

# Online Analytical Mining of Path Traversal Patterns for Web Measurement

Joseph Fong and H K Wong

Department of Computer Science

City University of Hong Kong, Hong Kong

Email: [csjfong@cityu.edu.hk](mailto:csjfong@cityu.edu.hk), [hkwong@cs.cityu.edu.hk](mailto:hkwong@cs.cityu.edu.hk)

**ABSTRACT** The WWW and its associated distributed information services provide rich world-wide online information services, where objects are linked together to facilitate interactive access. Users seeking information of Internet traverse from one object via links to another. It is important to analyze user access patterns which will help improve web pages design by providing an efficient access between highly correlated objects, and also assist in better marketing decisions by placing advertisements in frequently visited document. We need to study the user surfing behavior through examining the web access log, browsing frequency of web pages and computing the average duration time of visitor. This paper offers an architecture to store the derived web user access paths in a data warehouse, and facilitates its view maintainability by use of a metadata. The system will update the user access paths pattern with the data warehouse by the data operation functions in the metadata. Whenever a new user access path occurs, the view maintainability is triggered by a constraint class in the metadata. The data warehouse can be analyzed on the frequent pattern tree of user access paths on the website within a period and duration. The result is an online analytical mining path traversal pattern. Our experimental and performance studies have demonstrated the effectiveness and efficiency of our system with the following contributions: an architecture of online analytical mining (OLAM) using frame model metadata; a methodology (stepwise procedure) of implementing OLAM and the resultant cluster of web pages frequently visited by users for marketing use.

Keywords: OLAM, web measurement, path traversal patterns, view maintenance

## INTRODUCTION

As the popularity of WWW explodes, a massive amount of data is generated by web servers in the form of web access logs. This is a rich source of information for understanding web user surfing behavior. Web usage mining is one type of web mining activity that involves the automatic discovery of user access patterns on web server(s). On the other hand, it is an application of data mining algorithms to web access logs to find the trends and regularities in web users' traversal patterns.

Analysis of these access data can provide useful information for server performance enhancements, restructuring website, and direct marketing in electronic commerce. As a result, web usage mining has been widely used in improving website design, business and marketing decision support, user profiling, and web server system performance, etc.

Among discovering various kinds of knowledge in large databases, mining association rule has attracted

great attention in database research communities in recent years (Agrawal, Imielinski, & Swami, 1993; Agrawal, & Srikant, 1994; Miller, & Yang, 1997; Srikant, & Agrawal, 1995; Savasere, Omiecinski, & Navathe, 1995). Association rule mining is a form of data mining to discover interesting relationships among attributes in those data. The discovered rules may help marketing decision support, and business management. Association rules have two important measurements: Support and Confidence. Support is an argument that decides whether the candidate is frequent or not. Confidence is an argument that describes the believable degree of association rules.

The Frequent Pattern Growth (FP-growth) algorithm is one of the association rule algorithms to find frequent itemsets, but unlike Apriori, it avoids the expense of candidate generation by generating only candidate itemsets. Because FP-growth does not need to examine both candidate and non-candidate sets and requires only two scans of the database, it is a fast algorithm for mining association patterns. We will study this algorithm in depth in our proposed algorithm that is called Sequential FP-growth.

In this paper, we propose and develop an interesting method that is called online analytical mining of path traversal patterns, which integrates the recently developed data warehouse technology with efficient association mining methods. The system stores the derived web user access paths in a data warehouse, and facilitate its view maintainability by use of a frame metadata. The system will update the user access paths pattern with the data warehouse by the data operation functions in the frame metadata. Whenever a new user access path occurs, the view maintainability is triggered by a constraint class in the frame metadata. The data warehouse can be analyzed on the frequent pattern tree of user access paths on the website within a duration. The developed method achieves incremental, extensible, and multi-dimensional association rule mining with high performance.

## **RELATED WORK**

### **Association Rules Discovery**

The concept of association rules was first introduced in (Agrawal, Imielinski, & Swami, 1993). Since then, the problem of data mining for association rule has been studied extensively (Agrawal, & Srikant, 1994; Cheung, Han, Ng, & Wong, 1996; Han, Karypis, & Kumar, 1997; Park, Chen, & Yu, 1995; Savasere, Omiecinski, & Navathe, 1995; Svawagi, Thomas, & Agrawal, 1998). These studies covered a broad range of topics with variations studied, and aimed for further improvements of the performance of the algorithm. For example, fast algorithms based on the Apriori Algorithm (Agrawal, & Srikant, 1994), incremental updating and parallel algorithms (Cheung, Han, Ng, & Wong, 1996; Han, Karypis, & Kumar, 1997; Park, Chen, & Yu, 1995), mining of generalized, multi-level rules, and multi-dimensional rules (Zhao, Deshpande, & Naughton, 1997).

### **Sequential Patterns Mining**

The problem of discovering sequential patterns mining is to find inter-transaction patterns such that the presence of a set of items is followed by another item in the time-stamp ordered transaction set. It was first introduced by (Agrawal, & Srikant, 1995). The algorithm AprioriAll was the one that found all frequent

patterns. Later, the same authors (Srikant, & Agrawal, 1996) presented the GSP algorithm that outperforms AprioriAll by up to 20 times. The GSP algorithm they proposed was a variation of the Apriori algorithm in which candidate sequences are stored in a hash-tree.

Much more work has been done in user behavior analysis. In (Chen, Park, & Yu, 1998), methods were explored to mine path traversal patterns in a distributed information environment, but only one ordered dimension of the forward referenced pages/URLs accessed was considered. PSP (Masseglia, Cathala, & Poncelet, 1998) was developed to improve the way in which GSP stored candidate patterns. PSP created a prefix-tree, where any branch from its root to the leaf, stands for a candidate sequence, and the leaf node provides the support of the sequence. A prefix-tree structure costs less in term of memory as it organizes candidates according to their common elements. FreeSpan (Pei, Han, Mortazavi-Asl, & Zhu, 2000) was developed to substantially reduce the expensive candidate generation and testing of Apriori, while maintaining its basic heuristic. PrefixSpan (Pei, Han, Mortazavi-Asl, Pinto, Chen, Dayal, & Hsu, 2001) was developed to address the costs of FreeSpan. The idea is that, instead of projecting sequence databases by considering all the possible occurrences of frequent subsequences, the projection is based only on the frequent prefixes that greatly reduce the efforts of candidate subsequence generation. SPADE (Zaki, 2001) is a fundamentally different sequential pattern algorithm. In place of repeated database scans, this method uses lattice-search techniques and simple join operations to discover all sequence patterns.

### **Web Usage Mining**

In the recent years, there has been an increasing number of research work done in web usage mining (Chen, Park, & Yu, 1998; Wu, Yu, & Ballman, 1998; Buchner, Baumgarten, Anand, S.Mulvenna, & Hughes, 1999; Cooley, Mobasher, & Srivastava, 1999; Masseglia, Poncelet, & Cicchetti, 1999; Srivastava, Cooley, Deshpande, & Tan, 2000). Web usage mining is the type of web mining activity that involves the automatic discovery of user access patterns from one or more web servers. The main motivation of these studies is to get a better understanding of the reactions of customers who shop through electronic premises of a company in WWW or of users who are simply browsing these premises. Some studies also applied the mining results to improve the design of web sites, analyzed system performance and network communications or even built adaptive websites. In general, there are two main goals in the application of the discovered knowledge in web usage mining: general access pattern tracking for understanding access patterns and trends and customized usage tracking for adapting and personalizing the browsing experience for the users. The former is the aim of this paper.

Our proposed system, online analytical mining of path traversal patterns differs from the others because it includes scalable, continuously and incrementally web usage mining and the integration of the data mining with database systems and data warehouse systems.

### **Frame Model Metadata**

(Fong, & Huang, 1997) translated existing data models into a frame model of the universal database. The structure of frame model consists of several classes such as Header, Attributes, Methods, and Constraints

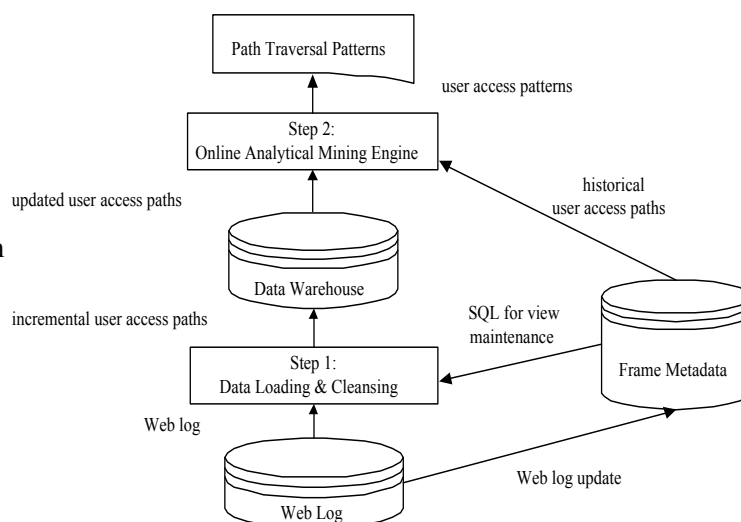
classes. According to the frame model, a universal database could be formed. As a result, the old and the new database systems could coexist to form a data warehouse for a decision support system.

(Fong, & Huang, 1999) investigated the architecture of a universal data warehousing for the connectivity of relational and OO data model using an ORDBMS. A frame model metadata was chosen to represent the conceptual and the logical schema of a universal data warehouse, which structures an application domain into classes, and its data in relational tables. The universal data warehouse using an ORDBMS offers a relational and an OO view for the data warehouse to accommodate different types of queries efficiently. (Fong, & Pang, 1999) proposed a frame metadata model approach to integrate existing databases and evolved them to support new database applications. This facilitates an evolutionary approach to integrate existing databases to support new applications.

### GENERAL ARCHITECTURE OF OLAM

We have developed a general architecture for Online Analytical Mining of Path Traversal Pattern on web usage, which is presented in (Fong, Wong, & Fong, 2000). The overall architecture for the web usage mining process is depicted in Figure 1.

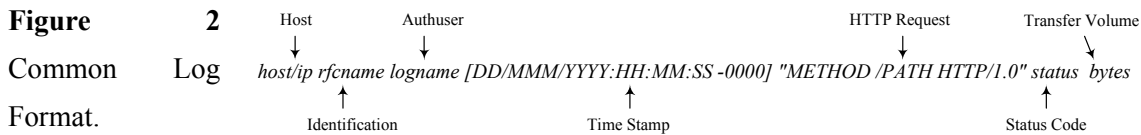
**Figure 1**  
General Architecture of OLAM on  
User Access Patterns.



The architecture divides the web usage mining process into two main parts. Firstly, the data collected from the web log goes through two steps. In the first step of data preprocessing, data loading and data cleansing, data is filtered to remove irrelevant information (i.e., server request failures, authentication failures, etc.). Then, all entries of the log were mapped one by one into a relational database. After the data was cleansed, the web log is loaded into data warehouse and new implicit data like frequency occurrence of access paths and the time spent by each visitor on each page is calculated. Also the database facilitates information extraction and data summarization based on the individual attributes. In the second step, web mining techniques are used to predict and discover interesting user access paths. After the initialization of loading the web log into the data warehouse, whenever a user access path is recorded in web log file, a corresponding update will be made to the frame metadata, which triggers the update of user access patterns of web pages online, and generates path traversal patterns.

## Web Access Log

An important source of information about website visitors is server transfer log known as the access log. This is where every transaction between the server and browser is recorded with a date and time, the IP address of the server making the request for each page on the site, the status of that request, and the number of bytes transferred to that requester, etc. We can analyze users' activities on a website using web servers' access log. There are several kinds of log format. The most popular one is Common Log Format, which was used by most web servers. The common log format appears exactly in Figure 2.



### Example: Raw Data of the Access Log

```
144.214.121.52 - - [31/Mar/2001:20:38:11 +0800] "GET /an_cityu.gif HTTP/1.1" 200 90713
```

```
144.214.121.52 - - [31/Mar/2001:20:39:31 +0800] "GET /Courses.htm HTTP/1.1" 200 1213
```

### Step 1: Date Preprocessing

One of the important core steps of knowledge discovery is data preprocessing. Since not all the materials within the access log are useful for the mining process, a data preparation process must be performed first. After the data cleaning, the log entries must be partitioned into logical clusters using one or many transaction identification modules, which includes user and session identifications.

#### Step 1.1: Data Loading and Cleaning

A large proportion of the access log is related to graphics, and pictures that constitute the pages and provide no information on the usage of the website. Thus, all the log entries with the picture filename suffix in the access path field will be removed. Moreover, those records with the methods other than using "GET" (i.e. "PUT", "POST", "HEAD") in the access method field to access the specified file will be eliminated. Also, it needs to separate the access time field because separating them will make it easier to calculate the time for staying on each page. Finally, those records with the status value other than 200 (successful file retrieval) will be eliminated. Figure 3 shows the pseudo code for data preprocessing.

```

begin
open access log;
do until end of file
read a line from the log file;
if((access_method = "get" or "GET") and (status = 200))
if (path = ".htm" or path = ".html") // exclude the graphics, pictures
use space as a separator to identify the field and determine array size
do {
for (int i = 0; i <= array size; i++) {
if (it is scanning the last field)
store it to the array's last element;
else {
if (the field starts with "[") { // Handling the time
// format
get the length of this field;
check the first colon occurrence from "[" position;
store the content from the character after this
character up to the colon character to array;
i = i + 1;
store the content from the position of the colon
character to the end of this field to array;
reset the colon index value to 0;
}

// For handling the case of "+0800"
else if (the field end with "]") {
storing content exclude the character "]" to array;
}

// For handling the "GET" method
else if (the field starts with " " character) {
storing content exclude the character of " " to array;
}

// Handling the case of HTTP/1.0"
else if (the field ends with " " character) {
storing content exclude the character of " " to array;
}
else {
storing field content fully to the array;
}
} // else
update field position value;
} // end for
} // end do

```

**Figure 3**  
Pseudo Code for Data Preprocessing.

After the removal of all the irrelevant records from the web log file, the valid records will be stored in the main table as shown in Table 1.

IP Address	Date	Time	URL Request
144.214.36.91	07/May/2001	22:42:04	A.htm
144.214.36.91	07/May/2001	22:45:06	B.htm
144.214.36.91	07/May/2001	22:49:15	D.htm
144.214.36.91	07/May/2001	22:52:44	E.htm
144.214.36.91	07/May/2001	23:40:00	B.htm
144.214.36.91	07/May/2001	23:42:00	A.htm
144.214.36.92	07/May/2001	23:43:05	A.htm
144.214.36.92	07/May/2001	23:46:06	B.htm
144.214.36.92	07/May/2001	23:47:30	C.htm
144.214.36.93	07/May/2001	23:47:50	E.htm
144.214.36.93	07/May/2001	23:48:15	C.htm

**Table 1**  
Cleaned Web Log Data Stored in Main Table.

### Step 1.2: User Identification and Session Identification

We use the host name incorporated with user session to identify a user. A user session is all pages references made by a user during a single visit to a site. The user interacts within a web site as a collection of user session whose information is logged in web server log. Thus, a user session can be inferred from a web log, which represents a sequence of requests made by the user within a defined time interval. We use thirty minutes as time interval between requests within a user session (Catledge, & Pitkow, 1995). Figure 4 illustrates the inferred user sessions from Table 1.

**Figure 4**  
User Navigation  
Session Inferred  
from Cleaned Web  
Log.

IP Address	URL Request	Time of the Request	X = 30 minutes
144.214.36.91	A.htm	07/May/2001:22:42:04	↑ Session 1 ↓
144.214.36.91	B.htm	07/May/2001:22:45:06	
144.214.36.91	D.htm	07/May/2001:22:49:15	
144.214.36.91	E.htm	07/May/2001:22:52:44	
144.214.36.91	B.htm	07/May/2001:23:40:00	↑ Session 2 ↓
144.214.36.91	A.htm	07/May/2001:23:42:00	
144.214.36.92	A.htm	07/May/2001:23:43:05	↑ Session 3 ↓
144.214.36.92	B.htm	07/May/2001:23:46:06	
144.214.36.92	C.htm	07/May/2001:23:47:30	
144.214.36.93	E.htm	07/May/2001:23:47:50	↑ Session 4 ↓
144.214.36.93	C.htm	07/May/2001:23:48:15	

Session 1: A.htm → B.htm → D.htm → E.htm  
 Session 2: B.htm → A.htm  
 Session 3: A.htm → B.htm → C.htm  
 Session 4: E.htm → C.htm

### Step 1.3: Data Warehousing

After finishing the data preprocessing, we remove all the irrelevant records (i.e., graphics, pictures, server request failures, authentication failures, etc.) from the web log and identify user and session identifications as described above. All the cleaned data are stored in the main table for further process. We can then store the web usage in a data warehouse such that the log of accessing the target web page and its previous web pages will be analyzed as traversal patterns. These web pages of the user access paths record will be stored in the fact table of the data warehouse with their dates stored in the dimension table. The algorithm for recording user access paths into a data warehouse is shown in Figure 5.

**Figure 5**  
Algorithm for Recording User  
Access Paths into Data  
Warehouse.

---

Given: materialized view V, auxiliary relations  $V_1, \dots, V_n$ , data to be updated  $\delta R$  into data warehouse view and data warehouse view  $V'$  after update.

```

begin
  for record added in log
    extract desired data fields and map into main table;
    if access path exists
      then increment the frequency pattern by 1;
    else
      add the new user access path into fact table;
    end if
  end for

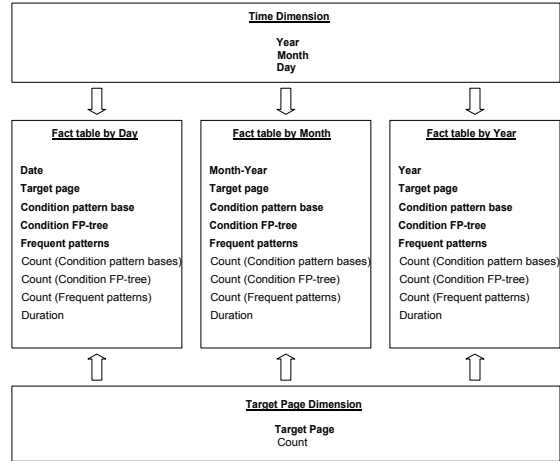
  // V' = V + Applied Group by on  $\delta R'$  with Aggregate
  // count by re-computing total and aggregate count
  if  $\delta R$  comes from updates to fact table destination relation
    then  $V' = V \cup \delta R'$ ;
  end if
end

```

---

Figure 6 shows the star schema of web usage in an access path for an interval in a period. In general, for any user with an UID or IP address, there are many navigation paths that the user browsed the website. For example, if the access path is P1, P2 and P3 in sequential order, its web page access path becomes from P1 to P2 to P3. Notice that frequency pattern count is the number of browsed frequency of the path.

**Figure 6**  
Star Schema of Frequency Pattern Count.



We can make use of the attribute event in the constraint class of the frame model metadata to automate the data warehouse data cube continuously and incrementally. For example, given the dimension table and the fact table are:

Year	Month	Day
Year1	Month1	Day1

Target Page	Count
T1	C1

Target Page	Date	CPB	CFP	FP	Count(CPB)	Count(CFP)	Count(FP)	Duration
T1	Date1	Path 1	Path 2	Path 3	C1	C2	C3	D1

Target Page	Month	CPB	CFP	FP	Count(CPB)	Count(CFP)	Count(FP)	Duration
T1	Month1	Path 1	Path 2	Path 3	C1	C2	C3	D1

Target Page	Year	CPB	CFP	FP	Count(CPB)	Count(CFP)	Count(FP)	Duration
T1	Year1	Path 1	Path 2	Path 3	C1	C2	C3	D1

To be updated dimension table tuple  $\delta R$  (data to be updated to data warehouse)

Year	Month	Day
Year2	Month2	Day2

Target Page	Count
T2	C2

To be updated fact table update  $\delta R$  (data to be updated to  $R_{FACT}$ )

Target Page	Date	CPB	CFP	FP	Count(CPB)	Count(CFP)	Count(FP)	Duration
T2	Date2	Path 4	Path 5	Path 6	C4	C5	C6	D2

Target Page	Month	CPB	CFP	FP	Count(CPB)	Count(CFP)	Count(FP)	Duration
T2	Month2	Path 4	Path 5	Path 6	C4	C5	C6	D2

Target Page	Year	CPB	CFP	FP	Count(CPB)	Count(CFP)	Count(FP)	Duration
T2	Year2	Path 4	Path 5	Path 6	C4	C5	C6	D2

If  $T1 = T2$ ,  $Date1 = Date2$ ,  $Path 1 = Path 4$ ,  $Path 2 = Path 5$ , and  $Path 3 = Path 6$ , then  $R_{FACT}$  become:

Target Page	Date	CPB	CFP	FP	Count(CPB)	Count(CFP)	Count(FP)	Duration
T1	Date1	Path 1	Path 2	Path 3	C1+C4	C2+C5	C3+C6	D1+D2

If they are not equal, it can be simply inserted the new records in the fact table directly.

Target Page	Date	CPB	CFP	FP	Count(CPB)	Count(CFP)	Count(FP)	Duration
T1	Date1	Path 1	Path 2	Path 3	C1	C2	C3	D1
T2	Date2	Path 4	Path 5	Path 6	C4	C5	C6	D2

If  $T1 = T2$ ,  $Month1 = Month2$ ,  $Path 1 = Path 4$ ,  $Path 2 = Path 5$ , and  $Path 3 = Path 6$ , then  $R_{FACT}$  become:

Target Page	Month	CPB	CFP	FP	Count(CPB)	Count(CFP)	Count(FP)	Duration
T1	Month1	Path 1	Path 2	Path 3	C1+C4	C2+C5	C3+C6	D1+D2

If they are not equal, it can be simply inserted the new records in the fact table directly.

Target Page	Month	CPB	CFP	FP	Count(CPB)	Count(CFP)	Count(FP)	Duration
T1	Month1	Path 1	Path 2	Path 3	C1	C2	C3	D1
T2	Month2	Path 4	Path 5	Path 6	C4	C5	C6	D2

If  $T1 = T2$ ,  $Year1 = Year2$ ,  $Path 1 = Path 4$ ,  $Path 2 = Path 5$ , and  $Path 3 = Path 6$ , then  $R_{FACT}$  become:

Target Page	Year	CPB	CFP	FP	Count(CPB)	Count(CFP)	Count(FP)	Duration
T1	Year1	Path 1	Path 2	Path 3	C1+C4	C2+C5	C3+C6	D1+D2

If they are not equal, it can be simply inserted the new records in the fact table directly.

Target Page	Year	CPB	CFP	FP	Count(CPB)	Count(CFP)	Count(FP)	Duration
T1	Year1	Path 1	Path 2	Path 3	C1	C2	C3	D1
T2	Year2	Path 4	Path 5	Path 6	C4	C5	C6	D2

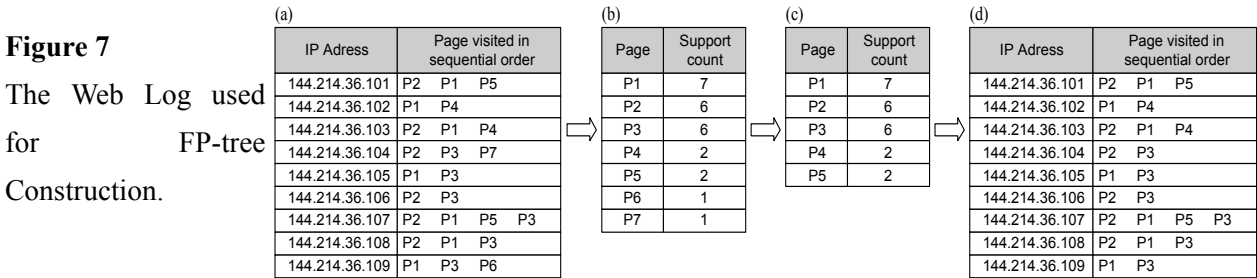
In order to get sequential pattern, if the analyzer wants to analyze the web usage within an interval, he/she can repeat the analysis on different dates. After accumulating the navigation paths of the analysis, the result is sequential patterns within a period.

**Step 2: Online Analytical Mining Engine**

Online analytical mining engine is the major component of the path traversal patterns mining system. The detailed algorithm will be discussed in the following.

**Step 2.1: Sequential Frequent Pattern Growth**

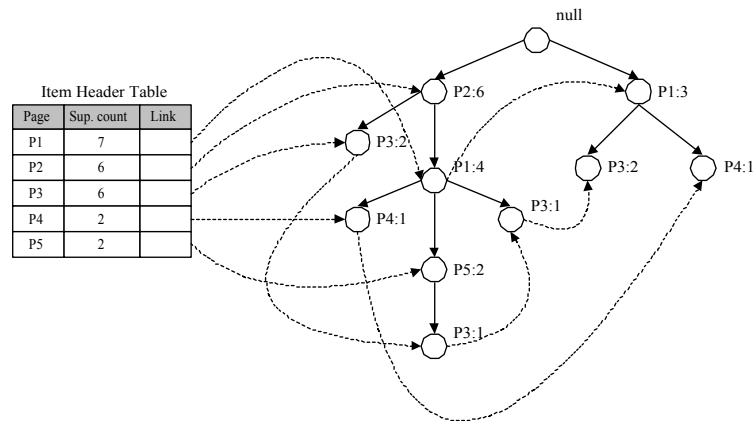
Given a web access pattern database in Figure 7, which contains a set of pages visited in sequential order, we assume the minimum support threshold is 2. Firstly, the database is scanned to derive a list of frequent items and the occurrences of these items. Remove all the items that do not satisfy the minimum support threshold. The resulting set or list is denoted L. Thus, we have  $L = [P1:7, P2:6, P3:6, P4:2, P5:2]$ . Then, the algorithm creates a tree with a root named ‘null’. Next, it scans the database again. The items in each transaction are processed in L order and a branch is created for each transaction.



For example, a scan of the first transaction, “144.214.36.101: P2, P1, P5” contains three visited pages (P2, P1, P5). It leads to the construction of the first branch of the tree with three nodes:  $\langle (P2:1), (P1:1), (P5:1) \rangle$ , where P2 is linked as a child of the root, P1 is linked to P2, and P5 is linked to P1. The second transaction, “144.214.36.102” contains the visited pages P1 and P4. It leads to the construction of the second branch with two nodes:  $\langle (P1:1), (P4:1) \rangle$ , where P1 is linked as a child of the root and P4 is linked to P1. The third transaction, “144.214.36.103” contains the visited pages P2, P1 and P4, which would result in a branch where P2 is linked to the root. P1 is linked to P2, and P4 is linked to P1. However, this branch would share a common prefix,  $\langle P2 \rangle$ , with the existing path for “144.214.36.101”. Therefore, we increment the count of the P2 node by 1, and create a new node, (P1:1), which is linked as a child of (P2:2). Also we create another new node, (P4:1), which is linked as a child of (P1:1). In general, when considering the branch to be added for a transaction, the count of each node along a common prefix is incremented by 1, and nodes for the items following the prefix are created and linked accordingly. The action in reading transactions and tree construction are iterative processed until the last transaction. The tree obtained after scanning all of the transactions is shown in Figure 8. To facilitate tree traversal, an item header table is built so that each item points to its occurrences in the tree via a chain of node-links. Therefore, the problem of mining frequent patterns in web log is transformed to that of mining the FP-tree. Figure 9 shows the pseudo code for sequential frequent pattern growth algorithm.

**Figure 8**

A FP-tree built from the Web Log.



```

begin
  build table T for 1-itemset;
  scan access log
  while not end of log file
    begin
      check T (each entry);
      if(new candidate)
        add new(candidate I);
        count = 1;
      else
        increment count by 1 (candidate I);
      end if
    end
  end while

  scan table T(each entry)
  while not end of table
    begin
      compare each candidate I(Count, min_sup, min_conf);
      if less than min_sup and min_conf
        ignore candidate I;
      end if
    end
  end while

  build tree F, root named "null"
  scan table T;
  for every candidate I
    call procedure insert_node(F, E);
    build a item header table H for each item;
    build a node-link for each item to point its occurrence in the tree;
  end for
end

```

```

procedure insert_node(S : start node, L : item list)
begin
  if L is null
    return S;
  else if (S(next) is null) or (node n not found)
    add new node(L(n));
    set n(count) to 1;
    S = insert_node(S(n->next), L(next n));
  else
    add 1 to n(count)
    S = insert_node(S(n->next), L(next n));
  end if
end

```

**Figure 9** Pseudo Code of Sequential Frequent Pattern Growth Algorithm.

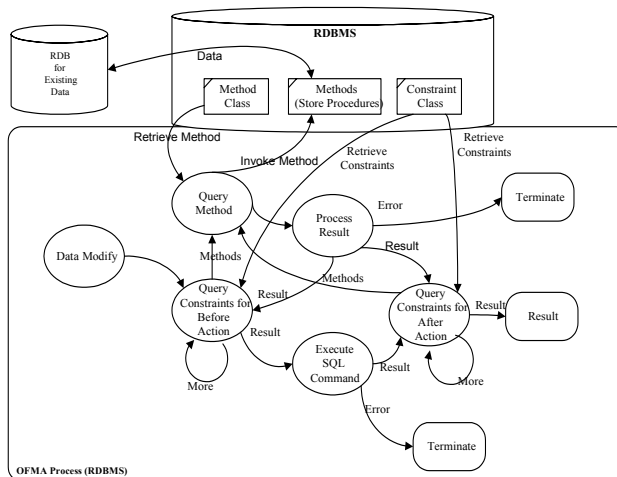
### Step 2.2: Data Warehouse Maintainability using Frame Model Metadata

The frame metadata (Fong, & Huang, 1997) consists of two classes: static classes and active classes. Static classes represent factual data entities and active classes represent rule entities. An active class is event driven, obtaining data from the database when invoked by a certain event. The static class stores data in its own database. With an object frame metadata model agent as shown in Figure 10, frame metadata can be processed with an object-oriented view and data operation functions. When an event occurs, it triggers a process in the constraint class, which calls for the operations in the method class for action. Data can be actively updated to maintain the view for decision support systems. The result is an active data warehousing view maintenance mechanism.

The frame metadata model can be used to implement an event driven active data mining. It suggests continuous data mining association rules as a result of the continuous base relation update.

Base relations are source data that are loaded into RDB tables in the data mining. Incremental data are data of  $\delta R$  which are produced from the source relation in order to update association rules after initial loading. Object Frame Model Agent is an executor, which invokes the constraint class and the method class in the frame metadata model to generate the required SQL transactions for computing association rules statistical figures.

**Figure 10**  
Data Flow of Frame Metadata Model Agent.



To implement the web usage mining for maintaining user access patterns online, we can use frame metadata to update user access paths continuously as follows:

Header class

Class Name	Parents	Operation	Class Type
V	0	Call Insert_path	Active

Constraint class

Constraint Name	Method name	Class Name	Parameter	Ownership	Event	Sequence	Timing
Insert_path	Insert_path	V	$\delta R$	Self	Insert	After	Repeat

Method class

Method_name	Class_name	Parameter	Method_type	Condition	Action
Insert_path	V	$R_s, \delta R$	Tuple	If Code = "GET"	Insert $\delta R$ into $R_s$

Consequently, the minimum support and confidence thresholds value must be specified by the analyst as input parameter to build the frequent tree patterns of user access paths, which in turn will derive the user access patterns (path traversal patterns) after data mining. Support and Confidence are two measures of rule interestingness: Support is an argument that decides whether the candidate is frequent or not. Confidence is an argument that describes the believable degree of association rules. They reflect the usefulness or certainty of discovered rules. Each measure is associated with a threshold that can be controlled by users or domain experts. Rules that do not meet the threshold are considered uninteresting, and hence are not presented to the user as knowledge. A strong association rule will have a large Support and high Confidence level.

## **APPLICATIONS OF OLAM OF PATH TRAVERSAL PATTERNS**

Each query to a web usage mining system returns a set of user navigation patterns. Then the analyst faces the nontrivial problem of evaluating these patterns and deriving reliable conclusions from them. A navigation pattern describes one or more routes among given web pages, along with the statistics on how often each page of each route has been accessed. The patterns and statistics provide rules with which the analyst can determine the output of coincidence. By studying this route closer and comparing it with the other routes crossing it, the web designer can detect pages that are not properly designed or linked and redesign them.

### **Restructuring a Website according to the Mining Results**

Path traversal patterns discovery should help the web designer in improving the design of website. Detecting user navigation paths and analyzing them may result in a better understanding of how users visit a site, identify users with similar information needs, or even improve the quality of information delivery in WWW using personalized web pages.

Also, the sequence of requests by visitors can help predict next requests or popular requests for given days, and thus improve the network traffic by caching those resources, or by allowing clustering resources in a site based on the user motivation.

### **Improving Customization**

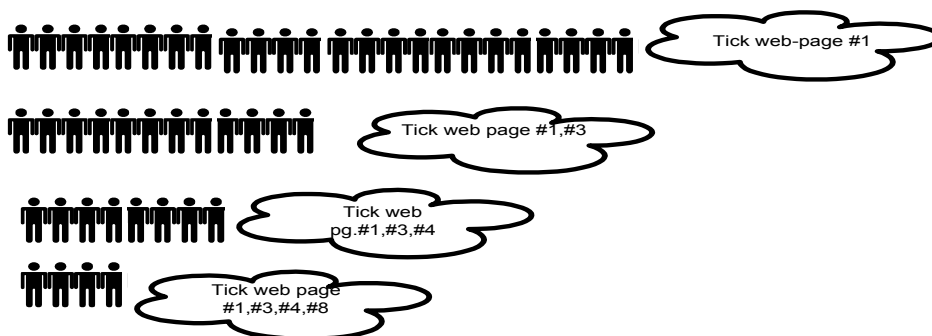
Customization involves learning about an individual user's preferences or interests based on user access patterns. As a result, customization aids in providing users with pages, sites and advertisements that are of interest to them. It may also be possible for website to automatically optimize their design and organization based on observed user access patterns.

### **Impact of Web Advertisements using OLAM of Path Traversal Patterns**

With WWW, it introduces risk for information security but also a huge issue in user's analysis; not because of its vast volume in eyeball count, but because of its random and extreme pattern of click sequence on the company/institution's website. We have therefore embedded the value of Confidence and Support level to accommodate these issues of boundary-less in our OLAM approach. Although user sets these two values that are solely based on heuristic, the criterion of optimality in different business domain should always associate with their expertise knowledge. As such, we restrict the user type within the Sales Management team of a company or the Public Relation team of an institution whose website is undergoing the mining process for increasing sales or promoting company images.

Our objective is that any e-customer should be able to come to the website and complete an e-service process from beginning to end in a user-friendly and intuitively correct manner. We need to encapsulate all our web site surfers' on line experience to discover the knowledge of customer behavior. Our OLAM could create a list of association rules for each targeted web-page determined by the user. The web-pages

click sequences are represented in path traversal patterns. These patterns are analysis to discover user's preference. The preferred web-page(s) could be identified and categorized by Internet surfer and/or e-customers. See below Figure 11.



**Figure 11**  
Identify Target Customers  
by Categorization.

From figure 11, web page #1 is the most frequently accessed web page. Web advertisement may then consider to place on this page. This is the most straightforward way without much need of data mining technique. Our approach offers more. For example, web-page #8 is the function page for registration as clients. All UIDs identified in their click sequence with the visit to web page #8 are grouped as the targeted e-customers. There may have many routes that could link to web page #8. Some users may have actually sent their registration and placed an order/enquiry whereas some may have skipped them. The unsuccessful cases are the target group, that need to be extracted and identified by their UIDs and their specific path traversal patterns are required for further study. The common web-page(s) in all these path traversal pattern that lead to unsuccessful registration are the critical web-page(s) which required the web advertisement, to influence the user behavior. They also target to change their subsequent path traversal patterns to stay on the web page #8 long enough for registration. Those web pages that never led to page #8 could consider to be contracted by revision in the web content, merging, consolidation or even elimination, depending on the individual cases and further studies on the web-page content. Many web pages impressed web surfers with non-focused content or overwhelmed the surfers or e-customers with too much advertising information. Our method could assist in filtering that only mission-critical web pages survived in the ultimate web-site infrastructure. As our targeted result is a list of potential e-customers for a certain product or service on a website, with the associated rules derived, we could trace these related knowledge from further analyzing the main tables in conjunction with the discovered associate rules. We could classify those UIDs by web page sequence. As the key of the main table – identification code tells the UID (User ID), we could identify the target e-customer further or even segment the target e-customers not only by their web-page preference, but also by their gender, occupation type, income range and age group. As such, more customer-oriented web advertisement(s) could be placed in their preferred web page(s) for more effective marketing.

## PROTOTYPE

In the following section, we demonstrate the process of online web usage mining. A university has a home page that contains a lot of useful information (For example, course information, facilities provided, etc.), which is distributed over several sub-pages. The person in-charged wants to know which sub-page is more popular and those visited a particular sub-page intended to visit other sub-pages. Then the person

in-charged can post relevant information or advertisements on the sub-pages more effectively.

The web log file was collected from the Computer Science Laboratory's websites of City University of Hong Kong. The site hosts a variety of information, ranging from department information and department course to individual websites. We are only interested in five pages for analysis. They are:-

Page 1: Department history, facilities and message from department head

Page 2: News, events & seminars notifications

Page 3: Listing of academic staff details

Page 4: Listing of programmes available by department

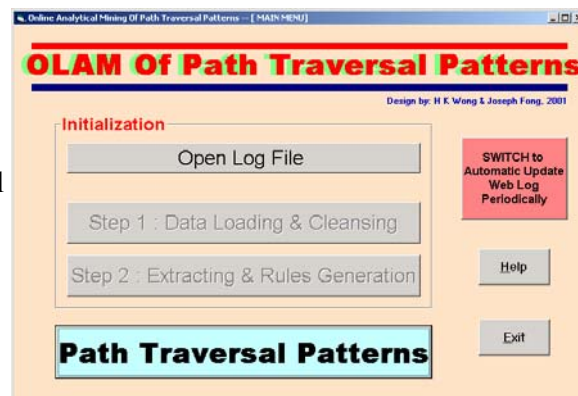
Page 5: Research groups, research projects, publications, etc

For simplification, the above five pages are classify as A, B, C, D, and E respectively.

Figure 11 shows the main menu of the online analytical mining of path traversal patterns. It consists of three parts: initialization, switch to automatic update web log periodically and path traversal patterns. After 'initialization' is executed, a set of potential user navigation paths and some user access statistics summary will be generated for analysis.

**Figure 11**

Main Menu of Online Analytical Mining of Path Traversal Patterns.

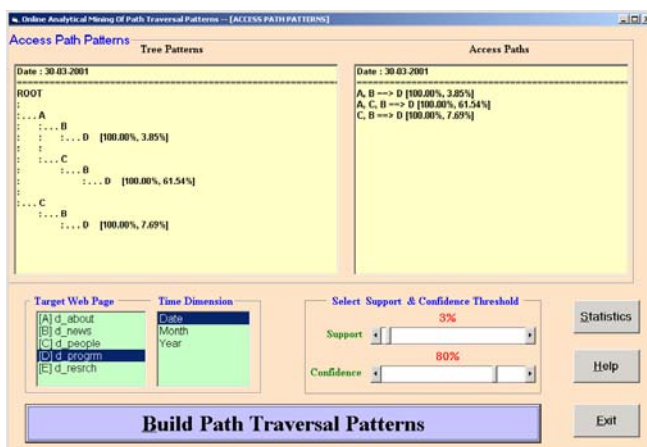


### **Initialization**

The initialization consists of three major components including: Open Log File, step 1 of data loading and cleansing and step 2 of extracting & Rule Generation. We can simply click the buttons sequentially and follow the instruction to complete the process.

Figure 12 shows the screen layout of access path patterns. The program will ask user to select / specify several parameters before building the potential user access paths and statistics summary. First user should select the target web page and time dimension that they are interested in. Also the two thresholds values: Support and Confidence can be set according to the user preference. Then by clicking the button "Build Path Traversal Patterns", a set of potential access paths will be generated. There are two windows in the screen. Both show the same information of user navigation paths. One is in graphic form, say FP-tree while the other one is in text form for easy readability. As a result, analyst can obtain their desired knowledge.

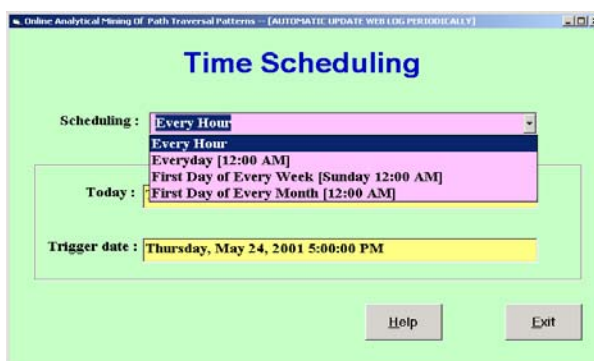
Example 1 For example, the target page ‘d\_prog’ and time dimension “Date” are selected and the confidence and support thresholds are set to 80% and 3% respectively. Then a set of access patterns is generated if its confidence and support levels is greater or equal to the values inputted by the user. Figure 12 displays the result of the query.



**Figure 12**  
All Significant Access Paths (Confidence = 80% and Support = 3%)

### Time Scheduling

Figure 13 shows the time scheduling menu where a user can set the time in which whenever the user accessed path is recorded in the web log file. A corresponding update will be made to the frame metadata, which in turns triggers the update of the user access patterns on the data warehouse. As a result, an up-to-date user access patterns can be maintained. The system provides four options for time scheduling.



**Figure 13**  
Time Scheduling.

### Online Analytical Mining of Path Traversal Patterns

In web usage, users activities on website are recorded into server log file continuously even though path traversal patterns have been derived before. As a result, the derived path traversal patterns will be out-dated soon. In order to maintain the current status of the path traversal pattern, we need to update the user access patterns continuously or periodically whenever the log file is being updated. This can be accomplished by time scheduling.

Suppose the access log is being updated after a period of duration according to the time set in the time scheduling part. As a result, an up-to-date user accessed patterns have been maintained.

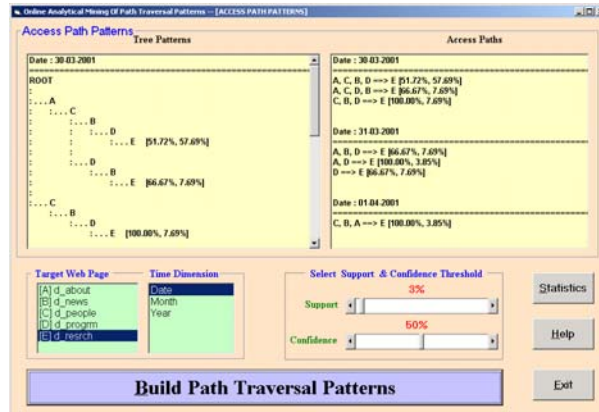
The system provided some Online Analytical Processing (OLAP) functions. It includes roll up and drill down. It will be demonstrated in the following examples. The following figures show the up-to-date user

access patterns and statistics summary.

**Example 2**

The target web page ‘d\_resrch’ and time dimension “Date” are selected and the Confidence and Support thresholds are set to 50% and 3% respectively. Then a set of access patterns is generated if its confidence and support levels are greater than or equal to the values inputted by the user. Figure 14 displays the result of the query.

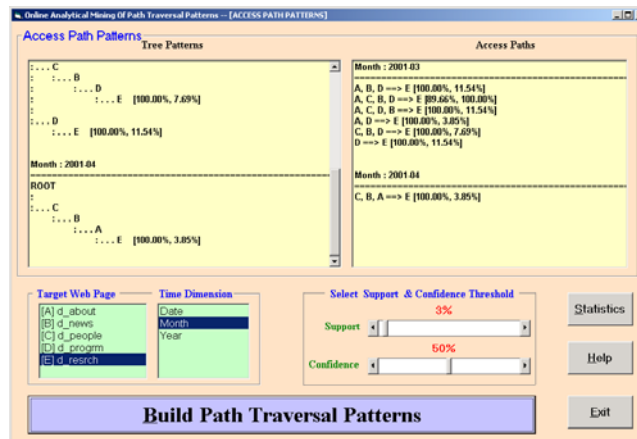
**Figure 14**  
All Significant Access Paths.



**Example 3**

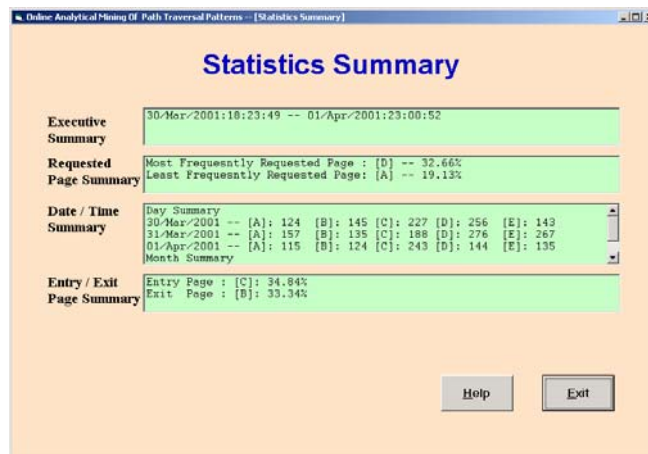
The target web page ‘d\_resrch’ and time dimension “Month” are selected and the Confidence and Support thresholds are set to 50% and 3% respectively. Then a set of access patterns is generated if its Confidence and Support levels are greater than or equal to the values inputted by the user. Figure 15 displays the result of the query.

**Figure 15**  
All Significant Access Paths.



Besides the user navigation paths, some useful statistics are also provided for analysis. By clicking the button “Statistics”, a screen will appear. Specifically, the main modules of online analytical mining of path traversal pattern provides four difference statistics. Executive summary provides a general statistic result for the entire time period of the log data. On the other hands, it specifies the time period of the log involved in the system. Requested page summary presents the most and least frequently requested pages by visitors of a website. Date / Time summary summaries information about the total number of pages viewed for the month, week and day. Entry page summary presents information about the entry pages viewed by visitor of a web site. Exit page summary presents information about the exit pages viewed by visitor of a web site.

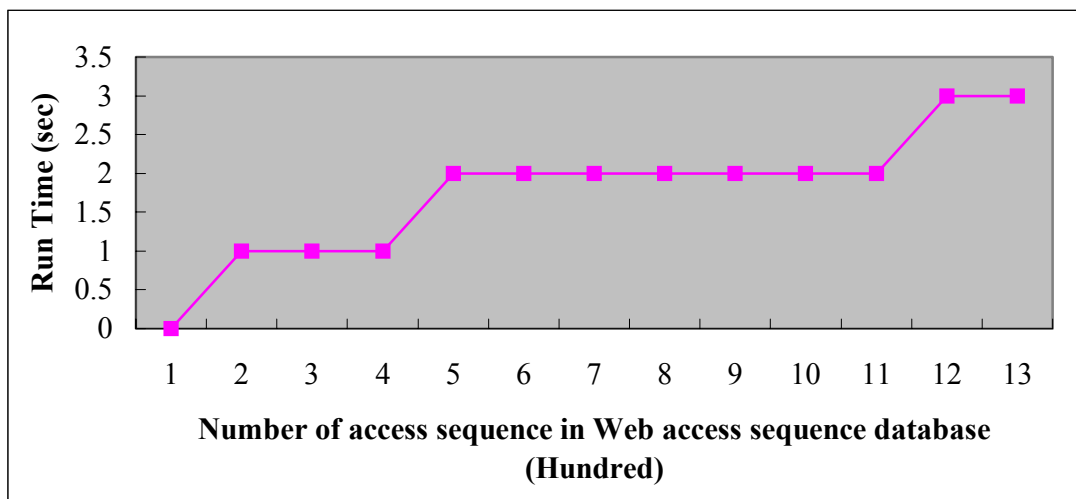
Figure 16 shows the statistics Summary of the web usage mining from the access log.



**Figure 16**  
Statistics Summary.

### PERFORMANCE EVALUATION

To access the relative performance of the algorithm for discovering path traversal patterns, we performed several experiments on an IBM compatible computer with a Mobile Intel Pentium III CPU clock rate of 750 MHz, 128 Megabytes of main memory, and running Windows 2000 Professional. The data resided in the FAT 32 file system and was stored on a 20 Gigabytes Ultra-ATA hard disk. The relational database is Sybase SQL Anywhere 5.0 for data storage. All programs are written in Microsoft Visual Basic 6.0. The web log covers the year of 2001 and its size is 102 MB.(Wong, 2001)



**Figure 17** Experimental Results.

The experimental result is shown in Figure 17. The FP-tree shows linear scalability with the number of access sequences in the databases. In case of large database, it is a good candidate to use for access patterns discovery. In our case study, we are only interested in five web pages. The total number of combinations of the traversal patterns is  $({}_5C_5 + {}_5C_4 + {}_5C_3 + {}_5C_2 + {}_5C_1 = 325)$  and the maximum depth of the FP-tree is 5. The FP-tree can be constructed within several seconds even though the numbers of transactions are greater than 100K. Thus, it is very efficient for online analysis purpose. The cost of FP-tree construction is  $O(|\text{number of frequent items in Transaction}| = 5)$ . In general, FP-tree is an effective structure facilitating web path

traversal patterns mining.

We believe that, with certain extensions, the methodology of FP-tree can be applied to perform many web usage mining tasks efficiently such as web user path traversal patterns mining.

## CONCLUSION AND FUTURE WORK

We have proposed and developed an OLAM methodology that provides the means for management investigation on e-customers' click behavior, so as to further analyze their scale of preference and habit on a website surfing for the web advertisement planning and design. In our approach, a mechanism of automating the view of the data warehousing has been introduced. The view is provided by joining a dimension table and a fact table, and keeps record of user access paths in a fact table. As the click sequence and path traversal patterns represent the customer's theme, these findings could also be translated into web site design and could then be utilized to refine the web-site infrastructure. The refinement of the web-site design could generate much different pattern of e-customer web-pages click sequence. This phenomenon is a cyclic circle. To ensure timeliness, our OLAM method takes a dynamic mining approach for most updated analysis, by providing continue refinement according to the change of the web-site environment. However, problem exists of how to synchronize the update of the based relations with the update of the view. This paper offers a frame model metadata to facilitate the trigger event, which will be invoked whenever an incremental update occurs in the based relation, i.e. access log. The frame model metadata consists of data operation, which can be used to update the user access path. As a result, with OLAM, we can transform the data warehousing into an active data warehousing which can activate the incremental data update from the based relation into an existing view, after update during time interval.

The discovery of e-customer click sequence and profile could help in designing a customer-focused web site in the following ways:

1. Make web site functionality intuitive by restructuring it around e-customers' preferring surfing routes and processes. The popular web pages with most diversified pre-requisite sequences and longest surfing time could be identified and refined appropriately with its page content and infrastructure.
2. The isolated and inactivate web-pages could imply that browsers are either incapable of access to it or simply not interested enough to arouse a click. Further analysis on these web-page content and its dynamic links are necessary, to decide upon whether metaphor on web site is necessary.
3. Relate utility<sup>1</sup> to relevant customer actions by easy accessible and visible utilitarian components.

The future direction of this research is to enhance our methodology with association rules established

---

<sup>1</sup> web site functionality that allow browsers do something useful to serve them better and faster, they normally addresses common areas of customer frustration or desire of new/extended activities

between the UID in the end result click sequence patterns and the UID associated attributes such as the user's personal particulars, for more association semantics discovery. The discovery of targeted customers' personal online preference and off-line particulars are important source for Customer Relation Management (CMR) to build customer-oriented websites in the future.

The future direction of this research is:

1. Since web log data provide information about what kind of users will access what kind of web pages, web log information can be integrated with web content and web linkage structure mining to help web page ranking, web document classification, and the construction of a multi-layered web information base as well.
2. Sequential pattern mining algorithms tend to generate a huge number of sequences, and at any given time, not all of those are of interest to the user. For example, a marketing analyst may only be interested in the activity of those online customers who have visited certain pages in a specific time period. In general, the discovered patterns must meet certain rules and conditions. As a result, certain constraints should be integrated with the web mining techniques in order to get a more reasonable and desired knowledge.

In conclusion, the importance of web usage mining will continue to grow with the popularity of WWW and undoubtedly will have a significant impact on the study of the online users' behaviors.

## REFERENCES

- Agrawal, R., Imielinski, T., & Swami, A. (1993). In Proc. 1993 ACM-SIGMOD Int. Conf. Management of Data. Mining association rules between sets of items in large databases, 207-216.
- Agrawal, R., & Srikant, R. (1994). In Proc. 1994 Int. Conf. Very Large Databases. Fast algorithms for mining association rules, 487-499.
- Agrawal, R. & Srikant, R. (1995). In Proc. 1995 Int. Conf. Data Engineering. Mining sequential patterns, 3-14.
- Buchner, A.G., Baumgarten, M., Anand, S.S., Mulvanna, M.D., & Hughes, J.G. (1999). In KDD Workshop on Web Usage Analysis and User Profiling (WebKDD'99). Navigation Pattern Discovery from Internet Data, 25-30.
- Chaudhuri, S., & Dayal, U. (1997). ACM SIGMOD Record. An overview of data warehousing and OLAP technology, (26), 65-74.
- Cheung, D.W., Han, J., Ng, V., & Wong, C.Y. (1996). In Proc. 1996 Int. Conf. Data Engineering. Maintenance of discovered association rules in large databases: An incremental updating technique, 106-114.
- Cooley, R., Mobasher, B., & Srivastava, J. (1999). Journal of Knowledge and Information Systems. Data Preparation for Mining World Wide Web Browsing Patterns, 1(1), 5-32.
- Catledge, L., & Pitkow, J. (1995). Computer Networks and ISDN Systems. Characterizing Browsing Behaviors on the World Wide Web, 27(6).
- Chen, M.S., Park, J.S., & Yu, P.S. (1998). IEEE Trans. on Knowledge and Data Engineering. Efficient

- Data Mining for Path Traversal Patterns, 10(2), 209-221.
- Fong, J., & Huang, S. (1997). Springer Verlag. Information Systems Reengineering, 179-212.
- Fong, J., Kwan, I., & Wong, H.K. (2001). The Fifth Pacific-Asia Conference on Knowledge Discovery and Data Mining. Online Marketing Support Using Online Analytical Mining Path Traversal Patterns.
- Fong, J., & Huang, S. (1999). International Journal of Cooperative Information Systems. Architecture of a Universal Database: A Frame Model Approach, 8(1), 47-82.
- Fong, J., & Pang, F. (1999). Proc. of Systems, Cybernetics and Informatics. Schema Evolution for New Database Applications: A Frame Metadata Model Approach, (5), 104-111.
- Fong, J., Wong, H.K., & Fong, A. (2000). Journal of Data Warehousing. Online Analytical Mining Web-Pages Tick Sequences, 5(4), 59-68.
- Fong, J., Wong, H.K., & Fong, A. (2000). The Second International Workshop on Information Integration and Web-based Applications & Services. Online Analytical Mining Association Rules on Web-pages Tick Sequences.
- Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., & Pirahesh, H. (1997). Data Mining and Knowledge Discovery. Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-totals, (1), 29-54.
- Han, J., & Kamber, M. (2001). Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers.
- Han, E.H., Karypis, G., & Kumar, V. (1997). ACM. Scalable Parallel Data Mining for Association Rules, 277-288.
- Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., & Hsu, M.C. (2000). In Proc. 2000 Int. Conf. Knowledge Discovery and Data Mining (KDD'00). FreeSpan: Frequent Pattern-Projected Sequential Pattern Mining, 355-359.
- Masseglia, F., Cathala, F., & Poncelet, P. (1998). In Proc. 1998 2nd European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'98). The PSP Approach for Mining Sequential Patterns, 176-184.
- Mohania, M., Madria, S., & Kambayashi, Y. (1999). Proc. of the 9<sup>th</sup> International Database Conference. Self-Maintainable Aggregate Views, 306-317.
- Masseglia, F., Poncelet, P., & Cicchetti, R. (1999). Networking and Information Systems Journal. An Efficient Algorithm for Web Usage Mining, 2(5-6), 571-603.
- Miller, R.J., & Yang, Y. (1997). In Proc. 1997 ACM-SIGMOD Int. Conf. Management of Data. Association rules over interval data, 452-461.
- Park, J.S., Chen, M.S., & Yu, P.S. (1995). In Proc. 4th Int. Conf. Information and Knowledge Management. Efficient parallel mining for association rules, 31-36.
- Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., & Hsu, M.C. (2001). In Proc. 2001 Int. Conf. Data Engineering (ICDE'01). PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth, 215-224.
- Pei, J., Han, J., Mortazavi-Asl, B., & Zhu, H. (2000). In Proc. 2000 Pacific-Asia Conf. On Knowledge Discovery and Data Mining (PAKDD'00). Mining Access Patterns Efficiently from Web Logs, 396-407.
- Roussopoulos, N. (1997). KRDB, SIGMOD Conference. Materialized Views and Data Warehouses,

316-327.

Srikant, R., & Agrawal, R. (1995). In Proc. 1995 Int. Conf. Very Large Data Bases. Mining generalized association rules, 407-419.

Srikant, R., & Agrawal, R. (1996). In Proc. 5th Int. Conf. Extending Database Technology. Mining sequential patterns: Generalizations and performance improvements, 3-17.

Srivastava, J., Cooley, R., Deshpande, M., & Tan, P.N. (2000). SIGKDD Explorations. Web Usage Mining: Discovery and Application of Usage Patterns from Web Data, 1(2), 12-23.

Savasere, A., Omiecinski, E., & Navathe, S. (1995). In Proc. 1995 Int. Conf. Very Large Databases. An efficient algorithm for mining association rules in large databases, 432-443.

Svawagi, S., Thomas, S., & Agrawal, R. (1998). ACM. Integrating Association Rule Mining With Relational Database Systems: Alternatives and Implications, 343-354.

Wong, H.K., & Fong, A. (2000). First IEEE Conference on Information Technology. Object-Relational Database Management System (ORDBMS) Using Frame Model Approach.

Wong, H. K., (2001). M.Phil Thesis of Computer Science Department of City University of Hong Kong. Online Analytical Mining of Path Traversal Patterns for Web Measurement.

Wu, K., Yu, P.S., & Ballman, A. (1998). IBM Systems Journal. Speedtracer: A web usage mining and analysis tool, 37(1), 89-105.

Zaki, M. J. (2001). In Proc. of Machine Learning Journal, special issue on Unsupervised Learning. SPADE: An Efficient Algorithm for Mining Frequent Sequences, 42(1/2), 31-60.

Zhao, Y., Deshpande, P.M., & Naughton, J.F. (1997). In Proc. 1997 ACM-SIGMOD Int. Conf. Management of Data. An array-based algorithm for simultaneous multidimensional aggregates, 159-170.

Zhuge, T., Molina, H.G., Hammer, J., & Widom, J. (1995). Proceedings of the International Conference on Management of Data. View maintenance in a warehousing environment, 316-327.