# ARCHCOLLECT
## *A Tool for WEB Usage Knowledge Acquisition from User's Interactions*

Ahmed Ali Abdalla Esmin
*Department of Computer Science (DCC), Federal University of Lavras (UFLA), Brazil*

Joubert de Castro Lima, Edgar Toshiro Yano
*Instituto Tecnológico de Aeronáutica (ITA), Brazil*

Tiago Garcia de Senna Carneiro
*Federal University of Ouro Preto (UFOP), Brazil*

Keywords: Data acquisition tool, Interactions.

Abstract: This paper presents a low coupled acquisition mechanism focused on users interactions, associated with semantic data. This tool, named ArchCollect, is used for collecting, transforming, loading and displaying user interactions. Its architecture is composed by seven components that gather information coming directly from the user, regardless the user monitored applications. The ArchCollect architecture has a relational model with capacity for keeping important information for two main areas: the commerce with products or services, quantities and prices, and applications with process, quantities, prices and employees. The relational model also added the possibility of obtaining the time spent to serve each user interaction on the application servers and on the ArchCollect servers. In this architecture, data extraction and analysis are performed either by internal algorithms, or by decision support tools, such as OLAP, Data Mining and Statistic.

## 1 INTRODUCTION

Knowledge acquisition mechanism and tools that monitor users' interactions are being actively developed on the Web. These tools are essential for Web usage mining projects, either in electronic commerce and electronic business.

Interaction is a general term used for classifying specific events that were emitted by users in any sort of application. These events are classified by clicks on elements on a page of an application. These elements are buttons, links and banners, the last one used particularly for commerce applications.

Especially in (Chen, 1996), (Gomory, 1999), (Lee, 2000), and (Kimball, 2000) the users interactions analysis area has been largely studied. There are also many commercial tools available for this area such as Andromedia, DoubleClick, Engage Technologies, IBM Corp´s SurfAid, Marketwave Corp´s, Media Metrix, net.Genesis´net.Analysis, NetRating Inc., Straight UP, Oracle 9i Inc.. All these tools extract the initial data from servers log files, what influences their portability and increases the complexity to establish the interaction pattern.

The purpose of this paper is to discuss ArchCollect internal components that are completely adjusted to the newest metrics developed by (Gomory, 1999), (Lee, 2000), and (Kimball, 2000), have low coupling to the existent application and represent a complete analysis model.

The internal components communication starts with the information collected by the user component, that are inserted by a parser software into the HTLM, XHTML or XML code of the existent application, and the information collected by the collecting component from a user who visits the application for the first time.

Since the data has been collected, forming the interaction pattern, this interaction pattern has to be transformed in a relational model that have granularity in the order of the users unique



Ali Abdalla Esmin A., de Castro Lima J., Toshiro Yano E. and Garcia de Senna Carneiro T. (2008).
ARCHCOLLECT - A Tool for WEB Usage Knowledge Acquisition from User's Interactions.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - AIDSS*, pages 375-380
DOI: 10.5220/0001722003750380

interactions. The relational model created in the architecture emphasize purchases, sales, actualization, business results, modules of the used systems and users that can be classified in clients or employees depending on the application that is going to be monitored. We also aggregate to this model the response and service time related to each user interaction on each server (application and ArchCollect).

The extraction of the stored data, according to the final pattern of interaction, can be implemented by OLAP tools, Data Mining, Statistics services, or by the personalization component, depending on the application.

The rest of this paper is organized as follows: Section2 explains, with details, the ArchCollect architecture components, its low coupling to the existent application and the relational model developed for the main areas of the architecture. In section 3, the related works are emphasized and compared to the developed architecture. Finally, Section 4 concludes this paper with some discussions on the relevant future work.

# 2 ARCHCOLLECT

ArchCollect components are separated in specific servers to enable the architecture to have a low coupling to the existent application and to be scalable. The components are: user component, collecting component, transforming component, loading component, duplication component, personalization component and visualization component. Figure 1 shows the architecture with the specific servers and their respective components.

## 2.1 User Component

The user component collect the name and the identifier of the clicked element, the complete interaction date with year, month, day, hour, minute and second, the user IP address, the page where the interaction occurred, the interaction position x and y, the user operating system, the screen resolution, and the viewed time. For specific elements called business elements, that exist in commerce application on the web, besides this information, the ArchCollect collects the quantity that was bought, sold or actualized and the amount in money that was bought, sold or actualized for each product or process. All this collected information is stored in a specific element on each page, typically a hidden text field, and afterwards re-passed to ArchCollect server components.

The only ArchCollect architecture's component that depends semantically on the existent application is the user component. This component is inserted by parser software into the HTML, XHTML or XML code of the existent application, causing no changes on its code, besides an increment on the number of lines of each page of the application. This mechanism allows more flexibility for the user component, once they fit perfectly to any application with no changes in the context, and consequently with no restructuring of the code at each application.

## 2.2 Duplication Component

When the user sends a request, a component (ASP, JSP, etc) in the existent application server is responsible for emitting a response to this request. Each request is an interaction, in the ArchCollect architecture's point of view. This interaction has to be received also by the ArchCollect server, regardless the existent application. To solve this problem, it's proposed the duplication component. This totally independent component operates after the server that is responsible for collecting all the interactions of all users. This server which belongs to the existent application can be a firewall, a load balancer, for example.

The duplication component starts listening to the port 80, which has already been defined. Doing this, all the requests made to the existent application will also be made to the duplication component.

Since all the requests are obtained, it is necessary to re-pass them to the collecting component and to the components of the existent application. We send an identical request to the existent application and a modified request to the collecting component. We analyze the URL sent by the user when carrying out an interaction, and change the name of the component written in this URL to the ArchCollect architecture's collecting component, named *CollectComp.class* and, finally, sent it.

As each server receives a request and each server sends a response. The ArchCollect architecture's server does not send the contents, only the heading containing the architecture's cookies. On the other hand, the existent application's server sends the heading and the contents. It gathers both answers in a unique answer and sends it to the client browser.

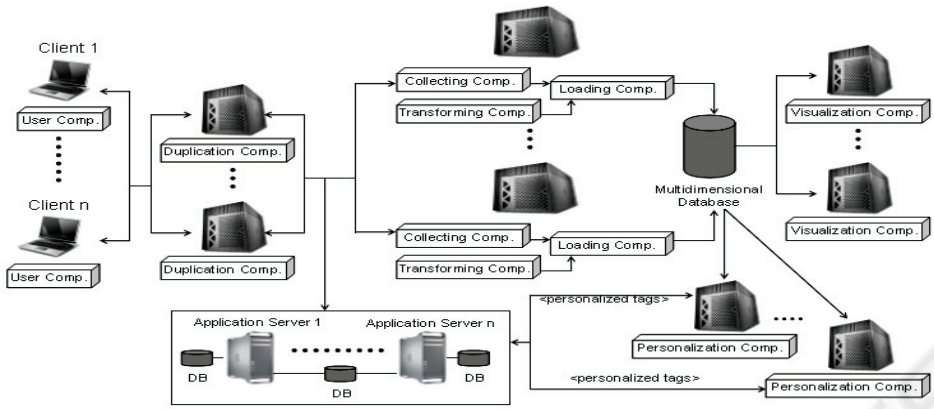


Figure 1: The ArchCollect Components.

## 2.3 Collecting Component

ArchCollect works with cookies to obtain unique users identifiers. The architecture adopted two kinds of cookies: one to identify the user permanence during a visit called session, and the other to identify the user in repeated visits, called persistent cookie.

Only for applications where the user can be anyone, the deadline of persistent cookie is set up in 30 days. For the rest of the applications the deadline is unlimited. Doing this we assure the host's identification. In cases where many users request a certain application from the same host or one user requests this certain application from many different hosts, it's not able to obtain a precise relation of identifiers.

After verifying the cookies and attributing or not these cookies to the users, the collecting component adds the attributed cookies to the information received by the user component. In some cases where the user does not have session cookie, we attribute a cookie to this user and add an identifier that characterizes the interaction as the first interaction of this user. This set of information is defined as the pattern of ArchCollect architecture's interactions.

This interactions pattern has sufficient information, which adjusts to the metrics developed by (Gomory, 1999), (Lee, 2000), and (Kimball, 2000) and to propose some questions that have not been emphasized yet, providing, then, dynamic content presentation forms to the users and administrators of an application on the web.

For the applications where the user cannot be previously identified, this information is obtained from the collecting component. When verify the existence of the cookies, in case the user does not have the persistent cookie, he receives one and extra information about this user is collected, e.g., the HTTP heading. This extra information is stored in the ArchCollect architecture's multidimensional model by the loading component. The multidimensional model and the loading component are described at the end of this section. For the rest of the applications the persistent cookies and the extra information are maintained, since in these cases the users are previously registered.

## 2.4 Loading and Transforming Components

The transforming component is responsible for identifying unique interactions and the unique sessions of each user. The transforming component obtains the unique interactions immediately, since it follows the interaction pattern. By analyzing the interaction pattern, it notices the existence of session identifiers for some interactions that characterize the beginning and the ending of a session. These identifiers are responsible for guaranteeing the entrance and the exit page of each user.

The unique sessions with unique interactions are stored in the multidimensional model by the loading component. The main purpose of the loading component is to obtain the information from different components, such as transforming, duplication and collecting component, and stores it on the multidimensional model.

## 2.5 Multidimensional Model

The relational model, presented in Figure 2, is the ArchCollect architecture's core. It reflects the ArchCollect architecture's final pattern of interactions.

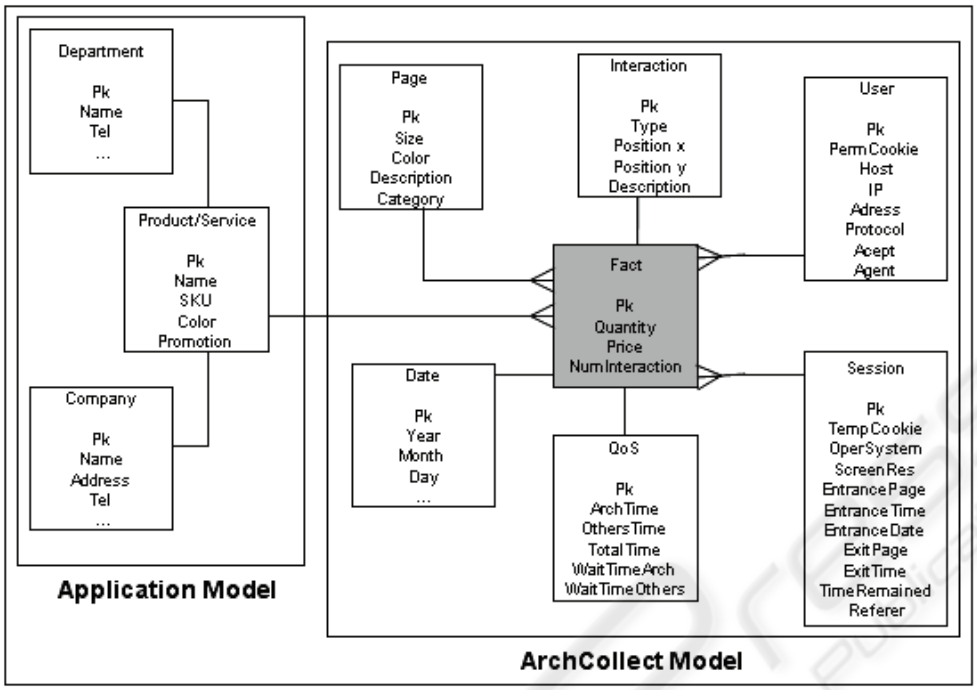


Figure 2. The ArchCollect Multidimensional Model.

Questions, illustrated by (Gomory, 1999) are emphasized, such as business results and purchases. Also, a new analyses focus can be obtained due to the collection of the times related to each interaction executed by the existent application's users. Information about costs together with information about elapsed times gives us a sophisticated multidimensional model that had not been proposed yet.

In this model, two types of applications are proposed. One type reflects e-commerce and e-business applications. In this model, features such as purchases, business results and the elapsed times related to each interaction are emphasized. For the other type of corporate interactions, it is relevant to know features such as process, sales, purchases, data updates and elapsed times related to each interaction. So, the architecture is adequate to innumerous contexts. For the general context of interactions it is relevant to know information such as date, user, page, page view, interaction positions, session, operate system, screen resolution, and, finally, the interaction itself according to the model proposed by (Gomory, 1999), (Lee, 2000), and (Kimball, 2000).

## 2.6 Visualization Component

Since the interactions are stored in the multidimensional model, the visualization component is used to visualize the answers to the following 30 questions, presented in Figure 3. These questions are divided in two parts: the first is composed by the questions from 1 to 15, that were formulated at (Gomory, 1999), (Lee, 2000), and (Kimball, 2000). The second part is proposed in this work.

## 2.7 Personalization Component

Web commerce applications deal with anonymous users who arrive, emit interactions and leave the application. Important information with granularity in the order of the users interactions are obtained by the components that were described until now.

This strategic and behavioral information has been passed online to the application managers, and then re-passed to the users through modifications in the application contents. Also, in order to re-pass these modifications online to the commerce application user, it is proposed the personalization component.

The information defined in the ArchCollect multidimensional model is used for the establishment of the user's profile. With this profile, the set of products, services or process that will compose the next page to be presented to the user is defined. When the monitored application needs to send a new page to the user, it may request the

personalization component for its content in terms of product's or service's advertisements, or it may ask what will be the better page layout for this user.

The personalization component task is to generate, in background and periodically, a list of products, services or processes, either associated to a purchase or just visited by the user. This list corresponds to the user profile, i.e., the items that can be used to generate the next pages to be presented.

# 3 RELATED WORKS

Some works were proposed to deal with data extraction under many different perspectives, in other words, to illustrate analyses offered to administrators (*Sites Modification, Systems Improvement, Business Intelligence*) and offered to the application users (*personalization*) (Spiliopoulou, 1998), (Wu, 1998), (Srivastava, 2000), (Shahabi, 2001).

A procedure for analyzing and describing the different works is using the three more important parameters proposed in (Srivastava, 2000).

1. Data source: data for analysis may come from many different sources such as the server, the client or the proxy. Projects such as WebSifit, SpeedTracer and WUM obtain their data from the servers. The Shahabi project obtains its data directly from the client and the ECI architecture analyzes data coming from both sources, in other words, the client and the servers.

2. Usage data: which categories have the data obtained for analysis? Projects such as the WebSifit use data called usage data, content data and structure data. The SpeedTracer project uses only usage data. The Shahabi project, as well as the ECI architecture and the WUM project, use usage data and structure data.

Table 1 summarizes each work's characteristics using some of criteria proposed by (Srivastava, 2000).

| Project | Data Source | Usage Data | Coupling |
|---|---|---|---|
| WebSifit | Web server | Usage/Content/ Structure | High |
| ECI IBM | Web server /user | Usage/Structure | High |
| SpeedTrace | Web server | Usage/Structure | High |
| WUM | Web server | Usage/Structure | High |
| Shahabi | User | Usage/Structure | Medium |
| Archcollect | User | Usage/Structure | Low |

1. Which users make more requests to the existent application?
2. In which period and at what time do users make more requests to the application?
3. How much time do users spend at the application in each session?
4. How many pages are visited during their permanence?
5. How did the users get to the application? How many users come from a certain place?
6. How many sales of the application occurred during a certain time?
7. How to decide if an advertisement is working or not?
8. How to decide if a page's layout is efficient?
9. Which advertisements are being converted into purchases?
10. Which products combinations are sold more frequently?
11. Which promotion sales are sold more frequently during a certain time?
12. Which products or services are being sold more frequently during a certain time?
13. Which company sells more products or services?
14. How frequently and how many of the products or services are being acquired?
15. Which products are left behind more frequently?
16. How many interactions are made so that the first purchase of a user can happen? And for the rest of the purchases?
17. How many interactions happen in a certain period?
18. How many interactions are made by each user in a certain period?
19. What percentage of all the interactions end in purchases?
20. How many interactions are made for each page element, page, session and visitor?
21. How much money was sold, bought or actualized by an employee in a session, in an interaction and in a certain time?
22. How many sessions does each user have during a certain time?
23. What is the relation between the time spent to carry out an interaction and the value given to this interaction?
24. How long on average does it take to carry out a certain service? Which is the average financial value related to this service?
25. What services have the better and what services have the worst cost-benefit relationship?
26. How many users the web application can deal with, if we establish that the service level of each interaction is equal to 30 seconds?
27. What will be the speed-up of the web application if we double the number of servers?
28. If a promotion makes the number of user increase at 10 percent, what will be the system behavior?
29. Where are the possible system bottlenecks?
30. How the web application performance is affected when the number of simultaneous users increases?

Figure 3. Questions to be answered by the Visualization Component.

# 4 CONCLUSIONS AND FUTURE WORKS

This paper presented architecture for collecting and analyzing user's interactions. The partial independence of the existent application showed us a model that is easy to be comprehended and implemented, called ArchCollect. Components with specific functions allowed the creation of a tool that keeps sufficient information for its purpose by the final pattern of interaction that was established in the architecture. All this was obtained by a single data source which is the application user.

The architecture has already been implemented and tested on (Lima, 2003), (Lima, 2004). The presented work can be extended to improve the understanding of the collected interactions. The personalization component is just one concept for the understanding of the collected interactions. Web usage mining algorithms or communities creation algorithms will be able to bring a huge contribution for the understanding of the interactions, and then providing a more sophisticated user's behavior profile. When establishing bigger sets called communities, we improve crucial questions such as a better performance in the obtainment of the profiles.

This version of ArchCollect collects the waiting time, the service time of all the interactions that are analyzed by the existent applications and by the ArchCollect architecture. These times show us how much it costs to carry out a certain service or process to the user. The length of time that is necessary for the administrator to analyze this interaction is still unknown, this information is very important to any web business. An extension of the architecture's internal analysis would be the development of time collecting in all components, allowing us to know which component, specifically, behaviors as the bottleneck of the complete architecture in an analysis moment previously specified.

# ACKNOWLEDGEMENTS

To FAPEMIG and CNPq for supporting this work.

# REFERENCES

Andromedia Inc´s Aria, http://www.andromedia.com

Chen M.S., Park J.S., Yu P.S. *Data Mining for Transversal Patterns in a Web Environment*. Proc. of 16th International Conference on Distributed Computing Systems, 1996.

DoubleClick Inc, http://www.doubleclick.com

Engage Technologies Inc, ttp://engagetechnologies.com

Gomory S., Hoch R., Lee J., Poldlaseck M., Schonberg E. *Visualization and Analysis of Clickstream Data of Online Stores for Understanding Web Mechandising*. IBM Watson Research Center, 1999.

Gomory S., Hoch R., Lee J., Poldlaseck M., Schonberg E. *Analysis and Visualization of Metrics for Online Merchandising*.IBM Watson Research Center, 1999.

Gomory S., Hoch R., Lee J., Poldlaseck M., Schonberg E. *Ecommerce Intelligence: Measuring, Analyzing, and Reporting on Merchandising Effectiveness of Online Stores*. IBM Watson Research Center, 1999.

IBM Corp´s SurfAid, http://surfaid.dfw.ibm.com

Kimball Ralph, Merz Richard, *Data webhouse tool kit*, 2000.

Lee J., Podlaseck M., Schonberg E., Hoch R., and Gomory S. *Understanding Merchandising Effectiveness of Online Stores*, published in the International Journal of Electronic Commerce and Business Media, January, 2000.

Lima J.C., Carneiro T.G.S., Pagliares R.M., et. Al. *ArchCollect: A set of Components directed towards web users' interaction*. ICEIS 2003: 308-316.

Lima J.C., Esmin A.A.A., et. Al. *ArchCollect Front-End: A Web Usage Data Mining Knowledge Acquisition Mechanism Focused on Static or Dynamic Contenting Applications*. ICEIS 2004: 258-262.

Marketwave Corp´s Hit List, http://www.marketwave.com

Media Metrix, http://www.mediametrix.com

net.Genesis´net.Analysis, http://www.netgenesis.com

NetRating Inc., http://www.netratings.com

Oracle 9i Inc., http://oracle.com/oracle9i

Rabenhorst, D. *Interactive Exploration of Multidimensional Data*, Proceedings of The SPIE Symposium on Electronic Imaging. 1994, pp 277-286.

Shahabi C., F. Banaei-Kashani, J. Faruque. 2001. *A reliable, eficient, and scalable system for web usage data acquisition*. WebKDD'01 Workshop, ACM-SIGKDD 2001, San Francisco, CA.

Spiliopoulou Myra and Faukstich Lukas C. WUM: A web utlilization miner, EDBT Workshop WebDB98, Valencia, Spain,1998.

Srivastava Jaideep, Cooley Robert, Deshpande Mukund, Tan Pang-Ning. Web usage minig: Discovery and applications of usage patterns from web data. SIGKDD, January, 2000.

Straight UP!, http://www.straightup.com

Wu Kun-Lung, Yu Philip S, and Ballman Allen. SpeedTracer: A web usage mining and analysis tool. IBM Systems Journal,37(1), 1998.

Youness Sakhr. Professional Data Warehousing with SQL Server7.0 and OLAP Services. 2000