

UNIVERSITY OF NAIROBI



SCHOOL OF COMPUTING AND INFORMATICS

**HOLISTIC APPROACH FOR EFFICIENT
EXTRACTION OF WEB DATA**

BY

MALEKIA DIDAS

P58/70606/2007

SUPERVISOR: ANDREW MWAURA

OCTOBER, 2011

**Submitted in partial fulfillment of the requirements of the Master of Science in
Computer Science**

University of NAIROBI Library

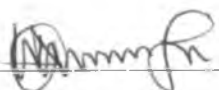


0439216 3

DECLARATION

This project, as presented in this report, is my original work and has not been presented for any other University award.

I understand that cheating and plagiarism constitute a breach of University regulations and will be dealt with accordingly.

Sign: _____

Date: 17-10-2011

Malekia Didas
P58/70606/2007

This project has been submitted as partial fulfillment of the requirements for the Master of Science degree in Computer Science of the University of Nairobi with my approval as the University supervisor.

Sign: _____

Date: 19/01/2011

Mwaura Andrew
Project supervisor
School of Computing and Informatics
University of Nairobi

ABSTRACT

There is a tremendous growth in the volume of information available on the internet, digital libraries, new sources and company database or intranets that contain valuable information. Information from World Wide Web has been a source of information which caters for different sectors ranging from social, political and economical spheres for decision making. Such information would be more valuable if it can be available to the end user and other application systems in required formats. This has caused the need for tools to assist users in extracting relevant information in a fast and effective way. We explore an efficient mechanism of extracting web data through analysis of HTML tags and patterns. HTML constitutes a large percentage of web content. However, much of this content lacks strict structure and proper schema. Additionally, web content has high update frequency and semantic heterogeneity of the information as compared to other format such as XML that are more firm in structure. We have managed to produce a customised generic model that can be used to extract unstructured data from the web and populate it to a database. The main contribution is an automated process for locating, extracting and storing data from HTML web sources. Such data is then available to other application software for analysis and other processing.

Keywords: Web data extraction, structured data, semi structured and unstructured data.

ACKNOWLEDGEMENT

First of all, I would like to pay my sincere regards to my family members who have taken an utmost concern and intense care for my research and my academic achievements in general. I pay them my deepest regards for all their inspiration and love, which has always remained invaluable to me, in each and every step of my life.

I would like to appreciate the assistance I got from various people to make my project successful. I would like to extend gratitude to the following: my supervisors and advisors, Mr. Andrew Mwaura for his excellence guidance and suggestions in various stages in this project. Dr. Wanjiku Nganga my second supervisor for her constructive ideas on how to organise my report. Also I would like to express my heartfelt appreciation and thanks to Mr. Shishiwa and Mr Njaala for their tireless positive and constructive contributions and proof reading of this report.

I would like to thank all lecturers through the entire course. Your knowledge and experience in the ICT industry opened my mind to new ideas and allow me to have confidence and dream big.

Last but not least, special thanks to my wife Audrey for her continuous support, patience, love and encouragements that helped me to finish my studies

TABLE OF CONTENTS

DECLARATION	ii
ABSTRACT	iii
ACKNOWLEDGEMENT	iv
LIST OF FIGURES	ix
LIST OF ABBREVIATIONS	x
CHAPTER ONE: INTRODUCTION	1
1.1 Background to the Problem	1
1.2 The Problem Statement	2
1.3 Purpose of the Project	4
1.4 Rationale/Justification	4
1.5 Research Questions/Objectives/Hypotheses	5
1.6 Assumptions and Limitations of the Research	6
1.7 Project Scope	6
1.8 Research Outcomes and their Significance to key Audiences	7
1.9 Definitions of important Terms	8
1.10 Overview of Dissertation	10
CHAPTER TWO: LITERATURE REVIEW AND THEORY	12
2.1 Introduction	12
2.2 Literature Review	12
2.3 Theories	15
2.4 Evolution of Data Extraction System (wrappers)	33
2.5 Conceptual Design	36
CHAPTER THREE: METHODOLOGY	37
3.1 Introduction	37
3.2 Research Methodology	37
3.3 Research Design and its Justification	38
3.4 Sources of Data/Information and Relevance of Data to the Problem ...	39
3.5 Tools, Procedures and Methods for Data Collection	39
3.6 Data Analysis Methods and their Justification	40
3.7 Limitation of Methodology and how they are overcome	40
3.8 Summary	41

CHAPTER FOUR: ANALYSIS AND DESIGN	42
4.1 Introduction	42
4.2 Algorithm Analysis and Design.....	42
4.3 Functional and Non-Functional Requirements	51
4.4 Architecture and Design.....	62
4.5 Summary.....	71
CHAPTER 5: IMPLEMENTATION.....	72
5.1 Introduction	72
5.2 Technologies	72
5.3 Working environment	74
5.4 Page analyser	74
5.5 Data Extractor	76
5.6 Data storage.....	77
5.7 User Interface	78
5.8 Summary.....	79
CHAPTER SIX: EXPERIMENTAL RESULTS	80
6.1 Introduction	80
6.2 Testing strategies for system development	80
6.3 Experimental Analysis	83
6.4 Methodology for Comparison of the two Systems.....	84
6.5 Performance Metrics	85
6.6 Algorithm Implementation	86
6.7 Accuracy Analysis	87
6.8 Time Complexity Analysis	89
6.9 Execution Time Analysis	91
6.10 Summary.....	92
CHAPTER SEVEN: DISCUSSION.....	93
7.1 Introduction	93
7.2 The main Findings and Observations.....	93
7.3 Exceptions.....	95
7.4 Relationship to previous work	95
7.5 Theoretical or practical Implications	95
7.6 Achievements.....	96
7.7 Constraints.....	97

CHAPTER EIGHT: CONCLUSION AND RECOMMENDATIONS	98
8.1 Introduction	98
8.2 Summary of the Results	98
8.3 Generalization of the Results	98
8.4 Applicability of the Results	98
8.5 Conclusions	99
8.6 Recommendation for further Studies	100
REFERENCES AND BIBLIOGRAPHY	103
APPENDICES	105
APPENDIX A: Data Extraction system installation and running guide	105
APPENDIX B: Data Extraction system source code listing	117
APPENDIX C: Glossary	122

LIST OF TABLES

Table 1: The comparison between surface web and deep web.....	18
Table 2: Major search engines	21
Table 3: Derivates of major search engines	21
Table 4: Major Directories	22
Table 5: Popular metasearch engines	22
Table 6: Knowledge Engineering Approach vs Automatic Training Approach	31
Table 7: A comparison of the strengths and weaknesses of SDLC	40
Table 8: Data Extractor.....	45
Table 9: System use-case	53
Table 10: Use-case Specification	54
Table 11: Functional Requirements.....	61
Table 12: Description of the fields of central table.....	64
Table 13: Field Names	64
Table 14: System Component Description.....	65
Table 15: Database Operations	68
Table 16: Comparison between Happy Harvester and Cluster system	87
Table 17: Comparison between Happy Harvester 2 and Cluster system	88
Table 18: Data Extractor	90
Table 19: Execution Time Analysis.....	91
Table 20: Web domain tested and the quality of the results produced.....	96

LIST OF FIGURES

Figure 1: Scope of the system	7
Figure 2: Taxonomy of Information Extraction	15
Figure 3: A diagram of the standard model of the information access processes	32
Figure 4: Proposed extraction system	36
Figure 5: Data flow for the system	50
Figure 6: Use case Diagram	53
Figure 7: System Architecture	63
Figure 8: System Component	65
Figure 9: Sequence Diagram	71
Figure 10: Page Analyser.....	75
Figure 11: Data storage	77
Figure 12: Some of the method tested.	80
Figure 13: Bottom up Integration	82
Figure 14: Methodology for Tools Comparison.....	84
Figure 15: Execution Time Analysis in graph	92
Figure 16: Unique tag Vs HTML Tags	94
Figure 17: Running time Vs Number of HTML Tags	94

LIST OF ABBREVIATIONS

GUI	Graphical User Interface
HTML	HyperText Mark-Up Language
IE	Information Extraction
IR	Information Retrieval
SDM	System Development Method
SQL	Structured Query Language
XML	Extensible Markup Language

CHAPTER ONE: INTRODUCTION

There is a growing tendency of many companies to depend on web as source of data and information for their daily use for the purpose of remaining competitive on this global market. Many researchers have been working on the different ways to make the use of web data to be available to the end users and other software application in automated ways. This section introduces the research problem and how it will be solved during this project work. It gives an overview on how the project was conducted in order to solve the research problem and indicates the limitations and scope of the project. Lastly the project organization is also presented.

1.1 Background to the Problem

World Wide Web is the largest database which contains various data that human being would like to consume for their needs. As a result of such massive growth of information on the web sources, it should be made available to the end users and other application such as e-commerce. However, the information in the web pages to the large extent is presented by HyperText Markup Language tags, which is used mainly for presentation purposes and not for the automation process of extracting of such data. Also HTML pages can be generated by hand or HTML generator tools, which results into lack of schema and ill-formatting pages that make human-friendly, not machine-friendly. Such a weakness makes the information from the web pages not available for computer application processing.

In order to make the information from the web sources available to the computer application, there is a need to make a clear separation between content and style by extracting and presenting data from web sources and put it into a suitable structured format, since HTML pages consist of either unstructured or semi structured data. There are various methods of information extraction from HTML documents which can be grouped into manual approach, supervised learning and automatic techniques. Each of these categories has advantages and disadvantages; for example, the manual method has high precision and recall values but difficult to apply to large number of pages. As a result, extracting data manually is error prone, tedious and impossible for huge amount of data. Supervised learning involves human interaction to create positive and negative samples which will result into good performance compared to manual approach. Lastly,

automatic techniques yield less human effort but they are not highly reliable regarding the information retrieval process.

HTML based web sites make vast majority of the web content. HTML technology has large volume compared to XML technology. XML technologies are more suitable for systems which are not compatible, since they separate web data from presentation which will make the information available for application processing. Therefore, there is a need to extract data from web source which is both unstructured and semi-structured into structured format which can be easily available to the end users or application programs for further processing.

This research will add value to already existing systems such as monitoring systems, notification systems and price comparison systems. It recommends the technique to automatically access the information from the corresponding web sources with less human intervention. Currently there are several technologies which uses XML technology for information exchange and processing. These technologies include Really Simple Syndication (RSS) for news updates and Simple Object Access Protocol (SOAP) for information exchange and inter process communications.

1.2 The Problem Statement

With explosions of web a lot of data is available online. This data range from unstructured, semi-structured to structured. Searching and extracting data in these innumerable formats is a big challenge for many businesses since such information is not organized or catalogued at one central location and there is a tendency of data change with time and also each industry has its own way of presenting information online. For example, finance industry to extract stock information, real estate industry where to extract property information and retail industry to extract different products and prices. The existing methods of information extraction from HTML documents include manual approach, supervised learning and automatic techniques. The manual method has high precision and recall values but it is difficult to apply to a large number of pages. Supervised learning involves human interaction to create positive and negative samples. Automatic techniques benefit from less human effort but they are not highly reliable regarding the information retrieved, Cosulschi Mirel et al (2006). Full automation is not always necessary in order to produce a useful tool, and may be undesirable in tasks requiring human judgment, Degbelo and Matongo (2009). In general the above methods have been characterized as manual, slow, complicated, delivering poor results and

dependent on only one type of web. Also they have been only applicable to surface web which is only 1% of the deep web. Although there are techniques used to reach web data such as browsing and keyword searching, there is no guarantee that users will be able to extract information even after knowing where to look for information by knowing its URLs. Web is designed as source of data for a human use, not for automation process where other application can benefit from it. This is because there is large amount of irrelevant information and the semantic meaning of different parts of an HTML document that may be encoded in ways that do not correspond in a simple way to a structured representation of data. Other factors, be web data are changing which make extracting data from such a web become difficult. Hence this problem motivates the need for the new information extraction techniques which will automate the task of locating data, extracting them and storing them for further application or decision making, Shaker Mahmoud et al (2010) . According to Agbogun (2010), the deep web contains 99% of information content of the web. However, most of this information is contained in databases and is not indexed by search engines. A complete approach to conducting this research is on the web which incorporates surface and deep web databases (holistic way) with a main focus on deep web since surface web has been well exhausted by other researchers. Most users of the internet are skilled in at least elementary use of search engines but the skill in accessing the deep web is limited to a much smaller population. According to a study by Bright (2005), the deep web is estimated to be up to 550 times larger than the 'surface web' that is accessible through traditional search engines and over 200,000 database-driven websites are affected by the problem. Gary and Sherman (2001), estimate the amount of quality pages in the deep web to be 3 to 4 times more than those pages accessible through search engine like Google. While the actual figures are debatable, it makes it clear that the deep web is far bigger than the surface web, and it is growing at a much faster pace. This is a problem that need to be attended to and that is the focus of this study. Hence this study will propose the efficient technique and develop a prototype which will extract any type of data irrespective of industry or formats. This technique will be semi-automatic which requires human intervention where necessary since the objective is to automate as much as possible the process of information extraction and at the same time give some flexibility to the users in the expression of their information need. This technique has been proposed by Robinson (2004).

1.3 Purpose of the Project

The broad purpose of this study is meant to aid web users to extracting higher quality information in less time from the deep web (surface web and deep web). This is because the web consists of two parts: the surface web and the deep web (invisible web or hidden web) but the deep Web impose a lot of challenges for locating and extracting data. Many extraction systems have been experiencing such challenges.

The specific purposes of this project is to implement and test a mechanism to automate the task of locating, extracting and storing the data from the website in order for such a mechanism to be used in any website without any further modification.

1.4 Rationale/Justification

There is progress in the utilisation of web data for various uses. Many researchers have been working on this field of data extraction by deploying various techniques and methods which in turn have been proved to work properly to some extent. Also, it has been used to real working system as explained above. But some of the methods and techniques involve human intervention in discovering the location of the data to be extracted from the web pages. This makes the work of extracting data more difficult and not reliable when the structure of the web page is changed. This research, therefore, puts effort and adds a value to web data extraction by examining in detail a method which has been proposed before for extracting web data and see what its weaknesses are. Then, the study addresses such weaknesses so that we can have 100% web data extraction. Therefore, this project automates the task of extracting the data from various sources so as to allow efficient extraction of data at the right time.

Besides, the second motivation for undertaking this project is that. The web is now over 20 years old. The information on the web has grown exponentially as result on probably every topic you can think of; there is some information available on some web page. As a result of such growth it is still very hard to find relevant information because the query interface to search engines has not changed since the early days of the web. Hard queries result into problem such as difficulties of finding the pages that contain relevant information, difficulties of extracting the relevant pieces of information from these pages, difficulties of connecting the information that is extracted from the pages into database and the problem of applying common-sense reasoning in all phases.

1.5 Research Questions/Objectives/Hypotheses

This research is being guided by the following questions:

- (i) What is the performance of various techniques for web data extraction and storage?
- (ii) What are the deep web problems that affect search engines, websites and searchers?
- (iii) What are the performance metrics for measuring the data extraction algorithm or technique?
- (iv) What are the strategies to improve the efficient and effectiveness of proposed technique?

The main objectives of this research are:

- (i) To investigate the performance of available web data extraction and storage techniques.
- (ii) To identify the requirements for data extraction and storage system.
- (iii) To explore the mechanism of identifying data clusters from the result web page which has the data that to be extracted.
- (iv) Evaluate the effectiveness of clustering algorithm in retrieving the most relevant documents based on user query
- (v) To explore the mechanism of storing the data to the databases.
- (vi) To propose techniques that will be efficient for data extraction and storage process. The techniques will eliminate manual work, slowness, poor result and lack of independence among web pages. The technique should be independent such that it can be used to any web site without modification.
- (vii) To test the proposed techniques for its effectiveness and efficiency by developing a prototype for information extraction that can automatically extract the appropriate information from the web pages and store it in the database. Such a model should be general in such a way that it can be applied into different website.

The following research hypotheses will also guide the research work:

- (i) The volume of web data from web sources in HTML technology is very big compared to that of XML technology. In this case, it is important to find a way of extracting the data from the existing legacy web document (unstructured or semi-structured HTML documents) to structured format. In so doing, the information

from the unstructured web data can be easily available to the end users or application programs.

- (ii) Web data extraction system such as searching engine, shopping agents, RSS, SOAP and other extraction systems rely on XML technology which deals with structured data. This results on false positive and false negatives to users query since there is small percentage of structured data, large percentage relying on semi-structured and unstructured.

1.6 Assumptions and Limitations of the Research

This research has the following assumptions and limitations which have been imposed by both time and funding

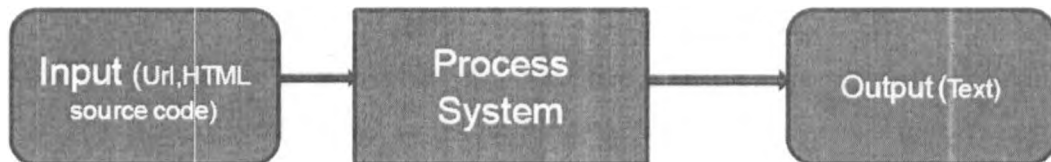
- (i) The software for data extraction which relate to the author research is commercial and very expensive. It was not available for free for full comparison with author prototype. This made the author to use its demo for comparison and assume that those few features could be enough for generalization hence reflect the complete functionality of it (Happy Harvester 2).
- (ii) The developed prototype cannot include one feature of integrating it to search engine so that it can help the user to search for required web domain before extracting the data due to time constraint. However, it is assumed that the user knows where she/he want to extract data either by using search engine strategies or he/she knows the URL of respectively web domain where data should be extracted. Hence the main focus is how to extract and store such a data.

1.7 Project Scope

The project focuses on demonstrating the powerfulness of proposed techniques by developing the prototype for data extraction. This employs clustering method using Java as a programming language and Msql as the database. Also the focus is to automate the task of locating, extracting and storing the data in the database from the deep web since surface web have been excellent been accessed by crawlable. This project will not deal with techniques for locating the web domain where data should be extracted. Instead, we assume that, the user knows it by using searching engine strategies. If the user knows the URLs of deep web databases and understands what information contained in these databases then she/he will be able to access the deep web information. The proposed technique helps an organization/individual that intends to extract data from the web to do

it in an efficient way (less human intervention, fast and in accurate way). The technique has been tested on the web domains with searching functionalities (this provides an opportunity to deal with dynamic web pages whose content changes as searching facilities access its database with corresponding keyword search). The input to the system is either URLs of respective websites or its source code and output is the text that is stored in database.

Figure 1: Scope of the system



The focus of this thesis is on HTML technology. The reasons include:

- (i) HTML is the predominant markup language for web pages.
- (ii) It is a kind of semi-structured data.
- (iii) Its information follows nested structure.
- (iv) It support World Wide Web consortium (W3C).
- (v) The language in all situations (which has been explained) above occurs.
- (vi) The volume of web data from web sources in HTML technology is very big compared to that of XML technology.

1.8 Research Outcomes and their Significance to key Audiences

The outcome of this research is the efficient mechanism for extracting web data which:

- (i) Helps the user to extract data from the web by reducing the amount of time spent in reading the long text in the web. It also filters irrelevant information and avoids manual copying and pasting. This reduces costs and errors associated with data extraction. Also, it leads to reduction in both false positives (useless answers delivered that fail to fulfill user's need) and false negatives (useful answers that the system fails to deliver to the user) in answering a query.
- (ii) Helps the user to classify and organize documents so that they can be stored in the database according to their content for quick retrieval purposes, thus minimization of the query response time and overall processing cost.

- (iii) Adds value to the literature of data extraction and storage techniques by deploying the new techniques. The study on deep web is necessary because it focuses on the problems encountered by search engines, websites and searchers. More importantly, the study provides information on the results of searches made using both surface search engines and deep web search tools. Finally, it presents the deep web not as a substitute for surface search engines, but as a complement to a complete search approach that is highly relevant to the academia and the general public.
- (iv) Helps the design of prototype system that can be used by any user and any application so as to produce the data in a structured format such as relation database or XML format or the prototype system. This can be used as a separate component to be used by other systems to extract data from any web pages.
- (v) Adds value to the information retrieval system such as searching engine, RSS, online shopping system, price comparison system and monitoring system. The process used is data extraction either by applying the technique used or use the data that will be extracted and stored in the database by using the prototype system. This will direct user queries to appropriate servers by constraining the search space through query refinement and source selection. This leads to elimination of unnecessary communication overhead over the global networks and over the individual information sources.

1.9 Definitions of important Terms

The **surface web (visible web)** is that portion of the World Wide Web that is indexed by convention search engine. Surface web consist of pages that can be browsed using normal web browser. Surface web is searchable through popular search engines. The collection of reachable pages defines the surface web. Narayan (2010)

The **deep web (also called Deepnet)** refers to the World Wide Web contents that are not part of the surface web, which is indexed by search engines. It is estimated that deep web is several order of magnitude larger than the surface web. The term “deep web” refers to web pages that are not accessible to search engines because those web pages are dynamically generated in response to queries through web forms or web services. The existing automated web crawlers cannot index these pages, thus they are hidden from web search engines. Narayan (2010)

Keyword is a word or phrase entered in a query form that a search system attempts to match in text documents in its database. Narayan (2010)

The Internet is a networking protocol (set of rules) that allows computers of all types to connect to and communicate with other computers on the Internet. Agbogun (2010)

The World Wide Web (Web), Is a software protocol that runs on top of the Internet, allowing users to easily access files stored on the Internet computers. Agbogun (2010)

HTML, It is known as HyperText Markup Language, It is the main markup language for Web pages where it provides a means that describes the structure of text-based information in a document. Jacob (2005)

Hypertext is a system that allows computerized objects (text, images, sounds, etc.) to be linked together. Agbogun (2010).

Hypertext link points to a specific object, or a specific place with a text; clicking the link opens the file associated with the object. Agbogun (2010).

XML it is known as Extensible Markup Language, It is a general-purpose specification for creating custom markup languages. It consist a rules for encoding documents in machine readable form. Jacob (2005)

XHTML it is known as Extensible Hypertext Markup Language, It is a markup language that has the same depth of expression as HTML, but also conforms to XML syntax. Jacob (2005)

RSS, is known as Really Simple Syndication, it is a family of web feed formats used to publish frequently updated works such as blog entries, news headlines, audio, and video in a standardized format. Jacob (2005)

Static web sites consist of HTML (hypertext markup language) pages that do not change unless the webmaster modifies the tags directly within the page. Static web sites are usually easier to develop but are very costly to maintain. They also often fall short on content because most of the information is not up- todate and is usually presented in general (because the pages are created for all users not for a specific user). The lack of content could eventually cause users to stop returning to the site on a regular basis. Agbogun (2010)

Dynamic web sites consist of HTML pages that are created on the web server before they are sent to the user. Most dynamic web sites use a relational database management system (RDBMS) to create the dynamic content. Dynamic web sites are usually more expensive to develop but cheaper to maintain, often full of content and timely information. Also, dynamic web sites are usually more popular because the information

displayed is regularly updated and can be customized specifically for users. The content may change according to the geographic location of the user, time of day, etc. Technologies for producing it include cgi scripts, server side includes (ssi), javascript etc. When capitalized, Dynamic HTML refers to new HTML extensions that will enable a Web page to react to user input without sending requests to the Web server. Microsoft and Netscape have submitted competing Dynamic HTML proposals to the Worldwide Web Consortium (W3C). Sites can include a combination of static and dynamic content. Information that does not change often is best created statically whereas information that changes often should be created dynamically. A web site that contains both static and dynamic content is usually the most cost-effective option in the long run. Agbogun (2010)

1.10 Overview of Dissertation

The contents of this thesis are enlisted as follows:

- Chapter 1 deal with introduction which contains the introduction to chapter one, problem statement, purpose of the project, project motivation, project scope, research outcomes and their significance to key audiences, research questions, research objectives, research hypotheses, assumptions and limitation of the research, definitions of important terms and lastly overview of dissertation.
- Chapter 2 deal with literature review and theory, which contains the background of the problem, researcher ideas that have contributed to formation of the thesis, theories that have contributed towards successful implementation of the project, which include web data extractor theories, web data extraction system and theory of data extraction process. Also the chapter concludes by including the conceptual design, which is a proposed system to be developed.
- Chapter 3 deal with methodology which contains different methods to the study. This includes research methodology, research design and its justification, source of data or information and their relevance to the problem, tool and procedure for data collection and its justification, data analysis method and its justification, limitations of methodology and how to overcome them and project schedule from the proposal which gives the details on how the project was carried out.

- Chapter 4 deals with analysis and design that is algorithm design and analysis, There is web page analysis and investigations on web page analysis for the purpose of detecting those problems which face data extraction process and how algorithm will be able to overcome them. It has separate algorithm for separate component and algorithm for the complete system. Also this chapter deals with functional and non functional requirements that will be provided by the system. Besides, it includes data extraction process perspective and function and user characteristics. Indeed the chapter includes architecture and system design which includes system architecture, database design, system component and component interaction.
- Chapter 5 deals with implementation. This includes the choice of the technologies for system implementation together with the reasons for the choices and the setting for the working environment of the system. This chapter also specifies how various components of the system have been implemented.
- Chapter 6 deals with testing of the system developed. This includes, testing strategies, experimental analysis, methodology used for comparison, performance metrics, algorithm implementation, accuracy analysis, time complexity analysis and execution time analysis
- Chapter 7 deals with discussion of results obtained. This includes main findings and observations, exceptions, relationship to previous work, theoretical or practical implications, achievements and the constraints of the project.
- Chapter 8 is the last chapter which is about the conclusion and future work. In this chapter the successes and failures of the system are discussed. It discusses much more on the conclusion and further researches to extend this project.

The dissertation ends with a list of references which have been used in this project together with the appendices which provide additional information of the dissertation. Appendix A is about data extraction system installation and running guide. Appendix B is about data extraction systems source code listing. Appendix C presents glossary.

CHAPTER TWO: LITERATURE REVIEW AND THEORY

2.1 Introduction

Information drives today's businesses and the internet is a powerhouse of information. Most businesses rely on the web to gather data that is crucial to their decision making processes. Companies regularly assimilate and analyze product specifications, pricing information, market trends and regulatory information from various websites but such a task is being performed manually. Many researchers have been trying to explore the best way of extracting web data and store it into the data warehouses or databases for further use. This chapter explores those techniques which have been proposed by various researchers by looking at their strengths and weaknesses. This chapter ends up by recommending and choosing one approach among the discussed ones to be extended so as to add value to data extraction.

2.2 Literature Review

The following sections present the literature reviewed from researches conducted by various researchers. The ideas from these researchers have contributed to formation of this project.

2.2.1 Degbelo and Matongo (2009)

Degbelo and Matongo (2009) point out that there is a very high growth of using electronically stored data in the web within organisations whereby human beings have been looking the ideal way for obtaining such a data from the web. According to Degbelo and Matongo (2009) the distribution of information within an organisation should be timely, selective and to some degree automatic whereby human beings should find some automatic system that should be able to select the right information for the right purpose.

2.2.2 Califf and Mooney (1999)

These researchers use the techniques of pattern matching whereby they build templates for the data to be extracted and later on the templates are filled with data from the result of web page queried by end user. The weakness of this method is that it is a manual system.

2.2.3 Kuhlins and Tredwell (2002)

Kuhlins and Tredwell (2002) use pattern recognition techniques based on the text constraints whereby data are extracted from the document depending on the predefined pattern. They describe and implement LAPIS toolkit whereby they treat the HTML web document as text document. Here with the help of the system, the user creates the pattern before the data is extracted. The weakness of this method is that, it is difficult to understand the toolkit and it involves a lot of manual work.

2.2.4 Myllymaki (2001)

Myllymaki (2001) points out that Andes framework uses the crawler technology and XML based data extraction techniques whereby there is restructuring of the HTML document to the XML specification before the document is parsed for data extraction. In this research, approach similar to Andes framework is used by considering only parts of HTML document with data clusters.

2.2.5 Embley (2005)

Embley model extracts data based on the extraction anthologies technique whereby the model extracts data based on the recognition and classification of data values from the web pages. The weakness of this approach is that it is limited to semi structured HTML web sources. This research extends also the ideas to unstructured HTML web sources

2.2.6 Hackathom (1998)

Hackathom (1998) describes four stages process to getting the web data into the data warehouse which is discovery, acquisition, structuring and dissemination. At discovery stage, the predefined pattern which shows the location of data is loaded and the HTML source code is parsed, then the information is ready for the stage of acquisition where data is structured and stored to the data warehouse. The weakness of this technique is that the whole process of identifying the location of data clusters is still manual. My research automates the process of identifying the location of data clusters.

2.2.7 Lam, Gong, and Muyeba (2008)

Lam, Gong, and Muyeba (2008) explains the history of extraction system by identifying evolution of extraction system and its correspond weaknesses. They later propose their own extraction system which belongs to the type of semi-automatic wrappers where it

utilizes the training process by learning rules for extraction with the assumption that with training, the system can be more adaptive to different types of pages if the training samples are broad enough. But yet there was human intervention for the training samples pages which mark the weakness of the method proposed.

The proposed system in this research belongs to the type of semi-automatic wrappers, where the use of repetitive pattern marking the existence of the data clusters is utilised. This technique belongs to clustering technique and cluster analysis has been playing an important role in solving many problems in medicine, psychology, pattern recognition and image processing. Hope if it is utilised well, it can solve the current problem of web data extraction

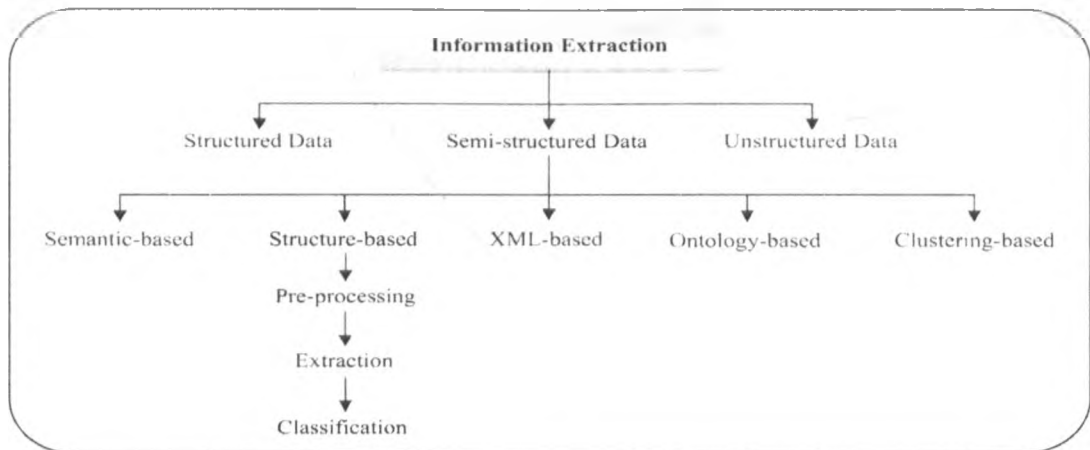
2.2.8 Robinson (2004) and Jacob (2005)

Robinson (2004) and Jacob (2005) point out the best way of identifying the data clusters and the best way the page descriptor information should be presented. Their researches have positive contribution to my research. Their ideas are extended in my research by exploring the best way to locate, extract and store the information by using their techniques. This study also explores the best way to make such information to be available to end users and other software application for further usage.

2.2.9 Shaker Mahmoud et al (2010)

Shaker Mahmoud et al (2010) propose a framework for extracting information from semi-structured web data source. This framework consists of three components, namely: (i) Query Interface (QI), (ii) Information Extraction (IE), and (iii) Relevant Information Analyzer (RIA). The weakness of their proposed technique is that, it depends only on structured-based data where single tag which is table tag is considered and the rest of tags are ignored.

Figure 2: Taxonomy of Information Extraction



Source: Shaker Mahmoud et al (2010)

In this research the above framework is utilised by exploring new area of clustering-based and the approach uses repetitive pattern which marks the existence of the data clusters as proposed by Robinson (2004) and Jacob (2005)

2.2.10 Happy Harvester 2 (2011)

Happy Harvester 2 is software for data extraction which utilises a technique which is close to my research. In this research it is used as the benchmark to assess my technique so as efficiency and effectiveness of my technique can be identified.

2.3 Theories

The following are theories that have contributed to successful implementation of this project.

2.3.1 Web Sources Theories

2.3.1.1 Functional Components of the Internet

According to Agbogun (2010) at present, the internet is functionally divided into two areas as follows:

- (i) The surface web which contains 1% of the information content of the web. Search engines crawl along the web to extract and index text from HTML. (HyperText Markup Language) documents on websites, then make this information searchable through keywords and directories.

- (ii) The deep web which contains 99% of the information content of the web. Most of this information is contained in databases and is not indexed by search engines due to some technical and business reasons which stand as obstacles. This information is made searchable by keywords only through the query engine located on the specific website of each database.

As the web evolves it is very certain that the deep web becomes more easily accessible. However, at present to access deep web information, one needs to go directly to the website containing the database of interest and use the website's query engine. To do this, you need to know the Uniform Resource Locator (URL) of the deep web site. Considering that there over 200,000 deep web sites, It is a challenge to know which site to use for a given research topic. Hence this project deployed a technique to explore the deep web.

2.3.1.2 The Nature of Web Sources

According to Narayan (2010) the web consists of two parts namely: Surface web and Deep web as described below.

Surface Web

The surface web (also known as the visible web or indexable web) is that portion of the World Wide Web that is indexed by conventional search engines. The part of the Web that is not reachable using this way is called the deep web. Search engines construct a database of the web by using programs called spiders or web crawlers that begin with a list of known web pages. The spider gets a copy of each page and indexes it, storing useful information that will let the page be quickly retrieved again later. Any hyperlinks to new pages are added to the list of pages to be crawled. Eventually, all reachable pages are indexed, unless the spider runs out of time or disk space. The collection of reachable pages defines the surface web. The surface web is mostly static in nature. Static web page means that for any new request of the page the same information is being presented. Most of techniques for extraction web data have managed to take advantage of surface web. The technique which has been implemented will take also advantage of surface web.

Deep Web

For various reasons (e.g., the Robots Exclusion Standard, links generated by JavaScript and Flash, password-protection) some pages can not be reached by the spider. These 'invisible' pages are referred to as the deep web.

The deep web (also called Deepnet, the invisible web, dark web or the hidden web) refers to World Wide Web content that is not part of the surface Web, which is indexed by standard search engines. Michael (2001), credited with coining the phrase, said that searching on the internet today can be compared to dragging a net across the surface of the ocean, a great deal may be caught in the net, but there is a wealth of information that is deep and therefore missing. Most of the web's information is buried far down on dynamically generated sites, and standard search engines do not find it. Traditional search engines cannot "see" or retrieve content in the deep Web – those pages do not exist until they are created dynamically as the result of a specific search. The deep web is several orders of magnitude larger than the surface web. A recent study in 2000, it was estimated that the deep web contained approximately 7,500 terabytes of data and 550 billion individual documents. Estimates based on extrapolations from a study done at University of California, Berkeley, show that the deep web consists of about 91,000 terabytes. By contrast, the surface web (which is easily reached by search engines) is only about 167 terabytes. The Library of Congress, in 1997, was estimated to have perhaps 3,000 terabytes. Most of deep webs are dynamic in nature. The dynamic web page is the web page whose contents changes for each new request of the web page.

2.3.1.3 Problems with Deep Web Sources

There are unabated growth of the web which has resulted in a situation in which more information is available to more people than ever in human history. Along with this unprecedented growth has come the inevitable problem of information overload. To counteract this information overload, users typically rely on search engines (like Google and the entire web) or on manually-created categorization hierarchies (like Yahoo! and the Open Directory Project). Though excellent for accessing web pages on the so-called "crawlable" web, these approaches overlook a much more massive and high-quality resource: The Deep Web (or Hidden Web) comprises of all information that resides in autonomous databases behind portals and information providers' web front-ends. Web

pages in the Deep Web are dynamically-generated in response to a query through a web site's search form and often contain rich content. A recent study by Narayan (2010) has estimated the size of the Deep Web to be more than 500 billion pages, whereas the size of the "crawlable" web is only 1% of the Deep Web (i.e., less than 5 billion pages). Even those web sites with some static links that are "crawlable" by a search engine often have much more information available only through a query interface. Unlocking this vast deep web content presents a major research challenge. In analogy to search engines over the "crawlable" web, we argue that one way to unlock the Deep Web is to employ a fully automated approach to extracting, indexing, and searching the query-related information-rich regions from dynamic web pages. In this project, we focus on three processes (extracting, indexing and querying the extracted data) from Deep Web. Extracting the interesting information from a Deep Web site requires many things: including scalable and robust methods for analyzing dynamic web pages of a given web site, discovering and locating the query-related information-rich content regions, and extracting itemized objects within each region. By full automation, we mean that the extraction algorithms should be designed independently of the presentation features or specific content of the web pages, such as the specific ways in which the query-related information is laid out or the specific locations where the navigational links and advertisement information are placed in the web pages, Narayan (2010).

The table below presents the comparison between surface web and deep web

Table 1: The comparison between surface web and deep web

Surface Web	Deep Web
It consist of Millions of web pages	Over 200,000 databases
It consist of 1 billion documents	550 billion documents
It consist of 19 terabytes	7,750 terabytes
As a weakness it has broad shallow coverage	As strength it has deep vertical coverage
As a weakness, its results contain additions	As strength, its results contain no additions
As a weakness, its content is unevaluated	As strength , its content is unevaluated by experts

Source: Agbogun (2010).

2.3.1.4 Properties of Deep Web

Deep web resources may be classified into one or more of the following categories Narayan (2010).

- (i) **Dynamic content:** dynamic pages which are returned in response to a submitted query or accessed only through a form, especially if open-domain input elements (such as text fields) are used; such fields are hard to navigate without domain knowledge.
- (ii) **Unlinked content:** pages which are not linked to by other pages, which may prevent web crawling programs from accessing the content. This content is referred to as pages without back links (or in links).
- (iii) **Private Web:** sites that require registration and login (password-protected resources).
- (iv) **Contextual Web:** pages with content varying for different access contexts (e.g., ranges of client IP addresses or previous navigation sequence).
- (v) **Limited access content:** sites that limit access to their pages in a technical way (e.g., using the Robots Exclusion Standard, CAPTCHAs, or no-cache Pragma HTTP headers which prohibit search engines from browsing them and creating cached copies).
- (vi) **Scripted content:** pages that are only accessible through links produced by JavaScript as well as content dynamically downloaded from Web servers via Flash or AJAX solutions.
- (vii) **Non-HTML/text content:** textual content encoded in multimedia (image or video) files or specific file formats not handled by search engines.
- (viii) Lastly Text content using the Gopher protocol and files hosted on the FTP protocol that is not indexed by most search engines. Engines such as Google do not index pages outside of the HTTP protocol.

2.3.1.5 How deep Web Problem affects Search Engines, Websites and Searchers

According to Agbogun (2010) the following are the effects of deep web

How does the deep Web problem affect search engines?

The search engines like Google, AltaVista and FAST claim to index the web, they are actually indexing a very small portion of it. Traditional search engines, using 'spiders' and 'crawlers', were designed to index simple HTML pages, that have incoming links from other pages on the web. But modern websites, operating databases to generate pages on-the-fly, are too sophisticated for these search engines to index their pages. So while these search engines do a very good job at indexing small sites and personal home pages, they cannot provide sophisticated sites such as eBay, Library of Congress and IMDB (Internet Movie Database) the means to expose their pages to searchers Agbogun (2010).

How does the deep Web Problem affect Websites?

Websites generated using dynamic pages generated by applications like CGI or ASP (Active Sever Page) are usually part of the deep web. The pages in these sites, while available to any human web user, are practically invisible to search engines. Affected sites, therefore, sustain a significant loss in targeted traffic, Agbogun (2010).

How does the deep Web Problem affect Searchers?

Naturally, a search engine that does not index most of the web is providing the user with a partial service. But the deep web is a bigger problem than that, the quality of information in the deep web is usually very high (compared to the surface web), because the deep web consists of the major authorative websites available on the net.

2.3.1.6 Search Tools

Various tools that help make researching more productive on both the surface web and the deep web as mentioned below, Agbonun (2010).

- **Search Engines**

Search engine can be said to be:

- (i) A software program that searches a database and gathers and reports information that contains or is related to specified terms.
- (ii) A website whose primary function is providing a search engine for gathering and reporting information available on the internet or a portion of the internet.

There are numerous search engines for the surface web. Which one should one use? If periodically one compares search results on various search engines, he/she may find that there is a little overlap between various search engines. For a thorough surface web search, you need to use multiple search engines. According to Search Engine Watch the following are the major search engines:

Table 2: Major search engines

About	Useful summary articles
Ask Jeeves	High relevancy searches, owns Teoma
Gigablast	Small but useful statistical result display
Google	Large crawler and directory
LookSmart	Human-compiled, owns WiseNut
Teoma	Ask Jeeves-crawler, high relevancy
Yahoo!	Crawler and tabs for images, video, etc.

Source: Agbonun (2010)

These are derivatives of the above search engines; they use the engines indicated:

Table 3: Derivates of major search engines

AllTheWeb	Bought by Yahoo
AltaVista	Yahoo-crawler and tabs
AOL Search	Google-crawler
HotBot	Ask Jeeves-crawler or Google-crawler
Lycos	LookSmart directory, Yahoo crawler
MSN	Yahoo crawler, Microsoft crawler pending

Netscape	Google-crawler
WiseNut	LookSmart owns

Source: Agbonun (2010)

- **Directory Browsing**

Directory browsing is another way of searching the surface Web. Directories are assembled by human beings who use editorial judgment to make their selections. To search directories, one clicks through a hierarchical set of hyperlinks. These are some of the major directories:

Table 4: Major Directories

Google
LookSmart
Yahoo!

Source: Agbonun (2010)

- **Metasearch Engines**

Metasearch engines search several search engines simultaneously and combine the results. Theoretically, you may get broader coverage in this way. Practically, you may lose precision because some metasearch engines cannot pass Boolean operators and most of the syntax does not work from the original engine. The following are popular metasearch engines, Agbonun (2010)

Table 5: Popular metasearch engines

Dogpile	Rated best
Kartoo	Visual output showing relations
Mamma	Crawlers, directories, specialty search sites
Vivisimo	Rated second best, organizes results

Source: Agbonun (2010)

- **Copernic Agent**

Copernic agent is a tool that is useful. It comes in three versions: freeware, personal, and professional. It searches using up to 90 search engines in 10 categories, it then combines results, eliminates duplicates, eliminates broken links and prioritizes the output. It installs

as a client on your computer and goes beyond what metasearch engines can do, Agbonun (2010).

- **Specialized Search Engines**

Specialized search engines search for databases by topic and help eliminate the “noise” associated with general search engines. Examples includes: Beaucoup, Search Enginez and Specialized Subject Indexes. Recall that there are over 200,000 databases on the Web. This specialized search engines are a big help in finding databases of interest to your research, Agbonun (2010).

2.3.1.7 Deep Web Search Tools

Various tools that help to research more productive on the deep web are mentioned below, Agbonun (2010).

- (i) **Complete Planet** uses a query based engine to index 70,000+ deep Web databases and surface Web sites. You can do a keyword search on all 70,000+ databases to find which databases to use for your search. You can also browse by category, and then search databases of interest.
- (ii) **ProFusion** is a combination of query based engine and a deep Web directory portal. The directory structure is accessed by clicking on Specialized Searches. With an account, you can setup custom “My Search Groups” to search customized lists of websites and/or databases of your choice. For example, you could create a group called Technology and add all the databases and websites of interest to you. This group is saved to your profile. You could then, at any future time, search this group on a research topic with keywords. This is a great time saver.
- (iii) **SurfWax** also uses a site's existing search capability as part of the meta-search process to tap the deep Web. They use proprietary algorithms to interpret the site's search criteria (Boolean, etc). With an account, you can also setup custom SearchSets to search customized lists of websites and/or databases of your choice. SurfWax also has a news accumulator feature with over 50,000 news topics in 84 categories. This news accumulator feature is a godsend providing high quality results. These are some useful news accumulator categories: all topics, networking, technology, telecommunication, and web services. In addition, this site has WikiWax which takes the online encyclopedia.

2.3.1.8 General Situation in the Web:

The following are facts about what is existing in the web:

- (i) Information of almost all types exists in the web such as structured, semi-structured and unstructured data, texts, and multimedia data.
- (ii) HTML pages can be written by hand or generated by some HTML generator tools.
- (iii) Web is designed as a source of data for a human use. It is built to guarantee that the content and the information could easily be understood and read by humans but not prepared to be used as data able to be treated by other applications. Since the trend has changed there is a challenge to structure this kind of information into the most appropriate way such that other applications or users can utilise it.
- (iv) Large percentage of information in the web is semi-structured due to the nested structured of HTML code.
- (v) Much of the information in the web is linked, since there are hyperlinks among pages within a site and across different sites.
- (vi) There is large percentage of information which is redundant since the same piece of information or its variants may appear in many pages.
- (vii) The web is noisy, since it consist of mixture of many kinds of information such as main contents, advertisements, navigation panels and copyright notices
- (viii) There is dynamic contents which are produced as result of submitted query.
- (ix) There is unlinked content where pages are not linked to by other pages. As result it can prevent web crawling programs from access the contents.
- (x) There is also private web which is password-protected resources, which requires registration and login.
- (xi) There is also contextual web whereby pages with content vary from different access contexts.
- (xii) There is limited access content which is sites that limit access to their pages in a technical way. As a result it may prohibit search engines from browsing them and creating cached copies.

- (xiii) There is scripted content, this includes pages that are only accessible through links
- (xiv) There is non-HTML or text content that is textual content encoded in multimedia files or specific file formats not handled by search engines.
- (xv) There is high growth of information in the internet, which increases exponential approx. There is 50 billion publicly accessible / index table web documents distributed all over the world on thousands of web servers and there are 1400 million of internet users as per January 2011.
- (xvi) The size of deep web is large compared to surface web, where surface web is 1% of the deep web where so far data extraction has been implemented.

The reasons for the above situation are as follows:

- (i) The web is a public product, no one own it, it is free accessed and it is difficult to impose rules upon it.
- (ii) There is emergence of web 2.0 technology where users are encouraged to contribute rich contents.
- (iii) Many companies, individuals want to be known by having the websites. Hence many pops up every day.
- (iv) There is high increase of users who possess devices that can access the internet such as mobile phone, notebook and personal computers.
- (v) There are social network such as facebook which promote the use of internet. as result users are encouraged to post notes.
- (vi) All hardware and software are cheaper to acquire. This promotes more people to possess it.
- (vii) Www is cheapest way get information from the web compared to other source of information because internet is cheaper.
- (viii) Since HTML pages can be written by hand or generated by some HTML generator tools, this encourages more users to design website and upload into free servers.
- (ix) There are free servers, which encourage users to own website since it is free to upload sites.

2.3.2 Web Data Extraction System

2.3.2.1 Introduction

A web data extraction system is a software system that automatically and repeatedly extracts data from web pages with changing content and delivers the extracted data to a database or some other application. Baumgartner, Gatterbauer, and Gottlob (2009). The task of web data extraction performed by such a system is usually divided into five different functions, Baumgartner, Gatterbauer, and Gottlob (2009): (1) web interaction, which comprises mainly the navigation to usually pre-determined target web pages containing the desired information; (2) support for wrapper generation and execution, where a wrapper is a program that identifies the desired data on target pages, extracts the data and transforms it into a structured format; (3) scheduling, which allows repeated application of previously generated wrappers to their respective target pages; (4) data transformation, which includes filtering, transforming, refining, and integrating data extracted from one or more sources and structuring the result according to a desired output format (usually XML or relational tables); and (5) delivering the resulting structured data to external applications such as database management systems, data warehouses, business software systems, content management systems, decision support systems, RSS publishers, email servers, or SMS servers. Alternatively, the output can be used to generate new web services out of existing and continually changing web sources.

2.3.2.2 Purposes of Web Data Extraction

According to Boronat (2008) the purposes of web data extraction are as follows:

- (i) Get information from the web to be used in other areas or by applications
- (ii) Information retrieval (e.g. Feeds Web search engines...)
- (iii) Let the user to access particular data from the Web
- (iv) Economical issues (e.g. stock market, shopping comparison...)

2.3.2.3 Main Problem during Data Extraction

According to Boronat (2008) internet is designed as a source of data for a human use. Problems appear when we want to extract data from HTML. The following are cases which make data extraction hard:

- (i) **Presentation of the Data without following a Structure**

Normally the content of a Web page is presented following structured patterns. This structure supplies the user an easy and logic way to find the information avoiding to waste his time. A good structure helps the data extraction tools to realize a good work. A suitable example could be a scenario of a digital newspaper. In this scenario we can find a table that contains all the news ordered by time of success. Each row is composed of a headline and a brief description of the news. This is simple way to structure if represented on a tree. It can be seen that some elements are appearing repeatedly. This helps our tools to extract the information. Lets think of the opposite example; a digital newspaper that doesn't use a main table with all the news and it doesn't follow a rule to present the information. It means some news could have photos, others videos and the information will be presented in a cell of a specified size and location that makes a nice end-view to the user. This kind of structure has more possibilities to generate problems to our data extraction tools, Boronat (2008).

(ii) Bad constructed HTML source documents

A well-built HTML document must follow some rules. Although most of the browsers could visualize the content of a page having some errors in the structure, it is highly recommended to follow the W3C standard of HTML. Some of these errors could consist of bad placed tags, repeated tags without sense and no closed tags. All these kind of mistakes could make harder our data extraction, Boronat (2008).

(iii) Nested data elements

These kinds of elements nest data and then element by element could appear different. We want to extract the part of information that is related to the auction. What happens here is that the second element is not new and then this type of information is displaced to the beginning which produces errors. Similar examples of this kind could be found on the Web, Boronat (2008).

(iv) Problem of choosing the correct Web page source example

This problem can be shown choosing a Web page whose content structure could change depending on some factors. One real example of this kind is the resulting page of Web search engines. If we perform a search using an input value we can get a result page with some entries. Depending of this value, this resulting Web page will change. We can get some image snapshots, video snapshots or some advertising related to this value. If the structure changes depending of this value, we can not use our data extraction tool with all the possible values to be sure that it uses exactly the best source. Because of this fact, we can say it is really important in this kind of pages to select a good sample to assure that

we are going to produce the minimum number of errors during the data extraction process, Boronat (2008).

(v) Problem of using scripts or dynamic content

Our data extraction tools read the HTML code to perform extractions. All the static content is written in HTML, it doesn't occur when speaking about dynamic content; such as Javascript, AJAX or Flash. Our data extraction tools cannot parse or treat all this information like with normal HTML. It doesn't follow the same syntax. Sometimes it has to be pre-processed before displaying a result or others. The result is only visual or changes could be introduced at any time the page is loaded. Some of our tools have support to treat dynamic content, especially Javascript, but often this kind of content generates difficulties to perform data extractions, Boronat (2008).

2.3.2.4 Type of Contents that can be extracted

According to Boronat (2008) the following types of content can be extracted from the web:

- (i) Free text (Unstructured Data): This type of text could be found in natural language texts, for example magazines or pharmaceutical research abstracts. Patterns involving syntactic relations between words or semantic classes of words are used to extract data from this type of sources.
- (ii) Structured text: This type of text is defined as textual information in a database or file following a predefined and strict format. It contains sufficient structure to allow unambiguous parsing and recognition of information where simple techniques are sufficient for extracting information from it. To extract this kind of data we have to use the format description.
- (iii) Semi-structured text: This type of text is placed in an intermediate point between unstructured collections of textual documents and fully structured tuples of typed data. To extract data we use extraction patterns that are often based on tokens and delimiters, for example the HTML-tags.

2.3.2.5 Ways to perform Data Extraction

According to Boronat (2008) the following are the ways of performing data extraction from the web:

(i) Manual extraction of the data

Manual extraction is the most precise option to extract data as we directly choose the data fields of our interest. The necessity to treat elements in an individual way takes a lot of time when treating large amount of data and hence makes to rule out that this option as not viable. This could be a good option for small and concrete data extractions. However, it is not the most common scenario when talking about Web data extractions. For these reasons, these extractions should be performed in a more automatically way.

(ii) Use a built API

API belongs to the owner of the Web page where we want to extract data. Normally, we can find APIs in few specific numbers of Web pages and its use and supply are limited by the specifications of the owner. To use them we have to take a look at the documentation and the method list of the owner.

(iii) Use wrapper

It is a procedure that is designed for extracting content of a particular information source for delivering the content of interest in a self-describing representation. The construction of a wrapper can be done manually or by using a semi-automatic or automatic approach. The manual generation of a wrapper involves writing of ad-hoc code. The creator has to spend quite some time understanding the structure of the document and translating it into program code. The task is not trivial and hand coding could be tedious and error-prone. On the other hand, semi-automatic wrapper generation benefits from support tools to help design the wrapper. By using a graphical interface, the user can describe which the important data fields to be extracted are. A specific configuration of the wrapper should be done for each Web page source as the content structure varies from each other. Expert knowledge in wrapper coding is not required at this stage, and it is also less error prone than coding. On the other hand, the automatic wrapper generation uses machine-learning techniques. The wrapper research community has developed learning algorithms for a spectrum of wrappers. This kind of wrapper requires a minimum intervention of human experts and systems which go through a training phase, where it is fed with training examples, and, in many cases, this learning has to be supervised.

2.3.2.6 Data Extraction process

According to Boronat (2008) the following are steps to extract information using a wrapper:

- (i) Load the information from the source page.
- (ii) Transform the source page into its posterior treatment.
- (iii) Identify the appearing elements.
- (iv) Filter these elements.
- (v) Export the final data to an output format.

The first and last steps are common to all types of wrappers as we need a data input and a data output to perform a data extraction but depending of the used wrapper type, the intermediate steps could vary.

2.3.2.7 Information Extraction (IE)

Degbelo and Matongo (2009) define IE as the process of picking information from the information source such as document collection in the web and using it afterwards for a purpose. There are various techniques for extracting information from the documents. Degbelo and Matongo (2009) argue that approach to the problem of information extraction can be differentiated from each other along a number of dimensions which are theoretical and practical. For theoretical side point of view, information can be extracted based on the knowledge engineering approach and the automatic training approach. On the practical side, information extraction can be based on those systems that require human intervention at run time and those that require little or no intervention. According to Degbelo and Matongo (2009), full automation is not always necessary in order to produce a useful tool, and may be undesirable in the tasks which require human judgement. It becomes difficult to decide which approach to adopt to solve a specific problem given the above arguments, but according to Degbelo and Matongo (2009), there is a guideline which someone can use to decide which approach to adopt as shown below.

Table 6: Knowledge Engineering Approach vs Automatic Training Approach

Use Knowledge Engineering Approach when	Use automatic Training Approach when
<ul style="list-style-type: none">• Resource (e.g. lexicons, lists) are available• Rule writers are available• Training data is scarce or expensive to obtain• Extraction specifications are likely to change• Highest possible performance is critical	<ul style="list-style-type: none">• Resources are unavailable• No skilled rule writers are available• Training data is cheaper and plentiful• Extraction specifications are stable• Good performance is adequate for the task

Source: Degbelo and Matongo (2009)

2.3.2.8 Query Construction

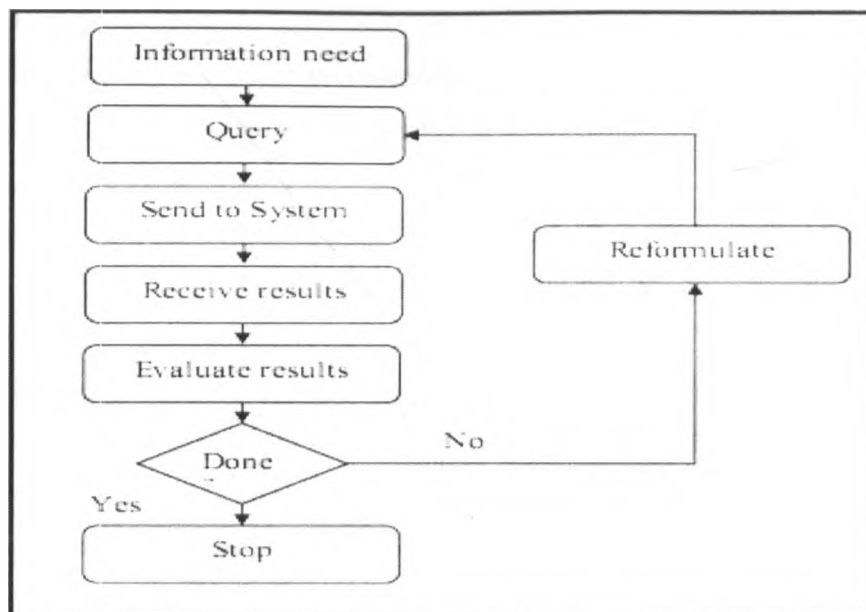
According to Degbelo and Matongo (2009) query is the formulation of user need. There are three methods for presenting a query:

- (i) Choosing parameters from a menu: in this method, the database system presents a list of parameters from which the user can choose according to his needs.
- (ii) Query by Example (QBE): here the system presents a blank record (a kind of form) and lets the user specify the fields and values that define the query.
- (iii) Query language: is defined as a programming language for formulating queries for a given data format. Query languages exist for both databases and information systems. This is the most complex method because it forces the user to learn a specialized language, but it is also the most powerful.

2.3.2.9 User Interface

User Interface (UI) is the program that allows the user to follow the information access process and therefore to interact with the system which can be a computer, a database and information system. Information access process allows user to building a query, sending this query into an information system or database that would retrieve relevant documents and send them back to user later on. Below is the information access process as proposed by Degbelo and Matongo (2009).

Figure 3: A diagram of the standard model of the information access processes



Source: Degbelo and Matongo (2009)

According Degbelo and Matongo (2009) the user interface, in order to be acceptable by the user and complete the user's need, should fulfil some requirements such as;

- (i) The User Interface should allow the user to reassess his goals and adjust his strategy accordingly: it will be useful when the user encounters a 'trigger' that causes him to use a different strategy temporarily.
- (ii) The UI should support search strategies by making it easy to follow tracks with unanticipated results. This can be achieved, in part, by supplying ways to record the progress of the current strategy and to store, find, and reload intermediate results.
- (iii) An important aspect of the interaction between the user and the UI is that the output of one action should be easily used as the input to the next.

User Interface can be in the form of graphical or command line. In this research the user interface will be in the graphical form to allow flexibility of the system to be used by both expert and non expert since a good user interface design can make a product easy to understand and use, which results in greater user acceptance.

2.4 Evolution of Data Extraction System (wrappers)

Introduction

In the very beginning, a wrapper is constructed to manually extract a particular format of information. However, the wrapper is not adaptive to change. It should be reconstructed accordingly to different types of information. In addition, it is complicated and knowledge intensive to construct the extraction rules used in a wrapper for a specific domain. Therefore, only experts may have knowledge to do that. No doubt, the inflexibility and the development cost for construction are the main disadvantages of using wrappers. Due to the extensive work in manually constructing a wrapper, many wrapper generation techniques have been developed. Those techniques could be classified into several classes, including language development based, HTML tree processing based, natural language processing based, wrapper induction based, modeling based and ontology based.

According to Lam, Gong and Muyeba (2008) and Shaker Mahmoud et al (2010) the following are techniques for data extraction and their corresponding challenges.

Language development based

In order to assist the user to accomplish the extraction task, a new language was developed for a language development based system. The famous systems for this type include TSIMMIS and Web-OQL. One of the drawbacks of such a model is that not all users are familiar with the new query language, so the performance of the system may not be as expected.

HTML tree processing based

Since most of the web pages are in HTML format, another type of extraction system, HTML tree processing based system, was proposed. By parsing the tree structure of a web page, a system is able to locate useful pieces of information. XWRAP and RoadRunner are examples in this respect. In this solution, web pages need to be transformed into XHTML or XML format due to limitations of the HTML format.

Natural language processing based

For some pages which are mainly composed of grammatical text or paragraphs, Natural Language Processing (NLP) systems can be used. NLP is popularly used to extract free text information, and makes use of filtering, part-of-speech tagging and lexical semantic tagging technology to build up the extraction rules. SRV, WHISH and KnowItAll are examples of this technique. However, for some pages which are composed of the tabular or list format, NLP based tools may not be effective since the internal structure of the page can not be fully exploited. Also they tend to be slow and this can be a problem as the volume of document collections on semi-structured data such as web pages can be large and the extraction is often expected to be performed on the fly. Therefore, NLP techniques are however not well suited for structured and semi-structured data as these techniques require full grammatical sentences.

Wrapper induction based

The wrapper induction based systems can induce the contextual rules for delimiting the information based on a set of training samples. SoftMealy and STALKER are typical examples.

Modeling based

In modeling based systems, according to a set of modeling primitives, for example tables or lists, the data are conformed to the pre-given structure. Then the system tries to locate the information against given structures. NoDoSe is an example of this type of systems.

Semantic-based

With the advent of the Internet, more information is available electronically, and the information on the Internet is generated in textual form which differs from the web page to another in semantics. Semantics generally deals with the relationships between signs and concepts (mental signs). Different kinds of semantics are Lexical Semantics, Statistical Semantics, Structural Semantics, and Prototype Semantics. The semantic-based information extraction from web pages depends on presentation regularities and domain knowledge where there is a need to divide a web page into information blocks or several segments before organizing the content into hierarchical groups. However during this process (partition a web page) some of the attribute labels of values may be missing which pose the weakness to this technique.

Structure-based

The structure based approaches employ assumptions about the general structure of tables on the web pages by analysing any given web page for the existence of tabular data, recognizes relations as implied by their spatial arrangement, extracts a number of n-tuples together with hierarchical information about relations between their entries and saves them in structured data format. The task of extracting web tables is formulated as the task of (i) finding all frames for a given web page, (ii) discerning those which adhere to the definition of tables where a 2-D grid is semantically significant from lists and other frames intended for non-relational layout purposes, (iii) transferring the content into a topological grid description in which logical cells are flush with neighboring cells and their spatial relations are explicit. The main obstacle of the structured-based approach is that it depends on the specific structure such as table tag <TABLE> tags hence fail to adapt new changes.

Ontology based

Ontology is a branch of philosophy and structures of objects, properties, events, processes and relations in every area of reality. Ontology techniques can be used to decompose a domain into objects, and describe these objects. This type of system does not rely on the structures of web pages or the grammars of texts but instead an object is constructed for a specific type of data. WebDax is a typical example in this respect.

XML-based

There are several challenges in extracting information from a semi-structured web page such as the lack of a schema, ill formatting, high update frequency, and semantic heterogeneity of the information. In order to overcome these challenges, some researchers have proposed approaches for transforming the page into a format called Extensible Mark-up Language (XML). In this situation the extraction task is only individual page based where all the fields for the same record are supposed to be contained in the same page. However, in many other situations, the fields may be located in different relevant pages, such as several linked web pages hence make poor performance.

Clustering-based

Cluster analysis has been playing an important role in solving many problems in medicine, psychology, biology, sociology, pattern recognition, and image processing. Clustering algorithms attempt to assess the interaction among patterns by organizing patterns into clusters such that patterns within a cluster are more similar to each other than are patterns belonging to different clusters.

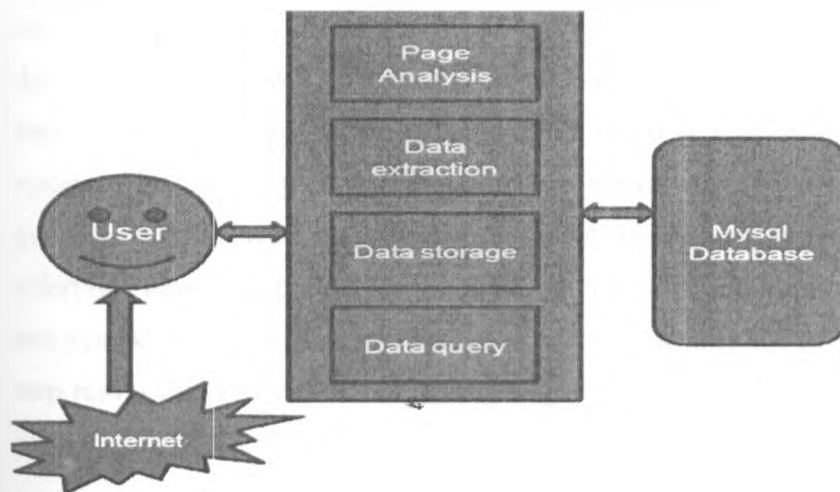
Other techniques

Besides classifying by the main techniques used, the wrapper can also be grouped into semi-automatic wrapper or fully-automatic wrapper. For the semi-automatic wrapper, human involvements are necessary. Most of the systems belong to this type, such as TSIMMIS and XWRAP. For the fully-automatic wrapper, no human intervention is needed; examples include Omini and STAVIES, which make use of tree structures, or the visual structures of pages to perform the extraction task.

2.5 Conceptual Design

From the explanation of literature review above, there is a need to automate the task of locating, extracting and storing data to data warehouse. Below is a diagram for the model.

Figure 4: Proposed extraction system



The system should be able to offer page analysis, data extraction, data storage and data query with the use of User interface for easy interaction.

CHAPTER THREE: METHODODOLOGY

3.1 Introduction

This chapter outlines the structure of the project work. It details the work done at each stage as milestones. The project follows an agile approach and is divided into five different phases which are performed iteratively and with a degree of overlap between them and with testing completely integrated into the implementation phase. The research utilizes the System Development Method (SDM) that is the research design methodology which combines both research and practice. The combination of research and practice is done in such a way that in one side the research raises a broad understanding of the problem and on the other side, the development of a program serves as a proof of testing the theories.

3.2 Research Methodology

The following are the phases which this project undertakes in an attempt to solve the research problems. It is based on SDM theories which are specific to the field of computer science, Degbelo and Matongo (2009).

3.2.1 Literature Survey

The first phase is concerned with study in the areas relevant to the thesis topic to determine the current definition of best practice. This includes the study of various research papers, journals articles, books and manuals that discusses the analysis of web data extraction and what the current trends on this field are. Also it looks at existing frameworks and tools for data extraction that could be used in the construction of the system. This step helps to build the concept where we have to construct our research questions by reviewing the literature to get related knowledge about the research and effort that have been done so far by scholars, in this field. It implied that we search, find and synthesize the existing knowledge in order to identify the scope of our work. This step results into an overview of the existing literature that is presented in the "Literature review and theory chapter"

3.2.2 Requirements Specification

This is the second phase which deals with formalizing the problem statement into a set of requirements which guide the design, implementation, testing and evaluation of the system. This includes specification of both function and non functional requirements.

3.2.3 Design

Design is the third projects phase which is suggesting a suitable design based on the requirements and the findings of the literature survey. The design included both logical design and physical design. In logical design, the proposed algorithm for data extraction is designed and tested whereas in physical design prototype system is designed. This includes Page analyser design and development, Data extractor design and development, data storage design and development and client user interface design and development. The program is developed using java as programming language.

3.2.4 Implementation

This is the fourth project phase which is concerned with implementing the suggested design as a component within this methodology. This includes actual implementation of page analyser, data extractor, data storage and user interface using java programming and java script pages.

3.2.5 Testing

Testing is the fifth phase. Testing is the act of designing, debugging, and executing tests. During the development of the system, there are testing strategies that are used to ensure that the system is error free. The main testing strategies which are used are Unit Testing, Integration Testing and System Testing.

3.2.6 Evaluation

It is a last project phase whereby the implementation is tested and the results are evaluated with respect to the theory and the requirement specification. Here the prototype is used for evaluation, where efficiency and effectiveness of proposed technique are evaluated. Recall and precision is used to evaluate the correctness and completeness of the algorithm.

3.3 Research Design and its Justification

This research is experimental where development of prototype is be used. Validation process involves collection of information from website to test its correctness and completeness of its proposed algorithm. The choice of websites is in such way that it cuts across all important sectors such as business, academic and industries where information extraction is their major activity so as to remain competitive. Experimental approach fits

this research since it involves implementation and evaluation of the techniques by quantifying its performance in terms of its efficiency and effectiveness. Hence quantitative approach is used in this study in order to obtain details of requirements for web data extraction.

3.4 Sources of Data/Information and Relevance of Data to the Problem

The data for literature and theory are from both primary and secondary source. From secondary sources, we have managed to make use of previous published information to formulate problem statement and to guide implementation process of proposed technique which is the extension of previous authors' work. From primary source we have managed to make use of new data for testing the algorithm, this new data include those websites which are identified to have problems which our algorithm is trying to solve. Both primary and secondary sources are relevant to the problem which the study addresses.

3.5 Tools, Procedures and Methods for Data Collection

This study uses a quantitative based approach to investigate the proposed algorithm for its efficiency and effectiveness. The main data collection techniques used are as follows:

- (i) Review of the current literature on techniques for data extraction process
- (ii) Research on existing models of extraction system.

Two sources of data have been used as mentioned below:

- (i) Happy Harvester 2 Software which is a commercially available product for web data extraction;
- (ii) Various web sites which can be grouped into business, educational and government websites.

The use of Happy Harvester 2 is justified since it is closely related to the approach/technique used in this study. The use of the web sites from business, educational and government is justified since these web sites experience the problems related to data extraction.

3.6 Data Analysis Methods and their Justification

The data analysis method which is used in this research is descriptive, the reason being that this research is experimental based where the use of precision and recall for identify correctness and completeness of proposed technique is applied. The arithmetic mean is used to look for the average of each precision and recall value.

3.7 Limitation of Methodology and how they are overcome

This research uses SDLC as a methodology; however it has been stated that SDLC is no longer applied to models like Agile computing, but it is still a term widely in use in Technology circles. The SDLC practice has advantages in traditional models of software development, which lends itself more to a structured environment. The disadvantages of using the SDLC methodology is when there is need for iterative development or (i.e. web development or e-commerce) where stakeholders need to review on a regular basis the software being designed, *Degbelo and Matongo (2009)*

Table 7: A comparison of the strengths and weaknesses of SDLC

Strengths	Weaknesses
Control.	Increased development time.
Monitor Large projects.	Increased development cost.
Detailed steps.	Systems must be defined up front.
Evaluate costs and completion targets.	Rigidity.
Documentation.	Hard to estimate costs, project overruns.
Well defined user input.	User input is sometimes limited.
Ease of maintenance.	
Development and design standards.	
Tolerates changes in MIS staffing.	

How to overcome the above Limitation

- (i) An alternative to the SDLC is Rapid application development, which combines prototyping, Joint Application Development and implementation of CASE tools. The advantages of RAD are speed, reduced development cost, and active user involvement in the development process
- (ii) Another alternative is Component-based software development technology. It removes the problems by supporting fast development, easy maintenance, good quality, easy creation/upgrade of application, low cost, etc. It also creates new software business area such as business component developer, sales and distribution vendors

3.8 Summary

This chapter discusses the methodology used to undertaking this project. It uses software development lifecycle. It consists of six steps which can be grouped into three major steps, namely; concept building step which consists of theoretical studies and analysis of the problem, system building step which consists of requirement analysis and specification, system design, system testing and lastly system evaluation step.

CHAPTER FOUR: ANALYSIS AND DESIGN

4.1 Introduction

This chapter analyses, designs and implements the proposed algorithm.

4.2 Algorithm Analysis and Design

4.2.1 Introduction

This section describes the main component of algorithm. It demonstrates the way it operates. It has two main sections which are web page analysis and investigation on web page analysis. The two components assist the development of the algorithm.

4.2.2 Web Page Analysis

4.2.2.1 Structure of Web page

The web page can be viewed as script which consists of the HTML tags together with the information visible on the web pages. HTML tags are those texts that are enclosed by angle brackets ("`<`" then "`>`"). The HTML tags may contain other information which is used to provide more information about the HTML tag. This information is called attributes of the tag that is name value pair. Here is an example of the HTML tag with Attribute, ``. Font is that HTML tag which describes the behaviour of the texts enclosed by that tag. Size is the attribute which shows size of the text enclosed by the tag. In this case, the value of the font size is 10. Further more the HTML tags should have start and end tags. The tags should conform to the XML quality standard, such as proper nesting of tags, quoted attribute tags and having a closing tag. The current browsers tend to ignore these qualities and the new extension of HTML which is XHTML enforces XML syntax.

4.2.2.2 Algorithm Approach

The approach of this algorithm is through analysing the structure of the web pages by analysing the HTML source codes. The aim of the page analysis of the web pages is to automate the process of identifying the data clusters which were generated as a result of user search query. The approach which I am investigating in this project is the use of repetitive pattern which marks the existence of the data cluster. This approach was introduced by Robinson (2004). The representation of the web page into tag-String and text-String provides a mechanism of separating the information visible through the browser and HTML tags. Some information on the HTML tags is also values and need to be extracted. This information includes links to images, video clips, other web documents

and other web resources. The analysis of the web page is done on the tag-String for locating the repetitive patterns for data clusters, Robinson (2004).

4.2.2.3 Components of Algorithm for Web page Analysis

According to Robinson (2004) the following are the components of algorithm for the page analyser.

(i) Tag String and Text String

It is the first step for page analyser where the HTML source codes are presented in tag string and in text string. It is important to represent the HTML web page as a sequence of numbered tag-String with the corresponding text-String. This is because the web page analysis will be based on the list of tag-String. It can be seen that in all cases, the number of the tag-String elements will be one more than that of text-String. This is because there is no text-String after the last tag-String. The attributes of the HTML tag are used to provide more information about the tag. So the tag attributes are not be used during the analysis for the repetitive pattern. Using tag string, the existence of data cluster can be observed by having repetitive pattern. But the weakness of this method is that when the order of HTML tags is different for the same presentation become difficult to locate repetitive pattern.

(ii) Tag Set

The tag-String is not used to provide similar sequence of number patterns for discovering the data cluster. Tag-Set method solves the problem when the order of HTML tags is different for the same presentation. For example the tag-String `<i>` should be considered as the same as `<i>` but the tag name for instance `td` is different from that of `/td` or `td/`. In this case, the tag-String is represented as a tag-Set. Tag-Set is sequence of non negative integers, which show how many times a tag appears in the given tag-String. The order of the sequence is arbitrary, which depends on the order of the list of all distinct HTML tags used in the web page. From a numbered list of tag-String the corresponding list of tag-Set is produced, for each tag-String. Then, a list of the distinct tag-Set is produced. The existence of the data cluster can be suspected when the size of the list of distinct tag-Set is less than that of all the tag-Sets. If this is the case, then a particular tag-String has been used more than once. This means that, there is a repetitive structure based on the tag-String.

(iii) Item Set

It is important to know which tag-Sets are similar to each other for each distinct tag-Set. So the item-Set is the group of tag-Sets having the same tag-Set. The item set is represented as series of numbers denoting the position of a tag-Set in the list of the tag-Sets.

(iv) Data clusters

The data cluster is discovered when there is a complete cycle when the trail of item number is followed. The technique used is known as the row succession graph. The main task of the analyser is to walk through the nodes and pick out those which are data clusters. The analyser starts at the entry point of the cycle and ends at the exit point of the cycle. When the analyser enters the cycle with a data cluster (the transition count is greater than one), it creates the record of the data cluster before it starts again for the next cycle. The number of records for a data cluster is equal to the number of cycles the analyser passes. Normally the tag-Set which represents first field of the first record is different from those of the data cluster. In this case, the analyser starts creating the first records from one node before the entry node.

(v) Page Descriptor

The final task of the page analyser is to create the page descriptor. The page descriptor is the information which is used to locate the data cluster in the result web page. In this report the page descriptor has three main kinds of information. There is a tag-Set number which shows the start of the data cluster. Then there is the size of the record which can be identified (using row succession graph) by counting the number of nodes forming the cycle of the data cluster. Also the size of the record is the common difference of the terms which form a row of the succession graph. There is also the tag-Set number which shows the end of the data cluster. Once the information for locating the data clusters in the web page has been produced, the data extractor can easily extract data. This is because each of the text-String number corresponds to the tag-Set number. So, the data extractor walks over the list of the text-String. When the text-String number that corresponds to the start of the data cluster is encountered, the data extractor starts creating a record. The size of the record is given as part of the information for page descriptor.

4.2.2.4 General formula for data extractor

If the size of the record is Z where Z is a positive integer and the start of the data cluster is tag-Set N where N is a whole number such that N is greater than or equal to Zero and N is less than the number of entries of the List of text-String, then the first record will be extracted as shown below. Given that the list of text-String items is TS .

Table 8: Data Extractor

Record	Field 1	Field 2	...	Field Z
1	$TS[N]$	$TS [N+1]$...	$TS [N+Z-1]$
2	$TS [N+Z]$	$TS [N+Z+1]$...	$TS [N+2Z-1]$
3	$TS [N+2Z]$	$TS [N+2Z+1]$...	$TS [N+3Z+Z-1]$
...
K	$TS [N+(K-1)*Z]$	$TS [N+(K-1)*Z+1]$...	$TS [N+K*Z-1]$

The data extractor will stop extracting at the end of the data cluster, which is marked by the tag-Set number say M , when M is the whole number such that M is greater than N and M is less than the number of entries of the List of text-String. The data extractor will stop extracting when the value of M is equal to $TS [N+K*Z-1]$ as shown below.

$$M = TS [N+K*Z-1]$$

The data are presented into a structured format, so that it can be stored into relational database tables, XML format, or on any data files such as Microsoft excel.

However, there are some web pages which produce irregular structures and hence it is difficult for the page analyser to identify the data clusters on the web page.

4.2.3 Investigations on Web Page Analysis

4.2.3.1 Introduction

This section presents the investigation on the natures of result web pages where in some cases, the structure of the data cluster is not easily identified. The suggested solution to those problems is discussed. The page analyser tends to use the repetitive patterns in order to identify the data clusters in the result web page. Some of the tag-Strings which are not part of the data cluster may be the same as those on the data cluster. The

following are problems which occur as result of web query and their solutions, Robinson (2004).

4.2.3.2 Restructuring of tag-Set

In some cases it is important to restructure the tpGrid, so that those tag-Set which are displaced from the data cluster are retained. This can occur when a tag-Set of the data cluster is the same as the one which is not part of the data cluster. Restructuring process is done by investigating the columns for each rows of the data cluster. By reading the first entries for each row, the analyser forms the items based on the consecutive values. The analyser will check if these items are in consecutive orders. Otherwise, the refined items are produced by identifying the location of the missing items. Technique such as splitting the row into rows is used so as all items which form the data cluster are grouped together.

4.2.3.3 HTML tags and tpGrid Structure

Some of the HTML tags are frequently used by most of the websites to provide the formatting which can be very useful for recognising the data clusters. In most of the websites, the HTML table tags are frequently used for presenting data in record format. Other web sites use tags like, link tags `<a>`, break `
`, paragraph `<p>`, lists `` and bold `` or `` tags for marking the fields for each record. In most cases these tags are associated with other HTML tags which format the data presentation such as font settings. One of the reasons that the tpGrid structure of irregular is that, some of the HTML tags are used for the purpose of putting more emphasis. Some of them are just used for formatting the presentation. The solution to this problem is to allow the analyser to ignore the HTML bold tag. This approach is very useful for some web sites which use bold HTML tag for emphasis. Other tags which need to be ignored include subscript or superscript tag, image tag just to mention few. But it can be seen that, these HTML tags can not be ignored by analyser in all cases. Some websites use bold for marking fields of data clusters. In these web sites, when the HTML bold tag is ignored, the record structure will be affected as well.

4.2.3.4 Records with Optional Field

It is not the case that always the data clusters have the same structure of each data record that is the same number of fields. Since there are some tags that occur once or rarely in some of the data records, this results into irregular structure which make hard for the analyser to identify the data cluster. To solve this problem, the algorithm uses technique for pattern recognition for clusters identification where the tag-Set number is written with the corresponding, the most first similar tag-Set number. The entries which occur only once are marked by entry -1. This approach has the benefit that it works with all the other web sites which works with row succession approach. This approach is more general, that is it solves the problems of the optional fields by providing the more general record templates. Analysing this trial of numbers, patterns can be identified for data clusters. Before the analysis of the patterns the clusters need to be sorted out. This means that, the items which occur only once should be marked as -1. The various groups of clusters – surrounded by -1's – can separately be analysed. The simplest pattern recognition approach is that, the first entry for each record occurs at a certain internal within the items in the cluster. Since the number of fields in each record is not the same because some of the field entries are optional fields then the analyser creates the template record for the structure of the data cluster. The template record for the data cluster is the one which has the highest number of items. The other records are aligned based on the structure of the template record. The items which are missing from the template record are marked by entry -1, this signifies that there is a missing item in that entry. It should be noted that, the last entry of the items of the last record does not mean that it is an optional value. This occurs due to the fact that with reference to the tpGrid, there is a hole at the end of the last row.

4.2.3.5 Nested Data Clusters

There is a situation whereby the analyser is able to identify data clusters and hence extract data from those pages. But in the data extracted it is found that, some of the data fields are located on the wrong columns. This happens when there is a structure of nested records. The problem can be solved by using pattern recognition approach although some items are not found on the chosen template record but existed in some other records, Robinson (2004).

4.2.4 Components of Algorithm

The focus of this research is dynamic web page whose contents change for each new request of the web page. Such a dynamic web pages are those which are used to get the information from the underlying databases by providing the search form where the user can query the information from those sites.

For the purpose of locating data to be extracted, the algorithm analyses the structure of the web pages by analysing the HTML source codes. This process helps to automate the process of identifying the data clusters which are generated as a result of user search query. The approach which is used is the use of repetitive pattern which marks the existence of the data cluster. The analysis of the web page is done on the tag-String for locating the repetitive patterns of data clusters. This approach was introduced by Robinson (2004). It consists of the following terms:

- a) Tag string and Text string (produce distinct tags)
- b) Tag set (produce distinct Tag set)
- c) Item set
- d) Data clusters
- e) Page Descriptor

The components are explained here below:

4.2.4.1 Algorithm for locating Data Cluster

This algorithm deals with page analyzer which is used to locate data cluster for data extraction.

1. Get HTML source code
2. Parse HTML source code to a sequenced list of HTML tags part and texts part as they appear on the HTML source code (both).
3. Using the information from 4.2.4.1 (2) above to create to lists of tag-String and text-String.
4. Use a list of tag-String to generate a list of all distinct tags used in the HTML source code. Hence we have a list of text Strings, a list of tag Strings and a set of distinct HTML tags.
5. Get a list of tag String
6. Generate a list of tag-Sets for each tag-String
7. Generate the list of distinct tag Sets from 4.2.4.1 (6) above

8. Get the list of tag-Set String and distinct Tag-Set String
 9. For each tag-Set string of the distinct tag-Sets generate the String of tag-Set numbers of similar tag-set from the list of tag-Set string
 10. Refine the Collection of item sets by comparing the tag-Set with those within the same cluster
 11. Get the trail of item sets.
 12. For each cluster identify the records based on the similarity of patterns on the trail of item sets.
 13. Identify the template record.
 14. Use the template record to align all the records and represent the optional items (items which are not on the record but on the template record) by -1.
 15. Produce the page descriptor using the item set numbers.
 16. Data extractor can extract the located data according to page descriptor above 4.2.4.1
- (15)

4.2.4.2 Algorithm for Data Extraction

- 1 Gets the page descriptor information and the collection of text-String from the component page analysis.
- 2 Iterated over the collection of text-String until the start of the data cluster is reached.
- 3 Creating the first record of the data cluster using the size of the record from the page descriptor.
- 4 Repeat the process 4.2.4.2 (3) until the end of the data cluster is reached.

4.2.4.3 Algorithm for storing and querying the Database

This algorithm is responsible for all database operations. The database operations range from database connection to execution of database table queries.

1. load database driver
2. get database connection
3. Database table is created for new data.
4. The data is inserted into the database table one record at a time.
5. Get the database connection object
6. Execute database query
7. For each record add the entry to the collection object.
8. Return the collection of result data

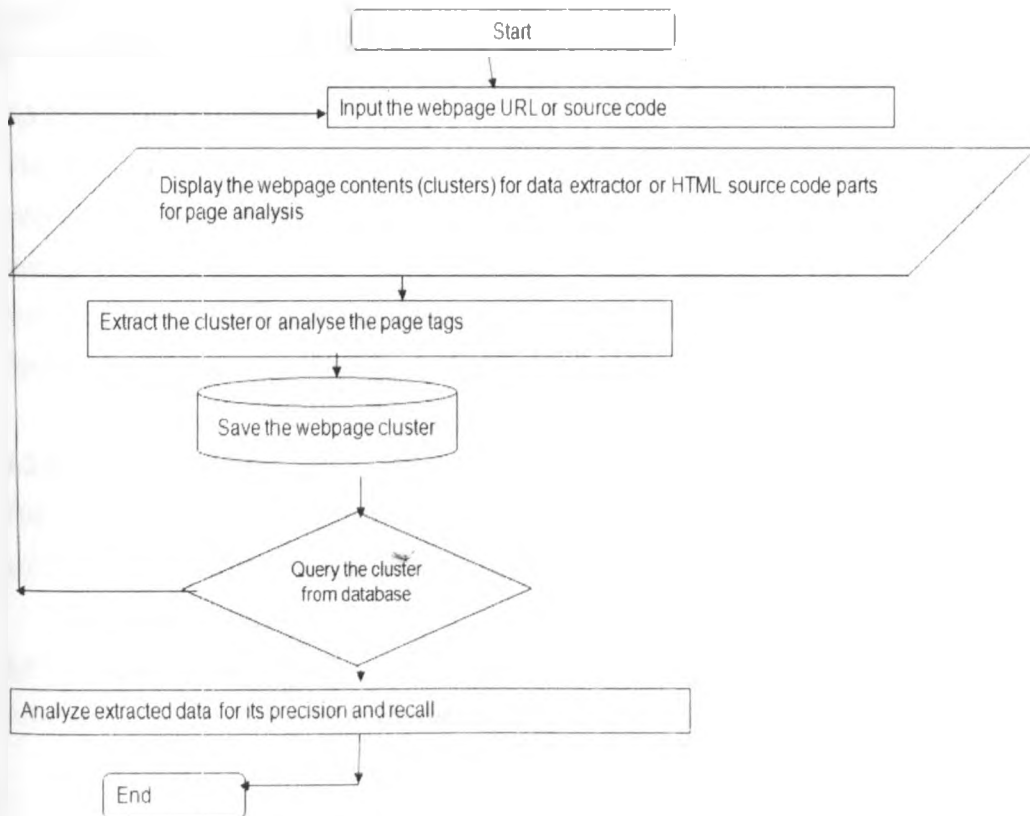
4.2.5 Algorithm for the complete System

The algorithm for extracting data from www based on HTML tags. It combines page analysis, data extraction, storage and query algorithm.

1. Start the application program
2. Input the web page by entering web page URL
3. Get the HTML source code from the user or from the URL.
4. Locate data clusters
5. Refine data clusters identified
6. Create page descriptor
7. Extract data from html source code specified by page descriptor
8. Refine the extracted data
9. Store the data into database
10. Query the data from the database
11. stop

4.2.6 Data flow

Figure 5: Data flow for the system



4.2.7 Summary

This section explains what algorithm is trying to achieve and type of problems which occur during data extraction and how it has been solved by proposed algorithm.

4.3 Functional and Non-Functional Requirements

4.3.1 Introduction

This section is about the services the system provides to the user. The main focus of this section is functional requirements of the system and non functional requirements which show the external behaviours of the system as seen by user. The system functionalities have been designed to show the real sense of data extraction and hence make the information available to end user and application software at any time. The section proceeds with common terms or process which occurs during data extraction.

4.3.2 Extraction Process Perspective

In data extraction process, user needs parse information data from respective site and store them to a local database. This process should locate, extract and store data automatically with less human intervention when users start running software and by specifying where to extract data through specifying the keywords.

4.3.2.1 System Interfaces

The data extraction process system to be developed as a stand-alone tool that within the internet. It consists of three major components: Web Access, Page analysis, data extract, data storage and data query. The Web is accessed through internet accesses where the data is extracted. The data extracted is passed to database for storage purpose. From the database data can be queried or used by any application.

4.3.2.2 User Interface

The data extraction process system must provide a user interface that is available through graphical user interface for page analysis, data extractor and data query interface.

4.3.2.3 Hardware Interface

All components must be able to execute on a personal computer.

4.3.2.4 Software Interfaces

The data extraction process must be Java Application Running Environment. The Web Access, page analysis, data extract and data storage must be integrating with each other. There should be individual interfaces for them.

4.3.2.5 Operation

The operation of the data extraction process must be easy and intuitive for any user. No specific knowledge is required to use the system. The operated process should check the respective web if it is accessible, particularly on whether the address is still valid and has enough memory room for database.

4.3.3 Extraction Process Function

The main function of the data extraction process system is to extract information or data from any website especially deep web and then to store it to a local database automatically. Another optional function is to group data into data clusters where the user decides which data to save in the database.

Data extraction process also provides two simple operations about database handle: first if there is any database existing on which the user can create a new local database; second if there is already a valid database data extraction process which can write meta-data into this database.

A timestamp record when the data is entered into database is provided by the system. It is also specific data or information with a unique identification. This unique identification is useful for the future work for searching purposes. The data extraction process should read web page URL address from user. When data extraction process is extracting all the data from the web page, user can not do any implementation. The process should handle some simple errors which may be caused by either web access or database handling. If an error happens, the data extraction process should have an output warning the user where this error throw out and may record this error. The software extraction process must get the total number of the data cluster and could estimate when the process should stop.

4.3.4 User Characteristics

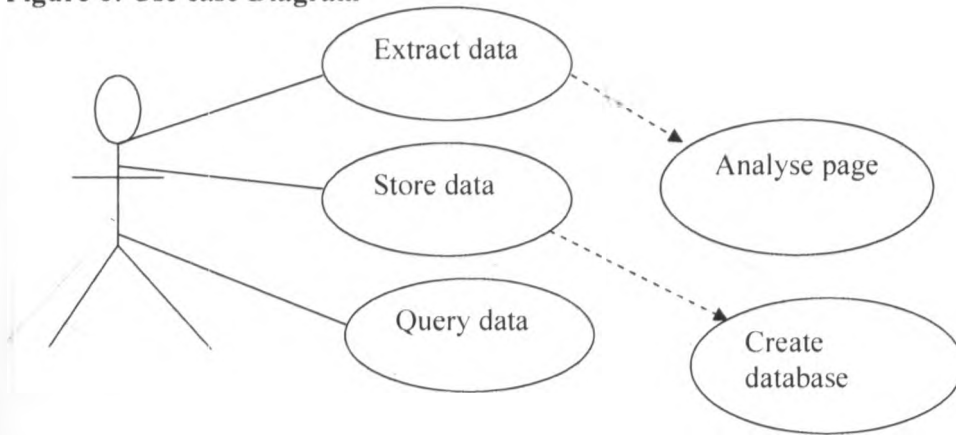
Users of the developed system can be any body from the business, educational and government sector including both experts and naïve users.

4.3.5 Use-case Model

To better understand the data extraction process requirement and how to interact between the system and users, here infer use-case model describes where the system behaviour originate from the user. Following section displays the use-case diagram shows use-case specification.

4.3.5.1 Use-case Diagram

Figure 6: Use case Diagram



This section explains the software extraction process use-case diagram. This use-case diagram has three actors: User, any website where data should be extracted and Database. There are five Use-Cases as displayed in table below.

Table 9: System use-case

Use case	Name
UC 1	Page analysis
UC 2	Data extraction
UC 3	Data storage
UC 4	Create new database
UC 5	Client User Interface

4.3.5.2 Use-case Specification

This subsection describes the Use-Case in detail; it defines a goal-oriented set of interactions between external actors and current system. By describing the features of the system as Use-Cases, makes it easier to transform them into requirements. See table below.

Table 10: Use-case Specification

A: Page analysis

Code	UC 1
Use Case Name	Page analysis requirement
Goal	The goal of this function is to provide information about the location of the web data to be extracted.
Pre-condition	<ol style="list-style-type: none"> 1. Start the process. 2. Get url or html source code from the user about specific website. 3. Success link to respective homepage for chosen url.
Post-condition	The output of this process will be HTML page modelled into different lists such as list of text string, tag string, distinct tag set, tpGrid, Semi refined tgGrid, trail for clusters and page descriptors. Hence this will produce the location of data to be extracted.
Actors	User
Triggering event	The user enters url or HTML source code.
Flow of events	<ol style="list-style-type: none"> 1. Start the process. 2. Get url or html source code from the user about specific website. 3. Incase of url a success link to respectively homepage for chosen url will be done. 4. Gets the HTML source code containing the searched data. 5. Click on the link (Page Analysis) and Paste the HTML source code from the clipboard. 6. Specify any settings for HTML tags to be ignored by the analyser 7. Click the button (Process page), the new page will be

	<p>shown. This page contains the combo box. Select the model for viewing and then click on the button (Submit)</p> <p>From above, the expectation from page analysis is</p> <ul style="list-style-type: none"> • The system should be able to get HTML source code from user. • The system should be able to identify the data clusters in the HTML source code. • The user should be able to see and select the data clusters using check box controls and command buttons. • The system should be able to create the page descriptor from the page specified.
Extensions	None
Alternatives	<ol style="list-style-type: none"> 1. When URL or HTML source code does not exist or it has a wrong format it will display a warning and stop the process. 2. When connection to a website fails it will display a warning and stop process. 3. When process cannot read information from particular page, it will display a warning and continue to next one.

B: Data extraction

Code	UC 2
Use Case Name	Data extraction
Goal	This component is intended to be used to extract the data from web pages before they are stored on the data store. This component will make use of the information from the page analyser to extract the information from the web page.

Pre-condition	UC1 successful implemented. User should be able to get source code.
Post-condition	The extracted data are ready to be stored to database or data warehouse.
Actors	User
Triggering event	User click extract data link. Or triggered by UC1.
Flow of events	<ol style="list-style-type: none"> 1. The system should be able to extract the data from the html source code specified by user. The system should be able to use the existing page descriptor for data extraction 2. The user should be able to see the data cluster extracted. 3. Start the data extraction process by clicking extract data link. 4. Getting Html source code Read url or get source code configuration from respective website after search the information of interest. For a given url a success link to respectively homepage will occur. This will be an extension of the first use case. From result web page obtain html source code by using view button on the browser (view->source->), then copy to the clipboard 5. Processing Html source. Click on the Link (Extract Data). Paste the HTML source code from the clipboard, and specify any settings for tags to be ignored by page analyser. Then click on the button (Process Page). Data cluster will be displayed. 6. Select data Cluster. Click on the button (Select cluster), one data cluster will be processed. Click on the check box to select the columns of the data of interest (some columns will be empty). Once the data cluster is refined, enter the information on the textbox. Then click on the button (Save Data) for the data to be stored into the database. The message will be shown to indicate that the information have been stored

	into the database.
Extensions	Extend from UC1
Alternatives	<ol style="list-style-type: none"> 1. When URL does not exist nor has a wrong format it will display a warning and stop process. 2. When the html source neither is empty or does nor exists, invalid display a warning and stop process. 3. When connecting to web failed it will display a warning and stop the process. 4. When process can not parse information from particular web page, it will display a warning stop process.

C: Data storage

Code	UC 3
Use Case Name	Data Storage requirement
Goal	This component uses the data produced by data extractor and store them to the database. The data store component will also receive user query to the database and get the query results to the user.
Pre-condition	<ol style="list-style-type: none"> 1. Local database existed and has correct format according to the structure proposed. 2. Read database configuration from configuration file. 3. Success link to local database. 4. Receive extracted data from UC1 and UC2. 5. Enough memory room to store the data.
Post-condition	The extracted information or data was inserted to database.
Actors	UC1 and UC2.
Triggering event	Add method in UC1 and UC2
Flow of events	<ol style="list-style-type: none"> 1. Raise add method from UC 1 or UC 2 2. Read database configuration from configuration file 3. 1&2 above will make success link to database 4. Receive data extraction from UC 1 or UC 2. 5. Write extracted data or information into database. The system should be able to store the data cluster specified by

	<p>user.</p> <p>⇒ The system should be able to create tables based on the structure of the page descriptor.</p> <p>⇒ The system should be able to insert the new data to the existing database table.</p> <p>6. The user should be able to query the database.</p> <p>7. The system should be able to provide the user with the result of the query.</p> <p>8. The system should be able to inform the user if there is no results of the query.</p> <p>9. Close the link to database</p>
Extensions	Extend from UC1 and UC2
Alternatives	<ol style="list-style-type: none"> 1. If the database configuration file does not exist or has the wrong format it will display a warning and stop the process. 2. If database does not exist or link to database failed, it will throw out a warning and stop process. 3. If information data has error syntax or format, it will throw out a warning. 4. If the system did not have enough hardware memory room to store data it will throw out a warning.

D: Create new database

Code	UC 4
Use Case Name	Create new database
Goal	Create a new database to store extracted information or data
Pre-condition	<ol style="list-style-type: none"> 1. No similar name database exists. 2. Read database configuration from configuration file.

	3. There is enough hard disk memory.
Post-condition	New database was created
Actors	User
Triggering event	The button Save has been applied by user
Flow of events	<ol style="list-style-type: none"> 1. User click the button enters command from command control panel. 2. Read database configuration from configuration file. 3. Create new database. 4. Process end
Extensions	Extend from UC1 and UC2
Alternatives	<ol style="list-style-type: none"> 1. If database configuration file does not exist or format, an error it will display a warning and stop the process. 2. If there exists already a database, it will display a warning and stop the process. 3. If there was not enough hardware memory, it will display a warning and stop the process.

E: Client User Interface

Code	UC 5
Use Case Name	Client User Interface
Goal	This component provides the user with the interface which will enable user to interact with the system. The user provides search terms and the results of the query will also be displayed for the user.
Pre-condition	There must be UC 1, UC 2, UC 3 and UC 4 already implemented
Post-condition	Interface for UC 1, UC 2, UC 3 and UC 4 integrated into one interface.
Actors	User
Flow of events	<ol style="list-style-type: none"> 1. The user should be able to load the html page to the system using the text area on the browser window and

	<p>click on the button.</p> <p>⇒ The System should be able to create the page descriptor for the result web page.</p> <ol style="list-style-type: none"> 2. The user should be able to select the cluster which contains data needed by user (see requirement 01-iii). 3. The system should be able to store on the file the data cluster specified by the user. Then, the user clicks the button to store the data to the database. 4. The user should be able to see in the tabled structure the result of the data extracted. 5. The system should be able to extract data specified by other queries using the stored page descriptor for that site. 6. The system should be able to store the data cluster specified by user to the database.
Extensions	None
Alternatives	<ol style="list-style-type: none"> 1. If database connection fails, it will display a warning and stop the process. 2. If link to another interface fail, it will display a warning and stop the process. 3. If there is no enough hardware memory it will display a warning and stop the process for the loading interface.

4.3.6 Functional Requirements

This section describes the functional requirements of the data extraction system. Each requirement results from the system features and the Use-Cases, which have been described above. Each requirement has its type Essential or Desirable (Essential: means system shall have this function; Desirable: means system should has this function).

Table 11: Functional Requirements

Functionality	Type
R1 Get HTML source code from URL or User.	Essential
R2 Locate data clusters	Essential
R3 Give Page descriptor.	Essential
R4 Get data.	Essential
R5 Refine data	Essential
R5 Store data	Essential
R6 Query data.	Essential
R7 Web access error handle.	Essential
R8 Insert selected data cluster with Java time stamp.	Essential
R9 Database error handle.	Essential
R10 Use interface to interact with system which integrate page analysis, data extract, data storage and data query.	Essential

4.3.7 Non-functional requirements

These describe the system from external point of view. In this system the following are non-function requirement.

4.3.7.1 Performance Requirements

The system caches the web data to the data store. This increases the efficient of the system in terms of processing time. Before the data is extracted, the page descriptor is produced. Then all the subsequent searches are used to produce page descriptor and hence response time will be low. The system takes up to 5 seconds at worse case during the production of the page descriptor.

4.3.7.2 Quality Attributes

4.3.7.2.1 Security

This system can be used by any user and any application, to produce the data in a structured format, such as relation database or XML format. The system requires no authentication to the users. This system can be used as a separate component which is used by other systems to extract data from the web pages.

The user should comply with the applicable data protection legislation in the respective jurisdiction. Since the user uses data from other business organisations, the data should be processed further by complying with the data protection legislation.

4.3.7.2.2 Availability

The system developed is available in the sense that it runs at any point of time. The system is able to detect the page format changes and hence informs the user. The system can produce the new page descriptor in response to the web page structural change. The system is intended to run at a certain time interval without crashing.

4.3.7.2.3 Reliability

The system can produce highly reliable data for the user. This is because the system uses the structures of HTML tags used to represent on the web pages which is the actual data needed by the user. The system will extract only the data item identified in the web page. The system is intended to be robust during the error conditions and the user can use the system at any time on demand.

4.3.7.2.4 Maintainability

The product to be developed is designed to be maintainable, in the sense that, it is component based. Each component is intended to be more independent in terms of modification. The functionality of page analyser can be modified without affecting the data extractor or data store.

4.4 Architecture and Design

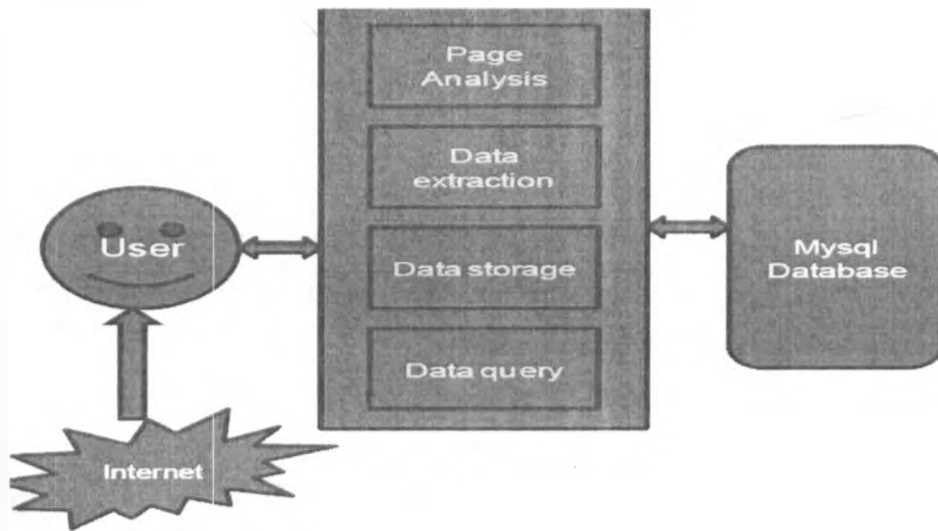
4.4.1 Introduction

This section provides a comprehensive architectural overview. It shows logical view of the data extraction process system.

4.4.2 System Architecture

The system is designed to follow three-tier system architecture. In the front end the system has JSP pages for presentation to the user. In the middle tier, the system uses system modules to model business logic. The business logic includes all the processing and the services provided by the system. In the business logic, there are two main components. These are page analysis and data store. With page analysis, all the operations for identification of data clusters are carried out. The data store is mainly responsible for database operations. The back end of the system is the database base which has been linked to the system by database connector.

Figure 7: System Architecture



4.4.3 Database Design

4.4.3.1 Overview

The data model of the system is designed for data extraction system especially data warehousing based application. So the information in the database table is stored as supplement and not updating the existing information. The system is designed to provide also a local cache for the web data. In doing so, the design of the data table is on the fly. Here it means that the database table schema is defined when the data is extracted. The information is stored incrementally and hence for each new extraction a new table is created or the information is inserted into the existing tables. The organisation of the data model follows the star schema of data extraction system design for data warehousing systems. In this case, there will be a central table which stores the subjects for the search information. Other tables will store the information that will be extracted from the web pages.

4.4.3.2 Fact Table Design

The central fact table designed contains information about the other tables. In other words it is a metadata table. In normal design of the data model of the data warehousing, the central fact table contains the reference (foreign key) information from the dimension tables, together with other aggregate information. In this data model design, the fact table contains the information about the web domain used for search, the subject for the search and the date when the information is extracted.

Below is the data table schema for the metadata table, which includes name of the data fields and data types.

Subject (ID, domain, keyword, search_date)

The name of the table is new which contains four fields. This new table is created during the system installation.

Table 12: Description of the fields of central table

Field name	Data type	Description
ID	Integer	This field is primary key and It is incremented automatically when the new record is inserted into the table
Domain	Text	This field stores information about the name of the web site domain URL. It contains the string of characters.
Keyword	Text	This field stores the keyword used during the searching of web information. It contains the String of characters. The maximum length of the String is 100.
Search date	Date	This field stored the system date when the record was stored into the table.

4.4.3.3 Data Table Design

The data tables stores the data that is extracted from the web pages. The design of the data tables depends on the structure of the data extracted. The names of the tables are identified by the search keyword of the query used to generate the search results.

In this design the table field names should have general names. A typical table with five columns is shown below. It should be noted that, the data extracted from the web page is not refined in response to the data types. So, the data field will be stored as text string regardless of what type they represent.

Table 13: Field Names

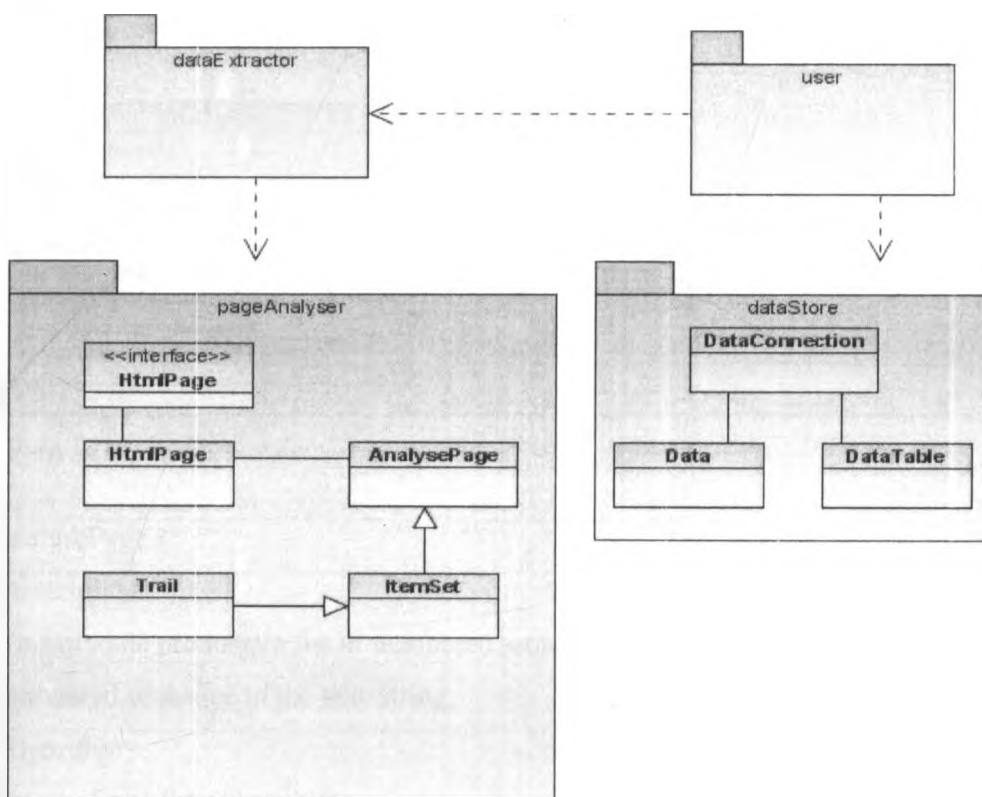
General schema for the data tables
Database table (field1, field2, field3, field4, field5);
The data type for each field is text string.

4.4.4 System Components

4.4.4.1 Component Architecture

This section shows the system components and system modules which model the system implementation. There are four main components with four modules in data analyser component. There are three modules in the data store component. The other components, data extractor and the user are mainly JSP pages. This component represents explicitly the structure and organization of the data or information extraction system

Figure 8: System Component



4.4.4.2 Component Description

Table 14: System Component Description

Page analyser

HtmlParser

Description

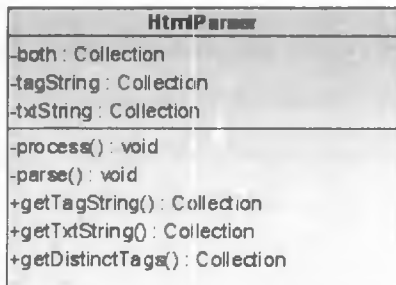
This module stores lists of tag-String, text-String and distinct HTML tags. The

module parses the HTML source code and sort out the HTML tags and text.

Algorithm

- i) Get HTML source code
- ii) Parse HTML source code to a sequenced list of HTML tags part and texts part as they appear on the HTML source code (both).
- iii) Using the list above (ii) to create to lists of tag-String and text-String.
- iv) Use a list of tag-String to generate a list of all distinct tags used in the HTML source code.
- v) Return a list of text Strings, a list of tag Strings and a set of distinct HTML tags.

UML Class diagram



analysePage

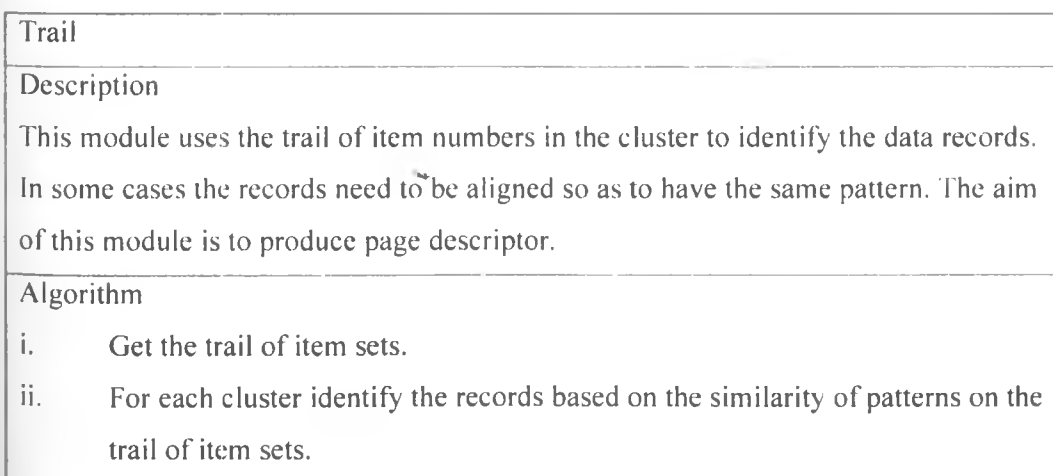
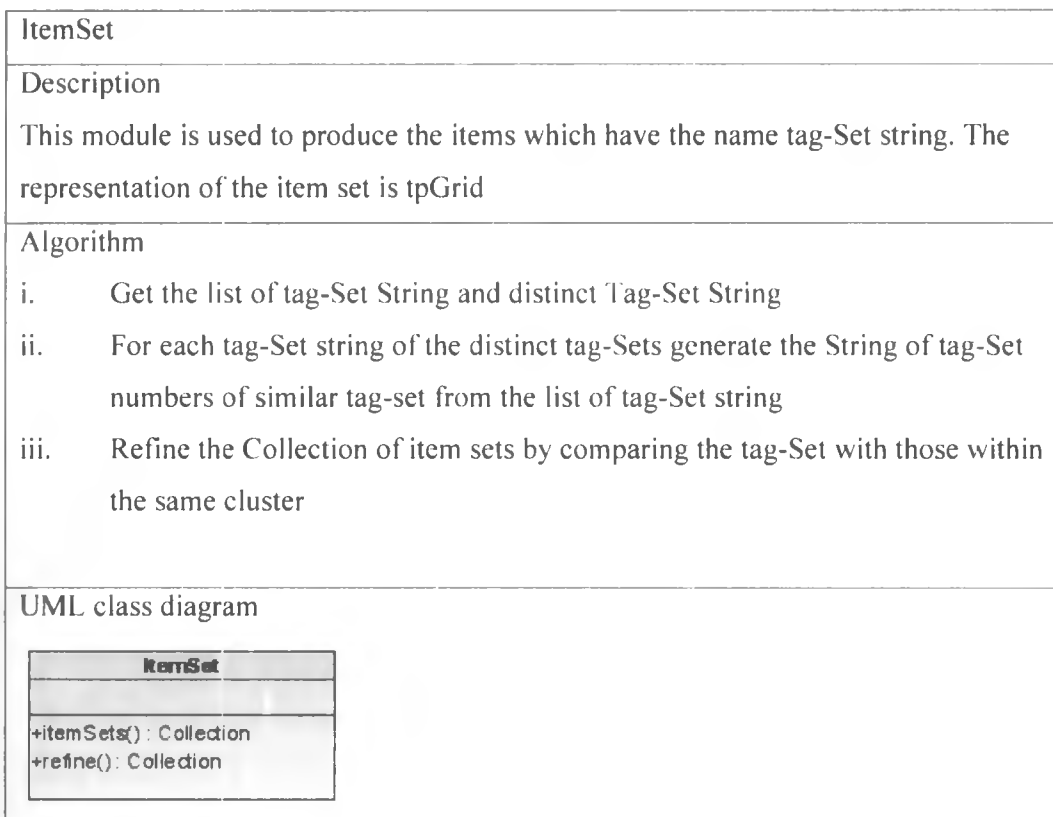
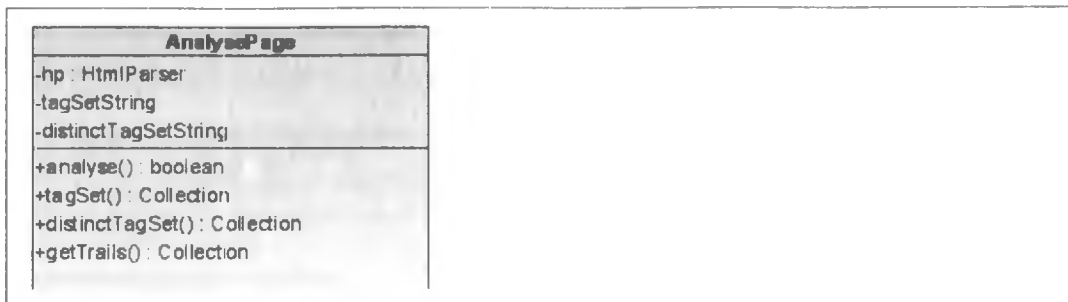
Description

This module produces a list of numbered sequence tag sets with respect to the numbered sequence of the text String.

Algorithm

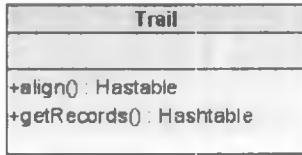
- i. Get a list of tag String
- ii. Generate a list of tag-Sets for each tag-String
- iii. Generate the list of distinct tag Sets from (ii) above

UML Class diagram



- iii. Identify the template record.
- iv. Use the template record to align all the records and represent the optional items (items which are not on the record but on the template record) by -1.
- v. Produce the page descriptor using the item set numbers.

UML class diagram



4.4.4.3 Data extraction

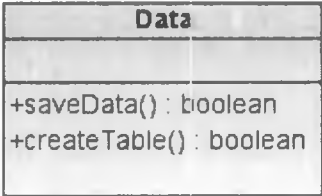
This component has a module which gets the page descriptor information and the collection of text-String from the component page analysis. The module is iterated over the collection of text-String until the start of the data cluster is reached. The module starts creating the first record of the data cluster using the size of the record from the page descriptor. Then the process repeats until the end of the data cluster is reached.

4.4.4.4 Data Store

This component is responsible for all database operations. The database operations range from database connection to execution of database table queries.

Table 15: Database Operations

DataConnection
<p>Description</p> <p>This module is responsible for preparing the database connection using mysql-connector component. The connection object is then used by other modules which are Data and DataTable modules.</p>
<p>Algorithm</p> <ul style="list-style-type: none"> i. load database driver. ii. get database connection.
<p>UML class diagram</p> <pre> classDiagram class DataConnection { +getConnection() : Connection } </pre>

Data
<p>Description</p> <p>Once the data is extracted and ready for storage to the relational database, the module uses java metadata API for creating generic table structures to the database. Once the table is created, then the data is stored to the created table.</p>
<p>Algorithm</p> <ol style="list-style-type: none"> i. Database table is created for new data. ii. The data is inserted into the database table one record at a time.
<p>UML class diagram</p>  <pre> classDiagram class Data { +saveData() : boolean +createTable() : boolean } </pre> <p>The UML class diagram shows a class named 'Data'. It has two public methods: '+saveData() : boolean' and '+createTable() : boolean'. The class name 'Data' is written in bold in the top-left corner of the class box.</p>

DataTable
<p>Description</p> <p>This module uses java metadata API to load the information about the tables already created in the database. It is mostly used when the user wants to view the information already stored.</p> <p>Also with this module, the data is queried from the database based on the name of the database table and user query. This module uses also java metadata API for getting the structure of the database table such as name of the columns created. In this case the content of the table records is read.</p>
<p>Algorithm</p> <ol style="list-style-type: none"> i. Get the database connection object ii. Execute database query iii. For each record add the entry to the collection object. iv. Return the collection of result data
<p>UML class diagram</p>

DataTable	
+getResults()	List
+loadTables()	List
+loadDates()	List
+loadDomains()	List

4.4.4.5 User Interface

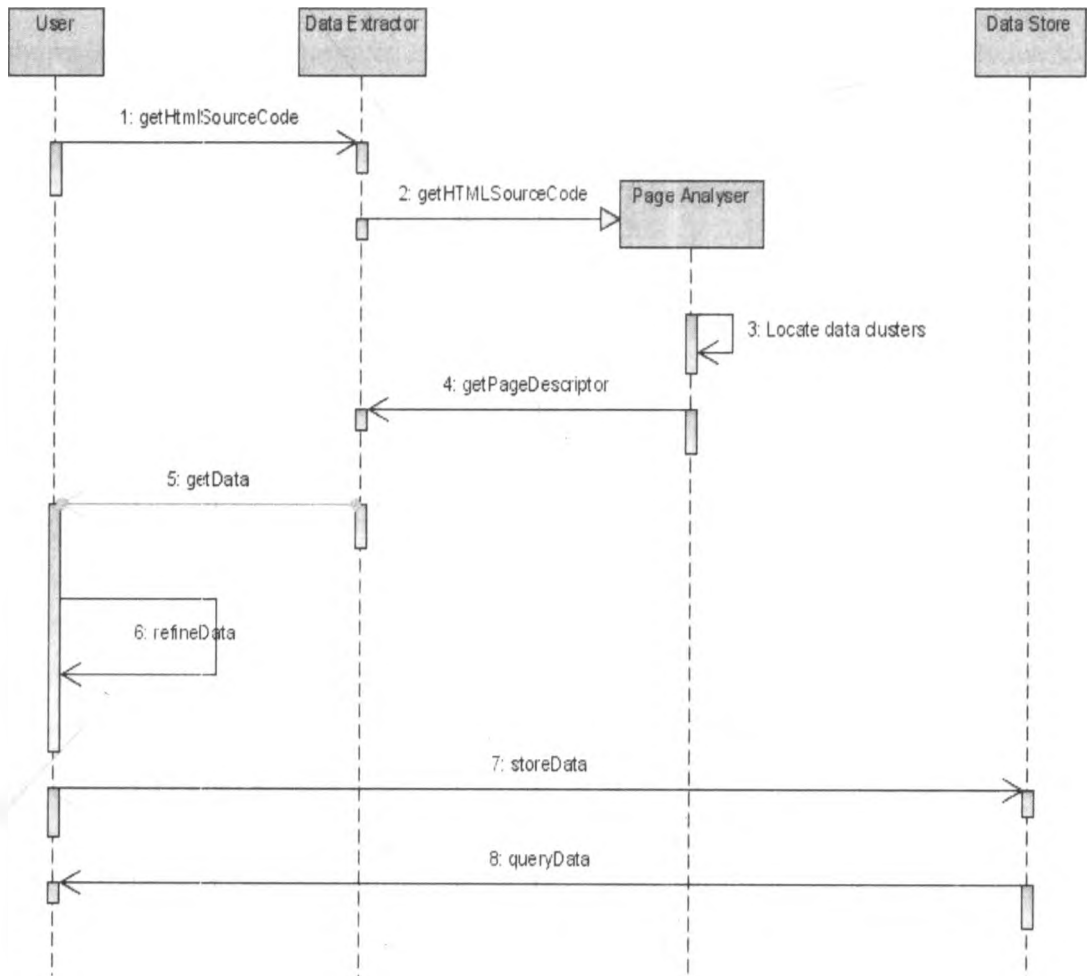
This component represents User interface, in this case, JSP pages, provide a way that the user can interact with the system. With this system. the user can perform two main functions. These functions are to extract data from the web page and to query the information already stored in the database.

When the user is extracting the data from the web pages, the system provides the link to the function for data extraction. The user inputs the text string which is the HTML source code of the web page for data to be extracted. In some web sites when the data is not embedded into HTML code but JavaScript, it is necessary to enter the result web page URL. A good example is the web domain www.ebay.com. User may specify other settings for ignoring some HTML tags to get better results.

After the processing of result web page is done, the data extracted is presented to user. The user in this application needs to select (refine) the information to be stored to the database. The user is presented with all the clusters identified in the web page. The user selects the cluster of his interest. The user also selects the fields of the record of his interest and then specifies the web site domain and the name of the table for the data. Then the data is stored after the user action.

4.4.5 Component Interaction

Figure 9: Sequence Diagram



4.5 Summary

The chapter has managed to analyse and design the algorithm for data extraction. The main challenge is to address the four common problems which face data extraction system as described in section 2.3.2.3. The next chapter will deal with actual implementation of proposed technique

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

In this chapter the implementation of the system is discussed in details. The section shows how the system has been implemented with respect to the suggested solution for the page analysis and data storage. The implementation is discussed basing on the order in which the extraction process takes place.

5.2 Technologies

The study utilizes various technologies that make the project successful, including tools, components and frameworks.

Types of Technologies

The technologies are divided into two categories that is, Application tier tools and components as well as data tier tools and components. The components are explained below.

i) Application Tier Tools and Components

From application point of view the following technologies are used

a) Java and Java Server Pages

Java is most modern the programming language which is gaining much popularity in various fields and applications. Java as an easy language to learn has been used much in academic institutions for teaching and research purposes. Many big software companies support Java. It has been found to be used by many applications and current technologies such as XML and SOAP.

Java has the following advantages:

- (i) It is easy to learn
- (ii) It is object oriented
- (iii) It is platform independent
- (iv) Portability to any machine.

Actually, Java has been chosen since it is easy to learn and understand. But the main reason is that, java has powerful collection framework. With Java collection various models can be produced for the web page during web page analysis. This web data extraction system is developed with the use of JSP technology. In this case it is easy to model three tier system. The JSP pages models, the presentation, and use of java beans in

the middle tiers which model business logical. This strategy has advantages of increasing the system performance and maintenance.

In this project, Java uses metadata which has enough function for performing metadata operations. Metadata technology is used because the database schema will not be defined in advance. When the new web data is extracted the table structure will be defined basing on the structure of the web data extracted.

b) JavaScript

JavaScript is a scripting language that is used for client side programming so as to make the web based applications more interactive. It was developed by Netscape. It was found that JavaScript has strong functions that can be used in this project for processing the HTML source code during page analysis phase. JavaScript has strong regular expression functionalities. Therefore, it is more interactive hence it makes the application to have high performance.

c) Metadata

Metadata can be defined as the information which describes other information. In other words it is a data that describes other data items. Metadata have been successful in writing general codes for software application. In this project, the structure of the database tables is not defined in advance. In this case, the metadata technology is used to get the information which is used to present the data in relation database. This information includes, the name of the table and name and type of the table columns. Metadata technique is used to write general queries to the database. This makes the application of the developed system more general to a variety of applications.

ii) Data Tier Tools and Components

From database point of view, for the purpose of storing the extracted data for the end user and other application software, the following are technologies which have been used in this project.

Microsoft SQL Server

In this project Mysql is used as a database management system. Mysql database management system has the following advantages; it is free, fast and portable database management system. Though it lacks some of the sophisticated features provided by relational database such as Oracle, it has small learning curve compared to oracle

relational database. Microsoft SQL server database lacks the advantage of not being platform independent as it runs on windows environment only.

5.3 Working environment

The system is intended to run on web server, in which case I have used GlassFish Server 3.0.1. The GlassFish Server 3.0.1 is used because it is an open source and it is free available. GlassFish Server 3.0.1 can be free downloaded from glassfish web site (<http://glassfish.java.net/>). Once the web server is installed into the system, settings for the environmental variable are performed.

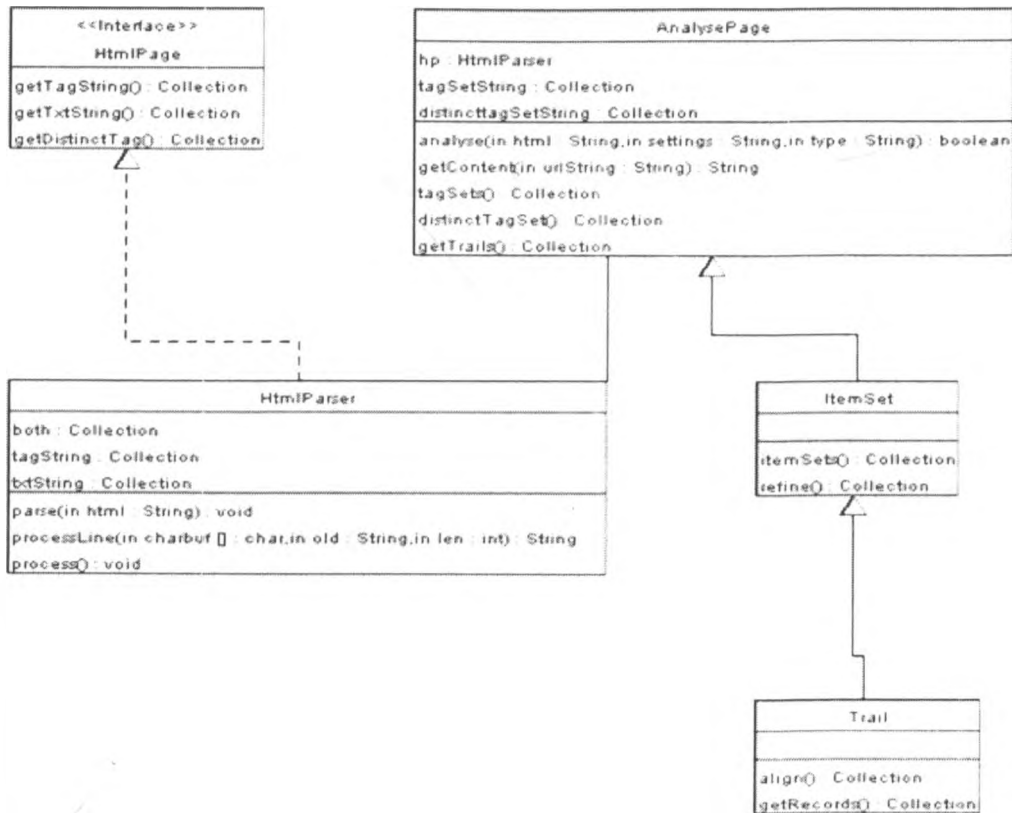
After the environmental variables have been set, the system is deployed on the “webapps” folder of GlassFish Server. The structure of the application is such that, the Java Server Pages files are located on the application folder. The application folder has WEB-INF as a sub-folder which in turn contains two sub-folders. These folders are “classes” which contains java beans components and “lib” folder which contains mysql-connector for mysql database connection.

In this project I have used Mysql as a database management system. The Mysql database can freely be downloaded from the www.mysql.com website. The setup file is downloaded and then being installed on the computer system. When the Mysql database is installed and all the installation settings are made, then the database is ready for use. In this project only one table is created during system deployment. This table has a schema as shown on the database design part. It has four fields, ID, the domain, keyword, and the date when the data is extracted. The value of date is taken from the system time. The other tables which store the data extracted form the database are created automatically data extraction and storing. The schema for these tables depends on the structure of the data extracted .The general row names are used to name the database table fields. Now the system is able to run under these environments.

5.4 Page analyser

In the component of the page analyser, there are four classes and one interface. These are HtmlParser, AnalyserPage, ItemSet, Trail and HtmlPage respectively.

Figure 10: Page Analyser



HtmlPage and HtmlParser

The class HtmlPage is the interface which is implemented by the class HtmlParser. The class HtmlParser parses the HTML source code which is in the form of a text string, and produce the list of HTML tags and text (non tag part) in the order they appear on the HTML source code. This list is further processed to create separate lists. One of the lists is a list for tag-String and the second one is the text-String. These lists are stored on the HtmlParser object fields. The list of tag-String is processed to get the list of distinct HTML tags that have been used in the web source code.

AnalysePage Class

This class creates an object HtmlParser. The AnalysePage uses the list of tag-String and text-String to produce the list of tag-Set and a list of distinct tag-Set. These lists are used for the purpose of creating the tpGrid. The class AnalysePage uses the method from the utility class for producing the trail of tag-Set numbers. This trail has two parts. The first part is the tag-Set number and the second part is the first similar tag-Set number when the list of tag-Set is traversed. In this implementation when the tag-Set number occurs only

once, the entry of -1 is assumed. The aim of doing this is to sort out the tag-Set with repetitions from those which occurs only once.

ItemSet class

This class extends the class AnalysePage and hence it uses all the methods and attributes of the super class. The aim of the ItemSet class is to create the list of items which shows the repetitions of similar tag-Set numbers. This means that the tag-Set numbers which have the same tag-Set will be grouped together. The model produced is the representation of the web page called a tag-Set progressive Grid. The tpGrid shows the blocks of data clusters and hence the analyser can automatically identify those clusters.

The class also has the refined representation of the tpGrid. This version of tpGrid tends to group those repeated similar items which are on the same cluster. This approach prevents the problem where the similar items are grouped from different clusters.

The Trial Class

The Trail class extends the AnalysePage class. This class has two main processes. First the class uses the list of trails which show the repetitive patterns, to produce the intermediate records. These records can be of the same pattern when there is no optional field or nested structures. When there are optional fields these records will have different number of fields for each record.

The second process aligns field patterns. The first template record is chosen with the assumption of the longest record. The other records are compared using the pattern matching technique. The algorithm for pattern matching is simply matching the string with the template method. In this case the items which are missing are substituted by index -1.

5.5 Data Extractor

This component does not have any class but Java server pages. This is because the extraction process is assisted by system user for data scrubbing.

Process.jsp

This page receives two items. The first item is the list of text-String and the second item is the page descriptors in form of tag-Set numbers. The tag-Set numbers are used to identify the corresponding text-String. In this case, data extracted is presented to the user

in form of a table showing the structure of the data cluster. Since in the page there can be more than one cluster, the system provides with user the option to select the cluster of interest. Once the cluster is chosen, the user is given another JSP page called cluster.jsp.

Cluster.jsp

This page shows the cluster which was chosen by the user. The user is given more option to select the columns of interest. This is because some of the columns just present labels or they are empty. The user is provided with check box control to select the columns of interest. Then the user is given another page called processcluster.jsp.

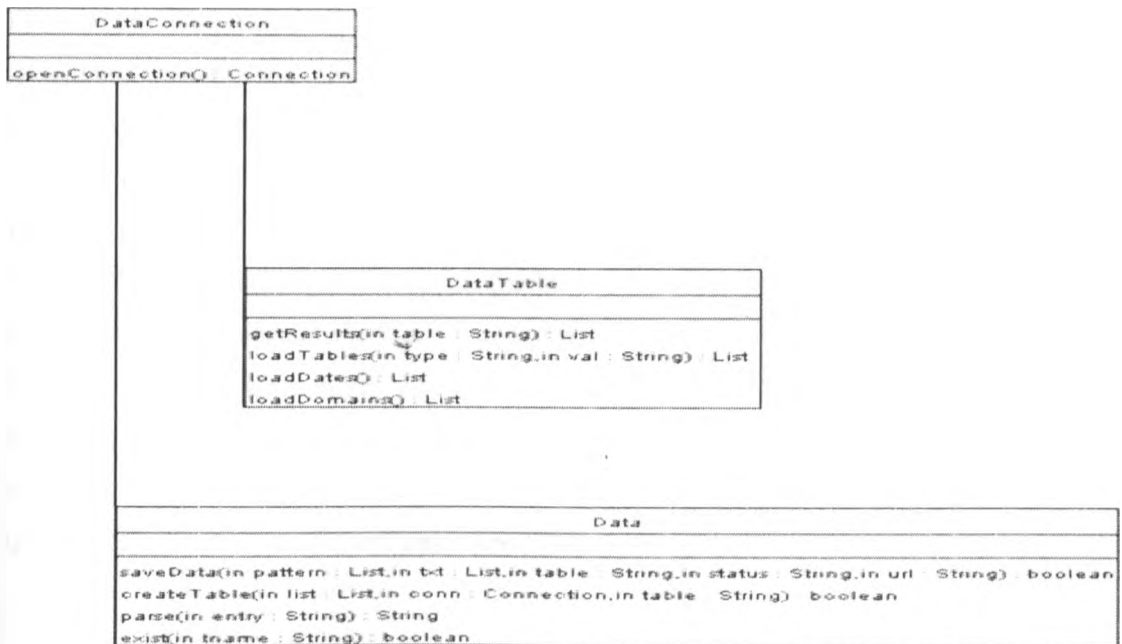
Processcluster.jsp

This page shows the refined data and provides the control for specifying the name of the table for storing the data to the database. The first control is text box for the name of the web site domain used for searching and extracting data. The second control is the text box for the name of the database table. The last control is the combo box for selecting the existing tables where the data is to be inserted.

5.6 Data storage

This component is composed of some Java classes and some JSP pages.

Figure 11: Data storage



The DataConnection Class

This class provides the connection object from the Mysql database. It is used by other classes for getting the connection object to the database. The objects of this class stores the database connection object.

The Data Class

This class has two main functions. The first function is to create the database table. The database table created is named after the name specified by the user. The fields of the database table use generic names and the data type are entirely text string. The class has another function of removing the single quotes which can occur on the texts and hence prevent introduction of SQL injection problems.

The DataTable Class

This class is used for querying the information from the database. It has the method `getResults()` to read data from the database tables as specified by user. There is a method `loadTables()` to show the tables which have already been created. There is a method `loadDates()` which shows the date when the specified table was stored into the database. There is also the method `loadDomains()` which specifies the web site domain where the data have been extracted.

5.7 User Interface

This component is composed of all the JSP pages which help the user to interact with the system for various functions. This is the main system interface. The user can get all the system services such as to extract data from web pages and query data from the database.

The index.jsp

This page is that start page. It contains the links where the user can select which service to perform. In this system there are four services. The first service is for data extraction from web pages. The second service is data query from the database as a local cache. The third service is page analysis which shows intermediate entities of the page during page analysis phase. These entities are the list of text-String, list of tag-String, list of tag-Sets, tpGrid, refined tpGrid, trails and page descriptors.

Data Extraction Service

In this service, the user is presented with first.jsp page which has text area for HTML source code, check boxes for HTML tag settings and a button control. When the user submits the entries, process.jsp page gets the HTML source code and call the java classes for page analysis and hence data extraction. The data extracted is presented to the user and the process of refining continues until when the data is stored to the database.

Data Query Services

This service has two JSP pages; queries.jsp and query.jsp. The queries.jsp page interacts with the data Store component to get the information about the database tables created such as date, keyword used for data searching on the web page and web domain. The user selects to query database tables based on the date, keyword or web domain.

The query.jsp web page is used by the user to specify the name of the database table using the combo box control. The user can restrict the query so that specific information is queried by entering the keyword. Then the information that is stored into the database table is displayed to the user.

Page Analysis

In this service, the user submits the HTML source code and specifies HTML tag settings to the page called exp.jsp. This page provides a combo box control and the user can select the intermediate entities produced during the analysis. This service is mainly used for investigation purpose during page analysis.

5.8 Summary

From the exploration of various technologies discussed above for the implementation of this system is possible. The system is implemented using java as a programming tool. The database management system for this application is Mysql database. The other tool is the web server application where the system will be running. The choice for the web server application is Glassfish server 3. The component which has been used by the system is mysql-connector which is used by java JDBC for Mysql database connection.

CHAPTER SIX: EXPERIMENTAL RESULTS

6.1 Introduction

This chapter deals with testing strategies, experimental layout and performance result of the system which has been developed and find out if it achieves the intended result. The testing is organised by setting the system in the order in which data was extracted. The testing is divided into two parts; the first part involves system testing and the second part involves algorithm testing. Comparison between this study and other corresponding systems is done using precision and recall to test completeness and accuracy of the proposed algorithm.

6.2 Testing strategies for system development

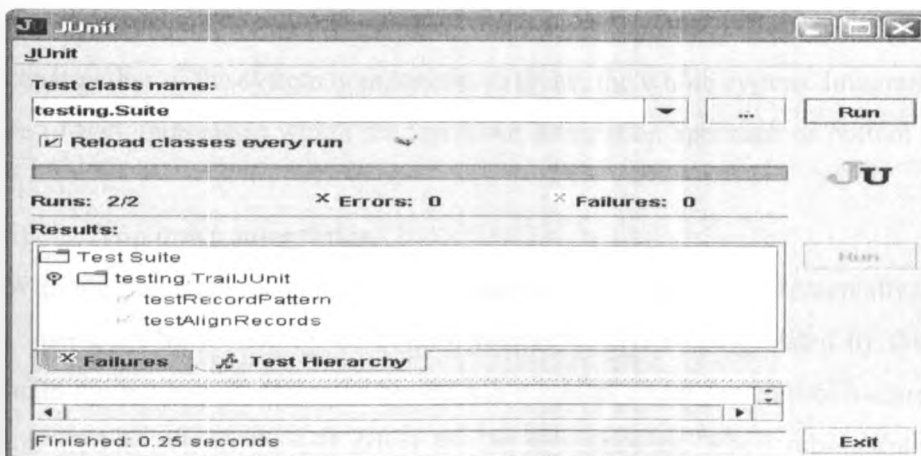
Since testing is the act of designing, debugging, and executing tests then during the development of the system, there are testing strategies that are used to ensure that the system is error free. The main testing strategies are Unit Testing, Integration Testing and System Testing during system development using JUnit Framework.

6.2.1 JUnit Framework

This is the framework developed for testing during the extreme programming. It is very useful for developing test cases. Using JUnit testing framework methods of each classes are tested by specifying the input data and comparing the expected output and the real output data.

Below is the graphical interface showing the results of the tests on some of the methods which have been tested.

Figure 12: Some of the method tested.



6.2.2 Unit Testing

During the implementation of the system unit testing has been done in the form of white box testing and black box testing. With white box testing, each statement of the code is tested against predefined input data. Then the result of the statement is compared with the expected output. If the case of branch statement paths is tested to ensure that the execution follows the expected path coverage. With loop structures, the first two iterations are tested and the last iteration to ensure the loop does not enter into infinity execution state under any input data. When the system is implemented, each line is tested to see if it produces the expected output. Otherwise if there is any deviation from the expected output the statement is re-defined. With the black box testing some system functions, modules or structures are checked against the input and the expected output. These structures are loop structures, methods (functions or sub routines). The internal workings of the component need not to be known. Black testing can be seen as redundant as the white box tests all the individual statement within the function. The black box is necessary to ensure that all the statements within the function produce the expected result from the functional point of view. Black box does not need programming skills but rather input and output data. During the black box testing, the functions are tested against three sets of data. These data sets are the data inside the test case, the data outside the test case and the data in the boundary case. The data from the outside set is used to test the system robustness.

6.2.3 Integration Testing

The integration testing is done when all the software units have been tested. In integration testing it is necessary to ensure that the data consistency is preserved across the interface. Also the integration test ensures that one component does not have effect on the functioning of the other component. The integration test is done systematically with the construction of the system components to create the whole system. Integration testing can be of two approaches which are top down integration approach or bottom up integration approach.

a) Top down integration

With top down approach the system structure is constructed incrementally. The system is constructed by following the control hierarchy. After being tested by the test stub the main top module is integrated by the sub-modules under the control hierarchy. The main top module acts as the test driver for the sub-modules. When sub-module is tested and

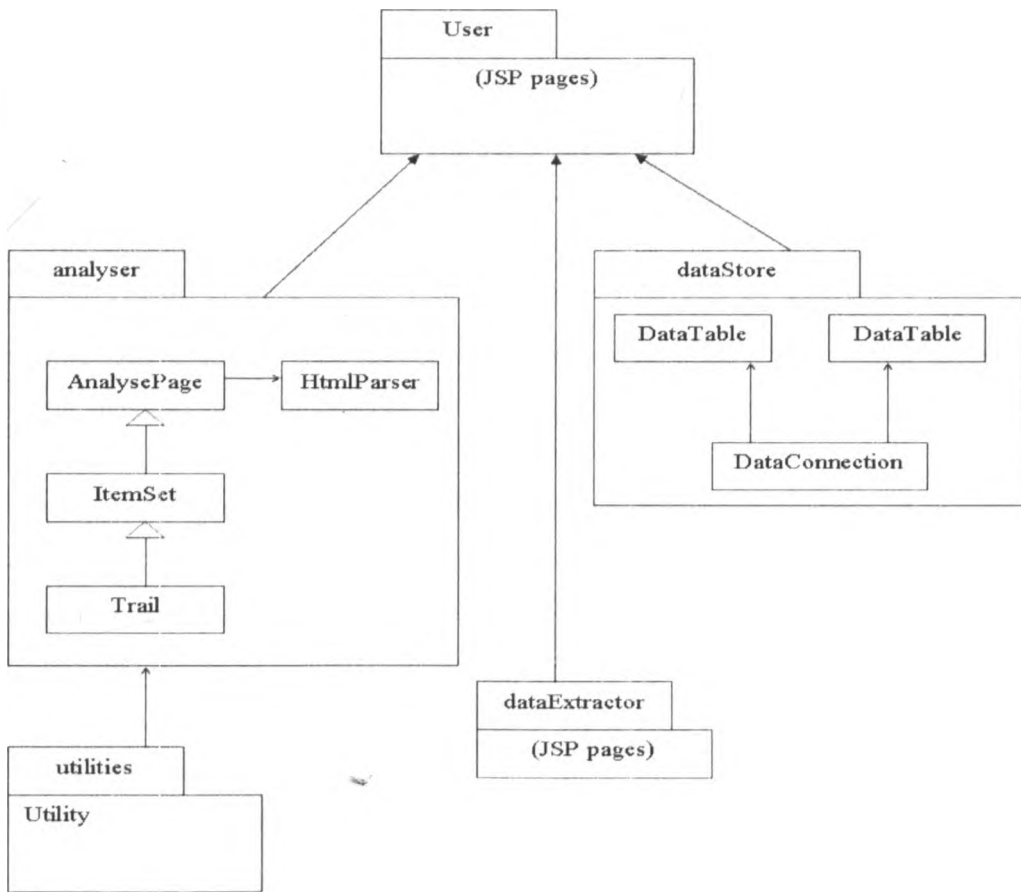
found to have no error it is integrated with the top modules. The process continues until all the sub modules are integrated.

b) Bottom up integration

With bottom up approach, the modules are being tested from down to the top of the control hierarchy. The lowest level modules are tested with the test driver and then are integrated to form a large sub system. The process continues in each case building a test driver and used for testing the sub modules until the whole system is integrated.

In this project bottom up approach is used for integration testing and system construction. Below is the figure showing the integration testing hierarchy from bottom to the top level.

Figure 13: Bottom up Integration



6.2.4 System Testing

When unit testing, component testing and integration testing have been done successful, then the whole system is tested against system behaviours. System testing explores system behaviours which are neither tested by unit testing, component testing nor integration testing. System testing involves performance testing and installation testing.

a) Performance Testing

Performance testing determines how fast the system will behave at a particular work load. The testing is done to check whether the performance of the system is affected by the working environment which includes amount of Random Access Memory, Size of the software package that is taken to the hard disk and the speed of the computer processor. The performance of the system is found to increase when the system runs of the computer with high performance measures, which are processor speed, size of Random Access Memory, and the amount of space on the Hard Disk. The other causes of the low system performance if the technology used to implement the system. The investigations show that Java programming language suffers performance drawbacks when compared with other programming languages including PHP, C++, ASP.NET and CGI.

b) Installation Testing

Installation testing is done to determine if the system can be installed outside the development environment. The testing is done to check whether the new environment affect the functioning of some system component or the performance of the system is affected by the new environment.

In this project the system has been installed in windows operating systems (Windows 7) where the Java Virtual Machine has been installed. The system was found to be working properly without any changes to system operations. Hence further testing can be done in other operating systems.

6.3 Experimental Analysis

6.3.1 Experimental Objectives

The purpose of this experiment is to evaluate both the effectiveness and the efficiency of this algorithm using two data sets. The second objective is to compare system developed performance with that of Happy Harvester 2. Since the comparison is between system developed and the Happy Harvester system, the two data sources are from the data set that has been utilised by Happy Harvester system and the other data set is the one which

is proposed. The two systems are tested by using the two data sets and their precision and recall have been identified.

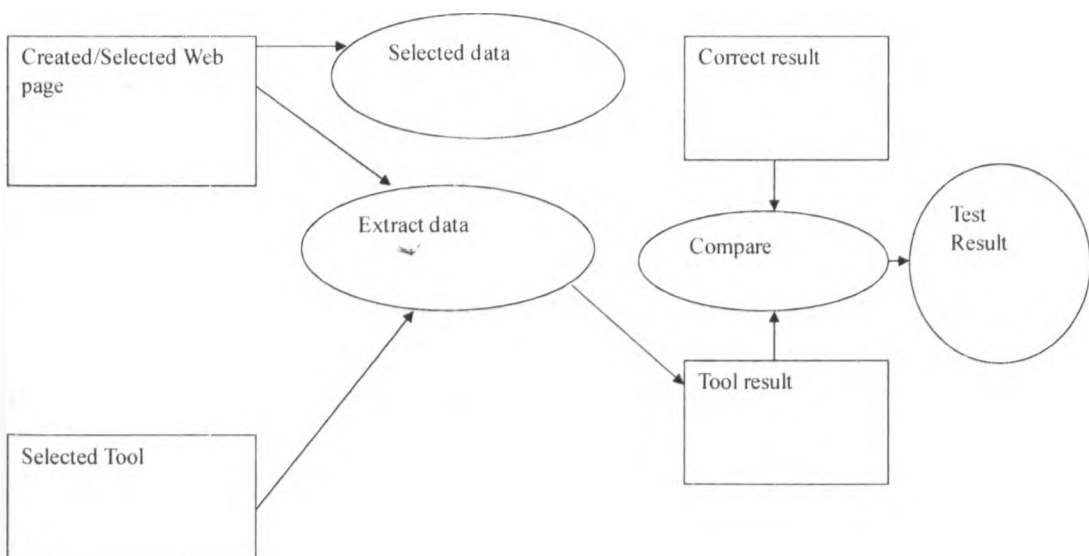
6.3.2 Experiment Setup

The experiment has been carried out on a Pentium 4 computer with a 3.2GHz CPU and 2G of RAM as minimum requirement and with windows operating systems (Windows 7) where the Java Virtual Machine is installed. The algorithm has been implemented using Java as programming language and the JavaScript has been utilised to take care of dynamic HTML pages. The source of data set is from the Happy Harvester system and the one which has been collected based on the factor affecting data extraction process. Happy Harvester is closely related system similar to one which is being proposed although it operates manually. The choice of web pages takes into consideration various sectors which include business, education as well as government websites. The choice of websites takes into consideration the factors affecting data extraction.

6.4 Methodology for Comparison of the two Systems

To realize the results, a test methodology is proposed which assists on getting the final results and also tries to direct steps which must be followed. The techniques used are manually compared with the extracted data with the original HTML pages data to check for their correctness and completeness.

Figure 14: Methodology for Tools Comparison



The above methodology operates as follow:

- (i) The first step consists of creating or selecting a web page source in which we want to extract data. This test uses web pages found in the web but self made pages can be created to focus in some specific features that can not be found in the web or the one which we want to test specifically. This can be possible by creating and locating such a web in the private free web server. e.g Gillfish server.
- (ii) Use our tool to extract data from respective web either created or selected web pages
- (iii) Extract the data using our tool
- (iv) The output from this tool is compared with the correct extracted data (either manually extracted data or by another tool)
- (v) Test the obtained results whether they qualify the data extraction results by quantifies, they are very good, good or poor or give out percentage depend on recall and precision value that will be obtained.

6.5 Performance Metrics

There are two metrics for measuring performance, precision and recall. In this project the performance metrics are tested using two variables known as recall and precision and two data sets. The explanation for recall and precision is as follow.

In information retrieval contexts, precision and recall are defined in terms of a set of retrieved documents (e.g. the list of documents produced by a web search engine for a query) and a set of relevant documents (e.g. the list of all documents on the internet that are relevant to a certain topic).

Precision

In the field of information retrieval, precision is the fraction of retrieved documents that are relevant to the search.

Miao Gengxin et al (2009), Precision can be defined using the following formula

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positive}}$$

This can be expressed as follow

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

Recall

Recall in information retrieval is the fraction of the documents that are relevant to the query that are successfully retrieved.

Miao Gengxin et al (2009), Recall can be expressed using the following formula

Recall = $\frac{\text{true positives}}{\text{ground truth}}$ which is the same as the following formula

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

It is trivial to achieve recall of 100% by returning all documents in response to any query. Therefore recall alone is not enough but one needs to measure the number of non-relevant documents also, for example by computing the precision.

Fall-Out

The proportion of non-relevant documents that are retrieved, out of all non-relevant documents available.

$$\text{fall-out} = \frac{|\{\text{non-relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{non-relevant documents}\}|}$$

It is trivial to achieve fall-out of 0% by returning zero documents in response to any query.

6.6 Algorithm Implementation

The algorithm has been successful implemented and it can produce the following:

- a) Tag string and Text string (produce distinct tags).
- b) Tag set (produce distinct Tag set).
- c) Item set.
- d) Data clusters.
- e) Page Descriptor.

The implemented algorithm has accuracy and precision of 100% respectively on doing the above important stages.

6.7 Accuracy Analysis

The experimental results for our algorithm compared with Happy Harvester are shown below. We run both algorithms for all of the Web pages in the first data set and second data respectively where recall and precision is calculated. The techniques used are manually compared the extracted data with the original HTML pages data to check for their correctness and completeness.

Table 16: Comparison between Happy Harvester and Cluster system

No	Web domain	Happy Harvester		Cluster System	
		Recall	Precision	Recall	Precision
1	http://www.happyharvester.com/example1	100%	100%	100%	100%
2	http://www.happyharvester.com/example2	100%	100%	100%	100%
3	http://www.happyharvester.com/example3	100%	100%	100%	90%
4	http://www.happyharvester.com/example4	100%	100%	100%	80%
5	http://www.happyharvester.com/example5	100%	100%	100%	100%
6	http://www.happyharvester.com/example6	100%	100%	100%	80%
7	http://www.happyharvester.com/example7	100%	100%	100%	50%
8	http://www.happyharvester.com/example8	100%	100%	100%	70%
9	http://www.happyharvester.com/example9	100%	100%	100%	80%
10	http://www.happyharvester.com/example10	100%	100%	100%	90%
11	http://www.happyharvester.com/example11	100%	100%	0%	0%
12	http://www.happyharvester.com/	0%	0%	0%	0%

	<u>example12</u>				
13	<u>http://www.happyharvester.com/ example13</u>	100%	100%	100%	100%
14	<u>http://www.happyharvester.com/ example14</u>	100%	100%	100%	100%
15	<u>http://www.happyharvester.com/ example15</u>	100%	100%	100%	100%
16	<u>http://www.happyharvester.com/ example16</u>	100%	100%	100%	100%
17	<u>http://www.happyharvester.com/ example17</u>	100%	100%	100%	90%
Average		94.12%	94.12%	88.24%	78.24%

As it can be seen from Table above, the developed system achieves better average recall and precision values which are 88.24% and 78.24% respectively. The Happy Harvester system, however, only achieves 94.12% for recall and 94.12% for precision, suggesting that the developed system performs well similar to the Happy Harvester system on these Web pages.

The second Data Set Results

The data have been sampled basing on the problems that face data extractor.

Table 17: Comparison between Happy Harvester 2 and Cluster system

No	Web domain	Cluster System		Happy Harvester	
		Recall	Precision	Recall	Precision
1	<u>www.uonbi.ac.ke</u>	100%	100%	0%	0%
2	<u>www.japanesevehicles.com</u>	100%	100%	0%	0%
3	<u>www.nation.co.ke</u>	100%	100%	0%	0%
4	<u>www.kelkoo.com</u>	100%	100%		
5	<u>www.tesco.co.uk</u>	100%	70%	0%	0%
6	<u>www.overture.com</u>	100%	100%	0%	0%
7	<u>www.bl.uk</u>	100%	100%	0%	0%
8	<u>www.planepictures.net</u>	100%	100%	0%	0%

9	www.alltheweb.com	100%	100%	0%	0%
10	www.encyclopedia.com	100%	100%	0%	0%
11	www.all4one.searchallinone.com	100%	100%	0%	0%
12	www.highbeam.com	100%	100%	0%	0%
13	http://campus.acm.org	100%	100%	0%	0%
14	www.yahoo.com	100%	95%	0%	0%
15	www.google.com	100%	100%	0%	0%
16	www.argos.co.uk	100%	60%	0%	0%
17	www.ebay.com	100%	95%	0%	0%
18	www.amazon.com	100%	65%	0%	0%
19	www.mamma.com	100%	100%	0%	0%
20	www.bbc.co.uk/food	100%	100%	0%	0%
21	www.ibm.com	100%	100%	0%	0%
Average		100%	94.52%	0%	0%

As it can be seen from Table above, the developed system achieves better average precision and recall values which are 94.52% and 100% respectively. The Happy Harvester system, however, achieves only 0% for precision and 0% for recall, suggesting that the developed system outperforms the Happy Harvester system on these Web pages.

6.8 Time Complexity Analysis

The algorithm consists of three steps. We analyze the time complexity for each step individually.

Page Analyser

It is the first step in this algorithm which provides information about the location of web data to extract. It consists of four classes; HtmlParser, analysePage, ItemSet and Trail. The purpose of these four classes is to scan the web page and mark the repetitive pattern which marks the existence of data cluster, hence data to be extracted. The process takes $O(L)$ time for HtmlParser and analysePage. where L is the total number of HTML tags occurrences in the web page. By calculating the itemSet and Trail for location of data to be extracted using distinct HTML tags, the process take $O(M \times L) + O(M^3)$ where M is the number of unique HTML tag. Thus, the step of identifying data cluster, hence data to be extracted takes $O(M \times L) + O(M^3)$ time in total.

Data Extraction

Once the information for locating the data clusters in the web page has been produced, the data extractor can easily extract data. This is because each of the text-String number corresponds to the tag-Set number. So, the data extractor walks over the list of the text-String. When the text-String number that corresponds to the start of the data cluster is encountered, the data extractor starts creating a record. The size of the record is given as part of the information for page descriptor. If the size of the record is Z where Z is a positive integer and the start of the data cluster is tag-Set N where N is a whole number such that N is greater than or equal to Zero and N is less than the number of entries of the List of text-String, then the first record will be extracted as shown below, given that the list of text-String items is TS .

Table 18: Data Extractor

Record	Field 1	Field 2	...	Field Z
1	$TS[N]$	$TS [N+1]$...	$TS [N+Z-1]$
2	$TS [N+Z]$	$TS [N+Z+1]$...	$TS [N+2Z-1]$
3	$TS [N+2Z]$	$TS [N+2Z+1]$...	$TS [N+3Z-1]$
...
K	$TS [N+(K-1)*Z]$	$TS [N+(K-1)*Z+1]$...	$TS [N+K*Z-1]$

The data extractor will stop extracting at the end of the data cluster, which is marked by the tag-Set number say M , when M is the whole number such that M is greater than N and M is less than the number of entries of the List of text-String. The data extractor will stop extracting when the value of M is equal to $TS[N+K*Z-1]$ as shown below.

$$M = TS [N+K*Z-1].$$

This step is intended to extract the data from web pages before the data are stored on the data store. This process makes use of information from the page analyzer to extract the information from the web page. The number of HTML tags visited is still less than L . Thus, the time complexity of the data extraction step is $O(L)$.

Data Storage and Query

This step intends to store data extracted into database or data warehouse and also allow user query to the database. This process makes use of information from page analyzer as well as from data extraction. Thus, the process of store and retrieve data from database takes $O(L)$ time.

Hence, in total, the time complexity of the algorithm is $O(M \times L) + O(M^3)$, where L is the total number of tag occurrences and M is the number of unique tag in the web page.

Compared to Happyharvest, whose time complexity is $O(L^2)$. Since it operates basing on similar approach where algorithms traverse a DOM tree and apply edit distance computation between sibling subtrees.

Let N be the number of children of each node. At the root, the algorithms compute the edit distance between its children with size L/N , taking $O((L/N)^2)$ time. Happy Harvester computes the edit distance N times. At depth d , there are N^d trees, each of which has N children of size L/N^{d+1} . The total cost is $\sum_{d=0}^{K-1} (L/N^{d+1})^2 N^k N^d = L^2 N^{k-2} \sum_{d=0}^{K-1} (1/N)^d < L^2 N^{k-2} \times N / (N-1)$ where $K=1$. Thus, the time complexity of Happy Harvester is $O(L^2/N)$. From this analysis, we conclude that Happy Harvester is efficient ($O(L)$) when the document structure is simple (and N is as large as L). However, if the document structure is complex, happy harvest is not as scalable. Hence my proposed algorithm outperforms that for happy harvest system.

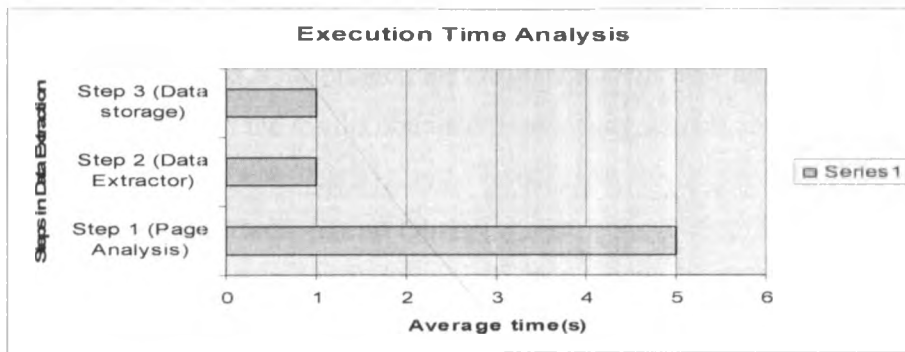
6.9 Execution Time Analysis

In this project, in the worse case the system has a performance of five seconds during the page analysis stage. When the page descriptor has been created the system is found to have a performance of less than one second.

Table 19: Execution Time Analysis

Function	Average time (s)	Percentage
Step 1	5	83.32%
Step 2	<=1	8.34%
Step 3	<=1	8.34%
Total execution time	<=6	100%

Figure 15: Execution Time Analysis in graph



6.10 Summary

- As it can be seen from the first table above, the developed system achieves good average precision and recall values which are 78.24% and 88.24% respectively. The Happy Harvester 2 system, however, achieves only 94.12% for precision and 94.12% for recall, suggesting that Happy Harvester using its data performs well compared to our system.
- As it can be seen from the second table above, our system achieves better average precision and recall values which are 94.52% and 100% respectively. The Happy Harvester 2 system, however, only achieves 0% for precision and 0% for recall, suggesting that our system outperforms the happy harvest system on these Web pages.
- Using the two tables above, we can conclude that, our system outperforms Happy Harvester 2 system.
- Both Happy Harvester 2 and the developed system are efficient when the structure is simple (and N is as large as L). However, if the document structure is complex, Happy Harvester is not scalable.
- The time complexity of our algorithm is $O(L)$ for practical data sets (it is linear in the document length) which are more efficient compared to Happy Harvest which is $O(L^2)$.
- The number of unique html tag does no increase as the number of HTML tags increases.
- As number of HTML tags increase ,the run time for page analyzer also increases
- Happy Harvester has managed to detect nested structures (manually) while the developed system has managed to detect nested structure automatically.

CHAPTER SEVEN: DISCUSSION

7.1 Introduction

This chapter evaluates the project; the chapter explains how the intended objectives have been achieved from the results obtained from testing section above.

7.2 The main Findings and Observations

The following are the main findings and observations as a result of running the system.

The following are the problems observed and be solved

(i) Distorted structure

There is displacement of data cluster whereby tag-set of the data cluster appear in another location which is not part of the data cluster. This make identification of rows which form data clusters difficult. To solve this problem restructuring of the tpGrid has been implemented in the algorithm automatically so as all the items which form the data cluster are grouped together.

(ii) HTML optional tags

There is irregular structure which hide data cluster, the irregular structure is caused by some html tags which is used to putting more emphasis or to formatting the presentation. To solve this problem, the algorithm has included an optional for analyzer to ignore those HTML tags. These tags include bold, subscript and superscript, image tag, link tag and break line tag. When HTML tags is ignored the good results is produced but HTML tags can not be ignored in all cases. So the problem is when to ignore the HTML tags

(iii) Records with Optional Field

There is irregular structure which occurs because of structure of data. The record is not the same for the same data clusters. To solve this problem, the record templates have been implemented where data record with the highest number of items will be taken as reference. The remain missing record will be marked with optional fields by entry -1, which indicates that there is missing item in that entry.

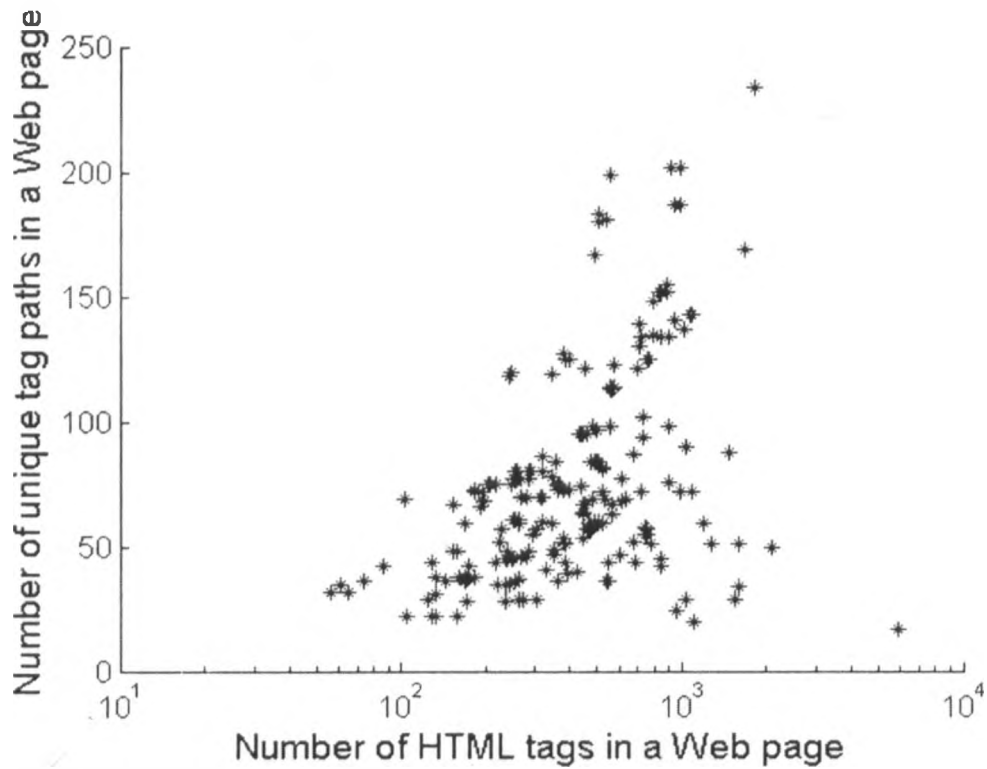
(iv) Nested data clusters

There is a problem of displaced data fields extracted in the wrong column. This is due to existence nested data clusters. To solve this problem, the same approach of introduced record templates have been considered with little success. More focus should be extended to this idea.

(v) HTML tags

The number of unique tag paths does not increase as the number of HTML tags increases.

Figure 16: Unique tag Vs HTML Tags

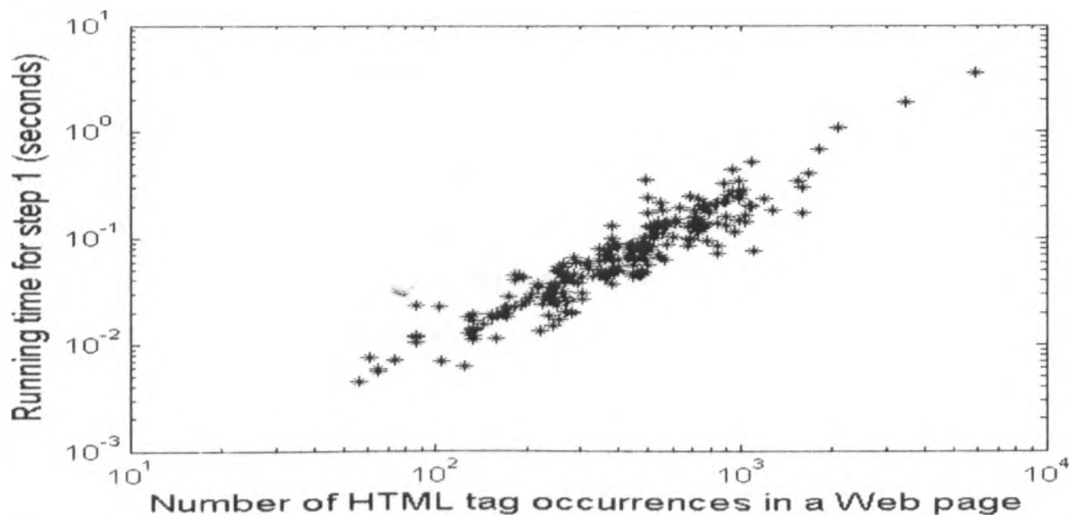


Source: Miao Gengxin et al (2009)

(vi) Running time for page analysis

Page analysis is linear in the document length.

Figure 17: Running time Vs Number of HTML Tags



Source: Miao Gengxin et al (2009)

(vii) Domain-independent technique

The algorithm depends only on the web page structure without examining the web page content, which makes it a domain-independent approach. The algorithm is suitable for handling web-scale data because it is fully automatic and does not need any information other than the Web page.

7.3 Exceptions

The approach proposes working on web page with database where user specifies the keyword so as to extract the data from the database. The system can not search or locate the respective website on its own. Instead the user must know the url of website and type of data she/he is looking for.

7.4 Relationship to previous work

This research is related to wrapper design, data extraction systems, web market monitoring, web process integration and web mashups. Where mashup is a new web application that combines a number of different websites into an integrated view. Usually, the content is taken via APIs by embedding RSS or atom feeds similar to REST (Representational State Transfer).

7.5 Theoretical or practical Implications

In this project work, we solve our research problem from two perspectives both theoretical and technical. From the theoretical perspective, our work summarizes the existing knowledge in our main field of work which is extraction models, information extraction and information retrieval techniques. Consequently, different approaches for extracting the request from the models and for expressing the request in a query have been discussed with their effects on the quality of the retrieved results. From a technical perspective the work aims at illustrating concrete application of the knowledge gained from the existing literature. The key applications include web market monitoring, web process integration and web mashups.

7.6 Achievements

There are a lot of achievements in this project. This project has been conducted with respect to investigations.

- (i) The task of identifying the data clusters has been automated with the use of repetitive features using tag-String. The method technique used in this project has been the extension to the project done by Robinson (2004). Where as other techniques tend to be specific to certain web pages. This technique is automatic in the sense that, the structure of the web page for data clusters is analysed automatically by the system.
- (ii) The settings for some HTML tags used by certain web domain for formatting the query results has been investigated and found to provide good results. This project has been successful in investigating some HTML tags which affect the process of clusters identification.
- (iii) The new method has been investigated and implemented in the problem where the numbers of fields of a record of a data cluster is different to other records. The developed system is found to produce good results. This main structure is found to be the presence of optional fields.
- (iv) The system to various web domains which provides data searching functionalities has been tested. The figure below shows the web domain tested and the quality of the results produced.

Table 20: Web domain tested and the quality of the results produced

Web domain	Quality of results
www.uonbi.ac.ke	100%
www.japanesevehicles.com	100%
www.nation.co.ke	100%
www.kelkoo.com	100%
www.tesco.co.uk	70%
www.overture.com	100%
www.bl.uk	100%
www.planepictures.net	100%
www.alltheweb.com	100%
www.encyclopedia.com	100%

www.all4one.searchallinone.com	100%
www.highbeam.com	100%
http://campus.acm.org	100%
www.yahoo.com	95%
www.google.com	100%
www.argos.co.uk	60%
www.ebay.com	95%
www.amazon.com	65%
www.mamma.com	100%
www.bbc.co.uk/food	100%
www.ibm.com	100%

- (v) It has been found that, metadata functionalities should be used to produce the general codes. These codes can work in any system. In this case the schema for the data in the database is defined automatically based on the structure of the data extracted.
- (vi) There are some web domains where the project has not yet implemented the proposed solution due to the time constraints. This is the case when there are nested data clusters. The good example of the web domain which produces results with nested structures is www.amazon.com. The current implementation works between 60%-70% of the good results.

7.7 Constraints

The test environment consisted of a laptop computer that had capacity of 230GB, 2GB memory, Intel(R) Pentium (R) Dual CPU T 2370 @ 1.73GHz with window 7, with internet connection so as to access various website. The system makes use of gillweb server and makes use of msq1 server which is not faster. These resources are very minimal to realise the true function of data extraction. Hence, the preferred setting with appropriate setting should be deployed such that the system runs in an environment which has internet connection so as to have an access to the web pages. Otherwise, it should be deployed online so as to be accessed. Otherwise self made website which focuses on specific problem can be utilised with the help of free web server for upload.

CHAPTER EIGHT: CONCLUSION AND RECOMMENDATIONS

8.1 Introduction

This chapter evaluates the project whether the intended objective have been achieved, give out conclusions and recommend the future work which can be done using this project.

8.2 Summary of the Results

Four modules have been developed during this project.

8.1.1 Page Analyser

The module has been implemented successfully. It has four classes and one interface. These are HtmlParser, AnalyserPage, ItemSet, Trail and HtmlPage respectively.

8.1.2 Data Extractor

The module has been implemented successfully and it consists of only java server pages. This is because the extraction process is assisted by system user for data scrubbing.

8.1.3 Data Storage

The module has been implemented successfully. It is composed of some Java classes and some JSP pages. It includes DataConnection class, the DataClass and Datatable class

8.1.4 Client user Interface

The module has been implemented successfully. It is composed of all the JSP pages which help the user to interact with the system to various functions. This is the main system interface. The user can get all the system services such as to extract data from web pages and query data from the database. The main interface includes the index.jsp, Data extraction service, Data query services and Page analysis

8.3 Generalization of the Results

This model can be used to any website range from unstructured, structured and semi-structured web pages. Also this model can be applicable to information extraction both internal data sources such as a local database which is the main focus of this project and external data sources such as remote collection of documents.

8.4 Applicability of the Results

The system which has been developed as prototype system can be used by any user and any application so as to produce the data in a structured format such as relation database or XML format. The prototype system can be used as a separate component (to be used by other systems) to extract data from any web pages.

The proposed techniques for data extraction can add value to the information retrieval system such as searching engine, RSS, online shopping system, price comparison system and monitoring system in the process of data extraction either by apply the technique used or use the data that will be extracted and stored in the database by using the prototype system. This directs user queries to appropriate servers by constraining the search space through query refinement and source selection. Therefore, it eliminates unnecessary communication overhead over the global networks and over the individual information sources

8.5 Conclusions

The growing bulks of information on the web data is a very valuable source of information. The web can now be viewed as a global and free data warehouse. But the web sources can not directly be interfaced with other applications for further processing. Many people have done the researches so that the web data can be available to the software applications. The main objective was to develop a wrapper which will extract data hence making them available to other applications. In this project tag-String as a structure for identifying the repetitive pattern has been presented. In so doing the data clusters can be identified and hence the data can be extracted from the result web page.

The main investigations done in this research are organised into stages. These stages are defined based on the regularity of the repetitive patterns of tag-String as they are found in various web domains.

- (i) The first stage is to investigate the web domain which produces the results in the way that, the structure of the data cluster is regular.
- (ii) The next stage is to investigate how HTML tags that have been used by some web domains. In particular, the tags which tend to produce irregular repetitive pattern on the tpGrid. These tags are bold tags for emphasising the search keywords. Examples of the web domains found with this pattern are www.yahoo.com, www.google.com, and www.ibm.com. Some of the words produced by search query are presented as subscript or superscript. Some site use links to mark some words as the data field such as title. So it is important for the page analyser to ignore these tags for better results.
- (iii) The other stage is to investigate the problems where the records of the data clusters have different number of data fields. In these cases other fields are optional, as they are not found on some other fields. The new approach is to find

the template record and use the template record to identify the missing fields and mark them as optional fields.

- (iv) The final investigation is on the clusters with nested data items. The proposed solution uses the information from the tpGrid so that the nested clusters can be sorted out. A good example of the site with this pattern is www.amazon.com. It has been found from searching information about the books that in some records there are nested data patterns. Moreover these nested patterns are optional field.

8.6 Recommendation for further Studies

The following are the further studies that can be originated by this project.

8.6.1 Challenges to address in some Web Domain

There is a lot to be done in relation to the data extraction from web sources. The main objective is to automate the task of extracting data. In some cases the web domain URL can be used to load the HTML source codes from the web servers and hence all the data produced as a result of the query can be extracted. The problem with some web domain is that, the result page URL is not available on the address line of the web browser. This is because many web domains use new technologies such as JSP and Active Server Pages, and PHP. Hence the following should be the extension of this project;

- (i) To investigate the production of the general browser which should be a wrapper to the search form of web sites with rich web information. In this case the task of getting the source code or using the URL will be solved since the source code will be available to the system automatically.
- (ii) To investigate the use of semantic techniques for scrubbing the data to their respective data types. This is because some of the data items are mixed with their labels. For example, "Author: Smith R. J." can be extracted. Therefore the author label needs to be separated from the author name.
- (iii) To investigate the use of information from the tpGrid to solve the problems of nested data items. Also, there are patterns where it is necessary to know in advance the structure of the record. In this case the information from tpGrid can be used. In most cases the difference of the consecutive items of the row in the tpGrid is the number of fields in the record.
- (iv) Data items which are stored on the HTML tag attributes. These data item are links to other web resources such as other web pages and images. The image tag stores the link of images in *src* attribute while the link tag to other web documents is

stored in *href* attribute. The images which are produced as a result of query result they normally have the same size. The width and height information of the image are found as attributes of image tag, which are *height* and *width*. The analyser will find the image tags and link tags in the tag-String which are part of the data clusters.

8.6.2 Challenges to address in Information Extraction

From data extraction process point of view, we know that, user is required in the information extraction process. A question is 'where should the person stop and the information extraction interface start?' Hence further studies should be conducted in order to clarify

- (i) The borders of the user involvement and system involvement in the information extraction process.
- (ii) Both how much activity and which type of activity the user should be able to direct the system to do at once.

8.6.3 Challenge to address in Data Storage or Warehouse

There should be the storage formats that allow interpretation of the content which favour any tools in order to improve the interoperability with other tools for information extraction. This will help to make the knowledge contained in the database sharable and herewith, available for re-use by other tools. Thus, future research could orientate towards:

- (i) How to store the content of extracted data in different format.
- (ii) How to make conversion of existing storage formats to other standard format such as XML.

8.6.4 Challenge to address in Information Retrieval

From the scope of this project where our system deals with data extraction only, with the assumption that user know exactly where to extract data by issue to the system the URL or web page source code of using search engine to locate the page. Hence further studies to extend scope can be:

- (i) Information retrieval from several sources

During this work, we have seen the search of information in several sources. But these searches were conducted separately and were independent from one another. Another field of study that could be addressed is the integration of results of searches from various

data sources so as to make the use of these results in a synergistic way, which will be of great benefit to the final user.

(ii) Information retrieval from unstructured data sources

Our results suggest that studies may be conducted towards finding 'how to improve the quality of information retrieval from unstructured data source using the same technique' because the outcomes were not satisfactory enough in our case especially for nested structure.

(iii) Defining the relevance of a returned result

Apart from recall and precision, other measurement metric should be used such as general data extraction tests and resilience against changing of web data since. The best way to define a relevance of a returned result remains an open question.

8.6.5 Application of the proposed Algorithm

The proposed algorithm can be applied to existing systems for data retrieval and extraction, since it is easy to implement and it has good performance. Therefore, the following are topics which can be implemented using this algorithm, just to mention few.

- (i) Using enterprise models as interface for information searching and extraction
- (ii) Designing and implementing a web-based data warehouse solution for cost analysis.
- (iii) Implementing best practices for fraud detection on an online advertising platform.
- (iv) Implementation and evaluation of a text extraction tool for adverse drug reaction information.
- (v) Extracting content from online news sites
- (vi) Detection of spyware by mining executable files
- (vii) Analysis of medical data
- (viii) Automated event extraction from e-mail.

8.6.6 Proposed Data extraction system.

For the purpose of improving the proposed data extraction system. especially to improve its functionality, reliability and usage, the following can be done

- Improve GUI for the system so as to provide nice usage to user.
- Retain the titles of extracted data clusters or allow user to label it.

REFERENCES AND BIBLIOGRAPHY

1. Baumgartner Robert, Wolfgang Gatterbauer and Georg Gottlob (2009) Web Data Extraction System.
2. Boronat Xavier Azagra (2008) A Comparison of HTML-aware tools for Web Data Extraction.
3. Califf Mary E, and Raymond J. Mooney (1999) Relational Learning of Pattern-Match Rules for Information Extraction.
4. Cosulschi Mirel et al (2006) HTML Pattern Generator - Automatic Data Extraction from Web Pages.
5. Degbelo Auriol and Tanguy Matongo (2009) Applying Enterprise Models As Interface For Information Searching.
6. Embley David W. (2005) Toward Tomorrow's Semantic Web, An Approach Based on Information Extraction Ontologies.
7. Gary Price and Chris Sherman (2001) The Invisible Web: Uncovering Information Source Search Engine Can't See.
8. Hackathom, Richard D. (1998) Web Farming for the Data Warehouse.
9. Jacob Ayubu (2005) A Warehouse Or Local Cache For Web Data.
10. Kuhlins S, and R Tredwell (2002) Toolkits for Generating Wrapper.
11. Lam Man I, Zhiguo Gong and Maybin Muyeba (2008) A Method for Web Information Extraction.
12. Miao Gengxin et al (2009) Extracting Data Records from the Web Using Tag Path Clustering
13. Michael K. Bergman (2001) The Deep We Surfacing Hidden Value.
14. Myllymaki J. (2001) Effective Web Data Extraction with Standard XML Technologies.
15. Narayan K.C.(2010) Analysis of Data Extraction Methods of Deep Web, A Dissertation On Analysis of Data Extraction Methods of Deep Web Submitted as a partial fulfillment of requirement of the degree of M. E. in Computer Engineering under Pokhara University.
16. Robinson Jerome (2004) Data Extraction from Web Data Sources.
17. Robinson Jerome (2004) Providing Robust Access to Data in Web Pages.
18. Shaker Mahmoud et al (2010) A Framework for Extracting Information from Semi-Structured Web Data Sources

URL References:

1. Agbogun, Joshua Babatunde (2010) The Deep Web As A Tool For Mathematical Science Education Research. A Paper Presented At National Mathematical Centre Capacity Building Workshop For Mathematical Sciences Lecturers In Tertiary Institutions [Online] available from www.nmcabuja.org/Lectures/the_deep_web.doc [29th May 2011]
2. Bright Planet. (2005). Deep Web FAQ [Online] available from <http://www.brightplanet.com> [29th May 2011]
3. Happy Harvester 2 (2011) <http://www.happyharvester.com/index.html> [29th May 2011]

APPENDICES

APPENDIX A: Data Extraction system installation and running guide

Introduction

This paper proposes a system which employs clustering techniques for automatic information extraction from HTML documents containing semi-structured data. Using domain-specific information provided by the user, the system parses and tokenizes the data from an HTML document, partitions it into clusters containing similar elements, and estimates an extraction rule based on the pattern of occurrence of html tags. The extraction rule is then used to refine clusters, and finally the output is reported. To demonstrate the effectiveness of this approach, the proposed approach is tested by conducting experiments on the University of Nairobi web-site and other websites range from education, business and government website; the results prove comparable to those reported in the literature.

Installation

- The system should be installed in Java runtime environment version 4 and above
- Copy the system folder and paste in a location of your choice
- Set the system environment include loading database library for connection purposes.

Data extraction

When the user is extracting the data from the web pages, the system provides the link to the function for data extraction. The user inputs the text string which is the HTML source code of the web page for data to be extracted. In some web sites when the data is not embedded into HTML code but JavaScript, it is necessary to enter the result web page URL. A good example is the web domain www.ebay.com. User may specify other settings for ignoring some HTML tags to get better results.

The system extract data from the HTML source code containing data. In this case to extract data from a page follow the following steps below.

1. Home

Start the process by starting at home. Click on the Link showing in the home page as shown on the diagram below, you may click either of the following link , Extract Data,

Query Data or Page Analysis.

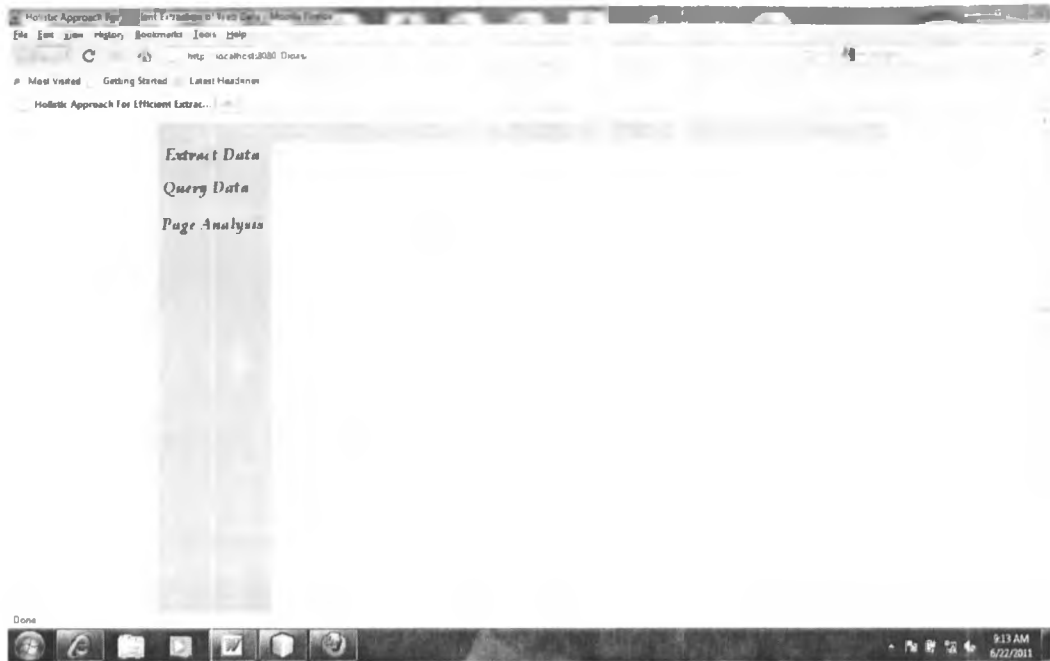


Figure 1

2. Getting HTML source code

Get the HTML source code from the result web page of your search. For example, to extract data from <http://www.uonbi.ac.ke/>

- Search the information of interest (Below is a typical result page) or following a link for static web page, until you found information of your interest.



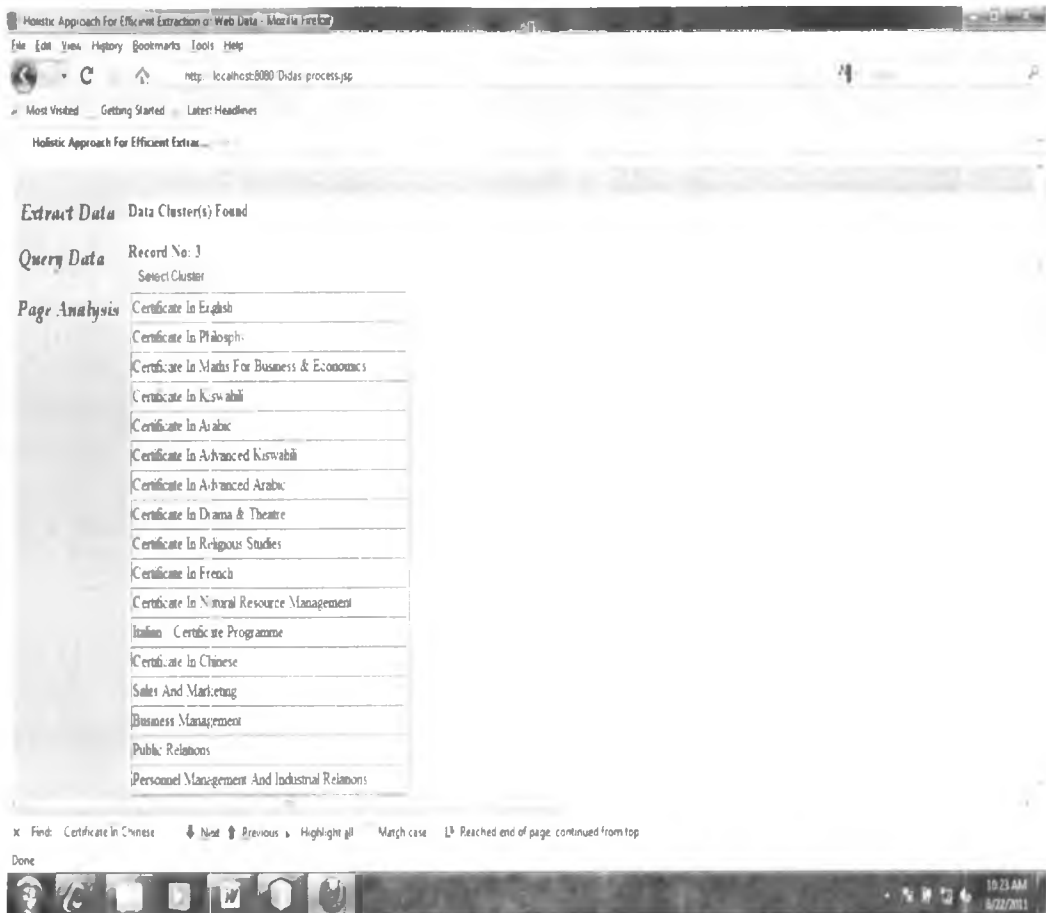
Figure 2

- Get HTML source code (view->source->), then copy to the clipboard so as to process it.

Example: From University of Nairobi website, we want to extract information about programme offered. Here we should follow the link so as to obtain exactly web page where we want to extract data.

http://www.uonbi.ac.ke/uon_programmes_type_index/certificate

Here the system will extract all certificate programme offered by UoN



Below is the procedure to obtain the above figure

3. Processing HTML source

- o Click on the Link (**Extract Data**), the page as shown below will be shown. Paste the HTML source code from the clipboard, and specify any settings for tags to be ignored by page analyser. Then click on the button (**Process Page**)

Enter the HTML Source Code Here

Tick tag to Ignore

- Bold tags
- Superscript <sup> tags
- Subscript <sub> tags
- Break
 tags
- Link <a>tags

Process Page

Clear Form

Use URL

Figure 3

- Example of data Clusters extracted (we have three clusters)

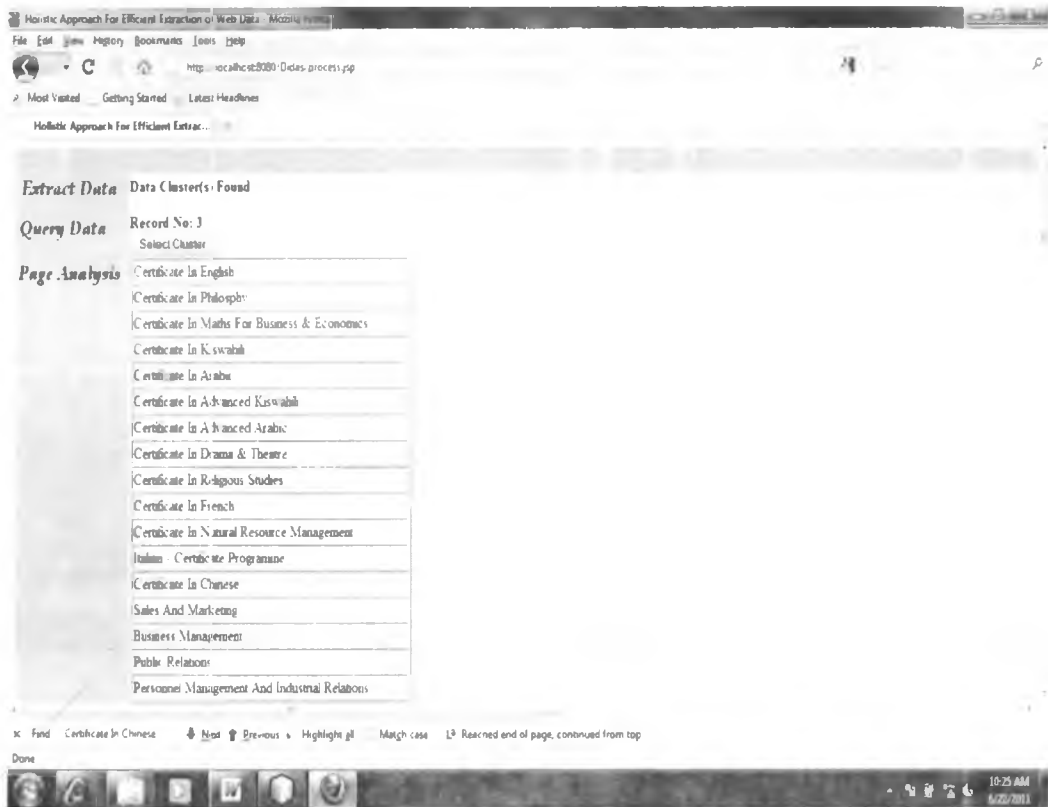


Figure 4

4. Selecting data Cluster

- Click on the button (**Select cluster**), one data cluster will be processed. The page as shown below will be shown. Click on the check box to select the columns of the data of interest (some columns will be empty).

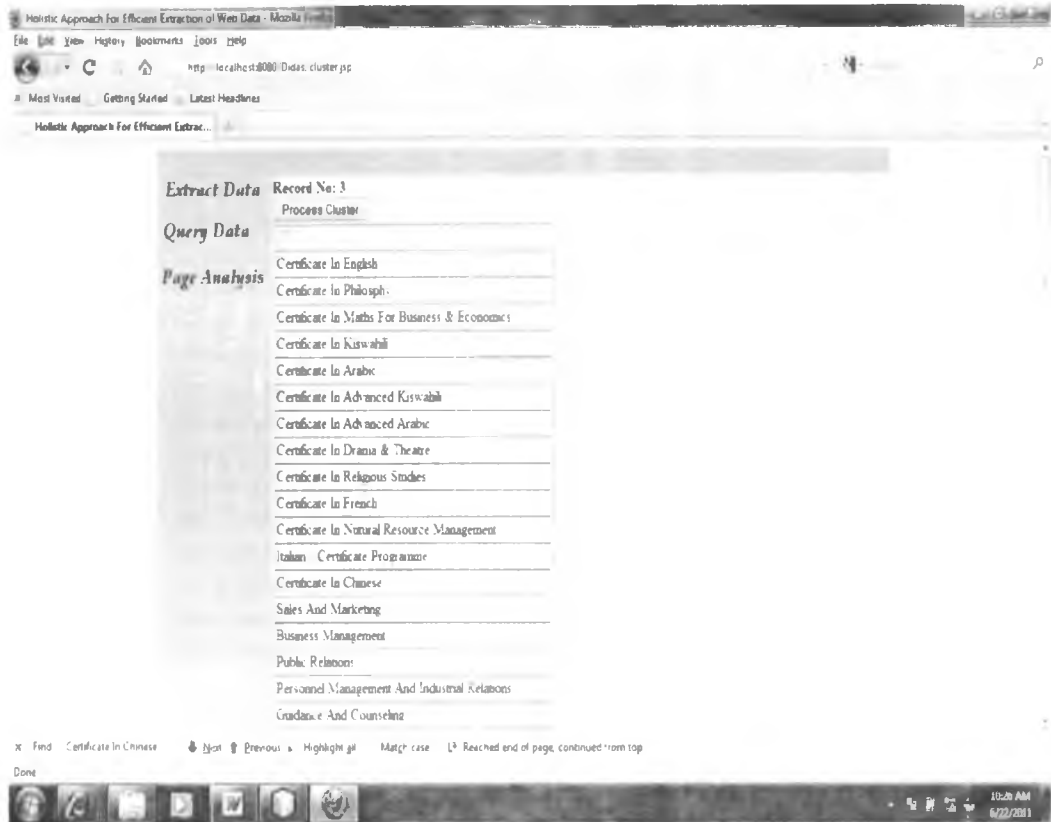


Figure 5

- Once the data cluster is refined, enter the information on the textbox as shown on the figure below. Then click on the button (**Save Data**) for the data to be stored into the database. The message will be shown to indicate that the information have been stored into the database.

Query Data Only For new Table
Enter site URL

Page Analysis Enter New Table Name
Save Data

Certificate In English
Certificate In Philosophy
Certificate In Maths For Business & Economics
Certificate In Kiswahili
Certificate In Arabic
Certificate In Advanced Kiswahili
Certificate In Advanced Arabic
Certificate In Drama & Theatre
Certificate In Religious Studies
Certificate In French
Certificate In Natural Resource Management
Italian - Certificate Programme
Certificate In Chinese
Sales And Marketing
Business Management
Public Relations

Figure 6

Data Query

This section, user can view the information that has been stored into the database.

Since the information is organised into database tables, the user selects the tables.

The tables can be queried by selecting the options

- i. All tables

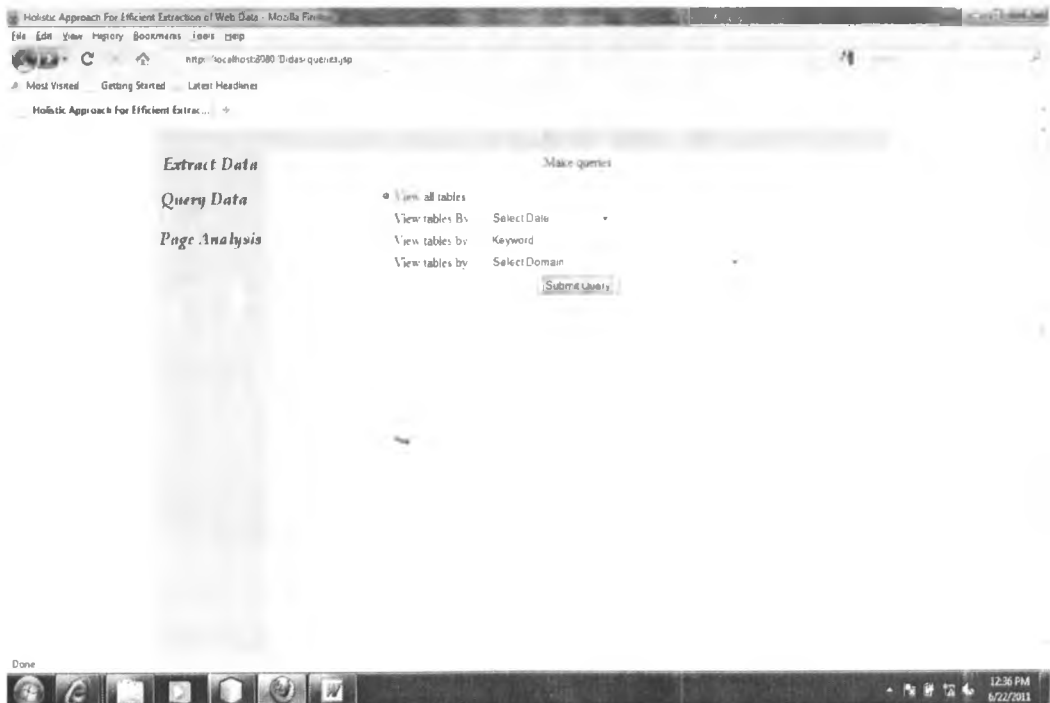


Figure 7

ii. Tables by date when the data was stored

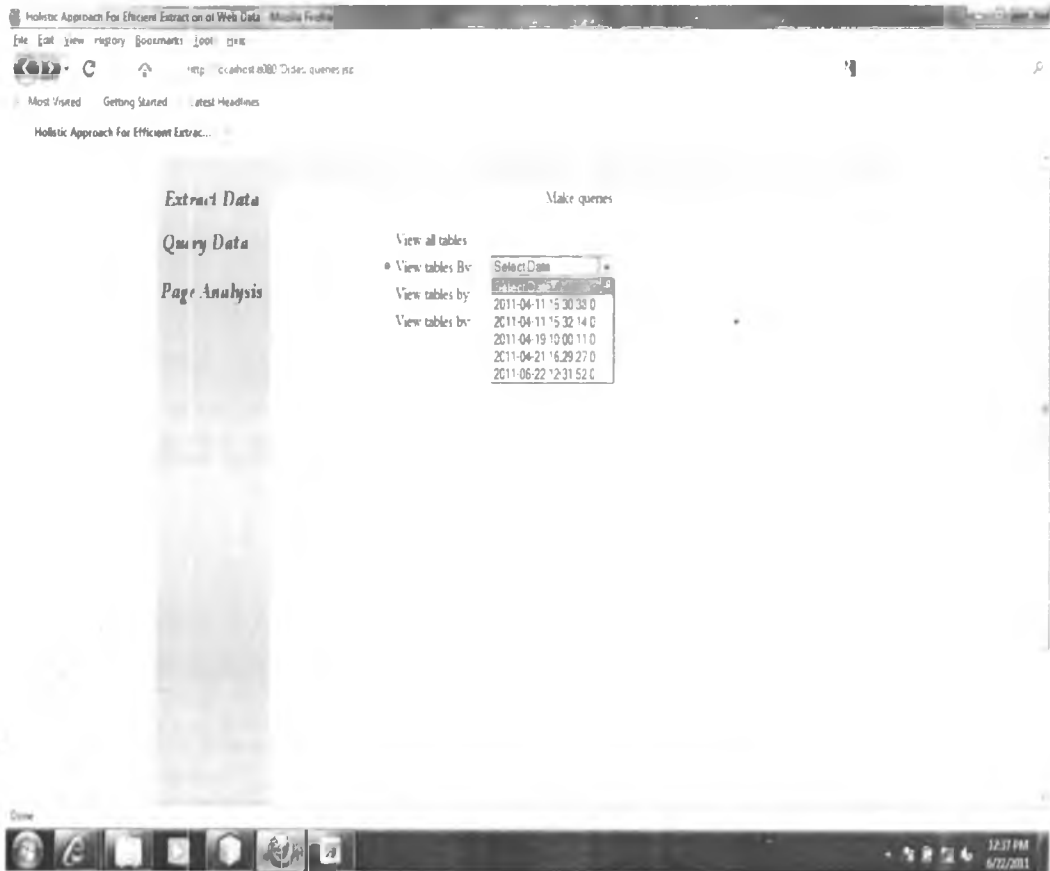


Figure 8

iii. Tables by keyword used for searching the data from the web sources

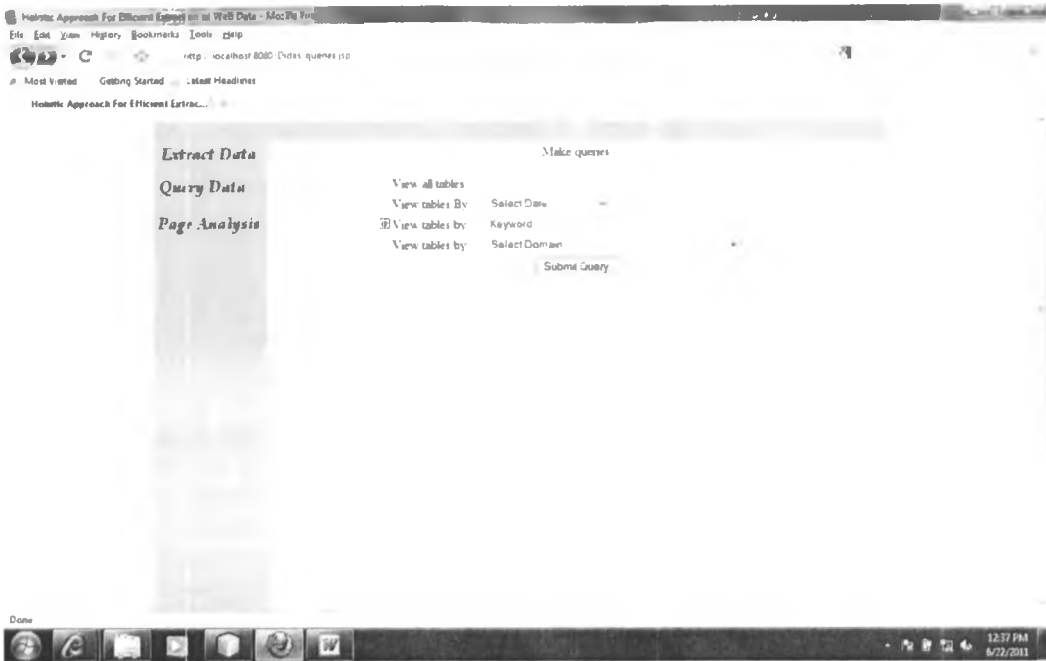


Figure 9

iv. Tables by web domains

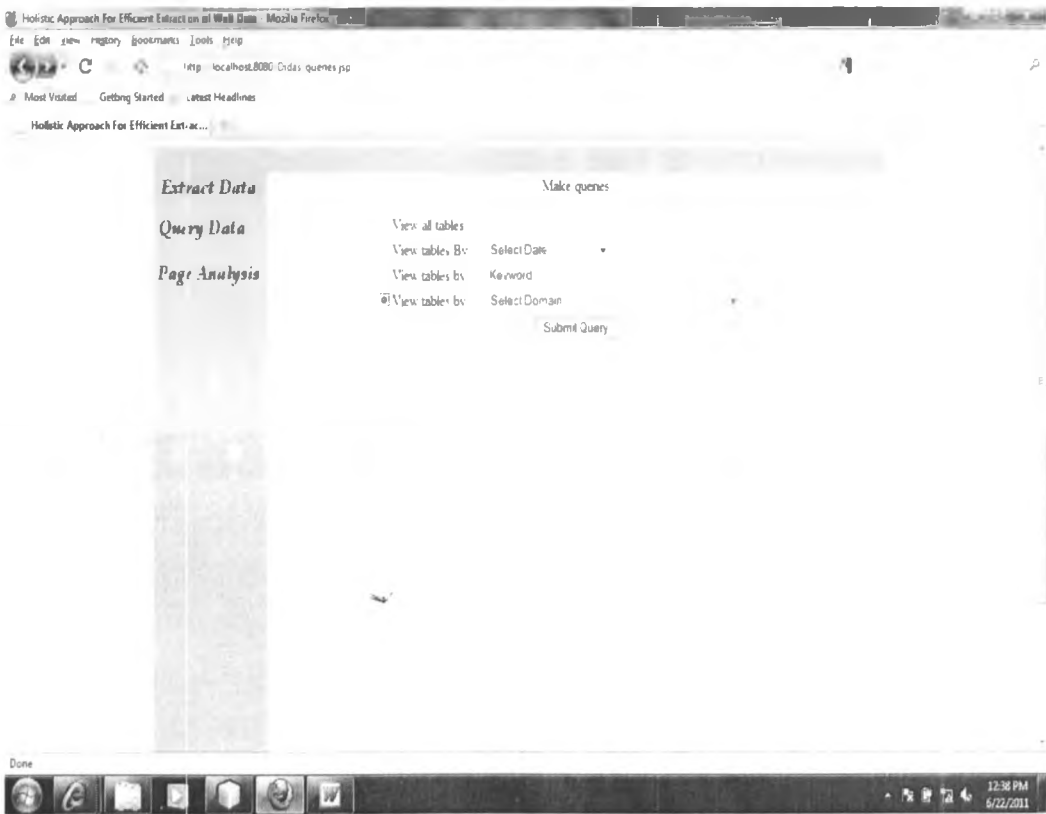


Figure 10

Select one option then click on the button labelled “submit query”. This will bring to the form which lists the name of database names.

Select the name of the table from the drop down combo box and optionally you can restrict the data to be shown by specifying the query keyword. See on the figure below.

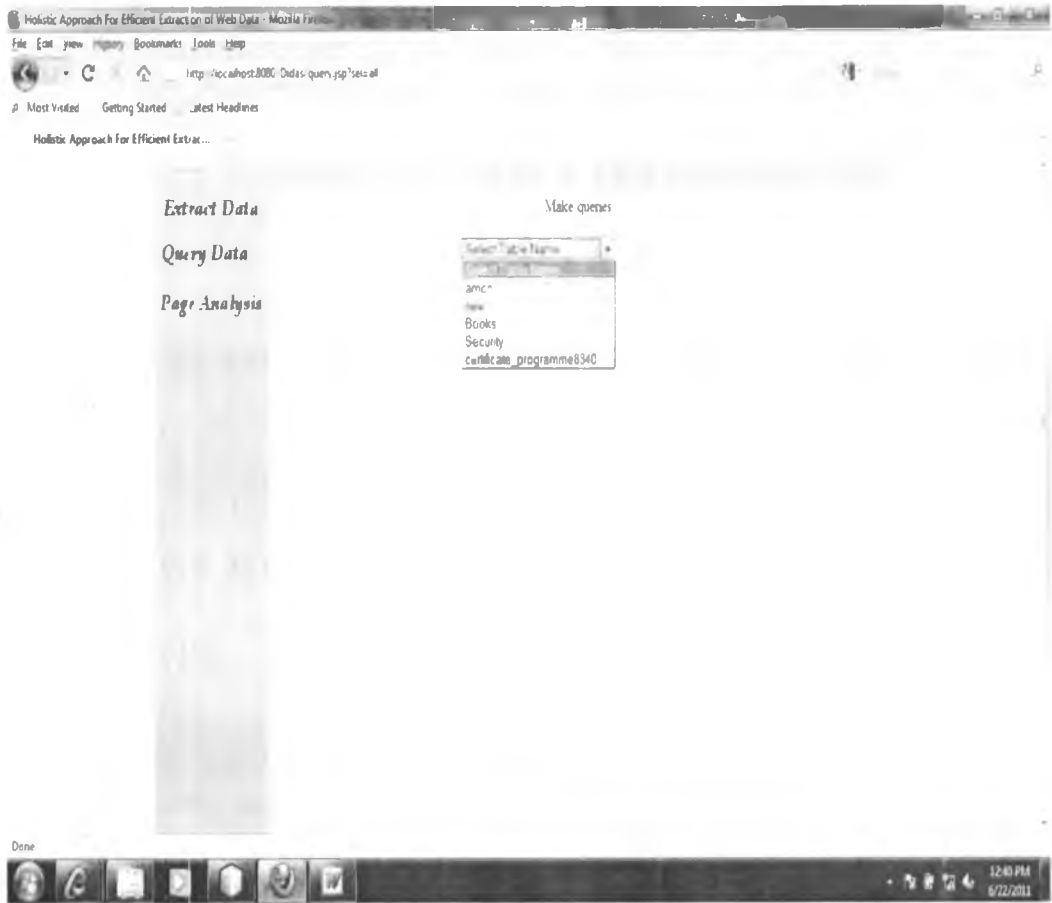


Figure 11

Below is the figure showing the results of the query.



Figure 12

Page Analysis

With this part you can see how the HTML page is modelled into

- Text-String
 - Tag-String
 - Tag-Set
 - Distinct Tag-Set
 - tpGrid
 - Semi refined tgGrid
 - Trial for clusters
1. Get the HTML source codes containing the searched data.
 2. Click on the link (**Page Analysis**) and Paste the HTML source code from the clipboard into the text area as shown in figure 3 above
 3. Specify any settings for HTML tags to be ignored by the analyser

4. Click the button (**Process page**), the new page will be shown. This page contains the combo box. Select the model for viewing and then click on the button (**Submit**)

Below is the typical output of the tag-Strings for the HTML result page from http://www.uonbi.ac.ke/uon_programmes_type_index/certificate web site

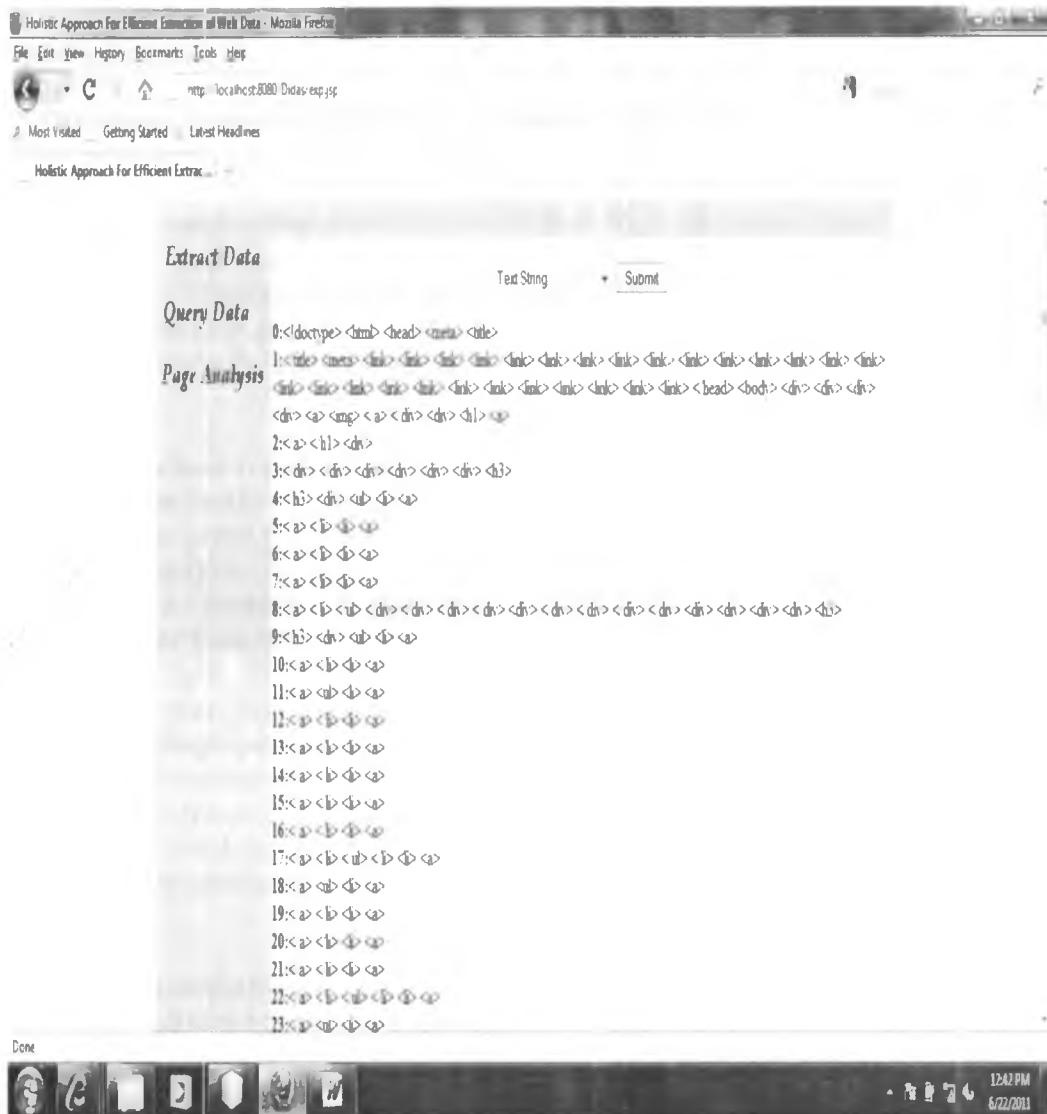


Figure 13

Source code for Page Analyser

```

package analyser;
import utilities.Utility;

import java.util.*;
import java.io.*;
import java.net.*;
/**
 * This class creates the HtmlPage as object and process
 * the information to collections of tagset, and trails.
 */
public class AnalysePage{
    protected Collection tagSetString;
    protected Collection distinctTagSetString;
    protected HtmlPage hp;
    public AnalysePage(){

/**
 * This method create HtmlPage object
 * @param html : String - Html source code or URL
 * @param settings : String - token of user options
 * @param type : String - (url or HTML source code)
 * @return : boolean - true of false
 * @throws Exception
 */
    public boolean analyse(String html,String settings, String type)throws Exception{
        List setting = new ArrayList();
        String htmlpage;
        String [] temp = settings.split(" ");
        for (int k=0;k<temp.length;k++){
            setting.add(temp[k]);
        }
        try{
            if (type.equals("url")){
                htmlpage = getContent(html);

                } else {
                    htmlpage=html;
                }
        }catch(Exception e){
            return false;
        }

        hp = new HtmlParser(htmlpage,setting);
        tagSetString=tagSets();
        distinctTagSetString=distinctTagSets();
        return true;
    }
}

```

```

}

/**
 * This method process the URL and return the HTML source
 * code
 * @param urlString : String - ur for web domain
 * @return - String the HTML source code
 */
private String getContent(String urlString) {
    String one="";
    try{
        URL url = new URL(urlString);
        URLConnection uc = url.openConnection();
        BufferedReader in = new BufferedReader(
            new InputStreamReader(
                uc.getInputStream()));
        String inputLine;
        String line="";
        String temp;
        while ((inputLine = in.readLine()) != null){
            temp = inputLine;
            line+=temp+"\n";
        }
        in.close();
        one=strip(line);
    }catch(Exception e){
        System.out.println(e.toString());
        return "";
    }
    return one;
}

/**
 * Method that strips all the HTML script tags,
 * style tags, and comments
 * @param temp :String - The HTML source code
 * @return
 */
private String strip(String temp){
    String tmp=temp;
    try{
        String reg="( ?i)<style(.*\n*)/style>";
        reg+="|<script(.*\n*)/script>";
        reg+="|<!-- (.*\n*) -->";
        reg+="|<!(.*)>";
        tmp=tmp.replaceAll(reg, "");
    }catch(Exception e){
        System.out.println(e.toString());
        return "";
    }
}

```



```

        return tmp;
    }

    /**
     * This method produced a Collection of tagSets
     * using the Collection of tagString
     * @return : Collection - of tag set string
     */
    public Collection tagSets(){
        Collection coll=new ArrayList();
        Collection temp= getTagString();
        Iterator iterator=temp.iterator();
        while(iterator.hasNext()){
            String tagString=(String)iterator.next();
            String tagset=tagSet(tagString);
            coll.add(tagset);
        }
        return coll;
    }

    /**
     * This method transforms the tagString
     * to tag-Set representation
     * @param tagString : String - tagString
     * @return : String - tag-Set string
     */
    private String tagSet(String tagString){
        String temp="";
        Collection dt = getDistinctTags();
        Iterator iter = dt.iterator();
        int i=0;
        while(iter.hasNext()){
            String aTag = (String)iter.next();
            if (i<dt.size()-1)
                temp+=(count(aTag,tagString))+", ";
            else
                temp+=(count(aTag,tagString));
            i++;
        }
        return temp;
    }

    /**
     * This methods calculated the number os times a tag
     * appears on the tagString
     * @param aTag : String - a tag
     * @param tagString : String - tag String
     * @return : int - number of times a tag is in tag String
     */
    private int count(String aTag,String tagString){
        if (tagString.indexOf(aTag)<0) return 0;
    }

```

```

int counter=0;
StringTokenizer st = new StringTokenizer(tagString," ");
while(st.hasMoreTokens()){
    String token = st.nextToken();
    if (token.equals(aTag)) counter++;
}
return counter;
}

/**
 * This method produces the Collection of distinct
 * tagset String
 * @return : Collection - of distinct tag set String.
 */
public Collection distinctTagSets(){
    Collection temp = new ArrayList();
    for (Iterator i=tagSetString.iterator();i.hasNext();){
        Object obj=i.next();
        if (!temp.contains(obj)){
            temp.add(obj);
        }
    }
    return temp;
}

/**
 * This method Gives a collection of text string
 * @return : Collection - text String
 */
public Collection getTxtString(){
    return hp.getTxtString();
}

/**
 * This method Gives a collection of tag string
 * @return : Collection - tag String
 */
public Collection getTagString(){
    return hp.getTagString();
}

/**
 * This method Gives a collection of distinct
 * tags
 * @return : Collection - distinct tags
 */
public Collection getDistinctTags(){
    return hp.getDistinctTags();
}

/**
 * This method Gives a collection of tags and text as they

```

```

* occur on the html source code
* @return : Collection - both text and tags
*/
public Collection getBoth(){
    return hp.getBoth();
}

/**
* This method return the transformed collection
* of tagset string as a trail of patterns.
* @return : Collection - trail.
*/
public Collection getTrails(){
    return Utility.series(tagSetString);
}
}
*****
package analyser;
import java.util.*;

/**
* This class is the interface that describes
* the services provided by a class implementing it
*/
public interface HtmlPage{
    /**
    * Collection of tag string
    * @return
    */
    public Collection getTagString();
    /**
    * Collection of distinct tags
    * @return
    */
    public Collection getDistinctTags();
    /**
    * Collection of text String
    * @return
    */
    public Collection getTxtString();
    /**
    * collection of both tag and text as they
    * occur in the html source code
    * @return
    */
    public Collection getBoth();
}

```

APPENDIX C: Glossary

An experiment	Means is a process or study that results in the collection of data.
Data Store	Means a data warehouse, local cache or database.
DBMS	Database Management System
Dynamic web page	Means the web page whose contents changes for each new request of the web page
Experimental design	Means is the process of planning a study to meet specified objectives.
HTML	Stands for HyperText Markup Language
JSP	Java Server pages
SQL	Structured Query Language
Static web page	Means that for any new request of the page the same information is being presented (surface web).
Tag-Set	A set of numbers for each Tag-String which shows how many times a particular HTML tag has been user with respect to all HTML tags used in the web page.
Tag-String	(In the HTML source code), is any String of characters which proceeds Text-String.
Text-String	Any string of characters which is not within the angle brackets < and > in the HTML source code.
UML	Unified Modelling Language
URL	Stands for Universal Resource Locator
XHTML	Extensible HyperText Markup Language
XML	eXtended Markup Language