Modeling and Analysis the Web Structure Using Stochastic Timed Petri Nets

Po-Zung Chen, Chu-Hao Sun
Department of Computer Science and Information Engineering, Tamkang University, Taipei County, Taiwan
Email: pozung@mail.tku.edu.tw, 894190320@s94.tku.edu.tw

Shih-Yang Yang
Department of Information Management, Kang-Ning Junior College of Medical Care and Management , Taipei, Taiwan
Email: shihyang@knjc.edu.tw

***Abstract*—Precise analysis of the Web structure can facilitate data pre-processing and enhance the accuracy of the mining results in the procedure of Web usage mining. STPN（Stochastic Timed Petri Nets） is a high-level graphical model widely used in modeling system activities with concurrency. STPN can save the analyzed results in an incidence matrix for future follow-up analyses, and some already-verified properties held by STPN, such as reachability, can also be used to solve some unsettled problems in the model. In the present study, we put forth the use of STPN as the Web structure model. We adopt Place in the STPN model to represent webpage on the websites and use Transition to represent hyperlink. Through the model, we can conduct Web structure analysis. We simultaneously employ the Web structure analysis information in the incidence matrix and the reachability properties, obtained from the STPN model, to help proceed with pageview identification and path completion at the data preprocessing phase.**

***Index Terms*—Web usage mining, data preprocessing, Stochastic Timed Petri Nets, reachability behavior, pageview identification, path completion.**

I. INTRODUCTION

As the Internet is increasingly prevalent worldwide and bringing out variations in business transactions, website management and design capability have been two of the potent constituents in the area of information science. To achieve better website management and design capability, many website management professionals started to examine site-users' webpage browsing frequency, sequence and even duration through the Web usage profiles. Hence, Web usage mining has become a hot research topic.

Web mining is the application of data mining techniques to discover patterns from the Web. According to analysis targets Web mining is divided into three types, namely Web content mining, Web structure mining and Web usage mining [1][2][3].

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 [4][6][7] and pageview identification [5] during data preprocessing.

In [8] we propose to use STPN to model a website structure, we can further apply the incidence matrix and related properties obtained from the STPN model to help proceed with pageview identification and path complete. The data preprocessing process includes into two parts. (Figure 1).

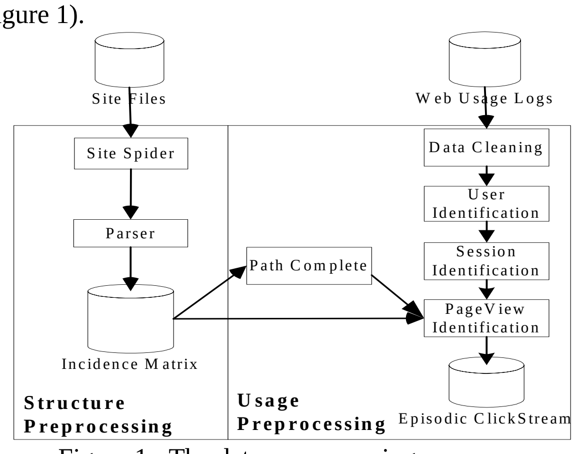

Figure 1: The data preprocessing process

In the part of structure preprocessing [6], 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 [9], 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 is referred to Figure 2.

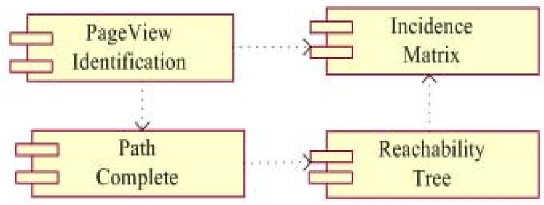

Figure 2: The Component Diagram of Pageview Identification and Path Completion

STPN （Stochastic Timed Petri Nets） is a high-level graphical model widely used in modeling system activities with concurrency. STPN can save the analyzed results in an incidence matrix for future follow-up analyses, and some already-verified properties held by STPN, such as reachability, can also be used to solve some unsettled problems in the model.

In the present study, we put forth the use of STPN as the Web structure model. Through the model, we can conduct Web structure analysis. We simultaneously employ the Web structure analysis information in the incidence matrix and the reachability properties, obtained from the STPN model, to help proceed with pageview identification and path completion at the data preprocessing phase.

The structure of this paper is as follows: Section 2 depicts the definition and related research of STPN. In section 3, introduces how to exploit the STPN to identify Pageview. In section 4, we present how to Use reachability characteristic in STPN to help path completion. Finally, we conclude this paper in section 5.

II. BACKGROUND AND RELATED WORK

A. Stochastic Timed Petri Nets (STPN)

In the research of Murata[10], Reisig[11],Marsan[12], STPN is the model used to analyze and represent the system activities and behaviors, which can apply the important structure property and behavior property to take a quality analysis.

A STPN model can be expressed as:

STPN = (P, T, F, W, M0, E)

P ={p1, p2, …, pm} is a finite set of laces, expressed as a circle in STPN graph. Each represents a place.

T = {t1, t2, … , tn} is a finite time-consumed transition set expressed as black solid rectangle in STPN.

F = {finput, foutput} is the set of all arcs in STPN, and finput (P x T) and foutput (T x P) are defined as input line and output line of the transition respectively.

W = Weighting Function(W: F ' N), in which W(p,t) indicates the weighting value of arc finput link from place p to transition t, and W(p,t) indicates the weighting value of arc foutput from transition to place p. In addition, the weighting value must be a positive integer.

M0 = The initial value of the system, representing the set of tokens in each place. Besides, M represents the set of all states of the system, so Mk = {M0, M1, M2, …, Mn}.

E={e0, e1, … , en} is the Exponential Time Rate with respect to the consumed time of a transition.

To describe the graph structure of STPN, a Place-Transition Table or called STPN Matrix has been defined; therein, C is a two-dimensional m x n matrix, m and n are the numbers of Places and Transitions respectively; STPN Matrix can be formulated by the approach that the negative value of W(P ' T) and positive value of W(T ' P), and then inserted the value into STPN Matrix C.

B.STPN reachability behavior characteristic

STPN is a tool which widely used for analyze and system confirmation. [10] mentioned that the postulation of reachability problem: " If the destination state [Md] may arrived by way of the original state [Mo], then by triggers a set which consisted by some transition, possible to be solved using the linear equation, moreover has at least a group of solutions. " That is, in STPN, its state transfer may express by the STPN reachability behavior characteristics equation:

$$[M0]_{nx1} + [C_{ij}]_{nxm}[t_j]_{mx1} = [Md]_{nx1} \quad (1)$$

n : represents total positions count in STPN.

m : represents total transitions count in STPN.

[M0]nx1 : unidimensional vector, represents the original state of STPN.

[M0]nx1 : unidimensional vector, represents the destination state of STPN.

[Cij]nxm : represents a adjacent matrix, indicate after expressed the jth shift movement triggers, the ith position mark count which obtains or loses.

[tj]mx1 : this unidimensional vector represents the solution of the linear equation, also is the answer which reachability problem ask. This group of equation solution expresses

the original state [M0] by way of a set which contains after some triggered transition in [tj], and then arrives the destination status [Md].

C. Construct a Web structure: using STPN

[3] use the following definition to define a Web structure:

M = [<F1; F2; ……; Fn>]
F = {hf, L1, L2, ……, Lm}
L = <r, (h1, g1) | …… | (hp, gp)>

M: represents webs which consist of many Web pages or frames (Fi).
Fi: Web pages or frames (Fi), it consisted of a HTML file name (hf) and its related hyperlink (Li).
Li: it's a hyperlink, it consists of a linking form (r) and the Web content scope in each different page frame combination composes. In here gi represents the Web file or page frame hi the position name which appears in the Web content scope after a hyperlink is triggered.

Take a Web structure as an example, if the hyperlink grammar in its Web file-links likes Table I:

TABLE I.
HYPERLINK CONTENT OF A WEB STRUCTURE

File	Content of Web pages
Default	<a href="index.htm">
Index	<frameset cols="20%,80%"> <frame src="1.htm" name="left" > <frame src="A.htm" name="right">
1	<a href="A.htm" target="right"> <a href="B.htm" target="right">
A	<a href="C.htm" target="_top">
1	<a href="A.htm" target="right"> <a href="B.htm" target="right">
B	<a href="D.htm" target="_top">
C	<a href="D.htm" > <a href="index.htm" >
D	<a href="C.htm" >
Default	<a href="index.htm">

According the Web structure definitions above, we can represent the Web structure in Figure 3:

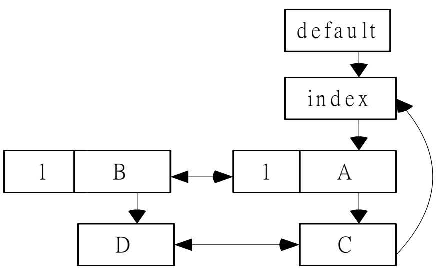

Figure 3: Web structure

When constructs the Web structure model, we starts from the main entrance of a web, analyzes the entire website structure in order. Basically, [8] we use a place in STPN to indicates a Web frame, and use a timed transition in STPN to indicates a hyperlink in a frame, finally we get a adjacency matrix [Cij]mxn (Table II), which represents a Web structure. The Stochastic Timed Petri Nets corresponding to the Web structure of Table II is shown in Figure 4.

TABLE II.
ADJACENCY MATRIX

	T0	T1	T2	T3	T4	T5	T6	T7
P0	-1	0	0	0	0	0	0	0
P1	1	0*	0*	-1	-1	1	0	0
P2	1	0*	0*	-1	-1	1	0	0
P3	1	-1	1	-1	0	1	0	0
P4	0	1	-1	0	-1	0	0	0
P5	0	0	0	1	0	-1	-1	1
P6	0	0	0	0	1	0	1	-1

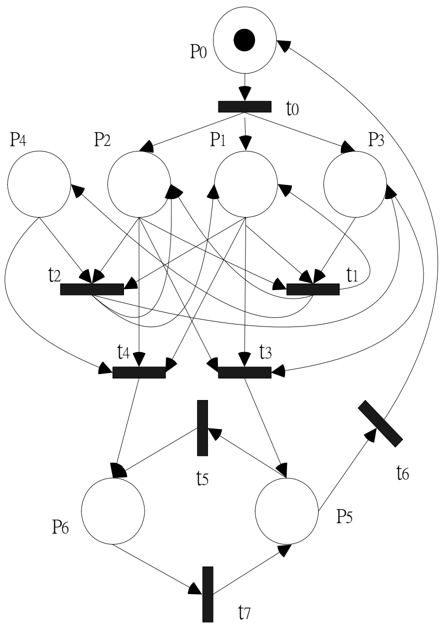

Figure 4 The Stochastic Timed Petri Nets corresponding to the Web structure of Table II

In really analyzes and makes the Web structure using STPN ,we represent a STPN status by [M]nx1, and represent the relationship between Web page (frame) and hyperlink by adjacency matrix [Cij]mxn.

In table II, A matrix element value 0* indicates a situation which a place has been obtained a token after the same place has been lose a token by a transition.

We start from the main entrance of a web, in order analyzes the entire website structure, to construct a Web structure model. We must pay attention to analyze a Web structure by breadth first method. This method is extremely important for asking the accuracy of the linear equation solution.

D. Relate Research of PN in Web Services

Many researchers apply PN in the Web Services. Thomas [13] analyzes the Web Services Description Language (WSDL) and constructs the related PN model. They also use the model to simulate the transitions of message and transaction. Rachid [14] uses Petri Net-Based algebra to simulate control flow of Web Service Composition. Lisa [15] implements a Coloured Petri Nets (CPN) to analyze the performance of HTTP Web Server. David [16] employs the PN model to assay the relationship among hyperlinks and observe the behaviors of users.

III. Pageview Identification

A pageview represents a display comprising all the Web pages 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 STPN to model a website structure and proceed with pageview identification with the help of an incidence matrix. After sorting out the procedures of data cleaning, session identification, the log file usually records the user IP, request time, request file and refers.

Since we can locate the use's current and subsequent pageview through request file URL, we can also find out which file the user adopts to click on the hyperlink so as to posit the request of webpage service. As a result, we exploit one user request record to obtain the following three various data:

- Refer Pageview (RPV) ：RPV represents the set of pageview which the user has browsed.
- Source Pageview (SPV)： SPV represents the set of pageview which the user is browsing.
- Destination Pageview (DPV)：DPV represents the set of pageview which the user may subsequently browse.

By the way of the using of RPV, SPV and DPV, we can then filtering those complicated Web content scope set between two neighboring Web session records. We represent these data set relationships below. Figure 5 indicates a triggered transition:

{RPV}→{SPV}→{DPV}

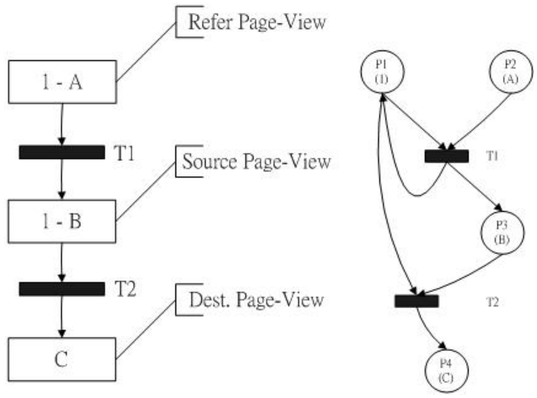

Figure 5: Web structure and STPN structure

B. Pageview Identification Algorithm

Figure 6 illustrates the algorithm of pageview identification. In pageview identification, we subsequently use each count of the user record to locate the possible pageview and then fill the count of the pageview into the pageview identification table. However, when the request file in the user records comprises the file belonging to the previous pageview, this indicates the request file belongs to another complicated pageview. To ensure which pageview it belongs to, we retrieve the next count of user records. We cross-examine to locate the only one pageview. But when two counts of user records do not link in sequence, we use the function to fill up the inadequacy of user records through Path_Completion ( ).

```
Input: User_Session , all logs in a user session.
Output: PageViewTable (PVT)
Data Structure:
SPV: a set of places which represent current pageview.
DPV: a set of places which represent next pageview.
WPV: a set of places which represent current calculated pageview.
PVT: a collection of pageview.

PageviewIdentification(User_Session)
{
    Current_Refer = null;
    WPV = null;
    while (!User_Session.EOF()) {
        Current_Log=User_Session.ReadLog();
        WPV = WPV.Add(Current_Log.URL);
        if (Current_Refer == Current_Log.Refer) {
            WPV = WPV.Add(Current_Log.URL);
        }
        else {
            DPV =  Get_DPV(Current_Refer);
            SPV = Get_PageView(WPV, DPV);
            PVT.Add(SPV);
            WPV = null;
            Current_Refer = Current_Log.Refer;
        }
        User_Session.MoveNext();
    }
    if (WPV != null)
```

```
21.     PVT.Add(WPV);
22.}
```

Figure 6: Algorithm of Pageview Identification

The input of the algorithm includes User_Session. User_Session is the log describing the user request record for a certain website in a period of time. Through the process of pageview identification, we can transfer the file from the file request record to more meaningfulness of user pageview record.

The output of the calculation method is PageViewTable which represents contents and the sequence of the data which the user extract the content range from the browser in a certain period of time.

There are two functions in this algorithm, namely Get_DPV and Get_PageView.

- Get_DPV:

This function will use the keyed-in count of the pageview to locate all of the possible subsequent count of the pageview.

- Get_PageView:

This function will read two sets of pageview and compare them. It will return the match pageview.

C.Complexity of Pageview Identification

If we provide the following defintions:

- US= number of logs in a user session
- AW=Average number of webpages in one pageview
- AL=Average number of Link in one webpage

The complexity of Pageview Identification is (US*AW*AL).

D.Example of Pageview Identification

Taking a user profile as an example as shown in Table III, we observe from the incidence matrix, when the system retrieves the first usage profile having a transition identification (TID) as 1, the transition related to P0, hence, consists of T0 only due to that the place corresponding to Default.htm is P0 and that [A0j]=(-1, 0, 0, 0, 0, 0, 1, 0); we can recognize that T0 indicates the transformation of the pageview comprising Index.htm, 1.htm, and A.htm from the pageview containing Default.htm only, for [Ai0]=(-1, 1, 1 ,1, 0, 0, 0). The first, second and third usage profiles in Table IV-1 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 T1 stands for the transition to Index.htm, 1.htm, and B.htm, and T3 the transition to C.htm as there are only two transitions, T1 and T3, related to Index.htm, 1.htm and A.htm; the request webpage of the fifth profile is B.htm. Hence, we can indicate that the fifth profile represents the user's entering the pageview comprising Index.htm, 1.htm and B.htm. Based on such an 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 IV.

TABLE III.
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

TABLE IV.
A USER SESSION AFTER PAGEVIEW IDENTIFICATION

Pageview ID	Request file
0	Default.htm
1	Index.htm
1	1.htm
1	A.htm
2	Index.htm
2	1.htm
2	B.htm

IV. PATH COMPLETION

In many cases, the Web log will lose some Web page file access records. For example, some users may used to click "back" button to go back to a previous Web content scope. On the other hand, a proxy server will make the same cache actions like "back" button. At these cases, the browser will get some cached Web files which are already saved in computer or proxy server before. Then the cached access event won't appear in a Web log. Moreover, if a user click back button time and time, there are many access records will lost in a Web log.

In these cases described above, we might utilize STPN's reachability behavior characteristic. To each pair in STPN status which is not connected by a transition directly, we can use reachability linear equation, and adjacency matrix to obtain some unknown, continual STPN status set and triggered transition set in order. These sets can help us to fill STPN status which between these two missed STPN status. Then it's an opportunity for us to use Path complete method.

A. Using Reachability to Assist Path Completion

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

Path completion needs should thus be activated to patch it once the Web usage logs are found incomplete during the pageview identification process. We will classify the problems of path completion into two categories. They are the lost single user-record and the lost multiple user-record. The method they use to deal with is described as follows:

- Lost Single User Records

Since the "refer" field in the Web log record keeps records the file location of the user requests. When the user record only lacks one count of data, we can use URL in Current_Log to locate the SPV of Current_Log, and use the item of refer to locate DPV of Current_Log.Refer. Eventually, we cross-examine the both data so as to rapidly locate the inadequate user records.

- Lost Multiple User Records

When the web log lost more than one count of user records, we cannot simply use refer to locate the lost path. Therefore, when facing such a problem, we will use the reachability property mentioned in section II to find out the lost path. We first calculated the state in the previous log and set it at M0. We then calculated the state, Md, in the next log. According to Equation (1), we use M0, Md and incidence matrix Cij to calculate matrix [Tj] representing the possible path of two logs. Finally, we further use [Tj] to locate the possible path of two logs and then fill the potential paths with relevant pageview identification table.

B. Path Complete Algorithm

Figure 7 illustrates the algorithm of path completion. We can first use the above-mentioned method to cross-examine the data. If the overlap between DPV and DPV of SPV is null, then it represents the lost record data of the two records can be more than one count. To tackle this problem, we further locate the lost record by using the reachability property in the STPN model.

```
Input: PageViewTable (PVT)
Output: PageViewTable (PVT)
Data Structure:
SPV: a set of places which represent current pageview.
DPV, DPV1: a set of places which represent next pageview.
PVT: a collection of pageview.

Path_Completion(PVT)
{
   If (PVT != null) {
      SPV = PVT[0];
      i = 1;
      while (i < PVT.count) {
         DPV = PVT[i];
         DPV1 = Get_DPV(SPV);
         bHasIntersect = CheckIntersect(DPV, DPV1);
         if  (bHasIntersect == false) {
            STPN_Reachability(PVT, SPV, DPV);
         }
         SPV = DPV;
         i = i + 1;
      }
   }
}
```

Figure 7: Algorithm of Path Completion

The input of the algorithm includes PageViewTable. There are three functions in this algorithm, namly Get_DPV, CheckIntersect, STPN_Reachability.

- Get_DPV:

This function will use the keyed-in count of the pageview to locate all of the possible subsequent count of the pageview.

- CheckIntersect:

This function will cross-calculate according to the logged-in DPV and DPV1. If both of them have the overlap, then the true value will be transmitted back. If their overlap is null, then the false value will be transmitted back.

- STPN_Reachability:

This function will use the reachability property in the STPN model to locate the reachable path for the two pageviews, and fill the related data of the path into Pageview Table.

C. Complexity of Path Completion

The complexities of path completion include two parts. There are "CheckIntersect", and "STPN_Reachability". If we provide the following defintions:

- F=Web page number
- PV=Pageview number of Pageview Table
- AW=Average number of webpages in one pageview
- AL=Average number of Link in one webpage
- M=Place number =F

The complexity of "CheckIntersect" is (AW*AL). We use the matrix function in MATLAB to implement the STPN_Reachability. The complexity of the matrix function is approximately [(4*F*F*F/3)] [17]. Therefore the worst-case complexity of "Path Completion" is [(4*F*F*F/3)*PV]. The best-case complexity of "Path Completion" is [AW*AL*PV].

D. Example of Path Completion

Taking the user session in Table III 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 M0= [1, 0, 0, 0, 0, 0, 0]T and the destination state marking Md=[0, 0, 0, 0, 0, 0, 1]T. According to (1), the equation is illustrated as in Figure 8.

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

Figure 8: The State Equation of Path Complete

We can also obtain the [Tk]=[1, 1, 0, 0, 1, 0, 0, 0]T, which means that the initial state can be transferred to destination state through continuous fired in T0, T1 and T4. To take one step ahead, we can find the firing sequence of M0 to Md is T0->T1->T4 in the assistance of the incidence matrix.

V. Conclusions

The significant contribution of the present dissertation is that we use STPN Model as website structure model. We simultaneously provide functions and properties to help tackle the current problems in the process of Web Usage Mining. Further, we calculate the use proportion of each pageview in the website as references for the design of website mangers and their modification. We propose the use of Stochastic Timed Petri Nets as a webpage structure model for website simulation, and demonstrate that using this model, we can not only adopt Stochastic Timed Petri Nests' incidence matrix to help carry out pageview identification but also utilize Stochastic Timed Petri Nests' reachability property to fulfill path completion.

In the future research, the present dissertation proposes the use of STPN model for the construction of website structure model. We explicate how to precede data preprocessing and pattern discovery in details after establishing the model. In the further research in the future, we hope to use the model to construct some basic pattern analysis as follows:

- In the process of data preprocessing, we save the duration the user remains on the website, and increase the time indicator, which allows us to grasp the importance of user's application of each webpage after analyzing the user property and behaviors.

- In the process of data preprocessing, we save the file amount and increase the file amount indicator, which allows us to analyze the use sequence and frequency. The analytic outcome can be used as references for deploying the website content.

References

[1] Sanjay Madria, Sourav S Bhowmick, W. –K Ng, E. P. Lim, "Research Issues in Web Data Mining", In proceedings of Data Warehousing and Knowledge Discovery, First International Conference, DaWak, '99, 1999, pp.303-312.

[2] Jose Borges and Mark levene, "Data Mining of User Navigation Patterns", In Proceedings of the WEBKDD'99, 1999, pp. 31-39.

[3] Raymond Kosala and Hendrik Blockeel, "Web mining research: A survey", SIGKDD Explorations, Vol.2, Issue 1, pp1-15, July 2000.

[4] 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, ppP.93-116, May 2003.

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

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

[7] Magdalini Eirinaki and Michalis Vazirgiannis, "Web Mining for Web personalization.", ACM Transactions on Internet Technology, Vol. 3, No. 1, Feb. 2003, pp.1-27. Tadao Murata, "Petri Nets: Properties, Analysis and Applications", Proceedings of the IEEE, Vol. 77, No. 4, 1989.

[8] Shih-Yang Yang , Po-Zung Chen, Chu-Hao Sun, "Using Petri Nets to enhance web usage mining", Acta Polytechnica Hungarica, 4, 3, 2007, pp. 113-125, ISSN 1785-8860

[9] Yan Wang, "Web Mining and Knowledge Discovery of Usage Patterns"

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

[11] W. Reisig, "Correctness Proofs of Distributed Algorithms", Lecture Notes in Computer Science, Vol. 938：Theory and Practice in Distributed Systems, pp.164-177, 1995.

[12] M. Ajmone Marsan, "Stochastic Petri Nets: An Elementary Introduction", Lecture Notes in Computer Science, Vol. 424：Advances in Petri Nets 1989, pp. 1-29, 1990.

[13] Johnson P. Thomas, Mathews Thomas, and George Ghinew, "Modeling of Web Services Flow", IEEE International Conference on E-Commerce(CEC'03), June, 2003.

[14] Rachid Hamadi and Boualem Benatallah, "A Petri Net-based Model for Web Service Composition", In Preceedings of the Fourteenth Australasian database conference on Database technologies, 2003, p.191-200, February 01, 2003, Adelaide, Australia.

[15] Lisa Wells S. Christensen, Lars M. Kristensen, and Kjeld H. Mortensen, "Simulation Based Performance Analysis of Web Servers", In Proceedings of 9th International Workshop on Petri Nets and Proformance Models, Sept. 11-14, pp 9-68, 2001

[16] P. David Scotts, Richard Furuma, "Petri-net-based hypertext: document structure with browsing semantics", ACM Transactions on Information Systems, Volume 7, Issue 1, pp.3-29, January 1989.

[17] Peter Linz, Richard L. C. Wang, "Exploring Numerical Methods: An Introduction to Scientific Computing Using MATLAB", Jones and Bartlett Publishers, Inc., 2003.

Po-Zung Chen received the Ph.D. degree in Computer Science from the University of Iowa in December 1989. From November 1989 to May 1990, he was a visiting Assistant Professor at Michigan Technological University (Houghton, Michigan). Since August 1990, he is an Associate Professor with the Department of Computer Science and Information Engineering at Tamkang University (Taipei, Taiwan). His research interests include object-oriented distributed programming, parallel & distributed systems, and simulation & modeling.

Chu-Hao Sun is currently a candidate of Ph.D. student in department of Computer Science and Information Engineering in Tamkang University (Taipei, Taiwan). He received his B.E. and M.E. degree from the same university in 1995 and 1998. His research interests include database management system, parallel processing, web technology, and data mining.

Shih-Yang Yang received the Ph.D. degree in Computer Science and Information Engineering from Tamkang University, Taiwan, in January 2008. Since January 2008, he is an Associate Professor with the Department of Visual Communication and Animation Art at Kang-Ning Junior College of Medical Care and Management (Taipei, Taiwan). His research interests include parallel & distributed systems, web technology, and multimedia.