Journal of Web Engineering, Vol. 8, No.1 (2009) 071-092

FINDING UNEXPECTED NAVIGATION BEHAVIOUR IN CLICKSTREAM DATA FOR WEBSITE DESIGN IMPROVEMENT

I-HSIEN TING

Department of Information Management

National University of Kaohsiung
700 Kaohsiung University Road, 811 Kaohsiung City, Taiwan
iting@nuk.edu.tw

CHRIS KIMBLE

Professeur Associé Management Information Systems

Euromed Marseille Ecole de Management

Domaine de Luminy, BP 921, 13288, Marseille Cedex 9, France
chris.kimble@euromed-management.com

DANIEL KUDENKO

The University of York
Heslington, York, YO10 5DD, United Kingdom
kudenko@cs.york.ac.uk

Received June 3, 2008
Revised July 29, 2008

This paper describes a novel web usage mining approach to discover patterns in the navigation of websites known as Unexpected Navigation Behaviours (UNBs). The approach provides a web designer with a means of identifying and classifying patterns of browsing and, by reviewing these patterns, the designer can then choose to modify the design of their site or redesign it completely. UNB mining is based on the Consecutive Common Subsequence (CCS), a special instance of Common Subsequence (CS), which is used to define a set of expected routes. The predefined expected routes are then treated as rules and stored in a rule base. By using the predefined route and the UNB mining algorithm, interesting navigation behaviours can be discovered. This paper will introduce the format of the expected route and describe the UNB algorithms. It will also describe a tool that a website designer can use to define the expected route more efficiently, which can help the website designer to make decision about where and how the design of website can be improved. The paper concludes with a series of experiments designed to evaluate how well the UNB mining algorithms work and demonstrate how UNB mining can be useful for improving website design.

Keywords: Navigation behaviour, web usage mining, clickstream data, sequential mining, website design, website designers
Communicated by: M. Gaedke & G. Rossi

1 Introduction

Rapid growth has made the World Wide Web a place full not only of opportunity but also of competition; unsurprisingly, articles on the topic managing a successful website are now a very popular [20]. The interface of a website confronts the website users directly, and its quality, usability and accessibility may affect the user's impressions of the quality of the website. In E-commerce, these may also affect customers' feelings about the transaction reliability of the website. Thus, website design is a very important criterion in building a successful website [19].

Finding and solving design problems in a website is the key to better performance and more successful website. Among the possible reasons that may cause the design problems of a website, Fang and Holsapple have shown that the website site navigation structure is a critical factor to affect the usability of a website [8]; Yen proposed several structure-based models to measure the accessibility of a website [30]. Therefore, one way to discover potential design problems is by developing an understanding of the navigation behaviour and experience of users when navigating a website [13].

Currently, there are many website usability research methodologies which focus on understanding the browsing behaviour of users, such as questionnaire survey, interview, ethnographic study, the "Think-Aloud" method [22], client-side behaviour monitor [24] and server-side data analysis [5]. While not a perfect as a source of data [4], the analysis service-side data is often seen as the most convenient and cheapest way to discover the browsing behaviour of users. Log files (clickstream data) are a source of server-side data that records users' requests to the web server; by using this kind of data for analysis, the website does not have the extra cost of installing special software on the client/server side in order to record users' browsing behaviour [12].

The amount of data (especially the server side clickstream data) in a website's log can sometimes be huge. Thus, an efficient tool to help analyse the browsing behaviour of users is necessary. Therefore, in this paper we focus on how to use data analysis techniques with clickstream data in order to identify the browsing patterns of users and discover potential design problems.

Web usage mining is now the most popular technique for analysing a user's navigation behaviour in clickstream data. The results can not only be used to understand how a user navigates their way through a site, but also to provide a better service [15], an adaptive website [21] and website personalisation [16][18]. It is therefore of value to be able to analyse navigation behaviour and apply the results for improving the design of website [10].

In this paper, we propose a novel web usage mining method called Unexpected Navigation Behaviour mining, or UNB mining. UNB mining is useful for website designers to understand how a user browses their website, especially for those website designers who want to redesign their website. The concept behind this method of web usage mining is that the designer of the site ought to be able to define patterns of 'expected navigation behaviour', and then by using this as a template, is able to discover any unexpected deviations from these routes.

The rationale behind this is that the designer of the site is the person who best understands the overall design concept of the site and so is best placed to define an expected route through it. Using this predefined expected route and the proposed UNB mining algorithms, navigation routes that do not match the expected route are then identified as instances of unexpected navigation behaviour. The

website designer can then use these to find the reason why this behaviour occurs, decide to accept the new routes as valid or reject them as errors, make decisions of change or modify the website accordingly. We consider UNB mining to be a form of data mining, since it discovers regularities (or actually "irregularities") in the clickstream data. The search space for these regularities is restricted by additional knowledge, in our case the expected navigation route.

The structure of this paper is as follows: a brief overview of the background and motivation is introduced in section 1. In section 2, some related literatures concerning the discovery of user's navigation behaviours and the concept of common sequence are discussed as well as the motivation of the UNB mining. The detail of the UNB mining algorithm is presented in section 3. In section 4, we introduce the UNB mining system and a tool to help produce a definition for an expected route. An empirical study about using the UNB mining for supporting the improvement of website is presented in section 5, and the performance of UNB mining is evaluated in this section as well. Section 6 presents the conclusions of this paper and outlines some directions for future work.

2 Literature Review and the Motivation

2.1 Discovering User's Navigation behaviours

In navigation behaviour discovery, there are two popular approaches to finding an interesting pattern. The first is to use a visualisation technique to present the user's navigation history in a graph or map. For example, Canter et al. [3] proposed a way to group different user's navigation behaviour into six indices that could characterise navigational behaviour [4]. The footstep map is another popular visualization tool that uses the concept of a spanning tree to convert the user's navigation route into a footstep map [7]. The Footstep graph is a visualisation technique that improves the footstep map by using an x-y plot to present the user's navigation pattern based on navigation sequence and navigation time. The user's navigation trend can be very easily extracted using the footstep graph [27]. The advantage of these visualisation tools is that the results are very easy to be read and can be understood by the human eye. However, the weakness of this kind of technique is that it is not robust enough to deal with a large amount of complex clickstream data [4].

The second technique is to use a data mining technique to analyse clickstream data; this is also known as web usage mining [6]. Applying the traditional mining algorithms, interesting navigation patterns can be discovered [25][29]. For example, a clustering algorithm can group the users into suitable clusters according to their navigation behaviour for measuring the similarity [17]. An association rule algorithm can discover the relationship between different user's navigation routes and trend analysis or sequential mining algorithms can be used to discover a sequential pattern in user's navigation behaviour [11]. Similarly, the HPG algorithm can be used to model user's navigation behaviour and find a user's preferred trail [3]. The advantage of this kind of navigation behaviour discovery technique is that a large amount of data can be processed very efficiently [16]. However, the results produced by these techniques are sometimes difficult to interpret and explain.

Nasraoui et al. proposed an approach to map user sessions in an evolving clickstream scenario through measuring the similarity (using single-pass mining method) to an expected (pre-discovered) user session profile [17]. In this paper, we propose another way to define the expected navigation routes through collecting website designer's 'design concept' for the site. Then, using this, an

algorithm will compare the difference between expected navigation routes and user's navigation behaviour in order to discover unexpected navigation behaviour.

Our aim here is to improve web site design, thus, even though the web usage mining algorithm is very efficient, it is of little practical use if the patterns it discovers are difficult to explain. In the case of website design, we believe that website designer is the person with most knowledge about the way the website should be browsed; consequently, we have based our technique on navigation routes that are defined by the web site designer. This approach is described in further detail in section 2.3; the problem of efficiently and effectively 'harvesting' the routes from the designer is covered in section 4.1

2.2 Sequential Mining, Common Subsequence (CS) and Longest Common Subsequence (LCS)

Sequential mining is a technique that can be used to discover patterns in time or sequence [9]. In recent years, it has been broadly used in the area of bioinformatics research to discover the pattern of the DNA sequence or in web usage mining to discover the pattern of users' navigation sequences. In web usage mining, the sequential pattern would be one web page browsed after another, or one set of web pages browsed after another set. For instance, a sequential pattern could be expressed as "30 % users' navigation behaviour follows the sequential pattern web page A, web page B, then web page C"[2].

In this paper, we propose a sequential mining algorithm called UNB mining. UNB mining is based on the concept of a consecutive common subsequence (CCS), which is special instance of a common subsequence (CS). CS is a well-designed sequential mining algorithm. Given that there are two sequences X and Y, then if Z is a subsequence of both X and Y, we say that Z is a common subsequence of both X and Y.

For example, if X={a,b,c,d,e,f,g} and Y={b,d,e,h,i} then the common subsequence of sequence X will be CS={b},{d},{e},{b,d},{d,e},{b,e},{b,d,e}. Generally, the intersections between X and Y are found by using the dynamic programming algorithm [1]. One of the most interesting applications of the CS is the identification of the subsequence with maximum-length. This is known as the longest common subsequence (LCS) or edit-distance. In above example, the LCS of sequence X and Y is LCS={b,d,e}.

In web usage mining, the CS and LCS are usually applied to clustering and sequential mining. The CS and LCS are also very useful in sequential mining to discover the relevance, co-occurrence and difference between sequences [14]. Finally, in some clickstream clustering algorithms, the LCS is use as the core technique for an algorithm to measure the distance between different sessions' sequences.

2.3 Our Approach to UNB Mining

In most examples of web usage mining research, the researcher uses a web mining algorithm to find an interesting pattern and the pattern is then 'explained' by the researcher using some characteristic of the pattern. However, when dealing with websites, identical patterns can have different meanings in different websites or even different meanings in the same website when considers at different time or viewed in a different context.

For example, in our previous research, a visualisation technique called footstep graph was developed to model the user's navigation behaviour and was used to identify interesting patterns [19]. The footstep graph is based on the use of a 2-dimensional x-y plot, where the x-axis is the browsing time between two web pages and the y-axis represents the web page in the browsing route of users. Thus, the distance on the x-axis means the time that the user has spent and a change in the y-axis is a transition from one web page to another.

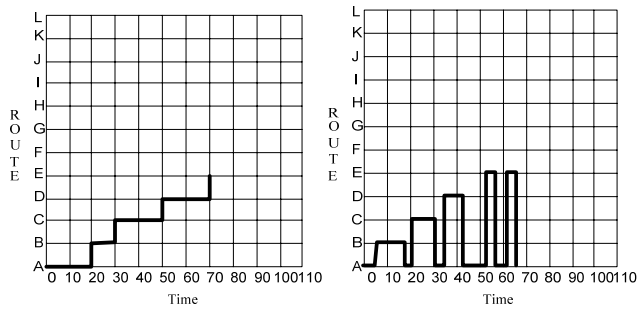

Figure 1 (a) The user's navigation pattern that presented as Upstairs pattern in footstep graph (b) The user's navigation pattern that presented as Fingers pattern in footstep graph

In figure 1(a), the footstep graph shows what we termed an Upstairs pattern. In some e-commerce websites, this kind of navigation pattern means the user is surfing the website smoothly. However, in another site the user may not be following the path the designer intended. Another example, shown in figure 1(b), is when the user's navigation behaviour presents a Fingers pattern. In some cases, this means the user has fallen into a navigation loop. Normally, this kind of navigation pattern will indicate that there is some problem in the website and a redesign of the website is necessary. However, this navigation pattern may be exactly what the website designer intended. Finally, a pattern such as a Fingers pattern can sometimes be a transitory pattern that simply indicates that a new user is exploring an unknown site [4].

In addition to the analysis of patterns, some website designers may want to have a detailed view of navigation path to discover some hidden problems in the navigation paths of users (or they want to understand more about how their website is used). The detail view of patterns sometimes is useful for the website designer to understand how their website is used. For example, there are two users' navigation path P1:{Page A→Page B→Page C→Page D} and P2:{Page A→Page B→Page E→Page D}, these two patterns will be discovered as stairs pattern and the distance of the co-occurrence Page A→Page D are both 4. However, some website designer may think the expected navigation route in their website is path P1 and path P2 is not what the website designer expected. Thus, a tool to assist the discovery of the unexpected navigation paths is benefit for website design to know how their website is used.

It would be helpful to develop a technique for more efficient analysis of the navigation behaviour of users, based on the viewpoint of website designers, the tool is called Unexpected Navigation

Behaviour Mining (UNB mining) in the paper. For such a technique, unexpected navigation routes can be discovered, and the results can directly help website designers to understand how websites are used by users.

In the paper, we take the view that website designers have their own rationale for how their websites are designed and what navigation paths they expect users to take. However, the website design concept is often embedded deep within website designer's mind and it can be a very difficult and time consuming task to transform this design concept into a specification suitable for UNB mining. To ease this problem, we created a technique for defining a so-called "expected route" (see section 4.2.4.(3)). Through this, the tacit and unstructured knowledge about the expected navigation behaviour in website designer's mind can be transformed into a set of explicit and structured rules (expected routes). These rules are also the base for automation, if it is necessary for further research in this area.

3 UNB Mining

Figure 2 shows the UNB mining process and its two prerequisites steps: data pre-processing and pattern restoration. UNB mining itself also consists of two steps: route segmentation and unexpected navigation behaviour discovery. A database of predefined expected routes is an essential component of UNB mining. The detail of these four steps and the components in the process is discussed in more detail below.

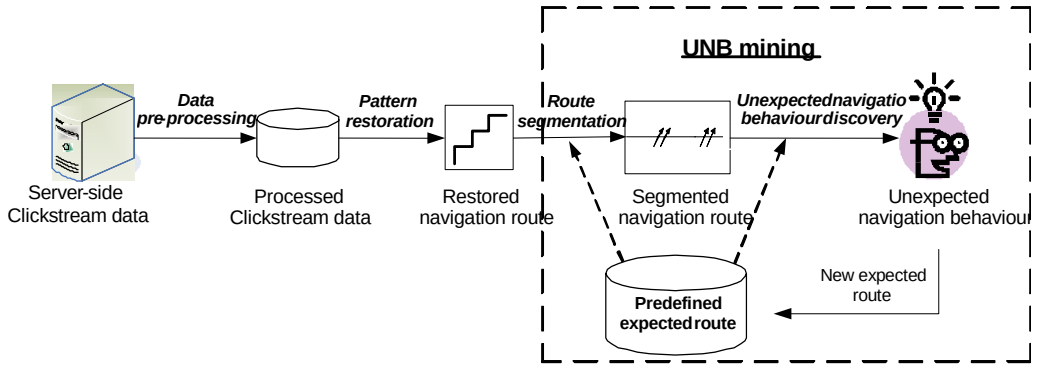

Figure 2 The process of the UNB mining and prerequisites steps

3.1 Data Pre-processing and Data Restoration

The raw server-side clickstream data must be pre-processed to clean the noise, incomplete or irrelevant data before using it for web usage mining. For example, clickstream data created by 'bots' needs to be removed to ensure the data is really from a user' [26]. User and session identification are also important steps of the data pre-processing.

1. 82.41.230.33 [09/Dec/2004:11:27:11] "GET /index.asp HTTP/1.1" http://www.google.com/search?q=skateboard+store 200 12121 Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
2. 82.41.230.33 [09/Dec/2004:11:27:13] "GET /images/portfolio/websites/nevisport3.jpg HTTP/1.1" http://www.google.com/search?q=skateboard+store 200 57888 Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
3. 82.41.230.33 [09/Dec/2004:12:25:09] "GET /microdesign/contact.asp HTTP/1.1" 200 19878 http://www.ch-6.co.uk/microdesign/websites_list.asp, Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
4. 82.41.222.96 [09/Dec/2004:12:17:19] "GET /dev/impact/login/applications_list.asp HTTP/1.1" 200 14712 http://www.ch-6.co.uk/dev/impact/login/admin.asp, Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.5) Gecko/20041107 Firefox/1.0
5. 82.41.230.33, [09/Dec/2004:12:25:22] "GET /microdesign/contact.asp HTTP/1.1" http://www.ch-6.co.uk/microdesign/contact.asp, Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)

Figure 3 A Sample of Raw Clickstream Data

User identification is in order to distinguish the clickstream data from different users. Generally, this is achieved by using the user's IP address, login / username or cookies to identify the user. Session identification then divides the user's navigation history into a number of distinct sessions. This can be a problem, when there is no login / logout data and is usually done by defining a time-out threshold (e.g. 30 minutes) to mark the end of a session. Figure 3 shows an example of a real raw clickstream data and figure 4 is a sample clickstream data after data pre-processing (including data cleaning, user identification and session identification)

User 82.41.230.33
Session 1:
82.41.230.33 09/Dec/2004:11:27:11 /index.asp http://www.google.com/search?q=skateboard+store, Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Session 2:
82.41.230.33 09/Dec/2004:12:25:09 /microdesign/contact.asp http://www.ch-6.co.uk/microdesign/websites_list.asp Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
82.41.230.33 09/Dec/2004:12:25:22 /microdesign/contact.asp http://www.ch-6.co.uk/microdesign/contact.asp, Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)

Figure 4 A Sample Pre-Processed Clickstream data: After Session Identification

An additional problem faced in data pre-processing, is that some data is lost due to caching in either the browser or a proxy server. This lost data must be restored to make sure the user's navigation pattern is as correct and complete as possible [28]. Some of the potential and actual problems of current pattern restoration techniques have been discussed in Clark et al. [4]. In figure 5, the sample

shows that some clickstream data is lost due to caching in browser, and figure 6 shows the restored clickstream data by using the pattern restoration techniques.

(1) 61.59.121.221, 16:02:54, /~kimble/research/research.html, /~kimble/
(2) 61.59.121.221, 16:03:00, /~kimble/teaching/teach.html, /~kimble/

Figure 5 An example of original clickstream data in a session

(1) 61.59.121.221, 16:02:49, /~kimble/, -, Restored
(2) 61.59.121.221, 16:02:54, /~kimble/research/research.html, /~kimble/, Original
(3) 61.59.121.221, 16:02:57, /~kimble, -, Restored
(4) 61.59.121.221, 16:03:00, /~kimble/teaching/teach.html, /~kimble/, Original

Figure 6 The clickstream data after first phase of the PRM algorithm

3.2 Predefined Expected Route Database

The predefined expected route-base is used to store the predefined expected route. For UNB mining, the expected route-base is the core of the entire technique; it is an essential component for both steps of UNB mining. The definition of the expected route is based on the concept of the consecutive common subsequence (CCS). The details of the concept of CCS and the expected route definition method are discussed below. In this paper, we also provide a useful tool to assist the definition of expected routes. The tool is introduced in section 4.1 of this paper.

3.2.1 Consecutive Common Subsequence (CCS)

In some cases, common subsequence (CS) and longest common subsequence (LCS) are useful to measure the similarity between sequences. However, in sequential mining for web usage mining, the user's navigation behaviour is a consecutive behaviour. The navigation behaviour will be very different if the navigation sequences between two are not identical. For example, consider the two navigation sequences A={a,b,c,a,d} and B={a,b,c,d}. These two sequences all match the LCS={a,b,c,d} and both are very similar, but in terms of their navigation behaviour, they are quite different. In order to discover a UNB, a concept based on a CCS is proposed in this paper.

As discussed in section 2 of this paper, a CCS is a special instance of a CS. Assuming there are two sequences X and Y then, for X={a,b,c,d,e,f,g} and Y={b,d,e,h,i}, the CS of the sequence X and Y will be {b},{d},{e},{b,d},{d,e}{b,e}{b,d,e}.

The CCS is the concurrence nodes in two different sequences are the same and consecutive. For example, the CCS in above two sequences X and Y will be {d→e}. Sometimes the CCS can be divided into many sub-CCS when necessary. For instance, there is a CCS A={a→b→c→d→e}, then it can be divided into A1={a→b}→{c→d→e} or A2={a→b→c}→{d→e} or A3={a}→{b→c}→{d→e}…etc.

3.2.2 The Expected Route

In this paper, the concept of a CCS is used to define the expected route. The expected route is predefined, usually by the website designer or the person who has overall responsibility for site content (e.g. marketing manager, website owner, or website manager).

There are two different kinds of subsequence of the expected route, restricted subsequence and flexible subsequence and it route must follow the concept of CCS. For example, consider a predefined expected route ER = $R_1\{x_1 \rightarrow x_2\} \rightarrow F_1\{<=p; \in a\} \rightarrow R_2\{x_3 \rightarrow x_4\}$. $R_1\{x_1 \rightarrow x_2\}$ which is a restricted subsequence. The subsequence of the user's navigation behaviour must be exactly the same as the restricted, in order not to be identified as a UNB. $F_1\{<=p; \in a\}$ on the other hand, is a flexible subsequence (pattern route). The p in F_1 represents a certain number of pages and the a in F_1 represents the attributes of those pages. As discussed in section 3, time can also be represented in the flexible subsequence as t, however, this will not be discussed in this paper.

The idea behind using a flexible subsequence is that the website designer may sometimes think several possible navigation routes all of which have similar pattern. The flexible route can allow the website designer to define only few expected routes to cover many expected routes with similar concept (navigation pattern). For example if we have three routes

Expected route 1 = {index→produt$_1$→cart→checkout}

Expected route 2 = {index→product$_2$→cart→checkout}

Expected route 3 = {index→product$_1$→product$_2$→cart→checkout}

Then, by applying the concept of flexible subsequence, the website designer would only need to define one expected route to cover all the three.

Expected route 4 = R_1{index}→F_1{<=2; ∈ product}→R_2{cart→checkout}

In above expected route 4, the flexible route F_1 means that if a subsequence contains no more than 2 product related web pages then the route is what the website designer expected. So, the expected route 4 can cover the requirements of expected route 1, 2 and 3. For example, the product 1 and product 2 in the expected route 3 are both product related page and there are only two product related pages in the subsequence. The advantage of using the concept of flexible subsequence in expected route definition is that it not only makes the definition of expected route more flexible but it also reduces the time and effort needed by the website designer to define an expected route.

3.3 Navigation Route Segmentation

After the data pre-processing and data restoration, the next step is data segmentation. In this step, the user's navigation route is broken into segments based on the pre-defined expected route. In addition, the segmentation algorithm also carries out some preliminary UNB detection. The pseudo code of the segmentation algorithm is presented in figure 7. To illustrate the whole process, consider the following expected route ER:

```
1: Input ERi←Expected route i
2: Input URj←User Navigation Route j
3: Output Segmented URj
4: Begin procedure Segmentation algorithm:
5: pointer=0
6: for k=0 to Number of R
7:     for m=0 to Number of Nodes in Rk
8:         for n=pointer to Number of Nodes in URj
9:             If all Rkm=URjn then
10:                SpName=Rk
11:                    pointer=CurrentNode
12:            else
13:                    URj→Unexpected route; end algorithm
14:            end if
15:        end for
16:    end for
17: end for
18: End procedure
```

Figure 7 The pseudo code of the segmentation algorithm

ER=R_1{index→product_index}→F_1{<=p_1; ∈ a_1}→R_2{cart→checkout}

And the following user navigation route UR:

UR=index→product_index→$product_1$→$product_2$→service→cart→checkout

Using the above notation, the user's navigation route can now be segmented into the following three sub-sequences:

UR=R_1{index→product_index}→F_1{$product_1$→$product_2$→service}→R_2{cart→checkout}

At the beginning, every user's navigation route is put into a pool of UNBs, and the UNB mining algorithm only processes the navigation routes in this pool. Once the navigation route has been put into a pool of expected navigation routes, the route will not be processed any more.

First, the segmentation algorithm will extract the first restricted subsequence (R_1) from an ER. Then, the algorithm will search every consecutive subsequence in a UR until every restricted subsequence (R) in the ER is matched. If not, the algorithm will put the UR into the UNB pool and the algorithm will not be performed again. The time-complexity of the segmentation algorithm is O(n3).

When there is a consecutive subsequence that matches a restricted subsequence, the algorithm will give the name R_k to the consecutive subsequence (SpName) that means it is an appropriate subsequence to the restricted subsequence R_k. The algorithm then will extract next restricted subsequence of the ER, and the algorithm will be performed again. Once all of the restricted subsequences in the ER are matched and the UR is still alive, the UR will be put into the expected navigation behaviour pool. After performing the segmentation algorithm, all consecutive subsequences of the ER that have not been recognized as a restricted subsequence will be treated as a flexible subsequence in next step.

3.4 Unexpected Navigation behaviour Discovery

The main work of the UNB discovery step is to test is a flexible subsequence in a UR matches the setting of an appropriate flexible subsequence in an ER. In this step, the UNB discovery approach will only process the UR in the expected navigation route pool. Assuming an ER:

ER=R_1{index→product_index}→F_1{<=p_1; ∈a_1}→R_2{cart→checkout}

And a segmented UR:

UR=R_1{index→product_index}→F_1{$product_1$→$product_2$→service}→R_2{cart→checkout}

The UNB discovery approach will test is every restricted rule (R_1 and R_2) of the ER exists in the segmented UR. To make sure it is necessary to perform the UNB discovery approach. Then, the approach will test is the F_1 in the segmented UR matches the threshold of the F_1 in the ER. For example, considering there are three different threshold settings of the F_1 in the ER.

(1) When p_1=2 and a_1=any in the F_1{<=p_1; ∈a_1} then:

In the F_1 of the ER, a_1=any means any page in the F_1 of the UR is expected. Under this situation, the UR will be recognised as an UNB that is verified by the predefined ER. The reason why is even the F_1 of the UR passed the threshold of a=any, but it cannot pass the threshold of p_1=2.

(2) When p_1=3 and a_1=any in the F_1{<=p_1; ∈a_1} then

Under this situation, the UR will be recognised as an expected navigation behaviour. The F_1 of the UR not only passed the threshold of a_1=any, but also passed the threshold of p_1=3.

(3) When p_1=3 and a_1=product in the F_1{<=p_1; ∈a_1} then

In the F_1 of the ER, a=product means only the product related page in the F_1 of the UR is expected. Under this situation, the UR will be recognised as an unexpected navigation behaviour. The reason why is even the F1 of the UR passed the threshold of p_1=3, but one of user browsed page is related to service. The UR therefore cannot pass the threshold a_1=product, and be recognized as an UNB.

Figure 8 is the pseudo code of the UNB discovery approach. After performing the UNB discovery approach, a UR that is still alive in the UNB pool will be recognised as a UNB, because there are not any ER support it to be an expected navigation behaviour. Once the UNB has been discovered, a website designer can by this to review their website, and to check is there any potential problem in the website. If the website designer thinks that the discovered UNB is acceptable to be an expected route, it can then be defined as an expected route and stored in the expected route-base. The time-complexity of the UNB discovery algorithm is $O(n)$.

1: Input ER_i←Expected route I
2: Input UR_j←Segmented User Navigation Route $_j$
3: Output matching result

4: Begin procedure UNB discovery approach:
5: ***for*** k=0 to number of $SubIR_i$ and $SubUR_j$
6: ***if*** $SubIR_{ik} \neq SubUR_{jk}$ ***then***
7: UR_j→unexpected route; end approach
8: ***end if***
9: ***end for***
10: ***for*** m=0 to number of F in ER_i, n=0 to number of F in UR_j
11: ***if*** $F_mp \geq F_np$ and $F_na \in F_ma$ ***then***
12: UR_j→expected navigation behaviour
13: ***else***
14: UR_j→unexpected navigation behaviour
15: ***end if***
16: ***end for***
17: End procedure

Figure 8 The pseudo code of the UNB discovery approach

3.5 A Measurement of the Most Similar ER

In UNB mining, there is a metric to measure the distance between discovered unexpected navigation routes and a set of pre-defined expected routes. This information enables the website designer to know the most similar expected routes of a UNB. If necessary, it can also provide a full list of the expected routes and the distance between each expected route and the UNB. The website designer can then use this information to find the most similar expected route and to think about why the UNBs happen.

The distance between UNBs and expected routes is measured by the Edit Distance algorithm. Edit Distance is a means of measuring sequence similarity. This measures the minimum number of simple string edit operations required to transform one string into another. The string edit operation of the edit distance includes insert, delete, replace and transpose [23]. For example, there is a UNB and an expected route ER5:

The UNB:/→/clients→/clients→/contact.asp

The ER5: /→services.asp→{<=2, ∈ Any}→/aboutus.asp→{4,Information}→/contact.asp.

The /clients in the UNB is an information related web page, therefore, the UNB and ER5 can now be transformed to

The UNB: /→/contact.asp

The ER5: /→services.asp→aboutus.asp→contact.asp

Therefore, the edit distance between the UNB and ER5 is 2 (insert services.asp and aboutus.asp to the UNB).

Figure 9 is figure that illustrates above distance measurement concept. Furthermore, this figure is also helpful for website designer to review the relationship between an unexpected navigation behaviour and expected route. For example, why did the user move directly from "/" to "/clients.asp" but did not move via /services.asp, and why did the user move directly from "/clients.asp" to "/contact.asp" but did not move via "/aboutus.asp".

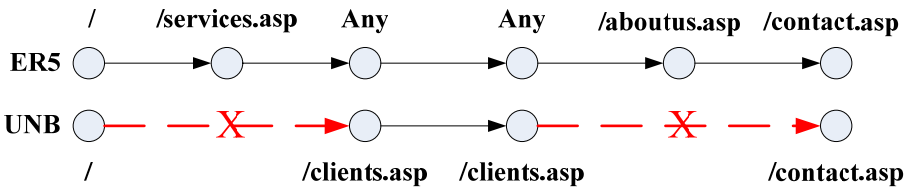

Figure 9 The concept of the edit distance measurement

4 The UNB Mining System as a whole

4.1 A Tool to Assist Designers with the Definition of an Expected Route

One argument that might be made about the UNB mining approach is that it could be a time-consuming and heavy work for website designer to define expected routes, especially for large and complex websites. In order to assist website designer to define the expected routes, we developed an expected routes definition assistance tool. By using this tool, the website designer can simulate a user's navigation behaviour and produce expected routes from their original website design concept in a very efficient way.

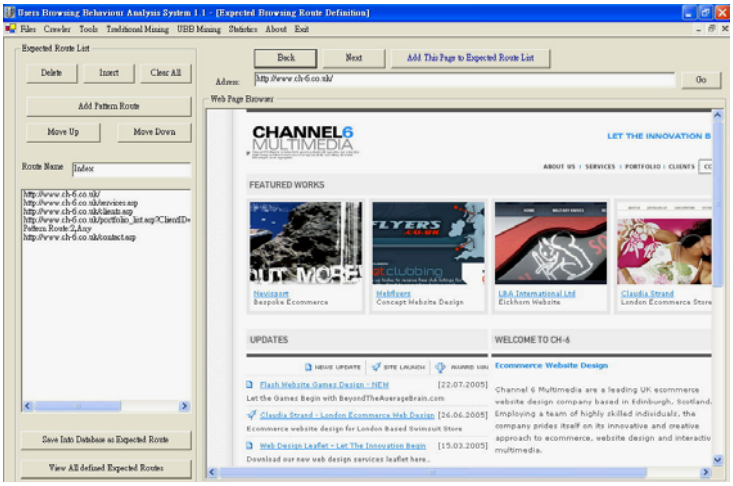

Figure 10 The interface of expected route definition assistance tool

This tool is consists of two components, one is mini-browser and another one is expected routes editor. Figure 8 shows the interface of the mini-browser in the expected routes definition assistance tool. This function allows website designer to define the expected routes by using "Navigation-Adding" approach. The mini-browser (in the right hand side of the interface) has some basic functions, which are similar to popular browsers, such as Internet Explorer, Netscape and Firefox. The design concept of this tool is to ensure the website designer can use the tool to define expect routes

without extra learning time. When the website designer browses their website, they can use the "Add this page to expected route list" button to add the URL address of current navigation page into the expected route list as a restricted subsequence of the UNB mining algorithm.

Figure 10 shows the interface of expected route definition assistance tool. The left hand side of the interface of this tool is the expected route list and the route editing area. It shows the subsequence (routes) which have already been added by the website designer. The editing area allows the website designer to edit the expected route, by using functions such as delete, insert, move-up and move-down. Finally, when the designer is satisfied with the route, it can be added to the pre-defined expected routes database (see the UNB mining process in figure 2).

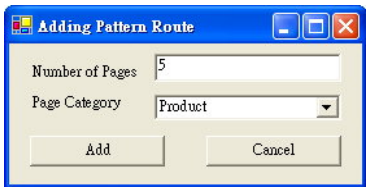

Figure 11 Flexible subsequence (pattern route) definition function

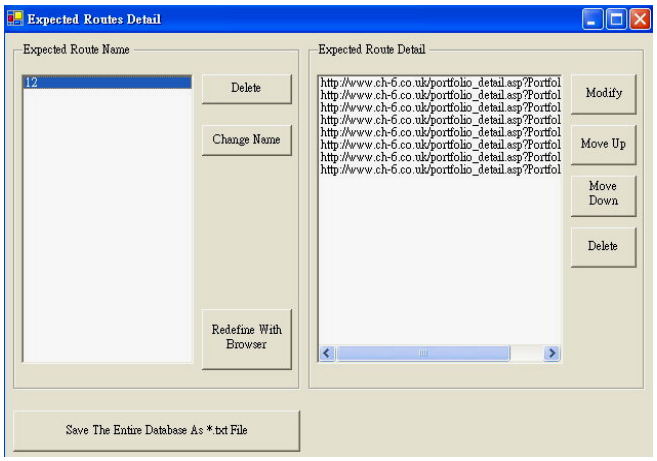

Figure 12 The expected route editor

In addition to "Navigation-Adding" approach to define the expected routes, the website designer can also add a flexible subsequence (pattern route) to the expected routes list through manual input. Using this function, the designer can manually input the number of pages as the p in flexible subsequence Fi{p;a} and page category as the a in Fi{p;a} (see figure 11). For example, if the website designer thinks that their users can navigate any six different web pages about information, his then can use the tool to input 6 in the field of number of pages and select the category of web pages as "information".

After saving a route into the expected routes database, the expected routes editor provides a function for the website designer to modify the predefined expected routes. Figure 12 shows the expected routes editor. The predefined expected routes can be deleted, the routes sequence can be moved up or down and new routes can be added into the expected routes list. These predefined expected routes can then be used by the UNB mining tool to discover instances of unexpected navigation behaviour.

4.2 The Main Interface of the UNB Mining System

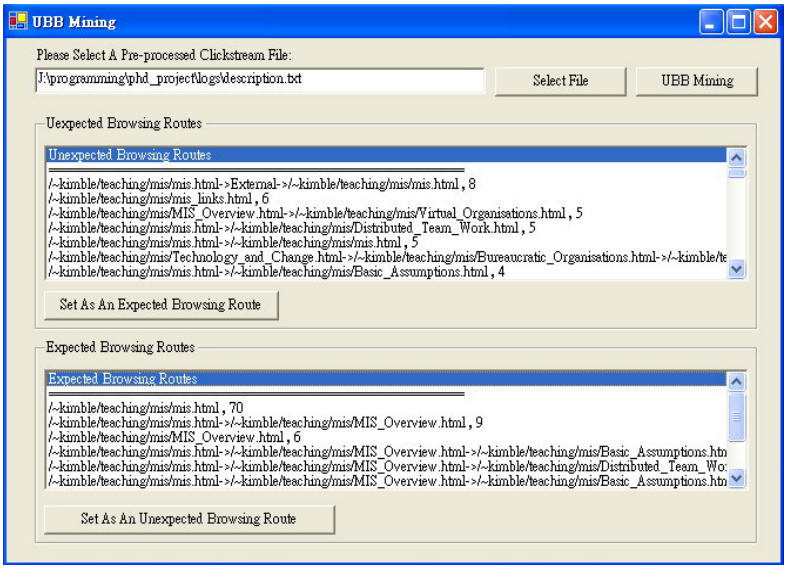

Figure 13 The UNB mining system interface

Figure 13 shows the interface of UNB mining system, which provides another way to help the website designer define an expected route in an efficient manner. After performing the UNB mining algorithm, the discovered unexpected and expected navigation routes are shown in different areas of the interface together with the detail of the routes (URL address) ranked by their frequency. The website designer can now review the results through the interface. If website designer think the discovered UNB is what they expected, they can select the UNB and set it as an expected navigation route. If, on the other hand, the website designer discovers an unexpected route it can be classed as a UNB in the same way.

In order to test the performance of this tool, we worked with a website designer to define expected navigation routes for three commercial websites. For each website, the designer defined five expected routes; on average, each expected route had 7 subsequences (mixing with restricted subsequence and flexible subsequence). Table 1 and 2 show two samples of expected navigation routes as defined by the website designer.

Table 1 A sample expected route of website 1

Prospective Customer Desired Route	
Type	URL Address
R_1	index.asp
R_2	services.asp
R_3	portfollio_list.asp
F_1	P=6, a=Clients
R_4	clients.asp
F_2	P=10, a=Clients
R_5	aboutus_news_list.asp
F_3	P=4, a=Information

Table 2 A sample expected route of website 2

User Looking for Job	
Type	URL Address
R_1	Index.asp
F_1	P=2, a=Information
R_2	jobs.asp

Before using expected navigation route definition tool, the website designer had found it extremely difficult to define an expected route. However, after only a short demonstration of this tool, the website designer was able to define other remaining expected routes without any further assistance. The whole process of defining 15 expected routes for 3 different websites took 21 minutes from start to finish.

5 An Empirical Study and the Performance Evaluation of UNB Mining

In order to test how the UNB mining works in real environment and to evaluate the performance of the UNB mining, an empirical study was undertaken with a website designer in a website design company: Channel 6 (http://www.ch-6.co.uk). In this section, we first present the collected expected routes and the UNBs that were discovered by using the UNB mining algorithm. In addition, possible recommendations for supporting website design improvement will also be discussed in this section according to those discovered UNBs. The second part of this section is a performance evaluation of the UNB mining algorithm using different numbers of predefined expected routes and user sessions.

5.1 Using the UNB Mining for Supporting Website Design Improvement

(1) The clickstream data and expected routes

The focus of the empirical study is the clickstream data from Channel 6's main website covers a period of three months; this was collected directly from the company's web server. The web server is based on Apache server and the clickstream data follows the format of W3C extended common log format. Because the format of the clickstream data follows the W3C ECLF, the data pre-processing step proposed in this paper is applied to pre-process the raw clickstream data.

The "Expect User Navigation Routes" are also essential data for UNB mining, which must be derived from transformation of the website design concept in the website designer's mind. Figure 14 shows two samples of expected user navigation routes, the first for job seekers and the second for prospective customers looking for E-commerce. These pre-defined expected routes then can be used to discover expected and unexpected user's navigation behaviour, applying the UNB mining technique

Channel 6 Job Seeker 1
http://www.ch-6.co.uk/index.asp
http://www.ch-6.co.uk/contact.asp
http://www.ch-6.co.uk/aboutus.asp
http://www.ch-6.co.uk/aboutus_vacancies.asp

Prospective Customer Looking for Ecommerce
http://www.ch-6.co.uk/index.asp
http://www.ch-6.co.uk/services.asp
Pattern Route:<=6, ∈ Information
http://www.ch-6.co.uk/services.asp
http://www.ch-6.co.uk/aboutus.asp
http://www.ch-6.co.uk/aboutus_testimonials.asp
http://www.ch-6.co.uk/contact.asp

Figure 14 Two sample expected users' navigation routes

(2) The discovered UNBs and recommendation for website design improvement

After six expected navigation routes were defined by the website designer, using the expected navigation route definition assistance tool, the UNB mining algorithm was performed to discover unexpected routes. The top 10 (sorted by frequency) unexpected navigation routes discovered are shown in table 3, in which the second field "URL" is the discovered unexpected users' navigation route and the third field is "Frequency", how often unexpected routes occur. The fourth field provides information about the most similar expected routes and the distance between the user routes. For example, ER1(6) means that the edit-distance between ER1 and unexpected navigation route NO.1 is '6'. In this paper, we consider that the shorter the edit-distance means the more similar between the ER and the unexpected navigation route.

From the unexpected navigation routes discovered, the website designer, using the information provided in the table, reviews the navigation behaviour of users, his design concept and the website structure. After matching the unexpected routes and the website design concept, the website designer can then redesign the website or add the unexpected routes to the expected one, if the discovered unexpected routes are acceptable.

Table 3 Unexpected Navigation Routes (Total sessions: 3749)

No.	Route	Frequency	Most Similar ER
1	/->/interactive->/interactive/main.asp->/interactive/content.xml	40	ER1(6) ER2(4) ER3(8) ER4(7) ER5(4) ER6(7)
2	/->/portfolio_list.asp->/clients	17	ER3(2) ER4(2)
3	/->/clients->/clients->/contact.asp	15	ER5(2)
4	/->/aboutus.asp->/contact.asp	14	ER2(1) ER5(1)
5	/aboutus_vacancies.asp->/aboutus_news_article.asp?ArticleID=24->/aboutus_vacancies.asp->/aboutus_news_article.asp?ArticleID=23	14	ER2(2)
6	/->/contact.asp->/aboutus.asp	13	ER1(1) ER2(1)
7	/->/portfolio_list.asp->/portfolio_detail.asp?PortfolioID=54	13	ER2(3) ER5(3)
8	/->/services->/services->/microstyle	12	ER5(2)
9	/->/portfolio_list.asp->/portfolio_detail.asp?PortfolioID=50	12	ER2(3)
10	/->/contact.asp->/portfolio_list.asp	11	ER2(2)

For example, the No.1 unexpected route in table 3 is:

/→/interactive→/interactive/main.asp→/interactive/content.xml

Here the most similar expected routes (with lowest distance) are ER2 (Channel 6 job seeker) and ER5 (prospective customer looking for e-commerce). Due to the edit-distance between ER2 (and ER5) and the unexpected navigation route NO.1 is '4 ', which is the lowest one. In this case, if the website designer thinks that the expected route could be a possible behaviour of a job seeker, then he can review the result based on matching the unexpected route and ER2. The ER2 is:

/→Pattern Route:<=5, ∈ Information→/aboutus_vacancies.asp

The first unexpected route in table 3 then has been chosen for further discussion. The unexpected route is:

/→/interactive→/interactive/main.asp→/interactive/content.xml

The greatest difference between this unexpected route and the expected route (/→Pattern Route:<=5, ∈ Information→/aboutus_vacancies.asp) is that users did not visit the /aboutus_vacancies.asp after they visited the information related pages. After reviewing the website structure and website design, we found that no link is provided in the /interactive/content.xml page.

Therefore, the appropriate recommendation from this analysis result could be "Would you please add a link in the /interactive/content.xml page to the /aboutus_vacancies.asp page".

Table 4 A sample heuristic for recommendations generation

1.	Selecting the top n unexpected navigation routes (and to fit the set frequency threshold).
2.	Selecting the most similar ER with highest edit distance.
3.	Checking the website design concept and reviewing the website structure of the selected ER.
4.	Generating recommendations or strategies to improve the design of website.
5.	Selecting another most similar ER and redo step 3 and step 4.

A sample heuristic is provided in table 4 to help the understanding of recommendation generation. These five steps in table 4 are normally performed manually but not automatically, (since they can be designed to be performed automatically) as we consider that, it is the best strategy to allow the website designer to review and generate recommendations based on their design concept. Thus, the system that we proposed in this paper can be treated as a system that can assist the website designer to make decision for website design improvement.

The generated recommendations may not always interesting/useful for website designer to redesign/restructure the website. However, it provides a good opportunity for the designer to review and check the design of their website. The website designer can also use the analysis results to evaluate the performance of the website, such as the accessibility, usability and the consistency between website designer's design concept and users' navigation behaviour.

5.2 Performance Evaluation of the UNB mining Algorithm

To evaluate the performance of the UNB mining algorithm, we tested the processing time of the UNB mining algorithm under different number of predefined expected routes (10, 20, 30, 40 and 50 routes) and different number of user sessions (100, 200… 900, and 1000 user sessions). The average number of browsed pages of a user session is six pages, and the average percentage of restricted rule and flexible rule in a predefined expected route are both 50%. The performance evaluation was performed under Windows XP operation system, 512Mb RAM, 1.3 GHz Intel Pentium 4 CPU. Figure 15 shows the performance evaluation result of the UNB mining algorithm.

The data of X-axis in figure 15 denotes the number of user sessions, and the data of Y-axis is the processing time of the UNB mining algorithm. The figure shows the performance of the UNB mining algorithm is very good. It not only processes the data and discovers the unexpected routes very fast (less than ten seconds), but the raise of the processing time is also not very intension with the increasing of the number of user sessions and expected routes.

Furthermore, the website designer also provided his opinion about the UNB algorithm and the assistant tool. At the outset, the website designer was requested to define his expected navigation routes without any assistance. However, the website designer found it very difficult to define the routes and did not even know how to start the task, especially to define the pattern routes of the expected route. Therefore, the tool to assist the definition of expected routes was introduced to the

website designer. After we had introduced the tool to the website designer, he found it easier than before to define the expected route simply by navigation and adding. Moreover, several tasks were assigned to the website designer, and it was helpful for him to start the definition of the expected user navigation routes.

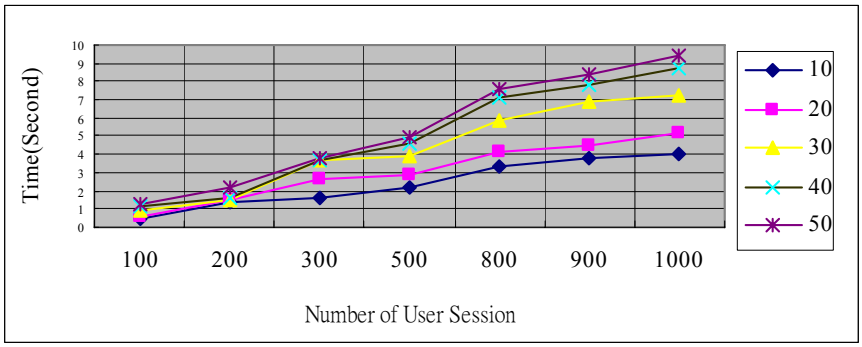

Figure 15 Performance evaluation of the UNB mining algorithm

6 Conclusions

In this paper, we proposed a novel approach to web usage mining based on using the web designer's 'design concept' for the site as the test for the value or otherwise of the patterns that were discovered: we called this approach UNB mining. Our objective in doing this was to create a tool that could be used to support the improvement of website design. The UNB mining approach described above can be applied to many different areas, such as E-commerce website design improvement, customer shopping process optimisation, student's behaviour in e-learning or adaptive user interface on web or multimedia systems.

UNB mining is based on detecting deviations from predefined patterns, which are specified by the designer or owner of the site. UNB mining is a sequential mining technique based on the concept of a consecutive common subsequence. There are two algorithms included in the UNB mining: the segmentation algorithm and the UNB discovery algorithm. Through the UNB mining, a website designer can discover interesting user's navigation behaviours, which are unexpected. The results can not only be used by a designer to review, improve or redesign their website, but can also be used to model a user's navigation behaviour.

In this paper, a useful tool to assist the definition of expected routes has been introduced. However, the system that we proposed in this paper is only a prototype system. The possible future development of this system can focus on the improvement of the system interface and the presentation of the analysis results, such as colourful and graphical presentation of the discovered unexpected navigation routes. Moreover, some arguments about possible techniques that may break-down the navigation behaviour of users still exist, such as navigation with tab function, searching but not browsing and some interactive techniques. These problems are interesting issues, and will be included in our future work to improve the proposed UNB algorithm and the system.

References

1. Banerjee, A. and Ghosh, J., Clickstream clustering using weighted longest common subsequences, In Proceedings of the 1st SIAM International Conference on Data Mining: Workshop on Web Mining, 2001.
2. Borges, J. and Levene, M., Data mining of user navigation patterns, In Masand B. M. and Spiliopoulu, M. eds. Web Usage Analysis and User Profiling, International WEBKDD'99 Workshop, San Diego, California, US, LNCS 1836, 2000, 92-111.
3. Canter, D., River, R. and Storrs, G., Characterizing user navigation through complex data structure, Behaviour and Information Technology, 4 (2), 1985, 93-102.
4. Clark, L., Ting, I. H., Kimble, C., Wright, P. and Kudenko, D., Combining ethnographic and clickstream data to identify user web navigation strategies, Information Research, Vol.11 No.2, paper 249, January 2006 [Available at http://InformationR.net/ir/11-2/paper249.html].
5. Cooley, R., Mobasher, B. and Srivastave, J., Web mining: information and pattern discovery on the World Wide Web, In Proceedings of the 9th IEEE International Conference on Tool with Artificial Intelligence, Newport Beach, CA, USA, 1997, 558-567.
6. Cooley, R., Tan, P. N. and Srivastava, J., Discovery of interesting usage patterns from web data, In Masand B. M. and Spiliopoulu, M. eds. Web Usage Analysis and User Profiling, International WEBKDD'99 Workshop, San Diego, California, USA, August 15, 2000, LNCS 1836, Springer-Verlag, 2000, 163-182.
7. Dömel, P., WebMap - a graphical hypertext navigation tool, In Proceedings of the Second International WWW Conference, Chicago, USA, 1994.
8. Fan, X. and Holsapple, C. W., An empirical study of web site navigation structures' impacts on web site usability, Decision Support Systems, 43, 2007, 476-491.
9. Eirinaki, M. and Vazirgiannis, M., Web mining for web personalization, ACM Transactions on Internet Technology, 3 (1), 2003, 1-27.
10. Fu, Y., Creado, M. and Ju, C., Reorganizing web sites based on user access patterns, In Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM'01), Atlanta Georgia, USA, November 5-10, 2001, 583-585.
11. G. Hooker, G. and Finkelman, M., Sequential analysis for learning models of navigation, In Proceedings of WebKDD 2004 Workshop on Web Mining and Web Usage Analysis, Seattle, WA, USA, 2004.
12. Kohavi, R., Mining e-commerce data: the good, the bad, and the ugly, In Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, April 16-18, Hong Kong, 2001, 8-13.
13. Kohavi, R., Mason, L. and Zheng, Z., Lessons and challenges from mining retail e-commerce data, Machine Learning, 57, 2001, 83-113.
14. Kothari, R., Mittal, R., Jain, V. and Mohania, M., On using page co-occurrences for computing clickstream similarity, In Proceedings of SIAM International Conference on Data Mining, San Francisco, CA, USA, May 1-3 2003.
15. Lee, W. P., Liu, C. H. and Lu, C. C., Intelligent Agent-based systems for personalized recommendation in internet commerce, Expert Systems with Applications, 22, 2002, 275-284.
16. Mobasher, B., Dai, H., Luo, T. and Nakagawa, M., Discovery and Evaluation of Aggregate Usage Profile for Web Personalization, Data Mining and Knowledge Discovery, 6, 2002, 61-82.
17. Nasraoui, O., Cardona, C. and Rojas, C., Mining evolving web clickstreams with explicit retrieval similarity measures", In Proceedings of International Web Dynamics Workshop, International World Wide Web Conference, New York, NY, USA, May 2004.

18. Nasraoui, O. and Pavuluri, M., Complete this puzzle: a connectionist approach to accurate web recommendation based on a committee of predictors, In Proceedings of WebKDD 2004 workshop on Web Mining and Web Usage Analysis, Seattle, WA USA, 2004, 47-60.
19. Nielsen, J., Farrell, S., Molich, R., and Snyder, C., E-Commerce User Experience, Nielsen Norman Group, 2001.
20. Pather, S., Erwin, G. and Remenyi, D., Measuring e-commerce effectiveness: a conceptual model, In Proceedings of SAICSIT Conference, 2003, 143-152.
21. Perkowitz, M. and Etzioni, O., Adaptive web sites, Communications of the ACM, 143 (8), 2000, 152-158.
22. Raphael, A. and Brower, G., Usability testing: Think-aloud protocol, User-Centered Information Design Workbook, [Available at http://www.washington.edu/webguides/workbook/] (Access date: 7 March 2008).
23. Ristad, E. S. and Yianilos, P. N., Learning string edit distance, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20 (2), 1998, 522-532.
24. Shahabi, C. and Kashani, F. B., Efficient and anonymous web usage mining based on client-side tracking", In Proceedings of WEBKDD 2001, 2001, 113-144.
25. Srivastava, J., Cooley, R., Deshpande, M. and Tan, P. N., Web usage mining: discovery and applications of usage patterns from web data, SIGKDD Explorations, 1(2), 2000,12-23.
26. Tan, P. N. and Kumar, V., Discovery of web robot sessions based on their navigation patterns, Data Mining and Knowledge Discovery, 6, 2000, 9-35.
27. Ting, I. H., Kimble, C. and Kudenko, D., Visualizing and Classifying the Pattern of User's Browsing Behaviour for Website Design Recommendation, In Proceedings of the First International Workshop on Knowledge Discovery in Data Stream (ECML/ PKDD 2004) Pisa, Italy, 24 September 2004, 101-102.
28. Ting, I. H., Kimble, C. and Kudenko, D., A pattern restore method for restoring missing patterns in server side clickstream data, In Zhang, Y. et al. eds. APWeb 2005, LNCS 3399, Springer-Verlag, 2005, 501-512.
29. Wu, H., Gordon, M., Demaagd, K. and Fan, W., Mining web navigations for intelligence, Decision Support Systems, 41, 2006, 574-591.
30. Yen, P.-C., The design and evaluation of accessibility on web navigation, Decision Support Systems, 42, 2007, 2219-2235.