Using Petri Nets to Enhance Web Usage Mining[1]

Shih-Yang Yang

Department of Information Management
Kang-Ning Junior College of Medical Care and Management
Nei-Hu, 114, Taiwan
Shihyang@knjc.edu.tw

Po-Zung Chen, Chu-Hao Sun

Department of Computer Science and Information Engineering
Tamkang University
Tamsui, 251, Taiwan
pozung@mail.tku.edu.tw, steven.sun@fubon.com

Abstract: Precise analysis of the web structure can facilitate data processing and enhance the accuracy of the mining results in the procedure of web usage mining. Many researchers have identified that pageview identification and path completion are of great importance in the result of web usage mining. Currently, there is still a lack of an effective and systematic method to analyze and deal with the two steps.

In the present study, we propose the application of Petri Nets (PN), a model used to analyze the framework of webpages in a website. We adopt Place in the PN model to represent webpage on the websites and use Transition to represent hyperlink. The study explores how to undergo the pageview identification after we use the PN model to conduct the analysis of the framework and then obtain incident matrix. Likewise, we use reachability property in the model to undergo path completion.

Keywords: Petri nets, Web usage mining, Data Preprocessing

1 Introduction

This study introduces a method to enhance a web usage mining using Petri Nets (PN) in modeling a web structure. We also observe that PN can help resolve pageview identification and path completion, particularly in a complex webpage comprising many frames in one single pageview.

[1] This work is supported by the National Science Council, Taiwan, ROC, under Grant # NSC 93-2213-E-032-016.

As the internet becomes globally popular, more and more business transactions have been done through websites nowadays. To achieve a better website management and design capability, many website management personnel started reviewing their site users' webpage browsing frequency, sequence and even duration on each webpage browsed, adopting the user's web usage profiles. Hence, web usage mining has become a hot research topic.

A website is comprised of a series of web pages and hyperlinks. Although most web-usage-mining related studies only focus on, and analyze, the users' web usage profiles, some related studies [1] [2] point out that a good-quality analysis on web structure can provide gains and benefits for web usage mining, too. Most previous web usage mining utilizes a webpage as an analysis component, but after the HTML standard language, such as 'frameset', started being applied to webpage design, a website user's browsed display might exhibit more than one web page concurrently; hence, an analysis based on a web page component can no longer truly reveal the user's usage states and behaviors. A pageview is defined as the visual picture of a web page in a specific client environment at a specific point in time [3]. The use of pageview as the analysis component for web usage mining could more accurately reveal a website user's browsing behaviors, but such method will increase the complexity of data preprocessing during web usage mining.

In general, the process of web usage mining can be divided into three parts, namely preprocessing, pattern discovery, and pattern analysis [4]. Preprocessing will process untreated site files and user profile data into page classification, site topology and server session files. Pattern discovery will process a server session file into rules, patterns, and statistics information. Pattern analysis looks into the rules, patterns, and statistics information obtained from pattern discovery for results that will be of interest to the management personnel.

The preprocessing step can be generally divided into content preprocessing, structure preprocessing, and usage preprocessing. Content preprocessing classifies site files into page classification to help pattern analysis. Structure preprocessing converts site files into a site topology to help pageview identification and path complete. Usage preprocessing coverts raw usage data into click stream of episodic user behaviors via some steps such as data cleaning, user identification, session identification, pageview identification, path complete (if necessary).

Concerning data cleaning, user identification, and session identification faced in a usage preprocessing process, many previous studies have investigated a great deal and rendered some processing steps and methods [5] [6]; although some researchers point out that an understanding on website structure can be useful in carrying out pageview identification and path complete [1][7][8], it is still hard to establish fine solutions for pageview identification and path complete.

Petri Nets (PN) is a high-level graphical model widely used in modeling system activities with concurrency. PN can store the analyzed results in a matrix for future follow-up analyses, and some already-verified properties held by PN, such as reachability, can also be used to resolve some unsettled problems in the model. According to the definition in [9], it is formally defined as a 5-tuple PN of (P, T, I, O, M_0), where

(1) P = $\{p_1, p_2 \ldots p_m\}$, a finite set of places;

(2) T = $\{t_1, t_2 \ldots t_n\}$, a finite set of transitions; $P \cup T \neq \emptyset$, and $P \cap T = \emptyset$;

(3) I: P x T→N, an input function that defines directed arcs from places to transitions, where N is a set of nonnegative integers;

(4) O: T x P →N, an output function that defines directed arcs from transitions to places;

(5) M_0: P →N, the initial marking. A marking is an assignment of tokens to a place;

PN is carried out by firing transitions. A transition, t, is said to be enabled if each input place, p, of t contains at least an amount of token equal to the weight of the directed arc connected to t from p. In a PN model, we can utilize the different token amounts in the places to represent the different system states. Since a fire of transition in the system often can be associated with a change of the token amount in a place, PN hence can represent, or model, the system dynamic behaviors via the fire of transitions. An incidence matrix records all token-amount changes in all places after all fired transitions. For PN with n transitions and m places, the incidence matrix A, where A=$[a_{ij}]$, is an n×m matrix of integers; its typical entry is given by

$$a_{ij}=a_{ij}^{+} - a_{ij}^{-}, \tag{1}$$

where

a_{ij}^{+}=O (t_i, p_j), the weight of the arc from Transition i to its Output Place j, and

a_{ij}-=I (ti, p_j), the weight of the arc to Transition i from its Input Place j;

a_{ij}^{+}, a_{ij}^{-} and a_{ij} represent the number of tokens removed, added, and changed in Place p_j, respectively, when Transition t_i fires once.

Concerning the PN properties, reachability is one of them that is often discussed and utilized; this property is originally expected to explore if the modeled system can be transited from one state to another. During the processing or operations, we can also simultaneously trace out what are the possible intermediate states during the transitions from the initial state to the destination one. In a PN model, a marking M_i is said to be reachable from a marking, M_0, if there exist a sequence of transition firings which can transform a marking, M_0, to M_i. According to the definition in [10], we can obtain the state equation as shown in (2) as follows, where A is an incidence matrix, $\{U_0, U_1, \ldots, U_d\}$, representing the firing sequence from M_0 to M_d if M_d is reachable from M_0:

$$M_d = M_0 + A^T \sum_{k=1}^{d} Uk \tag{2}$$

The purpose of this study is to enhance web usage mining by PN. We use a parsing algorithm to retrieve the website's webpage contnets, analyze the webpage's contents and find the incidence matrix which represents web structure. We can apply the web structure information in the incidence matrix and the reachability properties obtained from the PN model to help proceed with pageview identification and path complete. Also we apply Markov analysis to provide usage statistics in pattern discovery. Figure 1 represents our proposed PN based web usage mining structure.

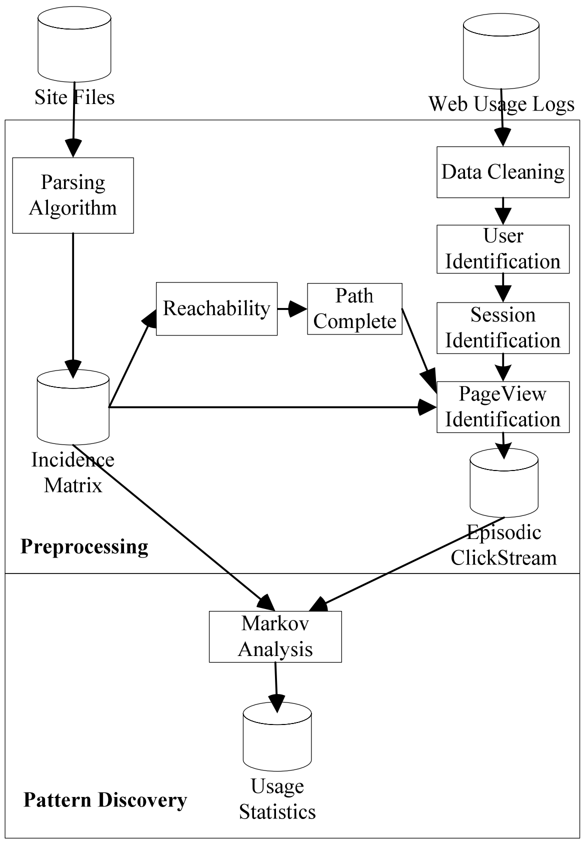

Figure 1
PN based web usage mining structure

In this paper, we focus on how to utilize a parsing algorithm to retrieve the website's webpage contents, analyze the webpage's contents and find the incidence matrix which represents web structure. Also we show how to apply the reachability properties to help pageview identification and path complete processes. For the part of how to apply Markov analysis to provide usage statistics in pattern discovery, please refer to [11].

2 Modeling a Website Structure with Petri Nets

A website is comprised of many web pages, where the web pages can be linked to from one another via hyperlinks. We will use places in PN to represent webpage, and transitions to hyperlinks.

Using PN as the website structure model, we can attain:

PN= (P, T, I, O, M_0),

Where

P stands for the web pages in the website;

T represents the hyperlinks in the web pages;

I denote the removed webpage volumes in the browser's pageviews after the hyperlinks are fired;

O denotes the added webpage volumes in the browser's pageviews after the hyperlinks are fired;

M_0 signifies the retrieved webpage in the browser after the first time the user enters the website.

In the parsing algorithm, we get the contents of the default root page first, and then traverse the web topology by visiting hyperlinks in all the web pages one by one to find the corresponding pageviews and construct the web structure accordingly.

There are three data structures, namely Pageview, PageviewQueue and VisitedPageviewTree. Pageview is a set of places, which represents the relevant information of web pages in the pageview. PageviewQueue keeps the pageview information about which pageview should be parsing next. By this structure, we implement a breadth-first constructing algorithm. VisetedPageviewTree maintains the tree of visited pageview, which is used to avoid repeatedly visiting the same pageview. There are five functions in this algorithm, namely GetWebPageContent, Parse, Generate_pageview, Generate_Transition and Generate_Aij. GetWebPageContent will get the content of the web page from a given web URL. Parse will find the HTML hyperlinks and its corresponding pageview from a given web page content. By visiting hyperlinks in all the web pages one by one, Generate_pageview will generate the corresponding destination pageview and add new place if necessary. Generate_Transition and Generate_Aij will construct the web structure represented by incidence matrix Aij.

The parsing algorithm is shown in Figure 2.

```
Input: Pageview, a set of places.
Output: A_ij, the incidence matrix of PN
Data Structure:
VisitedPageviewTree, maintain the tree of visited pageviews.
PageviewQueue, keep the pageview information about which pageview should be
parsing in the next step.
1.  Parsing_Algorithm (Pageview, Aij)
2.  home=Pageview;
3.  PageviewQueue.Enqueue(home);
4.  Do while  PageviewQueue.count > 0
5.      current_pageview = PageviewQueue.Dequeue();
6.      If VistedPageviewTree.Exist(current_pageview)=false Then
7.          VisitedPageviewTree.Add(current_pageview);
8.        For all places in current_pageview;      //current_place.
9.        web_page_URL=current_place.URL;
10.       web_page_content= GetWebPageContent(web_page_URL);
11.       Set_of_link=Parse(web_page_content);
12.          For each link in Set_of_link
13.              destination_pageview=Generate_Pageview(link,     current_pageview);
                 //add new place if necessary.
14.              transition=Generate_Transition(current_pageview,
                 destination_pageview);
15.              Generate_Aij(transition, Aij);
16.              PageviewQueue.Enqueue(destination_pageview);
17.          End For
18.      End For
19.    End If
20. Loop
```

Figure 2
The parsing algorithm

Taking the website represented in Table 1 as an example, Default.htm is the homepage of this website.

Table 1
A website example

Webpage names	The links in the webpage
1	<a href="A.htm" target="right">A</a>
	<a href="B.htm" target="right">B</a>
A	<a href="C.htm" target="_top">C</a>
B	<a href="D.htm" target="_top">D</a>

C	<a href="D.htm" >D</a><p>
	<a href="Default.htm" >Default</a><p>
D	<a href="C.htm" >C</a><p>
Default	<a href=”Index.htm”>Index</a>
Index	<frameset cols="20%,80%"> <frame src="1.htm" name="left" > <frame src="A.htm" name="right">

Table 2 illustrates the execution of the main loop in the parsing algorithm, ‘for all places in current_pageview(code# 8~18)’, where P# represents the place number of current_place in the current_pageview and T# is the transition number returned by Generate_Transition(), code# 19. Note that, the visited pageview (1,2,3)*, (6)*, (0)* and (5)* founded by the function VistedPageviewTree.Exist() are indicated by * in Table 2.

Table 2
The execution of main loop in the parsing algorithm

Current Pageview	P#	T#	Pageviw_Queue	Generate_Aij	
				Aij=-1	Aij=1
(0)	0	0	{ (1, 2, 3) }	A[0,0]	A[1,0] A[2,0] A[3,0]
(1, 2,3)	1		{ }		
	2	1	{ (1, 2, 4) }	A[1,1] A[2,1] A[3,1]	A[1,1] A[2,1] A[4,1]
	3	2	{ (1, 2,4), (5) }	A[1,2] A[2,2] A[3,2]	A[5,2]
(1, 2,4)	1		{ (5) }		
	2	3	{ (5),(1, 2,3) }	A[1,3] A[2,3] A[4,3]	A[1,3] A[2,3] A[3,3]
	4	4	{ (5),(1, 2,3), (6) }	A[1,4] A[2,4] A[4,4]	A[6,4]
(5)	5	5	{(1,2,3), (6), (6) }	A[5,5]	A[6,5]
		6	{(1,2,3),(6), (6),(0) }	A[5,6]	A[0,6]
(1,2,3)*			{(6), (6),(0) }		
(6)	6	7	{(6),(0),(5) }	A[6,7]	A[5,7]
(6)*			{(0) (5)}		
(0)*			{(5)}		
(5)*			{ }		

Table 3 shows the place added by the function Generate_Pageview() and its corresponding webpage. Table 4 shows the transition number generated by the function Generate_Transition() and its corresponding hyperlink.

Table 3
The place number and its corresponding webpage

Place #	Webpage names
0	Default
1	Index
2	1
3	A
4	B
5	C
6	D

Table 4
The transition number and its corresponding hyperlink

Transition #	Hyperink
0	<a href=”Index.htm”>Index</a>
1	<a href="B.htm" target="right">B</a>
2	<a href="C.htm" target="_top">C</a>
3	<a href="A.htm" target="right">A</a>
4	<a href="D.htm" target="_top">D</a>
5	<a href="D.htm" >D</a><p>
6	<a href="Default.htm" >Default</a><p>
7	<a href="C.htm" >C</a><p>

The web structure can be expressed by the incidence matrix, A_{ij}, in Figure 3, and its corresponding PN in Figure 4.

$$\lfloor A_{ij} \rfloor_{7\times 8} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0^* & -1 & 0^* & -1 & 0 & 0 & 0 \\ 1 & 0^* & -1 & 0^* & -1 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 \end{bmatrix}$$

Figure 3
The incidence matrix representing the webpage structure shown in Table 1.
(0* denotes that the values of the input and output function values of the place are equal when the transition is fired.)

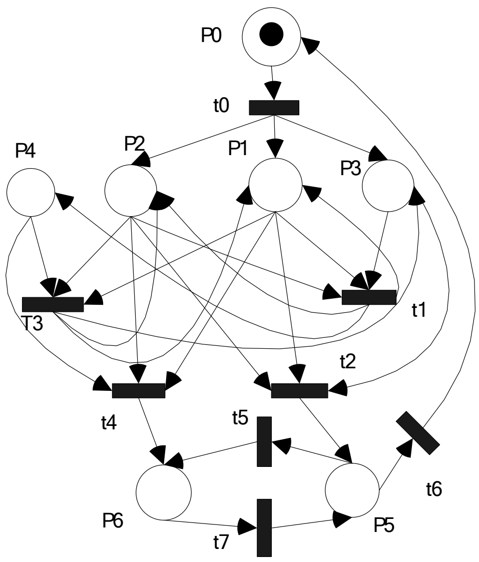

Figure 4
The Petri Nets corresponding to the website of Table 1

3 Using PN Model to Enhance Data Preprocessing

In the process of browsing a website, the user will only retrieve the first webpage through a direct key-in of the webpage's website address to request the webpage contents; very often the user's afterward browsed web pages are requested via service requests for new webpage contents toward the websites pointed to by the browser, according to the attributes of the hyperlinks, after the user clicks said hyperlinks within the browser. The browser will then display the related contents to the user based on the web pages' protocols after the requested web pages are transmitted to the user from the websites. Hence, the user's browsing process can be taken as an inter-webpage transition process.

The main task of web usage mining is to retrieve the information meaningful to the system management personnel from the web server's accumulated usage profiles left by all the browser users. As the profiles are only the sequentially recorded contents of the services provided by the web server, the profiles not only could contain multiple browsing profiles of different browser users but also could take in some extra or erroneous profiles. The website management personnel must proceed with preprocessing to these usage profiles if they are to correctly analyze said users' webpage-contents usage sequence. Hence, a data preprocessing is needed to enhance information processing before we can analyze the usage profiles. The first step of data preprocessing often is to delete the erroneous or useless data or columns in the usage profiles via data cleaning; after finishing data cleaning, we next need to extract different users' usage profiles with user identification, using the user's IP column. Each user's website usage profile might include his multiple website usage records within a period of time; hence, we need to divide said user's usage profile into his multiple browsing session log files.

After completing said session identification, we still need to face problems related to path complete and pageview identification during data preprocessing. As we propose to use PN to model a website structure, we can further apply the incidence matrix and related properties obtained from the Petri Nets model to help proceed with path complete and pageview identification.

In the part of structure preprocessing, we first utilize a site spider to retrieve the website's webpage contents; we then use parsing program to analyze the webpage's contents to locate all places, i.e. the web pages and transitions, which are the hyperlinks causing the transitions; we also analyze the incidence matrix between the places and the transitions.

In the part of usage preprocessing, we will sequentially finish data cleaning, user identification, and session identification. In pageview identification, we will proceed with it using the pageview information provided by the incidence matrix. If missing paths are found during the identification process, we will activate the path complete process to locate the possible missing paths to carry out the path complete process and, then, continue working on pageview identification. The related component diagram are referred to Figure 5.

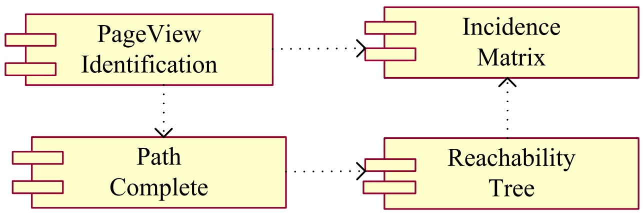

Figure 5
The component diagram of pageview identification and path complete

3.1 Pageview Identification

A pageview represents a display comprising all the webpages appearing concurrently in the browser during the process the user reads a webpage. The main function of pageview identification is to identify and mend the user session log file, with the help of the web structure information, to find out the real displayed contents and sequence in the browser during the user's browsing process. We adopt Petri Nets to model a website structure and proceed with pageview identification with the help of an incidence matrix.

Table 5
A user session before pageview identification

TID	Request file	Referring file
1	Default.htm	_
2	Index.htm	Default.htm
3	1.htm	Index.htm
4	A.htm	Index.htm
5	B.htm	1.htm
6	D.htm	B.htm
7	C.htm	D.htm

Taking a user profile as an example as shown in Table 5, we observe from the incidence matrix, as shown in Fig. 3, that when the system retrieves the first usage profile having a transition identification (TID) as 1, the transition related to P_0, hence, consists of T_0 only due to that the place corresponding to Default.htm is P_0 and that $[A_{0j}]=(-1, 0, 0, 0, 0, 0, 1, 0)$; we can recognize that T_0 indicates the transformation of the pageview comprising Index.htm, 1.htm, and A.htm from the pageview containing Default.htm only, for $[A_{i0}]=(-1, 1, 1 ,1, 0, 0, 0)$. The first, second and third usage profiles in Table 5 are Index.htm, 1.htm and A.htm, with a good match; hence, we can confirm that the three profiles represent the pageview comprising Index.htm, 1.htm and A.htm. As we proceed with the identification of the fifth data, we observe that T_1 stands for the transition to Index.htm, 1.htm, and B.htm, and T_3 the transition to C.htm as there are only two transitions, T_1 and T_3,related to Index.htm, 1.htm and A.htm; the request webpage of the fifth profile is B.htm. Hence, we can arrive at that the fifth profile represents the user's entering the pageview comprising Index.htm, 1.htm and B.htm. Based on such identification method, we can find out the sixth and seventh profiles representing, respectively, the user's entering the pageview comprising D.htm and C.htm; the user's profiles after completing the pageview identification are shown in Table 6.

Table 6
A user session after pageview identification

Pageview ID	Request file
1	Default.htm
2	Index.htm
2	1.htm
2	A.htm
3	Index.htm
3	1.htm
3	B.htm
4	D.htm
5	C.htm

3.2 Path Complete

Web Usage Logs are the records of the web content requests that all web users have made to that website. It's possible that web user encounters problem of Browser cache or Proxy server when requesting service from website. So it's not unusual some user's records are missing from the web usage logs. If it's not managed well, error will happen in Page identification process and, in turn, affect the correctness of web usage mining.

So Path Complete needs should be activated to patch it once the web usage logs are found incomplete during Pageview identification process

Taking the user session in Table 5 for example, if we lost the web logs from TID 2 to 5. After the pageview identification process identified the pageview of first transaction, it will find that the current pageview cannot transfer to next pageview directly. The process will launched the path complete process to find out the lost transition. Since we can have the initial state marking M_0=[1, 0, 0, 0, 0, 0, 0]^T and the destination state marking M_d=[0, 0, 0, 0, 0, 0, 1]^T. According to (2), the equation can be illustrated as Figure 6.

$$\begin{bmatrix}0\\0\\0\\0\\0\\1\\0\end{bmatrix} = \begin{bmatrix}0\\1\\1\\0\\1\\0\\0\end{bmatrix} + \begin{bmatrix}-1 & 0 & 0 & 0 & 0 & 0 & 1 & 0\\ 1 & 0^* & -1 & 0^* & -1 & 0 & 0 & 0\\ 1 & 0^* & -1 & 0^* & -1 & 0 & 0 & 0\\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & -1 & -1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & -1 & -1 & 1\\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1\end{bmatrix} \times [T_k]$$

Figure 6
The state equation of path complete

We can also obtain the $[T_k]$=[1, 1, 0, 0, 1, 0, 0, 0]^T, which means that the initial state can be transferred to destination sate through continuous fired in T_0, T_1 and T_4. To take one step ahead, we can find the firing sequence of M_0 to M_d is T_0->T_1->T_4 in the help of incidence matrix.

Conclusions

A good-quality data preprocessing is one of the keys in determining if the web usage mining will be a success; however, relatively few researchers have expressed how to proceed with pageview identification and path complete in data preprocessing. In this paper, we propose the use of Petri Nets as a webpage structure model for website simulation and demonstrate that with this model, we can not only adopt Petri Nests' incidence matrix to help carry out pageview identification but also utilize Petri Nests' reachability property to fulfill path complete.

References

[1] Robert Cooley: The Use of Web Structure and Content to Identify Subjectively Interesting Web Usage Patterns, ACM Transactions on Internet Technology, Vol. 3, No. 2, May 2003, pp. 93-116

[2] Berendt, B. Mobasher, M. Nakagawa, M. Spiliopoulou: The Impact of Site Structure and User Environment on Session Reconstruction in Web Usage Analysis, Proc. WEBKDD 2002: Mining Web Data for Discovery Usage Patterns and Profiles, LNCS 2703 Springer-Verlag, 2002

[3] W3C Web Characterization Terminology & Definitions Sheet, http://www.w3.org/1999/05/WCA-terms/

[4] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, Pan-Ning Tan: Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data, SIGKDD Explorations, Vol. 1, Issue 2, Jan. 2000, pp. 12-23

[5] Robert Cooley, Bamshad Mobasher, Jaideep Srivastava: Data Preparation for Mining World Wide Web Browsing Patterns, Journal of Knowledge and Information System, 1(1), 1999, pp. 5-32

[6] Murat Ali Bayir, Ismail H. Toroslu, Ahmet Cosar: A New Approach for Reactive Web Usage Mining Data Processing, Proceeding of the 22nd International Conference on Data Engineering Workshops (ICDEW'06)

[7] M. Spiliopoulou, B. Mobasher, B. Berendt, M. Nakagawa: A Framework for the Evaluation of Session Reconstruction Heuristics in Web Usage Analysis, INFORMS Journal on Computing, 2003

[8] Magdalini Eirinaki, Michalis Vazirgiannis: Web Mining for Web personalization, ACM Transactions on Internet Technology, Vol. 3, No. 1, Feb. 2003, pp. 1-27

[9] Jiacun Wang: Timed Petri Nets, Theory and Application, Boston: Kluwer Academic Publishers, 1998

[10] Tadao Murata: Petri Nets: Properties, Analysis and Applications, in Proceedings of the IEEE, Vol. 77, No. 4, 1989

[11] BinHong Wang: Markov Analysis for STPN Web Structure Model, Master thesis, Department of Computer Science and Information Engineering, Tamkang University, 2007, in Chinese