CRAWLING THE CONTROL SYSTEM *

T. Larrieu, JLAB, Newport News, VA 23606, U.S.A.

Abstract

Information about accelerator operations and the control system resides in various formats in a variety of places on the lab network. There are operating procedures, technical notes, engineering drawings, and other formal controlled documents. There are programmer references and API documentation generated by tools such as doxygen and javadoc. There are the thousands of electronic records generated by and stored in databases and applications such as electronic logbooks, training materials, wikis, and bulletin boards and the contents of text-based configuration files and log files that can also be valuable sources of information. The obvious way to aggregate all these sources is to index them with a search engine that users can then query from a web browser. Toward this end, the Google "mini" search appliance was selected and implemented because of its low cost and its simple web-based configuration and management. In addition to crawling and indexing electronic documents, the appliance provides an API that has been used to supplement search results with live control system data such as current values of EPICS process variables and graphs of recent data from the archiver.

INTRODUCTION

Although most web-based tools used by operators and support personnel at Jefferson Lab provide search functionality, staff have expressed ongoing frustration about the difficulty of finding information using such tools. The narrow application-specific scope of each search dialog requires the user to determine first where to search most profitably and if he or she guesses incorrectly, perhaps to repeat the same search on multiple systems. To address this problem a global search engine was implemented to searches to span all relevant control system applications and information sources.

The tools required to implement cross-application search include a crawler to find documents, an indexing engine to parse and score those documents, a query engine to find and retrieve relevant information from the index, and finally, a front-end interface for users.

An overwhelming selection of tools that perform some or all of those four functions can be found at websites such as www.searchtools.com, where more than 170 packages are listed. However, upon closer inspection, many choices lack either adequate documentation or support or lack critical features such as the ability to index proprietary Microsoft Office file formats. Of the commercial packages listed at searchtools.com, one of the least expensive and most fully featured is the Google Mini Search Appliance.

IMPLEMENTATION

The Google Corporation ships its search tools literally in the form of an appliance, a 1U rack-mountable computer system preconfigured with the necessary software (Figure 1).

Figure 1: The search appliance (see arrows).

To begin using the appliance, one simply unboxes it, places it on the network, and proceeds to configure search parameters via its web interface (Figure 2). No Operating System administration is required, nor even possible.

Figure 2: Web based configuration and management.

* Notice: Authored by Jefferson Science Associates, LLC under U.S. DOE Contract No. DE-AC05-06OR23177. The U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce this manuscript for U.S. Government purposes.

Table 1: Information Sources to be Crawled

Source	***Content Type***
Software, Engineering Group Wikis	HTML topics with PDF and Microsoft Office formatted attachments
Departmental Web Sites	HTML with links to PDF, Microsoft Office format documents
Software Program Documentation	HTML (the output of Doxygen, Javadoc, PHPDoc, etc.)
Operational Procedures	PDF
EDL, Alarm Handler Config, IOC Signal.list, Hardware Address files, etc.	Plain Text
Electronic Log, OPS Problem Reports, Downtime Logs	HTML (formatted output of Database content)
ATLis Work Plans, Software Test plans, Beam Test plans	HTML (formatted output of Database content)
Lessons Learned	HTML (formatted output of Database content)
Operations Bulletin Board	HTML (content of online phpbb discussion forums)
Accelerator Engineering Drawings	PDF (rendered from CAD software)

Data Sources

The first and most important configuration step was to identify data sources and the URLs to be used as the crawler's starting point. Table 1 lists the data sources to be crawled by the control system search engine. In some cases, such as the departmental web pages and the wikis, the home page made a suitable starting point, and simple regular expressions were sufficient to specify the URL patterns to be crawled and indexed.

In other cases, it was necessary to write simple scripts to generate URL listings for the crawler. In the case of the electronic logbook, for example, a script was written to feed the crawler only the most recent 12 months of entries. This was done because the cost of Google Mini Search Appliance is proportional to the number of documents it is licensed to index ($3,000 for 100,000 documents, $6,000 for 200,000 documents, etc.)

To prevent searches from returning obsolete files from old versions, the script that feeds the crawler hardware address files belonging to low level IOC application selects only the files from the most recent version.

Search Tuning

Several iterations were required to fine-tune the quality of the search appliance's document index. The "crawl diagnostics" reports generated by the search appliance revealed obsolete and forgotten web pages (e.g. for disused software) that were deleted to prevent future confusion. The same reports also identified numbers of pages with broken links that had to be fixed.

Further refinement of the search engine output was achieved with source-based results-biasing. Within the appliance's web interface it is possible to specify that documents matching certain URL patters should be ranked higher when search results are returned. For example, search results from the authoritative Operational Procedures repository should be given precedence over search results from Wiki documents or OPS discussion forums.

Web Server and Application Tuning

Running the web crawler for a period of time revealed that some tweaking would be required to optimize the response of web servers and web-based applications to being crawled.

Applications that serve as database front-ends (Electronic Logbook, Software Test plans, etc.) were modified to output database record timestamps into "Last-Modified" HTTP headers as part of dynamic page generation. Then on subsequent crawls, when presented with an "if-modified-since" header request by the crawler, the application can simply respond with "HTTP/1.1 304 Not Modified" and avoid unnecessary database workload.

To accommodate the increased load induced by the crawler without sacrificing performance for other clients, it was helpful to increase the ServerLimit and MaxClients parameters in the apache httpd.conf configuration files.

To allow clients to launch EDM to view the screen definition files in search results, a new mime type 'Application/edm' was mapped to *.edl files in the apache mime.types file. Correspondingly, an entry was added to /etc/mailcap on all control system Linux workstations to launch the program edmRun whenever such a document is retrieved by the web browser.

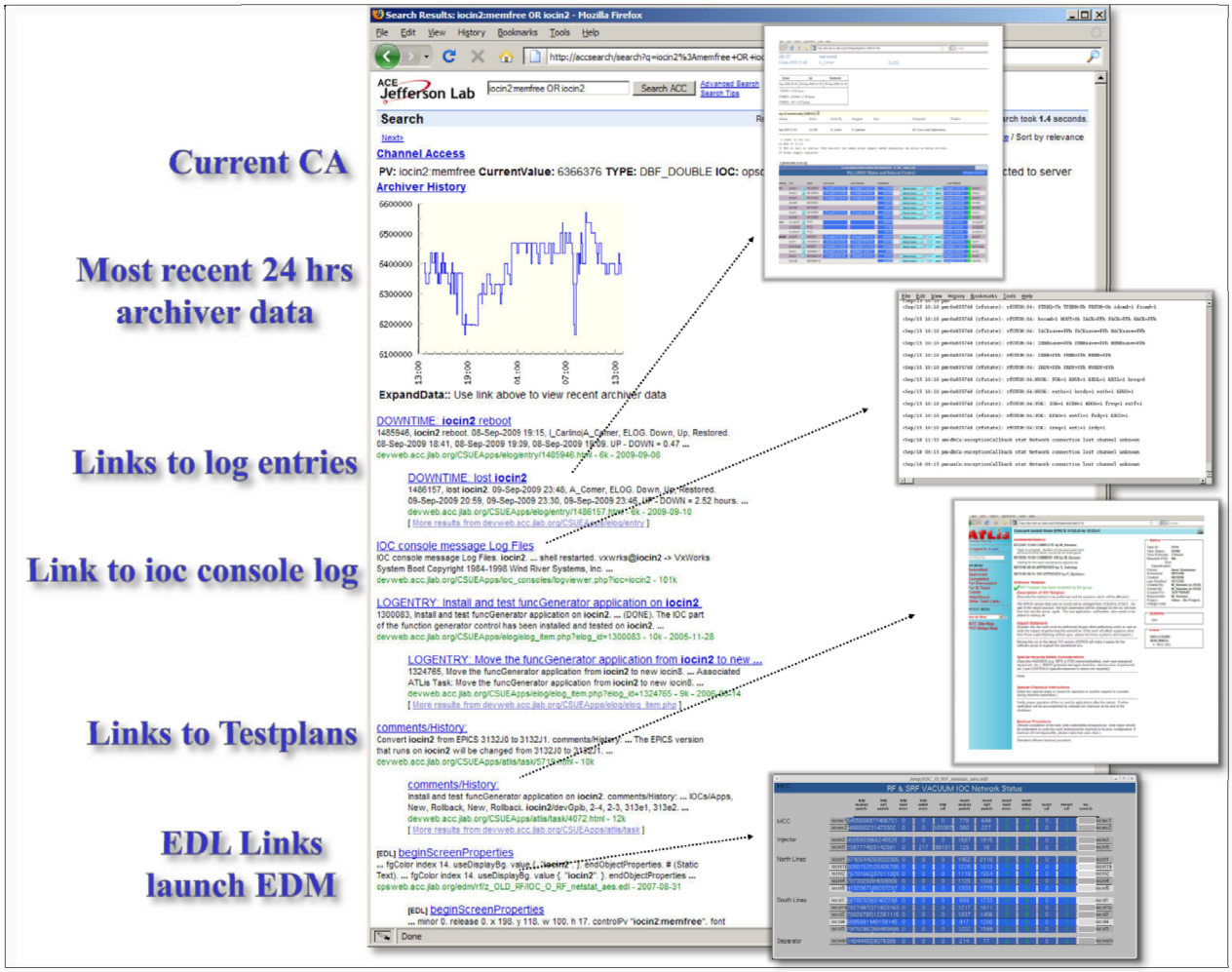

Figure 3: A single web search will now retrieve information from many applications and information sources, and can include live data from external sources such as channel access and the archiver.

External Data

The search appliance provides a programming interface ("Google OneBox API" [1]) that allows the search appliance to request real-time data from external sources. External modules can be invoked for every search, or for only those queries that match a certain regular expression. In either case the external sources are provided with the search query string and expected to respond quickly (< 3 seconds) with results in the proper XML format.

Similar to the manner in which google.com uses the API to retrieve advertisements, weather, and airline flight schedules, so can it be used to supplement control system search results with live EPICS data. Two API modules have been written for the JLAB control system search. One module retrieves current channel access data when a user searches on a PV name, while the second generates a graph of the past 24 hours data from the archiver.

CONCLUSION

The results of the search engine implementation are illustrated in Figure 3. It is now possible with a single search to retrieve results that span numerous applications and control system information sources.

REFERENCES

[1] "Google OneBox for Enterprise Developer's Guide", http://code.google.com/apis/searchappliance/documentation/52/oneboxguide.html.