VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

School of Mathematical and Computing Sciences
Computer Science

PO Box 600
Wellington
New Zealand

Tel: +64 4 471 5341, Fax: +64 4 495 5045
Email: Tech.Reports@mcs.vuw.ac.nz
Http://www.mcs.vuw.ac.nz/comp/Publications

Tool Support for Scholarly Web Sites

P. Martin

Technical Report CS-TR-CS-TR-98/8
August 4, 1998

Abstract

The researcher's notebook is a familiar tool in many disciplines and has several important uses: as a place to record detailed facts, as a place to store reminders of sources of useful information, and as a place to develop new ideas.

We have previously proposed a way of implementing an electronic version of the research notebook using a novel combination of hypertext and directories which we call the 'Scholarly Web Site' (SWS). The SWS is designed to be particularly supportive of the tasks of collecting information (or pointers to information) of diverse kinds, of finding particular items of information in a large collection, of discovering new information and relationships from a collection, and of refining the structure of an information collection.

In this paper we describe a set of tools to support the creation, the maintenance, and the integration of the SWS with other evolving information bases and we report the successes and challenges that resulted from our development of prototype versions of these tools.

Keywords Hypertext, electronic notebooks, research methodology, WWW.

1 Introduction

The researcher's diary or notebook is a familiar tool in many disciplines [3]. In this paper-based friend, the researcher accumulates material of numerous and diverse kinds including: data collected in the field, summaries and critiques of the literature, bibliographic details for source material, lists of relevant researchers, institutions and conferences. Notebooks often also accumulate more analytical material including: ideas for investigation, strategic plans for ongoing research, and outlines for possible papers. Finally, some notebooks are used by researchers to reflect on the progress made and external or self-imposed obstacles to progress.

In fact, not only do notebooks contain a diversity of information but they are used in several distinct ways. First, notebooks are used to look up specific facts, e.g. the page numbers of an article to be cited, and to find (or re-find) indirectly other material, e.g. by looking up the library's call number for an important text. Second, notebooks are used as a repository of information out of which new relationships may be discovered. Third, notebooks — especially those in the form of loose-leaf folders or index cards — are used as a practice ground for marshaling and re-marshaling ideas until a logical disposition is found.

The computer can extend the traditional paper-based notebook in at least three ways. At a trivial level, the word processor replaces the pen and paper (loose leaves or index cards) and facilitates the production of a neat and tidy record. More significantly, the computer can provide a fast and effective search tool for locating information in the online notebook. The computer can also provide different views of the material in the electronic notebook, and greatly speed the re-marshaling of material.

In [5] we introduced a general method for implementing online, electronic versions of the research notebook which we call 'Scholarly Web Sites' (SWSs). The main purpose of the SWS is to support the looking up of specific facts, the discovery of new information ([2] terms this 'information farming'), and the reorganising of existing information.

One of the nice properties of the SWS is that it can be implemented using existing Web technology. However, our experience using a SWS has shown that there are several ways in which the ease of use and effectiveness of the SWS can be improved by means of some new tools. In this paper we discuss what tools are useful and report our experience with some prototypes tools.

In the next section we review the main elements of the SWS. In Section 3 we describe three sets of tools that could be usefully implemented to support SWSs. In Section 4 we report our experience developing the prototype tools. Finally, Section 5 concludes the paper.

2 Scholarly Web Sites

In this section we review the main elements of the SWS as first proposed in [5].

SWSs are structured using the hypertext approach [4] with extensively interlinked 'directory', 'content', 'external' and 'interface' nodes. We choose to use hypertext for several reasons. First, we believe that browsing (a suitably presented) hypertext can be a more effective way to re-find information in a research notebook than using information retrieval methods [6] which require the user to specify their information need with great precision. Second, the hypertext links provide an efficient way for researchers to record new relationships that they discover in the notebook. Third, the division of material into nodes provides a convenient and natural basis for organising and reorganising material into various structures such as lists and hierarchies.

The technologies that we chose to implement the SWS are the standard Web technologies, so SWS nodes can be authored using conventional HTML editors and delivered to readers using standard Web servers. This choice, we felt, would also reduce the learning curve for those researchers who wanted to build their own SWS. Further, this choice means that a SWS can include any of the diverse kinds of information that are supported by the current range of Web browsers, plug-ins, helper applications, and other, 'Web-aware' office productivity tools.

In addition to the directory, content and external nodes mentioned above, SWSs also have directory, content and external links which by definition are links that point into directory, content and external nodes respectively. We will now summarise the function and use of nodes and links in the SWS.

2.1 Directory nodes

The directory nodes are the main navigational device in a SWS. They function in a similar way to the 'landmarks' described by [1] in that the directory nodes are typically few in number, they are extensively interlinked with the rest of the SWS, and they are sufficiently uniform (as described next) that the user can learn the purpose of a directory node from seeing only a small part of it. Having these directories as landmarks in an important technique for reducing the 'lost in hyperspace' problem described by [7].

Internally, a directory node is an ordered list of objects of the same type. For example, a SWS may have a directory of 'conference' type objects where each object represents knowledge acquired by the researcher about a particular conference. The type of an object determines the attributes of the object — a conference object might have 'title', 'acronym' and 'location' attributes, for example.

It is intended that the directory entry for a particular object will represent (directly or via links) *all* information known to the researcher about that object. Further, each object will have directory or content links to every node in the SWS where it is mentioned, and each mention will have a (directory) link back to the object. It is this richness of linking that makes directories useful 'jumping off points' and that makes browsing an effective way finding information.

Adaptive Hypertext and Hypermedia, Workshop on...

- AH '98, second, Pittsburgh, PA, June, in conjunction with Hypertext '98, {WWW: Home page}
- AH '94, first, Cape Cod, Massachusetts, August, in conjunction with {CONFERENCE: UM '94}

Education Multimedia and Hypermedia, World Conf. on...

- Ed-Media '99, Seattle, Washington, June, {WWW: Announcement}
- Ed-Media '98, Freiburg, Germany, June, {WWW: Home page}

Figure 1: Directory node fragment

Figure 1 shows part of a typical directory node. The node contain contains four conference objects and the objects are ordered alphabetically to allow linear browsing of the node.

2.2 Content and external nodes

Whereas directory nodes are highly structured, content nodes contain largely unstructured information. Examples of content nodes are descriptions and discussions of research topics, critiques of published articles, proposals for new work, and archives of work published in non Web-based forums. Figure 2 gives an example of a content node which explains the concept of a hypertext guided tour.

Guided tours The guided tour metaphor has been attributed to {ARTICLE: (Hammond 1987)}. {ARTICLE: (Marshall 1989)} also talks about guided tours, particularly the problems that authors have in selecting which nodes to include in each tour. For an online demonstration, see {PERSON: Watter}'s {WWW: Guided Tours Page}. See also {hypertext concepts: trails}, {hypertext concepts: paths}, and {hypertext concepts: footprints}.

Figure 2: Content node fragment

External nodes are Web pages owned by someone other than the SWS author and are not generally considered to be part of a SWS. However, they are important because they often contain frequently updated or copyright information that is not appropriate for including in the SWS verbatim.

2.3 Links

Clickable links are an important element of the hypertext interface and have two key purposes in the SWS: they enable a user to locate information by navigating to the appropriate node, and they allow an author to represent information about the relationship between a pair of nodes.

Our technique for making links an effective navigational device is based on link typing [8]. The set of link types in the SWS is intentionally small — to reduce 'cognitive overhead' [9] users must be able to interpret and apply link types consistently — and is defined by the author so that the types available are relevant to the domain.

The link types available include one fixed type, 'WWW', which is used exclusively for external links, and one type corresponding to each directory node. Thus, if the SWS author chooses to have directories for articles, conferences and people then appropriate types might be: WWW, ARTICLE, CONFERENCE, and PERSON. It is also convenient to treat each content node as having an associated type.

Each link in the SWS is annotated with its type and some free text. Link types are displayed to the user by prefixing the free text appropriately. A distinguished font, point size or colour is used for the prefix so that the reader can distinguish the link type from the surrounding text. The choice of font, size or colour can be made so as to reduce visual interruption in the text flow, typically by using a font size that is smaller than the normal text.

The free text for external links is chosen by the author and is typically the same as title of the external page. In contrast, the free text for content and directory links is derived from the text at the destination of the link. This is done to ensure that when the user clicks on a segment of linked free text they can quickly identify where to start reading at the destination and is particularly useful when a link points somewhere into the middle of a node. In fact, links may not be placed to arbitrary locations within nodes, they must point at text that has been defined as a heading.

This 'restriction' on the choice of free text is important for automatic insertion of links, as described in the next section.

Examples of link annotations can be seen in Figure 1 which contains a directory link with type CONFERENCE and free text "UM" and three external links annotated "WWW".

3 Tool support for Scholarly Web Sites

Having summarised the function and use of SWS nodes and links we now discuss six aspects of the SWS, each of which can be enhanced by support from automated tools. In Section 4 we will report the successes and challenges that resulted from our implementation of prototypes of these tools.

In what follows, the six aspects (and associated tools) are grouped into three categories relating to: the creation of new SWSs, the maintenance of existing SWSs, and the interfacing of existing SWSs to other, evolving information bases.

3.1 Creating new SWSs

SWS creation tools. We need a set of tools for creating new SWSs. Our experience using a SWS shows that the benefits obtained are proportional to the size of the SWS and inversely proportional to the number of times that the SWS must be abandoned for another information finding tool. To encourage new researchers to adopt SWSs we therefore require tools to import information from the most popular existing information bases, such as BibTeX files, browser bookmark files and ordinary Web pages, and to convert that information into SWS pages.

3.2 Maintaining existing SWSs

Link creation tool. A fundamental requirement for any hypertext is that its links function as intended. Yet, with HTML, it is very easy to miscode a link source or a link destination — and doing either will prevent the link from functioning as intended. Therefore, we need a tool to detect 'broken' links by checking that every link points to a defined location (in fact, to a heading) in a Web page that exists.

The same tool can be extended to replace 'hints' inserted by the SWS author with complete links. Such hints need contain only enough information in the way of keywords or substrings for the tool to pick the most likely directory or content node and the most likely destination within that node. Where several matches are possible, the author can be prompted to select the correct one.

This tool can also assist in the reorganisation of nodes within a SWS, allowing the author to move objects between directories and to move material between content nodes with the assurance that the necessary changes to links in the rest of the SWS will be made by the tool.

Link and directory presentation tool. To make the SWS an efficient tool for (re)finding information we must minimise the user's cognitive overhead in interpreting link function. As we indicated earlier, one way to do this is to employ a set of conventions for presenting links. For example, we show link types and we place certain restrictions on the free text in a link. Enforcing such conventions is tedious and error-prone so we propose the following tool.

A tool to verify link annotations by checking that every link has the type 'WWW' or a type corresponding to one of the directory or content nodes in the SWS, and by checking that all links (except external links) have free text that matches the link destination. The tool can also enforce the conventions for the typography of the link type and free text.

The same tool can be used to enforce a consistent presentation of objects in directories so that directories may be browsed linearly. This tool keeps the objects in a suitable order, date of entry or alphabetic by name for example.

The tool can also use information about object attributes to enforce a consistent presentation of the objects. For example, in a directory containing bibliographic objects the tool may use typography to highlight the title or authors of each object.

Landmark maintenance tool. As we described above, for directories to act as effective landmarks they need to be extensively interlinked with other nodes in the SWS. The high degree of interlinking soon becomes a burden on the author and so we propose a tool that can automatically insert some particular kinds of new link — without the need for intervention by the author.

'Reverse links' are an important example of links that can be inserted automatically. For each link placed into a directory by the author, a reverse link can be added from the directory back to the source of the author's link.

Links for new objects can also be placed automatically. For instance, when a new object is added for the person 'Smith' the tool can add links *from* all articles with author 'Smith' *to* the new object.

Object creation tool. A tool to suggest new objects for inclusion in a given directory. For example, suppose that an article by Smith is added to the ARTICLE directory but there is no object in the PERSON directory for Smith — then the tool can insert the Smith object.

3.3 Interfacing SWSs to other information bases

In subsection 3.1 we proposed tools for creating new SWSs by converting information from other formats. However, there are many examples of pre-existing and evolving information bases which are best manipulated in their native format and which should not be converted into SWS pages. Email and newsgroups, for example, are much better manipulated using existing email and news applications.

Interface presentation tool. What we propose here is a tool that creates 'interface nodes' for inclusion in the SWS. These nodes contain summaries or digests of items stored in non-SWS information bases and can be browsed, linked to, and annotated just as any other node in the SWS but they cannot be directly edited. Figure 3 shows two objects in an interface node that has been generated from a folder of email. Both objects have a heading that shows the sender and subject of the email, a date, and a digest of the first few lines of the message. The first object has been annotated (in SMALL CAPS) by the SWS author to provide a link from this interface node into the PROJECT and ARTICLE directory nodes. The appropriate entry in the PROJECT node will contain a reverse link back to the Herlocker message.

Continuing the example of the interface node generated from email, the tool that generates the digests for the email interface node will periodically update those digests and add or delete objects in the node as the user updates their email folder.

In addition to interfacing email folders to the SWS we could interface archives of Usenet news groups, Web sites, or Unix file hierarchies.

4 Results from developing prototype tools

Having reviewed the main elements of the SWS in Section 2 and discussed six aspects of the SWS which would be enhanced by tool support in Section 3 we now describe the successes and challenges that resulted from our development of prototype tools for supporting SWSs.

The prototypes were developed in perl and tested on a SWS of 150 content nodes and 6 directory nodes containing approximately 740 objects and 7700 links. This SWS was

Jon Herlocker **GroupLens Announcement**

- June 1997
- Greetings! The GroupLens Recommendation Service is now available to the public. We encourage everyone to return to the GroupLens web page (http://www . cd . umn . edu / Research / GroupLens) to register...
- FOR GROUPLENS SEE {PROJECT: GROUPLENS} OR {ARTICLE: RESNICK 1994}

MovieLens-info@cs.umn.edu **MovieLens Announcement**

- Tuesday
- As you have indicated your interest in being notified of any new services or other news regarding MovieLens, we are taking this opportunity to let you know...

Figure 3: Interface node fragment

constructed over a period of 10 months for the topics of hypertext, information filtering, information retrieval and related areas.

SWS creation tools. Our prototype tools can convert BibTeX and conventional HTML files into SWS nodes. These tools are simple, requiring little more than some regular expression search and replace operations. However, a different tool is needed for each kind of input and it has proven more efficient to implement quick and dirty filters and then to rely on some manual inspection and correction of the output.

Link creation tool. Our prototype tool verifies links and suggests appropriate corrections for broken internal links. While developing this prototype, we experimented with different algorithms for matching hints to possible link destinations. Clearly, the shorter the hint that the SWS author need enter, the better. However, more ambiguous hints lead in general to more possible matches being returned from which the author has to choose.

Our conclusion from testing the prototype is that even a fairly simple hint matching algorithm greatly enhances usability.

Link and directory presentation tool. Our prototype presentation tool proved simple to implement and significantly enhanced readability of the SWS. It also reduced the amount of formatting for the author to perform.

Landmark maintenance tool. With the link creation tool, the author signals that she wishes a new link to be created and the tool fills in the detail necessary to make a correct HTML link. The landmark maintenance tool, on the other hand, may make links in places where the author has not given any explicit indication that a new link is required.

Our experience developing the prototype tools strongly suggested that the tool should not make a change to the SWS without gaining the author's explicit permission first — the risk is that a wrongly inserted link is very hard to detect in a SWS with many thousands of links.

Fairly early on in testing our prototype tools we discovered that a mechanism is need for some particular kinds of automatic links to be suppressed. For instance, it is beneficial to have a link from the bibliographic details for a conference article to the corresponding object in the conference directory, but it is distracting to have the reverse links inserted (from a given conference to all articles presented at that conference).

We conclude that automatically inserted links are sometimes useful, but in general not as valuable as links explicitly placed by the author and in some may even corrupt the integrity of the information in the SWS.

Object creation tool. Just as there are disadvantages to automatically inserted links, so there can be disadvantages with an overly enthusiastic object creation tool. Use of our prototype tool suggest that directories become 'noisy' when large numbers of new objects are added automatically and that the decision of which objects to include in a directory should be left to the author.

We conclude that automatically inserted objects are not beneficial and we have therefore discontinued development of this particular tool.

Interface presentation tool. There are three practical issues with creating interface nodes. First, we know of no standard method for generating summaries that are concise enough to keep a large interface node browsable and yet selective enough to capture all information of relevance to the user.

Second, our preliminary results suggest we are going to need a different summarising method for each kind of external information base.

Third, for many external information bases it appears that several alternative orderings of objects are possible. For example, when interfacing to email folders, should the messages be ordered by date, sender, or subject? Supporting multiple interfaces for the same external information base is possible, but introduces further complications in terms of storing the annotations and links in the interface nodes.

5 Conclusions

In this paper we have described a set of tools which we believe enhance the usability of Scholarly Web Sites and increase the productivity of their users. We have also described our progress in implementing prototypes of these tools and the lessons we have learned. Whilst the project is still at an early stage, results are promising.

First, converting existing information seems generally simple, although a different tool is needed for each kind of imported file.

Second, tool support for link verification and hint completion greatly enhances usability for author and user.

Third, our experience with the landmark maintenance and object creation tools shows that a simple-minded approach to generating new links or new directory objects can produce distracting noise (i.e. irrelevant links or objects) and, in the worst case, corrupt the integrity of the SWS. To some extent the risks associated with corruption can be mitigated by visual indicators that distinguish author-placed links and objects from tool-inferred links and objects, but this also adds to the visual noise.

Fourth, we feel that the interface nodes are an appropriate way to link the SWS with external, evolving information bases but we have yet to demonstrate the effectiveness of browsing as a way to find information in, or to organise, those external information bases.

In terms of future work we are aware that we need to establish a basic kit of reliable tools for SWS creation and link and directory presentation so that we can have some test subjects evaluate the usefulness of SWSs. We also intend to put more study into the tools for landmark maintenance and interface presentation.

References

[1] V. Balasubramanian. State of the art review on hypermedia issues and applications. Available from http://www.isg.sfu.ca/~duchier/misc/hyper text_review, December 1993.

[2] Mark Bernstein. Enactment in information farming. In *Fifth ACM Conference on Hypertext*, pages 242–249, Seattle, Washington, November 1993.

[3] Loraine Blaxter, Christina Hughes, and Malcolm Tight. *How To Research.* Open University Press, Buckingham, United Kingdom, 1996.

[4] Jeff Conklin. Hypertext: An introduction and survey. *IEEE Computer*, 20:17–41, 1987.

[5] Paul Martin. Scholarly Web Sites: A general method for structuring online research notebooks. In *Third Australian Document Computing Symposium*, Sydney, Australia, August 1998.

[6] Charles T. Meadow. *Text Information Retrieval Systems.* Academic Press, Inc., 1992.

[7] Jakob Nielsen. The art of navigating through hypertext. *Communications of the ACM*, 33(3):296–310, 1990.

[8] Randy Trigg. *A Network-Based Approach to Text Handling for the Online Scientific Community.* PhD thesis, University of Maryland, 1983.

[9] Patricia Wright. Cognitive overheads and prostheses: Some issues in evaluating hypertexts. In *Proceedings of Hypertext*, pages 1–12. ACM Press, New York, 1991.