

Creating Educational Guided Paths over the World-Wide Web

Frank M. Shipman III, Catherine C. Marshall, Richard Furuta,
Donald A. Brenner, Hao-wei Hsieh, and Vijay Kumar

Center for the Study of Digital Libraries &
Department of Computer Science
Texas A&M University
College Station, TX 77843-3112, USA

ABSTRACT

The World-Wide Web has the potential to support education through access to extensive and encyclopedic materials. The materials must be tailored and contextualized to promote the instructional goals. To provide this contextualization, we have applied the concept of a *guided path*, an ordered list of pages independent of the existing Web structure. Besides creating the structure, a teacher may annotate the individual pages of a path to provide transition, emphasis, and missing explanation. In addition, some limited interactivity and control over the display of remote information is possible in our prototype path server. This prototype, called Walden's Paths, works with existing Web browsers and servers allowing its integration into an educational setting with existing hardware and software.

INTRODUCTION

Students develop knowledge-building skills and strategies through their exposure to an expanded discourse community and broad base of information resources [SB93]. The Internet shows promise of providing students with such exposure: a wealth of new material and a spectrum of new voices are becoming available to students and educators alike through networked electronic information resources like the Internet's World-Wide Web. The breadth of this material promises to increase as digital library efforts continue and research organizations recognize the importance of contributing to globally accessible multimedia databases.

As extensive and encyclopedic as these materials are and promise to become, they still must be tailored for use in a classroom setting. The bulk of the material available today is not aimed at the needs of K-12 students, although many elements—collections of images, simulations, segments of digital video, audio, electronic versions of well-known works of fiction and reference materials, library indices, databases, and hypertextual documents—have the potential to play a strong supporting role in the curriculum of tomorrow.

Access to extensive resources and a broader discourse community will be instrumental in supporting learning through exploration (a natural complement to what Pea and Gomez refer to as learning-in-doing [PG92, Pea93]). Exploration is a valuable mode of learning, but it is even more valuable when it is constrained by a curriculum developer's well-conceived ideas of which materials should be included, supplemental

text aimed at the particular level of student, and additional structure and ordering to help the student comprehend what he or she is discovering.

Scardamalia and Bereiter distinguish between knowledge reproduction strategies and knowledge building strategies: knowledge building strategies focus on the development of understanding, while knowledge reproduction strategies focus, very literally, on students' abilities to absorb passively, then recreate, what they have been told [SB93]. Our focus is on using guided exploration of large scale information resources to engage students in comprehending, interpreting, and evaluating materials—the substance of knowledge building and critical thinking.

The World-Wide Web (which we will refer to as the "Web") and its hypertextual paradigm are well-suited to form a basis for exploratory learning. One central theme of hypertext and the Web is traversal: a reader moves from one segment of material (a node or page) to another by following a link to related material. A reader's need for detail, explanation, alternative discussion, or related topics is guided by the reader's own desire to explore, to construct knowledge, to find information. Of course, without a particular aim in mind, or any sort of guiding purpose or instruction, link following can easily become a random walk. It is necessary to add meta-structure (in this case, meta-structure that reflects an instructor's curricular goals) to the underlying hypertextual network to make it suitable for exploratory learning and knowledge construction.

We can envision the future to some extent by looking at materials, media, and genres available on the Internet today. If we look, for example, at the NASA Spacelink Web site, we can find information for the public about NASA programs (including existing educational materials).¹ Or we can find movies of insects on Iowa State's entomology information server. Or we can view the Library of Congress's Soviet Archives Exhibit. Some of the information provides methods for interaction; for example, Xerox PARC's map viewer allows readers to zoom on map regions or search for place names as one would in a gazetteer. Simulations and visualizations are also available through the Web to help readers grasp more difficult materials and concepts. Authors have

¹Figure 1 contains the locations (the URLs) of the examples cited in this paragraph.

- NASA Spacelink: <http://spacelink.msfc.nasa.gov/>
- Iowa State's entomology information server: <http://www.public.iastate.edu/~entomology/ImageGallery.html>
- Library of Congress' Soviet Archives Exhibit: <http://www.ncsa.uiuc.edu/SDG/Experimental/soviet.exhibit/entrance.html>
- Xerox PARC's map viewer: <http://pubweb.parc.xerox.com/map>

Figure 1: Some Web examples

created a number of indices to effect additional structure on top of these diverse sites, but most of them are just lists of Web pages or Web server sites or hierarchies of such lists; few of them provide the additional rhetorical structure that one would encounter in materials for classroom use.

PROBLEMS OF GENERAL WEB ACCESS

What are the specific kinds of problems that we anticipate (and have observed) when students are given general access to the kind of large, heterogeneous collections of information that we find on the Internet? Basic needs for adapting and tailoring the Web for exploratory learning include mechanisms: for defining a territory or information space; for adding structure to the information space to promote comprehension and accessibility of the material; for tailoring existing content and links to meet curricular constraints; for tracking student progress and adapting paths to individual differences; for maintaining the quality and integrity of the derived instructional materials; and finally for sharing these metastructures and instructional strategies within a community of educators.

A significant amount of material is not organized for comprehension by a K–12 student. Much of the information on the Web assumes access by an information-seeking adult or possibly an adult who is casually browsing or “reading around.” This material, if left as is, will bore or frustrate most students, since they require a more structured presentation of background material on the way to exploring less organized information.

Given a relevant territory (which we will refer to as an information space) and a general organization for material, a problem still remains: the material—the content and links—still needs to be tailored to address the needs of school-age learners. Because the Web's hypertextual structure is represented by content mark-up [BL94] (i.e., links are denoted within the pages themselves), this sort of tailoring requires methods for changing material at a within-page (intra-node) level. For example, a given Web page may include too many links (and possibly links to material outside the information space). Or a Web page may need additional rhetorical structure to guide the student; because many Web document genres (such as home pages) are new, many authors who contribute valuable material are inexperienced in constructing readable hypertexts. Within-page tailoring may also be necessary to adapt material that is presented at the wrong level for a K–12 student. A second grader who is interested in the space program will not be able to understand a mathematical description of vehicle trajectory, but may be able to understand diagrams or a simple verbal account of the same material.

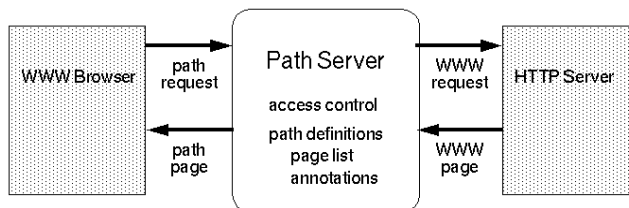


Figure 2: Diagram showing communication and information storage of path server.

APPROACH

The initial metaphor in implementing an environment to support focused network exploration is a generalization of the *guided path* [Zel87, Zel88, Tri88, Zel89]. As originally defined, the guided path provided the means for directing a reader's traversal along a path of components extracted from a set of documents. The ordering of components on the path is not constrained by that of the source documents—in other words, the components encountered do not have to follow the temporal orderings of the source. In essence, the guided path allows creation of a *presentation*, defining a meta-structure that is layered on top of the underlying documents' preexisting structures.

The guided path is well-suited for control of presentations and for communication of relationships. It serves as a meta-structuring mechanism that can be used to express an order over a large collection of information.

Besides providing an ordering of pages, a guided path can provide additional context for the page through annotation. By providing text or other annotations in addition to the content of the page, the path author may provide a rhetorical structure to the path as a whole, create transitions to fill in informational gaps between pages, and emphasize particular aspects of the materials.

Prototype Path Server

To take advantage of the variety of information and software that makes up the Web, a guided path mechanism must work with the common Web browsers and Web servers. We adopted a strategy to create a “path server” to operate in between a student's browser and the information server. The path server stores local information specific to the paths—namely the order of pages in the path and their annotations—and how to retrieve the original information as shown in Figure 2.

By leaving the content of the pages at the Web server and retrieving it when a Web browser requests a path page, the path server can take advantage of the evolution of the pre-

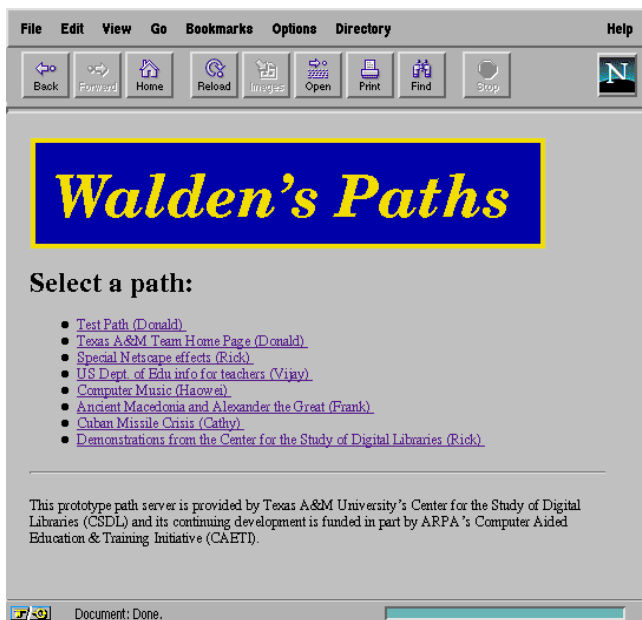


Figure 3: The opening page listing available paths.

existing Web structure and content. Once retrieved, the path server adds path controls and the path-specific annotations and returns the resulting page to the Web browser.

Our prototype path server, called Walden's Paths, operates as a CGI executable script. CGI stands for Common Gateway Interface, which refers to standards that allow communication between HTTP servers and the server-side gateway programs. The path server program is written in C.

Upon execution, the path server distinguishes between three basic situations: when a student is accessing the path server but not a specific path, when a student is accessing a particular page in a path, and when a student is accessing a page not on the path. The path server is able to identify the current situation as one of the above by examining an environment variable called the URL query string.

If the student is not currently viewing a path, the path server lists the possible paths that the student can choose from. This is shown in Figure 3. The list is built using an index of the available paths, which is stored with the path server.

If the student already is viewing a path page, the path server determines the appropriate URL to retrieve using the path information stored at the server. Paths are represented by a series of URLs and associated annotations. These paths, like the index of the available paths, are stored as files accessible to the server. The current prototype limits paths to be linear; no "side trips" or "alternate routes" can be specified by the path author.

After determining which URL is requested, the path server performs two tasks. The first task is to retrieve the Web page of the URL. Once the page has been retrieved, the path server adds path controls and annotation to that page. There are three elements of the path controls: forward and back arrows to permit traversal of the path and a Walden's Path

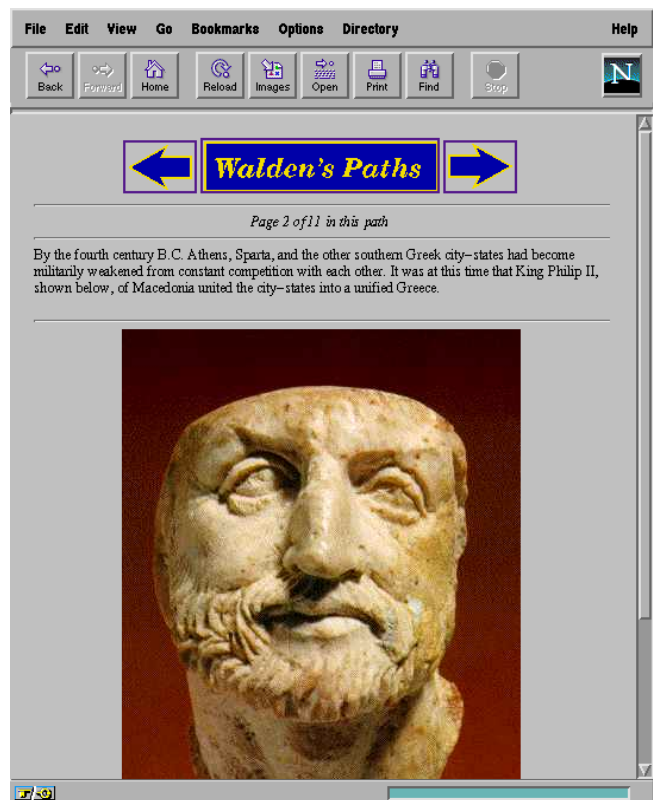


Figure 4: Page of path shows control buttons at top, followed by positional and annotation information, and the original material from the Web.

logo that, when selected, returns the student to Figure 3's display. Annotations, added to the page just below the path controls, can include plain text and HTML markup, including anchors, forms, and image sources. The path server then sends the resulting path page to the student's Web browser, such as Netscape or Mosaic, which renders the modified HTML markup. Figure 4 shows a page in a path on ancient Macedonia and Alexander the Great.

If the student chooses to follow a link that is not part of the authored path, the path server retrieves the requested Web page and prepends a Walden's Paths Logo that, when selected, provides a link back to the last path page seen by the student. Figure 5 shows a page off of one of the authored paths.

If the student traverses off the path, the server must make sure all the student's accesses go through the path server so the "back to path" controls can be added. To do this, the path server replaces all the URLs specified in HTML anchors to route requests through itself. One problem in implementing our prototype has been that many URLs are context-dependent, or "minimized". Minimized URLs must be expanded by the path server for this to work correctly.

Speeding up Path Access: Caching at the Path Server

There are possible limitations of network bandwidth and accessibility that need to be addressed for the use of the path server in educational settings. First, not all schools have network connections, much less fast network connections. By

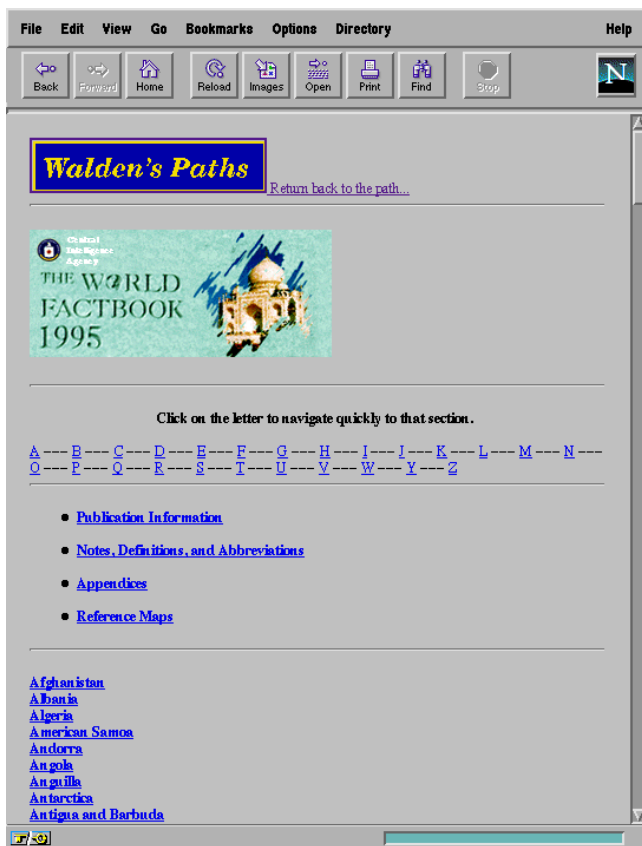


Figure 5: Web pages not on authored path have buttons added to get back to the path.

caching Web information when it is accessed, the path server can reduce the required network traffic between the school and the Internet. Additionally, as sites on the Internet are not always accessible, pre-caching pages that are part of a path allows a teacher to know that the information on the main path will be available in the classroom.

To support these goals of reducing the reliance on network bandwidth and guaranteeing the availability of information, we have been investigating two methods of caching documents at the path server. The first method, *regular-caching*, is like the caching done by Netscape and some other browsers. The path-server simply caches documents as they are retrieved from the Internet. This approach saves what the students are browsing, including Web pages not on the main path. The advantage of having the path server do this caching over having the Web browser cache information is that the cache is shared by all students of the path server. This means each page is only retrieved once (and stored once) for everyone using the path server.

The second method of caching, *pre-caching*, means the path-server caches what it may need later. Normally a document is not retrieved (and thus not cached) until a student requests it. Here, we can take advantage of the fact that we know beforehand the URLs that are going to be accessed during a session with the path server. In the case of guided paths, it is natural to assume that the pages on the path are likely to be

viewed later, especially if a student has begun traversing that path. Our current implementation is limited to accessing and caching the Web pages listed in the paths.

Caching problems. As with Web browser caches, using a cache at the path server leads to a number of issues concerning versioning, cache size, and cache time-out policies. First, local versions of a Web page may be out-of-date relative to the page available via the Internet. One method to address this problem partially is to have cached information “time out” so that after some length of time the information in the cache is discarded. A related issue is how much disk space should be used by the cache and what replacement policy is to be used when necessary.

These general issues concerning the caching of Web information can be better addressed by considering the particular application. We will first consider the issue of out-of-date information and time-out policies, then look at appropriate replacement policies for the path server application when cache space becomes an issue.

First, in the case of guided paths, static information may sometimes be an asset. A teacher may want to be sure that the material on this page will not change before the students access it. On the other hand, if the page is a real-time display, such as the current local weather map, having an old version may not be acceptable. This implies that each page should have its own policy regarding caching. There are at least four possibilities: (1) cache now and don’t remove (sets the exact information for the path); (2) cache during execution and don’t remove (ensures all students see the same up-to-date version); (3) cache during execution and remove as needed (implements the traditional caching scheme); and (4) don’t ever cache (retrieves dynamic information each time).

The problem of determining what to remove from the cache when the cache is full also is influenced by the particular application. A common strategy for this problem is the Least Recently Used (LRU) policy where the page least recently accessed is removed from the cache to make room for new pages. A problem for the path server cache is the variation in size of elements of the cache. Making space to cache a large color image could force a disproportionate number of documents out of the cache. Cache management is an area of active investigation in our project.

AUTHORING PATHS: EXPERIENCES AND ISSUES

Figure 3 showed the path server’s introductory page, listing a number of our experimental paths. Authoring these paths has augmented the lessons reported in our previous accounts of path construction [MI89], especially since much previous work on paths has relied on locally-controlled document collections. By contrast, we have used materials gathered from the Web as a basis for our paths. In our examples, existing Web pages from different servers around the world are structured as simple linear paths, and supplemented and explained by the authors’ narrative, implemented as a prepended annotation. Example paths include an artifact-centered description of Ancient Macedonia and Alexander the Great, a short account of the people and events of the Cuban Missile Crisis, a narrative explaining our digital library work with

local herbaria and botanists, and a tour through computer music resources. Several other paths also were written to test various features of the path server.

Our experiences highlight a set of issues and questions that will guide future development. We focus on four: (1) the interaction between available content and curricular goals; (2) author control of Web page markup to promote consistent styles and coherence within a particular path; (3) the potential for interactivity in our prototype; and (4) supporting and accelerating the path authoring process.

How does existing content interact with instructional goals? The Web is a rich source of content, but is it the content that an instructor needs to put together a lesson on the topic of choice? Will the availability of materials tend to drive the kinds of paths teachers create? Several of the example paths use sets of materials culled from different sites around the Web. It comes as no surprise that it is easier to construct an interesting path from an already coherent source (like a library of related images drawn from one or two Web sites) than it is to gather materials from many sites and integrate them. Text, as one would expect, is very difficult to reuse; it is frequently directed at a general Web audience, at a special interest group, or at an academic audience. Textual indices are slightly easier to reuse, but still must be adapted for instructional purposes (i.e., not all the links that they provide are appropriate). Images, sounds, and digital video clips are much easier to adapt for different uses. For example, Figure 6 shows a page from the Cuban Missile Crisis tour. Note that while the image fits the path author's intent, the Web page the author has selected has inappropriate text (which has been rendered in a very small font to de-emphasize it). In general, this characteristic of Web materials is leading us to look at ways of reusing document *components* (such as embedded images) rather than entire document pages.

What cues might a path author use to indicate path coherence? Most of these cues involve the introduction of additional markup in the path. For example, in our Cuban Missile Crisis path, the author has used a new font size to set off her annotations from the text of existing Web pages. It is easy to see how such markup—for example, markup designating a uniform background color, or using consistent image centering strategy—can promote a sense of coherence for the student. Many of our example paths introduce new markup elements to give the outside page and the author's annotations a particular "look". Our mechanism for indicating whether a reader is on or off the authored path (see Figure 5 in the previous section) is yet another means by which coherence is supported.

Promoting interactivity is an important goal of our project. When paths are simple, it is all too easy for a student to fall into passivity—just clicking his or her way from beginning to end, not exploring or constructing any new meaning. There are many ways around this apparent pitfall. The first, and most decidedly technological solution, is to provide authors with a more expressive path mechanism, one that promotes more active exploration. Our past work [MI89] has found that paths generally have "spines" and "side trips"; side trips are interesting digressions that allow a reader to pursue ma-

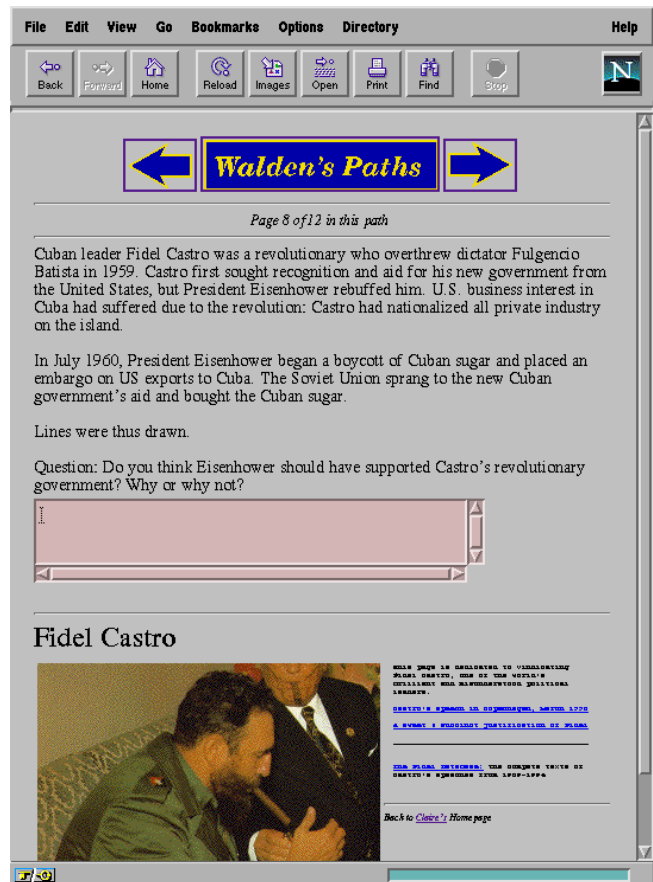


Figure 6: Page from path on Cuban Missile Crisis that includes an HTML form and alters the display of original material to provide emphasis.

terial in more depth. Other of our past work [SF89, FS94] provides a general mechanism for expressing paths through documents and their components. But we have seen exploration facilitated by simply adding links into the annotative text, and inviting the student to jump off the beaten path for awhile. The previous section discusses how the path server provides "off the path" feedback, and a mechanism to return. We are also considering ways to limit off-path exploration, so that an interesting digression doesn't become a distraction, or a foray into material well outside the classroom curriculum.

Of course, interactivity means more than additional places to click. Figure 6 shows how HTML forms markup has been used to invite student input; this markup, or "mailto:" links will enable a teacher to gather student reflections or responses to questions posed in the annotations. Other mechanisms have been developed for more traditional CAI-types of instruction, like system-evaluated multiple choice questions. We can also envision students authoring paths themselves (either for their peers, for their teacher, or for younger students) as a more interactive form of engagement with the material.

How can we support the path authoring process? First, instructional design templates can make it easier to create paths [JRJR89, CG91]. At the very least, most of our examples have an introductory page that sets the tone and style for the

rest of the path. In an ideal world, instructional designers and experienced path authors can create general types of paths to help newer path authors—teachers and students—satisfy their own instructional goals. Second, path cues—like the progress along the path that is shown at the top of the constructed page in Figure 6, “Page 8 of 12 in this path”—help the author by making certain common kinds of reader feedback available across all paths. Finally, we can support the reuse of *portions* of paths, including individual annotated pages. A discussion of John F. Kennedy from the Cuban Missile Crisis path might fit well into a unit on American Presidents.

Our initial prototype, and our experiences authoring paths in it, provides a promising artifact that can serve as the basis for interaction with classroom teachers and students. Our next step is to involve teachers and students in the project; from this ongoing feedback, we will continue to refine and develop the path authoring and path server capabilities.

CONCLUSIONS

In this paper we have discussed the use of guided paths to overcome some of the difficulties associated with using information available via the network in an educational setting. We have described our path server, Walden’s Paths, which allows the contextualization, annotation, and ordering of Web pages to provide paths on particular topics. The paths provide for a rhetorical structure independent of the materials existing structure.

The path server is independent from specific Web browsers and servers, enabling its use with the expanding variety of such software. This architecture also allows the information to remain at its original Web location or to be cached locally to improve access time and guarantee availability. There are a number of trade-offs in the caching of networked information that we have begun to investigate in the context of curricular use.

Through building a set of sample paths we have begun to gain experience with the issues of path authoring. Our initial prototype enables the path author to provide some limited interactivity and control over the display of remote information. This will be extended as we build tools to support the authoring of paths.

REFERENCES

BL94. Tim Berners-Lee. HyperText Markup Language (HTML): Working and background materials, 1994. URL <http://www11.w3.org/hypertext/WWW/MarkUp/MarkUp.html>.

CG91. Karen Smith Catlin and L. Nancy Garrett. Hypermedia templates: An author’s tool. In *Proceedings of Hypertext '91*, pages 147–160, 1991. San Antonio, TX, December.

FS94. Richard Furuta and P. David Stotts. Interpreted collaboration protocols and their use in groupware prototyping. In *Proceedings, ACM 1994 Conference on Computer Supported Cooperative Work*, pages 121–132. Association for Computing Machinery, October 1994.

JRJR89. Daniel S. Jordan, Daniel M. Russell, Anne-Marie S. Jensen, and Russell A. Rogers. Facilitating the development of representations with IDE. In *Hypertext '89 Proceedings*, pages 93–104. ACM, New York, November 1989.

MI89. C. Marshall and P. Irish. Guided tours and on-line presentations: How authors make existing hypertext intelligible for readers. In *Proceedings of Hypertext '89*, pages 15–26, 1989. November 5-8, 1989, Pittsburgh, Pennsylvania.

Pea93. Roy D. Pea. The collaborative visualization project. *Communications of the ACM*, 36(5):61–63, May 1993.

PG92. R. D. Pea and L. Gomez. Distributed multimedia learning environments: Why and how. *Interactive Learning Environments*, 2(2):73–109, 1992.

SB93. Marlene Scardamalia and Carl Bereiter. Technologies for knowledge-building discourse. *Communications of the ACM*, 36(5):37–41, May 1993.

SF89. P. David Stotts and Richard Furuta. Petri-net-based hypertext: Document structure with browsing semantics. *ACM Transactions on Information Systems*, 7(1):3–29, January 1989.

Tri88. Randall H. Trigg. Guided tours and tabletops: Tools for communicating in a hypertext environment. *ACM Transactions on Office Information Systems*, 6(4):398–414, October 1988.

Zel87. Polle T. Zellweger. Directed paths through collections of multi-media documents. In *Hypertext '87*, November 1987. Position paper.

Zel88. Polle T. Zellweger. Active paths through multimedia documents. In J. C. van Vliet, editor, *Document Manipulation and Typography*, pages 19–34. Cambridge University Press, April 1988. Proceedings of the International Conference on Electronic Publishing, Document Manipulation, and Typography, Nice (France), April 20–22, 1988.

Zel89. Polle T. Zellweger. Scripted documents: A hypertext path mechanism. In *Hypertext '89 Proceedings*, pages 1–26. ACM, New York, November 1989.