Computing, Information Systems & Development Informatics Journal
Vol. 9 No. 2, June, 2018 - www.cisdijournal.org

Web Content Usage Data Logging for Discovering User Interests

Ehikioya, S.A.
Department of Computer Science
Baze University,
Abuja, NIGERIA
E-mail: ehikioya@gmail.com

Jinbo Zheng
Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada R3T 2N2
E-mail: jinbo@cs.umanitoba.ca

ABSTRACT

Smart and efficient business intelligence solutions deliver great economic values for organizations. Web usage data mining offers businesses a better understanding of how their Web sites are used. Common traditional approaches of Web usage mining use Web server access log file in W3C (World Wide Web Consortium) extended log file format as data source. This Web access log file reports Web usage by URLs which indicates only the location of served Web pages and often nothing about the content. This log may lead to inaccurate business analysis, specifically for Web sites using dynamic Web pages. This paper presents a Web content usage logging system for Web administrators and business analysts to capture Web site visitors' interests at a fine granular level. This paper also presents how a Web site, designed using the object-oriented paradigm, can benefit from this logging system to capture interested objects' attributes and relationships among these attributes. The Web content usage log provides valuable data for Web usage data mining with minimal effort in data extraction, transformation, and loading (ETL).

Keywords: E-commerce, Web log, ETL, data mining, Web usage mining

CISDI Journal Reference Format
Ehikioya, S.A. & Jinbo Zheng (2018): Web Content Usage Data Logging for Discovering User Interests. Computing, Information Systems, Development Informatics & Allied Research Journal. Vol 9 No 2. Pp 43-50.
Available online at www.cisdijournal.org

1. INTRODUCTION

The value of business intelligence solutions has been acknowledged by a fast increasing number of customers in recent years. However, a business intelligence solution can be a very complex system and may be very costly, mostly because of the complexities of dealing with big data and data warehousing. According to Friedman and Strange [1], "A significant amount of time (typically 50 percent to 75 percent) spent in building and deploying a data warehouse is taken up with data infrastructure. Finding, analyzing, extracting, transforming and loading data to the warehouse is a huge challenge, which many enterprises dramatically underestimate, often by 100 percent or more. Understanding and resolving semantic issues across the data landscape and determining the rules by which data should be merged and integrated requires significant time and resources from the IS organization and business units". These issues are still much with us today, and they pose critical challenges to usage of big data repositories.

The Internet has become a borderless marketplace for transacting businesses (purchasing goods and services) and it is increasingly very popular. This kind of transaction is called electronic commerce (e-commerce). E-commerce offers consumers capability for comparing prices of goods and services and considerable amount of product lines to choose from. E-commerce systems are reactive and cut across multiple geographical locations and heterogeneous and autonomous systems. Thus, e-commerce applications are inherently distributed. An e-commerce transaction could involve the simultaneous execution of many processes on different computers (possibly at different locations) [13]. This feature of e-commerce makes e-commerce transactions to generate huge amount of data that needs to be intelligently integrated [9, 10, 11, 12] to be able to generate business actionable meaning from it. With the growing popularity of e-commerce, the volume of consumer usage data is also growing phenomenally. As consumers / customers become more and more discriminatory with their choices for products and services consumption patterns, consumer Web usage data offer potential gold mine for merchants that can take advantage of it.

According to Beamer [17], "Customer usage data is one of the most important assets of any enterprise. The best customers will be identified by data; the customer experience will be enhanced by data; and new products and services will be developed based on data". Consequently, merchants are challenged to turn Web usage data into new revenue opportunities. However, how the internal teams can capture and log the necessary customer usage data that is required to create new sources of revenue is crucial.

The key for merchants' success in this competitive marketplace is to have accurate knowledge about the needs of potential customers and the ability to establish personalized services that satisfy these needs. Discovery and analysis of data from Web sites enables businesses understand visitors' and customers' behaviours and expectations; and how a Web site is used. Web traffic analysis, through mining Web log data, is believed to be able to greatly assist overall business decision-making by coordinating sales and marketing efforts to turn customer information into sales.

Web sites are served by Web servers. Each Web page is referred by an Internet address, called Uniform Resource Identifier (URI). When a visitor (customer) requests a Web page, the Web server usually logs the visitor's requested URI into a log file while serving the Web page content to the visitor. The Web log file contains all user activities in a time period. However, mining this log file is not always enough to reveal visitors' interests, especially for dynamically generated content. Some research work have been done to combine Web log data with Web content data for mining the integrated data efficiently, such as WebSIFT in Srivastava et al [2], and Mobasher et al [3] framework for more efficient personalization.

In this paper, we present a Web content usage data logging system to capture Web site visitors' interests at a granular level flexibly. This system logs a subset of a Web page's content for each visitor's request instead of a traditional URI. The subset is defined in a configuration file according to the site owner's interest. We developed a prototype Web site running on a Web application server with the Web content usage data logging system to demonstrate the ability of logging Web content usage information for later efficient data mining purposes.

The contributions of this paper are:

- The Web content usage data logging system provides an extensible infrastructure to capture e-commerce Web site visitors' interests. The system provides the capability, and it has the flexibility, to capture web site visitors' interests at an atomic level. The captured data is an excellent data source for further business intelligence analysis. The flexible structured data allows the logged data to be analyzed in diversified ways at many granular levels.

- Provides an architectural model of e-commerce applications content usage data logging system. It has the ability and flexibility to handle and capture information from static Web pages as well as dynamic pages to log business interesting data. Our model provides an infrastructure for personalized digital marketing.

- The implementation of our design provides a test-bed platform / environment for learning and gaining practical experience in Web content usage data logging architecture (software) design and implementation.

This paper is significant for the following reasons: First, we present a simple generic Web content usage logging system that provides the ability to record a rich set of structured data from users' requests and responses for both static and dynamic web pages. This system does not use any proprietary APIs, and only adopts popular standards and techniques. The entire logging procedure is transparent to users. Second, the rich set of well-structured data greatly reduces the data preparation efforts for data warehousing and data mining, which enables the quick development and deployment of business intelligence solutions, dramatically reducing project time and cost. Third, the prototype implementation of our web content usage logging system offers capability for accurate web usage analysis for enhancing e-commerce activities. Thus, it provides an innovative solution to give business-oriented analysis of clients' activities for e-commerce web sites.

The rest of this paper is organized as follows: Section 2 describes the design of our web content usage logging system and examines the key features of the architecture suitable for e-commerce systems, which offers the critical link to web data usage logging systems and actionable business data. Section 3 describes the implementation of our model architecture. Finally, in Section 4 we discuss our future work.

2. DESIGN OF A WEB CONTENT USAGE LOGGING SYSTEM

Most current Web traffic analysis tools are inadequate in reporting the effectiveness of specific marketing and merchandising efforts for a Web site because request URIs from Web servers' W3C standard access log file do not have strong page content tracking and analyzing ability. Simply applying data mining [14, 15] technologies on transaction URIs is unable to reveal, or may even give misleading, relationships among items presented in the transactions in a dynamic Web content scenario. For example, when there are many items in one category, the products are usually shown on many numbered Web pages. So only a portion of all products will show on each page, sorted by some attributes. Because the product database can be updated very often, adding / deleting products may result in a product to be shown on a different page. So it is very likely that two requests with same URL and parameters may get different page contents. Integrating log with content correctly over time is very challenging and requires very complex procedures [10, 11, 12], such as adding historical data auditing functions, and URI parsing programs, etc. Although static page content seems to be easier to associate with request URI to provide a content utilization analysis, the result will be incorrect when the content is updated often.

The traditional logging approach does not yield strong page content tracking and analysis ability. This situation necessitates research work in the integration of Web usage data and Web content for data mining purposes, such as Cooley et al [5] and Zaiane et al [6]. This paper introduces a new and innovative Web content usage logging system which is different from other existing Web logging approaches. The Web content usage logging system presented in this paper is used to track what information / data is actually sent back to the clients. So instead of investing huge amount of effort in the later data warehousing stage to integrate and consolidate usage data and content data, our logging system records interesting response content to considerably reduce efforts in the complex data warehousing stage. This logging system is a relatively simple but very effective solution to capture interesting data attributes requested by users via a Web content usage logging module with great flexibility in the configuration of these data attributes names. Our logging system also has the ability to handle and capture information from static Web pages as well as dynamic pages. The Web content usage logging procedure is "transparent" to web developers and Web site visitors, and the logging information module is fully configurable by Web server administrators.

With the popularity of Model-View-Control (MVC) model [16, 20, 23] in modern Web application development to support a wide variety of clients' interfaces, such as WAP browsers, Web services, and email services, etc., while reusing business logics, HTTP Web request is not the only traffic that needs to be tracked for how it has been used nowadays. The MVC model completely separates the presentation and business logic layer while it keeps the integrity of data for business logic. It offers a different way of the working of traditional dynamic content publishing which edits mixed layout control tags and contents. It uses XML to describe business data, and uses XSL transformation capabilities to merge business data and presentation format. This allows each layer to be independently designed, created and managed, thus reducing management overhead, increasing work reuse, and reducing time to market.

The controller program running on Web application server usually acts as a gateway for business logic programs to render response data according to request type respectively. So logging data pass through this gateway and one can monitor what products and services customers are really interested in. The Web content usage logging system is designed to be an add-on module running on the gateway controller's Web application server. The response data results from business logic programs are much smaller in size with well-structured data, while the final rendered data sent to clients always have lots of extra rendering attributes with possible loose-structured data, such as HTML. In our prototype, the response data results are in XML format, which is less than 20% in size of the final HTML page. So the logging process on response data can be very efficient. Application developers also can benefit from much smaller and simpler result data for debugging and testing.

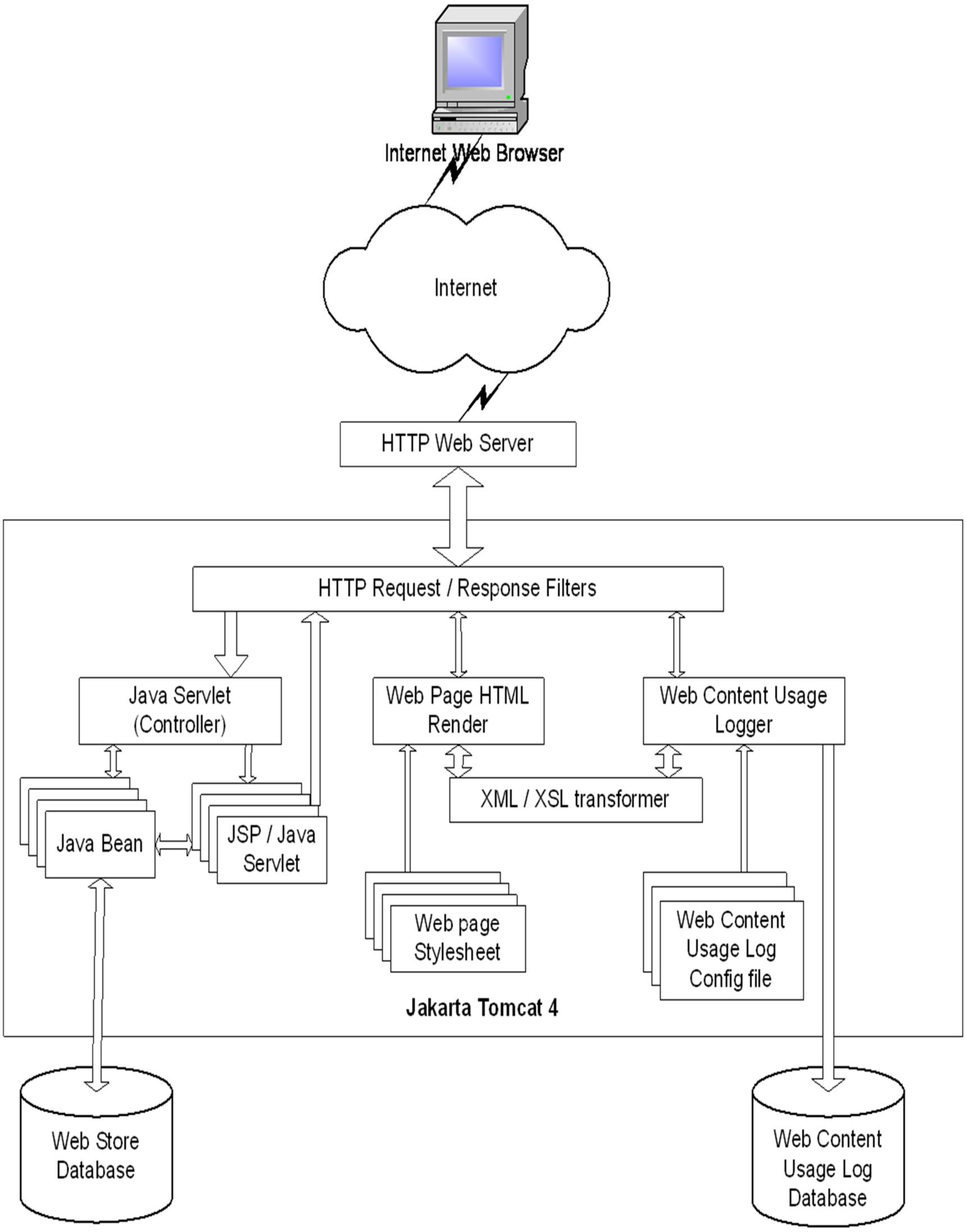

Figure 1. Web Content Usage Logging System Architecture

The Web content usage logging system prototype introduces two extra modules: the Web content usage logging module and the Web content rendering module, as add-ons for the Web application server. The (Java based) system architecture is shown in Figure 1. The Web content usage logging module reads the original response in XML format from business logic programs, instead of writing an extra content parser and define a configuration file, to simplify content usage logging process. This module uses W3C Extensible Stylesheet Language (XSL) Transformations (XSLT) to generate an abstract of the original response data as Web content log data. This XSLT process uses XSL configuration file(s) written by Web site administrators or business analysts. Any system administrator or analyst that has some knowledge of XML and XSL can control what to log from the content of his / her interest. Many recent projects (for Internet, Intranet and Extranet) adopt the XML technologies because XML solutions have a rich set of tools to facilitate data inter-exchange across applications to reduce the overall development and ownership costs.

The Web content rendering module reads the original response in XML format from business logic programs and replaces it by rendered response data in client preferred data format, such as HTML, SOAP, etc. The data rendering process also uses XSLT to produce final response in client preferred data format from the original response data from business logic programs.
This XSLT process uses XSL rendering file(s) written by Web designers or developers. The Web content rendering module selects the XSL file referenced in the *xsl:stylesheet* element in the original response, which normally is assigned by current Web application's controller. It is a very common approach in application development to use the *xsl:stylesheet* element to give the rendering file location for the current xml file. This element is defined as a standard element since W3C XSL Transformations Specification Version 1.0.

Figure 2 shows the response processing sequence on the Web application server with the Web content usage logging system running.

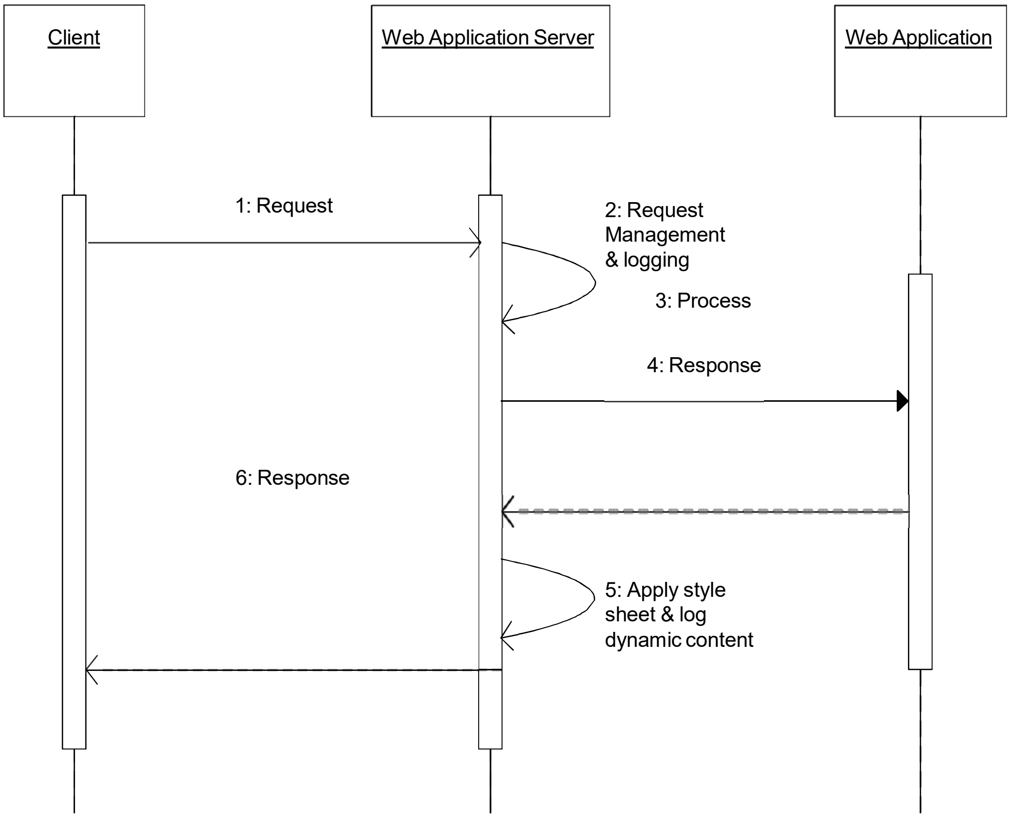

Figure 2. Request - Response Processing Sequence Diagram

3. IMPLEMENTATION OF OUR DESIGN

This Web content usage logging system prototype is implemented using Java programming language and runs on Jakarta Tomcat version 4.1.12. Java provides rich functionality in string processing, database interaction, XML processing, and Web server side programming. The implementation utilizes the filtering chain mechanism introduced in the Java Servlet Specification version 2.3 to achieve XML data XSL transformation for user interface rendering and content logging and to be transparent for clients and developers.

Although the filtering mechanism introduced in Servlet v2.3 looks similar to existing legacy module interceptor, also known as hooker, in Apache HTTP Server and most Web application servers, the Servlet 2.3 filtering mechanism is completely different architecturally. The previous hooked methods in interceptor modules are called on every request. Furthermore, method scoping and concurrency concerns in a multithread environment do not allow any easy or efficient sharing of variables and information between the different hooked method invocations when processing the same request [18, 19], causing severe impact on performance and stability on a busy site. However, the filtering mechanism in Java Servlet Specification 2.3 fully utilizes the object-oriented nature of the J2EE platform to make a chain of filters by nested method calls managed by Servlet containers in a multi-process and multi-thread approach. Filters are pluggable components that are declared in a configuration file and can be added and removed dynamically, and easily deployed in any compliant J2EE environment without touching any application code.

Using Log4J package [21, 22, 23], the logging process runs efficiently and supports a wide variety log data formats, such as tab delimited text file, XML formatted file, or database, locally and remotely without any change in the binary code. A Web site usually has limited number of XSL templates and these templates are usually relatively small in size (few kilobytes). Considering there may be too much overhead loading these files from hard disks or even remote web servers each time requested, a templates pool is also developed as a part of the Web content usage logging system, caching all the templates that may be requested in the memory.
A hash table is maintained for quick lookup of the entry memory address for each template.

A prototype Web store is developed in an object-oriented approach to demonstrate how to utilize the Web content usage logging module to capture business interested data for later data warehousing and data mining. This Web store uses the MVC architecture. The front presentation tier Web pages in this Web store use the Façade pattern [24, 25] to construct a generic and simple *Page* class for the "view" component by Web page template components, such as *PageHeader*, *PageFooter*, *PageNavigationBar*, and *PageMainPanel* classes. Data passing between tiers keeps using native format and protocol to minimize processing overhead until passing content to the "view" component. Java Architecture for XML Binding (JAXB) is used to map the instances of the *Page* class to XML formatted data as response data for user's request.

The Web content usage logging configuration file is an XSL file used to define interesting attributes in the *Page* class. So the XSL transformation in the Web content usage logging procedure will only save the value of these attributes in a tree structure to the log file.

The Web content usage logging module, as a Servlet filter, has a reference to the class instance of the *Request* class as well as a reference to the corresponding class instance of the *Response* class. So this module is also able to log all request attributes in the W3C Extend Log format standard. These attributes include client IP address, client hostname, client required URI, referrer (the reference page that led to this page) and request time. In addition, the random generated session id uniquely assigned to a Web session can also be recorded together with the rest of log attributes, eliminating one of most difficult problems in Web mining domain, the session identifying process. Any session related attributes, including customized attributes created by developers, such as username, can be recorded in the Web content usage log as well. To ensure the quality of Web site data, values of clients' request parameters can also be logged to compare content data. For example, comparing user's search keyword with the content he / she got, together with the navigation patterns; these can tell the effectiveness of the current search engine.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we present a simple generic Web content usage logging system. It provides the ability to record a rich set of structure data from users' requests and responses. To support logging the content of static Web page, we developed a static Web page handler for Tomcat Web Server. So the log information from Apache HTTP Server, which usually only contains requests for url files, is enhanced in our solution to capture both dynamic and static pages that provides the rich content for the data pre-processing stage for Web usage data mining. To support rich content logging and server side XML / XSL transformation, two plug-in modules for Jakarta Tomcat Server, the Content Transforming plug-in and the Content Logging plug-in, were developed. While the Content Transforming Plug-in is transforming the raw XML output from web applications, the Content Logging Plug-in is doing a quick parse of the same data. It generates an abstract of the interested data according to the Content Logging configuration file, the *log.xsl*.

This Web content usage logging system does not use any proprietary APIs, and only adopts popular standards and techniques (XML / XSL). The logging procedure is transparent to both the users and developers. Well structured data greatly reduced the data preparation efforts for data warehousing and data mining. This enables the later business intelligence solutions to be quickly built and deployed, dramatically reducing project time and cost.

Using façade pattern provides administrators and business users a simple interface to manage the attributes for content usage logging. A GUI interface for selecting attributes from a page can be very helpful for business users with little technical knowledge. In future, we plan to extend our current implementation to include this facility.

Further, we plan on integrating some of the formal components reported in [26] and preliminary results in [8] into our implementation and then do a performance evaluation of the effects of our Web content usage logging system plug-ins in the overall performance of our example e-commerce server. This will provide empirical data for the fine-tuning of our architecture and the implementation.

REFERENCES

[1] [1] T. Friedman and K. Strange, "BI and Data Warehousing Architecture Key Issues", *Gartner Research Reports*, March 2003.

[2] J. Srivastava, R. Cooley, M. Deshpande, P. Tan, "Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data", *ACM Special Interest Group on Knowledge Discovery in Data and Data Mining Explorations*, Jan 2000.

[3] B. Mobasher, H. Dai, T. Luo, Y. Sun, and J. Zhu, "Integrating Web Usage and Content Mining for More Effective Personalization", *1st International Conference on Electronic Commerce and Web Technologies*, London-Greenwich, United Kingdom, September 2000.

[4] A. Buchner and M. Mulvenna, "Discovering Internet Marketing Intelligence Through Online Analytical Web Usage Mining", *SIGMOD Record*, (4) 27, 1999.

[5] R. Cooley, B. Mobasher, and J. Srivastava, "Data Preparation for Mining World Wide Web Browsing Patterns", *Journal of Knowledge and Information Systems*, 1, 1999.

[6] O. Zaiane, M. Xin, and J. Han, "Discovering Web Access Patterns and Trends by Applying Olap and Data Mining Technology on Web Logs", In *Advances in Digital Libraries*, page 19-29, Santa Barbara, CA, 1998.

[7] Sing Li, "Filtering Tricks for your Tomcat: The Addition of Filtering to the Servlet 2.3 Spec Offers Enhanced Performance for your J2EE Apps", *IBM DeveloperWorks Technical Article*, June 2001.

[8] Jinbo Zheng, *Data Warehousing for Electronic Commerce*, Master Thesis, University of Manitoba, 2004.

[9] Adiele C. and Ehikioya S. A., "Evolving a "Wise" Integration System for E- Commerce Transactions", *Journal of Electronic Commerce Research and Application*s, Volume 6, No. 2, pp: 219 – 232, 2007.

[10] Ehikioya S. A. and Adiele C., "A Formal Model of Dynamic Identification of Correspondence Assertions for E-commerce Data Integration", *International Journal of the Computer, The Internet and Management (IJCIM),* Volume 13, # 2, pp 52-66, 2005.

[11] Adiele C. and Ehikioya S. A., "Algebraic Signatures for Scalable Web Data Integration for Electronic Commerce Transactions", Journal of Electronic Commerce Research, Vol. 6, #1, pp.56-74, 2005.

[12] Ehikioya S. A. and Adiele C., "Algebraic Signatures to Analyze Correspondence Assertions for Web Data Integration", International Journal of Computer and Information Science (IJCIS), Vol. 5, No. 3, 2004.

[13] Ehikioya S. A., "A Formal Characterization of Electronic Commerce Transactions",International Journal of Computer and Information Science, Vol. 2, No. 3, 2001.

[14] Jiawei Han, Micheline Kamber, and Jian Pei, Data Mining: Concepts and Techniques, Third Edition, Morgan Kaufman, 2011.

[15] Ian H. Witten, Eibe Frank, Mark A. Hall, and Christopher J. Pal, Data Mining: Practical Machine Learning Tools and Technologies, Fourth Edition, Morgan Kaufman, 2016.

[16] Gennadiy Zlobin, Chapter 1: "Model View Controller", in Learning Python Design Patterns, Packt Publishing, Nov. 2013

[17] Michael, "Monetizing Customer Usage Data", IoT Agenda, October 7, 2016. (Available at: http://internetofthingsagenda.techtarget.com/blog/IoT-Agenda/ Monetizing-customer-usage-data). Accessed on March 16, 2018.

[18] Sing Li, "Filtering Tricks for your Tomcat: The Addition of Filtering to the Servlet 2.3 Spec Offers Enhanced Performance for your J2EE Apps", IBM DeveloperWorks Technical Article, June 2001.

[19] "Tomcat Internals - The Design of Tomcat 3.x and the Reasons Behind It", (Available at: http://jakarta.apache.org/tomcat/tomcat-3.3-doc/intemal.html), 2000.

[20] Gennadiy Zlobin, "Exploring Model View Controller", November 2013. (Available at: https://www.packtpub.com/books/contents/exploring-model-view-controller). Accessed on March 18, 2018.

[21] Ceki Gulcu, The Complete Log4j Manual: The Reliable, Fast and Flexible Logging Framework for Java, QOS.ch, May 2003.

[22] Samudra Gupta, Pro Apache Log4j, 2nd Edition, Apress, November 2014.

[23] Apache Log4j 2 Package (Available at: https://logging.apache.org/log4j/2.x/). Accessed on March 18, 2018.