

The Internet Archives Project

The Internet Archives Project is an ongoing research project to demonstrate the feasibility of collecting and providing access to complete versions of the umich.edu domain. This term, under the guidance of Margaret Hedstrom, Ursula Arnold and I built on the work of past semesters. While Arnold focused on the appraisal side of the equation, I put my programming background to use on the technical side. In this endeavor I was largely successful. The result is a local version of the Internet Archive's Wayback Machine and thorough documentation of the technical procedures that made it possible. The online archive can be accessed here: <http://www.si.umich.edu/mirror>.

Throughout the process, an important goal was to create a record of our procedures. The online manual, http://www.si.umich.edu/mirror/how_to, provides the command files, scripts, and table structures of every phase of the project. It also serves an important function as an example of what it would take to duplicate our efforts at another institution. In fact, Eric Celeste at Internet2 was involved in a parallel project to ours, and by sharing our documentation both projects have benefited.

Three Steps

Early in the term we identified three steps in archiving a website. First, the information must be gathered from the website using a spider or crawler. Second, the information must be stored locally and processed. Third, the information must be made accessible. Each of these steps provided unique challenges.

Gathering Information

In early 2004 the Internet Archive performed three crawls of the umich.edu domain for the benefit of our project. The result was over 200 GB of information, which was delivered to us by mail on an external hard drive. As the old axiom goes, “Never underestimate the bandwidth of a station wagon full of tapes.” Along with this information came extensive crawl logs, which formed the basis of sampling analysis in previous years.

To augment this information, we wanted to run our own crawls of the umich.edu domain. While there are many suitable crawlers available, we settled on using Heritrix, the Internet Archive’s open source Java crawler. Using Heritrix gave us backward compatibility with the 2004 crawls and provided extensive metadata as well.

One challenge in using resources from the Internet Archive has been the difference in operating systems. While most of the computing resources at the School of Information are Windows or Macintosh machines, the Internet Archive uses Linux machines. This meant that the drive containing the 2004 crawls could not be read locally. After consulting with other students in the program, notably James Sweeney and Alec C. de Baca, we were able to find a patch for Macintosh that allowed us to read the Linux drive.

Another problem between Linux and Windows was in running the Heritrix program. The Internet Archive offers advice for Linux users, but all others are on their own. For a novice Java programmer like me, this presented a significant challenge. The solution to the problem was to create a Windows Command File which would execute the

necessary commands to start the Heritrix engine. Once that hurdle was crossed, it was a simple matter to access the web interface and perform crawls. This success formed the basis for the first online documentation of our procedures, so that future archivists would not need to go through the same discovery process.

Process & Appraise

Once we had gotten Heritrix to run locally, we were confronted with the question of what to do with the results. Heritrix produces ARC files, which are essentially binary concatenations of all the files downloaded by the web crawler, with metadata included right in the file. There are many ways to store information from a crawl, and the ARC file is probably the most elegant I've encountered.

Over the summer of 2004 I wrote my own crawler in Perl, and faced the same problem. My solution was to create a local directory structure that mimicked the directory structure of a website I was crawling, so that a URI such as `http://www.si.umich.edu/~nnnick/index.html` would be stored in my computer as `C:/crawls/www.si.umich.edu/~nnnick/index.html`. This worked well for small crawls, but larger crawls quickly got out of hand. The challenge of creating and storing tens of thousands of files in a complex directory structure taxed my system resources. Over ten percent of the file space was devoted simply to maintaining the integrity of the directory structure.

ARC files, in contrast, are concatenations of many files, so a single 100 megabyte file might contain thousands of smaller files. As the Internet Archive notes, such files are

much less taxing on file systems. In addition, because the complete, original binary file is stored with all its metadata within the ARC file, the file format offers a simple guarantee of authenticity and is itself easily archivable.

While Heritrix produces internally complete ARC files, it is up to the end user to find a way to extract the information. The Internet Archive recommends maintaining an external index of the files within each ARC file, so that the files can be programmatically extracted. This is how we proceeded with our first Perl script, in a simple attempt to “unpack” an ARC file into its constituent parts.

Space Issues

A quick analysis of the 2004 crawl revealed that there was significant duplication of files within the ARC files from the three different crawls that were performed. Within a single crawl a URI would be visited only once, but might appear with identical information in all three crawls. Therefore, I wrote a script to produce Optimized ARC files, using the extension “.arco” to distinguish them from the raw output of crawls. The results were ARCO files that contained only new and changed content.

While we had hundreds of gigabytes of offline storage space, our online storage space was constrained by SI Computing resources to fewer than 30 gigabytes. An analysis of our first local crawl by Arnold showed that fewer than ten percent of the files downloaded represented over ninety percent of the file size of the ARC files. These included presentations, movies, executables, and other rich media. In the interest of providing access to as many web pages as possible, it was decided to eliminate files

over one megabyte from the online version of the archive when creating ARCO files.

This somewhat arbitrary selection process enabled us to eventually mirror two and a half crawls of the entire umich.edu domain in 15 gigabytes of space. The files that were omitted remain offline in their original ARC files, waiting for the day when server space becomes cheap enough to post them.

Provide Access (Mirror)

Once we had the results of crawls from 2004 and 2005 in ARCO files with an external index, it was a relatively simple matter to write code that would display these files to users over the web. The index of the files was entered into a MySQL table, and included information such as the URI, date archived, and mime type.

Using PHP, the database could be searched for URIs matching a particular pattern, and the proper files could be returned to the user over the internet. For binary files such as JPEG images this was a simple matter of reading the proper portion of the ARCO file, appending the proper mime type to the header, and spitting the data back to the user.

For HTML, more complex measures were required. Building on the JavaScript employed by the Internet Archive's Wayback Machine, we used PHP to rewrite the HTML pages before returning them to the user. First, a base tag was added to the header to make all links in the file into absolute links pointing at the original URI. Next, JavaScript was added to the end of the file to dynamically change the absolute links to point to the archived versions of those files, rather than to the originals. Finally, code

was added to display information about the archive in the upper right corner of the archived page.

This complex rewriting was necessary for several reasons. Of primary importance was maintaining the look and feel of the original page. In order to do this, all the links to images and other visual content had to point at the archived versions rather than the original.

By making internal links point to the archived versions, we also preserved the most important feature of the original website, its linked structure. While an archivist could spend a lifetime arranging and describing the pages from just one crawl, thousands upon thousands of people have already invested time in the very same endeavor. The links between pages are the result of their effort, and preserving them is thus of critical importance.

Challenges & Implications

It is easy to look back on the technical challenges that were surmounted and forget the heartache and suffering that preceded success. Hopefully our documentation will provide a guide through the thicket for other professionals. There are some challenges that were not overcome, however, and these tend to carry with them important philosophical implications.

Server Space Limitations

We quickly found that it was much easier to gather information than it was to re-present

it. At an early stage of the process, I attempted to put the logs from the 2004 crawls into a MySQL table, only to run up against the 4 gigabyte size limit for tables. Even without this limit, the queries on such a large table took several minutes each to execute, which is unacceptable when delivering content over the internet.

While the metadata stretched the MySQL space, the ARCO files stretched the resources of the server. The day after I finished processing the ARCO files, Margaret Hedstrom was contacted by SI Computing. They had noticed a radical increase in the size of our account, and wanted to know how much longer we would need the space.

Thus, while we may have the resources to archive several complete copies of the umich.edu domain offline, we do not have the resources to put all of this information online. However, storage space continues to get exponentially cheaper. One megabyte of storage space cost one hundred dollars in 1985. By 1995 it was down to one dollar, and today that same dollar will buy one thousand times as much memory.¹ It is likely that the entire umich.edu domain from 1995 would fit within the space allotted to a single undergraduate in 2005. This suggests a simple solution; wait. While we should continue to gather information, we can hold off on making it accessible until the online computing resources come within reach.

¹ <http://www.americanscientist.org/template/AssetDetail/assetid/14750?print=yes>

Technical Limitations

Despite our success in archiving many pages, some pages simply don't archive well. These are often pages that include complex interactivity based on JavaScript or Flash. While it is a relatively simple matter to rewrite HTML code to keep pages from "breaking," it is several orders of magnitude more complex to do the same for JavaScript or Flash.

It is not a coincidence that pages that are designed with accessibility in mind are also easily archived. By relying on Cascading Style Sheets to determine the appearance of the page, rather than scripts and tables, a website can be much more accessible to the blind and visually impaired. After all, a screen reading program runs into the very same problems in parsing a complex web page that a web archive does. In arguing for archivability, we have the powerful argument for accessibility on our side.

Another hopeful point is that many design decisions which can break the archive are simply that; design decisions. The same look and feel can be achieved using code that is much more archivable. As with many other types of electronic records, the archivist must get out of the office and work with the creators to ensure the life of their content in the future. Archivists and records managers must work with website administrators to ensure that their designs are archivable, without sacrificing functionality.

The Future of the Web

Unfortunately, the most cutting edge applications on the web seem to be moving in the

opposite direction from accessibility and archivability. The latest trend in website design is asynchronous server interaction, termed Ajax (for Asynchronous JavaScript and XML) by Jesse James Garrett.² Under this model, constant interaction with the server creates the look and feel of the website.

Google Maps is the preeminent example of this trend. Instead of loading a page and waiting for the user to do something before loading the whole page again, JavaScript talks to the server continuously in the background. While the user is looking at a map, the JavaScript is loading information for adjacent regions. This way, when the user wants to scroll, the web browser already has the information ready to display. Each user action can result in multiple interactions between the browser and the server that the user never sees.

The result of Ajax design is a seamless user experience, without the usual wait while a page loads. While this is a wonderful development from the perspective of human computer interaction, it poses serious challenges for the archivist. We currently have no practical way of archiving this sort of interaction, and traditional web crawlers like Heritrix are utterly useless for this task. Even as we master the demands of current technology, the world speeds ahead.

² <http://www.adaptivepath.com/publications/essays/archives/000385.php>

Conclusions

The single greatest conclusion I have drawn from this project is that accessibility and archivability go hand in hand. A website that is accessible and user friendly is generally archivable, and archivists must work with website administrators to encourage these types of designs.

The technical challenges to archiving traditional websites are well understood by now, and within reach of any organization with significant computing resources. While we couldn't mirror all the content we gathered over the past two years, we have been able to store it safely for a future when the resources to provide access will be available. In the meantime, technology continues to change, posing new challenges for archivists.