A Survey on Website Structure Improvement Techniques

ChhayaShejul[1], Padmavathi[2]

[1]Pune University, GHRCEM, Wagholi, Pune, India

[2]Assistant Professor, GHRCEM, Pune University, Wagholi, Pune, India

Abstract: ***Evaluation of the link structure of a web site and its redefinition to achieve increased efficiency with regard to easier information retrieval is a common problem in website development. A primary reason is that the web developers' understanding of how a website should be structured can be considerably different from that of the users. This paper includes a technique that discovers the gap between Web site designers' expectations and users' behavior. The former are assessed by measuring the inter-page conceptual relevance and the latter by measuring the inter-page access co-occurrence. Also, in this paper we present a survey of the use of Web mining for Web personalization and web transformation approaches.***

Keywords: web personalization, web transformation, Website design, conceptual relevance, access co-occurrence

1. Introduction

The free-for-all Internet has led towards a massive publication of information. The Web nowadays seems to serve as a worldwide free library. This chaotic nature of the WWW is mirrored in the structure of websites, making it essentially haphazard. The continuous growth in the size and use of the World Wide Web imposes new methods of design and development of on-line Information Services. Most Web structures are large and complicated and users often miss the goal of their inquiry, or receive ambiguous results when they try to navigate through them.

On the other hand, the e-business sector is rapidly evolving and the need for Web market places that anticipate the needs of the customers is more than ever evident. Therefore, the requirement for predicting user needs in order to improve the usability and user retention of a Web site can be addressed by personalizing it. Web personalization is defined as any action that adapts the information or services provided by a Web site to the needs of a particular user. This research work was partially supported by the IST-2000-31077/I-Know U Mine R&D project funded by the European Union.

A primary cause of poor website design is that the web developers' understanding of how a website should be structured can be considerably different from those of the users [1], [2]. Such differences result in cases where users cannot easily locate the desired information in a website. This problem is difficult to avoid because when creating a website, web developers may not have a clear understanding of users' preferences and can only organize pages based on their own judgments. However, the measure of website effectiveness should be the satisfaction of the users rather than that of the developers. Thus, Webpages should be organized in a way that generally matches the user's model of how pages should be organized.

2. Related works

To assist the interpretation of frequent navigation patterns, some features that can characterize these patterns must be employed. The use of user classification is one approach, where frequent navigation patterns are interpreted differently based on the importance of users. For example, Spiliopoulou et al. compare navigation patterns of customers (Web site users who have purchased something) with those of non-customers. This comparison leads to rules on how the site's topology should be improved to turn non-customers into customers. Another approach uses hyperlink connectivity. For example, Perkowitz and Etzioni [2] find clusters of pages that tend to co-occur in visits but are not connected. For each cluster, an index page consisting of hyperlinks to the pages in the cluster is generated. In this way, more effective traversal between these pages will be achieved.

One drawback of these approaches is that they lack techniques to evaluate infrequent navigation patterns. In other words, they extract no navigation patterns that should be frequent (in the ideal Web site) but are actually infrequent because of poor site design. To find navigation patterns that should be frequent, we presume that page content analysis, as well as access log analysis, will be important.

Another drawback is that these approaches mostly concentrate on the hypertext topology and suggest no clues to improving the site design at the page layout level. In other words, if pages suggested for further improvement are already connected, Web site designers have to work on them without any help. The quality of page layout is dependent on many factors (e.g., topics, objectives, users, size, languages, the use of multimedia techniques, visual/logical consistency with other pages, and so forth), analysis of site-specific page layout features will be necessary.

The growth of the Internet has led to numerous studies on improving user navigations with the knowledge mined from webserver logs and they can be generally categorized in to web personalization and web transformation approaches. Web personalization is the process of "tailoring" web pages to the needs of specific users using the information of the users' navigational behavior and profile data. Perkowitz and Etzioni describe an approach that automatically synthesizes index pages which contain links to pages pertaining to

particular topics based on the co-occurrence frequency of pages in user traversals, to facilitate user navigation. The methods proposed by Mobasher et al. [13], [14], [15] and Yan et al. [16] create clusters of users' profiles from weblogs and then dynamically generate links for users who are classified into different categories based on their access patterns. Nakagawa and Mobasher [17] develop a hybrid personalization system that can dynamically switch between recommendation models based on degree of connectivity and the user's position in the site.

Web transformation, on the other hand, involves changing the structure of a website to facilitate the navigation for a large set of users instead of personalizing pages for individual users. Fu et al. [20] describe an approach to reorganize web pages so as to provide users with their desired information in fewer clicks. However, this approach considers only local structures in a website rather than the site as a whole, so the new structure may not be necessarily optimal. Gupta et al. [10] propose a heuristic method based on simulated annealing to re-link web pages to improve navigability. This method makes use of the aggregate user preference data and can be used to improve the link structure in websites for both wired and wireless devices. However, this approach does not yield optimal solutions and takes relatively a long time (10 to 15 hours) to run even for a small website. Lin [11] develops integer programming models to reorganize a website based on the cohesion between pages to reduce information overload and search depth for users. In addition, a two-stage heuristic involving two integer-programming models is developed to reduce the computation time. However, this heuristic still requires very long computation times to solve for the optimal solution, especially when the website contains many links.

Besides, the models were tested on randomly generated websites only, so its applicability on real websites remains questionable. To resolve the efficiency problem in [11], Lin and Tseng [19] propose an ant colony system to reorganize website structures. Although their approach is shown to provide solutions in a relatively short computation time, the sizes of the synthetic websites and real website tested in [19] are still relatively small, posing questions on its scalability to large-sized websites.

3. Discovering the Gap between Website Designers Expectations and Users Behaviors

In this section, we introduce a technique that discovers the gap between Web site designers' expectations and users' behavior. In this approach, the former is assessed by measuring the inter-page conceptual relevance whereas the latter by measuring the inter-page access co-occurrence.

3.1 Measurement of Conceptual Relevance

We employ the vector space model to measure the inter-page conceptual relevance. Given a Web site, we first remove HTML tags from each page. Second, we obtain content words (nouns, verbs, and adjectives) by performing morphological analysis and stop-word removal. Third, we compute the frequency of content words for each page. Fourth, we generate a list of content words weighted with their frequency. This list is viewed as a vector that represents page contents.

$p_i = (\omega_{i1}, \omega_{i2}, \ldots, \omega_{ik}, \ldots, \omega_{in})$, Where ω_{ik} the weight of the kth content word and n is is the number of distinct content words found in the Web site. We finally measure the inter-page conceptual relevance (SimC) for each page pair p_i and p_j using the cosine similarity formula as follows.

$$SimC(p_i, p_j) = \frac{\sum_{k=1}^{n}(\omega_{ik} \cdot \omega_{jk})}{\sqrt{\sum_{k=1}^{n}(\omega_{ik})^2 \cdot \sum_{k=1}^{n}(\omega_{jk})^2}},$$

Where, SimC is 0 if one of the pages contains no content words. If the number of content words that appear in both pages is 0, the value of SimC is also 0. If two pages contain identical content words with the same frequency (i.e., vectors of two pages are identical), the value of SimC is 1. Note that all pages are equally informative in the vector space model because of the page length normalization.

3.2 Measurement of Access Co-occurrence

We modify the vector space model to measure the inter-page access co-occurrence. Given access log data, we first remove accesses from search engine robots and proxy servers using heuristics (e.g., access to /robots.txt; exhaustive access in a short period). Second, we count IP addresses for each page. Third, we generate a list of IP addresses weighted with their frequency. This list is viewed as a vector that represents page users. We finally measure the inter-page access co-occurrence (SimA) for each page pair using the aforementioned cosine similarity formula.

$$SimA(p_i, p_j) = \frac{\sum_{k=1}^{t}(\lambda_{ik} \cdot \lambda_{jk})}{\sqrt{\sum_{k=1}^{t}(\lambda_{ik})^2 \cdot \sum_{k=1}^{t}(\lambda_{jk})^2}},$$

Where λ_{ik} is the weight of the kth IP address that visited p_i, and t is the number of distinct IP addresses found in the access log data. SimA is 0 if one of the pages has never been visited by anyone. If the number of users who visit both pages is 0, the value of SimA is also 0. If two pages are visited by identical users with the same frequency, the value of SimA is 1. Note that the inter-page access co-occurrence in this model is independent of the page popularity because the number of visits is normalized. It is also independent of hyperlink connectivity.

3.3 Gap Discovery

Before comparing the inter-page conceptual relevance with the access co-occurrence for each pair of pages, we introduce the notions of *content page* and *index page*. While the former is an ordinary page that conveys conceptual contents to users, the latter is a functional page for navigational help. In general, the index page has multiple

references to content pages of various topics, and tends to include content words of various topics without conceptual consistency. Consequently, measuring the conceptual relevance between the content page and the index page (or between two index pages) generates noisy data with a meaningless value. We therefore discard index pages in advance. The question here is how we should distinguish index pages. Because many pages actually have characteristics of a content page and an index page, the boundary between them can be subjective. In this paper, we use the number of references in a page as a guide, based on the intuitive idea that an index page should have more references than a content page. We consider a page with more than N references as an index page. To determine the optimal value of N, we compute the correlation coefficient (R_{CA}) between the inter-page conceptual relevance (SimC) and the access co-occurrence (SimA) using the following formula.

$$R_{CA} = \frac{s^2_{CA}}{\sqrt{s^2_C \cdot s^2_A}} \qquad (-1 \le R_{CA} \le 1),$$

Where

$$s^2_{CA} = \sum_{i=1}^{m} (SimC_i - \overline{SimC})(SimA_i - \overline{SimA}),$$

$$s^2_C = \sum_{i=1}^{m} (SimC_i - \overline{SimC})^2,$$

$$s^2_A = \sum_{i=1}^{m} (SimA_i - \overline{SimA})^2,$$

Where $SimC_i$ is the value of conceptual relevance for the ith pair of pages, $SimA_i$ is the value of access co-occurrence for the ith pair of pages, and m is the number of page pairs.

The correlation coefficient (R_{CA}) measures the degree of linear relationship between two variables (SimC and SimA). If there is an exact linear relationship, it is 1 or -1 depending on whether the variables are positively or negatively related. If there is no relationship, it is 0 (see [6] for more detail). Thus, if index pages (which generate noisy data) are properly discarded, the correlation coefficient will tend toward 1. Figure 1 show the result of this computation, where we used the Fuji Xerox Web site. The peak is observed at N = 8, i.e., pages with more than eight references can be considered as index pages in this Web site.

The correlation coefficient above can also be used as a criterion to indicate the overall design quality of the Web site. It would tend toward 1 if the overall site design were ideal. However, it does not indicate where the Web site requires improvement. For this purpose, we plot the inter-page conceptual relevance versus the access co-occurrence for each page pair as shown in figure 2. The straight line in the figure is a fit to the plot using least square regression. The markers on the lower right show the page pairs that rarely co-occur in visits even though they are conceptually related. Web site designers can locate the URLs of these pages by pointing at the markers. Our technique, depending on the size and quality of the Web site, may find many page pairs that should be improved, but it can also give a structural view for browsing assistance. The technique first transforms the set of page pairs into a set of distinct pages, and then applies a content-based agglomerative hierarchical clustering algorithm [13] to the new set.

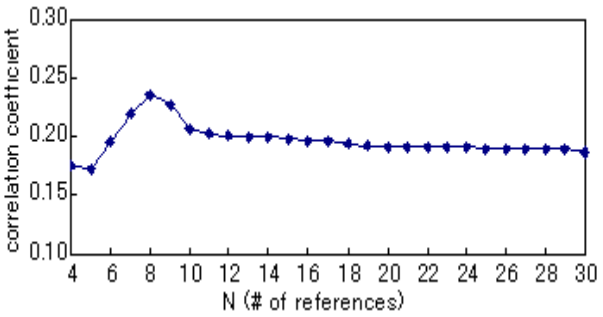


Figure 1: Index page determination

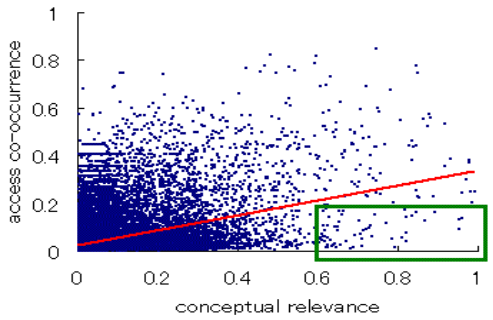


Figure2: Conceptual relevance versus access co-occurrence

It finally shows page clusters, which can help Web site designers to understand the design problem at a more abstract level. For example, we found that many pages in the lower right area in figure 2 were about products of the company.

4. Web Personalization

Web site personalization can be defined as the process of customizing the content and structure of a Web site to the specific and individual needs of each user taking advantage of the user's navigational behavior. The steps of a Web personalization process include: (a) the collection of Web data, (b) the modeling and categorization of these data (pre-processing phase), (c) the analysis of the collected data and (d) the determination of the actions that should be performed. The ways that are employed in order to analyze the collected data include content-based filtering, collaborative filtering, rule-based filtering and Web usage mining. The site is personalized through the highlighting of existing hyperlinks, the dynamic insertion of new hyperlinks that seem to be of interest for the current user, or even the creation of new index pages.

- ***Content-based filtering*** systems are solely based on individual users' preferences. The system tracks each user's behavior and recommends them items that are similar to items the user liked in the past.
- ***Collaborative filtering*** systems invite users to rate objects or divulge their preferences and interests and then return information that is predicted to be of interest for them. This is based on the assumption that users with similar behavior (for example users that rate similar objects) have analogous interests.

- ***Rule-based filtering*** the user is asked to answer to a set of questions. These questions are derived from a decision tree, so as the user proceeds on answering them, what she/he finally receives as a result (for example a list of products) is tailored to their needs. Content-based, rule-based and collaborative filtering may also be used in combination, for deducing more accurate conclusions.

The block diagram illustrated in Figure 3 represents the functional architecture of a Web personalization system in terms of the modules and data sources that were described earlier.

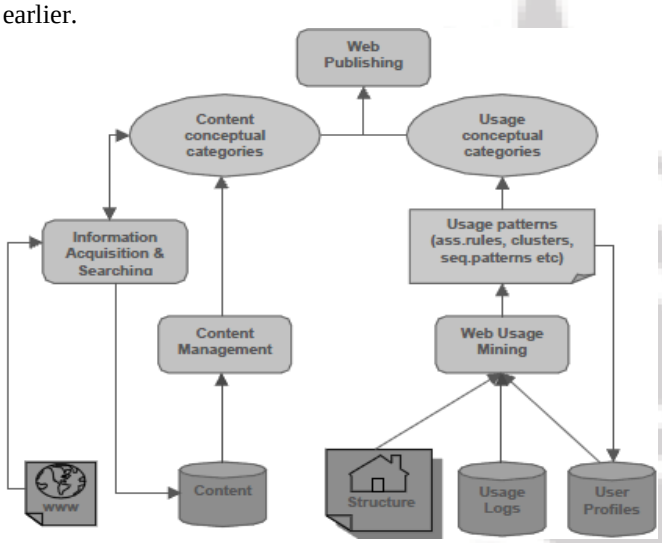


Figure 3: Modules of a Web personalization system

The content management module processes the Web site's content and classifies it in conceptual categories. The Web site's content can be enhanced with additional information acquired from other Web sources, using advanced search techniques. Given the site map structure and the usage logs, a Web usage miner provides results regarding usage patterns, user behavior, session and user clusters, click-stream information etc. Additional information about the individual users can be obtained by the user profiles. Moreover, any information extracted from the Web usage mining process concerning each user's navigational behavior can then be added to his/her profile. All this information about nodes, links, Web content, typical behaviors and patterns is conceptually abstracted and classified into semantic categories. Any information extracted from the interrelation between knowledge acquired using usage mining techniques and knowledge acquired from content management will then provide the framework for evaluating possible alternatives for restructuring the site. A publishing mechanism will perform the site modification, ensuring that each user navigates through the optimal site structure. The available content options for each user will be ranked according to user's interests.

5. Web Transformation

Considering transformations approaches mainly focuses on developing methods to completely reorganize the link structure of a website. The approach consists of three steps: pre-processing, page classification, and site reorganization. In pre-processing, pages on a Web site are processed to create an internal representation of the site. Page access information of its users is extracted from the Web server log. In page classification, the Web pages on the site are classified into two categories, index pages and content pages, based on the page access information. A page classification algorithm has been developed which uses data about a page's type, structure, and usage to determine its category. After the pages are classified, in site reorganization, the Web site is examined to find better ways to organize and arrange the pages on the site. An algorithm for the reorganization of the site has been developed.

5.1 Preprocessing

There are three tasks in Preprocessing. The first is Web site Preprocessing to obtain the current Structure of a Web site, i.e., how the pages are linked. The second is server log preprocessing to organize access records into sessions. The third is to collect access information for the pages from the sessions.

5.1.1 Web Site Preprocessing

The purpose of this phase is to create an internal data structure to represent the Web site. The Web site is represented as a directed graph in which a page is a node and a link is an arc. Each page of the Web site is parsed sequentially and the links in the page (tags beginning with <A HREF>) are extracted. Each page is assigned a unique page identifier (PID). For each page, PIDs of pages which has a link to it (called its *parents*) and pages which it links to (called its *children*) are stored. Currently, the Web pages are assumed to be static. Dynamic pages such as those generated by CGI or other server-side scripts are ignored. All non-HTTP references, e.g., "ftp://", "gopher://", "mail to:", etc., are filtered out because they do not represent site structure. In addition, all references to pages on other sites, e.g., a reference to Adobe site for Acrobat reader, are also removed. This is reasonable since these pages are not part of the Web site and cannot be modified, thus should not be included in the reorganization process. Also, multiple links between two pages are treated as one and intra-page links (an intra-page link is a link to the page it is in) are ignored.

5.1.2 Server Log Preprocessing

Since a lot of irrelevant information for Web usage mining such as background images is also included in the server log, it has to be processed first. A number of preprocessing algorithms and heuristics exist [5]. The steps involved in preprocessing of the server log are as follows.

- Records about image files (.gif, .jpg, etc) are filtered as well as unsuccessful requests (return code not 200).
- Requests from the same IP address are grouped into a session. A timeout of 30 minutes is used to decide the end of a session, i.e., if the same IP address does not occur within a time range of 30 minutes, the current session is closed. Subsequent requests from the same IP address will be treated as a new session.
- The time spent on a particular page is determined by the time difference between two consecutive requests.

The server log files are transformed into a set of sessions. A session represents a single visit of a user. Each session contains a session ID and a set of (PID, *time*) pairs, where

PID is the page identifier and *time* is the time the user spent on the page. There are some difficulties in accurately identifying sessions and estimating times spent on pages.

- Due to client or proxy caching of pages, the server log may not reliably detect the page requests from users. Some heuristics have been proposed, for example, in [4]. An intrusive method is to install a client-monitoring program. Generally, it is a hard problem.
- The users are identified by the IP addresses used. However this could be prone to errors since IP addresses could be reused or shared.
- The timeout technique helps to detect different users by setting a limit on idle time, although it is not always precise. It also helps to avoid endless sessions. If necessary, positive means of session identification, such as cookies or embedded sessions IDs, could be used. The amount of time a user spent on a page is determined by the time difference between two consecutive requests. This may not reflect the actual viewing time due to network congestions, transmission speed, and interruptions. Besides, the time the user spent on the last page can never be known since it is the last request of the session and there is no more requests after it.

Although the server log is not perfect for Web usage mining, it gives us rough idea about page access. Moreover, it is widely available without client-side programming or other intrusive methods. It provides a comprehensive source of access information with reasonable accuracy. For example, the Web server log will be organized into sessions as shown in Table 1.It should be noted that session IDs are not IP addresses since they may be reused or shared. Different visits from the same IP address will be identified as different sessions.

Table 1: Sessions from the server log.

Session ID	*IP Address*	*Time/Date*	*Requested Page*
1	dan.cs.umr.edu	01/Aug/1997:1 13:17:45	/~dan/a.html
		01/Aug/1997:13:17:48	/~dan/b.html
2	131.39.170.27	01/Aug/1997:13:17:47	/~white/Home.htm
		01/Aug/1997:13:17:51	/~white/hobby.htm

From Table1, it is possible to estimate how much time the user spent on each page by taking the difference in date and time between the current page request and the following page request. For example, in session 1, the user spent 3 seconds on the first page, /~dan/a.html.

5.1.3 Access Information Collections

In this step, the access statistics for the pages are collected from the sessions. The data obtained will later be used to classify the pages as well as to reorganize the site. The sessions obtained in server log preprocessing are scanned and the access statistics are computed. The statistics are stored with the graph that represents the site obtained in Web site preprocessing. The obvious problem is what should be done if a page happens to be the last page of a session. Since there is no page requested after that, we really cannot tell the time spent on the page. Therefore, we keep a count for the number of times that the page was the last page in a session. The following statistics are computed for each page.

- Number of sessions in which the page was accessed;
- Total time spent on the page;
- Number of times the page is the last requested page of a session.

5.2 Page Classification

In this phase, the pages on the Web site are classified into two categories: index pages and content pages [17]. An index page is a page used by the user for navigation of the Web site. It normally contains little information except links. A content page is a page containing information the user would be interested in. Its content offers something other than links. The classification provides clues for site reorganization. The page classification algorithm uses the following four heuristics.

- **File type-** An index page must be an HTML file, while a content page may or may not be. If a page is not an HTML file, it must be a content page. Otherwise its category has to be decided by other heuristics.
- **Number of links -**Generally, an index page has more links than a content page. A threshold is set such that the number of links in a page is compared with the threshold. A page with more links than the threshold is probably an index page. Otherwise, it is probably a content page.
- **End-of-session count -** The end-of-session count of a page is the ratio of the number of time it is the last page of a session to the total number of sessions. Most Web users browse a Web site to look for information and leave when they find it. It can be assumed that users are interested in content pages. The last page of a session is usually the content page that the user is interested in. If a page is the last page in a lot of sessions, it is probably a content page; otherwise, it is probably an index page. It is possible that a specific index page is commonly used as the exit point of a Web site. This should not cause many errors at large.
- **Reference length-** The reference length of a page is the average amount of time the users spent on the page. It is expected that the reference length of an index page is typically small while the reference length of a content page will be large. Based on this assumption, the reference length of a page can hint whether the page should be categorized as an index or content page.

5.2.1 Reference Length Method

The reference length method for page classification [3] is based on the assumption that the amount of time a user spends on a page is a function of the page category. The basic idea is to approximate the distribution of reference lengths of all pages by an exponential distribution. A cut-off point, *t*, for reference length, can be defined as follows.

$t = -ln(1-\gamma) / \lambda$

Where, γ= percentage of index page

λ= reciprocal of observed mean reference length of all pages

If a page's reference length is less than *t*, it is more likely an index page; otherwise, it is more likely a content page.

Calculated, which estimates the cut-off between index and content pages. In most cases, such a percentage is unknown and has to be estimated. For a Web site, the percentage of pages that are index pages can be estimated based on the structure and content of the site or the experience of the data analyst with related server.

5.2.2 Algorithm for Page Classification

An algorithm for page classification is introduced in this section which combines the heuristics mention above. To determine the category of a page, its file type is first checked. If it is not HTML, the page is certainly a content page and no other testing will be necessary. Otherwise, its end-of-session count, number of links, and reference length, are examined subsequently. Two thresholds, *count_threshold* and *link_threshold* are introduced. If a page's end-of-session count is greater than *count_threshold*, it is classified as a content page. If a page's number of links is greater than *link_threshold*, it is tagged as an index page. These thresholds should be selected conservatively so that they positively identify content or index pages. Finally, if necessary, the p age's reference length is checked against the cut-off point *t*. If its reference length is less than *t*, it is marked as an index page; otherwise it is marked as a content page.

The algorithm for page classification is outlined as follows.

(1) l = 1/(mean reference length of all pages)
(2) t = -ln(1-γ) /
(3) For each page *p* on the Web site
(4) If *p*'s file type is not HTML or
(5) *P's* end-of-session count > *count_threshold*
(6) Mark *p* as a content page
(7) Else If *p*'s number of links > *link_threshold*
(8) Mark *p* as an index page
(9) Else If *p*'s reference length < *t*
(10) Mark *p* as an index page
(11) Else
(12) Mark *p* as a content page

5.3 Site Reorganization

After preprocessing and page classification, we are ready to reorganize the Web site based on the access information. The goal of this phase is to reorganize the Web site such that its users will spend less time searching for the information they desire. The philosophy behind this is that a Web site will provide a better service to its users if it can cut down their navigation time by reorganizing the pages on the site. More specifically, we want to reorganize the pages so that users can access the information they desire with fewer clicks. Although other factors such as page layout affects navigation, the number of clicks a user has to go through is the dominant factor for navigation since every click requires active rather passive effort from users and often involves a request to and an reply from the server. The general idea of reorganization is to cut down the number of intermediate index pages a user has to go through. To achieve this, we need to place the frequently accessed pages higher up in the Website structure, i.e., closer to the home page, while pages that are accessed infrequently should be placed lower in the structure. In the meantime, we want to preserve the original site structure whenever possible, since it may bear business or organizational logics. Besides, dramatic changes of the site structure may confuse users. As a compromise between these two conflicting requirements, we introduce an evolutionary approach to Web site reorganization. The basic idea is to locally adjust the site when a frequently accessed page should be promoted. In addition, two thresholds are introduced, that is, maximum number of links in an index page (*I*) and maximum number of links in a content page(*C*). An index/content page will not have more than *I/C* links after site reorganization, unless it has more links before reorganization, in which case its links will be intact. These two thresholds are introduced to achieve two objectives. First is to limit the number of links in a page so its layout will be reasonable. This will prevent extreme cases, for example, a flat site structure where all pages are linked from the home page. Second is to somehow contain the changes in the site structure. The selection of these thresholds can be done by the Webmaster or data analyst.

5.3.1 Cases in Site Reorganization

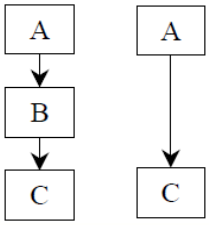
As mentioned earlier, in site reorganization, frequently accessed pages are put higher up in the site structure. On the contrary, infrequently accessed pages are placed lower in the site structure. In case such reorganization is not possible due to certain threshold such as maximum number of links in a page being exceeded, we will try to merge infrequent pages into a larger page. The mergers will reduce the number of clicks by users due to fewer page requests, thus decrease navigation time. To prevent spurious results, the merging pages must be HTML files and at most one of them can be a content page.

To decide if a page is frequently accessed, a parameter, minimum frequency (*F*), is introduced. A page's frequency is defined as the number of sessions it is in divided by the total number of sessions. If a page's frequency is greater than *F*, it is called a frequent page, otherwise, it is an infrequent page.

In site reorganization, the pages are examined sequentially starting from the home page. For each page, we consider its immediate parents and children, where a parent is any page that has a link to it and a child is any page that it has a link to. Depending on the number of children it has, there are different cases and for each case, different actions may be taken according to the frequency and category of the pages involved. For each page, we consider three cases depending upon the number of children it has: 1, 2, and 3+. The three cases are illustrated as follows. For the sake of simplicity and also since the processing is done one parent at a time, only one parent is considered in the cases.

(I) CASE I: The current page has one child. In this case, depending on the frequencies and categories of the pages, there are several possible outcomes, as shown in Figures 4 , 5, and 6, where page B is the current page.

(a) Page B is an index page.
Obviously, page B is redundant since it only serves as a link to page C. Thus the most obvious solution will be to delete page B and create a direct link from page A to page C, as shown in Figure 4.

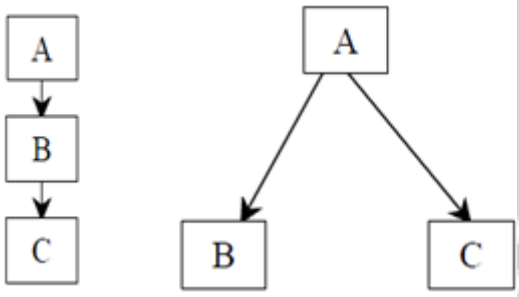


Before Processing After Processing

Figure 4: Case I, page B is an index page

(b) Page B is a content page and page C is frequent

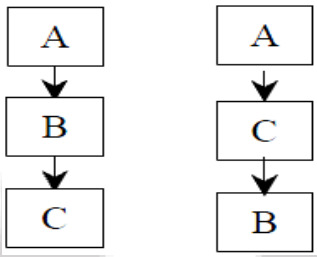
Since page C is frequent, it should be promoted by adding a direct link from page A to it as shown in Figure 5. This assumes that page A has a free link, i.e., adding a link will not exceed its number of links limit. The maximum number of links in page A is determined by its category (*I* for index page and *C* for content page).



Before Processing After Processing

Figure 5: Case I, page B is a content page and page A has a free link

If page A has used its links to full capacity, but page C have a free link, it is sometimes worthwhile to demote page B to be a child of page C as shown in Figure 6. This is done if page B is used mostly to fetch page C. This happens when the frequency of page C is more than half the frequency of page.



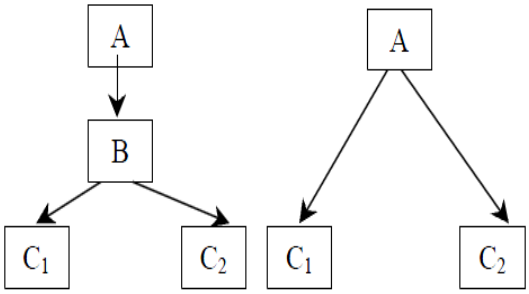
Before Processing After Processing

Figure 6: Case I, page A has no free links, but C does

(II)CASE II: The current page has two children. Again, depending on the frequencies and categories of the pages, there are several possible scenarios, as shown in Figures 7, 8, 9, 10, and 11, where page B is the current page and pages C1 and C2 are children of B. Without loss of generality, we assume that the frequency of page C1 is greater than that of page C2.

(a) Page B is an index page

Page B will be removed whenever possible. The easiest scenario is shown in Figure 7 where two direct links from page A to page C1 and page C2 are added. However, since two links will be added in page A while only one is deleted, it can only be possible if page A has an extra link to spare.

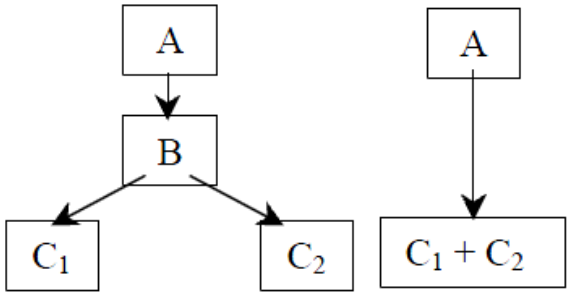


Before Processing After Processing

Figure 7: Case II, page A has a free link

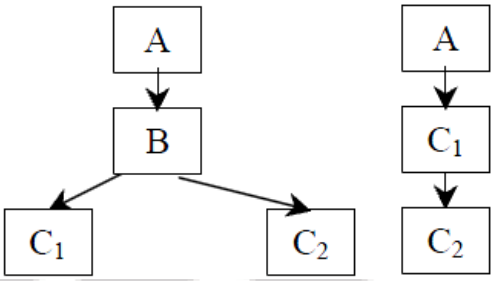
If page A does not have a free link, we will try to merge pages C1 and C2. Two or more pages can be merged if at most one of them is a content page and their total frequency does not exceed *F*. Moreover, the merged page will be a content page if a participating page is a content page; otherwise, the merged page is an index page. The limit on the number of links also applies to the merged page. If C1 and C2 can be merged, page A will link to the merged page, as shown in Figure 8.

If page A does not have a free link and pages C1 and C2 cannot be merged, page A will link to page C1 which will in turn link to page C2, as shown in Figure 9. Of course, this happens only when page C1 is frequent and has a free link.



Before Processing After Processing

Figure 8: Case II, page A does not have a free link, but C1 and C2 can be merged

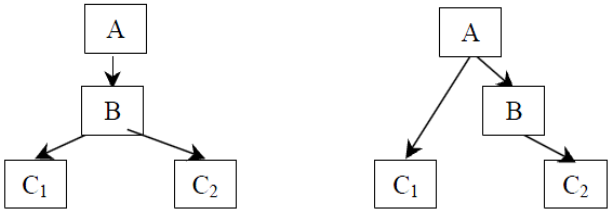


Before Processing After Processing

Figure 9: Case II, page A does not have a free link and C1 and C2 cannot be merged

(b) Page B is a content page

Since page B is a content page, it cannot be deleted. However, if C1 is a frequent page, it should be promoted higher in the structure. If page A has a free link, a link from page A to page C1 is added, and the link from page B to page C1 is removed, as shown in Figure 10.



Before Processing After Processing

Figure 10: Case II, page B is a content page and page A has a free link

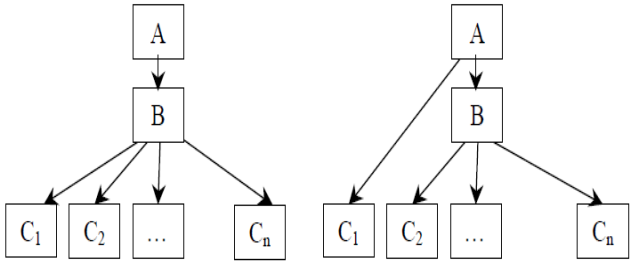
If both pages C1 and C2 are frequent, they should be promoted higher in the structure. If page A has two free links, links from page A to pages C1 and C2 are added, and the links from page B to pages C1 and C2 are removed, as shown in Figure 8. When C1 is not frequent, no change on the structure is proposed. Note in this situation, C2 will not be frequent either. If page A does not have enough links, the structure remains intact too.

Before Processing After Processing

Figure11: Case II, page B is a content page and page A has two free links

(III)CASE III: The current page has three or more children. There are several possible situations, as shown in Figures 12, 13, and 14, where B is the current page and C1… Cn are children of B. Without loss of generality, we assume that the child pages are ordered in decreasing order of frequency. That is, frequency of page C1 is greater than that of page C2 and so on until Cn. Since there are many possible combinations of the frequency and category of pages, we focus on page C1. If page C1 is a frequent page and is significantly more frequent than other children, i.e., its frequency is greater than or equal to the sum of the frequencies of C2… Cn, C1 should be promoted. Besides, we will try to merge infrequent pages.

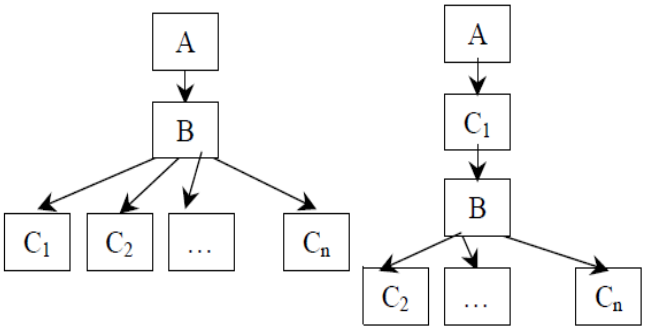
(a) Page C1 is significant and A has a free link
A link is added from page A to page C1, as shown in Figure 12.



Before Processing After Processing

Figure 12: Case III, page C1 is significant and page A has a free link

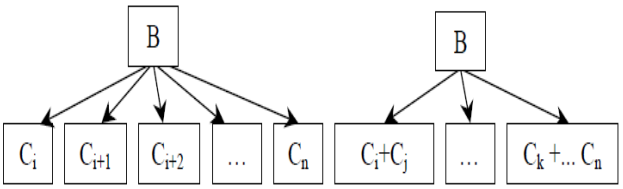
(b) Page C1 is significant and A does not have a free link, but C1 does
Since a significant number of requests is for page C1, but they have to go through page B, if page B is mostly traversed to get its children, it may be worthwhile to insert page C1 between page A and page B, as shown in Figure 13. This is done when the frequency of page C1 is more half the frequency of page B.



Before Processing After Processing

Figure 13: Case III, C1 is significant and A does not have a free link, but C1 does

(c) Merge of infrequent pages.
As explained in the beginning of this section, infrequent pages are merged if possible. Assume pages Ci, Ci+1… Cn are infrequent. They are added into a merged page in the ascending order of frequency, i.e., from Cn to Ci. When no more pages can be added into the merged page because it will become frequent, or its number of links will exceed its limit, or a second content page is being added, a new merged page starts. The remaining pages are added into the new merged page in the similar way, until all pages are done. An example is shown in Figure 14.



Before Processing After Processing

Figure 14: Merging infrequent pages

5.3.2 Algorithm for Site Reorganization

Based on the cases discussed in the previous section, the algorithm for site reorganization is outline as follows.

1) Initialize a queue Q
2) Put children of the home page in Q
3) Mark the home page
4) While Q not empty
5) Current_page = pop(Q)
6) Mark current_page
7) For each parent p of current_page
8) Local adjustment according to the cases in Section 5.1
9) Push children (maybe merged) of current_page into Q if they are not marked

6. Conclusion

In this survey, we have seen a technique that discovers the gap between Web site designers' expectations and users' behavior. The former are assessed by measuring the inter-page conceptual relevance and the latter by measuring the inter-page access co-occurrence. Web personalization is the process of customizing the content and the structure of a Web site to the specific and individual needs of each user,

without requiring from them to ask for it explicitly. This can be achieved by taking advantage of the user's navigational behaviour, as it can be revealed through the processing of the Web usage logs, as well as the user's characteristics and interests. Web transformations approaches mainly focus on developing methods to completely reorganize the link structure of a website. The approach consists of three steps: pre-processing, page classification, and site reorganization.

7. Future scope

Web personalization is a domain that has been recently gaining great momentum not only in the research area, where many research teams have addressed this problem from different perspectives. Enterprises expect that by exploiting the information hidden in their Web server logs they could discover the interactions between their Web site visitors and the products offered through their Web site. Using such information, they can optimise their site in order to increase sales and ensure customer retention. Web transformation could be further improved by incorporating additional constraints that can be identified using data mining methods. For instance, if data mining methods find that most users access the finance and sports pages together, then this information can be used to construct an additional constraint.

References

[1] T. Nakayama, H. Kato, and Y. Yamane, "Discovering the Gap between Web Site Designers' Expectations and Users' Behavior," Computer Networks, vol. 33, pp. 811-822, 2000.
[2] M. Perkowitz and O. Etzioni, "Towards Adaptive Web Sites: Conceptual Framework and Case Study," Artificial Intelligence, vol. 118, pp. 245-275, 2000.
[3] J. Lazar, User-Centered Web Development. Jones and Bartlett Publishers, 2001.
[4] Y. Yang, Y. Cao, Z. Nie, J. Zhou, and J. Wen, "Closing the Loop in Webpage Understanding," IEEE Trans. Knowledge and Data Eng., vol. 22, no. 5, pp. 639-650, May 2010.
[5] J. Hou and Y. Zhang, "Effectively Finding Relevant Web Pages from Linkage Information," IEEE Trans. Knowledge and Data Eng., vol. 15, no. 4, pp. 940-951, July/Aug. 2003.
[6] H. Kao, J. Ho, and M. Chen, "WISDOM: Web Intrapage Informative Structure Mining Based on Document Object Model," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 5, pp. 614-627, May 2005.
[7] H. Kao, S. Lin, J. Ho, and M. Chen, "Mining Web Informative Structures and Contents Based on Entropy Analysis," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 1, pp. 41-55, Jan. 2004.
[8] C. Kim and K. Shim, "TEXT: Automatic Template Extraction from Heterogeneous Web Pages," IEEE Trans. Knowledge and Data Eng., vol. 23, no. 4, pp. 612-626, Apr. 2011.
[9] M. Kilfoil et al., "Toward an Adaptive Web: The State of the Art and Science," Proc. Comm. Network and Services Research Conf., pp. 119-130, 2003.
[10] R. Gupta, A. Bagchi, and S. Sarkar, "Improving Linkage of Web Pages," INFORMS J. Computing, vol. 19, no. 1, pp. 127-136, 2007.
[11] C.C. Lin, "Optimal Web Site Reorganization Considering Information Overload and Search Depth," European J. Operational Research, vol. 173, no. 3, pp. 839-848, 2006.
[12] M. Eirinaki and M. Vazirgiannis, "Web Mining for Web Personalization,"ACM Trans. Internet Technology, vol. 3, no. 1 pp. 1-27, 2003.
[13] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, "Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization," Data Mining and Knowledge Discovery, vol. 6, no. 1, pp. 61-82, 2002.
[14] B. Mobasher, R. Cooley, and J. Srivastava, "Automatic Personalization Based on Web Usage Mining," Comm. ACM, vol. 43, no. 8, pp. 142-151, 2000.
[15] B. Mobasher, R. Cooley, and J. Srivastava, "Creating Adaptive Web Sites through Usage-Based Clustering of URLs," Proc. Workshop Knowledge and Data Eng. Exchange, 1999.
[16] W. Yan, M. Jacobsen, H. Garcia-Molina, and U. Dayal, "From User Access Patterns to Dynamic Hypertext Linking," Computer Networks and ISDN Systems, vol. 28, nos. 7-11, pp. 1007-1014, May 1996.
[17] M. Nakagawa and B. Mobasher, "A Hybrid Web Personalization Model Based on Site Connectivity," Proc. Web Knowledge Discovery Data Mining Workshop, pp. 59-70, 2003.
[18] B. Mobasher, "Data Mining for Personalization," The Adaptive Web: Methods and Strategies of Web Personalization, A. Kobsa, W. Nejdl, P. Brusilovsky, eds., vol. 4321, pp. 90-135, Springer-Verlag, 2007.
[19] C.C. Lin and L. Tseng, "Website Reorganization Using an Ant Colony System," Expert Systems with Applications, vol. 37, no. 12, pp. 7598-7605, 2010.
[20] Y. Fu, M.Y. Shih, M. Creado, and C. Ju, "Reorganizing Web Sites Based on User Access Patterns," Intelligent Systems in Accounting, Finance and Management, vol. 11, no. 1, pp. 39-53, 2002.
[21] M.D. Marsico and S. Levialdi, "Evaluating Web Sites: Exploiting User's Expectations," Int'l J. Human-Computer Studies, vol. 60, no. 3, pp. 381-416, 2004.
[22] J. Palmer, "Designing for Web Site Usability," Computer, vol. 35, no. 7, pp. 102-103, June 2002.
[23] J. Liu, S. Zhang, and J. Yang, "Characterizing Web Usage Regularities with Information Foraging Agents," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 5, pp. 566-584, May 2004.

Author Profile

Mrs. Chhaya Shejul – Student of ME Computer engineering, G.H.Raisoni College of Engineering & Mgnt, Wagholi, Pune, Pune University. India.

Mrs. B. Padmavathi – Asst. Professor in Computer Engineering Department , G.H.Raisoni College of Engineering & Mgnt, Wagholi, Pune, Pune University. India.