugs are ironed out from the software one file will suffice. Some problems with the new version of the robot meant that some sites were not being trawled completely. It may be useful to bear this in mind while considering this analysis.

Recommendations

Based on this crawl we are able to make a number of recommendations:

- Set your server to generate appropriate response codes (RCs) when there is an error. Robots normally index or audit resources if they receive an appropriate response code. Servers that are not configured to produce the correct response code may cause problems if a robot accesses your site. For example, in a small number of cases our robot audited an error message from a CGI script.
- Within a page, use relative links or ensure the machine name is the same as the publicized machine name. Since the WebWatch robot (like most other robots) does not perform DNS lookups it will assume a different hostname is an external host. For example for the site http://www.ambridge.ac.uk/ all hyperlinks should be of the form `http://www.ambridge.ac.uk/papers/paper.html` not `http://ambridge.ac.uk/papers/paper.html`.

Issues

A number of issues emerge from the survey.

Domain Naming

Some eLib project sites had their own domain name or used a significant machine name in the domain. A small number had an entry point at the top level. However most projects had an entry point which was located within the institution's directory structure and a small number used the tilde (~) convention. The persistency of these URLs may be a concern in the longer term.

Metadata

Dublin Core metadata does not yet appear to be widely used. This is perhaps due to the uncertainty of the HTML conventions for embedding Dublin Core metadata within HTML documents, which has only recently been resolved.

Server Usage

Most eLib project web sites use one of a small number of server software packages (Apache, NCSA, Netscape). Projects which make use of a little-used server package may wish to consider migrating to a more widely deployed package.

Future Trawls

Plans for future trawls include:

- More detailed analyses of the use of technologies (e.g. Java, Javascript, SGML, etc)
- More detailed analyses of the use of hyperlinks.
- More detailed analyses of the use of Dublin Core metadata.
- Detailed reports on broken links and other errors.
- Validation of HTML (and possibly stylesheets etc.).

An object-oriented re-write of the software (currently under development) will simplify the addition of future enhancements. Ideas for this version include validation and detailed reports of errors.

Appendix 6 Trawl of UK Academic Libraries

This appendix is based on an article originally published on the web. See <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/reports/hei-lib-may1998/` >.

This trawl took place in May 1998.

A Survey of UK Academic Library Web Sites

In May 1988, the WebWatch robot 'watched' UK University and College library web sites. This report contains an analysis of the data obtained from this web community.

A list maintained by NISS [1] provided entry-point URLs for 99 library sites. From these we were able to analyse 81 sites, some sites from the list were not analysed for reasons explained later. The analysis included looking at HTML and image resources and the HTTP headers associated with them. Resources will refer to HTML and images unless otherwise specified.

The data collected was processed with various Perl scripts into a form suitable for subsequent analysis with Excel and SPSS.

Analysis

Resources Collected

The summaries are of HTML and image resources and are based on the contents of 81 sites. A total of 122Mb of HTML pages representing 17,580 files and 115Mb of image data representing 14,277 images was analysed.

Figure A6-1 shows the distribution of number of resources per site.

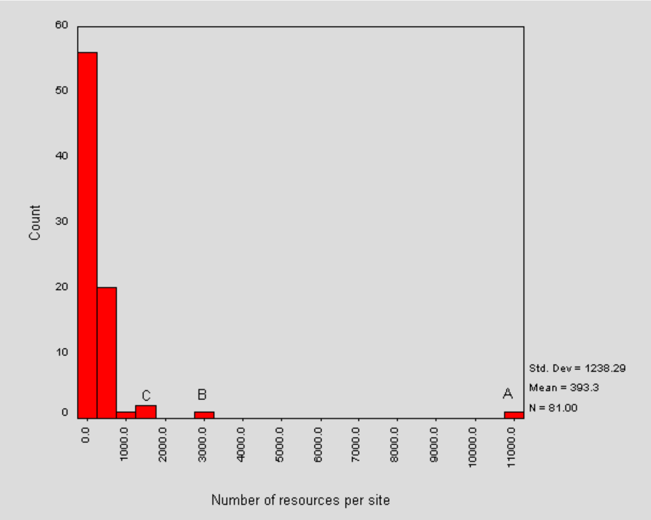


Figure A6-1 - Distribution of Number of Resources per Site

Note that a majority of sites have less than 2,000 resources under the initial path. The mean number of resources per site is around 393 and the most frequent number of resources is contained within the range 0-500. The outliers in Figure A6-1 are looked at further on.

The robot only follows resources available via hyperlinks from the entry point and so the summaries represent a users-view of the site. Any resources disallowed under the robots exclusion protocol would not be summarised (although 78% of sites (63) did not have a /robots.txt file). In total, 31,935 URLs were traversed. The domain names of these URLs correspond to the particular academic institution. The URL of most of the sites entered into the robots input file implied that the library shared space on a web server. In such a case, the robot based its traversal on remaining within the directory-path of the entry-point URL. For example, the site `http://www.bath.ac.uk/Library/` would be summarised for resources underneath `/Library/`. This applies to inline resources also, for example, a library site may include an image from a server-wide directory of images that would not be summarised.

Around 10% of the input URLs implied that the library site ran on its own server (though this could physically share a machine). These are shown in Figure A6-2.

```
www.lib.cam.ac.uk
www.lib.uea.ac.uk
libwww.essex.ac.uk
libweb.lancs.ac.uk
libweb.lancs.ac.uk
rylibweb.man.ac.uk
oulib1.open.ac.uk
www.lib.ox.ac.uk
www.library.sunderland.ac.uk
www.lib.ed.ac.uk
www-library.st-and.ac.uk
www.lib.strath.ac.uk
libweb.uwic.ac.uk
```

Figure A6-2 - Servers Dedicated to Library Information

Resource Sizes and Frequencies

Figure A6-2 shows the distribution of total size of site.

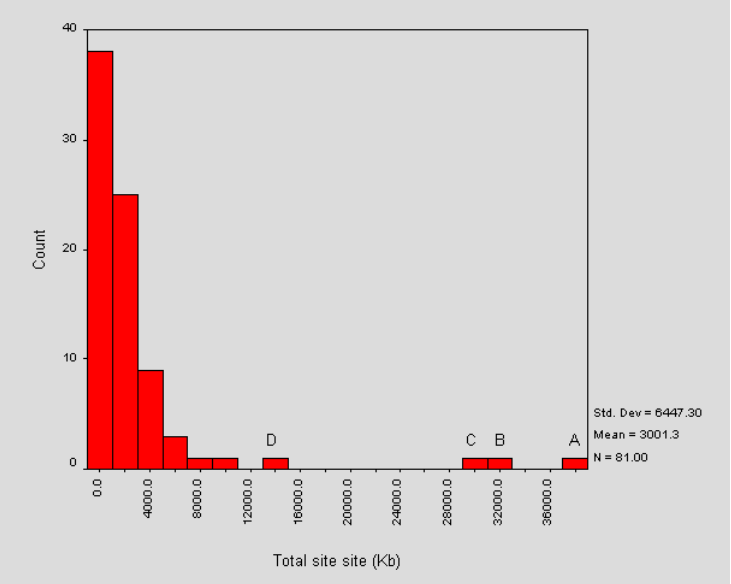


Figure A6-3 - Distribution of Total Size of Sites

Figure A6-3 initially shows an unbroken decay, which corresponds to the majority of sites. Further on there are four disconnected bars, the larger three of these contain significantly more content. The largest site analysed (labelled A in Figure A6-3), is at the University of Salford. It is 39,039Kb (38Mb). This site shares space on the institution's main web server and contains a wide range of information services material (including IT information). Site B is at the University of Aston and has size 31,137Kb (30Mb). This site contains general library and information material. This site is on a shared server and is merged with the institution's IT information point. The third outlier, C, is at the University of Manchester and corresponds to a site with its own server. The information contained on it is purely library-related. Outliers A and B of Figure 3 correspond to outliers A and B of Figure 1, i.e. the larger sites contain more resources. Outlier D in Figure 3 is part of the unbroken section of the histogram in Figure 1 (there is another site containing more resources that is within the unbroken section of Figure 3).

The mean size of an academic library web site is around 3Mb. The bars within Figure 2 have width 2000Kb, which indicates that almost half of the site sizes are within the range 0-2000Kb (0-2Mb). The site containing the least content (consisting of one page of HTML) had a total size of just over 4Kb and corresponded to a server at Roehampton Institute London.

The entry points of the smallest five sites are shown in Figure A6-4.

URL	**Size (Kb)**	**HTML**	**No. of Images**
http://www.roehampton.ac.uk/support/library/lms.html	4.3	1	0
http://www.tasc.ac.uk/lrs/lrs3.htm	7.9	5	0
http://www.sihe.ac.uk/Library/Library.html	9.0	1	1
http://www.library.sunderland.ac.uk/default.asp	10.4	1	0
http://www.mdx.ac.uk/ilrs/lib/libinfo.htm	20.2	14	0

Figure A6-4 - The Five Smallest Sites

Figure A6-5 shows the HTML and image components of Figure A6-3.

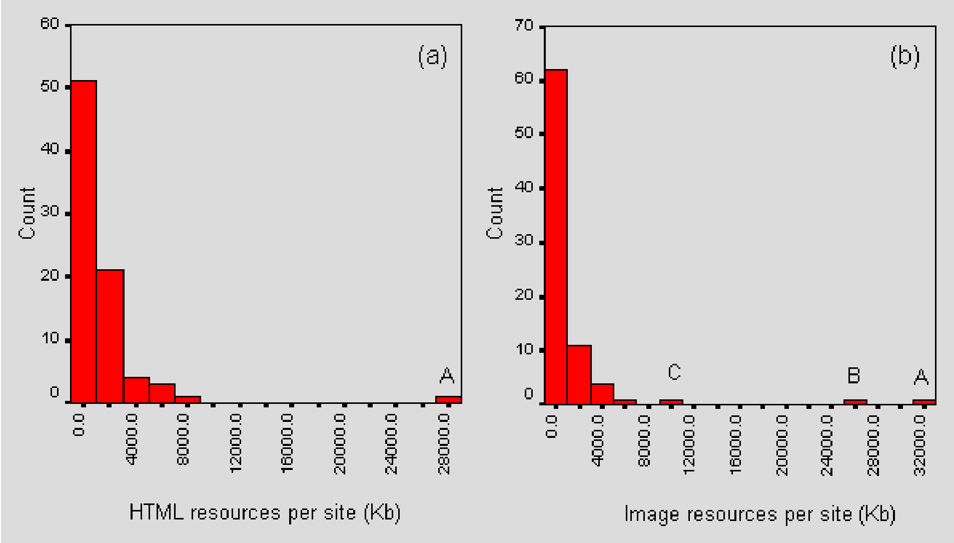


Figure A6-5 - Distributions of Total HTML Size and Total Image Size

The mean size of the HTML content of a site is around 3411Kb. We found a range from 1.5Kb to 28Mb. More than half of the sites contain under 2Mb of HTML.

Note that the image distribution is more broken than the HTML distribution suggesting that sites are more characterisable in their image size profiles than HTML size profiles. The mean total amount of image data per site is around 1.5Mb and a majority contain under 2Mb of image data. We encountered a range from 0Kb (i.e. containing no images under the entry-point path), of which there were 8 cases, to 31Mb.

The HTML size outlier (Figure 3a label A) does not correspond to any of the image size outliers (Figure A6-3b labels A-C). The correlation coefficient for the two sets of sizes is 28% (a fairly mild correlation). The furthest two image outliers (B and C) correspond to the outliers A and B of Figure A6-2. A preliminary inspection also indicates that bars to the right in Figure A6-3b tend also to be to the right in Figure A6-2. The HTML outlier corresponds to outlier B in Figure A6-2, but to none of the image outliers. We can conclude that the larger sites overall tend to contain a larger image content.

Figure A6-6 shows the HTML and image components of Figure A6-1.

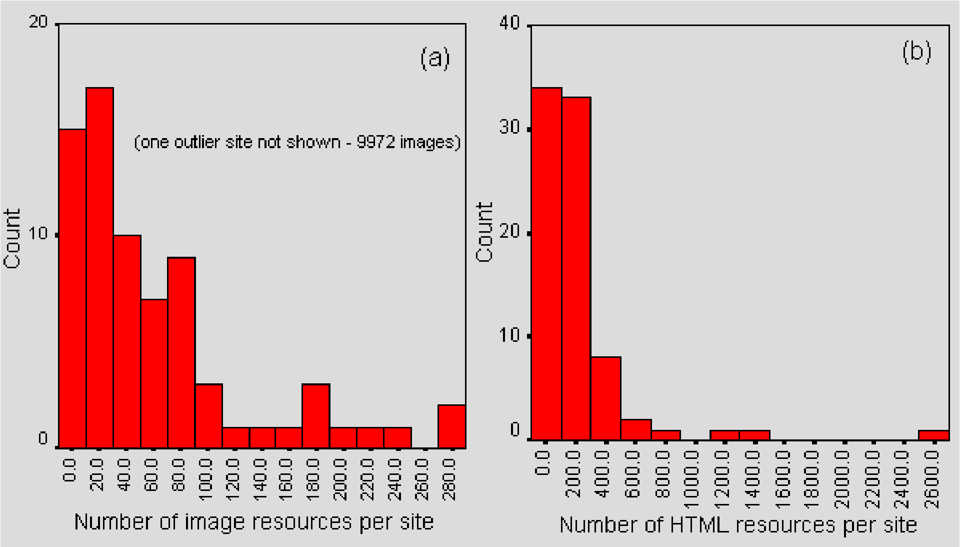


Figure A6-6 - Number of HTML and Image Resources per Site

There are, on average, 217 pages of HTML and 196 images per site (skewed by a couple of sites with many images - the standard deviation is 1162). Over 90% of sites have less than 400 pages of HTML and all but five sites have less than 200 images.

The mean size of an academic library HTML document is 6.5Kb (with standard deviation 4Kb). The distribution is nearly symmetric between 0Kb and 9Kb (the full range is 0-22Kb) and the modal value with intervals of width 1000 bytes is the range 3,500-4,500 bytes.

The mean size of an image from an academic library web site is 15Kb (with standard deviation 22Kb). The distribution decrements unbroken from 0-60,000 bytes (0-59Kb) and then becomes broken up. The modal value with intervals of size 10,000 bytes is the interval 0-10,000 bytes (0-10Kb).

Figure A6-7 shows a list of percentage correlation coefficients between counts of resources within a site and site size.

Correlation sets	% Correlation
Number of HTML resources and total size of HTML	91%
Total size of image resources and total size of site	86%
Number of HTML resources and total size of site	81%
Number of image resources and total size of images	77%
Total size of HTML resources and total size of site	73%
Number of image resources and total size of site	64%
Number of image resources and number of HTML resources	23%

Figure A6-7 - Percentage Correlation between Measures of Resource Sizes

From Figure A6-7, we can see that there is a high correlation - 91% - between the number of HTML documents within a site and the total file size of HTML resources for that site. This seems logical, however it is interesting to compare the analogous case for image resources which is lower at 77%. These percentages imply that HTML resources within a site tend to be more bounded in terms of their size than the image resources. We can also see that the size of all image data correlates quite highly with the total size of a site (86%) and that the analogous metric for HTML is lower at 73%. This would imply that the total size of a site is more dependent on its image content an its HTML content (images are usually bigger). It is interesting to note that there is a relatively weak correlation between the number of image resources and number of HTML resources (averaged over all the sites looked at). This could imply that most pages do not fit into a standard template, however multiply referenced images within a site are indexed only once. See HTML analysis below for a more in-depth look at images.

An academic library web page contains on average 0.5 of an image, with a standard deviation of 1.33. On average, almost all HTML documents include less than 2 images. Since each index was indexed only once as it is found, this figure could be misleading. However, since client-side caching will cache each image this figure could represent the total downloaded resources.

Given the means calculated above, the mean size of an academic library web page complete with inline images is 11Kb.

Use of Hyperlinks

The HTML elements which provide hyperlinking (`A` and `AREA`) were extracted for analysis. Figure A6-8 shows the distribution of hyperlink elements per site.

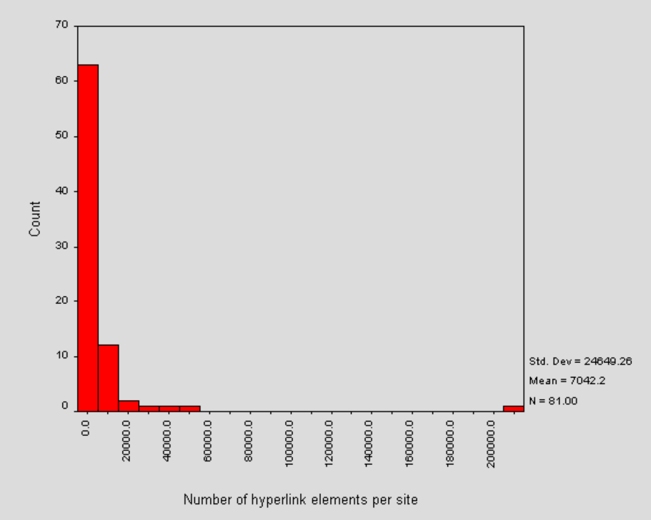


Figure A6-8 - Number of Hyperlinks per Site

The outlier is the server at Aston University and so corresponds to outlier B in Figure A6-3 and this skews the mean. On average, around 99% of HTML documents within an academic library site will contain the HTML elements that provide hyperlinking. Figure A6-9 shows the distribution of mean number of hyperlinks per document.

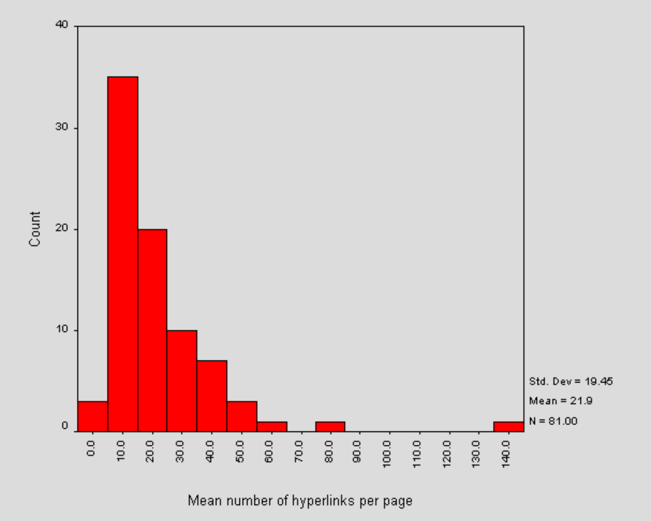


Figure A6-9 - Mean number of Hyperlinks per Document

The mean number of hyperlink elements per document is about 22 and the most frequent the interval 5-15 hyperlinks.

The URLs referenced in the hyperlinks were also profiled. For those URLs followed by the robot (i.e. within the site and under the entry-point path) the mean number of characters in a path is 25. More generally, an average 38% (with standard deviation 21%) of hyperlink URLs were qualified with `http://`. The distribution of this is approximately symmetric about 35% (the modal value); on average 1% of these URLs contained `#` and 1% contained `?`. For those that were not `http://` prefixed, around 2% contained # and under 1% contained `?`.

The Internet addresses used in the URLs were analysed. We extracted the server name and performed a reverse DNS lookup for IP addresses. The text to the right of the final '.' within the result was used to determine the domain of the server. Figure A6-10 shows the top five domains.

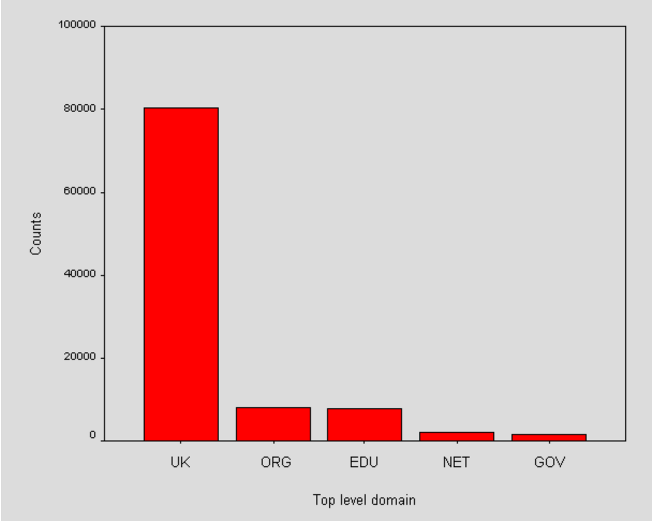


Figure A6-10 - Top Five Hyperlinked Domains

With the approach that the UK domain is geographically within the UK and that packets are not routed via another country, the chart indicates that this web community will generate traffic mostly within the UK. This may be significant when transatlantic bandwidth is charged for.

HTML Elements and Technologies

The distribution of mean total number of elements per page of HTML is shown in Figure A6-11.

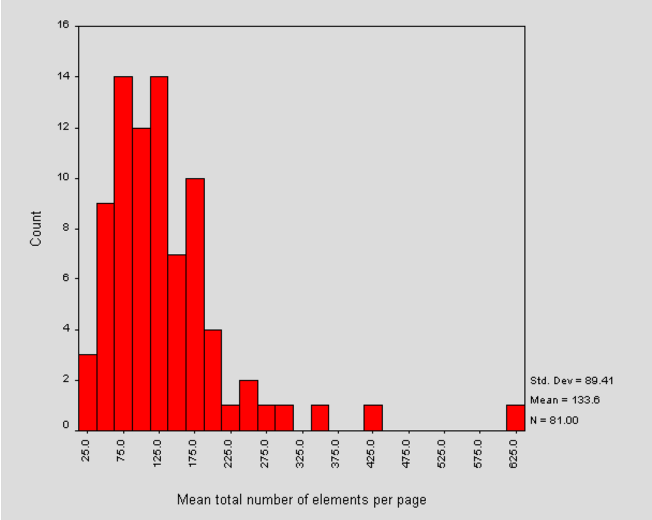


Figure A6-11 - Mean Total Number of Elements per HTML Page

The mean number of elements per page is around 134.

Interestingly the furthest outlier in Figure A6-11 does not correspond to any of the site size outliers in Figure A6-3, endorsing the earlier idea that the outliers of Figure A6-3 tend to be as a result of images. It does in fact correspond to the 8000 bar in Figure A6-5a. This site contains the greatest number of elements per page (on average) but this does not make it the site with the most HTML. However, the correlation coefficient between the mean total number of elements per page of HTML and the total site HTML content is 98%.

Figure A6-12 shows the mean number of unique elements per page of HTML.

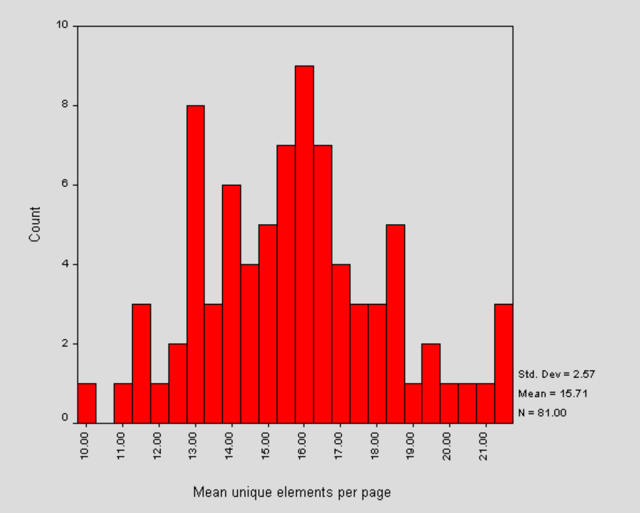


Figure A6-12 - Mean Unique Number of Elements per HTML Page

Comparing to Figure A6-11, we see that the mean of the average number of elements per page is around 16 which is also the modal value. Note that the outlier in Figure 11 has no representation in Figure A6-12. The distribution in Figure A6-12 is more symmetric and approximately normal than that of Figure A6-11. This indicates that the number of unique elements per page is likely to be bounded. In the future it will be interesting to compare a case of XML.

In contrast to the previous calculation on the mean number of images per page we also counted the number of image tags per page. Figure A6-13 shows the distribution of the mean number of images per page per site.

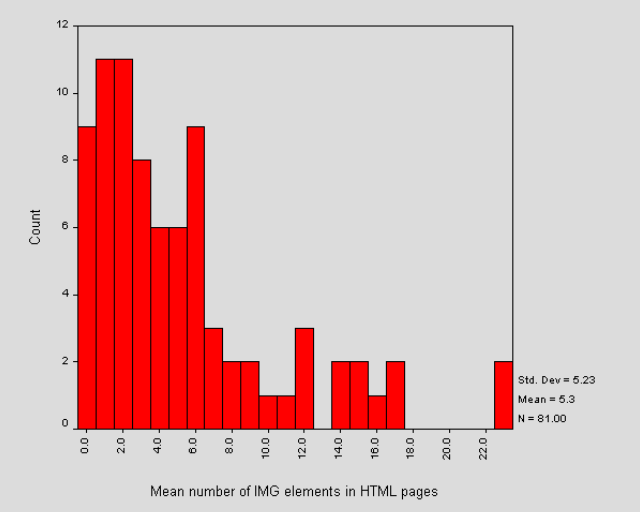


Figure A6-13 - Distribution of Mean Number of Images per Page per Site

Figure A6-13 characterises individual sites in their overall use of the `<IMG>` element. We can see that the average number of images over all sites is around 5. There is variation, but sites seem to stick within bounds (as seen by the small number of bars of height 1 count). The most frequent number of `<IMG>` elements per page is between 0.5 and 2.5.

These figures contrast with the previous calculation that there is, on average, 0.5 of an image corresponding to each page. Since the first analysis was based on number of image resources, rather than number of tags, we could deduce that image replication accounts for an average of almost 5 images per page. Client-side caching of images could therefore result in saving 75Kb of download data.

From our previous conclusions of image sizes and total size of site we might guess that the right outlier in Figure A6-13 is one of the larger sites, since the larger sites are dominated by large image resources.

This is not, in fact, the case and we conclude that this outlier includes the same images a number of times. This could suggest that the pages are written within a standard template.

Note that these figures are averages across each site - there is variation within each site. The page with the most <IMG> tags across all site contained 499 and this site corresponds to the largest outlier of Figure A6-3. The mean maximum across all sites is 23.

Each page of HTML retrieved was validated against an HTML 3.2 DTD using nsgmls [2] and the output stored. Only 12% of HTML pages validated with no errors. Although some sites contained compliant HTML documents, no site contained only compliant documents.

Within each HTML element, the following event handlers were looked for: onBlur, onChange, onClick, onFocus, onLoad, onMouseOver, onSelect, onSubmit and onUnload. 22% of sites used event handlers to some extent. The findings are shown in Figure A6-14.

Handler	No. of Occurrences	% Sites using [count]
OnBlur	0	0.0% [0]
OnChange	1	1.1% [1]
OnClick	66	7.4% [6]
OnFocus	0	0.0% [0]
OnLoad	10	6.2% [5]
OnMouseOver	1013	13.6% [11]
OnSelect	19	3.8% [3]
OnSubmit	17	3.8% [3]
OnUnload	0	0.0% [0]

Figure A6-14 - Event Handlers Found within HTML Documents

Further analysis shows that 22% of sites use event handlers.

Figure 14 shows that no use was made of the onBlur, onFocus or onUnload handlers. The most popular event-handler was onMouseover with 1013 occurrences over 11 sites. Figure 10 shows that this event-handler is used multiply on pages.

21% of sites (17 sites) made some use of the <SCRIPT> element. Where this tag contained attributes it was invariably language=JavaScript or language=JavaScript1.1.

Only one of the analysed referenced a Java applet with the <APPLET> element. There were no cases of technologies being included with the <EMBED> element or the<OBJECT> element.

Use of the <META> element was analysed. We specifically looked for the attributes shown in Figure A6-15 (case insensitively and accounting for the use of single and/or double quotes).

HTTP-EQUIV = CONTENT-TYPE	Used to specify the MIME type and character set of the document
HTTP-EQUIV = DC.*	Used to include Dublin Core metadata
HTTP-EQUIV = REFRESH	Used to tell the server to update the page
HTTP-EQUIV = PICS	Used to include PICS content-ratings
HTTP-EQUIV = REPLY-TO	Used to include an author email address
NAME = GENERATOR	Used to indicate software used in creating the page
NAME = DC.*	Used to include Dublin Core metadata
NAME = AUTHOR|DESCRIPTION|KEYWORDS	Used to include search-engine type metadata

Figure A6-15 - <META> Element Attributes Searched For

The <META> element applications are not mutually exclusive (this is illustrated in Figure A6-15). We found that 40% of sites used no search-engine type metadata within their whole site, 27% included no Generator and 35% included no Content-type. A small number of sites used Refresh (7%), DC (6%), Reply-To (1%) and PICS (1%).

Figure A6-16 shows the percentage of all pages using various <META> attributes.

72% of sites (59) made some use of the Generator attribute, and across these sites Generator appears in 38% of pages. 61% (49) used Search-engine type metadata which occurred across these sites in 28% of pages. 57 (46) used Content-Type across 24% of their pages. Refresh was used in 4% of sites (6) and across these sites was used in 4% of pages. Dublin Core was used by 6% of sites (5) and was used within 5% of the pages of these sites. PICS was used by 1% of sites (1 - EHCHE).

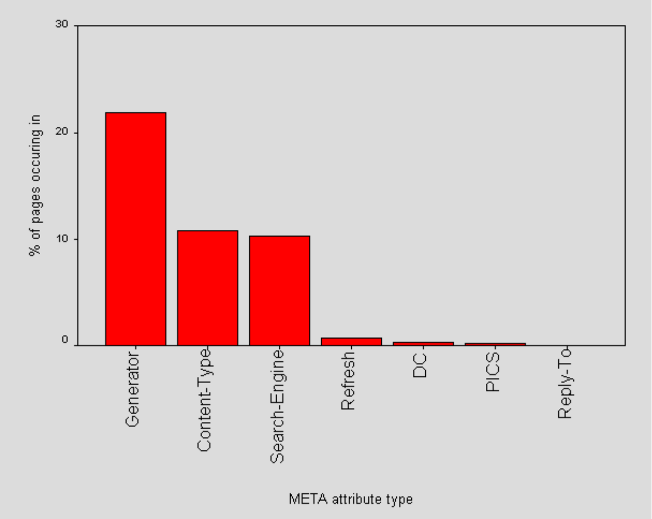


Figure A6-16 - Percentage of all Pages using Various META Attributes

Figure A6-17 shows how pages simultaneously use the <META> element.

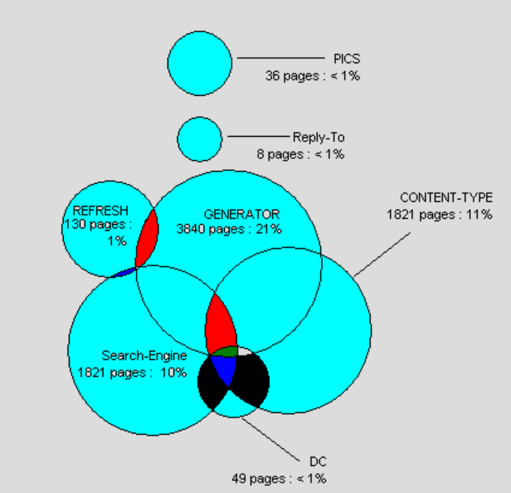


Figure A6-17 - Simultaneous Uses of the META Element

HTTP Header Analysis

All HTTP headers are recorded by the robot. For this analysis we looked at the server header, various caching headers and the HTTP-version.

Figure A6-18 shows a chart of the servers encountered.

Apache is most popular by a large margin, which probably suggests that Unix is the most popular operating system (we found nothing within the headers to suggest that it was Apache on another operating system).

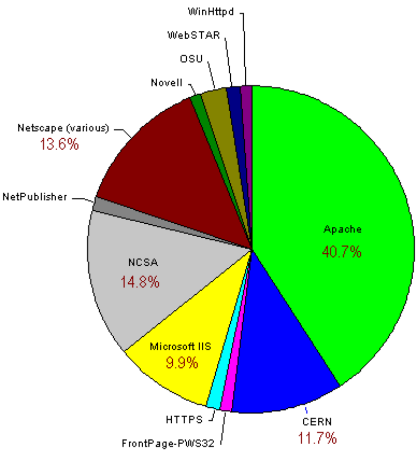


Figure A6-18 - HTTP Server Breakdown

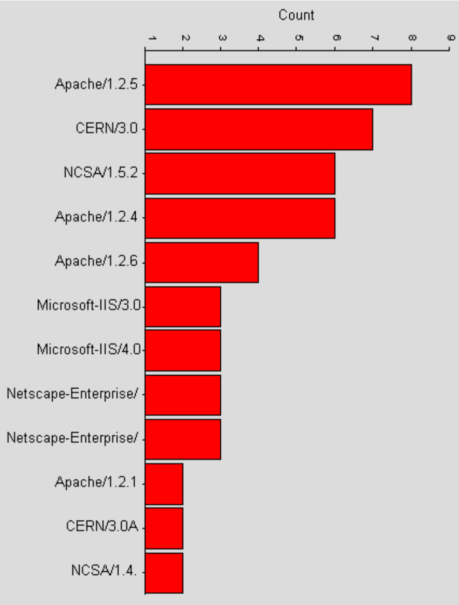


Figure A6-19 - Recurring HTTP Server Versions

The categories shown in Figure A6-19 represent supersets of the servers (the server name). The server field also contains a version number and any extensions, such as FrontPage extensions. Figure A6-19 shows the fully qualified server field for that occurred more than once.

The multiple Netscape-Enterprise bars refer to Netscape-Enterprise 2.01 and Netscape-Enterprise 3.0 (we also noticed individual cases of 3.0J and 3.0F which are not included).

Only Apache had extra information other than version and included UUOnline-2.0, mod_perl, PHP-FI-2.0 and FrontPage/v3.0.x.

There was no indication in any of the server headers as to the platform/operating system that the server runs on other than the fact that some servers only run on certain platforms (e.g. Microsoft IIS). We can infer the potential use of HTTP/1.0 or HTTP/1.1 from the servers used.

The HTTP caching headers `Etag`, `Last-modified` and any header of the form `Expir*` were looked for. Figure A6-20 shows the percentage of resources (images and HTML documents) per site that contained one or more of these headers.

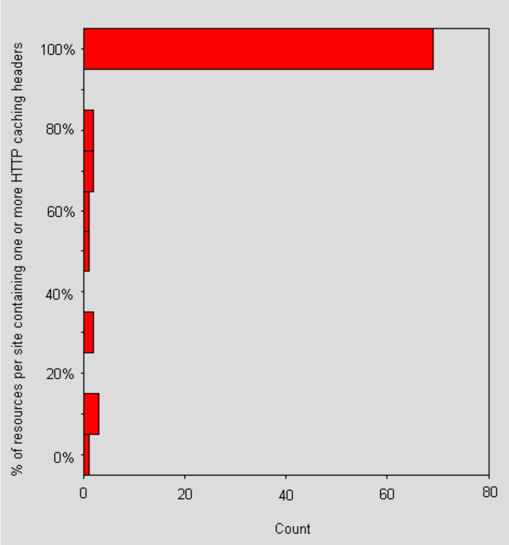


Figure A6-20 - Percentage of Resources per Site Containing one or more HTTP Caching Headers

Figure A6-20 shows that a majority of sites include one or more of the above headers. A closer examination reveals that 36% of sites have more than one such header in their HTML and a fraction more have more than one in their image headers. There doesn't appear to be any great bias in attaching caching headers to HTML or image resources. Although 37% of sites has slightly differing ratios of Cachable HTML/Total HTML to Cachable images/Total image, the mean difference across sites is only 0.1%. This is probably related to the high use of apache, which automatically includes some caching headers.

Errors

We encountered 1,151 client errors other than broken links (these include bad requests, unauthorized and forbidden). There were 935 broken links across the community (error code 404) spread over 59 sites. The distribution is shown in Figure A6-21. Note the tendency, as the number of broken links decrease, for the actual number of broken links between sites to become similar.

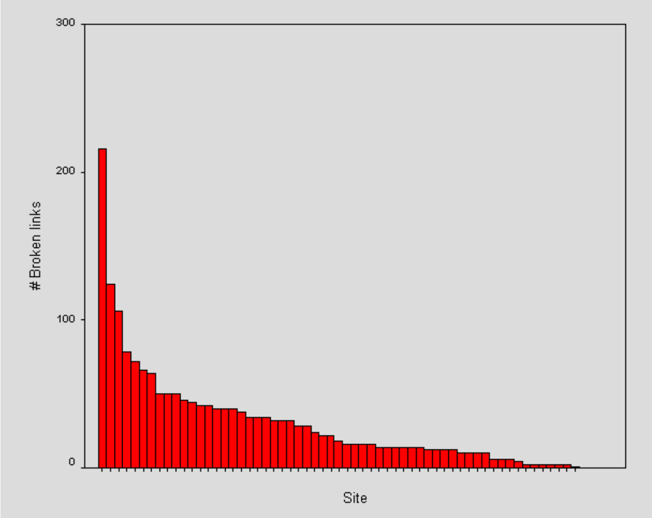


Figure A6-21 - Broken Links across Sites

The first few sites with the largest number of broken links do not correspond to the largest sites.

The sites that appeared within the input file but not in the final analysis were either dropped, due to dynamic content which confused the robot, or else at the time of crawling the entry-point could not be contacted.

References

1 *Alphabetically Sorted List of UK HE Campus Information Services*, NISS
<URL: `http://www.niss.ac.uk/education/hesites/cwis.html` >

Appendix 7 Third Trawl of UK Academic Entry Points

Introduction

On 25 November 1998 the WebWatch robot trawled the entry points for UK academic Web sites. This report is an analysis of the findings. This is the third Web crawl of the UK HEI entry points and completes a series of three snapshots of this community. The first crawl is available from the reports area of the WebWatch pages [1] and the second was published in the Journal of Documentation [2].

The input file of URLs obtained from NISS for the previous crawl was used. Of the 170 sites in this list, 150 sites were successfully crawled. Network/connection errors, out of date URLs and so on account for the 20 unexplored sites.

Size Metrics

Figure A7-1 shows a histogram of the total size of entry points. Total size is defined as the HTML page with inline images. A number of linked resources which may be downloaded by modern browsers, including external style sheets, external client-side scripts, resources requiring 'plugins' and background images are not included.

The range of sizes spans from around 5kb (`<URL: http://www.rcm.ac.uk/>`) to around 200Kb (`<URL: http://www.kiad.ac.uk/>`). The second large outlier at 192Kb corresponds to `<URL: http://www.scot.ac.uk/>`.

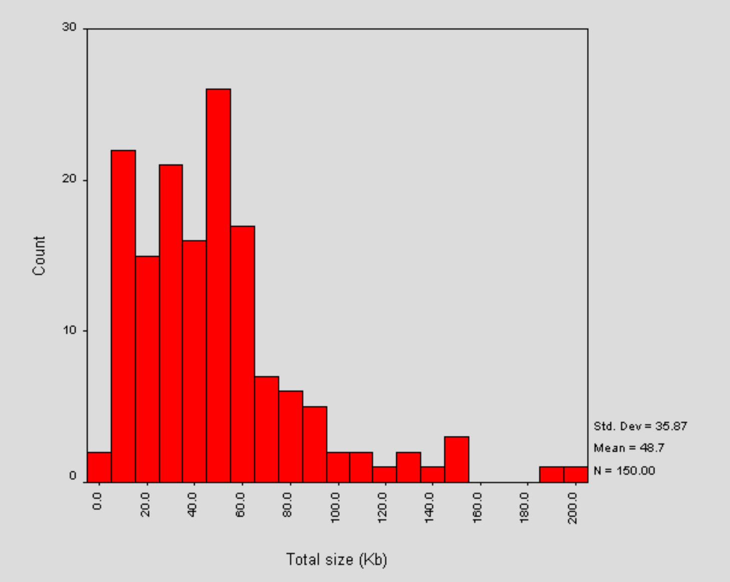


Figure A7-1 - Total Size of Entry Points

Hyperlinks

Figure A7-2 shows the number of hyperlinks within each site. These are obtained from the `A` element and from image map `AREA` elements. This data may include duplicate *URLs* where more than one hyperlink to the same URL exists.

Note that the outlier corresponds to `<URL: http://www.rhbnc.ac.uk/>`.

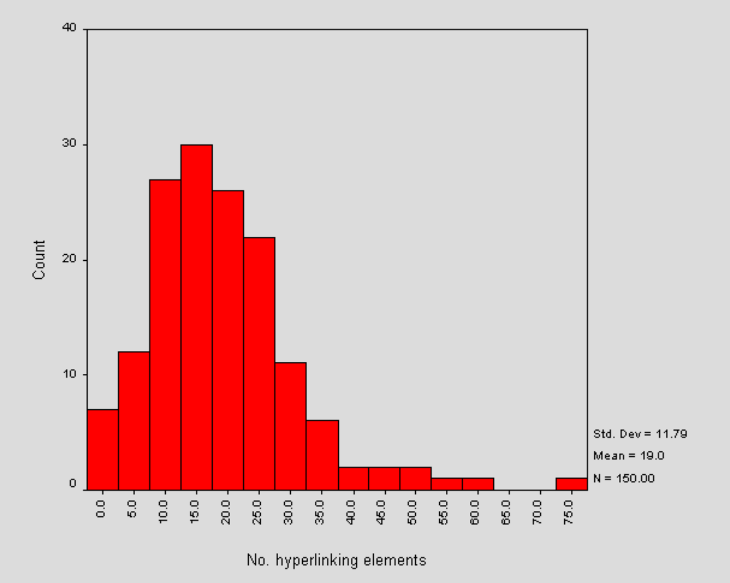


Figure A7-2 - Total Number of Hyperlink Elements per Site

HTTP Servers

Figure A7-3 shows a pie chart of the server software encountered during the crawl. This information is based upon the HTTP `Server` header returned by the web server.

The *Other* category consists of the servers listed in Table A7-1.

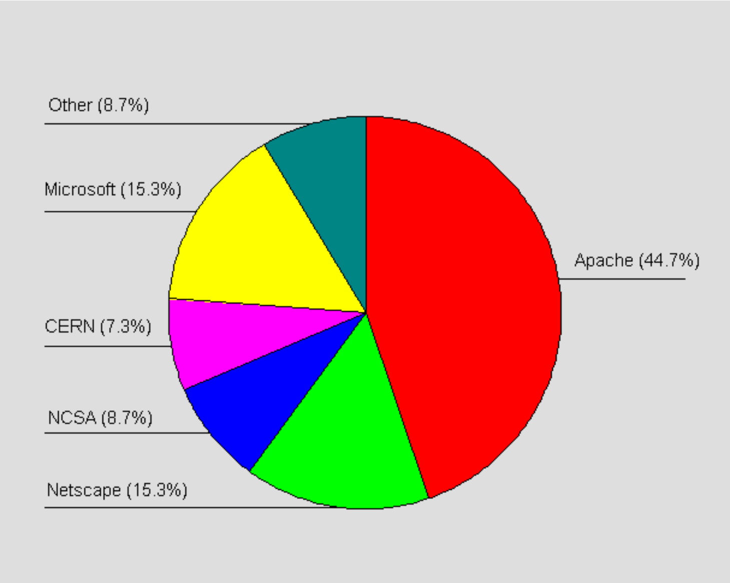


Figure A7-3 - Server Software Encountered

Server	Count
Borderware	1
Lotus Domino	1
Novell	2
OSU	2
SWS-1.0	1
WebSTAR	4
WinHttpd	1

Table A7-1 - Components of the 'Other' slice from Figure A7-3

A more detailed table of the servers found is shown in Table A7-2.

Server	Count
Apache/1.0.0	1
Apache/1.0.3	1
Apache/1.1.1	1
Apache/1.1.3	1
Apache/1.2.0	2
Apache/1.2.1	2
Apache/1.2.1 PHP/FI-2.0b12	1
Apache/1.2.4	7
Apache/1.2.4 FrontPage/3.0.2	1
Apache/1.2.5	12
Apache/1.2.6	6
Apache/1.2b10	2
Apache/1.2b7	1
Apache/1.3.0 (Unix)	7
Apache/1.3.0 (Unix) Debian/GNU	1
Apache/1.3.0 (Unix) PHP/3.0	1
Apache/1.3.1 (Unix)	6
Apache/1.3.2 (Unix)	1
Apache/1.3.3	2
Apache/1.3.3 (Unix)	6
Apache/1.3.3 Ben-SSL/1.28 (Unix) PHP/3.0.5 od_perl/1.16	1
Apache/1.3.3 UUOnline/1.4 (Unix)	1
Apache/1.3a1	1
Apache/1.3b3	1
Apache/1.3b5	1
BorderWare/2.	1
CERN/3.0	8
CERN/3.0A	3

HTTPS/2.12	1
Lotus-Doino/4.5	1
Microsoft-IIS/2.0	3
Microsoft-IIS/3.0	4
Microsoft-IIS/4.0	15
Microsoft-Internet-Inforation-Server/1.0	1
NCSA/1.	2
NCSA/1.4.	1
NCSA/1.5.1	3
NCSA/1.5.2	7
Netscape-Comunications/1.1	1
Netscape-Comunications/1.12	1
Netscape-Enterprise/2.01	3
Netscape-Enterprise/2.0a	2
Netscape-Enterprise/3.0	4
Netscape-Enterprise/3.0F	2
Netscape-Enterprise/3.0K	1
Netscape-Enterprise/3.5-For-NetWare	1
Netscape-Enterprise/3.5.1	4
Netscape-FastTrack/2.0	1
Netscape-FastTrack/2.01	1
Netscape-FastTrack/2.0a	1
Netscape-FastTrack/2.0c	1
Novell-HTTP-Server/2.5R	1
Novell-HTTP-Server/3.1R	1
OSU/1.9b	1
OSU/3.2	1
SWS-1.0	1
WebSTAR	2
WebSTAR/1.2.5 ID/13089	1
WebSTAR/2.0 ID/44693	1
WinHttpd/1.4a (Shareware Non-Commercial License	1
Total	150

Figure A7-2 - Table of all Servers Encountered

Of these servers, 40% used HTTP/1.0 and 60% used HTTP/1.1.

The Queso [3] software was used to get an idea of platforms. The high level results are summarised in Table A7-3. A more detailed breakdown is presented in Figure A7-7.

	Estimated	
OS	**Min**	**Max**
Unix	97	108
OS2	0	5
MacOS	6	11
Netware	3	3
Windows NT/95/98	20	20
Other	7	7
Unknown	6	6

Table A7-3 - Operating Systems as Reported by Queso

Note that the 'Other' category in Table A7-3 corresponds to the Queso output categories Figure A7-7) 'Cisco...' and the 'Unknown' category corresponds to the Queso output categories 'Unknown OS', 'Firewalled host/port or network congestion' and 'Dead Host, Firewalled port or Unassigned IP'.

Note that the estimated minimum and maximum values in Table A7-4 may be skewed because of the Queso unknowns referred to above.

Operating System	**Count**
BSDi or IRIX	1
Berkeley: Digital, HPUX, SunOs4, AIX3, OS/2 WARP-4, others...	5
Berkeley: HP-UX B.10.20	1
Berkeley: IRIX 5.x	3
Berkeley: usually Digital Unix, OSF/1 V3.0, HP-UX 10.x	14
Berkeley: usually HP/UX 9.x	1
Berkeley: usually SunOS 4.x, NexT	5
Cisco 11.2(10a), HP/3000 DTC, BayStack Switch	7
Dead Host, Firewalled Port or Unassigned IP	2
FreeBSD, NetBSD, OpenBSD	1
IBM AIX 4	2
IRIX 6.x	2
Linux 1.3.xx, 2.0.0 to 2.0.34	5
Linux 2.0.35 to 2.0.9	1
MacOS-8	6
Novell Netware TCP/IP	3
Reliant Unix from Siemens-Nixdorf	1
Solaris 2.x	60
Standard: Solaris 2.x, Linux 2.1.???, MacOS	5
Windows 95/98/NT	20
Firewalled Solaris 2.x	1
Firewalled host/port or network congestion	3
Unknown OS	1
Total	150

Table A7-4 - Queso Output

Metadata Profile

The attributes of the HTML `<META>` element were examined for known metadata conventions. Table A7-5 shows the results.

Metadata	**Number of `META` elements**	**No. sites**
PICS	1	1
HTTP-EQUIV="Refresh"	9	9
Reply-To	3	3
Search Engine	190	95
Dublin Core	102	11
HTTP-EQUIV="(Dublin Core)"	8	1

Table A7-5 - Types of Metadata Encountered

Technologies

Scripting

29 pages used the <SCRIPT> element to include a client-side script block. Of these, 23 pages included the attribute-value LANGUAGE="JavaScript".

All HTML elements were searched for the set of defined JavaScript event handlers. The results are shown in Table A7-6.

Handlers	Count	Sites
onChange	1	1
onClick	13	4
onLoad	10	8
onMouseOver	320	36

Table A7-6 - Event Handlers Encountered

Java

Two Java applets were referenced by the site <URL: http://www.uwic.ac.uk/>.

The site <URL: http://www.luton.ac.uk/> referenced a plugin using the OBJECT element.

Frames and "Splash Screens"

A total of 21 sites used framesets to provide a framed interface to the institutional entry point.

A total of 10 sites use HTTP-EQUIV="refresh" to provide a client-side redirect of a "splash screen" for the entry point.

Cachability

Table A7-7 shows a summary of the cachability of crawled resources.

Cachable resources	72.5% of HTML pages, 80.9% of images
Non-cachable resources	4.4% of HTML pages, 0.2% of images

Table A7-7 - Cachability of Resources Encountered

Additionally, 40% of HTML pages and 45% of images contained the HTTP/1.1 Etag header.

A resource is defined as cachable if:

- It contains an Expires header showing that the resource has not expired
- It contains a Last-modified header with a modification date greater than one day prior to the robot crawl
- It contains the Cache-control: public header

A resource is defined as **not** cachable if:

- It contains an Expires header showing that the resource has expired
- It contains a Last-Modified header with a modification date coinciding with the day of the robot crawl
- It contains the Cache-control: no-cache or Cache-control: no-store headers
- It contains the Pragma: nocache header

The cachability of resources is not determined if the resource used the Etag HTTP/1.1 header, since this would require additional testing at the time of the trawl which was not carried out.

Comparisons with Previous Crawls

Server Profiles

As shown in Figure A7-4, the Apache and Microsoft servers have shown increasing adoption. The Netscape server has fluctuated (perhaps due to a period of experimentation). The NCSA and CERN servers have shown a decrease in usage.

The growth of Apache and Microsoft servers has also resulted in a decrease of the 'Other' category, i.e. sites are subscribing to the more popular servers.

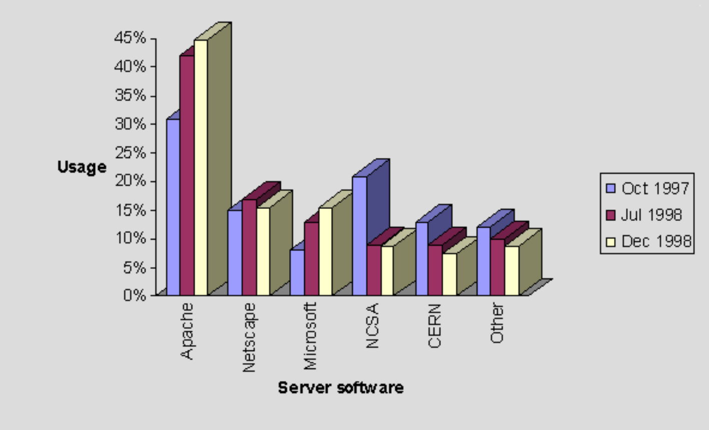


Figure A7-4 - Use of Server Software Over Three Crawls

A chart showing the growth of various servers is shown in Figure A7-5. This chart shows the contribution of growth for the period Oct 1997 - Jul 1998 and Jul 1998 - Nov 1998. Note that negative growth is interpreted as decline.

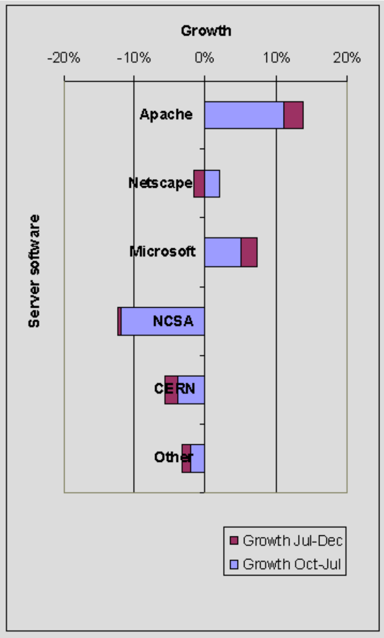


Figure A7-5 - Growth of Servers over Three Crawls

Size of Entry Points

A set of sites was isolated, for which reliable measurements of size exist for two previous web crawls. The results are shown in Figure A7-6.

Note that a majority of sites have not undergone great fluctuations in size. The outlier corresponds to <URL: `http://www.scot.ac.uk/` >. The pages for this site are different since this site has become part of a larger institution.

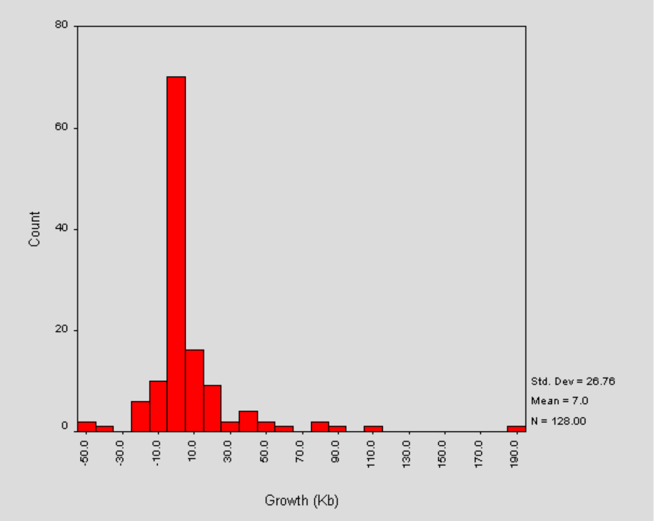


Figure A7-6 - Changes in Size of Entry Points

"Splash Screens"

The number of institutional entry points which make use of "splash screens" or redirect has shown a steady increase from five sites (Oct 97) to seven sites (July 1998) to ten sites in the current trawl.

Hyperlink Profiles

The domains referenced by hyperlinks in the three crawls have been dominated by ac.uk and this domain has shown an overall increase. Figure A7-8 shows the contribution of different types of domain name as a percentage of all hyperlinks in the site.

Domain	October 1997	July 1998	November 1998
Total .uk	97.31%	97.13%	98.00%
ac.uk	96.63%	95.94%	97.68%
net	0.30%	0.16%	0.11%
com	0.82%	0.61%	0.63%
org	0.34%	0.08%	0.18%
Other	0.15%	0.08%	0.10%
IP address	0.00%	0.12%	0.04%
Badly formed URL	1.10%	1.72%	0.91%

Table A7-8 - Domains Referenced in Hyperlinks

Note in Table A7-8, that the ac.uk data is a subset of the uk data.

Use of Metadata

In each crawl, we have looked for search-engine (SE) type metadata and Dublin-Core (DC) metadata. The findings for the three crawls are shown in Figure A7-7.

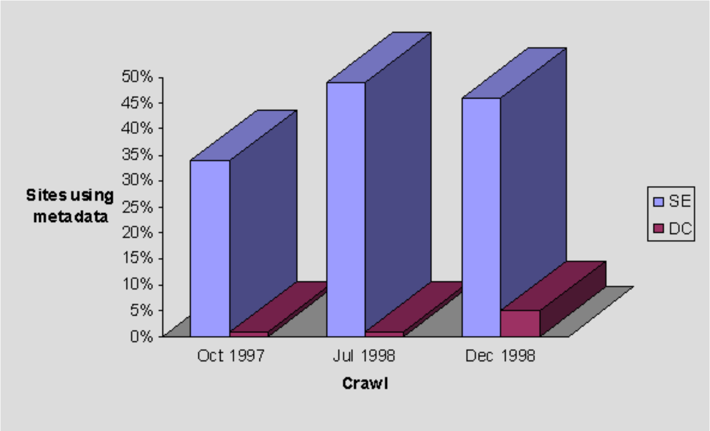


Figure A7-7 - Trends in Metadata Usage

Figure A7-7 shows that the use of Dublin Core metadata has increased considerably over the three crawls, from one site in October 1997 to 11 sites in November 1998.

References

1. **A Survey of UK Academic Library Web Sites**
<URL: http://www.ukoln.ac.uk/web-focus/webwatch/reports/hei-lib-may1998/ >

2. **How Is My Web Community Doing? Monitoring Trends in Web Service Provision**, Journal of Documentation, Vol. 55 No. 1 January 1999, pp 82-95

3. **Questo**
<URL: http://www.apostols.org/projectz/queso/ >

Appendix 8 Library Technology News Articles

The following articles were published in the News section of the Library Technology News magazine.

Public Library Domain Names - February 1998

This is the first of a series of regular News items by the WebWatch project providing information on the status of Public Libraries on the Web.

WebWatch is a project aiming to develop and use robot software for analyzing various aspects of the World Wide Web. The project is funded by the British Library Research and Innovation Centre (BLRIC) and is based at UKOLN.

Domain names can indicate the nature of the organization hosting the server. The Hardens' list of public library websites is used as the input for the WebWatch surveys. The list currently shows 133 sites. From this list the following domains were found: 43% gov.uk , 41% org.uk (of which 41% are part of EARL), 7% co.uk, 5% ac.uk, 5% com, 1.5% org and 1.5% net. The domains refer respectively to UK local government sites, UK organizations, commercial providers, UK academic institutions, companies, organizations and commercial providers.

It will be interesting to see how the proportions change as more public library websites come online.

WebWatching Public Library Web Site Entry Points - April 1998

In March the WebWatch project analysed the entry-points of public library websites. The analysis looked at the total size of the entry-points (including images) and profiled the hyperlinks contained in each (which included hyperlinks and "active maps").

Figure 1 shows a frequency distribution of the total-size of entry-points.

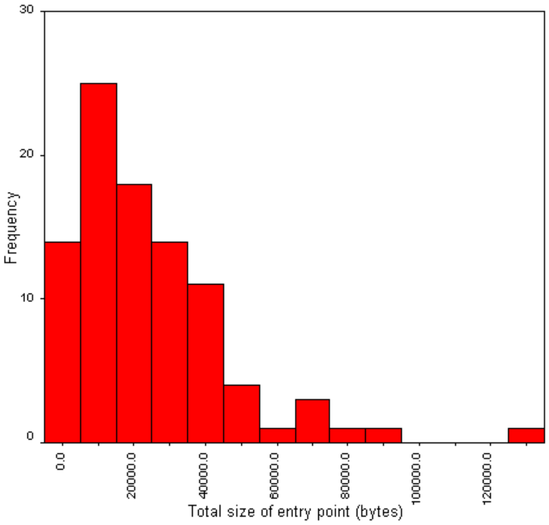


Figure 1 - Frequency Distribution Of The Total-Size Of Entry-Points

Figure 1 shows that the sizes of entry-points is approximately normal. The mean size of a page is about 23Kb, the median is around 19Kb. The trail to the right and the extreme outlier correspond to sites using larger images or more images than most, for example, a large detailed logo.

Figure A8-2 profiles the number of hyperlinks contained within each entry-point.

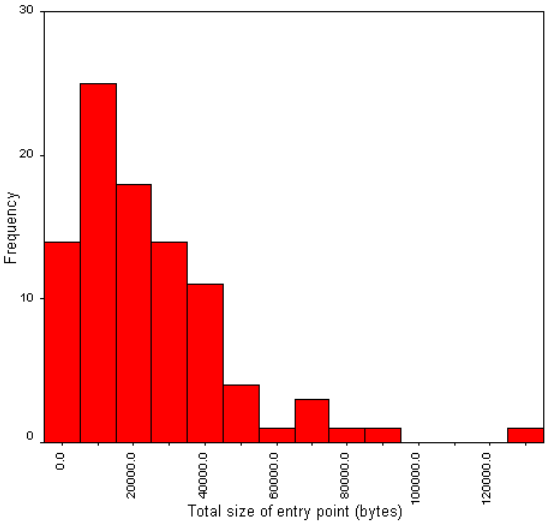


Figure 2 - Number Of Hyperlinks Contained Within Each Entry-Point

The average number of hyperlinks per entry-point is about 13. The outlier represents a cluster of pages that provide links to local branch libraries and other local information.

It will be interesting to see how these profiles change as public libraries gain experience in managing websites. Will, for example, the entry-point sizes grow (making more use of images) or shrink (providing a faster response)?

WebWatching Academic Library Web Sites - June 1998

Following on from UKOLN's analysis of public library websites, we analysed 86 University and college library websites.

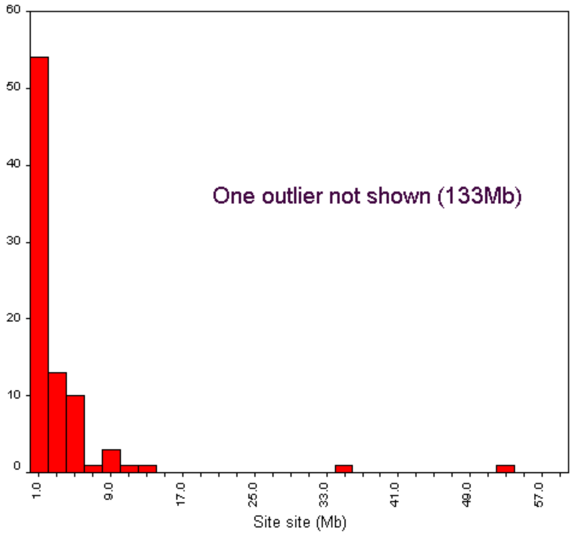


Figure 1 - Size Of Academic Library Web Sites

The average size of an academic library website is around 4.6 Mb. The histogram intervals in Figure 1 are 2Mb, so the chart indicates that most sites are less than 2Mb in total. The smallest site was around 4 Kb (consisting of one page). The largest was 133Mb (not shown in Figure 1 in order to keep the scale manageable). These figures include all resources (HTML, images and so on).

Academic library sites are larger than their public library counterparts and make greater use of web technologies, such as dynamically generated pages. Note that dynamically generated pages from four web sites were excluded from the analysis, due to a hyperlink recursion that was unsuitable for robot traversal.

A more detailed report will soon be available from the WebWatch web area.

Academic and Public Library Web Sites - August 1998

The WebWatch project has analysed both academic library web sites and public library web sites in the UK earlier this year. The reports on these web crawls can be found at the WebWatch web-area, <URL: http://www.ukoln.ac.uk/web-focus/webwatch/>. This column looks at some comparisons between the two analyses.

Academic library sites contain a lot more content. There are more resources (primarily HTML documents and images) within academic library sites than public library sites resulting in an overall larger site size in Kb for academic libraries. The average size of a page (HTML plus inline images) is roughly the same for both kinds of site, so it takes no more time to download a page from either site. The academic libraries do however show a greater use of technologies such as JavaScript and CGI.

We found the structure of academic library sites more suited to robot traversal (apart from their dynamic content) than the public library sites, primarily because there is greater structuring of directories within academic library sites.

Since academic libraries are part of institutions which have been networked for some time, it is not surprising to find their web sites more developed than public libraries. It will be interesting to continue comparing the two communities as public libraries gain experience in providing information on the web.

Academic Libraries and JANET Bandwidth Charging - Nov. 1998

UK Universities form part of the global Internet via a connection to JANET, the UK academic and research network. On 1 August 1998, charging was introduced for each institution's use of JANET transatlantic links. Currently, traffic from the US is charged at a rate of 2 pence/Mb during the hours 06:00 to 01:00 (though many universities will obtain subsidies from HEFCE or DENI).

By identifying domains within links to resources in a web site, it is possible to get a rough idea of the potential of the site to generate transatlantic bandwidth (although, of course, the situation is complicated by the use of caches, users entering URLs directly etc.). WebWatch therefore analysed academic library websites to explore this idea further.

Figure 1 shows the five most popular domains hyperlinked to from academic library web pages. Assuming that the UK domain implies that packets are not routed via the US, the chart shows that most links point to resources that will not directly consume transatlantic bandwidth. If the small number of other domains represent very popular resources, then there is still the potential to attract charging. In this case, such hyperlinks were found relatively deep within the web sites which may suggest that these will not be frequently selected.

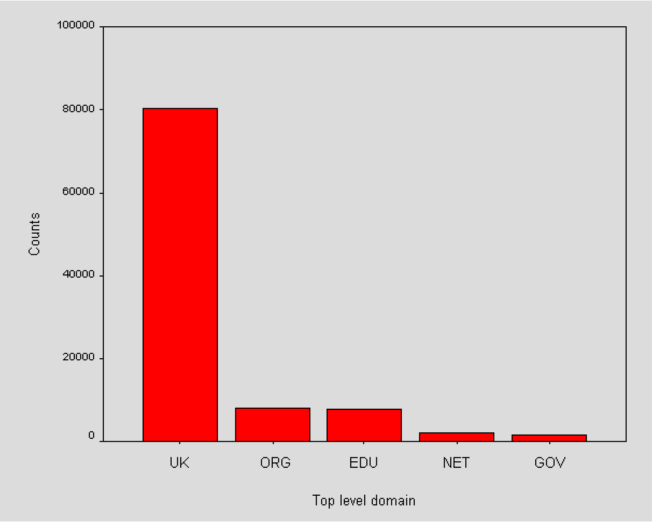


Figure 1 - Top Five Hyperlinked Domains

Final WebWatch News Column - February 1999

This month sees the last WebWatch news column. Although UKOLN hopes to continue with some WebWatch activities, the project has now formally terminated and the final report is available from the WebWatch reports area at <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/reports/`>.

WebWatch analyses have been amongst the first to attempt characterisation of UK Web communities such as public libraries and UK academic institutions. We hope that the analyses have provided useful information for relevant communities as well as laying some foundations for further work.

All reports, articles and WebWatch material can be obtained from the web site, at <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/`>.

Readers may find the various WebWatch services useful to analyse their own web pages, these can be found at <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/services/`>.

Appendix 9 Virtually Inaccessible?

The following article was published in the *Library Technology February 1999 Vol 4 (1)*. We are grateful to Library Technology for granting permission to republish this article.

Sarah Ormes, Public Library Networking Research Officer, UKOLN, s.l.ormes@ukoln.ac.uk

Ian Peacock, Technical Development and Research Officer, UKOLN, i.peacock@ukoln.ac.uk

Making the Library Virtually Accessible

Public Libraries have always tried to provide services which are as accessible as possible to as many users as possible. This commitment to accessibility extends well beyond simply ensuring library buildings are wheelchair friendly. Public libraries provide many specifically tailored services for disabled readers. Audio books, large print material, text readers, housebound services are just some of the services that most libraries support as part of their core activities.

Libraries are now facing a new accessibility challenge as they start to develop an ever increasing number of electronic services. Many public library services have, for example, been developing web sites providing online information about their services. These web sites range from simple opening time and phone number information to providing online access to local history photo collections and the library catalogue. Currently most of these web sites are still in an early stage of development but it is likely that in the short term future they will become an important gateway to many of the library's services. Already other libraries around the world are using their web sites to deliver online reference services, access to library held databases, community information services, gateway information to Internet resources and local history/family history resources. It is obviously important that these new 'virtual' service points need to be as accessible as any other part of the library service.

Accessibility and the Web

The article will explore the issues surrounding making web sites accessible to the profoundly visually impaired. (There are obviously many other accessibility issues to do with the Web which unfortunately this article does not have space to cover. Many of the resources listed in this article's references provide information on this wider issues). Obviously the very visual nature of most web resources make them potentially inaccessible to the visually impaired. Without awareness of this issue and ways that web page authors help make their web pages accessible to the visually impaired there is a danger that the visually impaired will be excluded from public libraries online services.

Web pages can be accessed by the visually impaired through the use of technology which 'reads aloud' web pages in much the same way that Kurzweil text readers can be used to read aloud books. However, for these web readers to work effectively web page authors need to develop their web pages with some simple guidelines in mind. The list below summarises the main points of these guidelines:

- Make use of the ALT TEXT tag when using images in a web page. This tag is used to provide a textual description which can be read by the text reader. If the ALT TEXT is not used someone using a text reader will have no idea what the is represented in the image.
- Image maps are problematic for text readers. It is important if a web page has an image map that the ALT tags are used or separate textual hypertext links are provided.
- Forms should be able to be downloaded and then posted or e-mailed. A large number of text readers have problems with HTML forms. A phone number should also be provided for the user to be able to phone in the necessary information.
- Tables are a somewhat contentious issue. They are frequently used for formatting pages in order to make them more visually attractive. At a very simple level they remain accessible but when they become very complex they do not work effectively with all text readers. Text readers tend to read across the screen in a way that runs all of the text on a line together. If an entry in a cell occupies more than one line the first line of each cell would be read together. One solution to this problem in the future could be through the use of stylesheets [1] but until then tables should be used with care.

- Frames cannot be read by much of the access technology used by visually impaired people. Homepages are recommended to be frame free and provide a link to a text only, non-frame version of the site.
- Material which is provided in PDF formats, a method often used by many government departments to make reports and other papers publicly available, are completely inaccessible to text readers. It is recommended that any document made available in a PDF format should also be available in HTML or ASCII.

More information on how to write 'accessible html' is available on the web and the references section of this article lists several good places to start [2]. This information covers other accessibility issues such as the recommended use of colours and fonts, general design and layout and awareness of keyboard friendliness e.g. for people who cannot use a mouse.

There are a number of tools available on the web which will freely provide a report on the level of accessibility of any web page. One of the most well known is BOBBY [3] which has been developed by the Center for Applied Special Technology (CAST). This is currently available on CAST's American server it should soon be mirrored in the UK on the Royal National Institute for the Blind's (RNIB) server [4].

So How are UK Public Libraries Currently Doing?

We explored the accessibility of UK public library web pages for the visually impaired using the WebWatch robot [5] on the 5th January 1999. The WebWatch robot is a software tool which is used to collect data about web pages. Using the list of UK public library websites available on the UK Public Libraries Page [6] 97 home pages were analysed by the robot. The robot collected data on images and the use of alt text tags, the use of image maps, the use of frames, tables and valid HTML. By analysing this data it was possible to get an indication of how accessible UK public library pages currently are. It should be noted that the robot only looked at the front page of each public library site.

Images

Of the 93 sites that contained images, 72 used the `ALT` attribute within the `IMG` element. The ratio between the number of `IMG` elements in the site with an `ALT` attribute and the total number of `IMG` elements in the site was calculated. This ratio showed how many images in the site had an ALT tag. If a site used the ALT tag with every image the ratio was 100%; for those sites that didn't use `ALT` attributes at all the ratio was 0%. This ratio for each site is shown in Figure one.

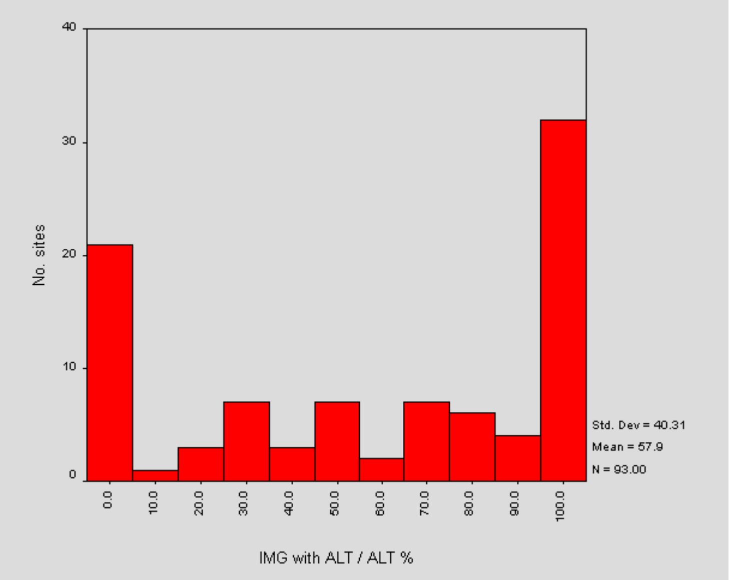


Figure 1 - Histogram of the ratio (images with `ALT` / total no. images)

Figure 1 shows that on over 30 homepages the `ALT` tag is used with every `IMG` element. However on over 20 sites the ALT tag is not used at all. These two cases account for 53 of the sites (over half). The other cases show inconsistent use of the ALT attribute and may be a result of a number of different individuals contributing HTML to the site.

Image Maps

Seventeen sites used client-side image maps and nine sites used server-side image maps. The sites using server-side maps also used client-side maps (this provides back compatibility for clients not supporting client-side maps). For the sites using client-side maps, only two sites consistently used the ALT attribute of the AREA element. The other 15 sites used no ALT attributes in any AREA elements. This means that most of these image maps are not accessible to text readers.

Tables

Seventy seven sites used HTML tables. For these sites, the number of tables per site is shown in Figure 2.

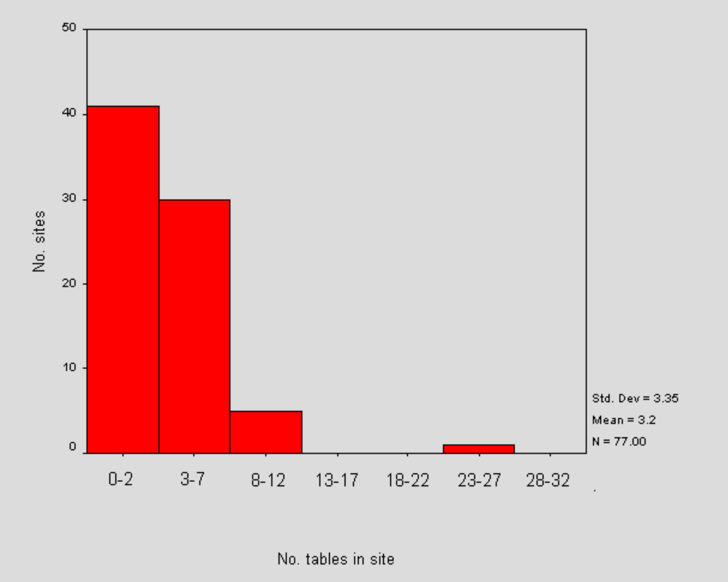


Figure 2 - No. of tables per site

Tables are therefore heavily used on public library web pages. (The outlier in the graph corresponds to a quirk of the robot, which has considered all the libraries hosted under EARL as one. This should be ignored).

Frames

Eleven sites contained framesets. Of these, eight sites used the NOFRAMES element that is displayed in web browsers that do not support frames. All 11 sites used the NAME or TITLE attribute within the FRAME elements.

HTML Validation

All sites were validated against the HTML3.2 DTD. Only 1 site validated correctly. This shows that nearly 100% of all public library homepages contain potentially problematic HTML! As incorrect HTML can cause many problems for text readers this is a cause for some concern.

Summary

The WebWatch robot has shown that the level and awareness of accessible HTML in public library web pages is very patchy. There seems to be a reasonably high level of awareness of the importance of using ALT tags with images but this is not consistent between sites or even within sites. The use of ALT tags with image maps is low and tables are heavily used. A cause for concern is the high level of invalid HTML. This is probably due to the developmental state of these web sites and will no doubt improve as libraries become more familiar with web site technology. The increased availability of HTML editing software e.g. FrontPage and HotMetal will also ensure that valid HTML is created.

In summary UK public library web sites are in the early stages of website development and this is reflected in the general low level of fully accessible sites.

Ways Forward

Fully accessible web pages are still quite rare on the Internet however with more sophisticated software becoming available and a greater awareness of accessibility issues developing this will soon change. In line with many council's policies on ensuring full accessibility to all their services all council services should also be applying this principle to their web services. Already some local governments in the United States are implementing world wide web accessibility standards which all their departments must follow [7]. It is conceivable that such standards could become a legislative obligation of all public bodies [8]. Public libraries need to be aware of these issues and ensure that their high level of commitment to accessibility covers all the media in which they provide and deliver services.

References

[1] W3C Stylesheets Information
<URL: http://www.w3.org/Style/ >

[2] Writing Accessible Web Pages Tips and Guidelines

- Accessibility Quick Reference Guide, Sun Microsystems
<URL: http://www.sun.com/access/acess.quick.ref.html>
- WAI Accessibility Guidelines: Page Authoring
<URL: http://www.w3.org/TR/WD-WAI-PAGEAUTH/>
- Hints for Designing Accessibly Website, RNIB
<URL: http://www.rnib.org.uk/wedo/research/hints.htm>
- Designing More Accessible Web pages, TRACE
<URL: http://trace.wisc.edu/world/web >

[3] BOBBY
<URL: http://www.cast.org/bobby/>

[4] RNIB
<URL: http://www.rnib.org.uk/ >

[5] WebWatch
<URL: http://www.ukoln.ac.uk/web-focus/webwatch/>

[6] UK Public Libraries Page
<URL: http://dspace.dial.pipex.com/town/square/ac940/ukpublib.html>

[7] City of San Jose World Wide Web Page Disability Access Design Standards.
<URL: http://www.ci.san-jose.ca.us/oaacc/disacces.html>

[8] Waddell, C. (1998) Applying the ADA to the Internet: A Web Accessibility Standard.
<URL: http://www.rit.edu/~easi/law/weblaw1.htm>

Appendix 10 WebWatch Dissemination

Presentations

The following presentations about WebWatch have been given.

An Introduction to the WebWatch Project

A talk on "*An Introduction to the WebWatch Project*" was given by Ian Peacock to UKOLN staff on 22 September 1997.

The slides are available at <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/presentations/intro-sep1997/`>

Web Developments For The JANET Community

A talk on "*Web Developments For The JANET Community*" was given by Brian Kelly at the London JANET Regional User Group meeting at ULCC on 13th October 1997. The talk covered emerging web developments and described how the WebWatch project could monitor deployment of such technologies.

The slides are available at <URL: `http://www.ukoln.ac.uk/web-focus/events/meetings/jrug/london-jrug-oct97/`>

Web Developments For The JANET Community

A talk on "*Web Developments For The JANET Community*" was given by Brian Kelly at South West JANET Regional User Group meeting at the University of Bath on 8th October 1997. The talk covered emerging web developments and described how the WebWatch project could monitor deployment of such technologies.

The slides are available at <URL: `http://www.ukoln.ac.uk/web-focus/events/meetings/jrug/sw-jrug-oct97/`>

WebWatching the UK: Robot Software for Monitoring UK Web Resources

A talk on "W*ebWatching the UK: Robot Software for Monitoring UK Web Resources*" was given by Ian Peacock at Networkshop 26 held at the University of Aberdeen in March 1997.

The slides and the paper are available at <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/presentations/networkshop-proc-jan1998/`>

WebWatching the UK HE Community

A workshop on "*WebWatching the UK HE Community*" was given by Brian Kelly at the UCISA TLIG workshop on New Opportunities: Information Services for the Next Millenium at the University of Southampton on 31st March 1998.

The slides are available at <URL: `http://www.ukoln.ac.uk/web-focus/events/conferences/ucisa-tlig98/workshop/`>

WebWatch: Robot Crawls of UK HEIs

A talk on "*WebWatch: Robot Crawls of UK HEIs*" was given by Ian Peacock at the Institutional Web Management workshop held at the University of Newcastle in September 1998.

The slides are available at <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/presentations/webmanage-pres-sep1998/`>

Publications

The following articles related to the WebWatch project have been published in print or web journals and magazines.

How Is My Web Community Doing? Monitoring Trends in Web Service Provision

This paper was written by Brian Kelly and Ian Peacock. It will be published in Journal of Documentation, Vol 55 No. 1 January 1999, pp 82-95.

Showing Robots the Door

This article was written by Ian Peacock and published in the web version of the Ariadne magazine, issue 15, May 1998.

See <URL: `http://www.ariadne.ac.uk/issue15/robots/`>

Robot Seeks Public Library Websites

This article was written by Brian Kelly, Sarah Ormes and Ian Peacock and published in the *LA Record*, December 1997 Vol. 99 (12).

See <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/articles/la-record-dec1997/`>

WebWatching UK Universities and Colleges

This article was written by Brian Kelly and published in the Web version of the *Ariadne* magazine, issue 12, November 1997.

See <URL: `http://www.ariadne.ac.uk/issue12/web-focus/`>

Library Technology News Articles

A brief series of news articles have been published in the *Library Technology News*. These include:

- *Public Library Domain Names*, February 1998
- *WebWatching Public Library Web Site Entry Points*, April 1998
- *WebWatching Academic Library Web Sites*, June 1998
- *Academic and Public Library Web Sites*, August 1998

See <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/articles/lt/` >

Articles

The following articles related to the WebWatch project have been published on the Web.

Report of WebWatch Crawl of eLib Web Sites

See <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/reports/elib-nov1997/`>

Analysis of Links from UK Universities and Colleges Institutional Home Pages

See <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/reports/hei-links-oct1997/report.html`>

A Survey of UK Academic Library Web Sites

See <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/reports/hei-lib-may1998/`>

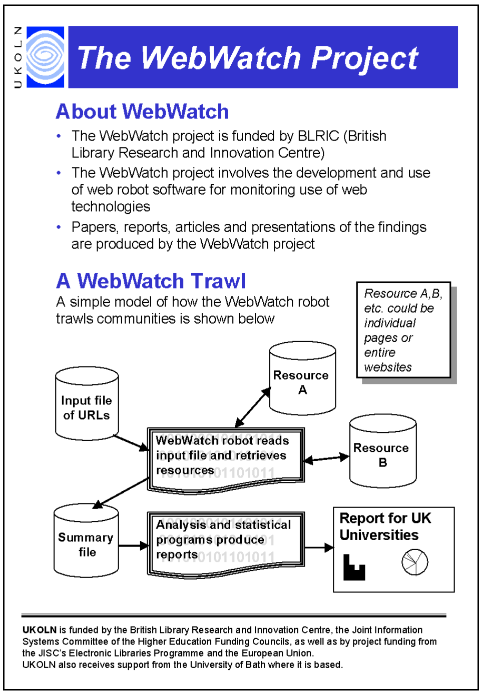
Third Crawl of UK Academic Entry Points

See <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/reports/hei-nov1998/`>

Poster Displays

A poster display about the WebWatch project was given at the Institutional Web Management workshop held at the University of Newcastle in September 1998.

The posters are available at <URL: `http://www.ukoln.ac.uk/web-focus/webwatch/info/` >. A selection of the posters are illustrated below.



UKOLN

The WebWatch Project

About WebWatch

- The WebWatch project is funded by BLRIC (British Library Research and Innovation Centre)
- The WebWatch project involves the development and use of web robot software for monitoring use of web technologies
- Papers, reports, articles and presentations of the findings are produced by the WebWatch project

A WebWatch Trawl

A simple model of how the WebWatch robot trawls communities is shown below

Resource A,B, etc. could be individual pages or entire websites

Input file of URLs

Resource A

WebWatch robot reads input file and retrieves resources

Resource B

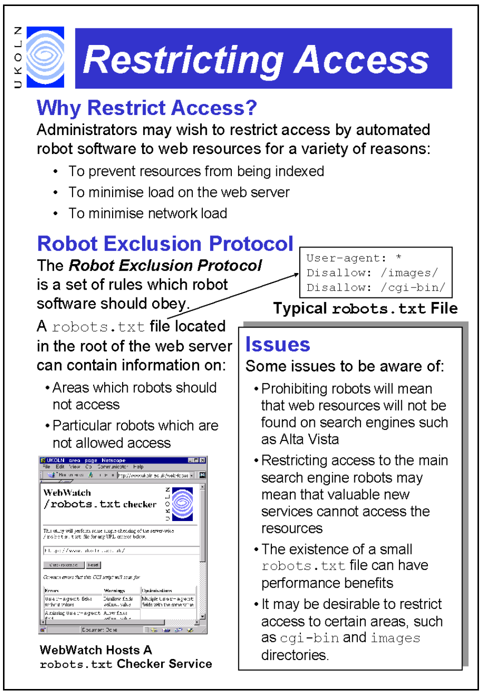
Summary file

Analysis and statistical programs produce reports

Report for UK Universities

UKOLN is funded by the British Library Research and Innovation Centre, the Joint Information Systems Committee of the Higher Education Funding Councils, as well as by project funding from the JISC's Electronic Libraries Programme and the European Union.
UKOLN also receives support from the University of Bath where it is based.

Figure A10-1a About WebWatch



UKOLN

Restricting Access

Why Restrict Access?

Administrators may wish to restrict access by automated robot software to web resources for a variety of reasons:

- To prevent resources from being indexed
- To minimise load on the web server
- To minimise network load

Robot Exclusion Protocol

The ***Robot Exclusion Protocol*** is a set of rules which robot software should obey.

```
User-agent: *
Disallow: /images/
Disallow: /cgi-bin/
```

Typical `robots.txt` File

A `robots.txt` file located in the root of the web server can contain information on:

- Areas which robots should not access
- Particular robots which are not allowed access

UKOLN area page Netscape

WebWatch
/robots.txt checker

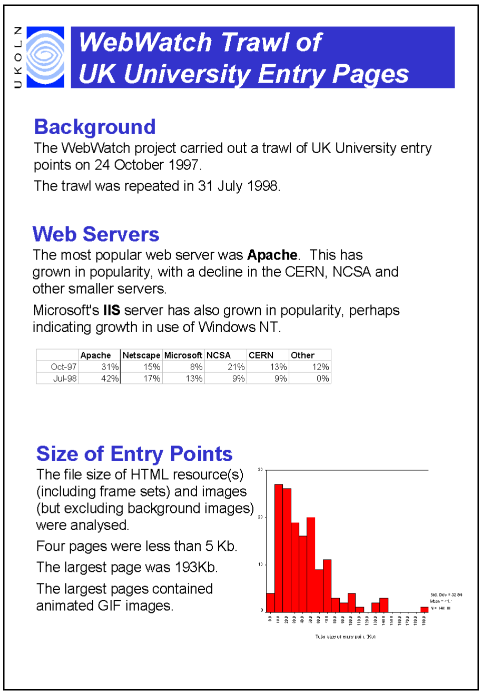
WebWatch Hosts A `robots.txt` Checker Service

Issues

Some issues to be aware of:

- Prohibiting robots will mean that web resources will not be found on search engines such as Alta Vista
- Restricting access to the main search engine robots may mean that valuable new services cannot access the resources
- The existence of a small `robots.txt` file can have performance benefits
- It may be desirable to restrict access to certain areas, such as `cgi-bin` and `images` directories.

Figure A10-1b Restricting Access



UKOLN

WebWatch Trawl of UK University Entry Pages

Background

The WebWatch project carried out a trawl of UK University entry points on 24 October 1997.

The trawl was repeated in 31 July 1998.

Web Servers

The most popular web server was **Apache**. This has grown in popularity, with a decline in the CERN, NCSA and other smaller servers.

Microsoft's **IIS** server has also grown in popularity, perhaps indicating growth in use of Windows NT.

	Apache	Netscape	Microsoft	NCSA	CERN	Other
Oct-97	31%	15%	8%	21%	13%	12%
Jul-98	42%	17%	13%	9%	9%	0%

Size of Entry Points

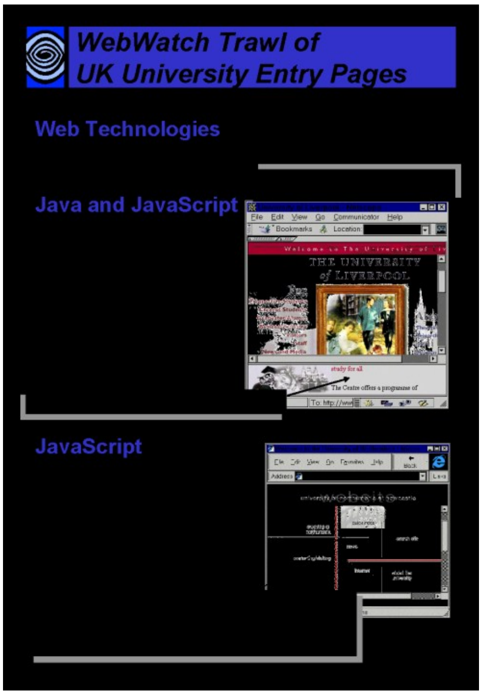
The file size of HTML resource(s) (including frame sets) and images (but excluding background images) were analysed.

Four pages were less than 5 Kb.

The largest page was 193Kb.

The largest pages contained animated GIF images.

Figure A10-1c UK HEI Trawl



WebWatch Trawl of UK University Entry Pages

Web Technologies

Java and JavaScript

JavaScript

Figure A10-1d UK HEI Technologies