CLASSIFYING USER EXPERIENCE OF WEB APPLICATIONS IN REAL TIME USING CLIENT LOGS

Leandro Guarino de Vasconcelos[1], Rafael Duarte Coelho dos Santos[1]
and Laercio Augusto Baldochi Jr.[2]
[1]*Brazilian Institute for Space Research*
Av. dos Astronautas, 1758, São José dos Campos, SP, Brazil, 12227-010
[2]*Itajubá Federal University, Av. BPS 1303, Itajubá, MG, Brazil, 37500-903*

ABSTRACT

With the increasing use of the Internet, the Web has become the predominant means by which people obtain information. However, due to the fast growth of the amount and (sometimes competing) sources of resources available on the Web, users want to find information quickly and efficiently. Currently, web personalization has been explored in order to encourage user's feedback, improve usability and provide interesting content. In the literature, the most common approach is to analyze server logs, which contain information about what pages the user accesses during browsing. However, client logs contain more information about the user navigation. The amount of data of the client logs is significantly greater than the amount of server logs, and this is one factor that discourages analysis of client logs. In this paper, an approach is presented to classify the level of user's experience in real time, using indices of efficiency and effectiveness. The proposed approach, called RUX (Real-time User eXperience), contains an efficient algorithm for analyzing the user's behavior of web applications in real time using client logs. RUX focuses on the paths that a user goes through during the interaction, comparing them to previously defined tasks. RUX approach can be used by application developer to consume the classification of user's experience in real time, previously programming actions that can be taken. Experimental results show that the approach is efficient for aspects of data collection, latency and scalability.

KEYWORDS

Usability, Task analysis, Real time, User's experience

1. INTRODUCTION

The Web has revolutionized communication for government, businesses and people. With the increasing use of the Internet, the Web has become the predominant means by which people obtain information. However, due to the fast growth of the amount of resources available on the Web, users want to find information quickly and efficiently. Thus, the success or failure of commercial applications on the Web is due to the potential of its website to attract and retain visitors. Therefore, the analysis of user's behavior in Web applications has become increasingly important (Gunduz and Ozsu, 2003).

An essential feature of successful Web sites is the ability to deliver the right content at the right time for the user (Velasquéz and Palade, 2008). In this sense, research on adaptive Web sites has become more frequent.

Velasquéz and Palade (2008) say that adaptive Web applications are the next generation of Web development. It would be ideal the automatic readapting on the Web application. However, this creates a high risk, especially on dynamic Web applications, because damaging adaptations can be performed instead of benefiting the user. Therefore, the semi-automatic adaptation is the most common approach.

Some motivations for the development of adaptive Web applications are (Velasquéz and Palade, 2008): (i) different people imply different users, (ii) different users have different goals, (iii) user's behavior changes over time, (iv) Web applications must be restructured as soon as they grow, in order to meet the changing needs, and (v) when a user searches for a specific Web information, s/he can feel lost in the "hyperspace".

The adaptation of applications covers the analysis of user's behavior, identifying his interests and/or needs and recommending personalized content. Early last decade, Huang et. al. (2002) has emphasized the demand for systems that identify user's needs and recommend relevant information.

Techniques for Web Usage Mining (WUM) have been used due to their ability to provide useful information on the application usage by exploiting patterns extracted through the analysis of logs (Pierrakos et. al., 2003). The analysis of user's behavior during browsing can provide useful insights that lead to the customization and personalization of the user experience. Therefore, e-marketing and e-commerce professionals have great interest in WUM (Aghabozorgi and Wah, 2009). For example, clustering techniques have been used to group users that behave similarly during browsing and for extracting usage patterns (Joshi et. al., 2000; Eirinaki and Vazirgiannis, 2003).

Every day, the use of computer systems and web applications produces a huge amount of information. Most of this information is stored in logs in order to maintain the historical record on the system usage. Web application logs are divided into server logs and client logs. Server logs store data about resources accessed by users like pages, images and other files. Client logs, on the other hand, provide more detailed information about the user interaction, as mouse movements, scrolling, and keyboard events. This is possible because these logs are collected by the browser, in the user side.

The amount of data resulting from client logs is significantly greater than the amount of server logs, and this is one factor that discourages analysis of client logs in the literature. However, according to marketing thought, *"more customer data, more knowledge about his behavior"* (Kotler and Armstrong, 2000), it is important to search for solutions that exploit client logs.

This paper presents an approach for classifying user experience in real time, from the analysis of the client logs in time they are collected. The approach proposed, named RUX (Real-time User eXperience), allows to evaluate the effectiveness and efficiency during user's interaction, considering tasks pre-defined by the Web application evaluator. Classification indicate the level of user experience with the interface (beginner, intermediate, expert). RUX can be used by the application developer to analyze the result of the classification of the user's experience in real time, previously programming actions that can be taken, for example, to customize the application interface. The USABILICS tool was used to collect and analyze logs (Vasconcelos and Baldochi, 2011). This tool collects a huge amount of detail about the elements of the page and performs the task analysis from an optimal path defined by the application evaluator. Experimental results show that the approach is efficient for aspects of data collection, latency and scalability.

This paper is structured as follows: Section 2 presents the main concepts and challenges in log analysis, and briefly describes the operation of USABILICS tool used to collect client logs that are used in RUX. Section 3 details the RUX approach. Section 4 describes experiments performed to validate our approach and discuss their results. Section 5 brings our conclusions and discuss future work.

2. COLLECTING AND ANALYZING LOGS

According to Géry e Haddad (2003), analyze logs is one of the most important challenges for providing intelligent Web services. Santos et al. (2012) discuss the importance of log analysis. According to them, log data may be used for resource monitoring, analysis of security and user profiles, as well as identifying security problems in server infrastructure (usually software-related), and the potential identification of suspicious user activity. In addition to these goals, analyzing interaction logs can provide useful information that helps a web engineer to improve the quality of a Web application, making it easier and faster to use (Khasawneh and Chan, 2006).

Every day a large amount of information is produced by computer systems. Most of this information is stored in logs, in order to maintain the historical record on the system usage. Although they are seldom used, these data are stored for security reasons. A popular website can generate gigabytes of logs in a short period of time (Velasquéz and Palade, 2008).

Exploring logs is not a trivial task because the analysis of large amounts of data is computationally expensive. However, logs contain hidden information that can be valuable to an organization, such as patterns of system usage.

Unfortunately, users do not feel comfortable providing their personal data. This makes the interactive methods of data collection impracticable (Bothorel and Chevalier, 2003) and therefore there is no significant amount of personal information from users to analyze. Although it is complex to analyze logs, log data is the best way to extract knowledge about user's behavior. The most effective approaches reported for the observation of user's behavior are based on log analysis. It has been an important subject of study in computing due to the variety of techniques to the use and exploration of logs (for example: data mining, artificial intelligence, data warehousing, high processing performance, data traffic).

Zulkernine et al. (2013) say, for example, that the log analysis is crucial to retrieve knowledge about the system state to solve performance problems. Velasquez et al. (2003) consider that, among the available data from the Web, the most relevant for the analysis of user's behavior and user's preferences during browsing are the logs and Web pages.

According to Géry e Haddad (2003), the easiest way to find information about the navigation of the users is using server logs. However, for analysis of user's behavior, the most relevant studies use client logs to capture data in browser. WELFIT (Santana e Baranauskas, 2010), WAUTT (Rivolli et al., 2008), WebHint (Vargas et al., 2010), WAUTER (Balbo et al., 2005), UsaProxy (Atterer, 2006) and USABILICS (Vasconcelos and Baldochi, 2011) are tools that collect data for usability evaluation and analysis of user's behavior.

These tools collect the actions that the user performs in client-side, retrieving information about objects manipulated in the interface and the time when the actions occurred, for example (Vasconcelos and Baldochi, 2011; Vasconcelos and Baldochi, 2012b) in Section 2.1.

2.1 USABILICS Tool

USABILICS is a tool that performs remote and semi-automatic usability evaluation of Web applications. The tool consists of three main modules:

- **Logging module:** USABILICS embeds in Web pages a script that recognizes all page elements through the Document Object Model (Hégaret et al., 2000) and binds events to page elements, allowing a detailed gathering of user actions. Automatic data collection of user's interactions is accomplished through communication between a client application and a server application. The client application, implemented in the JavaScript language, captures all data about events performed by the user in the browser. So, data gathering occurs during user interaction, and real-time data is compressed and sent to the server application, implemented in Java Server Pages (JSP). Then, the data is decompressed and stored in a relational database.
- **Task definition module UsaTasker** (Vasconcelos and Baldochi, 2012a): this module allows the evaluator to record web application paths that correspond to a task from the point of view of the user. The main advantage of UsaTasker is to reduce the effort to define the paths of tasks. This is done by using an interface model called COP (Container, Object, Page). UsaTasker exploits the COP model in order to allow the generalization of events for similar objects and containers. This feature allows to represent a large group of similar tasks using a single captured task.
- **Task analysis module:** after the definition of the tasks in UsaTasker, USABILICS compares the users' interactions with the optimal path of the tasks, resulting in an index of usability that is the effectiveness and efficiency of the interaction, i.e. the error rate and the completeness of the task.

Section 2.1.1 details how the task analysis of USABILICS works.

2.1.1 Task Analysis

Considering a "Sign in" task that requires filling out a registration form with name and email, Figure 1 contains two sequences of actions related to the task. The sequence at the top represents the optimal path defined by the evaluator and the sequence at the bottom represents the actions performed by a user.

ISBN: 978-989-8533-24-1 © 2014

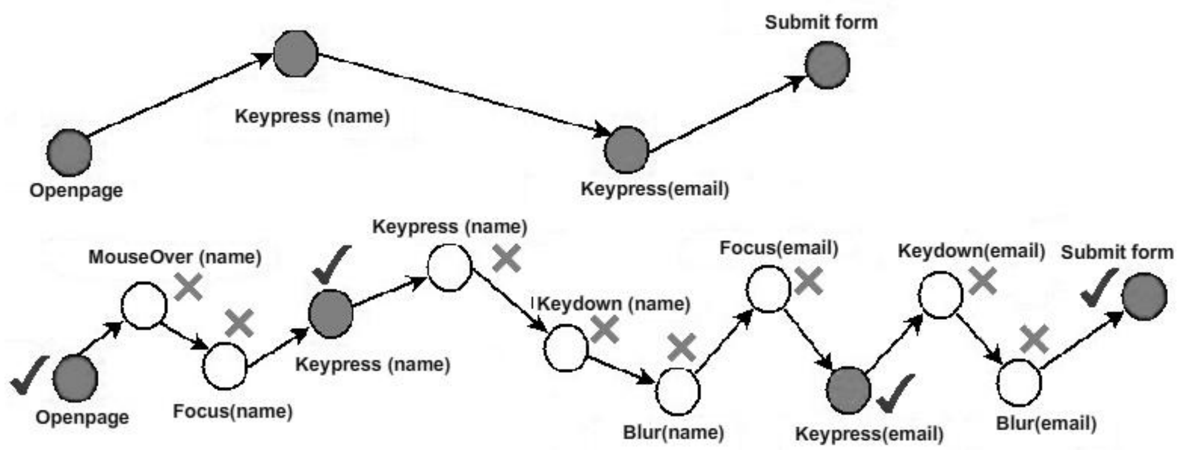

Figure 1. Comparison between the optimal path defined by the evaluator and the path traveled by the user.

A task analysis of USABILICS compares the actions of the path to the actions of the optimal path (expected actions) using a similarity measure based on the COP interface model.

For example, the first action on the path traveled by the user (OpenPage) is the first action of the optimal path; it means that the user initiated the task of filling out the form. The second action performed by the user (Mouseover (name)) indicates that he positioned the mouse over the text box called "name", but the expected event in the optimal path is different (Keypress). This comparison generates what we call "wrong action". However, there is a similarity between the two actions, differing only in the type of action. The similarity measure is detailed in Vasconcelos and Baldochi (2012b).

So, all actions performed by users are compared to the expected optimal path defined by the evaluator, resulting in a measure of the efficiency and effectiveness of Web application interface.

In USABILICS, task analysis is performed in offline mode, that is, after the user's interaction.

The proposed approach in this paper performs the task analysis in online mode, at the time the client logs are collected in order to classify the user experience in real time and notify the Web application for a possible decision. Section 3 presents the RUX (Real-time User eXperience) approach.

3. RUX (REAL-TIME USER EXPERIENCE)

According to Velasquez e Palade (2008), an advanced application for Web personalization is to generate recommendations of content and navigation during the user interaction. However, it is particularly challenging. In this sense, we propose an approach that analyzes user's behavior in real time, in order to classify the user experience for a possible decision-making during user's interaction.

For analysis of user's behavior in real time, the following points must be observed:

- **Data collection:** according to the marketing reasoning, as we collect more data about the customer, there is more knowledge about their behavior, therefore analyzed logs should provide as much detail as possible about user's interaction. As described in Section 2, client logs contain more detail than server logs.
- **Latency:** response time must be fast enough so as to allow an action to be initiated in the application to retain the user, offering him help or relevant content, before he exits the application.
- **Scalability:** considering the increasing amount of data collected and also the demand of the user web application, the analysis of user's behavior should be scalable in order to preserve the previous discussed features.

RUX approach is based on the task analysis of USABILICS, comparing the optimal path defined by the evaluator to the paths traveled by users. However, unlike USABILICS that performs an analysis in offline mode, the RUX performs analysis of user's behavior in online mode during user's interaction.

In order to provide real time analysis of client logs, we developed the algorithm presented bellow.

Algorithm *TaskAnalysisRealTime*

INPUT events collected by USABILICS and tasks defined by the evaluator
OUTPUT Efficiency and effectiveness index

foreach event *E*
  if *E* is an initiating event of a task *T* **then**
    if the user *U* is not performing any task *T'* **then**
      - Copies the event *E* to a temporary repository that stores the necessary events to analyze the current task *T*.
    else if the user *U* is performing another task *T'* **then**
      - Finishes the task *T'* that he is running.
      - Deletes the temporary repository events related to finished task *T'*.
  else
    if *E* is an end event of a task *T* **then**
      - Saves the indexes of efficiency and effectiveness.
      - Ends the current task *T*.
      - Deletes the temporary repository events related to finished task *T*.
  - Calculates the indexes of efficiency and effectiveness from the last event of the optimal path travelled by the user.
  - Stores the last event of the optimal path travelled by the user *U*.

Through this algorithm, the temporary repository stores only events related to tasks that users are running. When a task is finished (whether or not to start another), events related to that task are excluded, reducing the computational cost for the calculation of the indices of efficiency and effectiveness. In addition, calculation of indices is performed partially as soon as the user completes the steps of the task defined by the evaluator. Therefore, even if a task has many actions in the optimal path, the computational cost does not increase linearly.

Following, we present an experiment performed in order to validate the approach proposed for RUX.

4. VALIDATION AND RESULTS

To validate the RUX approach, we selected a Quiz Web application in which the user answers questions and earns points for correct answers. This web application has an interface with few features. When a question is displayed to the user, a clock counts the time decreasingly, and the options "Reply" and " I answer later" are displayed.

With UsaTasker module, the "Log in and respond" task was defined, and corresponds to the following actions:

1. Type the e-mail;
2. Type the password;
3. Click the "Login" button;
4. Wait for the presentation of the question;
5. Select an alternative; and
6. Click the "Reply" button.

The RUX approach was implemented as a Web application to get data from USABILICS and run the algorithm presented in Section 3. The Web Application was hosted on a server with 1 Core processor with 2.0 GHz, 2GB RAM, 40 GB HD and operating system Ubuntu Server 12.04 64-bit.

To optimize the performance of the index calculation, the temporary repository has been implemented on a standalone database USABILICS in the MySQL database, using the Memory engine, which stores the data in memory.

Over a week, the data was collected and analyzed in real time. 41 interactions were performed in the task defined by the evaluator. For each user interaction, we calculated the indices of efficiency and effectiveness.

Also, the latency between the sending of events by USABILICS and the return of the indices calculated by RUX has been measured. The average time spent to calculate the indices was 194 ms.

For the classification of the user's experience, we arbitrarily defined the Beginner, Intermediate and Expert levels, and two classification methods based on efficiency and effectiveness of user's interaction were used. These methods are presented in Sections 4.1 and 4.2.

The effectiveness is the amount of steps that the user has fulfilled the optimal path of the task. The efficiency is the rate of errors between the traveled path and the optimal path. Thus, an interaction may have a small error rate, but failed to complete the task; in this case, the efficiency index would be high and the efficacy rate was low.

4.1 Empirical Classification

In this method of classification, arbitrary values were used for the Beginner (values less than 0.3), Intermediate (values between 0.3 and 0.7) and Expert levels (values between 0.7 and 1.0). These arbitrary values were used to filter the indices of efficiency and effectiveness of each interaction. The result of this classification is shown in Figure 2 and demonstrates that most users are experienced in the evaluated Web application interface.

4.2 Statistical Classification

In this method of classification, users were classified through a process of collaborative filtering, where the level of experience was evaluated in relation to the level of experience of previous users. For each iteration, the average efficiency scores of previous users and the standard deviation were calculated, in order to identify the lower limit and the upper control limit. Users classified as Beginner received a lower index than the lower control limit. Expert users had highest index than the upper control limit, while Intermediate users obtained index within the control limits.

Figure 3 presents the result of this classification, and shows that the level of experience relating to other users is predominantly Intermediate. Through statistical classification, it is possible to identify users that present nonstandard behavior. From the standpoint of usability evaluation, the identification of inexperienced users (Beginners) is important for a possible discovery of usability problems in the interface.

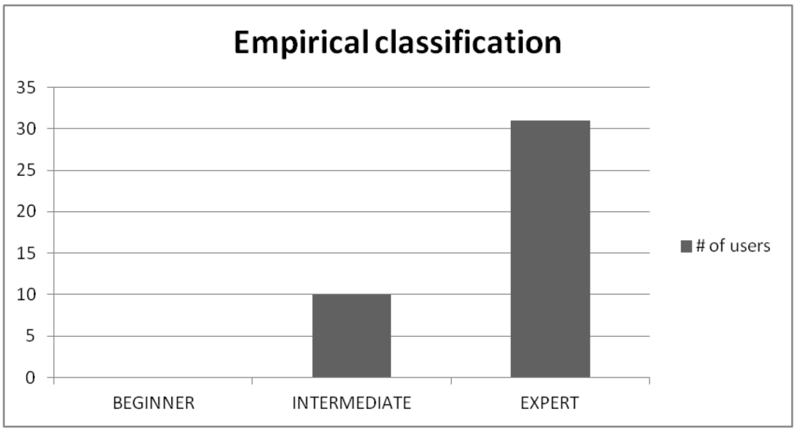

Figure 2. Level of user experience based on the empirical classification

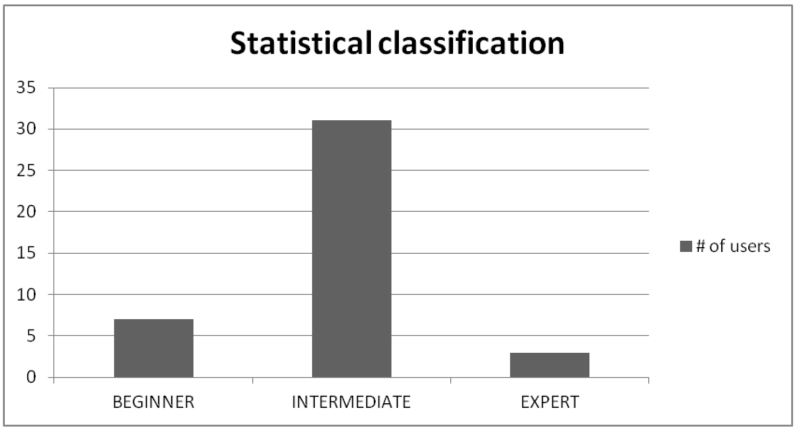

Figure 3. Level of user experience based on the statistical classification

5. CONCLUSIONS AND FUTURE WORK

Processing client logs is computationally expensive due to the large amount of data. This is one of the main reasons that discourage the development of solutions to analyze these logs in real time. However, the hidden information in the client logs is valuable for analysis of user's behavior.

The approach proposed in this paper presents an efficient way to process client logs in real time, during users' interaction in the Web applications. One factor for the efficiency is the limitation of the logs that are analyzed considering only the subset of data related to the tasks defined by the evaluator. Another factor is the proposed algorithm, which calculates the indices of effectiveness and efficiency as the user complete the task. Thus, it is possible to quickly notify the Web application on the classification of the current user's experience.

With RUX approach, it is possible for the developer to acquire information about the user's experience in real time and perform the adaptation of the interface or offer assistance, for example.

In future work, we intend to provide other information in real time beyond the level of user experience, evaluating the scalability of the algorithm simulating thousands of concurrent users and implement technical solutions beyond relational database to investigate the possibility of reducing the latency time.

REFERENCES

Aghabozorgi, S. R. and Wah, T. Y, 2009. Recommender systems: Incremental clustering on web log data. *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*. New York, NY, USA. p. 812–818.

Atterer, R., 2006. Logging usage of ajax applications with the usaproxy HTTP proxy. *Proceedings of the WWW 2006 Workshop on Logging Traces of Web Activity: The Mechanics of Data Collection.*

Balbo, S. et. al, 2005. Leading web usability evaluations to WAUTER. *Proceedings of the Eleventh Australasian World Wide Web Conference*.

Bothorel, C.; Chevalier, K., 2003. How to use enriched browsing context to personalize web site access. *In Proceedings of the 4th International and Interdisciplinary Conference on Modeling and Using Context*. Berlin, Germany. p. 419–426.

Eirinaki, M. and Vazirgiannis, M, 2003. Web mining for web personalization. *In ACM Transactions on Internet Technologies.* Vol. 3, n. 1, p. 1–27.

Géry, M. and Haddad, H., 2003. Evaluation of web usage mining approaches for user's next request prediction. *Proceedings of the 5th ACM International Workshop on Web Information and Data Management*. New York, NY, USA. p. 74–81.

Gunduz, S. and Ozsu, M., 2003. A poisson model for user accesses to web pages. *In Computer and Information Sciences - ISCIS 2003.* (Lecture Notes in Computer Science, v. 2869). p. 332–339.

Hégaret, P. L. et al, 2000. What is the Document Object Model? http://www.w3.org/TR/DOM-Level-2-Core/introduction.html.

Huang, E. et al, 2002. Comparison of interestingness functions for learning web usage patterns. *Proceedings of the Eleventh International Conference on Information and Knowledge Management.* New York, NY, USA. p. 617–620.

Joshi, A. et. al, 2000. On mining web access logs. *In ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*. p. 63–69.

Khasawneh, N. and Chan, C., 2006. Active user-based and ontology-based web log data preprocessing for web usage mining. *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*. Washington, DC, USA. p. 325–328.

Kotler, P.; Armstrong, G. *Marketing an Introduction*. Prentice Hall, 2000.

Pierrakos, D. et al, 2003. Web usage mining as a tool for personalization: A survey. *In User Modeling and User-Adapted Interaction*, Vol. 13, n. 4, p. 311–372.

Rivolli, A. et. al, 2008. Wautt: a tool for tracking the user interaction in interactive web applications (in portuguese). *Companion Proceedings of the XIV Brazilian Symposium on Multimedia and the Web*. p. 179–181.

Santana, V. F. and Baranauskas, M. C. C., 2010. Summarizing observational client-side data to reveal web usage patterns. *Proceedings of the 2010 ACM Symposium on Applied Computing*. p. 1219–1223.

Santos, R. D. C. et. al, 2012. Analysis of web-related threats in ten years of logs from a scientific portal. *In Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*. Vol. 8408.

Vargas, A. et. al, 2010. A method for remote and semi-automatic usability evaluation of web-based applications through users behavior analysis. *Proceedings of the 7th International Conference on Methods and Techniques in Behavioral Research*. p. 191–195.

Vasconcelos, L. G. and Baldochi, L. A., 2011. USABILICS: remote usability evaluation and metrics based on task analysis (in portuguese). *Proceedings of the 10th Brazilian Symposyum on Human Factors in Computer Systems & 5th Latin American Conference on Human-Computer Interaction*. p. 303–312.

Vasconcelos, L. G. and Baldochi, L. A., 2012a. UsaTasker: a task definition tool for supporting the usability evaluation of Web applications. *Proceedings of the IADIS International Conference WWW/Internet - ICWI 2012*. p. 307-314.

Vasconcelos, L. G. and Baldochi, L. A.,2012b. Towards an automatic evaluation of web applications. *In Proceedings of the 27th Annual ACM Symposium on Applied Computing*. New York, NY, USA: ACM, p. 709–716.

Velasquéz, J. et. al, 2003. Combining the web content and usage mining to understand the visitor behavior in a web site. *Proceedings of Third IEEE International Conference on Data Mining*. Melbourne, Florida, USA. p. 669–672

Velasquéz, J. D. and Palade, V. *Adaptive Web Sites: A Knowledge Extraction from Web Data Approach*. IOS Press, 2008.

Zulkernine, F. et. al, 2013. Capri: A tool for mining complex line patterns in large log data. *Proceedings of the 2Nd International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*. New York, NY, USA. p. 47–54.