A Heuristic Based Approach for Improving Website Link Structure and Navigation

Haider A Ramadhan
Computer Science Department, Sultan Qaboos University, Oman
haiderr@squ.edu.om

***Abstract-* As web sites evolve over time, their complexity in both the content and the link structure also tend to increase, hence, causing disorientation and cognitive overload on the user side. User traversals tend to become more cumbersome, and information get hidden deep inside long chains of web pages and links. As a result, we need more creative design features to allow automatic and dynamic improvement of web site link structure. This paper describes a set of heuristics to optimize the web site usability and link structure. In particular, we show how these heuristics can be used to (1) simplify the user navigational needs through automatically reducing the transient links and hence providing shortcuts to pages in demand, and (2) automatically provide users with redirects to popular pages in the web site. These heuristics are hoped to compliment previously reported graph-based techniques for web site link optimization.**

***Index Terms* – Web usage, web access patterns, link optimization, experimental evaluation, web graphs.**

I. Introduction

As websites evolve over time, issues related to their growth and complexity become critical to address. If a website is considerably extensive and large, such as a website for an academic institution, such evolution in its content and complexity pose heavy navigational burden on visitors. Moreover, such growth may result in information that is nested deep inside the hierarchical structure of the website. Thus, making site surfing more difficult for the users.

It has been reported [1] that such deeply nested information tend to cause disruption, disorientation, and cognitive overload for the users. Therefore, it is becoming a major requirement for a website design to automatically and dynamically improve its link structure, and hence, get adapted to changes in the navigational patterns of the users. In this paper, we discuss a novel heuristic based approach to automatically optimize the link structure of a website. The approach uses a web graph model to track the user patterns from the server log files and recognize opportunities for link optimization.

In particular, we focus on accomplishing the following two objectives: (1) automatically short-cutting the navigational path for the users by providing them with *shortcuts* to the pages in high demand, hence skipping irrelevant transient pages, and (2) automatically redirecting the users from a page to another through *redirects*. In both cases, we discuss static and dynamic (with temporal factors) shortcuts and redirects. The impetus for the work reported in this paper came from manual observation and discovery of various navigational patterns of the students accessing pages related to courses and assignments on the website of an academic institution. From this analysis, it became evident that there is plenty of room for the link structure improvement to allow more smooth and optimized access of information sought. To lay down the path for the future implementation of a prototype system which would incorporate our heuristic based design, it was decided to observe the student patterns over one full semester. This decision helped in compiling adequate navigational activities from the server log files, which in turn assisted in coming up with the heuristic based framework to improve the link structure of the website.

The website usage was monitored using two commercial monitoring programs, namely LiveStat [2] and WebTrends [3]. Both programs track online behavior of users and use server log data to generate various statistical reports identifying page popularities. LiveStat was used to identify total visits per page and total time spent on that page. WebTrends was used to generate a tree describing various link traversals and visited paths. These information were utilized to produce the web graph to identify optimizations for improving the link structure. For detailed description of the various issues involved in analyzing the log files and generating the usability statistics needed for pattern discovery, we recommend referring to M. Eirinaki and M. Vazirgiannis [22]. As mentioned earlier, the main goal of our approach is to (1) bypass unnecessary transient pages through automatic shortcuts, (2) provide automatic redirects to popular pages, and (3) personalize some navigational processes on individual user basis.

II. Related Work

Previous work focusing on analyzing navigational patterns of the users and on improving the link structure of a website, fall under web usage mining [4], which is a subfield of data mining. Past research mainly centered around the analysis of server log files to (1) identify user access behavior [5], suggest opportunities to redesign the structure of a website [6], and develop adaptive websites [7,8]. The overall objective of these research directions is to simplify user navigational process. Various approaches have been reported for achieving the above objectives. These approaches include: visual, cognitive, statistical, survey oriented, and graph-based models.

A. Visual Model

The visual model combines a website structure, i.e. pages and links among them, with the user traversal patterns. In general, the model uses software visualization techniques, such as hyperbolic trees and discs, to represent low level data extracted from the log files into high level visual abstractions. Several systems have been developed which incorporate the above model [9,10]. However, the size of a website imposes a severe limitation on these systems. Since hyperbolic trees show all connections in the web graph, they are useful for visualizing only small websites. Large websites are normally visualized using discs, which display graphs as a directed acyclic graph (DAG), hence reducing the complexity. However, discs also reduce the accuracy of the visual displays, since only high-level aggregate information is handled. Detail information related to specific paths the users traversed tend to get lost. Hence, it may not be useful to develop a full understanding of the navigational patterns which is required to improve the link structure.

B. Cognitive Model

The cognitive model is based on cognitive walkthrough for web site applications, which is becoming popular in website usability evaluation [11,12,13]. The concept is based on simulating users performing navigations tasks on a website by assuming that users perform goal-driven explorations. The method is derived from the cognitive process theory that controls goal-driven exploration. However, the underlying model is the Comprehension-based Linked Model of Deliberate Search (CoLiDes) [11]. Here, web navigation is mainly considered to be a process of comprehending texts and images. Users select links, images or other objects found on the website according to their perception about which selection is more semantically closer to their navigation goals. The model uses latent semantic analysis as a mathematical technique to estimate the semantic relatedness of objects based on a statistical analysis of a large corpus.

C. Statistical Model

This model is based on Markov Chains, which describe, at successive times, the states of a system to identify any changes in these states. Due to their stochastic nature, Markov chains have gained wide popularity among researchers attempting to model navigational patterns and behavior of the users [14,15]. One of the main applications of Markov chains within the usability context is the prediction of future pages and paths that users may explore. This task is done through analyzing the log files to track the web pages visited by each user from the start to the end of the navigation process, and then predict future pages of interest [15], or propose a new hybrid Markov model to discover interesting navigational patterns [16].

D. Survey Model

This model is based on conducting usability investigations to identify and study usability issues using expert surveys [17,18,19]. These investigations carry out protocol oriented analysis of users using a system, and can be defined as (1) heuristic based evaluation, and (2) user testing. In heuristic evaluation a group of experts explore a website and try to find out usability problems which may affect the visitors. With user testing, a group of users are asked to use the website and then report features which they consider ineffective.

E. Graph Model

This model applies graph theory concepts and definitions in the context of the website usability [20,21]. In addition to its quantitative attractiveness, the model allows an efficient representation of the user navigation process in simple and comprehensive graphs, trees, and images. The model perfectly fits the website definition, which is a graph of nodes (pages) connected through arcs (links).

Our research uses survey and graph models. The former model was used to manually monitor the navigational behavior of users while exploring the website to come up with heuristics. The later model was used to model the website and the user navigational patterns as a weighted directed graph, and also to model the process of link structure improvement. It is believed that the proposed heuristic based approach extends the work reported on using graphs to model the website improvement process.

III. WEB GRAPHS FOR PATH ANALYSIS

Automatic improvement of the website link structure requires analysis of various user navigational patterns. This analysis will then be used to identify and optimize candidate pages for redirects and links for shortcuts. To achieve that, we approach these optimizations in terms of operations on web graphs, which represent a complete or partial website. The link structure of a website is represented by G = (N, P, L, W) as a weighted directed graph [20], where:

N = total number of pages in the website (total number of nodes in the graph)
P = set of all nodes in G, $\{P_i | i \varepsilon [1, N]\}$
L = set of all arcs (edges) in G, $\{L_{i,j} | i \neq j, i, j \varepsilon [1, N]\}$, where $L_{i,j}$ is a link from P_i to P_j.
W = set of all arc weights in G, $\{W_{i,j} | i \neq j, i, j \varepsilon [1, N]\}$. $W_{i,j}$ is the conditional probability of $L_{i,j}$ selected by users who have visited $P_{i.}$, and it is computed as follows:

$$W_{i,j} = V_{i,j} / \sum_{k=1}^{D_i} V_{idk}$$

D_i is defined as the set of all destinations of P_i. Hence, $V_{i,j}$ becomes the associate degree from P_i to P_j.

These web graphs are constructed by processing web server log files, as mentioned earlier. In the first graph used for identifying the shortcuts (shown in figure 1), we represent the frequency of link traversals by the thickness of the arcs. No time attribute is needed for this step. For the second graph used for optimizing the redirects (shown in figure 2), both total traversals of the links and time attribute are used. The time attribute of a node shows the average time spent in that node before the user leaves that node. If no links leave out of a node, the time attribute is not shown. We use operations on these graphs in general terms and base our optimizations on heuristics for (1) adding shortcut links to automatically provide quick paths to target pages with high demand, and (2) adding automatic redirects from pages visited to pages which are actually sought by the users.

IV. Optimizing the Link Structure

In general, web site designers tend to follow a top-down design during the development. With this approach, information is organized in deep hierarchies, hence no single page is overly loaded with links. As a consequence, visitors have to traverse many transient pages before arriving to the needed page. However, if some pages with very high popularity among visitors can be recognized, we should be able to automatically provide *shortcuts* and *redirects* to these target pages. This should reduce the navigational burden on the visitors, and should allow automatic link structure improvement.

Previously reported research [20] had proposed a novel approach to adding links when one is needed to connect a pair of pages. The approach is based on graph theory, and it computes the average connectivity among the pages found on a path. Using the graph definition presented in the previous section, the average connectivity (E) is defined below. In the formula, N *(N - 1) is the total number of page pairs in a website, C_{ij} is the connectivity from P_i to P_j. The larger C_{ij} is, the easier the user can find P_j from P_i. $C_{ij} = w1 + w2 + \dots + w_m$, where m is the number of link routes from P_i to P_j. The larger the E, the better is the link structure. It shows convenience degree for users to obtain information following existing hyperlinks.

$$E = \sum_{i \neq j} \frac{C_{ij}}{[N * (N-1)]}$$

The basic idea is that if there is no link from page P_i to page $P_{j,}$ then link $L_{i,j}$ should be added to include E. In this approach, links are first added to all page pairs with no link in page p, and then a new link structure G'= (P, L', W') is obtained. However, this approach does not always increase E, since the probability of a link may be decreased by adding a new link, which in turn would decrease the connectivity of other page pairs. This may happen especially when many new links are added. To solve this problem, the following extra steps would be needed: first recalculate the new connectivity C'_{ij} and then find the new average prediction E'. If $C'_{ij} > C_{ij}$ and E" > E, then adding the link $L_{i,j}$ can improve the overall link structure. With this approach, paths can be made more effective.

However, adding extra links to improve the overall connectivity may increase the time users need to find the right link on a page. It may be argued that a more simple approach would just be to add links from all pages to all pages. As stated above, adding links from all pages to all other pages does not always increase E, since the probability of a link may be decreased by adding a new link, which in turn would decrease the connectivity of other page pairs. To overcome this problem, we propose adding links from all pages on the same path to the target pages, i.e. pages of interest. Only at the end of the experimental evaluation, we would be able to find out if this approach does in fact pose extra time burden on the users.

Even with this minor improvement, the basic question on finding pages in demand, referenced in our paper as *target pages*, is still not fully addressed. Finding page pairs with large C_{ij} values may well single out paths that are heavily traversed, but would not recognize target pages for which shortcuts or redirects need to be created within the link improvement strategy. It must be pointed out though, that the original work on web graphs [] did touch on the concept of target pages, however, not in an efficient way. The following criteria was proposed to recognize target pages for the link improvement: compute the weighted average of all C_{ij} and then find the average connectivity (E). The idea here is to identify target pages and then increase their C_{ij}.

We believe this approach is useful for identifying popular paths but not target pages. Of course, a very simple way to find out target pages would be to ask the visitors or the owners of the website to point out these pages, and then find out the largest C_{ij} that connects these target pages. Pages that have large connectivity to target pages should then include shortcut links. This research argues that we need a more concise definition of the target pages. In the following discussion, we present a heuristic based definition for (1) automatically identifying target pages in a website, and (2) creating shortcuts and redirects.

A. Optimization for Shortcuts

In this research, we use the following two heuristics to define the target pages for optimizing shortcuts.

h1: target pages receive large number of hits and no links are followed out from them.
h2: target pages are visited through heavily traversed paths.

From these heuristics, we can imply that (1) target pages are terminal pages or leaf nodes in a graph, and (2) a shortcut can be placed in all the pages on a path to a target page. Figure 1 below models such situation.

To present a real life scenario, let us consider our academic environment. Suppose a faculty has a page with links to his academic activities. On that page, he/she has links to other pages, which include a page on teaching duties, which in turn contains links to all courses being taught by this faculty. As the semester progresses, the faculty adds updated information to the course pages, for example new assignments. The faculty decides to organize the page structure in a hierarchical order.

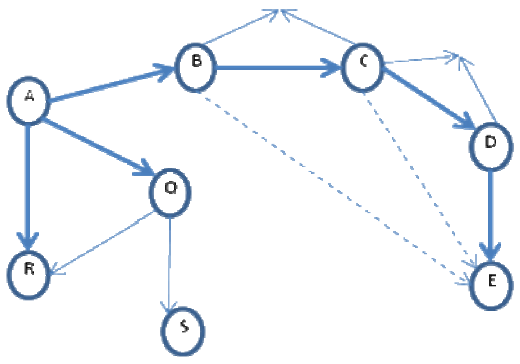

Figure 1: A web graph with shortcuts

Rather than adding course information at the top level, information is added to pages which are deeply nested. Now, users face heavy extra navigational burden in order to arrive at the needed page for the updated assignment. They have to visit the faculty main page, the teaching page, the course page, the assignments page, and finally the current assignment page. What we need here is a shortcut from the teaching page or the course page directly to the current assignment page.

In the above graph, bold arcs represent heavily visited paths, in terms of total traversals, excluding the time factor. We can observe that many users visit A, B, C, D, E to arrive at node E. Once there, users stay at node E and do not follow any link out from it. Hence, node E is a target page, and we can insert a shortcut from nodes B and C to node E, shown in dashed edges. Note that it is not required that all users follow the same path, as shown by the other edges leaving nodes B, C, and D. The main requirement for inserting a shortcut into a node is that it should be heavily visited by users on a path leading to a target page, hence becomes worth optimizing for.

The path from node A to node E does not satisfy the second heuristic *h2*. As shown in figure 1, paths A, Q and A, R are more heavily traversed than path A, B. This implies that more users leave A for Q and R than they do for node B. Hence, a shortcut from node A to node E will not be considered an optimization. Path B, C, D does satisfy the optimization heuristic h2 but node D fails to satisfy heuristic h1, i.e. it is not a target page, hence no shortcut is added from node B to node D. To prevent pages from ending up with too many shortcuts, shortcuts should be evaluated against the heuristics over time. Whenever any of the heuristics fail to satisfy, shortcuts should be cancelled.

There are situations where the temporal factor would need to be taken into account when adding shortcuts. For the above shortcut, we did not consider any temporal aspects when adding it. However, it is possible that over time a particular shortcut may become less popular. This happens if any of the above two heuristics fail to be satisfied, i.e. either a path becomes less visited or a target page becomes a non-target page by having outgoing paths. Now instead of re-analyzing the paths, we should consider only recent navigational behavior of the users, defined within a given time interval, and delete shortcuts whenever they become less popular among the users. This can be done by looking only at the most recent paths within the specified time frame.

B. Optimization for Redirects

Websites often have transient pages, which either contain links that are of interest to the users or just contain announcement information which is of no interest to the users. Considering our academic institution scenario mentioned earlier, we expect a faculty, especially a senior one, to maintain pages for several courses he/she teaches. Normally such a faculty teaches only one single course per semester. In this case, forcing the users to reach the page for this course through the main faculty page would pose a heavily extra navigational burden on the users. We can improve the user navigation here by automatically redirecting the visitors from the faculty main page or the page on teaching to the page for the course currently being taught by the faculty, hence bypassing the transient page on courses. To allow the visitors to visit the pages for other courses, the optimization should provide the visitors with a back link to the main page on teaching without being redirected. In this case, it is important that the redirect does not apply if the page whose accesses are being redirected is accessed from the target of the redirect. This would prevent navigational loops or cycles. Following are the two heuristics we propose to identify a transient page for redirects:

h1: most users visiting it leave towards a single page.
h2: most users spend very little time on it.

Figure 2 models the above scenario, where T stands for the average time in seconds spent on a node and V for the total visits. Despite its heavy traffic in terms of total visits, users spend very little time on node E. Moreover, all the visitors leave node E towards node G. Thus a redirect should be automatically created to redirect the visitors from E to G. Node B satisfies the heuristic h1 for the optimization but not heuristic h2. While node B has a single outgoing link, users spend a reasonably good amount of time in it.

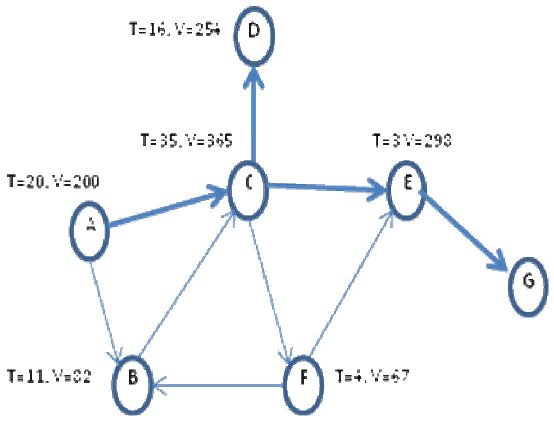

Figure 2: A web graph for redirects

This implies that users seem to be interested in the information found in node B, and hence we should not redirect them to node C. For node F, only heuristic h2 is satisfied. Despite the fact that the time spent in node F is little, the node appears to be an interesting point in the navigational patterns of the visitors. From this node they can leave for node B or node E. If we automatically redirect the visitors from node F to either node E or node B, we may be wrong 50% of the time. Hence, no redirect is created. The criteria for identifying the nodes for redirects requires searching all the paths visited by users through the web site. For each path, we need to identify nodes in which users spend on average less time than the average total time spent on the entire path, and leave these nodes for the next node on the path with frequency larger than the average frequency of visiting the nodes on the path.

When applied globally, redirects may pose a problem. Consider our faculty who currently teaches two courses, one at the undergraduate level and the other one at the graduate level. From the main page on teaching, now visitors can go either to the undergraduate course page or the graduate course page. One redirect would not serve the purpose any more in such a situation. If we still want to create a redirect, then it must be personalized for every visitor. This can be done by tracking previous navigational patterns for every user, from the server log files, to determine his/her interest. In this case, a redirect will be provided on an individual basis rather on a global basis. However, it will still remain difficult to provide a shortcut for those students who take both courses, i.e. graduate and undergraduate.

V. Experimental Evaluation

At present, we are still analyzing the web graphs to get some useful insights into the effectiveness of our heuristic based framework. In particular, our evaluation attempts to answer the following questions:

(1) How many target pages were recognized?
(2) How many shortcuts and redirects were created?
(3) Were the shortcuts and redirects considered useful by the users?
(4) Overall, how much improvement in the website link structure was achieved?

The evaluation methodology we follow focuses on having each department website as a separate website. A total of five departmental websites of the college are considered for this evaluation. The impetus behind this decision is twofold (1) to apply the heuristic based approach against five different websites instead of just one, and hence test the generality of the approach, and (2) to control the complexity of the web graphs generated from the log files. Academic websites tend to be very large, comprising hundreds of pages nested deep in hierarchical levels. Hence, it is believed that focusing on a department website would provide adequate insight into the usefulness of our approach. The questions mentioned above represent the measures against which our approach will be assessed. The following three table headers should provide an overview of what the evaluation is trying to accomplish when completed.

TABLE I
AN OVERVIEW OF DEPARTMENTAL WEBSITES

Dept	Tot. Pages	Tot. Links	Tot. Paths	Tot. Visits

The purpose of table 1 is to provide aggregate detail about the website for each department. In particular, the table shows for each department website total number of pages, total number of links found on all pages of the website, total number of complete paths that exist among the pages (nodes) of the website (graph), and the total number of visits the departmental website received. This information should provide us with a snapshot of the scope of the web graph for the departmental website.

TABLE II
USEFULNESS OF SHORTCUTS

Dept	Tot. Targets	Visits Before	Visits After

Table 2 is intended to show if the shortcuts automatically inserted were useful to user navigational patterns. The table covers the total number of target pages identified, total number of visits to the target pages through traversals of the current paths leading to these pages, and total number of visits to the same target pages following the shortcuts instead of long paths. The idea is that if the users reach the target pages, even after the shortcuts are provided, using the same path traversals of long graph links, then shortcuts may not be of interest to the users. This is to be determined by comparing the values of the visits before against the values of the visits after, i.e. directly following the shortcuts. Values for the first two columns are currently being extracted from the server log files. However, finding the values for the third column (after inserting shortcuts), requires monitoring the user behavior for another period of time, e.g. another full semester.

TABLE III
USEFULNESS OF REDIRECTS

Dept	Tot. Redirects	Visits Before	Visits After

Table 3 is intended to show if the redirects automatically inserted were useful to user navigational patterns. The table covers the total number of redirect pages identified, total number of visits to the target pages through traversals of the current paths leading to these pages, and total number of visits to the same target pages following the redirects instead of long paths. The idea is that if the users reach the target pages, even after the redirects are provided, using the same path traversals of long graph links, then redirects may not be of interest to the users. This is to be determined by comparing the values of the visits before against the values of the visits after, i.e. directly following the redirects. Values for the first two columns are currently being extracted from the server log files. However, finding the values for the third column (after inserting redirects), requires monitoring the user behavior for another period of time, e.g. an extra semester. We hope to complete our full analysis in the near future.

VI. Conclusion

This research attempts to compliment the website link improvement strategy proposed by traditional web graph theories through heuristics. The main focus of this research is to use a heuristic based framework to identify target pages (nodes) in a website (graph), and automatically create shortcuts and redirects to improve user navigational patterns and website link structure. To get some insights into the usefulness of our approach, five academic departmental websites of a college were monitored for their web usage using two commercial web monitoring packages. These packages helped us in generating web graphs and tress from the server log files. At present, we are still analyzing the results, and hope to publish our findings in the near future.

References

[1] E. Zhu. Hyper Media interface design: the effects of number of links and granularity of nodes. *Journal of Educational Multimedia and Hypermedia*, 8(3), 1999.

[2] http://www.mediahouse.com

[3] http://www.webtrends.com

[4] D. Arotaritei and S. Mitra. Web mining: a survey in the fuzzy framework. *Fuzzy Sets and Systems*, 148, 5-19, 2004.

[5] F. Michele and P. Federico. Mining interesting knowledge from weblogs: a survey. *Data and Knowledge Engineering*, 53 (3), 225-241, 2005.

[6] T. Nakayama and H. Kato. Discovering the gap between website designers' expectations and users' behavior. *International World Wide Web Conference on Computer networks, Amsterdam, Netherlands*, 811-822, 2000.

[7] P. Bra and A. Aerts. Aha!: the adaptive hypermedia architecture. *Proceedings of the ACM Hypertext Conference*, 2003.

[8] E. Ramp, P. Bra and P. Brusilovsky. High-level translation of adaptive hypermedia applications. *Proceedings of the ACM Hypertext Conference*, 2005.

[9] T. Munzner. Drawing large graphs with H3viewer and site manager. *Proceedings of the 6th International Symposium on Graph Drawing*, 384-393, London, UK, 1998.

[10] A. Wexelblat and P. Maes. Footprints: history-rich tools for information foraging. *Proceedings of SIGCHI Conference on Human Factors in Computing Systems*, 270-277, New York, USA, 1999.

[11] H. M. Blackmon, M. Kitajima and G. P. Polson. Repairing usability problems identified by the cognitive walkthrough for the web. *Proceedings of SIGCHI conference on human factors in computing systems*. Florida, USA, 497-504, 2002.

[12] H. M. Blackmon. Cognitive walkthrough for the web. *Proceedings of SIGCHI conference on human factors in computing systems*, Minnesota, USA, 463-470, 2003.

[13] A. Karoulis, S. Sylaiou and M.White. Usability evaluation of a virtual museum interface. *Informatica*,17(3), 363-380, 2006.

[14] P. Cairn, M. Jones and H. Thimbleby. Usability analysis with Markov models. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 8 (2), 99-132, 2001.

[15] M. Kitajima. Evaluation of website usability using Markov chains and latent semantic analysis. *IEICE Transactions on Communication*, E88-B (4), 1467-1475, 2005.

[16] Y. Yu. Mining interest navigation patterns based on hybrid Markov Model. *Lecture Notes in Computer Science*, 4027, 470-478, 2006.

[17] S. Y. Chen and R. D. Macredie. The assessment of usability of electronic shopping: a heuristic evaluation. *International Journal of Information Management*, 25 (6), 516-532, 2005.

[18] M. Allen. Heuristic evaluation of paper-based web pages: a simplified inspection usability methodology. *Journal of Biomedical Informatics*, 39 (4), 412-423, 2006.

[19] G. Branjik. Automatic web usability evaluation: what needs to be done? *Proceedings of the 6th conference on human factors and web*. Texas, Retrieved November 17, 2007 from: www.dimi.uniud.it/giorgio/papers/hfweb00.html.

[20] B. Zhou. Website link structure evaluation and improvement based on user visiting patterns. *Proceedings of the 12th ACM conference on hypertext and hypermedia*. Denmark, 241-244, 2001.

[21] J. Blazewicz, E. Pesch and M. Sterna. Novel representation of graph structures in web mining and data analysis. *Omega*, 33 (1), 65-71, 2005.

[22] M. Eirinaki and M. Vazirgiannis. Web Mining for Web Personalization. *ACM Transactions on Internet Technology*, 3, 1, 1-27, 2003.

Haider Ali Ramadhan is an Associate Professor of Computer Science at Sultan Qaboos University, Oman. He received his BS and MS in Computer Science from The University of North Carolina, USA, and the Ph.D. in Computer Science and AI from Sussex University, UK. His research interests include Software Visualization, Intelligent Programming Environments, Web Mining, and Alternative Programming Enviornments.