Candle Project

A Conceptual design for Establishing a Web Site Archive

By

Leo D. Geoffrion, Ph.D
Skidmore College

It is better to light one candle
than to curse the darkness.

Introduction

As the web becomes an increasingly central component to the overall campus experience, it is clear that the site and its contents should become part of the school's archives. Along with the College catalog, school newspaper, yearbook, and other similar documents, the web will provide historians with a glimpse of academic life in the 21st century.

On first glance, this appears to be little more than a technical task to record a snapshot of the entire web site onto durable media such as CD-ROM or DVD. Closer examination, along with a review of the literature concerning electronic archiving, reveals that this task is surprisingly daunting for small colleges with limited resources.

To begin with, the web is an information environment whose boundaries are – at best – very fuzzy. It is not self-evident how many linked resources need to be included in the site capture, how many different dynamic html pages should be included or how much of the actual server software must be included in order to render correctly the archived pages.

Meanwhile, archivists have correctly pointed out that even a brute-force snapshot of the entire web site may not be sufficient if one does not also record the metadata that establishes the overall context surrounding these many pages. For example, without metadata, future scholars may not have difficulty distinguishing between critical web contents and experimental items that were never intended for public use. Similarly, many faculty-composed course sites use the course title without any good information to identify where this course fits within the overall curriculum.

While the HTML structure has long supported header metatags, few amateur web authors employ them consistently because they are largely invisible and irrelevant to everyday page authoring. This means that many people may forget to enter the data when creating a web page. Equally important, the author may forget to update the metadata when updating or revising the page. Thus, even when metadata is present it may not be an accurate or current

record of the context surrounding this page. As a result, metatags remain a largely unexploited resource for capturing important metadata.

These problems are particularly acute in small colleges where it is clearly impractical to hire professional archivists to maintain the accuracy of the web information. At best, the typical small-college might hire one archivist whose role has traditionally centered on print media. The typical small college web site may contain 25,000 or more HTML pages, and this number grows to about 250,000 items if one includes all of the supporting images, style sheets, and related page components. Clearly, this workload cannot be added to the duties of existing professional staff.

Web page creation in higher education is inherently a decentralized process. While the school may hire one or more web specialists, or may outsource the development of its top level pages to a design consultant, most of the detailed content is created by dozens of staff working in the individual offices and academic departments. Furthermore, departments often delegate the actual page creation to student workers whose term of employment may span only a few weeks or months. Thus, it is not feasible to delegate the metacontent generation to the page authors unless the process can be highly automated and self-evident to novice authors.

At this point it is very tempting to "curse the darkness;" to declare that the task is impossible and to forsake any attempt to implement web archiving.

This proposal instead seeks to light one candle, by developing a modest technology proposal that may lead to a feasible protocol for preserving web information. This is only one candle, in that the proposed actions do not constitute a comprehensive or definitive solution to this important problem. It is also a solution that is tailored to the specific culture that characterizes our type of institution.

This proposal assumes that the most effective time to capture critical archival information is at the time of object creation or revision instead of attempting to reconstruct this information at a later date when the creators may be no longer available or may have not remember the details of the previous projects.

STEP 1: Establish a list of monitored pages.

It is neither feasible nor desirable to maintain accurate metadata on every digital resource that constitutes the web site for a typical college. Instead, attention should focus on the index page for each individual directory since this page typically establishes the connections among the other pages in that directory.

The initial list of such pages can be generated via an automatic disk sweep, can consist of a manually entered list, but the most effective will likely be a combination of the two methods.

The first pass for the list of monitored pages can be generated automatically via one of the system-specific file search commands. For example Unix provides a powerful find command that can quickly build a list of all index files within a file tree. Comparable commands exist for Windows and MacOS systems.

This automatic list should be supplemented with a list of files to exclude or to include. Some examples of exclude files might be the index pages for student sites or portions of the web site that are known to have no archival value. Similarly, include lists can add the names of key web pages that deviate from the standard index file naming or other site files that are known to be worth archival monitoring.

Step 2: Periodic Scanning.

A script on the web server then scans through each of the files identified in Step 1 to determine whether or not its contents have changes since the previous sweep.

At the simplest level, the MD5 file hash can be used to compare the current file to its previous edition. MD5 has been shown to be highly efficient for detecting any change to the file contents in a manner that is very immune to tampering.

Unfortunately, MD5 is too efficient, and will mark the page as modified if even a single byte within the file has changed. For example, a trivial change such as the addition of a single space or return may have no impact on the rendering of the final page, but will still trigger an MD5 hash mismatch. Hence, it is desirable to implement a more sophisticated change detection protocol that can distinguish between trivial and significant modifications. The Prism Project at Cornell University is currently experimenting with such a tool, but has not yet made it available for general distribution. This tool should be adopted when it becomes available.

Step 3: Reporting the modification.

When the scanner detects that one of the scanned pages has been modified, it e-mails an automatic message to the last known manager of that section of the web site. The message contains a link to a web form that contains the following user options.

1. The recipient can indicate that this page is no longer managed by this person and can refer the scanner to a different individual. The corresponding database entries are updated automatically and the e-mail is sent to the new author.

2. The recipient can indicate that the page is "in transit", (still being modified). In that case, the scanner is set to ignore the page for a preset time period (e.g. 1 week). At the end of that time period, the scanner retransmits the reminder e-mail.

3. The recipient can indicate that the change is trivial and therefore no further information update is necessary. In this case, the scanner simply resets the MD5 hash key.

4. Finally the maintainer can indicate that the update is complete and significant. The form handler then presents a second form where the maintainer can update the page metadata.

When generating the metadata update page, the software scans the web page to extract the previous metadata values and displays them as preset values in the data entry form. In some blank fields, the script can insert default values unique to that school. The maintainer can

then update the information as needed. When completed, the script automatically inserts the metadata as header tags in the monitored page and triggers the final stage of processing.

DCDOT, the Dublin Core Metadata Editor, at http://www.ukoln.ac.uk/metadata/dcdot/ provides a good model for how this process might result in a self-evident data-entry task. In this case, since all work is taking place within the campus web server, it is feasible to automate DCDOT even further by automatically updating the metatags into the web page.

Step 4: Final Processing.

The metadata update triggers the final stage of the page monitoring. This includes the following steps.

1. The MD5 change key used in the second stage is reset to mark the revised page as the new comparison standard.

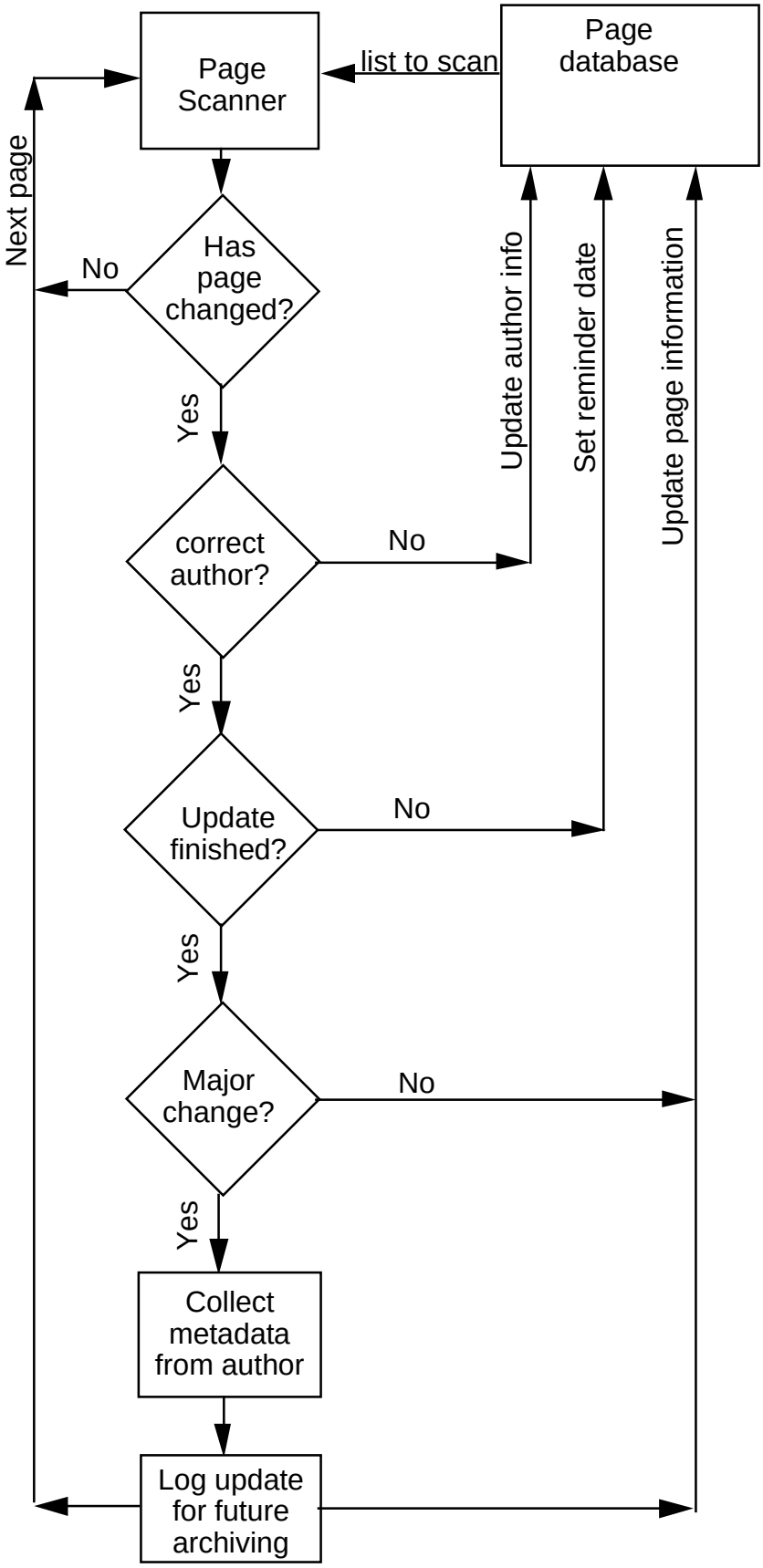
2. The page modification is logged in a manner that the campus web managers can use to provide summary data on the timeliness of the campus web site and the frequency of page updates.

3. The log file can also trigger supplementary archival activities to move the pages to more durable storage. For example, the college might choose to generate a periodic CD-ROM of all sites that have been updated since the last cycle.

Appendices

Schematic Flow Chart

Metadata Guidelines


Page Scanner
list to scan
Page database
Next page
Has page changed?
No
Yes
Update author info
Set reminder date
Update page information
correct author?
No
Yes
Update finished?
No
Yes
Major change?
No
Yes
Collect metadata from author
Log update for future archiving

Appendix 2

Metadata Guidelines

The metadata form should cover the key information needed for efficient use and archival storage of this page. Wherever feasible, the metadata should be based on established standards such as the Dublin Core since the use of these standards increases the likelihood that the data can be imported into more sophisticated archival services that may emerge in the coming years. Dublin Core already includes a rich set of tags for logging information concerning the author, owner, creation date, modification history, and other similar contextual information.

In addition to the standard information, the metadata tags can include:

KEYWORDS: The tags used to guide correct page placement in the various search engines that may index the site. Since most amateur page authors forget to insert this metatag, its inclusion here should improve the accuracy of site searches.

ROBOT EXCLUSION: The tags can include those used to signal to the search engines whether or not to index this site or the pages below this one. See http://vancouver-webpages.com/META/metatags.detail.html#robots for discussion of this metatag and its use to control search engines.

LOCAL TAGS: A specific college may wish to include special tags, such as an expiration date or other information that is subsequently used for system administration. In general, this practice should be discouraged since these tags will be the most likely ones to cause problems whenever the information is migrated to a future archival system.