Semantic Hierarchical Abstraction of Web Site Structures for Web Searchers

L. Wookey

Department of Computer Engineering
Sungkyul Univ. Anyang, Korea
wook@sungkyul.edu

J. Geller

Department of Computer Science
New Jersey Institute of Technology, NJ
geller@homer.njit.edu

The hierarchical abstraction of a Web site is useful in organising information and reducing the number of alternatives that must be considered at any one time when browsing the site. We present such an abstraction, which is computed from a user's query and a Web site. A Web site is viewed as a directed graph with Web nodes and Web arcs, where the Web nodes correspond to HTML files (i.e., Web pages) and the Web arcs correspond to hyperlinks. The abstraction is based on computing semantic weights for the hyperlinks within the site. We have developed a pilot system, called Anchor Woman, for displaying a map representing the abstraction of a Web site. This map helps a user to avoid the feeling of being "lost in space" and makes it easier for him/her to browse the site effectively and efficiently.

Keywords: Hierarchical abstraction, web browsing, site map, web site abstraction, directed graph, semantic weights.

1. INTRODUCTION

The World-Wide Web has become ubiquitous. Web browsers allow users to access on the order of four billion Web pages. As a browser session progresses, the users sometimes feel "being lost in hyperspace" (Conklin, 1987; Lau, 1999). Users of Web information, accessible over the global Internet, require assistance by appropriate visualisation methods. The overview visualisation of Web sites has provided many benefits such as enhancing spatial context, easing jumps to specified URLs, reducing disorientation, providing a sense of the extent of a particular Web site without getting any details, and acting as a visual surrogate for the users (Risse, 1998). There is a natural mapping of a Web site structure onto a directed graph where the nodes correspond to Web pages and the arcs to URLs. A Web site is viewed as a specific directed graph that consists of an initial node (called homepage) and other nodes connected to it. Complex, static Web abstractions do little to help a user who wants to get oriented within a Web site and often cause navigation problems of their own. What is needed is a simple Web site abstraction, which is dynamically constructed based on the current query of a user. In this paper, we will present such an abstraction.

Manuscript received: 6 March 2002
Communicating Editor: Sidney A. Morris

A hierarchical abstraction provides an improvement compared to searching blindly with a full set of links. However, the abstraction loses the richness of hyperlinks. Web designers can also (to some degree) support the needs for abstraction in their site designs (Hasan, 1996). The hierarchical abstraction in this paper is introduced for the following reasons. (1) It is helpful for users in organising information and reducing the number of alternatives that must be considered at any one time (Garofalakis, 1999; Huang, 1998; Wookey, 2000). (2) In our prototype system, a direct access path is implemented from a home page to all Web pages in a site. This means that once the (hierarchical) structure of a Web site has been constructed, any page can effectively be accessed by jumping to (or clicking on) a specific Web link in the abstraction. (3) The Web site structure is dynamically constructed, using a semantic weight formula (which will be explained in following sections). Thus, the abstraction is dynamically reorganised, every time a user issues a query. (4) The abstraction highlights the most likely node to which a user will navigate next.

This paper is organised as follows. In Section 2 we suggest a hierarchy of Web objects for modeling a Web site. In Section 3 a Web site graph is processed to manipulate interior and exterior arcs correctly and a naïve tree abstraction for Web sites is analysed. A weighting mechanism for arcs is introduced in Section 4. In Section 5 the prototype system is explained. Related work is discussed in Section 6. Conclusions follow in Section 7.

2. THE WEB DATA MODEL

2.1 Web Objects

In this paper, we view the World-Wide Web as a hierarchy of Web objects with a schema represented in Figure 1. The WWW is viewed as a set of Web sites, and a Web site as a set of Web pages with arcs and content elements. We focused on the Web site that is modeled as a directed graph with Web nodes and Web arcs, where the Web nodes correspond to HTML files with page contents, and the Web arcs correspond to hyperlinks interconnecting the Web pages. This is depicted in Figure 1.

2.2 Web node definition

A Web site is defined as a directed graph G (**W, E**) consisting of a finite Web node bag **W** and a finite Web arc bag **E** of ordered pairs of Web nodes. **W** and **E** are represented as a bag of Web node

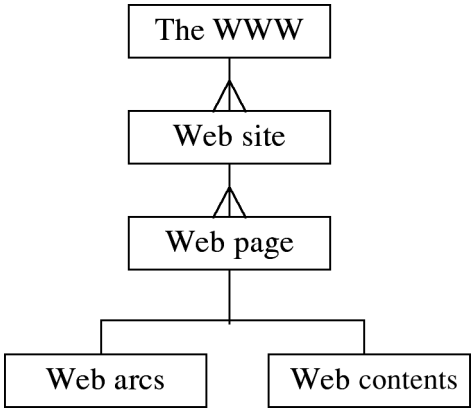

Figure 1: The schema of the World-Wide Web. (It is based on an ERD (Entity Relationship Diagram) description, but some optional details are simplified. A box represents an entity, a line a relationship. A triple line represents cardinality (one-to-many). The plain line at the bottom represents a Generalisation-Specialisation.)

elements Wi and a bag of Web arc elements (Wi, Wj) respectively, where i, j $\in \psi$ = {0, 1, 2, 3, …, n-1}, n represents the cardinality of a Web page, n =|**W**|, and ψ a Web node set. (Alternatively, for Web nodes Wi, Wj $\in$ **W**, a hyperlink from Wi to Wj can be denoted by Wi $\rightarrow$ Wj as used in Figure 2.) A Web node Wi is defined to contain additional information as follows:

$$Wi = [depth(Wi), \{Arc(Wi)\}, \{wt_{ik}\}, contents] \text{ for all } i, j \in \psi \quad (1)$$

where Wi represents an HTML file. The depth(Wi) represents how many steps the Web page Wi is away from the homepage. A Web arc bag, Arc(Wi), is represented by a pair of Web nodes {(Wi, Wj)} from a Web node Wi to Wj. $\{wt_{ik}\}$ represents a keyword bag of a Web node Wi that will be described in Section 5.

The data model, suggested in equation (1), is a nested bag notation having multiple (duplicated) keyword vectors. From a database point of view, the structure of a Web data model which has nested relations or structured objects is a kind of NFNF (non-first normal form) model (Roth, 1988). The bag of arcs, {Arc(Wi)}, may contain duplicate Web arcs such as a recursive cycle or a multi arc cycle. The duplicate Web arcs are eliminated in preprocessing steps in Section 3. On the other hand, we resolve the keyword bag in two ways. (1) One approach is to link a distinct attributive table that consists of a set of keywords with relevant numbers so that the keyword bag can be changed to a keyword set. (2) A data warehouse approach is applied to generate the mapping table between a user's query and the set of keywords (see Section 6). In other words, the NFNF properties can be resolved by a data warehouse approach (Abiteboul, 2003; Magai, 2003).

The homepage (W0) is defined as the default start up page (for example, index.html or default.asp or index.php, etc.) predetermined by the Web server. The depth(Wi) represents the *minimum* number of steps that a user needs in order to reach Wi from the homepage. For example, in Figure 2, the depth of W0 is 0 and the depth(W2) = 1.

We will use the notation "{}" for both, sets and bags. Thus, {Arc(Wi)} is the bag of URLs of the Web page Wi to other Web pages. The weight entry of a node will be discussed later. The Web page content information includes attributes of the Web page, such as the identifier, title, Meta, format, size, modified date, text, figures, multimedia files, etc. This paper deals with text-based information retrieval only. The limitation of multimedia properties can be relaxed with SMIL, XML, Audio retrieval, Graph retrieval, VIDEO retrieval, imagery retrieval, etc. (Zwol, 2000).

3. WEB ARC REPRESENTATION

3.1 Interior and Exterior Arcs

In this paper, hyperlinks (URLs) are assumed to be of two types: interior arcs and exterior arcs. The interior arcs are the URLs that point to HTML files within the same Web site and the exterior arcs point to other Web sites. We are interested in the interior arcs only, for we are focusing on the structure of a Web site that can be represented as a graph. During preprocessing of the URLs of a Web site, the standard IP addresses of every Web page are derived. The exterior arcs can be easily recognised, for they have a different server IPs and are discarded in the preprocessing phase. It should be noted that in some publications (Ng, 1998; Mendelzon, 1997), the interior arcs are additionally classified into two types, such as interior and local arcs. In our approach it is unnecessary to differentiate the interior arcs into different types. Once a Web page has been transferred from the Web server, there is no need to access the same Web page physically again. In some Web sites, there is only a frame in the default page (e.g., index.html). In that case, we use the URLs of those Web pages from which the frame includes text. It was found that the average node has roughly seven hyperlinks to other pages (Kumar, 2000).

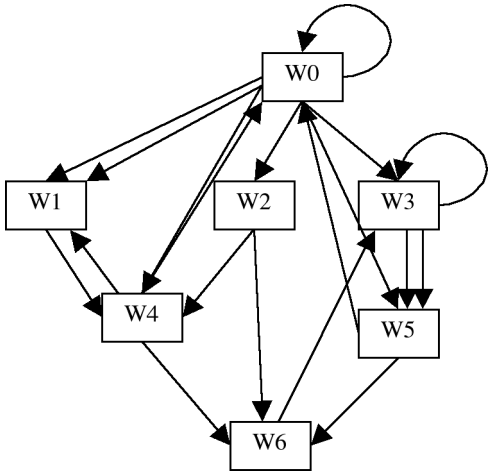

Figure 2: Example Web Site. (A box represents a Web page and an arrow a hyperlink.)

Example 1: An example Web site consists of 7 Web pages numbered from W0 to W6, connected by hyperlinks as follows. By the definition above, the node W0 has a depth(W0)=0, the bag of arcs of W0 is {(W0, W0), (W0, W1), (W0, W1), (W0, W2), (W0, W3), (W0, W4), (W0, W5)}, and the weight and the contents are not defined yet. Similarly, the node W3 has the *depth*(W3) = 1 and the bag of child URLs {(W3, W3), (W3, W5), (W3, W5)}. The relevant graph is represented in Figure 2. The textual description of the Web site looks as follows.

W0 = [0, {(W0, W0), (W0, W1), (W0, W1), (W0, W2), (W0, W3), (W0, W4), (W0, W5)}, weight, contents]
W1 = [1, {(W1, W4)}, weight, contents]
W2 = [1, {(W2, W4), (W2, W6)}, weight, contents]
W3 = [1, {(W3, W3), (W3, W5), (W3, W5)}, weight, contents]
W4 = [1, {(W4, W0), (W4, W1), (W4, W6)}, weight, contents]
W5 = [1, {(W5, W0), (W5, W6)}, weight, contents]
W6 = [2, {(W6, W3)}, weight, contents]

3.2 Conventional Spanning Tree Generation Approaches

The depth first algorithm (DFA) is easy to implement to find a spanning tree of a directed acyclic graph. But the DFA does not seem useful in the Web environment. Since most Web pages have complicated interconnections with other Web pages, DFA may bring about a deep spanning tree. Furthermore, in view of browsing performance, a long path of Web pages may imply long time delays for accessing a specific Web page. Researchers studying a mental model of Web navigation (Larson, 1998) suggest balancing the depth and the width of a Web site, which contradicts the behaviour of the DFA.

There are several advantages of applying the breadth first algorithm (BFA) to finding a spanning tree for representing a Web site graph. With the BFA, a Web page can typically be accessed quicker than with the DFA. It is easy to compute a spanning tree that minimises the average depth of nodes, using BFA. However, BFA, may give rise to flat trees such that the root node is liable to have direct links to many of the pages, which means in the Web context an overwhelming number of choices

for the user. Another weak point of BFA is related to the semantics of the Web site structure. It can be claimed that there is no semantic reason why a specific node is located at a specific place in the resulting spanning tree.

3.3 Preprocessing of the Web Arcs

A Web node may be the source of various arc types. In this paper, Web arcs are analysed in a preprocessing phase with the purpose of simplifying multiple arcs, recursive cycles, and multi-arc loops. Some problems arising here are discussed in (Huang, 1998; Mendelzon, 1997).

Multiple arcs can be defined such that one Web page has two or more hypertext links that point (directly) to the same Web page. In the preprocessing phase, multiple arcs are replaced by a single arc. This can be represented formally as follows. We use the notation {Arc(Wi)} for the bag of all links starting at Wi. For a given node Wi, if (Wi, Wj) ∈ {Arc(Wi)} and (Wi, Wk) ∈ {Arc(Wi)} then after preprocessing it must be that i≠k. For example, between W3 and W5 exist multiple (two) arcs; one of them can be eliminated. The multiple arcs can be detected and removed by eliminating "parallel" arc(s) from one Web page X to the same Web page Y.

A *recursive cycle* is defined such that a Web page has an arc pointing to itself. The reason for eliminating recursive cycles is that there is no need to access the same Web page physically again. (Local interior links can also be eliminated by this constraint.) For example, the node W3 has a recursive cycle that can be eliminated. This can be represented formally as follows: If (Wi, Wj) ∈ {Arc(Wi)} then after preprocessing it must be that i≠j.

Next, *multi-arc loops* have to be eliminated. A multi-arc loop can be defined as a set of Web pages with links that form a cycle. A cycle starting at a Web page W0 would lead back to W0. By eliminating cycles, we are producing a directed acyclic graph (DAG). Since one of the goals of this paper is to derive a tree-structured map, cycles must be eliminated. For example, the arcs (W4, W0) and (W5, W0) are liable to cause multi-arc loops, thus they have to be eliminated. This can be represented formally as follows: If (Wi, Wj) ∈ {Arc(Wi)} then after preprocessing there does not exist a sequence of arcs <(Wj, X1), (X1, X2), …(Xn-1, Xn), (Xn, Wi)> for n ≥ 0.

Example 2: By the preprocessing phase of the Web nodes (Wi) described above, the multiple arcs, the recursive cycles, and the multi-arc loops are eliminated. The resulting Web node (Wi') representation is shown below (but W1, W2 are not changed).

W0' = [0, {(W0, W1), (W0, W2), (W0, W3), (W0, W4), (W0, W5)}, weight, contents]
W3' = [1, {(W3, W5)}, weight, contents]
W4' = [1, {(W4, W6)}, weight, contents]
W5' = [1, {(W5, W6)}, weight, contents]
W6' = [2, { }, weight, contents]

4. SEMANTIC WEIGHT PRODUCTION

In our system, Anchor Woman, a graphical abstraction of a Web site is presented to a user to make navigation easier. Some experimental research indicates that graphical representations support better navigation because this type of representation more closely matches a user's mental model of the system (Hasan, 1996; Zwol, 2000; Risse, 1998). Textual tools, however, also offer advantages by allowing users to rapidly grasp the extent of a site and to search visually, in an efficient manner, for particular information (Glover, 2002). The tree abstraction, derived using the BFA, is simple and

easy to implement. However, it is not useful for finding a significant page in a Web site or for clustering pages based on semantics.

To overcome the above problems, it is possible to assign a weight to a Web page, which reflects how important it is (Chang, 1999; Chen, 1999; Garvano, 1999; Garofalakis, 1999). There are several alternatives how to compute document weights, such as by probabilistic models (Kumar, 2000), and by natural language analyses (Brasethvik, 2001). In this paper, we introduce a keyword-based weight computed from a query and a Web page and assign it to the page.

One common way to represent a Web node Wi is to use a word vector weighted by *tf-idf* (term frequency times inverse document frequency). This measure balances a term's frequency in the document and its frequency in a collection of documents. In such a vector Wi' = < wt_1',…,wt_m'>^{T}, each wt_i' is the product of a term frequency (*tf*) factor and an inverse document frequency (*idf*) factor. The *tf* factor in the i^{th} position is equal (or proportional) to the frequency of the i^{th} word within the document. The *tf* factor itself is sometimes normalised by dividing it by the frequency of the most-frequent non-stop term in the document as $tf_{norm} = tf/tf_{max}$. The *idf* factor corresponds to the content discrimination power of the i^{th} word: a word that appears rarely in the document set has a high *idf*, while a word that occurs in a large number of documents has a low *idf*. Typically, *idf* is computed by $[df(wt_i)/N]^{-1}$, and most often the $log_2[N/ df(wt_i)]$ is used, where N is the total number of documents and $df(wt_i)$ is the number of documents containing the i^{th} word. (If a word appears in every document, its discriminating power is 0. If a word appears in a single document, its discriminating power is maximal.) Once Wi' has been computed, the normalised vector W*i* is typically obtained by dividing each wt_i' by its norm.

We will now produce the weights of Web nodes in terms of a user query and *tf-idf*. These weights will be used to determine the topological ordering of Web nodes in our Web site map. A user issues a query *Q* according to his/her interests in a Web page Wi. We assume a representation of the query $Q=\{q_i\}$ for keywords q_i relative to Web pages Wi = $\{wt_j\}$, where q_i and wt_j are in the same language Σ for all $i, j \in \psi$. Then the weight of a specific Web page is defined as a scalar derived as the inner product of the query vector with the Web page vector:

$$\text{weight(Wi)} = Q \bullet Wi = [q_i] \bullet [(tf_i \cdot \log_2[N / df(wt_i)])]^T = \sum_i q_i \cdot (tf_i \cdot \log_2[N / df(wt_i)]) \quad (2)$$

where *Q* represents a query vector of query keywords q_i, for $i \in \psi$.

Using equation (2) the details of the weight computation are implemented in Figures 3(a) and 3(b). For the example 3-1, if a user generates a query *Q*=<site, graph, structure, visualisation>, the weight(W0) = <1, 1, 1, 1>•<0, 0.37, 0.32, 0.09>^{T} = 0.78. Similarly, the weight(W1) = <1, 1, 1, 1>•<0, 0.55, 0.16, 0.18>^{T} = 0.89, the weight(W2) = 0.09, the weight(W3) = 0.51, the weight(W4) = 0.00, the weight(W5) = 1.23, and the weight(W6) = 0.18. In Figure 3(b) W vectors appear as columns.

5. THE PROTOTYPE SYSTEM

5.1 Frame and Arc Extractor

The Anchor Woman system was developed to provide users with high-level abstractions of Web sites. The system consists of an engine and a viewer. The *engine* provides a frame and arc extractor module, a keyword database, and a weight calculation module. The *viewer* displays a Web site structure as a "tree view," and includes the HTML source, and a browser window. A screen shot of the viewer is shown in Figure 5. Anchor Woman v2.0, has been implemented with VB 6.0 as a client and with SQL SERVER 7.0 as the server database.

The system begins at a homepage which a user enters or which is identified by a search engine. The anchors of the Web pages are classified into two categories, (interior) anchors and frames. As explained above, the system extracts links within the Web site, but ignores internal bookmarks or exterior arcs. If the homepage consists of a frame without any interior anchors, the system extracts relevant anchors from the HTML code, which the frame includes. The resulting anchors with page contents are stored in the database. Those can be clicked on in the site map of the system, which is represented in Figure 5.

5.2 Weight Calculation Module

Next the system generates the weights of Web pages relative to a given query. If we select four search terms 'site, graph, structure, visualisation,' then the weights are calculated as in Figure 3(a). The word 'site,' for example, appears 5 times in the homepage W0 (the page number in the system appears as page 1), and in turn 2, 1, 5, 2, 6, and 2 times in W2.html to W7.html, respectively. Finally, we get the normalised vectors as follows: W0 = *<0, 0.37, 0.32, 0.09>*, W1 = *<0, 0.55, 0.16, 0.18>*, …, W7 = *<0, 0.18, 0, 0>* (Figure 3(b)).

	site	graph	structure	visualizat
1	5	2	2	1
2	2	3	1	2
3	1	0	0	1
4	5	0	1	4
5	2	0	0	0
6	6	4	2	2
7	2	1	0	0
N	7	7	7	7
DF	7	4	4	5
IDF	0	0,24	0,24	0,15

(a) Term Frequency and Relevant *TF* and *IDF* Values

Word	W'						
site	0,00	0,00	0,00	0,00	0,00	0,00	0,00
graph	0,37	0,55	0,00	0,00	0,00	0,73	0,18
structure	0,32	0,16	0,00	0,16	0,00	0,32	0,00
visualizat	0,09	0,18	0,09	0,35	0,00	0,18	0,00

(b) Normalised Weight Vector Generator

Figure 3: Weight Production Module of the System

5.3 Tree Generation Steps

In order to abstract a Web site graph into a tree structure, the following three steps were introduced in the Anchor Woman system: (1) Depth Extraction, (2) Semantic Weight Production, and (3) Spanning Tree Generation. The Depth Extraction step is needed to compute the depth of a node in a Web site and to count the number of incoming arcs of the node. The Spanning Tree Generation step converts the graph of the Web site into a tree structure. The Weight Calculation step updates the weight of a node from the current position up to the parameter level's parent.

The Spanning Tree Generation step converts the graph into a tree. The primary rule for selecting a node with multiple parents is that the node with the largest number of parents should be selected first. The secondary rule is that if there are many nodes with the same number of parents, and then the (topologically) upper node should be selected. In this context, 'upper' means the node that is

closer to the home page (which is determined by the *depth* parameter). If there exist ties both in the *indegree* as well as in the topological position, then one of the nodes is randomly selected.

As mentioned before, it is assumed that the difference of the weights of two Web pages indicates how closely related they are. We use the Euclidean distance of weights to measure which of multiple parents is closest to a node. When there are several parents, we always choose the closest one as tree parent.

Example 3: The Web node W4 (in Figure 4) has three parents (W0, W1, and W2). The distances from W4 to the three parents are |0.78–0.00|, |0.89–0.00|, and |0.09–0.00| respectively. The distance to W2 is the smallest of all. Thus, the node W2 becomes the parent of W4. If we continue to the end, then the results are as follows. The next node to be processed is W6, because the maximum indegree of W6 is 3. By comparing the distances ($Min\{0.18, 0.09, 1.05\}=0.09$), W2 becomes W6's parent. Then the node W0 becomes W1's parent. For W3, W0 is its parent.

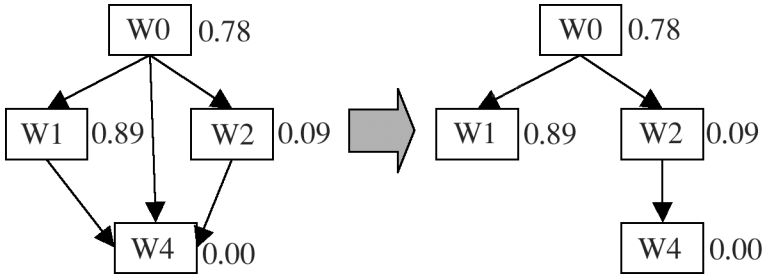

Figure 4: Example of Resolving Multiple Parents according to the Distance from a Web Node W4

5.4 Assessment of the System

The system starts at a URL that may be given by the user or via a search engine such as Yahoo, Lycos, Google, etc. Once a Web site (starting at index.html) has been transferred, our steps are applied. The system calculates the necessary weights from the user query and the page vectors, and generates a hierarchical abstraction of the Web site. A result screen of the sample site (http://www.perinatal.org) is represented in Figure 5. The resulting hyperlink structure has the added advantage of human-understandable labels (in the form of the page names) and a uniform granularity of detail, both of which are lacking in clustering steps (Wang, 2000).

The problem of finding a tree structure of a Web site from a directed graph is $n \log n$, for there must exist a Web page having the highest weight within a Web site (Garvano, 1999; Wookey, 2000). If the problem domain were enlarged to an Intranet (Chen, 1999) or the whole Web (Kumar, 2000; Mendelzon, 1997), then the time complexities would be exponential or NP-hard respectively (Ng, 1998).

One of the strong points of our system is to provide an abstract structure with respect to a query of a user. That means that if a user wants to find many Web pages in a site, issuing only one single query, our system facilitates this. It is likely that he will already find what he is looking for by clicking on the first link in the abstract map, because the first node has the highest weight. In other words, it provides a guide to the searcher in terms of the weights of different pages. For example, the four nodes at the first level (*depth* = 1) are sorted as follows: W5 (1.23), W1 (0.89), W3 (0.51), and W2 (0.09). The two nodes at the second level (*depth* = 2) are W6 (0.18) and W5 (0.00). This is the order (top to bottom) in which links to these nodes appear.

Figure 5: The Tree Structure with Weights and the Browser Session of Anchor Woman

6. RELATED WORK

There have been efforts to visualise Web information with a number of different approaches. Some experimental approaches maintain that graphical representations support better navigation because this type of representation more closely matches a user's mental model of the Web (Larson, 1998). Risse *et al* (1998) presented a conceptual approach to generating three-dimensional VRML scenes dynamically from the information stored in a database system. They extended VRML with a new data type in order to support server side information generation as well as to support a trigger mechanism.

Textual tools, however, exhibit other advantages by allowing users to rapidly circumscribe the extent of a Web site and to search visually in an efficient manner for a particular item of information (Ng, 1998). Hasan *et al* (1996) developed a system that allows users to generate graphical overviews of browsing sessions in a Web-browsing tool. The system provides views of the history of the navigation and of the containment of links in visited documents. The views, however, do not take into account the weights of documents corresponding to a user's queries.

Garofalakis *et al* (1999) developed the *Soala* system with an arc-editing step based on the relative page popularity. It can automatically revise a Web site's page structure to create a more effective hypertext scheme. Unless the mesh structure of a Web site is changed to a binary tree, the manipulation based on page popularity appears to be trifling. The SiteTree system (Pilgrim, 1999) operates on a client side WWW site map that represents the contents of documents visually on a

screen in such a way as to provide orientation and interactive navigation. But the hierarchical scheme used in SiteTree is limited by its breadth first approach.

Huang *et al* (1998) suggested a dynamic Web navigation model that filters the Web, producing a tree with a visual surrogate using a spring step. The dynamic Web graph model simplifies a real-world Web graph with a meaningful formal approach. It is a similar approach to ours in weeding out the internal anchors of the displayed graph. But the tree construction does not incorporate a weight or a relevance for each Web node while our model does. The displayed tree does not suggest to the user a next step or a direction. One cannot explain why the displayed tree structure is better than that of alternative trees. It just displays a trajectory that the user has followed.

Kumar *et al* (2000) considered a Web random graph model according to the number of incoming links and the reference number of major Web sites. The Cha-Cha model (Chen *et al*, 1999) imposes an organisation on Web site search results by recording the shortest paths, in terms of hyperlinks, from a server root page to every Web page within the intranet. After the user issues a query, the shortest paths are dynamically combined to form a hierarchical outline of the context in which the search results occur. The goals of the Cha-Cha system do not include the analysis of Web sites and the structuring of a Web site (or an intranet) is not of interest, either.

There are other related topics, such as clustering analysis of effective retrieval in a distributed environment (Wang, 2000), concept based relevance, text-database resource discovery (Chang, 1999; Garvano, 1999), and formal representation of Web sites (Kumar, 2000, Mendelzon, 1997).

7. CONCLUSIONS

This paper views a Web site as a directed graph where the nodes correspond to Web pages and the arcs to URLs. The overview visualisation of Web sites provides a number of benefits, such as highlighting the spatial context of a query, reducing disorientation, allowing direct jumps to specified URLs, providing a sense of the extent of a particular Web site, and acting as a visual surrogate.

Our hierarchical abstraction is a compromise between searching blindly with a full set of links and maintaining pivot weights in a Web "jungle." It is useful in organising information and reducing the number of alternatives that must be considered at any one time. A direct access path for each Web page is implemented in the prototype system.

The main contributions of this paper can be summarised as follows:

- A formal representation is used to tackle the abstraction of a Web site graph.
- The abstracted structure includes semantic weights by which the Web nodes are organised *dynamically* in the system.
- All the nodes in a Web site topology are sorted by weight. By assigning semantic weights to Web pages it becomes possible to decide which of several Web pages is the best for a given query. In other words, the abstraction provides a dynamic guide to the searcher that takes the user query into consideration.
- We have developed a pilot system, called Anchor Woman, with a site map user interface for browsing Web sites effectively and efficiently.

ACKNOWLEDGEMENT

We would like to thank T. Hwang for his help with implementing a version of Anchor Woman. This work was partially supported by the Korea Science and Engineering Foundation (KOSEF) through the Advanced Information Technology Research Center (AITrc).

REFERENCES

ABITEBOUL, S. (2003): Managing an XML warehouse in a P2P context. Proc. *CAiSE: International Conference CaiSE,* Klagenfurt, Austria, LNCS 2681: 4–13, Springer.

BRASETHVIK, T. and GULLA, J. A. (2001): Natural language analysis for semantic document modeling. *Data and Knowledge Engineering* 38(1): 45-62.

CHANG, C. and HSU, C. (1999): Enabling concept-based relevance feedback for information retrieval on the WWW. *IEEE TKDE* 11(4): 595–609.

CHEN, M., HEARST, M., HONG, J. and LIN, J. (1999): Cha-Cha: A system for organizing intranet search results, *Proc. USENIX Symposium on Internet Technologies and Systems*, Boulder, CO, USA: 11–14.

CONKLIN, J (1987), Hypertext, an introduction and survey. *IEEE Computer* 20(9): 17–41.

GAROFALAKIS, J., KAPPOS, P. and MOURLOUKOS, D. (1999): Web site optimization using page popularity, *IEEE Internet Computing* 3(4): 22–29.

GARVANO, L., GARCIA-MOLINA, H. and TOMASIC, A. (1999): GIOSS: Text-source discovery over the internet. *ACM TODS* 24(2): 229–264.

GLOVER, E. J., TSIOUTSIOULIKLIS, K., LAWRENCE, S., PENNOCK, D. M., and FLAKE G. (2002): Using web structure for classifying and describing web pages. *Proc. WWW*: 562–569.

HASAN, M. Z., MENDELZON, A. O. and VISTA, D. (1996): Applying database visualization to the world wide web, *ACM SIGMOD RECORD* 25(4): 45–49.

HEARST, M., A. (1999): The use of categories and clusters in organizing retrieval results. In *Natural Language Information Retrieval*, TOMEK STRZALKOWSKI (eds), 333–374. Kluwer Academic Publishers.

HUANG, M., EADES, P., WANG, J. and DOYLE, B. (1998): Dynamic web navigation with information filtering and animated visual display, *Proc. APWeb98*: 63–71.

KUMAR, R. RAGHAVAN, P., RAJAGOPALAN, S., SIVAKUMAR, D., TOMPKIN, A. and UPFAL, E. (2000): The web as a graph, *Proc. ACM SIGMOD-SIGACT-SIGART on PODS*: 1–10.

LARSON, K. and CZERWINSKI, M. (1998): Web page design: Implications of memory, structure and scent for information retrieval, *Proc. CHI International Conference on Human Factors in Computing Systems*, Los Angeles, California, USA. 25–32, ACM Press.

LAU, T., ETZIONE, O., and WELD D. S. (1999): Privacy interfaces for information management, *Communications of the ACM* 42(10): 89–94.

MAGAI, R. and WOOKEY, L. (2003): Designing and implementing a geospatial data warehouse in enterprise forestry. *Proc. IUFRO International Conference on Information Interoperability and Organization For Global Forest Information Systems*. Elsevier (in print).

MENDELZON, A. O. and MILO, T. (1997): Formal model of web queries, *Proc. ACM PODS*: 134–143.

NG, W. K., LIM, E. P., HUANG, C. T., BHOWMICK, S., and QIN, F. Q. (1998): Web warehousing: An algebra for web information, *Proc. of the IEEE ADL*: 228–237.

PILGRIM, C. J. and LEUNG, Y. K. (1999): Designing WWW site map systems, *Proc. DEXA*: 253–258, Springer-Verlag.

RISSE, T., LEISSLER, M., HEMMJE, M., ABERER, K. and KLEMENT, T. (1998): Supporting dynamic information visualization with VRML and databases, *Proc. Workshop on New paradigms in information visualization and manipulation*: 69–72.

ROTH, M. M., KORTH, H. F. and SILBERSCHATZ, A. (1988): Extended algebra and Calculus for Nested Relational Databases, *ACM TODS* 13(4): 389–417.

WANG, S., GAO, W., LI, J., HUANG, T., and XIE, H. (2000): Web clustering and association rule discovery for web broadcast. *Proc. WAIM*, Shanghai, China, LNCS 1846, Springer.

WOOKEY, L. and KIM, J. H. (2000): Visualization of web site information with semantic weights, *Proc. Int'l Conference on Internet Computing*: Las Vegas, USA, 253–258.

WOOKEY, L., HWANG, Y., KANG, S., KIM, C., KIM, S., and LEE, Y. (2002): Self-maintainable data warehouse views using differential files. *Proc. DEXA*: 216–225, Springer-Verlag.

ZWOL, R. and APERS P. (2000): The webspace method: On the integration of database technology with multimedia retrieval. *Proc. CIKM International Conference on Information and Knowledge Management*, McLean, VA, USA, 438–445, ACM Press.

BIOGRAPHICAL NOTES

L. Wookey

L. Wookey is an Associate Professor in the Department of Computer Engineering, Sungkyul University, S. Korea. He received the BSc (1987), MSc (1991), and the PhD (1996) in industrial engineering from Seoul National University, Korea. He completed an MSE in the Department of Computer Science, Carnegie-Mellon University in 2000. He received a diploma of TEFL (Teacher for English as a Foreign Language), ISS Canada. He was a visiting professor in the Department of Computer Science, UBC, Canada from March 2002 to August 2003. Dr Wookey has published many journal and conference papers in Distributed Database Systems, Data Warehouses, and Web IR. He is an ACM and IEEE member.

J. Geller

J. Geller is a professor in the Computer Science Department of the New Jersey Institute of Technology, Director of the Semantic Web and Ontologies Laboratory and Vice Chair of the MSc and PhD programs in Biomedical Informatics. He received an MSc degree (1984) and PhD degree (1988) in Computer Science from the State University of New York at Buffalo. Dr. Geller has published numerous journal and conference papers in knowledge representation, parallel artificial intelligence and medical informatics. His current research interests include ontologies, medical vocabularies and Web mining. He is an AAAI and ACM member and a past SIGART Treasurer.