DEVELOPMENT OF A CASE STUDY "MULTI-SITE" ON THE WORLD WIDE WEB

BRIAN R. JOHNSON AND MARIETTA S. MILLET

University of Washington
Department of Architecture
BOX 355720
Seattle, WA 98195-5720
USA

Abstract. Seeking to enhance the availability and quality of architectural case studies, we examine the web, believing that it may help to simplify authorship, distribution, and navigation of a catalog of case studies. A disk-based multi-media prototype from an earlier effort has been converted to the web and generalized to create a template. Warehousing and navigation of multiple case studies forms the main focus of the current project. Two existing models, monolithic web sites and web rings, are considered and rejected. A third approach is developed which promises to provide a low-budget low-overhead infrastructure within which to house an indefinite number of case studies, while permitting free-form browsing of the collection. The approach which we have developed creates what we call a "multi-site".

1. Introduction

Case studies are frequently used as a vehicle for teaching in architecture. This use takes the form of one or both of two modes. In an undergraduate or introductory course students might read about several different construction strategies for minimizing heat gain in hot climates. In a history course they might study different Palladian villas or Frank Lloyd Wright houses. In the more advanced courses and seminars the students might be asked to create a new case study, possibly utilizing an important piece of nearby architecture, as a means of demonstrating mastery of an area of analysis, such as environmental factors or formal decomposition.

In the ideal world these intellectual products would become part of an ongoing enrichment and enhancement of knowledge. The newly authored case studies would become part of a general resource for beginners and advanced users alike to consult when needed. Unfortunately, few student projects are entirely polished when first submitted. Often a second or third iteration of refinement may improve them significantly.

One format in which case studies might be published and distributed is that of computer files, particularly multimedia computer files. In the early 1990's the original *ArchiMedia* project was begun at the University of Washington. The aim was to create a collection of high quality case studies using disk-based multimedia (Millet, *et al.*, 1991, 1993). Faculty, working together with their students, developed a prototype case study. The project was created as a prototype, with carefully considered categories and an extensible structure. It was received enthusiastically by the audiences to which it was presented, validating the use of both multimedia and the case study approach.

Unfortunately, efforts to secure funding from traditional publishing sources for distribution on CD-ROM, or for initiating a periodic CD-ROM publication, were not successful. The emergence of the world wide web raised the possibility of "self-publication" and in 1996 a project was undertaken to convert the disk-based multimedia prototype to a web-based model (Johnson & Millet, 1997).

2. Assessment of the World Wide Web as a Medium

Documents on the web are written using HyperText Markup Language (HTML) "tags". Some tags are used for formatting, while others provide the basis of links between documents. Links between documents are created using "Uniform Resource Locators" (URLs), a means of specifying how to retrieve files on the Internet. Using any text editor it is relatively easy to create content and WYSIWYG (what you see is what you get) editors are becoming more common, but management of a site's links remains a difficult problem in most environments.

There are three common structures for serving content on the web, each of which we examined as a possible model for serving case study material. Most web sites utilize a mixture of the three.

2.1 TRADITIONAL WEB SITES

The most common web site structure consists of a single collection of HTML documents linked together in some logical fashion, usually hierarchical, but often with cross-linkages between pages. Unless the content is completely static, or automated editing tools are available, the creation and maintenance of this kind of web site involves significant amounts of time spent adjusting the links.

Previous experience had demonstrated how little funding is available for such activities even when content is high quality. While we might manage a case study or two of our own, we felt that the challenges of creating, maintaining, and serving a large collection of multi-media projects on the web would probably prove insurmountable. This suggested that we should explore a

decentralized collaborative approach, in which a long time-line and contributed labor might overcome the lack of funding.

2.2 DATABASE WEB SITES

For sites providing access to dynamic content, an alternative approach has emerged, in which the contents of a traditional database are presented to the web through software which translates between the database and HTML. Such an approach is usually tied to specific technologies (database and software), requires significant centralized server resources, and is not very portable.

We felt that a template based solely on this technology would not be viable due to limited support. At the same time, the integrating a database into the overall site strategy offered possibilities of interaction and interconnection which might eliminate some of the the problems of multiple servers.

2.3 SEARCH SITES

The third common type of web site or web resource also involves database software. Over time a software robot, or spider, examines a site, distilling the content of each page, extracting the URLs of linked pages, and storing the results in a database. The user requests a search of the database, looking for certain words or phrases, and the database software returns the URLs of all matching pages. The user is then able to access each of the pages, in turn.

The difficulty is that many sensible searches produce thousands of irrelevant "hits" (Johnson and Millet, 1997) and browsing constantly involves backing up to the search engine in order to establish what document to examine next. Nonetheless, we felt that it was important to provide some free-form searching capability.

3. Issues Related to Decentralized Approaches

3.1 ADVANTAGES OF DECENTRALIZATION

Decentralization has several advantages, not the least of which is that no one institution "owns" the collection. The case studies are served by web servers located at numerous institutions. Local hosting recognizes that some case studies might well be refined over a period of time as interest and resources permit. In addition, this low-tech approach may enhance the sense of pride attendant on authoring or refining a case study precisely because the material is created, housed, and served from a local server.

Local hosting also minimizes problems with copyright liability. Case studies of buildings in the neighborhood of the student's institution would allow the students to do original research (interviewing designers, clients, occupants,

etc.), but would also give them the opportunity to take their own pictures. Local files will help keep faculty oversight at an appropriately high level.

3.2 POSSIBLE RELATIONS BETWEEN FREESTANDING SITES

"Hot Links"

The simplest means of providing "cross links" to pages outside your own site is to insert a *full* or *absolute* URL for the target page as a link on one of your pages. Such lists are fragile, however, since the author never knows for sure if the target document is still available or in the same location.

Static Pages & Search Engines

The combination of static web pages and database search engines has proven quite powerful, though the user is often overwhelmed by the number of candidate documents and the process of browsing the results does not facilitate continuity of thought.

Web Rings

One response to the "info glut" provided by search engines has been the emergence of web rings. In these collaborative schemes a central site maintains a list of participating home pages. Each of these home pages includes a set of "forward" and "back" links which access the centralized list in order to forward the browser to the next/previous site in the list. Repeated use of the "next site" link will eventually return the browser to the point where they started (hence the term "ring").

4. The Cross-Link Problem

The template enables creation of traditional stand-alone web sites or web-site sections. However, we needed users to be able to hit a "Find Similar" buitton and jump from case study to case study following a topic or subject line, rather

than following the pre-built links within a case study. We call this "cross-linking" or "cross-grain browsing". Since the URLs must specify the exact name of the server and the file path for any files they reference, the URLs for these links are not knowable ahead of time, and will change over time as new projects are added to the collection. They can't be built into the template.

At first it may seem that this problem attaches only to the distributed environment, but inasmuch as we expect the pool of case studies to grow over time, with each case study created as a free-standing body of work, the challenge would apply equally to a series of case studies being served from a

single server. We did not want to have this problem.

5. The Archimedia Multi-site Components

The approach we adopted for development of the Archimedia project, (at http://www.caup.washington.edu/archimedia/) involves three components: The case study template; an indexing robot and search engine which provide a means of linking between case studies; and an on-line catalog of the case studies which can be searched or sorted various ways.

5.1. THE CASE STUDY TEMPLATE

The template (c.f. Johnson and Millet, 1997) fosters visual coherence in the collection and provides a well-structured starting point for new case studies. We wanted a robust, easy to edit, internally linked case study template that could be downloaded to any server and with which various kinds of case studies can be created.

The Structure of the Template

The logical structure of the template, shown in Figure 1, was derived from the earlier mult-media project, with some modifications. The decision was made to use "generic" HTML tags rather than some of the more limited-circulation tags specific to certain browsers or scripting languages. This meant that the original interface had to be simplified somewhat, but the major elements were preserved.

The template consists of a set of folders, each containing an HTML page addressing a category of information and a folder of related picture files. Pictures are provided in two forms, a thumbnail GIF file and a larger size JPEG file. The images folder also contains a series of HTML files, each of which provides the associated JPEG image with context (title, photo credits, link back to the main page, etc.) The topical HTML file contains references and links to the pictures, as well as place holder text to facilitate editing.
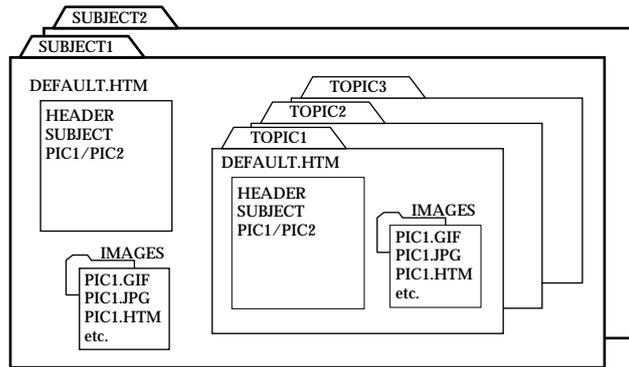
Figure 1. *Case Study Template Structure.*

Topics are collected together into larger subject areas (and directories), which also include an introductory HTML document and associated image files. Finally, subjects are collected into one directory, called the source directory. Additional files exist at this level to provide navigation control, but there should be little or no need to edit these files.

Navigation of the Template: The use of HTML Frames

It would be straight forward to place links in all the HTML files by which users would navigate the site. However, these links would be distributed throughout the various HTML documents, which makes updating them awkward in case of a change in the fundamental structure of the site, and which exposes them to accidental alteration during the process of content creation.

Partly in order to avoid both of these problems, and also in order to enable "crosswise browsing" (discussed later), we decided to implement the template using the HTML *frame* element. Frames permit one HTML document to control the display of another one in another portion of the browser's window (Netscape, 1977). In our case we created a set of HTML documents which display navigation links along the left side and bottom edge of the browser window, and which permit any of the case study topic pages to be brought up in the main window.

The result of this approach is an easy to navigate and edit, but difficult to corrupt template project. Since the HTML files implementing the frames are part of the template there is no requirement to preserve a link to the "home site" during case study development, or even to develop in an on-line setting.

5.2 THE CROSS-LINK REDIRECTOR

A transparent solution to the cross-link problem is illustrated in Figure 2. It

utilizes a database and a web server response called an "http redirect."

When a server is asked to provide a particular document it may respond by sending the document, or it may respond to the effect that, "The file you requested may be found at *<some URL>*". The browser then requests the document using the new URL. This URL may indicate a file on the same server, or it may indicate a file on another server. This entire interaction is extremely quick and usually invisible to the user.
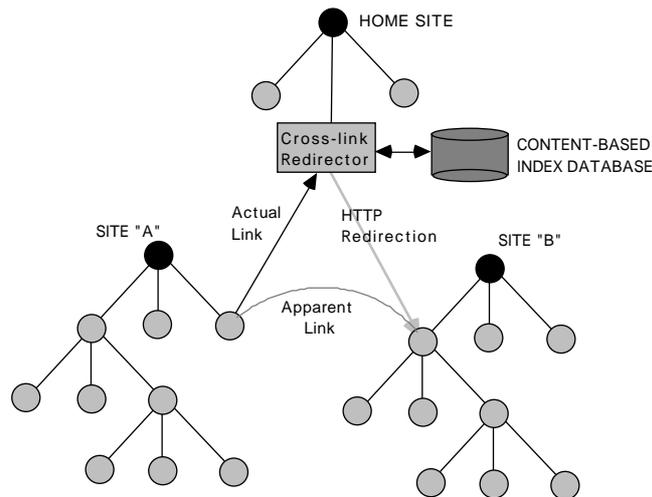
Figure 2. *The ideal crosswise browse.*

The redirection response can be generated by a program running on the server, and taking into account information from the web browser, such as the URL of the page from which the reference comes. The URL of the program can be built into the template files, because it does not change. The program's response will depend on the current state of the Archimedia catalog and search engine index data.

A fully implemented multi-site would utilize a "redirection server". Given the referring URL, and based on rules for determining the next most appropriate site, it would provide a "one click" link to similar pages in other case studies, without intervention.

The only way this can be done is if the redirection server has a database containing the URLs of those web pages which cover the same topics. The only off-the-shelf software which is available is site indexing agents. When queried, they provide a set of URLs from which to choose, as illustrated in Figure 3.

In the current implementation this means that during a cross-link there is a pause at a page where the user selects the exact target of the link. To narrow the searches conducted by the search engine a pre-processor routine uses the referring URL to establish the subject for the search and then redirects the query

to the search engine proper. To guarantee that pages contain a unique piece of text for the robot to record, each page of the template contains a string that uniquely defines it within the case study context, in terms of subject and topic. This keeps the (human) browser from being inundated by pages that do not really match the initial page.
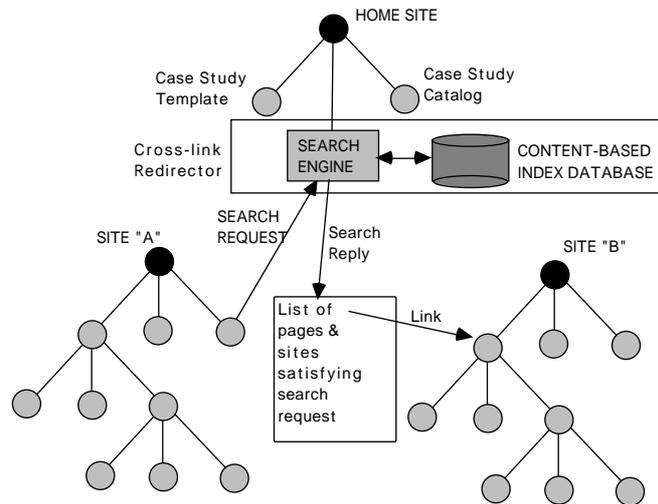
Figure 3. *The current cross-wise browse*

5.3 THE CATALOG

There are many excellent architectural case studies being served on the web already. These studies were obviously not done using our template, and they do not include the keywords that enable our "Find Similar" operation to function. They should, nonetheless, be included in the collection.

We created an automated catalog function which uses HTML forms to acquire the information needed to add a new case study to the catalog. This information is also posted to the robotic web indexing engine, so that the pages will be indexed for subject searches. All case studies can be searched, but case studies done with our template, or into which the appropriate key words have been inserted, are simply easier to index and search.

6. Results

We have developed and deployed a template using HTML frames. We have tested the template by converting the earlier Alvar Aalto project to the new format as well as developing a new project for Arthur Ericksson's Museum of Anthropology, in Vancouver, B.C., Canada.

The template should be easy to use in a classroom setting ("Team A gets this

folder, Team B will work on this one", etc.). All image and navigation links are in place. The files also include identifying strings that make the material easier to search. To facilitate customization, each subject area includes a separate background texture, and page headers are designed to present the case study title and institution name prominently so that authorship is clear.

Once a case study is completed, it is submitted to the catalog through a simple on-line form. It becomes immediately available through the catalog, and is indexed within a few days, so that it shows up during a "Find Similar" and other content-based searches.

The catalog database is maintained in FileMaker. The indexer/searcher function is run using Maxum Corporation's Phantom. The form processors and cross-link redirector CGIs are written in AppleScript.

Existing case studies may be submitted to the catalog without alteration, or the appropriate topic "markers" may be inserted into the text of the case study to facilitate indexing and searching.

6.1 REMAINING PROBLEMS

The "Find Similar" Rules

There is a problem, alluded to earlier, with the implementation of cross-grain browsing. It is necessary to of identify suitable rule(s) for the "Find Similar" operation. While browsing within a case study is structured around a particular building, cross-grain browsing might conceivably be guided by the architect, the region, the building type, the construction material, the climate, and so on. Some interface needs to be assembled to readily facilitate browsing along different threads of information.

In addition, the stateless nature of web browsing, in which each document request occurs by itself, without information about previously visited pages, makes it quite difficult for the server to identify which pages the user may have already visited, so as to avoid sending them there again.

Interface

It is difficult to design a frames-based interface which is as graphically interesting as other page design strategies. In general, we felt that the results warranted the compromise. However, if the interface isn't visually appealing, we may well have problems enticing users. For this reason we are looking at alternative window layouts, including frameless designs.

7. Conclusions

We began with the desire to increase the availability and quality of case studies for architects at a very low cost. Our investigation suggested the need to address four problems: funding and maintenance of a suitable server; development of a case study prototype; resolution of the cross-link maintenance problem; and incorporation of existing case studies into the collection. While imperfect, we have addressed all of these in the ArchiMedia system, which combines distributed server capacity with intelligent linking to create a "multi-site".

We believe that the availability of an on-line resource will contribute to retaining and enhancing the best case studies and that the use of local servers actually strengthens the sense of ownership and responsibility that accompanies authorship.

Finally, we believe that topic oriented multi-sites will eventually fill the gap between personal "hot links" and the common "30,000 pages found" response to many queries on the major indexing services. They may not take quite the form we suggest here, but it is a powerful infrastructure for distributed, cooperative web-based projects.

References

Johnson, B., and Millet, M. "A Web-based Architectural Case Study System", *Architecture: Material and Imagined: Proceedings of the 85th ACSA Annual Meeting and Technology Conference*, Dallas, Texas, pp 491-496.

Millet, M., Cohan P., and Hildebrand G., "ArchiMedia; Interactive Architecture." *Educational Multimedia and Hypermedia Annual 1993*, (Association for the Advancement of Computing in Education, Charlottesville, VA, p. 616.

Millet, M., Cohan P., Hildebrand G., and Read M., "ArchiMedia Case Studies: Integrative Architectural Education," *Reality and Virtual Reality: Proceedings of ACADIA 91*, Los Angeles, California (New Jersey Institute of Technology, Newark, NJ, 1991), pp. 127-134.

Netscape Communications Corp, Frame Syntax,
<http://home.netscape.com/assist/ net_sites/frame_syntax.html>
(January, 1997)