Discovery of Popular Structural Properties in a Website for Personalization and Adaptation

Haider Ramadhan Al-Lawati, Ahmed Al-Hosni, Abdullah Al-Hamadani, Mohammad Al-Badawi
Computer Science Department, Sultan Qaboos University
PO Box 36 Muscat 123, Oman
Email: {haiderr, alhosni, ahamadani, mbadawi}@squ.edu.om

***Abstract*- The massive growth in the size and complexity of websites, lead to increased demand on personalization systems and tools which can help in providing users with what they want or need without them having to ask for it explicitly. In this paper, we present a novel approach towards the discovery of target pages for shortcuts. The approach is based on the Maximal Forward Reference algorithm. Few changes to this algorithm are suggested to make it more suitable for the discovery of popular paths, pages and individual user behaviors in relation to the structural design of the sit. The major impetus for the selection of Maximal Forward Reference approach in our research was driven by two of our own convictions. First, forward traversals more realistically represent the navigational intentions of the user. Second, the algorithm has already been proven to generate a complete set of maximum references from the processed log file. The proposed approach aims at limiting the consolidation process of the MFR to the level of individual users which should help in providing more detailed site adaptation, personalization, and visualization on the user level.**

***Index Terms*- Web usage mining, user traversal patterns, site personalization, site adaptation, maximum forward traversals, web graphs.**

I. INTRODUCTION

With continuous growth in the size and complexity of websites, the issue of identifying better means for providing the users with highly demanded information with less burden is becoming too critical not to address. Due to increase in the size and content of websites, developers find it challenging to efficiently design, represent and organize the information in a website structure, while users find it difficult to access the desired information in a simple, friendly and time-saving manner. The implication is that such growth can easily result in having popular or rich content pages nested deep in the lower levels of a website, thus making it difficult for the users to find or reach.

In addition, with rapid evolution in the e-service sectors, e.g. e-commerce, e-business, e-learning, site owners are more interested than ever in making their sites automatically predict future navigational patterns of the users to improve the usability, structure, and user retention of their sites.

In general terms, the two issues of site personalization and adaptation involve three phases. First, preparation, preprocessing and categorization of web server log files to identify various data on user navigational patterns, such as frequency and time for pages visited and paths traversed [1,2,3,4]. Second, extraction of correlations and relationships between and across different kinds of such data to determine popular hot spots in a website, better classify the users and the services, formulate models for predicting future user interests, and even allow dynamic visualization of such correlations for providing high level models for the usability and popularity of various elements of the site [5,6,7]. Third, dynamically recommend to the site maintainers and developers actions to improve the structural and navigational efficiency of the site and to better provide the users with popular services [5,6,7].

Various commercial software solutions can be used to extract these correlations among data from the log files. These include LiveStat [10], WebTrend [33], and AnaLog [34]. The first solution is used to generate information on the popularity of each page in the site, e.g. total visits and total time spent. AnaLog and WebTrend are used to generate an abstract web site graph describing the site structure in terms of pages and links and user traversal patterns. However, statistical reports generated by these log file analysis software provide only data numbers on the frequency of page visits and time spent, and fail to provide information on how different structural properties of the site are related or can be adapted [7,8]. Hence, when the task is to (1) discover popular hot spots such as candidate pages, within the context of the overall site, for navigational shortcuts or redirects, (2) determine popular site components which should be redesigned or restructured, or (3) predict future user interests, then more elaborating techniques are needed. The objective of this research is to report a novel approach towards the discovery of target pages for shortcuts. The approach is based on the Maximum Forward Reference algorithm reported in [6]. Few changes to this algorithm are suggested to make it more suitable for the discovery of popular paths, pages and individual user behaviors in relation to the structural design of the site.

The next section summarizes important related work reported on determining popular structural elements of a website. Section 3 formulates the problem, summarizes the behavior of the Maximum Forward Reference algorithm, and suggests a modified version of the algorithm to cater for the discovery of shortcut pages. Section 4 explains in more detail the modified discovery process. Section 5 outlines a plan for the implementation

doi:10.4304/jetwi.3.3.253-260

and experimental analysis of the procedure reported in this paper.

II. RELATED WORK

Work in the area of web site personalization and adaptation has been well documented and reported. Research on the analysis of log files, discovery of useful user navigational patterns, recommendations for website adaptation, and visualization of the popular site elements through web graphs has received considerable attention. Prior work mainly focuses on the analysis of the server log files, and can be classified by approaches applied to such analysis. Main approaches include software visualization models [11,12], cognitive models [13,14,15], statistical models such as Markov models [16,17,18], survey models [19,20], client-side navigational assistance models [23], and algorithmic models [21,22]. Since the work reported here is related to the algorithmic approach, the following discussion briefly outlines distinctive splits among models used in such approach.

Almost all the models depend on the analysis of user access patterns through the application of graph-based theory within the context of website usability [5,8,21,22,24]. Web graphs are used to represent the site structure and user sessions. Analysis of such graphs is then conducted to discover (1) popular elements of the site, (2) opportunities for adaptation and personalization, (3) related clusters and classifications of users and pages for better efficient marketing, improved services and user retention, and (4) structural site components which require better design and organization.

The work reported in [6] on the concept of Maximal Forward Reference (MFR) for characterizing user episodes for the mining of traversal patterns is regarded a pioneer effort in this area. The objective is to locate “hot” access pattern in an information-providing service. This work is based on statistically dominant paths, and a maximal forward reference is defined as the sequence of pages requested by a user up to the last page before backtracking. This approach is similar to that of finding large item sets for association rules. The main difference is that in MFR the reference sequence describing user traversal patterns has to be consecutive references, whereas in the association rules it is just any combinations of items in a transaction. Our research proposes a modified version of the MFR algorithm to discover target pages which can serve as candidate pages for the shortcuts to be placed in pages on the path leading to the candidate page, and hence allow personalization and adaptation.

The idea of uncertainty in Web usage mining to discover clusters of user session profiles using robust fuzzy algorithms was reported in [25]. In this approach, a user or a page can be assigned to more than one cluster. After preprocessing the log data, a dissimilarity matrix is created and that used by the fuzzy algorithms in order to cluster typical user sessions. To achieve this, a similarity measure is used to takes into account both the individual URLs in a Web session, as well as the structure of the site.

The work reported in [3], regards Web usage mining as a three-phase process, consisting of preprocessing, pattern discovery, and pattern analysis. The prototype system introduced, WebSIFT, first performs intelligent cleansing and preprocessing for identifying users, server sessions, and inferring cached page references through the use of the referrer field, and also performs content and structure preprocessing. Pattern discovery is accomplished through the use of general statistic algorithms and data mining techniques such as association rules, sequential pattern analysis, clustering, and classification. The results are then analyzed through a simple knowledge query mechanism, a visualization tool, or the information filter, that makes use of the preprocessed content and structure information to automatically filter the results of the knowledge discovery algorithms.

In a similar work [26], data mining techniques, such as association rules and sequential pattern discovery, are applied to Web log files to customize the server hypertext organization dynamically. Here, the Web usage mining is regarded as a two-phase process, consisting of the preprocessing phase where all irrelevant data are removed and log file entries are clustered based on time considerations, and the Web mining phase where data mining techniques are applied. A generator of dynamic links uses the rules generated from sequential patterns or association rules, and each time the navigation pattern of a visitor matches a rule, the hypertext organization is dynamically modified.

On the information discovery side, [27] used a knowledge discovery process in order to discover marketing intelligence from Web data by proposing an environment that combines existing online analytical mining, as well as Web usage mining approaches and incorporates marketing expertise; [28] have designed a sequence mining system for the specification, discovery, and visualization of interesting navigation patterns, called “trails” and used concept hierarchies along with site semantics as the basic method of grouping Web pages together, where accessed pages or paths are abstracted based on page content, or by the kind of service requested.

Regarding self-adaptive web sites, [29] proposed a framework for self-adaptive Web sites, taking into account the site structure except for the site usage; [2] defined the notion of adaptive Web sites as sites that semiautomatically improve their organization and presentation by learning from visitor access patterns; [3] proposed a framework for mining Web log files to discover knowledge for the provision of recommendations to current users based on their browsing similarities to previous users. The framework relies solely on anonymous usage data provided by logs and the hypertext structure of a site. After data gathering and preprocessing (converting the usage, content, and structure information contained in the various data sources into various data abstractions), data mining

techniques such as association rules, sequential pattern discovery, clustering, and classification are applied, in order to discover interesting usage patterns.

In our previously reported work [30] along with the work reported by [31] on the discovery of popular pages as candidates for shortcuts, several optimization and heuristics based techniques were suggested. Both approaches utilized the web graph and searched for specific connectivity features in the web graph to identify the shortcuts. Although being novel, both approaches, optimization and heuristics based, suffered one major shortcoming, namely being rigid in the criteria for a page to satisfy to be a candidate for a shortcut. For example, in our previous work mentioned above, two heuristics were used to define the target pages for the shortcuts. First, the page must receive large number of visits and no links should follow out from it. Second, target pages should be reached through heavily traversed paths.

The formal experimental analysis of the above heuristic based approach is yet to be completed. However, the manual observational analysis of the web graph for the synthetic site used in the experiment tend to point out some degree of rigidity in the second part of the first heuristic as well as in the second heuristic. This observational analysis to be shortly published clearly indicates that by requiring target pages for the shortcuts to be the leaf nodes on the heavily visited paths exclude large number of popular pages with high visit frequency which are not terminal nodes in the web graph, i.e. links were following out from them. In regard to the second heuristic, the observational analysis shows the existence of several pages with high visit frequency that can be reached through several lightly traversed paths. Hence, the analysis provided us with useful insights into the usefulness of the approach, but also indicated the need for the heuristics to be refined.

III. PROBLEM FORMULATION

As mentioned earlier, when the web sites are evolved, their complexity tends to increase, the need for site adaptation and personalization arises, and the requirement for an efficient site maintenance and restructuring becomes more pressing. The merit of the work reported in this paper is to present a modified version of the Maximum Forward Reference (MFR) algorithm [6] for the discovery of target pages with high popularity to provide the users with shortcut links to these pages. The main goal is twofold (1) reduce the navigational burden on the users, and (2) support a decent visualization environment to facilitate adaptation, personalization, and structural redesign of a web site.

In the MFR algorithm, visit sequences for each user are regarded as one single traversal of the path until a backtracking is encountered. Hence, only consecutive forward references are recorded as a traversal of a path. Backward references are mainly regarded as transient actions as opposed to navigational interests. On a backtracking, the forward reference path is terminated. The resulted path is termed a maximum forward reference.

As an illustrative example, suppose the traversal log contains the following path for a user {A, B, C, D, C, B, E, G, H, G, W, A, O, U, O, V}, as shown in the simple web graph of figure 1. The MFR algorithm will produce the following set of forward references for this user {ABCD, ABEGH, ABEGW, AOU, AOV}. Duplicate traversals of the same paths are pruned from the set. However, for the discovery of the target pages, the visit frequency of the duplicate paths needs to be added to the overall frequency count of the path, otherwise it will be difficult to keep track of popular paths with high visits needed to determine target pages for the shortcuts. For example, if a user visits the path ABCD four times, the path in the MFR set will be shown only once, thus will not reflect how popular that path is. Therefore, the frequencies should be added to the visit counter for the path.

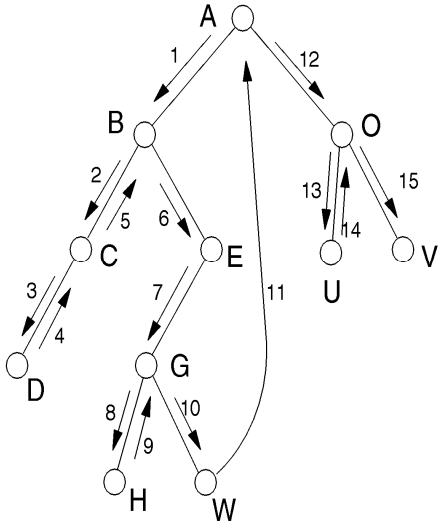

Figure 1. Maximum Forward References

Next, the MFR algorithm processes sets of all the users and produces an overall set of non-intersecting sects by pruning all partial paths found in the sets. This step may be very useful in pointing out some very popular structural elements of the site, but it further masks out information needed to determine popularity of individual paths for the shortcuts. Since the final superset consolidates all subsequences into the containing sequences, and clustering or classifying pages or users tend to become a challenging task as well. To adequately detail such shortcomings, let us consider the following MFR sets for three hypothetical users navigating the same web graph shown in figure 1.

U1: {ABCD, ABEGH, ABEGW, AOU, AOV}
U2: {ABC, ABEG, ABE, AO}
U3: {AB, ABEGH, AOU}

Next, re-occurring partial sequences among all the sets are pruned by intersecting MFRs for each user and then

for all the users. The result is a consolidated set representing structural elements of the site which are considered popular, but at the expense of information about visit frequency of individual paths, pages, and users, hence making it difficult to figure out target pages on target paths for target users. For example, the MFR set of user U2 is {ABC, ABEG, ABE, AO}. Since reference ABE is fully contained in ABEG, ABE is pruned, and the resulting set would be {ABC, ABEG, AO}. Before pruning, page E had two distinct visits, but after the pruning it shows only one visit. Hence, it is important to modify the MFR algorithm to take into account the frequency of all individual visits for the partial paths and to retain user patterns. The consolidated sets for the above three users would like the following:

U1: {ABCD, ABEGH, ABEGW, AOU, AOV}
U2: {ABC, ABEG, AO}
U3: {ABEGH, AOU}

Finally, the MFR algorithm prunes all the partial paths among all the sets and generates one super set. The resulting set would be {ABCD, ABEGH, ABEGW, AOU, AOV}. As shown, the process pruned ABC, ABEG, AO for U2 and ABEGH, AOU for U3. In fact, both MFR sets for U2 and U3 were eliminated. As can be seen from the process, the algorithm does point out to the overall popularity of the substructure ABEG, for example, but it masks out the information needed to determine the visit frequency of ABEG in relation to the number of different users visited the path. It is possible in some cases that a page or a path receives large number of hits from only a few users. In this case, it may not satisfy the criteria for being a target page.

The original MFR sets showed that page G, for example, was visited four times and by three different users. The consolidated set shows that it was visited twice. Number of users visited page G is also not shown. Hence, if the criteria for the discovery of target pages for the shortcuts, for example, was set to page depth on the path to 3, visit frequency to 4, and number of users to 3, then page G would not be recommended for the shortcut since the last two conditions of the criteria are not met by the final set, which in fact is not true.

The major impetus for the selection of maximum forward traversal approach in our research was driven by two convictions. First, forward traversals more realistically represent the navigational intentions of the user whereas the backward traversals in general tend to represent travelling intentions. Second, the algorithm has already been proven to generate a complete set of maximum references from the processed log file and consolidate such references in one final super reference for the discovery of some general hot spots in the web site.

From the above process we can conclude that while the final superset could be of high importance in targeting popular structural elements of the site, it would help less in tasks related to detailed site adaptation, personalization, visualization. For such tasks, initial MFR sets are more suitable. The consolidation process should be limited to the level of individual users only. In other words, pruning of partial paths should be implemented within a single MFR set and not across all the MFR sets for all the users. Visit frequencies of the pruned paths to be added to their parent paths. The main use of the superset would to be provide information regarding lengths of various paths. Such information would be used to identify the depth of target pages. Finally, visit frequencies for both the paths and pages from all the sets should be reflected in the web graph. This modified process should prune redundant subsequences for a user and still provide information for the discovery of popular paths and pages for the shortcuts in addition to determining navigational patterns of the individual users. The above procedure can be summarized as follows:

1. Generate the MFR sets for all the users
2. Consolidate partial paths for each user by pruning subsequences at a set level
3. Reflect the visit frequencies of the pruned paths in their parent paths
4. Reflect visit frequencies for all the users in the web site graph

Following this procedure, the resulting web graph would look as follows:

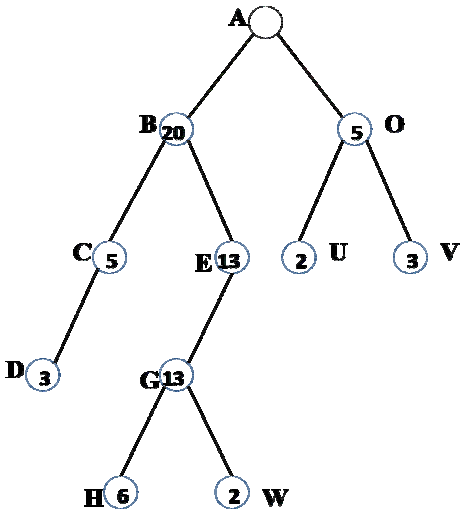

Figure 2: Total visit frequencies

Since all the visits are assumed to start from the initial page of the web graph, i.e. A, the visit frequency for a page also indicates the path popularity up to that page. For example, in the above graph it can be concluded that page G had 13 visits and the traversal frequency of the path ABEG is at least 13.

IV. DISCOVERY OF TARGET PAGES

In this section we detail the modified procedure for the discovery of candidate pages for the shortcuts. The following steps capture the overall algorithmic behavior

of the procedure. It is worth noting that we yet to implement the algorithm for the experimental purposes. Thus, the following steps describe the functional characteristics of the modified approach.

Step 1: Process the log file for data preparation, cleaning, and session identification using widely accepted conventions reported in [4].

Step 2: Generate a traversal log containing for each visited link a pair (s, d), where s denotes the initial start page and d denotes the destination page on the visit path. We intend to use the procedure reported in [32] to generate this log which is called ***referer log***.

Step 3: From the referer log, generate the MFR sets for individual users using the MFR algorithm described in [6].

Step 4: Consolidate partial paths for individual users with their parent paths in each MFR set, while adding path frequencies of the pruned subsequences to their super sequences. The resulting MFR set for each user should have no duplicate paths. However, the overall visit counters for the paths and pages would still retain information on the navigational patterns of the user. For example, if the MFR set for a user is {ABC, ABC, ABC, ABC, ABCD}, then after pruning, the set would contain only {ABCD}, but the frequency counters would show 5 visits for pages B and C and hence 5 traversals for the path ABC and one traversal for the path ABCD. As a result, navigational patters for individual users are clearly retained. If, however, the consolidation process is further applied to all the MFR sets for all the users, then the popularity of pages and paths would still be maintained but the behavior of individual users would not be easily inferred.

One of the criteria for the shortcuts as reported in [29, 30] is the number of users interested in a page and not only number of visits to that page. It is possible to discover a page with high visit frequency, but with interest from a few visitors. For example, let us consider the following two MFR sets for U1 {ABC, ABC, ABC, ABC, ABCD} and U2 {ABE, AOU}, using our example web graph. After the initial pruning process, we would have U1 {ABCD} and U2 {ABE, AOU}. With the second consolidation process, the resulting super MFR set would be {ABCD, ABE, AOU}. From the visit counters, we can still find out the overall popularity of page C, but it would be difficult to associate these visits with either of these two users. We know that C was mainly visited by user U1, but from the final MFR set it is not easy to determine that.

Step 5: Determine the supporting thresholds (ST) for identifying the target pages for shortcuts. This step is intended to provide site designers with means to analyze various navigational patters of users in close relation with the structural elements of the site. When used in a visual environment, such patterns should greatly assist the designers to experiment with various support thresholds for the redesign and personalization purposes. Following is the proposed set of such thresholds to be used for the discovery of shortcuts. It is worth noting that these thresholds are derived from our extensive academic and professional experience with the design and adaptation of web applications. Once the formal experimental analysis is completed, we should be in a better position to reflect on these support thresholds.

T1 (p_n):	visits (p_n) >= vsupport (p_n)
T2 (p_n):	users (p_n) >= usupport (p_n)
T3 (p_n):	depth (p_n) >= dsupport (p_n)

Threshold T1 determines the support for the total visit frequency of the target page on a popular path. This threshold covers both the popularity of the page and the path. For example, setting the visit frequency for a target page to 10 implies that all page on the path to the target page would have a visit frequency >= 10. Threshold T2 determines the support for the number of different users visited the page, and T3 determines the support for the depth of the page on a visited path or in the web graph. To give an illustrative example on the application of these thresholds, let us consider a situation where we need to find target pages at *dsupport*=3. The process is as follows:

a) From all the pruned MFR sets, determine the set of paths with depth >= *dsupport*, call it the Candidacy Set (CS).
b) Since users are expected to traverse similar paths, CS is expected to include duplicate paths or subpaths. Hence, we need to perform exactly what the original MFR algorithm does, namely to generate the MFS superset, call it the Candidacy Superset, which consolidates all the MFR sets for all the users into one non-intersecting set where all the subsequences are pruned. However, to retain the information about different users visiting similar paths, we need to maintain a counter for each duplicate path or subpath found in different MFR sets. This will help us in determining the *usupport* threshold when looking for the target pages.
c) From Candidacy Superset, identify the set of pages with depth >= *dsupport*, call it the Target Set (TS).
d) For each page in TS, test the values for the remaining thresholds.
e) Determine the set of pages whose thresholds conditions are satisfied, call it Shortcut Set (SS).

To show a complete example of the procedure described in this paper, let us consider the following initial MFR sets for five hypothetical users:

U1:{AB,AB,ABC,ABCD,ABEG,ABEGH, BEGH}
U2:{AOV, AOU, ABCD, ABC, ABEG, ABEG}
U3:{ABEGH, ABEGH, ABEGW, ABEGH}
U4:{AOU, AOV, AOV, ABCD, ABEG}
U5:{ABEGH, ABEG, ABEGW}

The consolidated MFR sets would look as follows:

U1: {ABCD, ABEGH}
U2: {AOV, AOU, ABCD, ABEG}

U3: {ABEGH, ABEGW}
U4: {AOU, AOV, ABCD, ABEG}
U5: {ABEGH, ABEGW}

The superset would include {ABCD, ABEGH, ABEGW, AOV, AOU}. The resulting web graph would look like the one shown in figure 1:

When *dsupport* is >=3, for example, then candidacy superset would contain {ABCD, ABEGH, ABEGW}. The path ABCD shows only once in the superset, but from Step 2 outlined above, we now that this path is visited by three different users, namely U1, U2 and U4. The target set would contain {D, G, H, W}, showing pages with depth >= 3. Now assuming that the remaining thresholds are: *vsupport* >= 10, and *usupport* >= 4, then the final shortcut set would contain {G}, indicating that G is a target page for a shortcut. Following previously reported convention on the location of the shortcut on a heavily visited path [30,31], a shortcut link to G would be placed on page B, i.e. all the pages on the path from A to G, excluding the initial root node A and the parent node E, which naturally should have a direct link to G.

The reason for excluding the root node has to do with the nature of the tree structure. In a tree, a node can be duplicated and hence can be reached through two different paths. For example, in the tree of figure 3, it is possible to have page G visited from page U but through an unpopular path AOUG, hence not satisfying the criteria for being a target page on that path. However, if the criteria dealing with the path popularity is relaxed, then a short cut for G should also be placed on the initial page A.

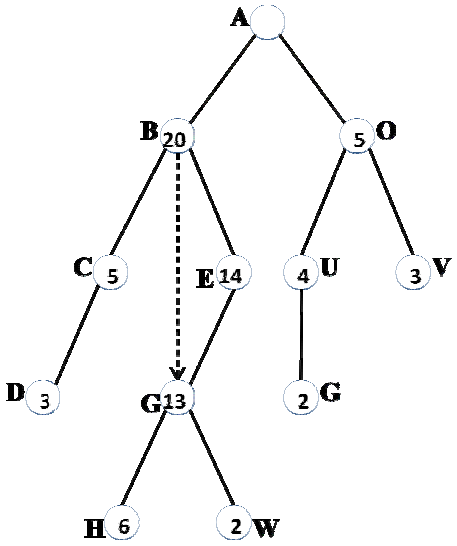

Figure 3: A shortcut to a target page

V. PRELIMINARY EXPERIMENTATION

To assess the impact of the modified MFR procedure proposed in this paper, we managed to conduct a preliminary experimental evaluation on an actual website. The website was originally developed for a different research project. In that project, the goal was to discover target pages using heuristics based approach [30], which basically aimed at optimizing the overall link structure of a website. However, the experimental results [37] showed that the heuristics introduced a level of rigidity in discovering target pages. As a consequence, it was decided to focus on a more robust and general procedure to accomplish the discovery task, which is reported in this paper. Since these two approaches are significantly different in their underlying algorithmic procedure but still are related in the objective, it was decided to use the same server log file for the above website usage. In fact, it would constitute a useful future study for comparing the results of both approaches for the discovery of target pages, i.e. heuristics based and the modified MFR approaches.

The website was designed to closely imitate the actual website of an academic department. Academic websites tend to be very large, comprising tens of pages nested deep in hierarchical levels. Hence, it was believed that focusing on an academic department website would provide an adequate insight into the usefulness of any proposed optimization or adaptation procedures. The website consisted of 200 pages, 5 levels of depth, and a total of 320 links across all the pages. The breakdown of the pages for each level showed 10 pages at level 1 (depth 1), 18 at level 2, 35 at level 3, 58 at level 4, and 79 pages at level 5, totalling up to 200 pages. The traversals were generated by asking 40 students to use the site over a semester. Howewver, only visits covering 3 months were collected from the log file. This was done to avoid any criticism regarding using the same log file for two different experiments. Pages did not include any particular rich-content, apart from links to other pages organized in a way to provide a decently solid hierarchical structure.

The experiment closely follwed the steps presented in section 3 of this paper. For the cleaning of server log file and identifying user sessions, we implemented the widely accepted convention reported in [4]. The referer log was generated using the procedure reported in [32]. The processed data identified a total of 1830 log records and 323 different sessions for all the users. The MFR sets for individual users were generated using the Maximum Forward Reference (MFR) algorithm [6]. Next, the modified approach outlined in section 3 was applied to consolidate the MFR sets through prunning and to identify page and path popularities of the website graph. The MFR procedure identified a total of 462 initial paths (sets) for the traversals, containing both partial and complete paths. The modified approach identified a total of 267 consolidated paths, along with their popularity in terms of frequency of visits.

Only three main performance measures were used to analyze the results, namely, the depth threshold (dsupport), the frequency of visits (vsupport), and the number of target pages discovered. Since users tend to arrive to pages at shallow levels in the website without extra navigational burden, it was decided to identify target pages at depths 3 to 5. For the visit threshold, pages with

visit frequency less than 10 were excluded from being considered as candidates for the target pages. The justification is that in our judgment a popularity of less than 10 in a large website does not indicate a heavily visited path or page.

Table 1: Discovery of target pages at Depth = 3

Visit Frequency Support	Target Pages Found
vsupport >= 10	27
vsupport >= 20	12
vsupport >= 30	9

From table 1, we can see that when dsupport is set to 3 a total of 27 pages were identified to have a visit frequency of 10 and more. For a visit frequency of 20 and more, 12 pages were identified, and for a visit frequency of 30 and more 9 pages were identified. Since pages with visit frequency of 10 and above already include in its count pages with vsupport >= 20 and vsupport >= 30, it can be concluded that when dsupport = 3, a total of 27 pages are identified to be target pages for the shortcuts. This figure represents 77% of the pages at level 3 of the web tree. Table 2 shows the results for dsupport=4. As mentioned erlier, there were a total of 58 pages found at level 4 in the web tree of the site. With a total of 36 pages identified as target pages, which include pages for all 3 values of vsupport, 62% of the pages at level 4 are found to be target pages. This outcome is no surprise since during navigation, users surfing academic sites tend to visit pages at higher and middle levels more frequently while trying to reach pages at lower levels in the website. Table 3 shows that as the depth increases, number of target pages tends to decrease. When dsupport=5, out of 79 pages only 20 (25%) pages were identified for the shortcuts for all three vsupport values. Overall, across 200 pages making up the test website, a total of 83 pages were identified for the shortcuts, representing 41% of the web tree.

Table 2: Discovery of target pages at Depth = 4

Visit Frequency Support	Target Pages Found
vsupport >= 10	36
vsupport >= 20	16
vsupport >= 30	9

Table 3: Discovery of target pages at Depth = 5

Visit Frequency Support	Target Pages Found
vsupport >= 10	20
vsupport >= 20	8
vsupport >= 30	4

VI. CONCLUSION AND FUTURE WORK

In this research, we have introduced a modified version of the MFR algorithm which identifies target pages for the shortcuts within the context of the overall website. The main objective of the research is twofold: (1) allow site developers to efficiently optimize the design and structure of their websites, and (2) reduce the navigational burden on the users. The preliminary evaluation of the modified procedure has shown to be very useful in identifying popular spots in a website in close relation to the overall structure of the site.

To better generalize the initial findings reported in this paper, we plan to conduct few more simulated experiments in the near future. The effort would involve simulating various website structures by using different distribution algorithms [36], namely the uniform, normal and triangular distributions with different parameters. We will generate various websites with different number of web pages and different links from each page. Various navigational patterns of the users will be simulated using the above three distribution algorithms with different parameters. This step will provide us with a decent referer log needed to run the modified MFR procedure explained in this paper for the discovery of shortcuts in relation to various support thresholds and structural properties of the simulated site.

Since visualization is among the core requirements of our research, the planned system should allow the site designers and the experimenters to visually analyze various visit scenarios against different support thresholds. This feature should enable them to graphically represent a website and to determine various popular structural spots for better personalization and adaptation. We plan to implement the system using Java platform. The processed log file, the referer log, and the MFR sets will be stored in a relational DBMS that can support the communication through JDBC/ODBC driver. Visualization features are to be provided by VGJ library, including the visual display of the website graph, which is to be described using the GML format.

REFERENCES

[1] M. D. Mulvenna. Personalization on the net using web mining. *Commun. ACM*, 43, 123-125, 8, 2000.

[2] B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on web usage mining. *Commun. ACM*, 43, 142-151, 8, 2000.

[3] J. Srivastava, R. Cooley, and M. Deshpande. Web usage mining: Discovery and applications of usage patters from web data, *SIGKDD Explorations* 1, 2, 12-23, 2000.

[4] R. Cooley. *Web usage mining: Discovery and Application of Interesting Patterns from Web Data*. PhD thesis, U. of Minnesota, 2000.

[5] M. Eirinaki. Web mining for web personalization, ACM Transactions on Internet Technology, 3, 1, 1-27, 2003.

[6] M. Chen. Efficient Data Mining for Path Traversal Patterns, *IEEE Transactions on Knowledge Engineering*, 10, 2, 209-221, 1998.

[7] Y. R. Srikant. Mining web logs to improve web site organization, *Proc. WWW01*, 430-437, 2001.

[8] A. P. D. Bra. Aha! The adaptive hypermedia architecture. *Proceedings of the ACM Hypertext Conference*, 2003.

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

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

[11] A. Wexelblat and P. Maes. Footprints: history-rich tools for information foraging. *Proceedings of SIGCHI conference on human factor in computing systems*, pp 270-277, NY, USA, 1999.

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

[13] 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.

[14] 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.

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

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

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

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

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

[20] 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.

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

[22] www.dimi.uniud.it/giorgio/papers/hfweb00.html.

[23] 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.

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

[25] A. Joshi. On mining web access logs. *Proc. ACM SIGMOD*, 63-69, 2000.

[26] F. Masseglia. Web usage mining: How to efficiently manage new transactions and new customers. Proc. PKDD, France, 2000.

[27] A. Buchner and M. D. Mulvenna. Discovering internet marketing intelligence through online analytical web usage mining. SIGMOD, 4, 54-61, 1998.

[28] M. Spiliopoulou. Web usage mining for web site evaluation. *Commun. ACM* 43, 8, 127-134, 2000.

[29] M. Perkowitz and O. Etzioni. Adaptive web sites. *Commun. ACM* 43, 8, 152-158, 2000.

[30] H. Ramadhan. A Heuristic Based Approach for Improving Website Link Structure and Navigation. *Journal of Emerging Technologies in Web Intelligence*, Vol. 1, 1, 88-93, 2009

[31] C. Doerr and D. von Dincklage. Simplifying web traversals by recognizing behavior patterns. Proc. HT'07, UK, 2007.

[32] R. T. Berners-Lee. Hypertext transfer protocol-HTTP/1.0, *Internet Draft*, 2, 1996.

[33] http//www.webtrends.com

[34] http//www.analog.cx

[35] Ramadhan, H. Identification of Target Pages for Shortcuts in a Website: An Experimental Analysis, IEEE Conference on Data Engineering and Internet Technology (DEIT), Bali, Indonesia, to appear, March, 2011.

[36] A. M. Law, W. D. Kelton, "*Simulation Modeling and Analysis*", McGraw Hill, 1991.

Call for Papers and Special Issues

Aims and Scope

Journal of Emerging Technologies in Web Intelligence (JETWI, ISSN 1798-0461) is a peer reviewed and indexed international journal, aims at gathering the latest advances of various topics in web intelligence and reporting how organizations can gain competitive advantages by applying the different emergent techniques in the real-world scenarios. Papers and studies which couple the intelligence techniques and theories with specific web technology problems are mainly targeted. Survey and tutorial articles that emphasize the research and application of web intelligence in a particular domain are also welcomed. These areas include, but are not limited to, the following:

- Web 3.0
- Enterprise Mashup
- Ambient Intelligence (AmI)
- Situational Applications
- Emerging Web-based Systems
- Ambient Awareness
- Ambient and Ubiquitous Learning
- Ambient Assisted Living
- Telepresence
- Lifelong Integrated Learning
- Smart Environments
- Web 2.0 and Social intelligence
- Context Aware Ubiquitous Computing
- Intelligent Brokers and Mediators
- Web Mining and Farming
- Wisdom Web
- Web Security
- Web Information Filtering and Access Control Models
- Web Services and Semantic Web
- Human-Web Interaction
- Web Technologies and Protocols
- Web Agents and Agent-based Systems
- Agent Self-organization, Learning, and Adaptation
- Agent-based Knowledge Discovery
- Agent-mediated Markets
- Knowledge Grid and Grid intelligence
- Knowledge Management, Networks, and Communities
- Agent Infrastructure and Architecture
- Agent-mediated Markets
- Cooperative Problem Solving
- Distributed Intelligence and Emergent Behavior
- Information Ecology
- Mediators and Middlewares
- Granular Computing for the Web
- Ontology Engineering
- Personalization Techniques
- Semantic Web
- Web based Support Systems
- Web based Information Retrieval Support Systems
- Web Services, Services Discovery & Composition
- Ubiquitous Imaging and Multimedia
- Wearable, Wireless and Mobile e-interfacing
- E-Applications
- Cloud Computing
- Web-Oriented Architectrues

Special Issue Guidelines

Special issues feature specifically aimed and targeted topics of interest contributed by authors responding to a particular Call for Papers or by invitation, edited by guest editor(s). We encourage you to submit proposals for creating special issues in areas that are of interest to the Journal. Preference will be given to proposals that cover some unique aspect of the technology and ones that include subjects that are timely and useful to the readers of the Journal. A Special Issue is typically made of 10 to 15 papers, with each paper 8 to 12 pages of length.

The following information should be included as part of the proposal:

- Proposed title for the Special Issue
- Description of the topic area to be focused upon and justification
- Review process for the selection and rejection of papers.
- Name, contact, position, affiliation, and biography of the Guest Editor(s)
- List of potential reviewers
- Potential authors to the issue
- Tentative time-table for the call for papers and reviews

If a proposal is accepted, the guest editor will be responsible for:

- Preparing the “Call for Papers” to be included on the Journal’s Web site.
- Distribution of the Call for Papers broadly to various mailing lists and sites.
- Getting submissions, arranging review process, making decisions, and carrying out all correspondence with the authors. Authors should be informed the Instructions for Authors.
- Providing us the completed and approved final versions of the papers formatted in the Journal’s style, together with all authors’ contact information.
- Writing a one- or two-page introductory editorial to be published in the Special Issue.

Special Issue for a Conference/Workshop

A special issue for a Conference/Workshop is usually released in association with the committee members of the Conference/Workshop like general chairs and/or program chairs who are appointed as the Guest Editors of the Special Issue. Special Issue for a Conference/Workshop is typically made of 10 to 15 papers, with each paper 8 to 12 pages of length.

Guest Editors are involved in the following steps in guest-editing a Special Issue based on a Conference/Workshop:

- Selecting a Title for the Special Issue, e.g. “Special Issue: Selected Best Papers of XYZ Conference”.
- Sending us a formal “Letter of Intent” for the Special Issue.
- Creating a “Call for Papers” for the Special Issue, posting it on the conference web site, and publicizing it to the conference attendees. Information about the Journal and Academy Publisher can be included in the Call for Papers.
- Establishing criteria for paper selection/rejections. The papers can be nominated based on multiple criteria, e.g. rank in review process plus the evaluation from the Session Chairs and the feedback from the Conference attendees.
- Selecting and inviting submissions, arranging review process, making decisions, and carrying out all correspondence with the authors. Authors should be informed the Author Instructions. Usually, the Proceedings manuscripts should be expanded and enhanced.
- Providing us the completed and approved final versions of the papers formatted in the Journal’s style, together with all authors’ contact information.
- Writing a one- or two-page introductory editorial to be published in the Special Issue.

More information is available on the web site at http://www.academypublisher.com/jetwi/.