UNIVERSITY OF ESSEX

Department of Computer Science

WAREHOUSE or LOCAL CACHE FOR WEB DATA

Jacob Ayubu

MSc Computer Science 2004 – 2005

SURNAME	JACOB
OTHER NAMES	AYUBU
QUALIFICATION SOUGHT	MSc COMPUTER SCIENCE
TITLE OF PROJECT	A Warehouse Or Local Cache For Web Data
SUPERVISOR	Dr. Nick Mitchell
DATE	September 2005

Abstract

The internet web is becoming important resource as a source of information. Various companies make the information from their databases available through search web page forms. The information ranges from indexed documents to product information. This information becomes more valuable if can be available to other software applications for further processing. In this project I present the efficient mechanism of extracting web data. The approach which I am using is based on analysing the patterns of the HTML tags. The customised general model of the web page is produced and is used for extracting data produced by subsequent web queries. The data extracted is populated to the database. The motivation behind this project is that the information is available to other software applications and hence can be used for decision making systems.

This project dissertation is in accordance with Examination Regulations 6.12 and 6.13.

Contents

Figures

ACKNOWLEDGMENT

I would like to appreciate the assistance I got from various people to make my project successful.

Dr. Nick Mitchell. my project supervisor for his excellence guidance and suggestions in various stages in this project.

Dr. Ing. Norbert Völker was very helpful during proposal writing for this project. He helped me a lot in the ideas and motivation in the project and how it should be planned.

Mr. J. Robinson helped me a lot in the research area.

CHAPTER 1

Introduction

1. 0 Overview

Human form of communication and information processing has become very important due to the popularity of conveying information through internet technology. There is massive growth of information on the web sources, which is available to users and application such as electronic commerce, news and notification of events.

The information on the web pages is presented by HyperText Markup Language (HTML) [42, 43, 47, 48] tags, mainly for presentation purposes and not the automated processing. This makes the information from the web pages not available for computer application processing. The HTML pages can be written by hand or generated by some HTML generator tools.

To make the information from the web sources available to the computer application, there is a need of extract them and presented into a suitable structured format. This is because the HTML pages are unstructured or semi structured documents.

Currently there is a development to the Internet technology so that web content should be presented in Extensible Markup Language (XML) [47, 48] format. This format is known as Extensible HyperText Markup Language (XHML) [47, 48]. In this case the web content will be separated from presentation, so that the information will be available for application processing. The information in XML format is more suitable for exchange with systems which are not compatible. Hence, applications such as monitoring systems, notification systems and price comparison systems will automatically access the information from the corresponding web sources. There are currently several technologies which uses XML technology for information exchange and processing. These technologies include Really Simple Syndication (RSS) [49] for news updates and Simple Object Access Protocol (SOAP) [47, 48] for information exchange and inter process communications.

The volume of web data from web sources in HTML technology is very big compare 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 end users or application programs.

1. 1 Project Motivation

There are various techniques used to extract the information from the web sources. Many researchers have been working on this research area which is found to be very important source for information. Some of these techniques (see section 1.4) include creating a predefined pattern which can be used to mark and hence extract data from the web pages. Along with these techniques other complex technologies such as artificial intelligence has been used to assist web data extraction process.

The result web pages produced by dynamic web pages in response to the user query have a distinct pattern, from which the data cluster can be easily identified. Those part of the HTML page that are mainly used for publication, advertisement or navigations are sorted out from the data resulted from the user search query.

In this project, I am going to explore the mechanism of identifying the cluster of data items from the web pages produced as a result of the search query (result web page). This mechanism is found to be most effective than other methods. This is because, it deals with the structures of the HTML tags that are used to represent the data resulted from the used search query.

The project investigates techniques not only for identifying the data clusters from the result web pages, but also for providing a mechanism for storing those extracted data to the relational databases. In this case, the information stored on the local cache can easily be used by end user or application software for further processing. The information can be rendered so that other presentation mechanisms can be produced depending on the application requirements such as displaying on mobile screens and /or small devices.

The information extracted can be processed inline with XML technology. In this case, the legacy HTML sources can be easily transformed into XML format. The information in XML form can be easily used exchangeable with other incompatible systems.

1.2 Goal of the Project

The goal of this project is to explore the mechanism of identifying the data clusters from the result web page and hence contains data that need to be extracted. The mechanism employed in this project is the use of analysis of the sequence of HTML tags which occurs before a non HTML tag of HTML source code (tag-String). These tag-Strings are used to identify the pattern which can be used to sort the part of the HTML page that corresponds to the results produced by the user query against those which are static and mainly used for navigation or labelling.

Further more, the web pages from different sites have different use of HTML tags for presenting the result of search query. This makes the process of identifying the data cluster more complicated. In this project, the mechanism is aimed to be more general in that it can be used to any web site, this is because many web sites have been explored and the system has been trained to learn these differences. This mechanism makes the system to be used to a large number of web sites.

After the identification of the result data locators from the analysis phase, the project goes further to explore the mechanism of storing the data to the relational databases. In this project, the best way for storing the information to the relational database is presented.

1.3 Project Tools and languages

The following project tools and languages have been used to make the project successful.

1.3.0 Java

The project is mainly build using Java Server Pages Technology [38]. Java language is used as programming language in JSP server scripts. There are java modules which are used together with JSP to model the middle tier of the application. These modules model the business logic of the application.

1.3.1 JavaScript

I have intended to use JavaScript in the JSP pages. for input validation and provide some processing as they have more string powerful processing functions than java

1.3.2 Smart draw

I have been using smart draw software for producing system models. These models are helpful for system development and project monitoring. For instance use case diagrams. component diagrams. User interface design and others for system interactions. Together with Smart draw I have used Visual Paradigm for UML. [50] which has user friendly tools for producing diagrams for system models.

1.3.4 Microsoft project

This software is very useful for project management process. I use this software to produce Gantt charts which shows the work break down structure of the project. The Gantt chart shows the tasks dependences and duration taken by the task. So. it is very easy to know if the project is successful or there are some delays and what to resolve those delays.

1.4 Related Works

There is a good number of researchers who have tried to find the best way of extracting web data and hence to be store into the data warehouses [4]. Most of the solutions provided, involves human intervention in the 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.

The process of extracting the data from the wed pages involves extracting the relevant data against HTML tags and those used for advertisements and navigations. Before data is extracted from the web page, the pattern is specified which located the relevant data items. This pattern is called wrapper for the web page. Many researchers have tried to develop wrappers for the web pages [1, 2, 3, 6, 10, 12, 24, 29, 30, 31]. Some of these wrappers are based on the predefined pattern [26, 27, 28] of HTML tags which encloses data [5, 13]. Some wrappers have been developed based on building schema that locates data [25] using the HTML table tags to define pattern structure while others have been developed using inductive techniques [10, 29] for automated processing.

Some researchers have used the technique for pattern matching. In their paper, "Relational Learning of Pattern-Match Rules for Information Extraction" Califf and Mooney [20] start by building the template for the data to be extracted and then the template is filled with data from the result web page. Other researchers have used machine learning techniques [19, 21]. The system is trained to learn the patterns which identify the data to be extracted. These methods are not automatic in the sense that they use manual systems.

Kuhlins and Tredwell [6] made researches on various toolkits for wrapper productions. They have described and implemented LAPIS toolkit. They treat the HTML web document as text document. To identify the data from the document they have used pattern recognition techniques based on the text constraints. The data is extracted from the document depending on the predefined pattern. The user will first create the pattern with the help of the system before the information is extracted. This is difficult to understand the toolkits and they involve a lot of manual work.

ANDES framework [7] has used the crawler technology and XML based data extraction techniques. The main motivation is that they restructure the HTML document to the structure defined by XML specification (XHTML) before the document is parsed for data extraction. Similar approach has been used by Robert Baumgartner, Sergio Flesca, and Georg Gottlob [14]. My approach will only consider parts of HTML document with data clusters.

Embley models the web data extraction based on the "extraction anthologies" [8]. In his work, he tried to model the extraction process based on the recognising and classifying data values from the web pages. This approach is limited to semi structured HTML web sources. The approach I am using can be applied to unstructured HTML web sources.

Richard D. Hackathorn [4] in his paper titled, "Web Farming for the Data Warehouse", described the process of getting the web data into the data warehouse. He described the process into four stages, which are discovery, acquisition, structuring and dissemination. During the stage of discovery, 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, being structured and store to the data warehouse. The process of identifying the location of data clusters is still manual system. In this case a more general and automatic approach is presented in this work.

The paper, presented by J. Robinson - *Data Extraction from Web Sources* describes the techniques of identifying the data clusters [1, 2, 3, 12]. The data clusters are identified automatically when the HTML source code is presented into tag-Set and text-String form. When the tag-Sets are represented with respect to distinct tag-Set they show a distinctive pattern of groups of tag-Set. These groups mark the regions which contain data clusters. Then the page descriptor is generated which describes the start and the end of the data cluster. The number of tag-Sets which form a unit distinctive pattern provides the size of the record of data cluster. This project do not only find the best way of identifying the data clusters, but also provides the best way that the page descriptor information should be presented. This is the case when the data clusters have records which have different size of records. The extension to this work is to store the information to the data warehouse. And so the information can be available to external users and software applications.

1.5 Overview of Dissertation

The dissertation has two main parts. The first part concerns with the theory about the project. It constitutes all the researches and investigations made during the project. The second part concerns with the software developments process that is system requirement specifications. software design, implementation and software testing.

Chapter two is about the overview of the investigations. The Structures and technologies used for presenting information on the web page have been discussed. Moreover, the over view for cluster identification using the tag-String techniques has been discussed.

Chapter three is about the investigations made in this project. The web pages which are produced as a result of the query results have different characteristics from different web page domains. These differences make the process of identifying the clusters on the web pages to vary for different web domains. The investigations have been done on the problem of structure of the tag-Set progressive Grid (tpGrid) which affects to identify automatically the clusters within the web page. Also the HTML tags which affect the structure of the records of the clusters such as HTML bold tag as used by some web domains for emphasising the search keywords has been investigated. Moreover in this chapter, the problem of clusters with optional fields has been investigated. Finally the problem of records with nested structures has been discussed together with techniques which are used to identify clusters in these situations.

Chapter four is about the technologies used during the project management, project investigations and system development. The justification on the choice of the technologies which can suitably fit with this project has been discussed in detail. Other technologies which can be used for system implementation are also discussed with the reasons as to why they have not been used in this system implementation.

Chapter five is about system analysis and software design. The requirements of the typical system for web data extraction system and data warehousing or local cache for the web data has been specified. Also the software design is also discussed in this chapter, where various design models and system architecture are presented.

Chapter six is about the detailed design and software implementation. In this chapter, the choice of the technologies for system implementation has been specified together with the reasons for the choices. Moreover, the setting for the working environment of the system has been discussed. This chapter also has specified the how various components of the system have been implemented.

Chapter seven is about testing where various level of testing to the system has been discussed. The levels of testing which includes unit testing, integration testing and system testing have been discussed. JUnit testing technique has been used for testing the methods using black box testing where the input data is tested against the output data. With integration testing, the components have been tested that they work together. Finally the system testing is done to check for the system behaviours which include performance and installation testing.

Chapter eight is the last chapter which is about the evaluation conclusion and future work. In this chapter the successes and failure of the system has been discussed. More has been discussed on the conclusion and what can be done as a future work as an extension to this project.

The dissertation ends with a list of references which has been used in this project together with the appendices which provides additional information of the dissertation. Appendix A is about web page analysis, appendix B is about investigations made, appendix C is about project management, appendix D is about system user manual, appendix E is about software source code, appendix F is about software system testing, and appendix G is about glossary.

CHAPTER 2

Web Page Analysis

2.0 Overview

The World Wide Web technology makes the information to be presented in the internet for various end users. The information seen on the internet is viewed through a special program know as web browsers. Examples of web browsers are Internet Explorer, Mozila and Netscape Navigator. There are special tags (delimiters) that are used to provide presentation of the information as seen on the browser. These tags are called HTML tags. The browser has the ability to process these HTML tags and produce the corresponding presentation of the information.

The web page contains the information to be presented on the browser together with the HTML tags which defines the way the information should be presented. There are tags that are used to present information as a list of item, tabular structure, and emphasis such as bold, italic, colour and so many other presentation styles.

The information presented on the web pages can be either static or dynamic. Static web page means that for any new request of the page the same information is being presented. The dynamic web page is the web page whose contents changes for each new request of the web page.

These contents of the dynamic pages is made to be changed from the fact that, there are server side scripts that are used to execute the user query and dynamically produce the information in response to the user query, either from database sources or from other data sources. Examples of server side scripts are JSP, PHP [51] and ASP. Also there is extension of HTML which makes use of the client side scripts and style sheets to change the content of the some parts of HTML page.

In this project I will discuss on the dynamic pages that are used to get the information from the underlying databases. Example of these sites range from online shopping applications; stock quotes figures, weather data processing to soccer information

agents. These sites provide the search form when the user can query the information from these sites.

The web page can be viewed as scripts which consist 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 in it which is used to provide more information about the HTML tag. This information is called attributes of the tag, which is that name value pair. Here is an example of the HTML tag with Attribute, `<font size="10">`. Font is that HTML tag which describes the behaviour of the texts enclosed by that tag. Size is the attribute which show that 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 currents browsers tend to ignore these qualities and the new extension of HTML which is XHTML enforces XML syntax.

The figure 1 and 2 below show part of the HTML source codes with the corresponding web page as seen on the browser.

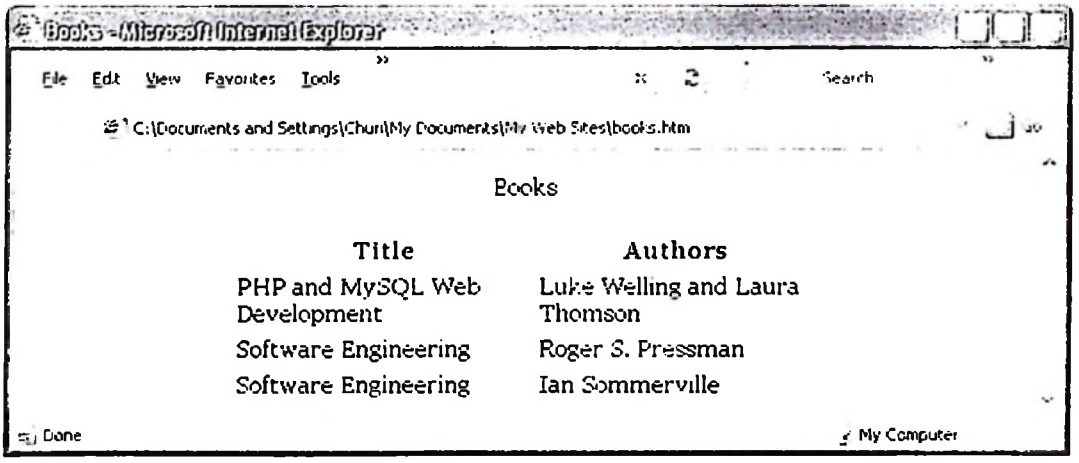


Figure 1 - The information as viewed on the browser

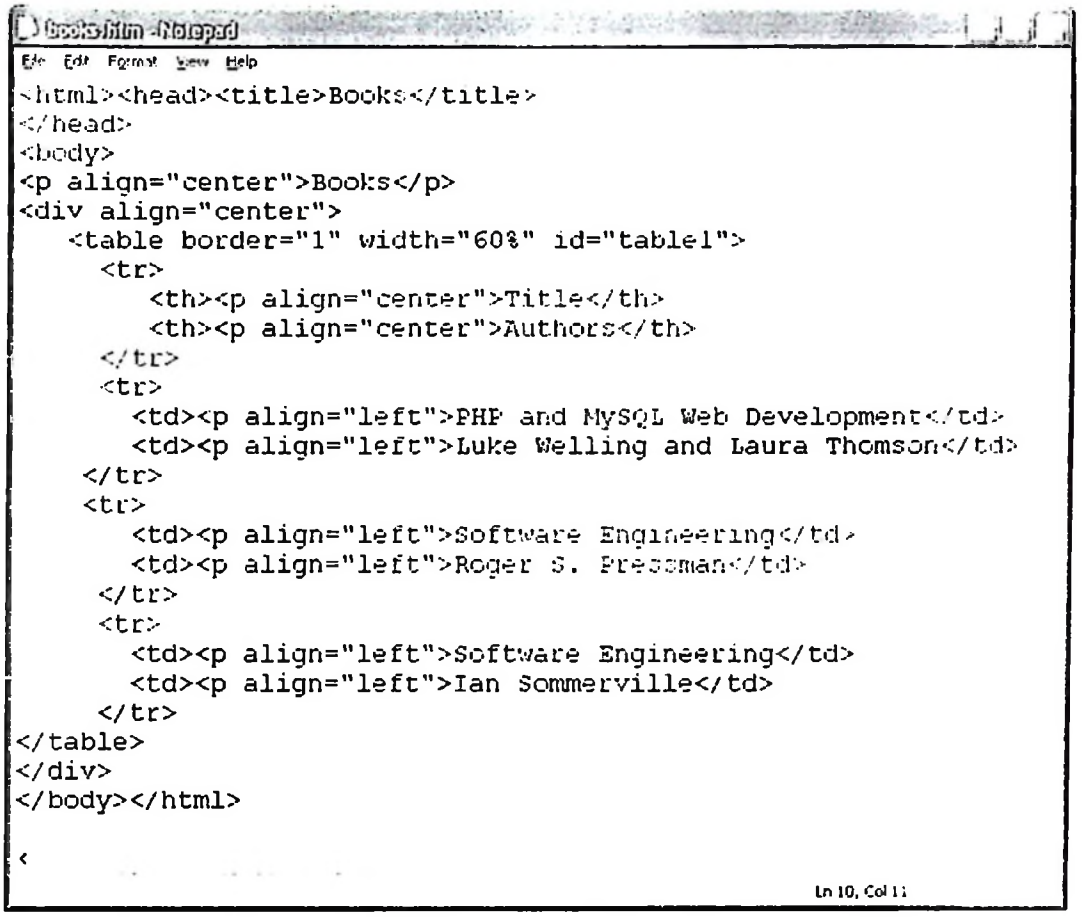


Figure 2 - The HTML source code

The investigations done in this project is to analyse the structure of the web pages by analysing the HTML source codes. The aim of the page analysis of the web pages 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 J [1]. From the figure 2 above, there is a very simple (to understand) sample of HTML source code, which generates the web page as seen on the browser as in figure 1.

The texts which are seen on the browser as seen on the figure 1 are *Books*, *Title*, *Authors*, *PHP and MySQL Web Development*, *Luke Welling and Laura Thomson*, *Software Engineering*, *Roger S. Pressman*, *Software Engineering*, and *Ian Sommerville*. The text marked as title (*Books*) is also seen on the browser on the title bar of the browser. Other texts which cannot be seen on the browser (figure 1) but are part of HTML source code (figure 2) are enclosed by angle bracket. These are HTML tags. It can be seen that before each of the text which is visible to the browser, there is a sequence of tags, which is called tag-String. The figure below shows the representation of the web page as a numbered sequence of tag-String followed by the text-String (text which is not inside the angle bracket).

S/No	Tag-String	Text-String
0	<html><head><title>	Books
1	</title></head><body>	Books
2	<table border="1" width="60%" id="table1"><tr><th>	Title
3	</th><th>	Authors
4	</th></tr><tr><td>	PHP and MySQL Web Development
5	</td><td>	Luke Welling and Laura Thomson
6	</td></tr><tr><td>	Software Engineering
7	</td><td>	Roger S. Pressman
8	</td></tr><tr><td>	Software Engineering
9	</td><td>	Ian Sommerville
10	</td></tr></table> </body></html>	

Figure 3 - tag-String and text-String

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.

There is something interesting about a list of tag-String as shown from figure 3 above. The tag-String number 4 (tag-String 4) is similar to that of number 6 and 8. Also tag-String 5 is similar to that of 7 and 9. This observation show that there is a data cluster associated with this pattern of repeated tag-String. This is what I will use to discover the data cluster and hence generate the page descriptor for that page.

2.1 Tag String and 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 not 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 will not be used during the analysis for the repetitive pattern. The figure 4 before shows a numbered sequence of the tag-String with no attributes.

S/No	Tag-String
0	<html><head><title>
1	</title></head><body>
2	<table><tr><th>
3	</th><th>
4	</th></tr><tr><td>
5	</td><td>
6	</th></tr><tr><td>
7	</td><td>
8	</th></tr><tr><td>
9	</td><td>
10	</td></tr></table> </body></html>

Figure 4 - Numbered sequence of tag-String without attributes

Having a numbered sequence of the tag-String, the list of numbers `[j, k]` can be created. The `j` item represents the number of the tag-String and `k` represents the number of the first occurred similar tag-String. The figure 5 below illustrates this observation with reference to the tag-String in figure 4.

S/No	Tag-String number
0	0
1	1
2	2
3	3
4	4
5	5
6	4
7	5
8	4
9	5
10	10

Figure 5 - Similarity Sequence Numbers for tag-String

From figure 5 above, with the second column, it can be seen that there is a repetitive pattern, in that a particular tag-String have been used more than once in the HTML code. The pattern of these tag-String forms a distinct structure which suggests a data cluster.

2.2 Tag Set

The tag-String is not used to provide the similarity 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 `<font`

size="10"><b><i> should be considered as the same as <b><font size=10><i>. 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. For example of a typical tag-String produced by the web site http://campus.acm.org of the result page produced by the search keyword data extraction was used is shown below.

```
</a> </td> </tr> </table> </td> </tr> <tr> <td> <hr>
<table> <tr> <td> <strong>
```

Figure 6 - A typical tag-String

It should be noted that, the tag name for instance td is different from that of /td or td/. The distinct tag names from this tag-String with its corresponding count are

```
/a - 1, /td - 2, /tr - 2, /table - 1, tr - 2, td - 2,
hr - 1, table - 1, strong - 1.
```

Figure 7 - a tag name with its number of counts

There are 31 distinct tag names used in this result web page, as shown below.

```
table, input, title, html, !doctype, form, /head, /title,
col, tr, /strong, td, div, a, /b, /div, img, /form, hr,
body, /body, /table, /html, /td, link, /tr, b, strong,
head, br, /a.
```

Figure 8 - A list of distinct tag names used on the web page

Now, the tag-Set is produced based on the arbitrary order of the distinct tag names. As an example, the tag-String in figure 6 above can be represented as tag-Set as shown on the figure below.

```
1,0,0,0,0,0,0,0,0,2,0,2,0,0,0,0,0,0,1,0,0,1,0,2,0,2,0,1,
0,0,1
```

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 have been used more that once. This means, there is repetitive structures based on the tag-String.
The result page produced by a search query on the `http://campus.acm.org` web site shows that there are 193 text-String. 194 tag-String and hence 194 tag-Sets. Since there are only 26 distinct tag-Sets, this shows that other tag-Sets have been used more than once.

2.3 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.

Refer to the Appendix A for the list of tag-Sets and a list of distinct tag-Sets as produced by the result web page web page of `http://campus.acm.org` when the search keyword *data extraction* was used. Below in figure 9 is the typical list of item-Sets which is a tag-Set progression Grid (tpGrid)[11].

The numbers with bold font are the serial number for the distinct tag-Sets. The number not bolded represents the serial number for the tag-String, and hence the serial number for the tag-Sets. From figure 9 it can be seen that, the tag-Set numbered as 0, 1, 4, 5, 6, 7, 8, 9, 16, 18, 43, 163, 188, 189, and 194 are used only once. The tag-Set 2 is used five times, tag-Set 9 is used four times and so the other item-Set with long rows are used several times.

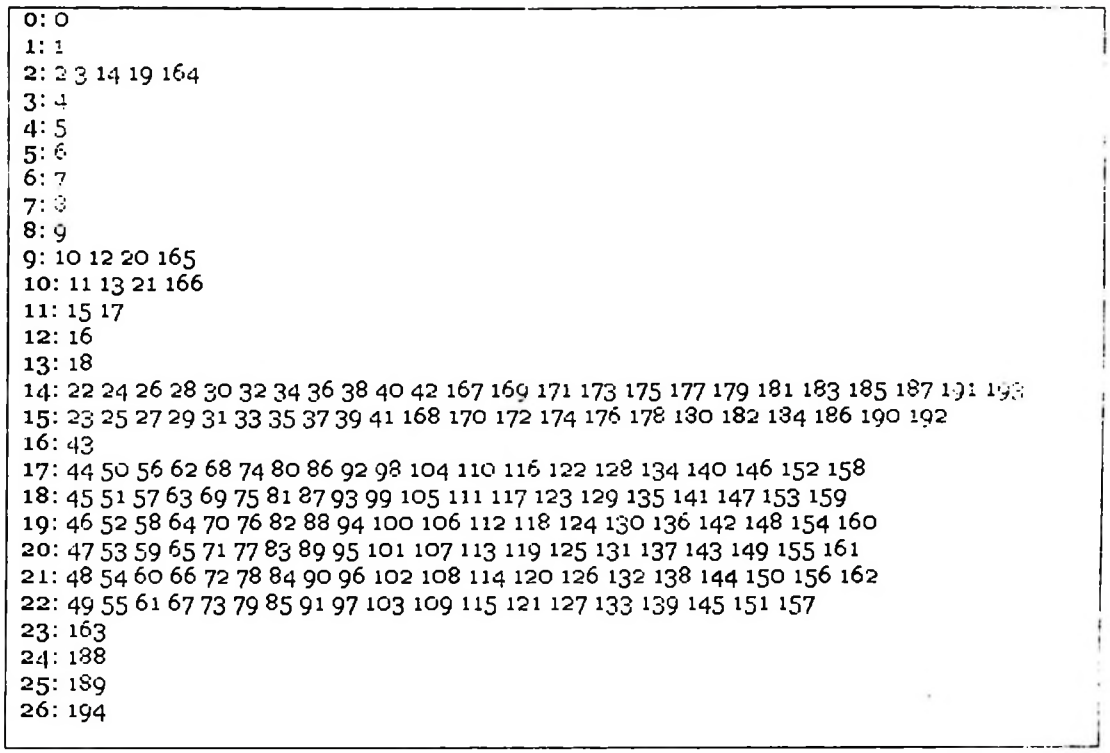


Figure 9 - The typical tpGrid showing the item-Set of Similar tag-Sets

The tpGrid in figure 9 above have structures which suggests that, there is existence of data clusters. If I follow the trail of tag-Set numbers starting at row number 17, it can be seen that, I have tag-String number 44, then 45, 46, 47, 48 and 49. These numbers have unique pattern similar to that of 50, 51, 52, 53, 54 and 55. They have similar appearance and hence this suggests that it is a record of 6 fields. The same pattern goes from left to the right along the rows 17, 18, 19, 20, 21 and 22.

Another cluster can be seen at rows with number 14 and 15, where the tag-Set number have a pattern of [22, 23] then [24, 25] up to the end of the rows. These items have similar appearance and they suggest a data cluster for a record with 2 fields. These rows have something unique compare to the other rows. The pattern goes with consecutive items, until item 42, then the pattern changes to [167, 168], [169, 170] until the end of the rows. This means that there are two data clusters in different places on the web page but have the similar tag-Set. These are page navigation links above and below the data cluster.

When the tgGrids are produced from different web pages, the data clusters have a unique feature. The shape of the data cluster as seen on the tpGrid is as shown in figure 10 below. If I look at row 22 it is found that the last entry missing. That is the number after 162, would be 163 which is not on row 22, but 23. The entry 163 is not part of the data cluster. If I look the tag-Set for the item 43 at row 16, it is found that it is different from those of the tag-Sets in the data cluster. This behaviour is the same for the first item of the data cluster.

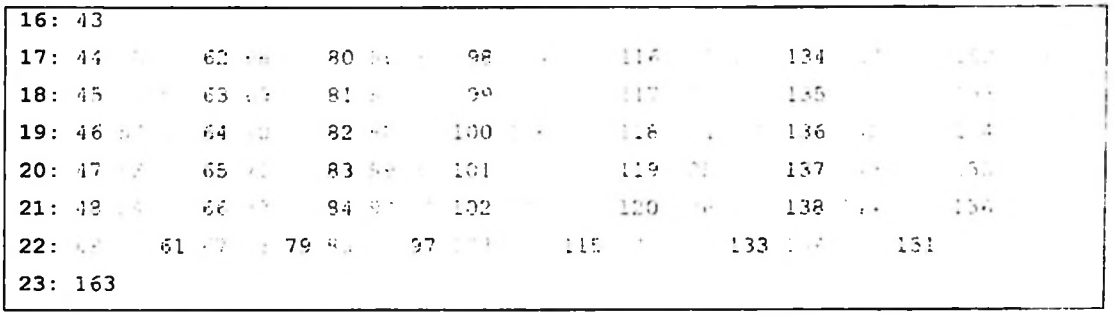


Figure 10 - Part of tpGrid showing data cluster

So, the data cluster starts at the item 43 up to 162. The record structure is constructed starting with 43 and group the items by 6 items. In this case the first record is [43, 44, 45, 46, 47, and 48], the next record is [49, 50, 51, 52, 53, and 54] and so on.

2.4 Data clusters

From the figure 9 above it is evident that there are three clusters. The main task is to identify which tag-Set number belongs to the data cluster. The analyser should discover those parts of tpGrid which are forms data cluster. The first task is to discover cycles when the trail of item number is followed. The technique used here is to generate the row succession graph (rsGraph) [3].

The row succession graph shows rows as nodes and the directed path showing a row following another row. The graph also shows the number of times a particular row follows another row. To construct the row succession graph, use the trail of item numbers. A typical model for the number representation of a graph is [17, 18,

20]. This means that row 17 follows row 18 20 times. The figure 11 below shows the complete row succession graph for the tpGrid from figure 9.

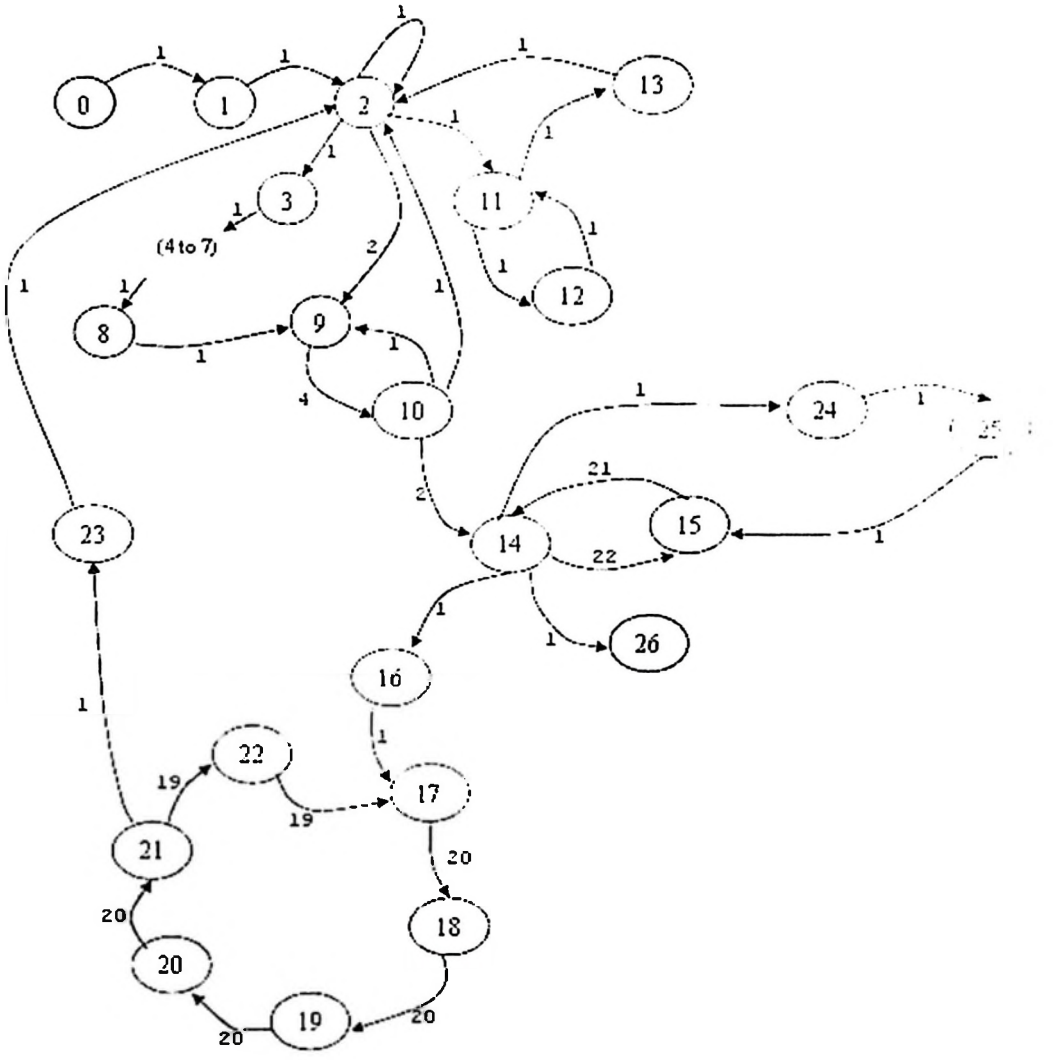


Figure 11 - Typical row succession graph

The row succession graph above shows two complete cycles, whose transition count is more than one. The number of nodes in a cycle is the size of the record for the data cluster. The nodes numbered 14 and 15 represent a cluster whose record size is two. The nodes 17, 18, 19, 20, 21 and 22 represent a data cluster whose record size is six.

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.

Previously I said that, 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. The analyser starts with node 16, then keeps on creating a record until six nodes are passed. The next record starts at node 22 (see figure 11 above).

2.5 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 have three main kind of information. There is 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 tpGrid. 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. If the size of the record is Z where Z in 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.

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 will be 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. Chapter 3 provides the investigations on the problems of locating the data clusters on the web page. The solution of how to tackle those problems has been discussed.

CHAPTER 3

Investigations on Web Page Analysis

3.0 Overview

This chapter presents the investigation on the natures of result web pages where in some cases, the structure of the data cluster not easily be 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.

3.1 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. A typical example of the page with distorted structure is from the web site http://www.mamma.com. When the search keyword "data extraction" is used, the part of typical tpGrid with data cluster, generated for page one is shown below.

```
35: 70
36: 71 77 83 89 95 101 107 113 119 125 131 137 143 149 155
37: 72 78 84 90 96 102 108 114 120 126 132 138 144 150 156
38: 73 79 85 91 97 103 109 115 121 127 133 139 145 151 157
39: 74 75 80 81 86 87 92 93 98 99 104 105 110 111 116 117 122 123 128 129 134 135 140 141 146 147 152
153 158 159
40: 76 82 88 94 100 106 112 118 124 130 136 142 148 154
41: 160
42: 176
```

The above figure shows that, the tag-Set 74 and 75 have the same tag-Set since they are on the same item-Set 39. The analyser restructures the tpGrid by splitting the row 39 to two rows. 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 will

form the items based on the consecutive values. From the above figure, the analyser reads 71, 72, 73, 74, and 76. 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, in this case is 75.

The item 75 is found at row 39, so that the analyser will split the row 39 to items that corresponds to 75. The figure below shows part of the tpGrid, which has been refined.

```
35: 70
36: 71 77 83 89  95 101 107 113 119 125 131 137 143 149 155
37: 72 78 84 90  96 102 108 114 120 126 132 138 144 150 156
38: 73 79 85 91  97 103 109 115 121 127 133 139 145 151 157
39: 74 80 86 92  98 104 110 116 122 128 134 140 146 152 158
39: 75 81 87 93  99 105 111 117 123 129 135 141 147 153 159
40: 76 82 88 94 100 106 112 118 124 130 136 142 148 154
40: 160
```

When this approach was used to the web page from web domain www.ibm.com it was found that, one of the rows of the data cluster was displaced to other rows. In doing so, it was very difficult to identify the rows which form data clusters. Below is a typical tpGrid from www.ibm.com when the search keyword "db2" was used.

```
31: 39
32: 40 43 60 121
33: 41 64 71 76 81 86 91 96 101 106 111 116 125
34: 42
35: 44
36: 45
37: 46
38: 47 48 49 50 51 52 53 54 55
39: 56
40: 57 118
41: 59 120
42: 65
43: 66
44: 67
45: 68 73 78 83 88 93 98 103 108 113
46: 69 74 79 84 89 94 99 104 109 114
47: 70 75 80 85 90 95 100 105 110 115
48: 72 77 82 87 92 97 102 107 112
49: 117
```

From the figure above, the data cluster is formed by rows 45, 46, 47, and 48. The items 68, 69, 70 and 72 are read. But these items are not in proper consecutive order. The item 71 is missing and is not part of the rows which form the data cluster. The item 71 has similar tag-Set with the items on the row 33, which has other items which are not part of the data cluster. It is important to refine the tpGrid so that, all the items which form the data cluster are grouped together. Below is the figure which shows the refined tpGrid so that all the similar clusters are grouped together.

```
51: 67
52: 68 73 78 83 88 93 98 103 108 113
53: 69 74 79 84 89 94 99 104 109 114
54: 70 75 80 85 90 95 100 105 110 115
55: 71 76 81 86 91 96 101 106 111 116
56: 72 77 82 87 92 97 102 107 112
57: 117
```

3.2 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 used tags like, link tags <a>, break
, paragraph <p>, lists <ul> and bold <b> or <Strong> 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.

When I was extracting the information from the web site www.ibm.com I was surprised to see that, I can not get the data clusters. But when I look on the result web page there is a clear data in clusters. When I tried to investigate the real cause of the irregular structure I found that, www.ibm.com web site used the bold HTML tag to put emphasis on the keywords used for searching see Appendix B.

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. Examples of these web sites are www.google.com. www.ibm.com.

When the bold tag was made to be ignored, the data cluster was clearly identified and the good results of data were extracted. Below is a typical part of tpGrid which shows the tag-Set numbers which marks the clusters of data.

```
44: 51
45: 52 57 62 67 72 77 82 87 92 97
46: 53 58 63 68 73 78 83 88 93 98
47: 54 59 64 69 74 79 84 89 94 99
48: 55 60 65 70 75 80 85 90 95 100
49: 56 61 66 71 76 81 86 91 96
50: 101
```

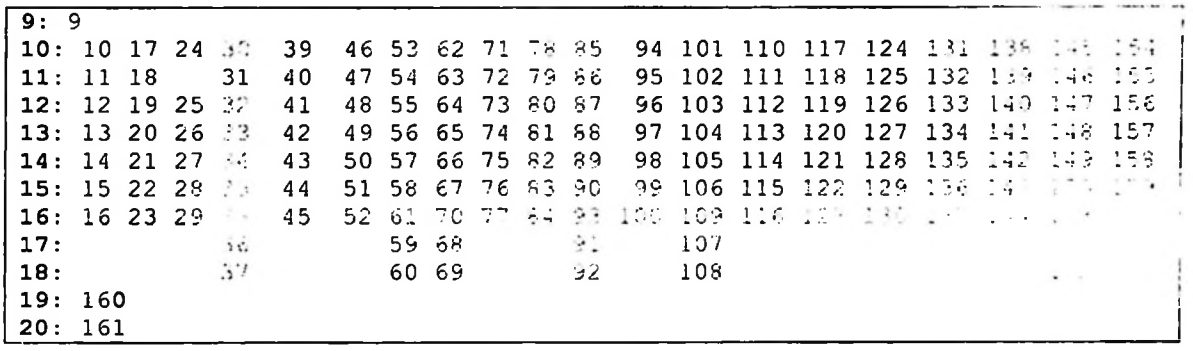
The other tag which needs to be ignored is subscript HTML tag <sub>. When I was analysing the results from the www.logitech.com, with "sdk" as a search keyword, it was found that subscript tag was used to present some of the information as subscript. Because I am using the tag-String for identifying the data fields, the data item within subscript was considered as a distinct data item. Since the occurrence of subscript tag is not uniform then the structure of the tpGrid was irregular. In order to solve this problem, the analyser ignores all the tags which are delimited by subscript tags. The result was found to be good when the subscript tag was ignored by the analyser. See the appendix B.

When I was analysing the web page from www.tesco.com, I found that the structure of the tpGrid was not so regular enough for the analyser to identify the data cluster. When I checked on the tag-Strings I found that the pattern was affected because of the image HTML tag <img> that was used in some records, and missing in some other records and hence makes the pattern for the data cluster to be irregular. When the HTML image tag was ignored by the page analyser the good results was produced.

It can be seen that, these HTML tags can not be ignored by analyser in all cases. Some websites used 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.

3.3 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. When I was analysing the web site from www.kelkoo.com I found that there are some tags that were occurring once or rarely in some of the data records. The trail formed when analysing the tpGrid shows that there is irregular structure for the analyser to identify the data cluster. Below is a typical tpGrid showing the results from www.kelkoo.co.uk web site, when the search keyword "database systems" was used.



Clearly from the figure above it can be seen that, the analyser located the data clusters at rows 10, 11, 12, 13, 14, 15 and 16. The trail breaks out when the record with items 24, 25, 26, 27, 28 and 28 is created. The item 25 was supposed to be on row 11.This shows that there is optional field that should be inserted on the hole as seen on the figure above. Note also that the items 36 and 37 should be inserted between row 15 and row 16. That is the record with items 30, 31, 32, 33, 34, 35, 36, 37 and 38 should be created. It was found that, with the use of the row succession graph it is not easy to sort out the pattern for the data clusters. Then the technique for pattern recognition was found to be very helpful for clusters identification.

In this approach, 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.

The figure below shows the typical trail for patterns that reveal the structure of the record. 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 in that is solves the problems of the optional fields by providing the more general record templates. Below is a typical trail of tag-Set from the web domain www.kelkoo.co.uk. which shows the tag-Set numbers of the result web page.

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 10,
11, 12, 13, 14, 15, 16, 10, 12, 13, 14, 15, 16, 10, 11, 12,
13, 14, 15, 36, 37, 16, 10, 11, 12, 13, 14, 15, 16, 10, 11,
12, 13, 14, 15, 16, 10, 11, 12, 13, 14, 15, 36, 37, 16, 10,
11, 12, 13, 14, 15, 36, 37, 16, 10, 11, 12, 13, 14, 15, 16,
10, 11, 12, 13, 14, 15, 16, 10, 11, 12, 13, 14, 15, 36, 37,
16, 10, 11, 12, 13, 14, 15, 16, 10, 11, 12, 13, 14, 15, 36,
37, 16, 10, 11, 12, 13, 14, 15, 16, 10, 11, 12, 13, 14, 15,
16, 10, 11, 12, 13, 14, 15, 16, 10, 11, 12, 13, 14, 15, 16,
10, 11, 12, 13, 14, 15, 16, 10, 11, 12, 13, 14, 15, 36, 37,
16, 10, 11, 12, 13, 14, 15, 160, 161, 162, 163, 163, 163, 163
3, 13, 169, 170, 171, 13, 3, 13, 3, 13, 3, 13, 3, 13, 181,
182, 183, 183, 183, 186, 183, 183, 183, 186, 183, 183, 183,
186, 183, 183, 183, 186, 183, 183, 183, 186, 183, 183, 183,
186, 183, 183, 183, 186, 183, 183, 183, 186, 183, 183, 183,
186, 183, 183, 183, 186, 183, 183, 183, 186, 183, 183, 183,
186, 183, 183, 183, 186, 183, 183, 183, 186, 183, 183, 183,
186, 183, 183, 183, 186, 183, 183, 183, 186, 183, 183, 183,
186, 183, 183, 281, 282, 3, 284, 3, 286, 3, 286, 3, 286, 3,
286, 3, 286, 3, 296, 162, 3, 13, 3, 13, 3, 13, 3, 13, 3, 13,
3, 13, 3, 13, 3, 13, 3, 13, 316, 317]
```

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.

```
[-1, -1, -1, 3, -1, -1, -1, -1, -1, -1, 10, 11, 12, 13, 14,
15, 16, 10, 11, 12, 13, 14, 15, 16, 10, 12, 13, 14, 15, 16,
10, 11, 12, 13, 14, 15, 36, 37, 16, 10, 11, 12, 13, 14, 15,
16, 10, 11, 12, 13, 14, 15, 16, 10, 11, 12, 13, 14, 15, 36,
37, 16, 10, 11, 12, 13, 14, 15, 36, 37, 16, 10, 11, 12, 13,
14, 15, 16, 10, 11, 12, 13, 14, 15, 16, 10, 11, 12, 13, 14,
15, 36, 37, 16, 10, 11, 12, 13, 14, 15, 16, 10, 11, 12, 13,
14, 15, 36, 37, 16, 10, 11, 12, 13, 14, 15, 16, 10, 11, 12,
13, 14, 15, 16, 10, 11, 12, 13, 14, 15, 16, 10, 11, 12, 13,
14, 15, 16, 10, 11, 12, 13, 14, 15, 16, 10, 11, 12, 13, 14,
15, 36, 37, 16, 10, 11, 12, 13, 14, 15, -1, -1, 162, 163, 163,
163, 163, 3, 13, -1, -1, -1, 13, 3, 13, 3, 13, 3, 13, 3, 13, -
1, -1, 183, 183, 183, 186, 183, 183, 183, 186, 183, 183, 183,
186, 183, 183, 183, 186, 183, 183, 183, 186, 183, 183, 183,
186, 183, 183, 183, 186, 183, 183, 183, 186, 183, 183, 183,
186, 183, 183, 183, 186, 183, 183, 183, 186, 183, 183, 183,
186, 183, 183, 183, 186, 183, 183, 183, 186, 183, 183, 183,
186, 183, 183, 183, 186, 183, 183, 183, 186, 183, 183, 183,
186, 183, 183, 183, 186, 183, 183, 183, 186, 183, 183, 183,
186, 183, 183, -1, -1, 3, -1, 3, 286, 3, 286, 3, 286, 3, 286,
3, 286, 3, -1, 162, 3, 13, 3, 13, 3, 13, 3, 13, 3, 13,
3, 13, 3, 13, 3, 13, -1, -1]
```

From the figure above. I explain the pattern on the clusters with the items shown on the figure below.

```
[10, 11, 12, 13, 14, 15, 16, 10, 11, 12, 13, 14, 15, 16, 10,
12, 13, 14, 15, 16, 10, 11, 12, 13, 14, 15, 36, 37, 16, 10,
11, 12, 13, 14, 15, 16, 10, 11, 12, 13, 14, 15, 16, 10, 11,
12, 13, 14, 15, 36, 37, 16, 10, 11, 12, 13, 14, 15, 36, 37,
16, 10, 11, 12, 13, 14, 15, 16, 10, 11, 12, 13, 14, 15, 16,
10, 11, 12, 13, 14, 15, 36, 37, 16, 10, 11, 12, 13, 14, 15,
16, 10, 11, 12, 13, 14, 15, 36, 37, 16, 10, 11, 12, 13, 14,
15, 16, 10, 11, 12, 13, 14, 15, 16, 10, 11, 12, 13, 14, 15,
16, 10, 11, 12, 13, 14, 15, 16, 10, 11, 12, 13, 14, 15, 16,
10, 11, 12, 13, 14, 15, 36, 37, 16, 10, 11, 12, 13, 14, 15]
```

The simplest pattern recognition approach is that. the first entry for each record occurs at a certain internal within the items in the cluster. When the item 10 is used as a control it can be seen that, the pattern for the record structure is created are shown below.

```
1.  [10, 11, 12, 13, 14, 15, 16]
2.  [10, 11, 12, 13, 14, 15, 16]
3.  [10, 12, 13, 14, 15, 16]
4.  [10, 11, 12, 13, 14, 15, 36, 37, 16]
5.  [10, 11, 12, 13, 14, 15, 16]
6.  [10, 11, 12, 13, 14, 15, 16]
7.  [10, 11, 12, 13, 14, 15, 36, 37, 16]
8.  [10, 11, 12, 13, 14, 15, 36, 37, 16]
9.  [10, 11, 12, 13, 14, 15, 16]
10. [10, 11, 12, 13, 14, 15, 16]
11. [10, 11, 12, 13, 14, 15, 36, 37, 16]
12. [10, 11, 12, 13, 14, 15, 16]
13. [10, 11, 12, 13, 14, 15, 36, 37, 16]
14. [10, 11, 12, 13, 14, 15, 16]
15. [10, 11, 12, 13, 14, 15, 16]
16. [10, 11, 12, 13, 14, 15, 16]
17. [10, 11, 12, 13, 14, 15, 16]
18. [10, 11, 12, 13, 14, 15, 16]
19. [10, 11, 12, 13, 14, 15, 36, 37, 16]
20. [10, 11, 12, 13, 14, 15]
```

The figure above shows a data cluster with 20 records. But the number of fields in each record is not the same. This is because some of the field entries are option fields. The analyser creates the template record for the structure of the data cluster. The template record for the data cluster is the one which have the highest number of items. In this case, the template record with items [10, 11, 12, 13, 14, 15, 36, 37, and 16] is chosen. 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. When the items of the first record are compared with the template record, it can be seen that, the items 36 and 37 and missing. This means that these entries in the items of the first records are option items. So, the structure of the items of the first record is [10, 11, 12, 13, 14,

15, -1, -1, and 16]. Below is a complete refined cluster of data item and hence can be used by page analyser to be transformed to the corresponding tag-Set numbers.

```
1.  [10, 11, 12, 13, 14, 15, -1, -1, 16]
2.  [10, 11, 12, 13, 14, 15, -1, -1, 16]
3.  [10, -1, 12, 13, 14, 15, -1, -1, 16]
4.  [10, 11, 12, 13, 14, 15, 36, 37, 16]
5.  [10, 11, 12, 13, 14, 15, -1, -1, 16]
6.  [10, 11, 12, 13, 14, 15, -1, -1, 16]
7.  [10, 11, 12, 13, 14, 15, 36, 37, 16]
8.  [10, 11, 12, 13, 14, 15, 36, 37, 16]
9.  [10, 11, 12, 13, 14, 15, -1, -1, 16]
10. [10, 11, 12, 13, 14, 15, -1, -1, 16]
11. [10, 11, 12, 13, 14, 15, 36, 37, 16]
12. [10, 11, 12, 13, 14, 15, -1, -1, 16]
13. [10, 11, 12, 13, 14, 15, 36, 37, 16]
14. [10, 11, 12, 13, 14, 15, -1, -1, 16]
15. [10, 11, 12, 13, 14, 15, -1, -1, 16]
16. [10, 11, 12, 13, 14, 15, -1, -1, 16]
17. [10, 11, 12, 13, 14, 15, -1, -1, 16]
18. [10, 11, 12, 13, 14, 15, -1, -1, 16]
19. [10, 11, 12, 13, 14, 15, 36, 37, 16]
20. [10, 11, 12, 13, 14, 15, -1, -1, -1]
```

It should be noted that, the last entry of the items of the last record does not mean that it is 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 (see figure 10 in section 2.3).

3.4 Nested data clusters

The analysis of the web page from the web site domain www.amazon.com, showed that, the analyser was able to identify data clusters and hence extract data from those pages. But in the data extracted it was found that, some of the data fields were located on the wrong columns. The figure below shows the data extracted from the web domain http://www.amazon.com. It can be seen that the data item "$99.00" has been extracted to the wrong data field, as it was supposed to be in the last column.

1.	Database Systems: Design, Implementation, and Management, Sixth Edition	- January [illegible]	Avg. Customer Rating:	List Price:	$[illegible]
2.	Database Systems Concepts	- May 17, 2005)	Avg. Customer Rating:	List Price:	$[illegible]
3.	Fundamentals of Database Systems, Fourth Edition	- July 23, 2003)	Avg. Customer Rating:	List Price:	$107.40
4.	Database Systems : A Practical Approach to Design, Implementation and Management (4th Edition) (International Computer Science Series)	- May 27, 2004)	Avg. Customer Rating:	$99.00	optional
5.	Database Systems : An Application-Oriented Approach, Introductory Version (2nd Edition)	- March 30, 2004)	optional		Buy new
6.	A First Course in Database Systems (2nd Edition)	- October 2, 2001)	Avg. Customer Rating:	$84.00	:
7.	Database Systems: The Complete Book	- October 2, 2001)	optional		Buy new
8.	Designing Effective Database Systems	- January 10, 2005)	Avg. Customer Rating:	:	$[illegible]
9.	An Introduction to Database Systems, Eighth Edition	- July 22, 2003)	optional		Buy new
10.	Database Management Systems	- August 14, 2002)	Avg. Customer Rating:	List Price:	$[illegible]

Then I realised that there is something more to be done for pattern recognition for generalised record structures. This happens when there is a structure of nested records. Using the approach as described on the section 3.4 above, some items were not found on the chosen template record but existed in some other records.

The typical result is shown on the figure below, for one of the data clusters which had 10 records.

1. [129, 117, 131, 132, 133, 134, 135, 136, 137, 138, 139, 18, 137, 142, 143, 18, 137, 137, 147, 148, 149, 150]
2. [129, 117, 131, 132, 133, 134, **157**, 148, 149, 150]
3. [129, **162**, 132, 133, 134, **166**, 137, 138, 139, 18, 137, 142, **173**, 148, 149, 150]
4. [129, 117, 131, 132, 133, 134, **157**, 148, 149, 150]
5. [129, **162**, 132, 133, 134, 157, 148, 149, 150]
6. [129, 117, 131, 132, 133, 134, **166**, 137, 138, 139, 18, 137, 142, **173**, 148, 149, 150]
7. [129, 117, 131, 132, 133, 134, 135, 136, 137, 138, 139, 18, 137, 142, 143, 18, 137, 137, 147, 148, 149, 150]

8. [129, 117, 131, 132, 133, 134, **157**, 148, 149, 150]
9. [129, 117, 131, 132, 133, 134, 135, 136, 137, 138, 139, 18, 137, 142, 143, 18, 137, 137, 147, 148, 149, 150]
10. [129, 117, 131, 132, 133, 134, **157**, 148, 149]

From here we get the template record as [129, 117, 131, 132, 133, 134, 135, 136, 137, 138, 139, 18, 137, 142, 143, 18, 137, 137, 147, 148, 149, 150]. When the first record is compared with the template record there is no problem since all the items are already aligned. But when the second record is compared with the template record it can be seen that, the item 157 is not on the template record.

The tag-String for the item 157 is [
  <table> <tr> <td> <table> </table> </td> <td> <a>]. The tag-String for the items [135, 136, 137, 138, 139, 18, 137, 142, 143, 18, 137, 137 and 147] is shown below. The main aim is to investigate the relationship of these items and item 157.

```
135:<br> </span> <span>
136:</span> <table> <tr> <td> <table> <tr> <td>
137:</td> <td>
138:</td> <td> <span>
139:</span> </td> </tr> <tr> <td> <a>
140:</a>
141:</td> <td>
142:</td> <td> <b>
143:</b> </td> </tr> <tr> <td> <a>
144:</a>
145:</td> <td>
146:</td> <td>
147:</td> </tr> </table> </td> <td> <a> <span>
```

The tag-String 157 shows the presence of a table that has no contents. The tag-String 136 shows that the table structure that start <table> and go through to the tag-

String 147 when it closes by the tag </table>. This observation means that the item 157 is exact substitute for the items from 135 through 147. This is a nested optional structure.

From these observations, the item 162 replaces the items 117 and 131 of the template record. The item 166 replaces the items 135 and 136, and item 173 replaces the items 143, 18, 137, 137, 147 of the template record. For template matching the items for record 2 can be aligned as shown below.

```
[129, 117, 131, 132, 133, 134, 135, 136, 137, 138, 139, 18, 137,
142, 143, 18, 137, 137, 147, 148, 149, 150]
[129, 117, 131, 132, 133, 134,  -1,  -1,  -1,  -1,  -1,  -1,  -1,
-1,  -1,  -1,  -1,  -1, 157, 148, 149, 150]
[129, -1, 162, 132, 133, 134,  -1, 166, 137, 138, 139, 18, 137,
142, -1,  -1,  -1,  -1,  173, 148, 149, 150]
```

This part of the suggested solution is not implemented in this project due to the limitation of time. This can be extended from this project for more sophisticated websites.

There are cases where the information from the tpGrid can be useful for pattern recognition and alignment processes. Below are the examples of the pattern that can be produced during the page analysis.

[52, 52, 52, 52, 52, 52, 52, 59, 52, 52, 52, 52, 52, 52, 59, 52, 52, 52, 52, 52, 52, 52, 52, 59, 52, 52, 52, 52, 52, 52, 52]

With this trail, when the item 52 is chosen as a control (as it is the first in the data cluster) some misleading information can be produced, in that the data cluster is a record of one field. The structure is then broken after the eighth item. The information is needed from the tpGrid to predict the size of the record cluster. With the trail above it can be observed that the cluster is eight field records. So, the analyser ignores the 52's until the eighth element is encountered. The size of the

record cluster can be obtained by finding difference of the consecutive terms of a tpGrid row.

```
Row 25: 51
Row 26: 52 53 54 55 56 57 58 60 61 62 63 64 65 67 68 69
70 71 72 73 74 76 77 78 79 80 81 82
Row 27: 59 66 75
Row 28: 83
```

The above figure is part of the tpGrid showing the data cluster. The rows 26 and 27 form part of the data cluster. When the common difference is found with row 26 it can be seen the common difference of 1 is found. Then the common difference is changed to 8 when items 58 and 60 are analysed. But when the common difference is calculated with items in row 27. the value of 8 is found. This suggests that the size of the record cluster is 8.

The technique for identifying the pattern of records can be done using the algorithm that was used by Smith and Waterman [32] which they used to identify the common pattern for molecular subsequence. This technique has been used by Gao. Andreae and Collins [30, 33] for their research they did for data extraction. on their paper titled. "Approximately Repetitive Structure Detection for Wrapper Induction". This technique has not been used in this project and hence can be done for further extension for this project.

Chapter 4

Technologies

4.0 Overview

I have explored various technologies which fit with my project. Some of them depend on the background knowledge I have while others depends on how powerful and useful can be used to model business logic. Below I have some technologies that can be used in my project and the reason for choosing them for the system developments.

4.1 Java and Java Server Pages

Java [38] is the programming language which is the most modern and 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 application and current technologies such as XML and SOAP.

Java has some of the following advantages

- ⇨ It is easy to learn
- ⇨ It is object oriented
- ⇨ It is platform independent

Actually I have chosen Java as I have learnt it and have more understanding on how to program with it comfortably. But the main reason is that, java have powerful collection framework. With Java collection I can produce various models for the web page during web page analysis.

In this project I am developing a web data extraction system 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 my project I have been using Java metadata which has enough function for performing metadata operations. I need to use metadata technology because the database schema will not be defined in advanced. When the new web data is extracted the table structure will be defined based on the structure of the web data extracted (see on the section of system design).

4.2 Database Management System

In this project I will used Mysql [52] as a database management system. Mysql database management system has advantages in that 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.

4.3 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 advanced. 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. I have used metadata technique to write general queries to the database. This makes my application more general to variety of applications.

4.4 JavaScript

JavaScript [53] 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. I came to find that JavaScript have strong functions that I can use for my project for processing the HTML source code during page analysis phase. With JavaScript I found that it has strong regular expression functionalities. Since it is more interactive then it makes the application to have high performance.

CHAPTER 5

System Analysis and Design

5.1 Overview

This section is about the services the system provides to the 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.

This chapter has four sections, with the first section showing functional requirements of the system and the second section showing the external behaviours of the system as seen by user known as non functional requirements of the system. The system architecture has been described showing the main components of the system and how they are related to each other. The last sections that have been discussed here are database design and system components.

5.2 Function requirements

The functional requirements have been identified by the main functionalities of the system. The identified functional requirements are page analysis, data extraction, data storage and User as shown below.

Use case diagram

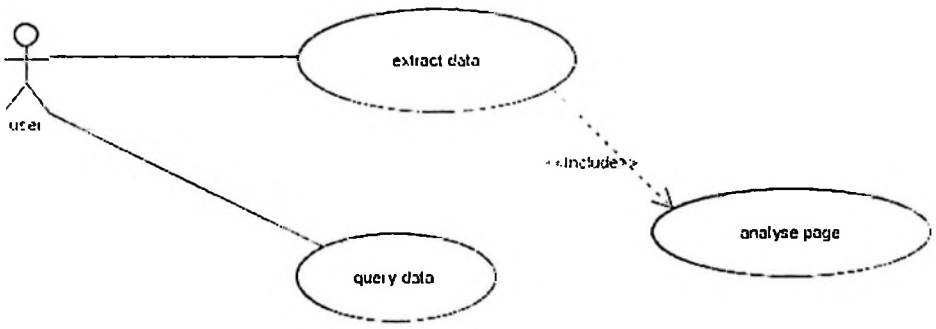


Figure 12 - Use case Diagram

5.2.1 Page analysis

Code	01
Name	Page analysis requirement
Purpose	The purpose of this function is to provide information about the location of the web data to be extracted.
Description	i. The system should be able to get HTML source code from user. ii. The system should be able to identify the data clusters in the HTML source code. iii. The user should be able to see and select the data clusters using check box controls and command buttons. iv. The system should be able to create the page descriptor from the page specified.

5.2.2 Data extraction

Code	02
Name	Data extraction
Purpose	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.
Description	i. 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 ii. The user should be able to see the data cluster extracted.

5.2.3 Data storage

Code	03
Name	Data Storage requirement
Purpose	This component use the data produced by data extractor and store then to the database. The data store component will also receive user query to the database and get the query results to the user.
Description	i. The system should be able to store the data cluster specified by user. ⇨ The system should be able to create tables based on the structure of the page descriptor. ⇨ The system should be able to insert the new data to the existing database table. ii. The user should be able to query the database. iii. The system should be able to provide the user with the result of the query. iv. The system should be able to inform the user if there no results of the query.

5.2.4 Client User Interface

Code	04
Name	Client User Interface
Purpose	This component will provide the user with the interface which will enable user to interact with the system. The user will provide search terms and the results of the query will also be displayed for the user.
Description	i. The use should be able to load the html page to

	the system using the text area on the browser window and click on the button. ⇨ The System should be able to create the page descriptor for the result web page. ii. The user should be able to select the cluster which contains data needed by user (see requirement 01-iii). iii. The system should be able to store on the file the data cluster specified by the user. After the user click the button to store the data to the database. iv. The user should be able to see in the tabled structure the result of the data extracted. v. The system should be able to extract data specified by other queries using the stored page descriptor for that site. vi. The system should be able to store the data cluster specified by user to the database.

5.3 Non functional requirements

5.3.1 Performance requirements

The system will cache the web data to the data store, this increase the efficient of the system in terms of processing time. Before the data is extracted the page descriptor will be produced. Then all the subsequent searches will use the produced 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

5.3.2 Quality Attributes

Security

This system will 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 required no

authentication to the users. This system can be used as a separate component to be used by other systems to extract data from the web pages.

The user should comply with the data protection act. Since the user uses data from other business organisations, the data should be processed further by complying with the data protection acts.

Availability

The system to be developed will be available in the sense that it runs at any point of time. The system will be able to detect the page format changes and hence inform the user. The system will 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.

Reliability

The system will 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 user can use the system at any time on demand.

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 term of modification. The functionality of page analyser can be modified without affecting the data extractor or data store.

5.4 System architecture

The system is design 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.

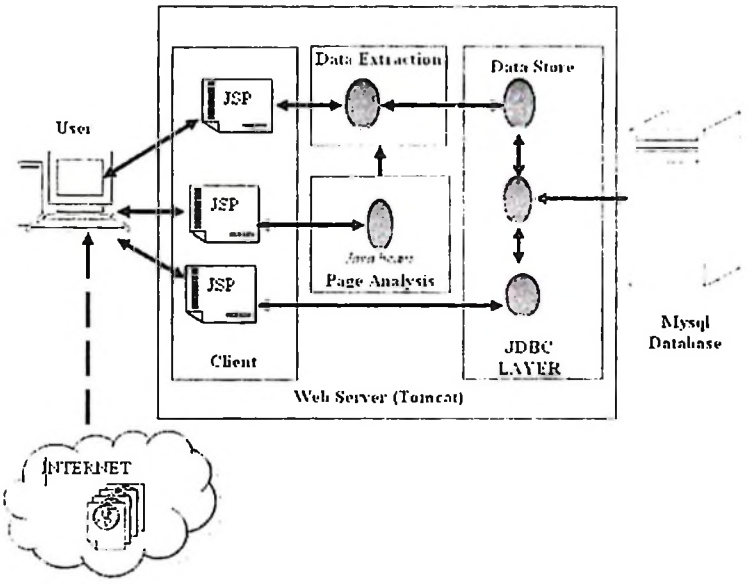


Figure 13 - System Architecture

5.5 Database design

5.5.1 Overview

The data model of the system is designed for data warehousing based application. So the information in the database tables is stored as supplement and not updating the existing information. The system has been design to provide a local cache for the web data. In doing so, the design of the data tables will be 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 design for data warehousing systems. In this case, there will be a central table which store the subjects for the search information and other tables will be the information that will be extracted from the web pages.

5.5.2 Fact table design

The central fact table is designed to contain 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 contain the reference (foreign key) information from the dimension tables, together with other aggregate information. In this data model design the fact table will contain the information about the web domain used for search, the subject for the search and the date when the information was 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 `subject` which contains four fields. This `subject` table will be created during the system installation.

Description of the fields of central table:

Field name	Data type	Description
ID	Integer	This field is primary key and It will be 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.

5.5.3 Data table design

The data tables stored the data that is extracted from the web pages. The design of the data tables depends on the structure of the data extracted. The name of the tables will be identified by the search keyword of the query used to generate the search results.

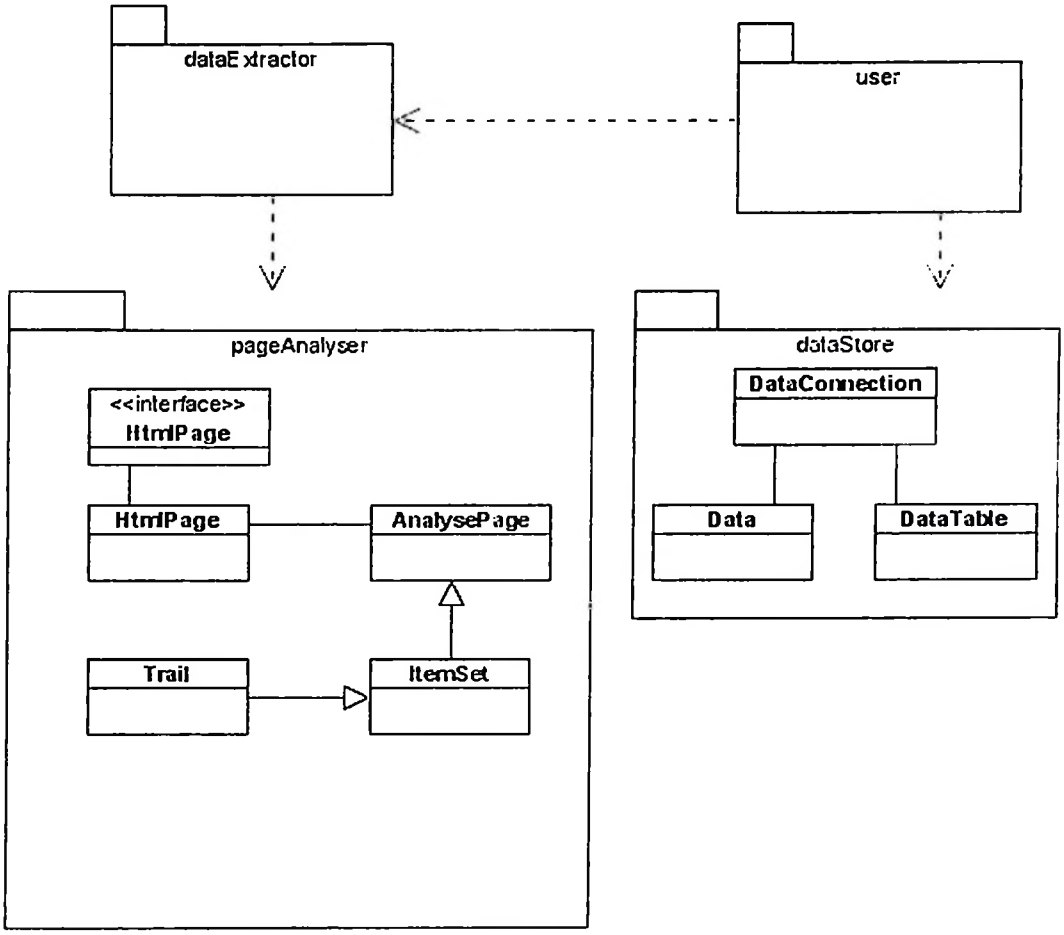
In this design the table field names with have general names. A typical table with five columns is shown below. It should be noted that, the data extracted from the web page is not refined in response to the data types. So, the data field will be stored as text string regardless of what type they represent.

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

5.6 System Component

5.6.1 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 user are mainly JSP pages.



5.6.2 Component description

Page analyser

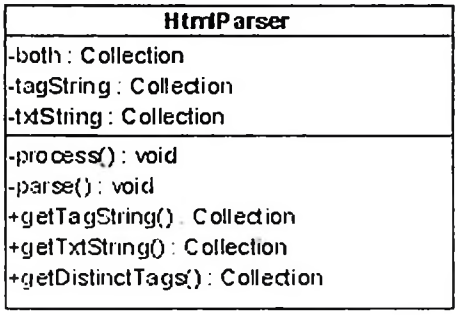
HtmlParser
Description This module stores lists of tag-String, text-String and distinct HTML tags. The module parses the HTML source code and sort out the HTML tags and text.
Algorithm i) Get HTML source code ii) Parse HTML source code to a sequenced list of HTML tags part and texts part as they appear on the HTML source code (both). iii) Using the list of from (ii) above 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 produce a list of numbered sequence tag sets with respect to the numbered sequence of the text String.

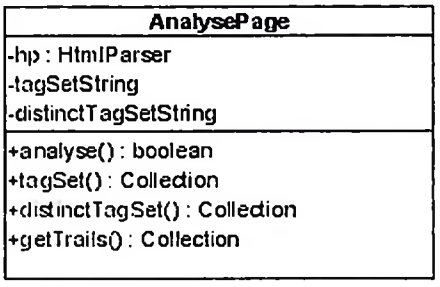
Algorithm

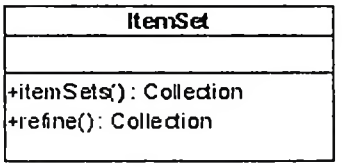
i. Get a list of tag String

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



ItemSet
Description This module is used to produce the items which have the name tag-Set string. The representation of the item set is tpGrid
Algorithm i. Get the list of tag-Set String and distinct Tag-Set String ii. 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 iii. Refine the Collection of item sets by comparing the tag-Set with those within the same cluster
UML class diagram 

Trail
Description This module uses the trail of item numbers in the cluster to identify the data records. In some cases the records needs to be align so as to have the same pattern. The aim of this module is to produce page descriptor.
Algorithm i. Get the trail of item sets. ii. For each cluster identify the records based on the similarity of patterns on the trail of item sets. 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

Data extraction

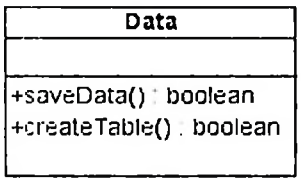
This component has a module which gets the page descriptor information and the collection of text-String from the component page analysis. The module iterated over the collection of text-String until the start of the data cluster is reached. The module starts creating the first record of the data cluster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

Data store

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

DataConnection
Description 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.
Algorithm i. load database driver ii. get database connection
UML class diagram DataConnection +getConnection() : Connection

Data
Description 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.
Algorithm i. Database table is created for new data. ii. The data is inserted into the database table one record at a time.
UML class diagram 

DataTable
Description 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 want to view the information already stored. Also with this module the data is queried from the database based on the name of the database table and user query. This module use 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.
Algorithm 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
UML class diagram

User

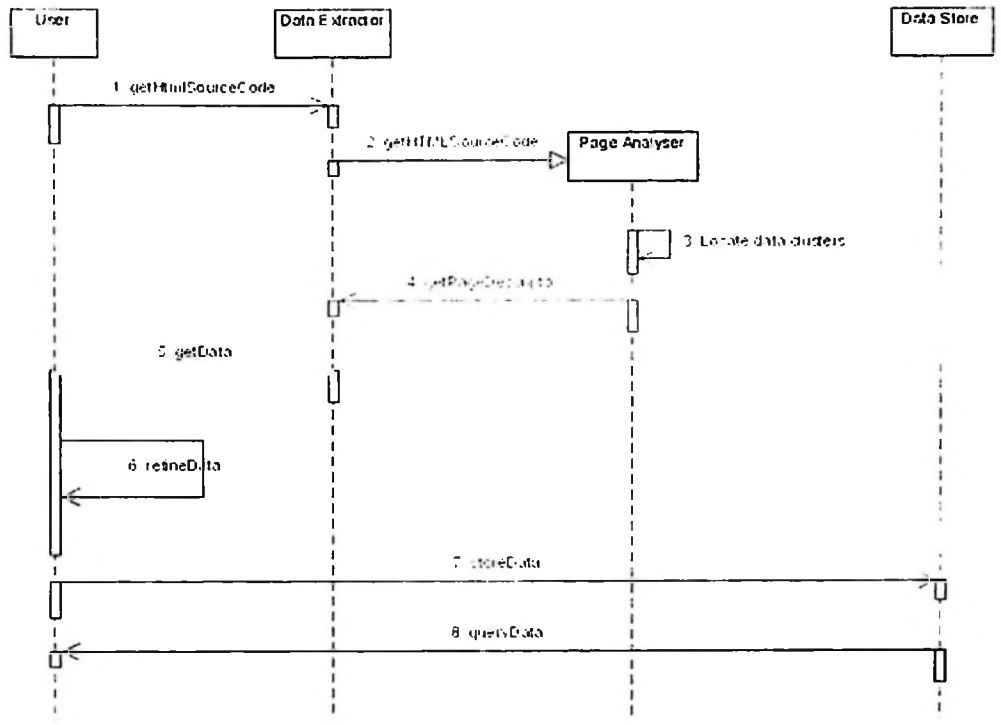
This component represents User interface, in this case, JSP pages, which provides a way that the user can interact with the system. With this system, the user can perform two main functions. These functions are to extract data from the web page and to query the information already stored in the database.

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 site 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 (see section 3.2 above).

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 will select the cluster of interest. The user also selects the fields of the record of 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.

5.7 Component interaction

Sequence diagram



CHAPTER 6

Detailed Design and Implementation

6.1 Overview

In this section I will discuss the implementation of the system in details. This section shows how the system has been implemented with respect to the suggested solution for the page analysis and data storage.

From the exploration of various suitable technologies to be used for the implementation of this system, now I present the technologies which I have used for implementation. 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 TOMCAT 4.1 [35]. The component which has been used by the system is mysql-connector which is used by java JDBC for Mysql database connection.

The implementation of this system has been organised into components. There are mainly four components where some are composed of java classes and others are composed of JSP pages. The implementation is discussed based on the order in which the extraction process takes place.

6.1.1 Working environment

The system is intended to run on web server, in which case I have used TOMCAT 4.1. The TOMCAT 4.1 was used because it is free web server and is easily available. TOMCAT 4.1 can be free downloaded from apache Jakarta web site (www.apache.org). Once the web server is installed into the system, settings for the environmental variable are set for CALALINA_HOME and CLASSPATH.

After the environmental variables have been set, the system is deployed on the "webapps" folder of TOMCAT HOME. 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.

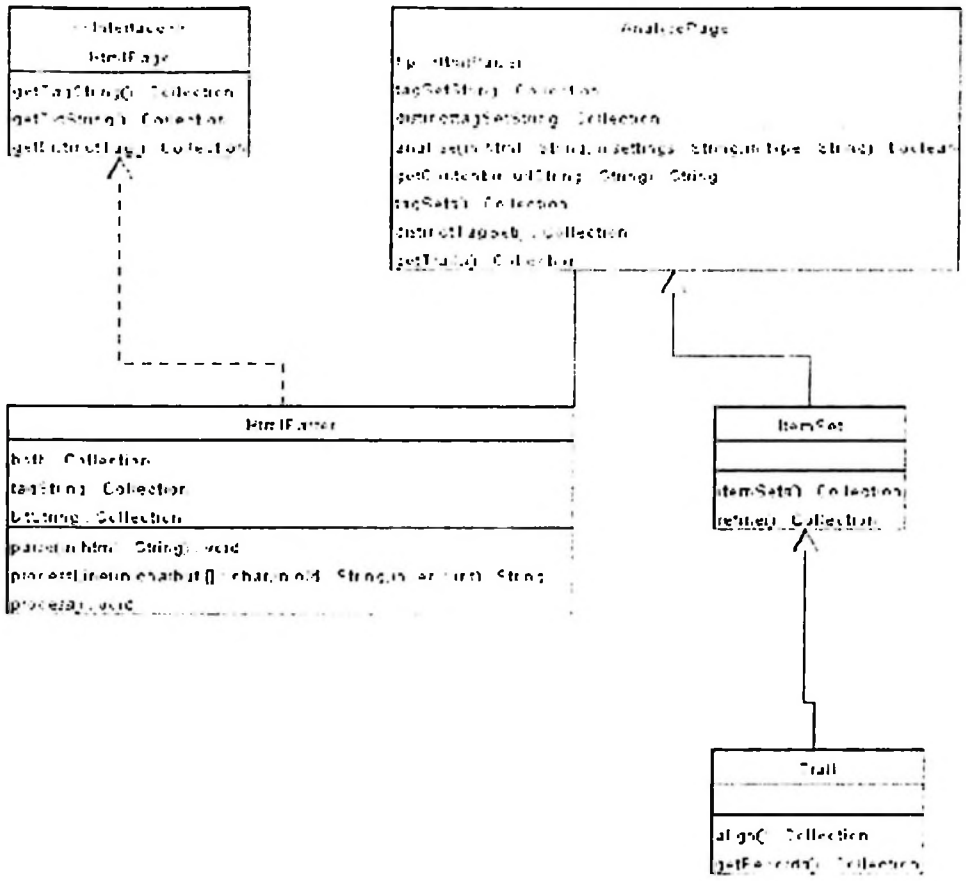
In this project I have used Mysql as a database management system. The Mysql database can freely be downloaded from the www.mysql.com website. The setup file is downloaded and then being installed on the computer system. When the Mysql database is installed and all the installation settings are made, 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 (see section 5.5.3). 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 and the general row names are used to name the database table fields.

Now the system is able to run under these environments.

6.2 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.



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 produced 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 to 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 have 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 (see chapter 3).

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 shows 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.

6.3 Data Extractor

This component has no 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 inform 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

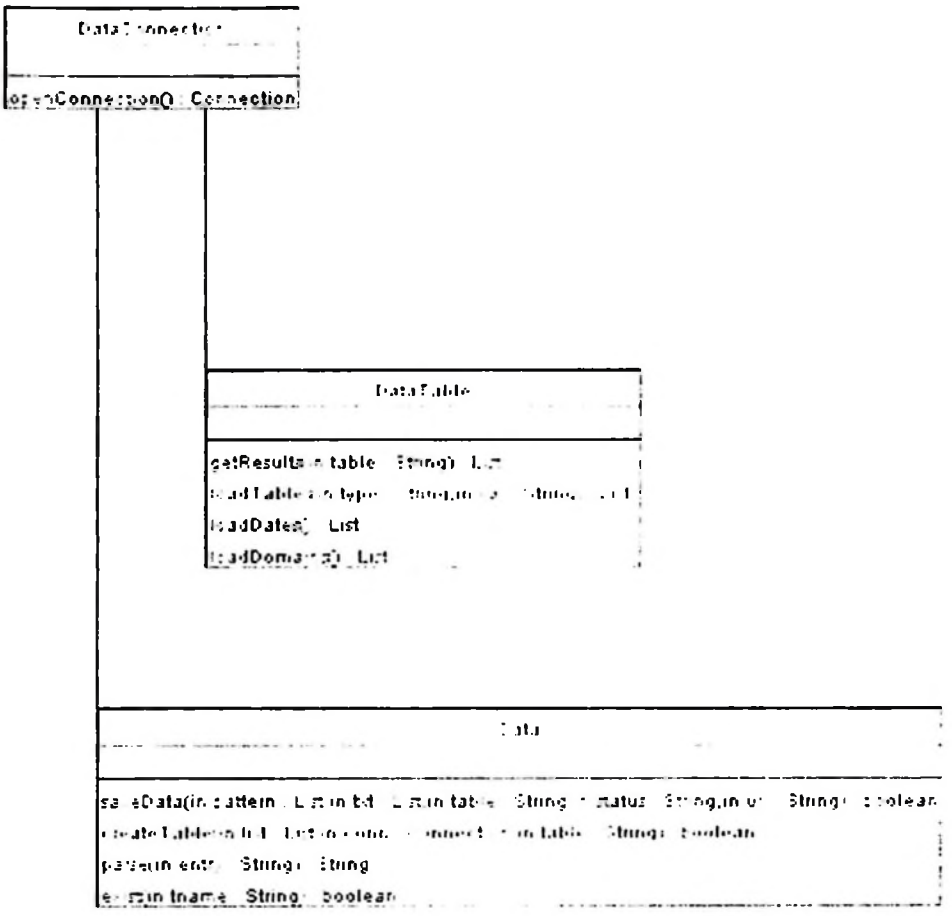
This page shows the cluster which was chosen by the user. The user is given more option to select the columns of interest. This is because some of the columns just present labels or they are empty. The user is provided with check box controls to select the columns of interest. Then the user is given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

6.4 Data storage

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



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 store 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 method for 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 specified the web site domain where the data have been extracted.

6.5 Client 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 submit the entries, process.jsp page get 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 for refining continues until when the data is stored to the database (see section 6.3).

Data query services.

This service has two JSP pages, these are 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 use for investigation purposes during page analysis.

CHAPTER 7

Testing

7.1 Overview

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

7.2 Unit Testing

During the implementation of the system unit testing was done in form of white box testing and black box testing. With white box testing, each statement of the code was tested against predefined input data. Then the result of the statement was compared with the expect output. If the case of branch statement paths was tested to ensure that the execution follow the expect 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 produced 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 working of the component need not to be known.

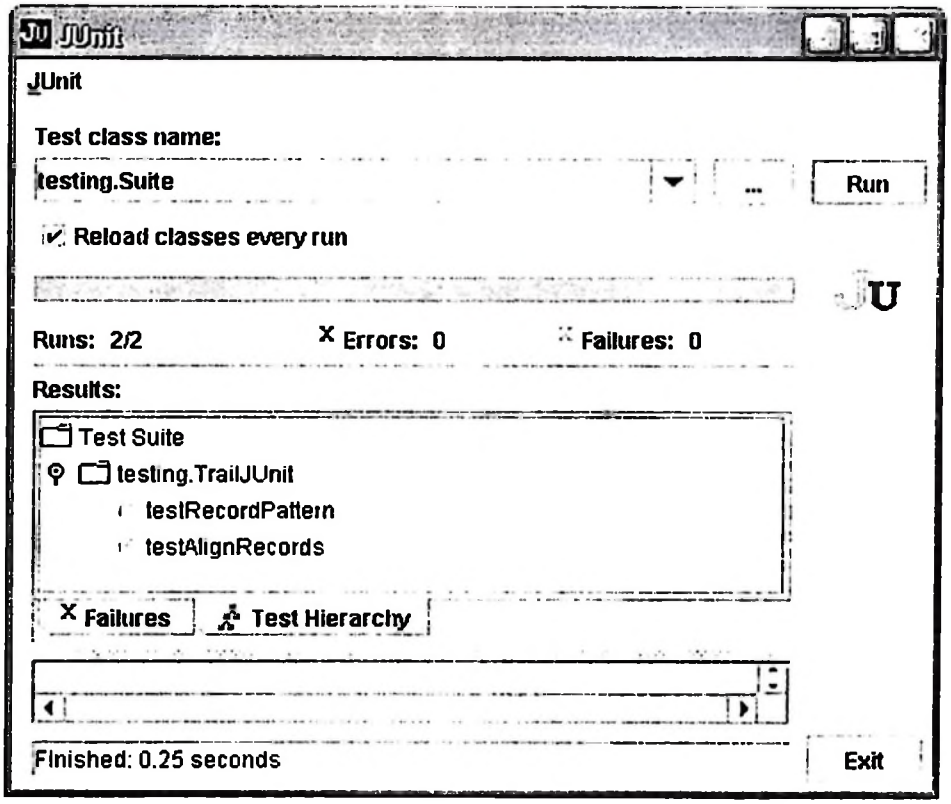
Black testing can be seen as redundant as the white box test all the individual statement within the function. The black box is necessary to ensure that all the statement within the function produces the expected result from the functional point of view. Black box needs no programming skills but rather input and out put 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.

7.2.1 Unit testing using JUnit Framework

This is the framework developed for testing during the extreme programming [40]. It is very useful for developing test cases. Using JUnit testing framework I have tested methods of each the classes by specifying the input data and compare 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 were tested.



Below is the summary of some of the methods which has been tested using the JUnit testing Framework.

Class Name	Method name	Input data	Output data	Evaluation
Trail	cluster Input: a list of trail with two items. The first item is the tag-Set number and the second item is the similar tag-Set number.	[20 20, 21 21, 22 22, 23 20, 24 21, 25 22, 26 20, 27 21, 28 22, 29 29, 23 20, 31 21, 32 22, 33 20, 34 21, 35 22]	[20: 20 21 22, 23: 20 21 22, 26: 20 21 22 29, 29: 20 21 22, 32: 20 21 22]	Success
Trail	getCluster Input:List of record pattern	[20: 20 21 22, 23: 20 21 22, 26: 20 21 22 29, 29: 20 21 22, 32: 20 21 22]	[20: 20 21 22 -1, 23: 20 21 22 -1, 26: 20 21 22 29, 29: 20 21 22 -1, 32: 20 21 22 -1]	success

7.3 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.

7.3.1 Top down integration

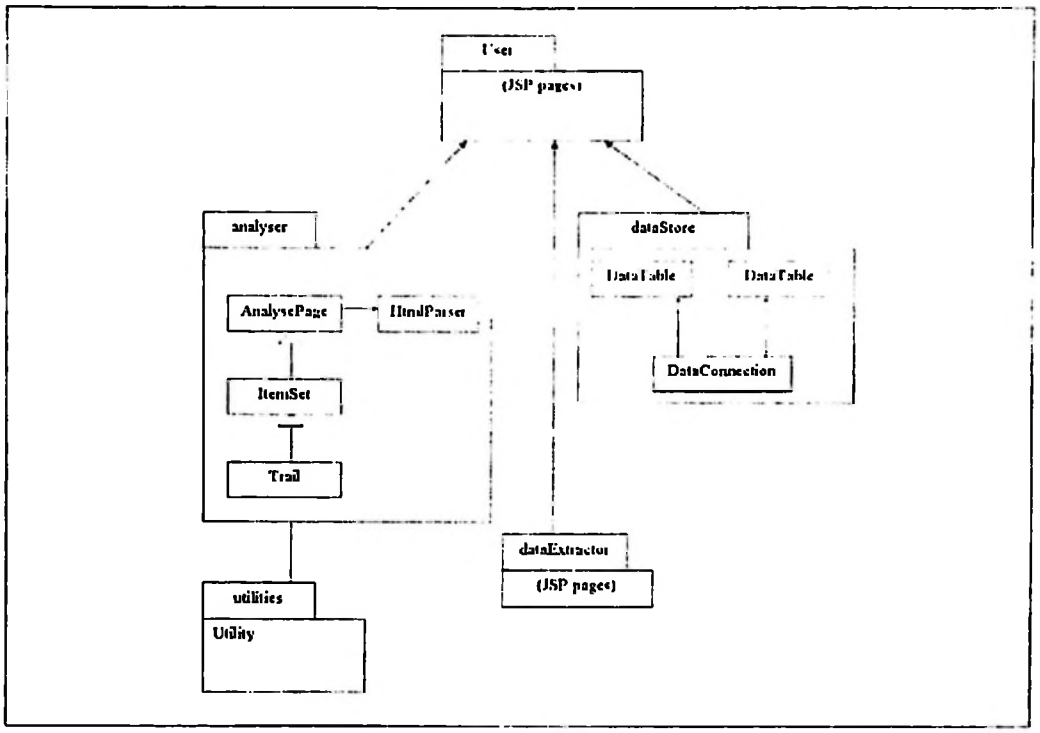
With top down approach the system structure is constructed incrementally. The system is constructed by following the control hierarchy. The main top module after being tested by the test stub is integration by the sub modules under the control hierarchy. The main top module acts as the test driver for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

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



7.4 System Testing

When unit testing, component testing and integration testing have been successful done, then the whole system is tested against system behaviours. System testing explores system behaviours which are not tested by unit testing, component testing nor integration testing. System testing involves, performance testing, installation testing.

7.4.1 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 takes to the hard disk and the speed of the computer processor.

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.

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.

7.4.2 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 XP) where the Java Virtual Machine has been installed. The system was found to be working properly without any changes to system operations.

CHAPTER 8

Conclusion, Evaluation and Future Work

8.1 Evaluation

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

- ⇨ 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 was extension to the project done by Robinson J. [1, 2, 3, 12]. 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.
- ⇨ 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 was successful to investigate some HTML tags which affect the process of clusters identification.
- ⇨ 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 result was found to produce good results. This main structure was found to be the presence of optional fields.
- ⇨ I have tested the system to various web domains which provides data searching functionalities. The figure below shows the web domain tested and the quality of the results produced. See appendix H for more information.

Web domain	Quality of results
www.mamma.com	100%
www.a1books.com	100%
www.ibm.com	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%

⇨ It was found that, metadata functionalities should be used to produce the general codes. These codes can work to any system. In this case the schema for the data in the database is defined automatically based on the structure of the data extracted.

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 for 65% of the good results.

8.2 Conclusion

The growing bulks of information on the web data is the 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 research so that the web data can be available to the software applications. The main objective is to develop a wrapper which will extract data to hence make available to other application.

In this project I have presented the use of tag-String as a structure for identifying the repetitive pattern. 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 is organised into stages. These stages are defined based on the regularity of the repetitive patterns of tag-String as they found in various web domains. 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.

The next stage was to investigate how HTML tags has 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 was important for the page analyser to ignore these tags for better results.

The other stage was to investigate the problems where the records of the data clusters have different number of data fields. In these cases other fields are optional fields. as they are not found on some other fields. The new approach was to find the template record and use the template record to identify the missing fields and mark them as optional fields.

The final investigation was on the clusters with nested data items. The proposed solution was to use 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 when I searched information about DVDs. It was found that. in some records there are nested data patterns. Moreover these nested patterns where optional field.

8.3 Future work

There is a lot to be done on the project for 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.

The extension to this project is to investigate the production of the general browser which should be a wrapper to the search form of web sites which has 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.

The other work which should be extended to this research is to investigate the use of semantic techniques for the 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.

The extension to this work is to investigate the use of information from the tpGrid to solve the problems of nested data items. Also, there are pattern 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.

The other part of the research which has not been done in this project is the 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.

Bibliography

Data extraction resources

[1]. J. Robinson. "Data Extraction from Web Data Sources". Proc WBC'04. 4th International Workshop on Web Based Collaboration. 2004

[2]. J. Robinson," Data Extraction from Web Database Query Result Pages via TagSets and Integer Sequences". Proc IADIS WWW/Internet International Conference 2003.

[3]. Robinson J., "Data Extraction from Web Pages containing Query Result". http://cscourse.essex.ac.uk/course/cc433/publications/magazine.pdf. 2005
Last accessed by September 2005.

[4] Richard D. Hackathorn. "Web Farming for the Data Warehouse". The Morgan Kauimann Series in Data Management Systems. Jim Gray. Series Editor. November 1998.

[5] J. Hammer. H. Garcia-Molina. J. Cho. R. Aranha. A. Crespo. "Extracting Semistructured Information from the Web". Proc. of the Workshop on Management of Semistructured Data. Tucson. Arizona. May 1997.

[6] Kuhlins. S. and Tredwell. R.. "Toolkits for Generating Wrapper". http://www.dit.hcmut.edu.vn/~tru/SPECIAL-STUDIES/survey-on-wrappers.pdf. 2002.
Last accessed by September 2005.

[7] Myllymaki. J.. "Effective Data Extraction with Standard XML Technologies". http://www.research.ibm.com/people/j/jussi/papers/ANDES/ANDES.pdf. 2001.
Last accessed by September 2005.

[8] Embley D. W., "Toward Tomorrow's Semantic Web -- An Approach Based on Information Extraction Ontologies", Position Paper for Dagstuhl Seminar, January 2005, http://www.smi.ucd.ie/Dagstuhl-MLSW/proceedings/embley.pdf
Last accessed by September 2005.

[9] The RISE Repository of Online Information Sources, Used in Information Extraction Tasks, University of Southern California, Information Sciences Institute http://www.isi.edu/info-agents/RISE/index.html, 1998.

[10] Eikvil L., "Information Extraction from World Wide Web-A Survey", Norweigan Computing Center, 1999,
http://citeseer.ist.psu.edu/cache/papers/cs/13218/http:zSzzSzwww.nr.nozSzbildzSzPostScriptzSzwebIE_rep945.pdf/eikvil99information.pdf
Last accessed by September 2005

[11] Embley D. W., Jiang Y., NG Y., "Record – Boundary Discovery in Web Documents", 1999.
http://www.sigmod.org/sigmod/sigmod99/eproceedings/papers/vkng.pdf,
Last accessed by September 2005

[12] Robinson J., "Analysing Web Pages for Automatic Wrapper Production in Data Extraction", 2004.
http://escourse.essex.ac.uk/course/cc433/publications/wrappers.pdf,
Last accessed by September 2005

[13] Arvind Arasu and Hector Garcia-Molina, "Extracting Structured Data from Web Pages", In the proceedings of ACM SIGMOD International Conference on Management Data (SIGMOD 2003), San Diego, California, USA, June 9-12 2003, ACM Press.

[14] Robert Baumgartner, Sergio Flesca, and Georg Gottlob, "Visual Web Information Extraction with Lixto", In the Proceedings of 27th International Conference on Very Large Data Bases (VLDB 2001), pages 119-128, Rome, Italy, September 11-14 2001.

[15] Valter Crescenzi, Giansalvatore Mecca, and Paolo Merialdo, "Road Runner: Towards Automatic Data Extraction from Large Websites." In the Proceedings of 27th International Conference on Very Large Data Bases (VLDB 2001), pages 109-118, Rome, Italy, September 14 2001.

[16] Adelberg B, "NoDoSE: A Tool for Semi-Automatically Extracting Structured and Semi-Structured Data from Text Documents", SIGMOD Record 27, 2 (1998), 283-294.

[17] Ashish N, Knoblock C, "Wrapper Generation for Semi-Structured Internet Sources" SIGMOD Record 26(1997), 8-15.

[18] Embley D W, Jiang Y S, Campbell D M, Liddle S W, Kai Ng Y, Quass D and Smith R D, "Conceptual-Model-Based Data Extraction from Multiple-Record Web Pages", Data and Knowledge Engineering 31, 3(1999),227-251.

[19] Freitag D, "Machine Learning for Information Extraction in Informal Domains", Machine Learning 39, 2/3 (2000), 169-202.

[20] Califf M E, and Mooney R J, "Relational Learning of Pattern-Match Rules for Information Extraction", In Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence, (Orlando, Florida,1999), pp. 328-334.

[21] Stephen Soderland, "Learning Information Extraction Rules for Semi-structured and Free Text, Machine Learning" 34(1-3) pp 233-272, 1999.

[22] S.W. Liddle, D.W. Embley, D.T. Scott, and S.H. Yau, "Extracting Data Behind Web Forms", Proc Wkshop on Conceptual Modeling Approaches for e-Business, 2002.

[23] C-H Chang, S-C Lui, and Y-C Wu, "Applying Pattern Mining to Web Information Extraction", Proc PAKDD 2001, 5th Pacific-Asia Conf on Knowledge Discovery and Data Mining, pp 4-16, LNAI 2035.

[24] I. Muslea and S. Minton and G. Knoblock, "A Hierarchical Approach to Wrapper Induction", Proc 3rd Conf on Autonomous Agents (1999).

[25] Sahuguet A, and Azavant F, "Building Intelligent web applications using lightweight wrappers", Data and Knowledge Engineering 36, 3 (2001), 283-316.

[26] Hemnani A and Bressan S, "Information Extraction – Tree Alignment Approach to Pattern Discovery in Web Documents" Proc. DEXA 13th International Conference on Database and Expert System Applications 2002.

[27] Gao X et al, "Learning Information Extraction Patterns from Tabular Web Pages without Manual Labelling" Proc. IEEE/WIC International Conference on Web Intelligence (WI2003), pp495-498, 2003.

[28] Wang J and Lochovsky F.H, "Data Extraction and Label Assignment for Web Databases", Proc. 12th International World Wide Web Conference, WWW03, 2003

[29] Boris Chidlovskii, Jon Ragetli and Maarten de Rijke. *Wrapper Generation via Grammar Induction.* Proc. Eleventh European Conference on Machine Learning, Barcelona, 2000. Springer Lecture Notes in Artificial Intelligence vol. 1810, pp. 96 – 108.

[30] Jon Ragetli, "Semi-Automatic Parser Generation for Information Extraction from the WWW", Master's Thesis for the studies Artificial Intelligence at the Faculty of Mathematics, Computer Science, Physics and Astronomy, University of Amsterdam, August 1998.
http://home.12move.nl/~sh364624/publicaties.html Last accessed by September 2005.

[31] Kushmerick Nicholas, "Wrapper induction: Efficiency and expressiveness.", Artificial Intelligence J. 118(1-2):15-68 (special issue on Intelligent Internet

Systems). http://www.cs.ucd.ie/staff/nick/home/research/download/kushmerick-aij2000.pdf, 2000.

[32] Smith T., F., and Waterman M., S., "Identification of common molecular subsequences", J. of Mol. Biol., 147:197. 1981.

[33] Xiaoying Gao, Peter Andreae, Richard Collins, "Approximately Repetitive Structure Detection for Wrapper Induction", Technical Report CS-TR-04/9 June 2004 http://www.mcs.vuw.ac.nz/comp/Publications/archive/CS-TR-04/CS-TR-04-9.pdf last accessed on September 2005.

Database resources

[34] Welling L. and Thomson L., PHP and MySQL Web Development, Indianapolis, IN : Sams, c2001.

[35] http://www.apache.org

[36] Elmasri R. and Navathe S. B., "Fundamentals of database systems", 4th Ed., Addison Wesley, 21st August, 2003

[37] Russell, Chad; Stephens, Jon, "Beginning MySQL Database Design and Optimization: From Novice to Professional", APRESS, Dec 2004.

Java resources

[38] Java technology, http://java.sun.com.
This site contains the information of all the technologies which uses java as programming language. There are also software products which are freely downloaded.

Java JDBC

[39] Taylor Art, "JDBC developer's resource: database programming on the Internet", Prentice Hall Prt; Bk & CD-ROM edition, 1997

[40] Norman, Metthew, "Database Design Manual: using MySQL for Windows", London: Springer, c2004.

Others

[41] Volker N. "Advanced Java and Software Engineering", Lecture Notes CC438. 2005. There are good notes useful especially for system testing which includes tools such as JUnit testing.

[42] Graham, Ian S., "HTML 4.0 sourcebook", New York: John Willey & Sons. 4th Ed., 1998.
It is a good text books for understanding structures for building web pages using HTML tags.

[43] Castro, E., "HTML for the World Wide Web", Berkeley: Peachpit. 5th Ed., c2003

[44] Connolly T., M., Begg C., E., "Database systems: a practical approach to design, implementation, and management", Harlow: Addison-Wesley, 4th Ed. 2005.

[45] Conallen J., "Building Web applications With UML", Addison-Wesley, 2000
It's very useful to understand the whole process for analysis and design using UML.

[46] E.Gamma and K. Beck, "Simple Java regression unit testing framework "http:www.junit.org.

[47] W3C, www.w3.org/MarkUp. This site contains the specification for the standard technologies for mark-up language which includes HTML, XHML and XML.

[48] _ _, www.w3schools.com. This site contains the tutorial which can be used at any level of expertise (novice, intermediate of advanced user) for web based technologies which includes HTML, XHTML, and XML.

[49] W3C RSS 1.0 News Feed Creation How-To http://www.w3.org/2001/10/glance/doc/howto

[50] _ _. Unified Modeling Language
http://www.omg.org/technology/documents/formal/uml.htm

[51] _ _. HPH. http://www.php.net/

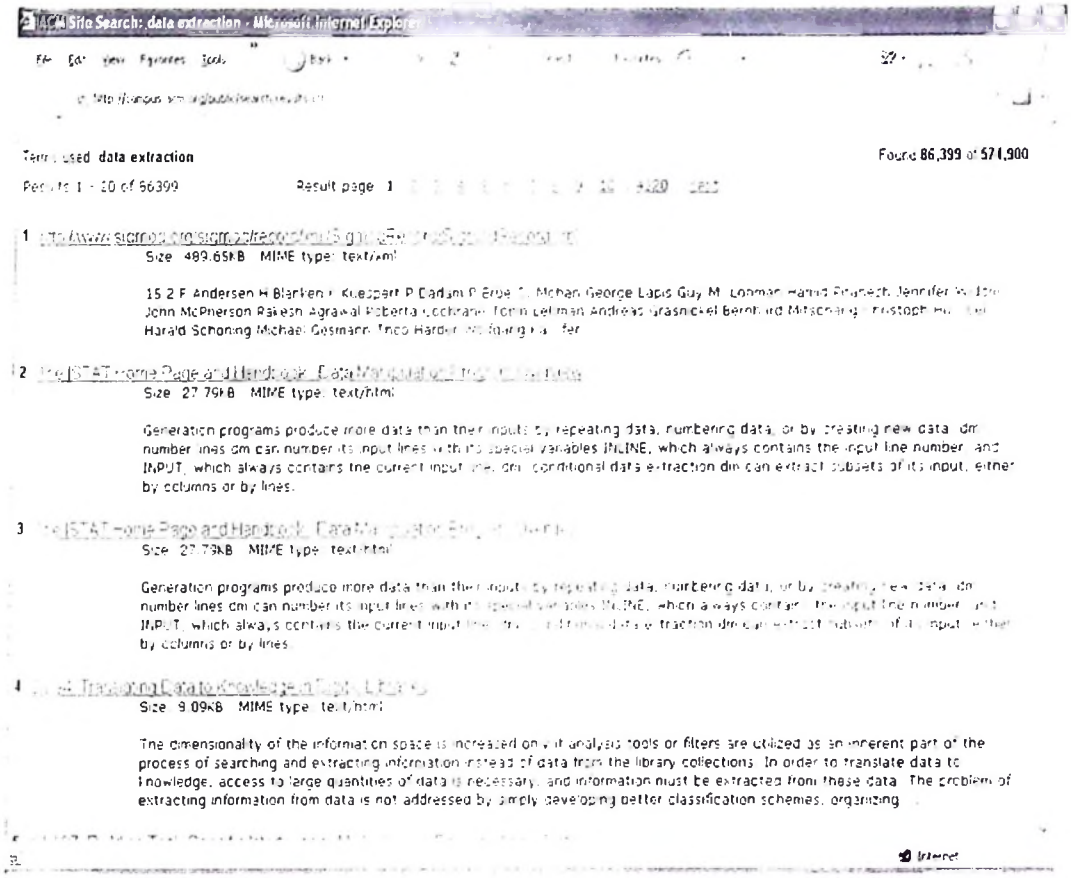
[52] _ _. Mysql Database resources. http://www.mysql.com

[53] Holzner, Steven. "JavaScript complete". New York : McGraw-Hill, c1998

Appendices

Appendix A – Web page analysis

The typical web page showing the request in response to the search query using the keyword "data extraction" on the web domain http://campus.acm.org.



The HTML source code from this page is obtained by clicking on the "view" menu of the browser, and selecting the option "source". The window will show up containing the HTML source code. Select all from the edit menu and copy to the clip board. Then this HTML source code is input to the System. Below is the typical text String as sorted from the HTML source code.

```
0: ACM Site Search: data extraction
1:
2:
```

```
3:
4:
5: Web Site Search:
6:
7:
8: Search Tips
9: Terms used:
10: data
11:
12: extraction
13:
14: Found
15: 86.399
16: of
17: 574.900
18: Results 1 - 20 of 86399
19: Result page:
20: 1
21:
22: 2
23:
24: 3
25:
26: 4
27:
28: 5
29:
30: 6
31:
32: 7
33:
34: 8
35:
36: 9
37:
38: 10
39: ...
40: 4320
41:
42: next
43: 1
44: http://www.sigmod.org/sigmod/record/xml/SigmodRecord/SigmodRecord.xml
45:
46: Size: 489.65KB   MIME type: text/xml
47: 15 2 F Andersen H Blanken K Kuespert P Dadam R Erbe C. Mohan George Lapis
Guy M. Lohman Hamid Pirahesh Jennifer Widom John McPherson Rakesh Agrawal
Roberta Cochrane Tobin Lehman Andreas Grasnickel Bernhard Mitschang Christoph
Hu¨:bel Harald Schöning Michael Gesmann Theo Härder Wolfgang Ka¨:fer
48:
49: 2
```

50: The ISTAT Home Page and Handbook : Data Manipulation Program Overview
51:
52: Size: 27.79KB MIME type: text/html
53: Generation programs produce more data than their inputs by repeating data. numbering data. or by creating new data. dm: number lines dm can number its input lines with its special variables INLINE. which always contains the input line number. and INPUT. which always contains the current input line. dm: conditional data extraction dm can extract subsets of its input. either by columns or by lines.
54:
55: 3
56: The ISTAT Home Page and Handbook : Data Manipulation Program Overview
57:
58: Size: 27.79KB MIME type: text/html
59: Generation programs produce more data than their inputs by repeating data. numbering data. or by creating new data. dm: number lines dm can number its input lines with its special variables INLINE. which always contains the input line number. and INPUT. which always contains the current input line. dm: conditional data extraction dm can extract subsets of its input. either by columns or by lines.
60:
61: 4
62: DL94: Translating Data to Knowledge in Digital Libraries
63:
64: Size: 8.09KB MIME type: text/html
65: The dimensionality of the information space is increased only if analysis tools or filters are utilized as an inherent part of the process of searching and extracting information instead of data from the library collections. In order to translate data to knowledge, access to large quantities of data is necessary, and information must be extracted from these data. The problem of extracting information from data is not addressed by simply developing better classification schemes. organizing ...
66:
67: 5
68: CHI 97: Building Task-Specific Interfaces to High Volume Conversational Data
69:
70: Size: 42.62KB MIME type: text/html
71: Phoaks maintains an experimental web site (http://www.phoaks.com/phoaks/) that contains over 37.000 pages of recommendation data for these newsgroups. The first Phoaks application attacks the problem of extracting recommendations of Web resources (URLs) from Usenet messages and creating interfaces to the recommendation data. Phoaks maintains a database of recommended resources and associated contextual information. and generates web pages as an interface to the recommendation data.
72:
73: 6
74: pods2000.dvi
75:
76: Size: 243.34KB MIME type: application/pdf
77: 2 For the following examples we use the expression E1 E2 to represent the regular expression that recognizes the regular set LE1 LE2 Example 46 (Maximal Extraction Expressions Although it might not be immediately obvious both (p hpi and (qp ((p qhpi are maximal extraction expressions 2 Example 47 (Unambiguity and Maximality

Given a nonmaximal unambiguous extraction expression E over if there exists a maximal extraction expression E 0 over such that E E 0 , we say that extraction expression...
78:
79: 7
80: http://www.acm.org/~perlman/stat/handbook.pdf
81:
82: Size: 168.65KB MIME type: application/pdf
83: 1- 3 Data Manipulation Programs Data Analysis Programs 1.4 Table of UNIX and MSDOS Utilities ..... With data manipulation programs, extractions from the master data file are transformed into a format suitable for input to an analysis program. 3- 2 Conventions STAT Handbook Section 3.2 Command Formats STAT programs are run on UNIX and MSDOS by typing the name of the program, program options, and ...
84:
85: 8
86: http://www.acm.org/~perlman/stat/handbook/handbook.txt
87:
88: Size: 130.60KB MIME type: text/plain
89: With data manipulation programs, extractions from the master data file are transformed into a format suitable for input to an analysis program. The following command tells {dm} to repeat each input line with {INPUT}, and then print the weighted sum of columns 4, 5, and 6, treated as numbers. The {regress} program assumes its input has the predicted variable in column 1 and the predictors in following columns.
90:
91: 9
92: |STAT Handbook
93:
94: Size: 145.83KB MIME type: text/html
95: With data manipulation programs, extractions from the master data file are transformed into a format suitable for input to an analysis program. |STAT programs are run on UNIX and MSDOS by typing the name of the program, program options, and program operands (e.g., expressions or file names). Simple Commands A simple command consists of a program name, program options delimited with minus signs, and program operands, such as file or variable names.
96:
97: 10
98: pods.dvi
99:
100: Size: 408.11KB MIME type: application/pdf
101: it D. E. I. S. Universit ` a della Calabria 87036 Rende (CS), Italy ABSTRACT We present the Lixto project which is both a research project in database theory and a commercial enterprise that de velops Web data extraction (wrapping and Web service denition software We discuss the projects main moti vations and ideas in particular the use of a logicbased framework for wrapping Then we present theoretical re sults on monadic datalog over trees and on Elog its close relative which is used as ...
102:
103: 11
104: http://acm.org/sigs/sigkdd/explorations/issues/6-2-2004-12/1-Ruth-Zhang.pdf

105:
106: Size: 255.73KB MIME type: application/pdf
107: We apply an approximate string matching method to detect those kinds of duplicates After the four steps of an iteration a set of records is ob tained To track the relationships between occurrences and patterns for each record we store a list of indices of patterns that this record matches Intuitively the records that have more occurrences are more likely to be correct The records that have more than one occurrence can be selected as the seed set of the next it eration One of the advantages ...
108:
109: 12
110: http://www.sigada.org/wg/asiswg/specs/asis-data_decomposition.ads
111:
112: Size: 61.47KB MIME type: text/plain; charset=UTF-8
113: ---------- -- 22.3 type Array_Component ---------- -- Type Array_Component describes the components of an array valued field for a record -- type. ---------- -- Type_Definition - Specifies the array type definition to query -- Component - Specifies a component which...
114:
115: 13
116: http://www.sigada.org/wg/asiswg/specs/asis20s.txt
117:
118: Size: 616.36KB MIME type: text/plain
119: -- ---------- -- Package ASIS Types: -- -- The following types are made visible directly through package Asis: -- type ASIS_Integer -- type ASIS_Natural -- type ASIS_Positive -- type List_Index -- type Context -- type Element -- type Element_List -- Element subtypes -- Element Kinds (set of enumeration types) -- type Compilation_Unit -- type Compilation_Unit_List -- Unit Kinds (set of enumeration types) -- type ...
120:
121: 14
122: Extracting Semantic Metadata and Its Visualization
123:
124: Size: 40.15KB MIME type: text/html
125: As no other relations refer to the 'SALARIED_PERSON' relation, the relationship between the relations is initially detected as a binary relationship. In this example, the 'SALES_PERSON' relation's foreign key attribute (i.e., 'EmpNo') is constrained as a primary key attribute for the 'SALES_PERSON' relation and the 'EMPLOYEE' relation contains an existing super/subclass relationship with the 'SALARIED_PERSON' relation. Therefore, it is concluded that the super/subclass relationship between ...
126:
127: 15
128: Microsoft Word - p0-cover-logos.doc
129:
130: Size: 6,491.03KB MIME type: application/pdf
131: 5M. J. Carey and J. Han RESEARCH ARTICLES AND SURVEYS Peer- to-Peer Management of XML Data: Issues and Research Challenges ..... More details

about distributed data mining could be found in [47]. Recently, the data generation rates in some data sources become faster than ever before. 3- Mining Techniques Mining data streams has attracted the attention of data mining community for the last three years.
132:
133: 16
134: Microsoft Word - editorial-v2.doc
135:
136: Size: 159.64KB MIME type: application/pdf
137: The research in Web mining aims to develop new techniques to effectively extract and mine useful knowledge or information from these Web pages [8]. Due to the heterogeneity and lack of structure of Web data, automated discovery of targeted or unexpected knowledge/ information is a challenging task. In the past few years, there was a rapid expansion of activities in the Web mining field, which consists of Web usage mining, Web structure mining, and Web content mining. Web content mining ...
138:
139: 17
140: survey.dvi
141:
142: Size: 247.97KB MIME type: application/pdf
143: , qm as sm ', where the qi 's are queries and each of the si 's is either a string query or the keyword schema The \as clauses create the URLs s1 s2 : :: , sm , which are assigned to the new pages resulting from each query qi . We illustrate StruQL with a query dening a web site starting with a Bibtex bibliography le modeled as a labeled graph The web site will consist of three kinds of pages a PaperPresentation page for each bibliography entry aYear page for each year pointing to all ...
144:
145: 18
146: http://www.sigmod.org/sigmod/record/issues/0109/a2-pvur.pdf
147:
148: Size: 61.87KB MIME type: application/pdf
149: · To analyze and interpret data using simulation models and other complex analytical programs, thereby generating new value- added data. Publishing programs in addition to data sources, and the ability to embed calls within SQL queries proved very useful when dealing with autonomous and heterogeneous data sources. Similarly to data, models and programs, data processing chains should be published through Le Select.
150:
151: 19
152: Microsoft Word - tia_detailed_info__5_19_500pm_.doc
153:
154: Size: 1,350.46KB MIME type: application/pdf
155: 1 Program Information DARPA's Information Awareness Office Since 1996, the Defense Advanced Research Projects Agency (DARPA) has been developing information technologies to counter asymmetric threats. 3 Data Search, Pattern Recognition, and Privacy Protection Programs - Genisys (data base access, data repository, and privacy protection) - Evidence Extraction and Link Discovery (EELD) - Scalable Social Network Analysis (SSNA) - MisInformation Detection (MInDet) - Bio- Event Advanced Leading ...

156:
157: 20
158: sigmodrecord.dvi
159:
160: Size: 204.07KB MIME type: application/pdf
161: A Sample Document XWRAP Elite System Human Input Tagging Rules Objects Elements XML Output Real Documents Objects Elements Output Tagging XML. Output A Generated Wrapper Analysis Element Sep. Object Extraction Subtree. Obj. always a single HTML tag that marks the boundary between objects is not valid when applied to elements As we pointed out earlier an element separator can also be a text delimiter We choose two dierent approaches to discover tag separators and textseparators First we build...
162:
163: Results 1 - 20 of 86399
164: Result page:
165: 1
166:
167: 2
168:
169: 3
170:
171: 4
172:
173: 5
174:
175: 6
176:
177: 7
178:
179: 8
180:
181: 9
182:
183: 10
184: ...
185: 4320
186:
187: next
188: Association for Computing Machinery. Copyright © 2005 ACM. Inc.
189: Privacy Policy
190:
191: Code of Ethics
192:
193: Contact Us

The tag String from this site is shown below as sorted from the HTML source code

0:<!doctype> <html> <head> <link> <link> <title>
1:</title> </head> <body> <table> <table> <tr> <td> <a> <img> </td>

<td> <table> <tr> <td>
2:</td> <td>
3:</td> <td>
4:</td> <td> </td> </tr> </table> <table> <form> <input> <input> <tr> <td> <img>
</td> </tr> <tr> <td>
5:</td> <td> <a>
6:
 <input>
7:<input>
8:<img> <a>
9: </td> </tr> </form> </table> </td> </tr> </table> <tr> <td> <table> <tr> <td>
<table> <tr> <td>
10:
11:
12:
13:
14:</td> <td>
15:
16:
17:
18: </td> </tr> </table> </td> </tr> <tr> <td> <table> <tr> <td>
19:</td> <td>
20:
21:
22:<a>
23:
24:<a>
25:
26:<a>
27:
28:<a>
29:
30:<a>
31:
32:<a>
33:
34:<a>
35:
36:<a>
37:
38:<a>
39:
40:<a>
41:
42:<a>
43: </td> </tr> </table> </td> </tr> <tr> <td> <hr> <table> <tr> <td>
44: </td> <td> <table> <col> <col> <col> <tr> <td> <a>
45: </td> </tr> <tr> <td> </td> <td>
46:</td> <td>
47:

48:

```
49:</div> </td> </tr> </table> </tr> <tr> <td> <strong>
50:</strong> </td> <td> <table> <col> <col> <col> <tr> <td> <a>
51:</a> </td> </tr> <tr> <td> </td> <td>
52:</td> <td> <div>
53:<br> <br>
54:<br>
55:</div> </td> </tr> </table> </tr> <tr> <td> <strong>
56:</strong> </td> <td> <table> <col> <col> <col> <tr> <td> <a>
57:</a> </td> </tr> <tr> <td> </td> <td>
58:</td> <td> <div>
59:<br> <br>
60:<br>
61:</div> </td> </tr> </table> </tr> <tr> <td> <strong>
62:</strong> </td> <td> <table> <col> <col> <col> <tr> <td> <a>
63:</a> </td> </tr> <tr> <td> </td> <td>
64:</td> <td> <div>
65:<br> <br>
66:<br>
67:</div> </td> </tr> </table> </tr> <tr> <td> <strong>
68:</strong> </td> <td> <table> <col> <col> <col> <tr> <td> <a>
69:</a> </td> </tr> <tr> <td> </td> <td>
70:</td> <td> <div>
71:<br> <br>
72:<br>
73:</div> </td> </tr> </table> </tr> <tr> <td> <strong>
74:</strong> </td> <td> <table> <col> <col> <col> <tr> <td> <a>
75:</a> </td> </tr> <tr> <td> </td> <td>
76:</td> <td> <div>
77:<br> <br>
78:<br>
79:</div> </td> </tr> </table> </tr> <tr> <td> <strong>
80:</strong> </td> <td> <table> <col> <col> <col> <tr> <td> <a>
81:</a> </td> </tr> <tr> <td> </td> <td>
82:</td> <td> <div>
83:<br> <br>
84:<br>
85:</div> </td> </tr> </table> </tr> <tr> <td> <strong>
86:</strong> </td> <td> <table> <col> <col> <col> <tr> <td> <a>
87:</a> </td> </tr> <tr> <td> </td> <td>
88:</td> <td> <div>
89:<br> <br>
90:<br>
91:</div> </td> </tr> </table> </tr> <tr> <td> <strong>
92:</strong> </td> <td> <table> <col> <col> <col> <tr> <td> <a>
93:</a> </td> </tr> <tr> <td> </td> <td>
94:</td> <td> <div>
95:<br> <br>
96:<br>
97:</div> </td> </tr> </table> </tr> <tr> <td> <strong>
98:</strong> </td> <td> <table> <col> <col> <col> <tr> <td> <a>
```

```
99:</a> </td> </tr> <tr> <td> </td> <td>
100:</td> <td> <div>
101:<br> <br>
102:<br>
103:</div> </td> </tr> </table> </tr> <tr> <td> <strong>
104:</strong> </td> <td> <table> <col> <col> <col> <tr> <td> <a>
105:</a> </td> </tr> <tr> <td> </td> <td>
106:</td> <td> <div>
107:<br> <br>
108:<br>
109:</div> </td> </tr> </table> </tr> <tr> <td> <strong>
110:</strong> </td> <td> <table> <col> <col> <col> <tr> <td> <a>
111:</a> </td> </tr> <tr> <td> </td> <td>
112:</td> <td> <div>
113:<br> <br>
114:<br>
115:</div> </td> </tr> </table> </tr> <tr> <td> <strong>
116:</strong> </td> <td> <table> <col> <col> <col> <tr> <td> <a>
117:</a> </td> </tr> <tr> <td> </td> <td>
118:</td> <td> <div>
119:<br> <br>
120:<br>
121:</div> </td> </tr> </table> </tr> <tr> <td> <strong>
122:</strong> </td> <td> <table> <col> <col> <col> <tr> <td> <a>
123:</a> </td> </tr> <tr> <td> </td> <td>
124:</td> <td> <div>
125:<br> <br>
126:<br>
127:</div> </td> </tr> </table> </tr> <tr> <td> <strong>
128:</strong> </td> <td> <table> <col> <col> <col> <tr> <td> <a>
129:</a> </td> </tr> <tr> <td> </td> <td>
130:</td> <td> <div>
131:<br> <br>
132:<br>
133:</div> </td> </tr> </table> </tr> <tr> <td> <strong>
134:</strong> </td> <td> <table> <col> <col> <col> <tr> <td> <a>
135:</a> </td> </tr> <tr> <td> </td> <td>
136:</td> <td> <div>
137:<br> <br>
138:<br>
139:</div> </td> </tr> </table> </tr> <tr> <td> <strong>
140:</strong> </td> <td> <table> <col> <col> <col> <tr> <td> <a>
141:</a> </td> </tr> <tr> <td> </td> <td>
142:</td> <td> <div>
143:<br> <br>
144:<br>
145:</div> </td> </tr> </table> </tr> <tr> <td> <strong>
146:</strong> </td> <td> <table> <col> <col> <col> <tr> <td> <a>
147:</a> </td> </tr> <tr> <td> </td> <td>
148:</td> <td> <div>
```

```
149:<br> <br>
150:<br>
151:</div> </td> </tr> </table> </tr> <tr> <td> <strong>
152:</strong> </td> <td> <table> <col> <col> <col> <tr> <td> <a>
153:</a> </td> </tr> <tr> <td> </td> <td>
154:</td> <td> <div>
155:<br> <br>
156:<br>
157:</div> </td> </tr> </table> </tr> <tr> <td> <strong>
158:</strong> </td> <td> <table> <col> <col> <col> <tr> <td> <a>
159:</a> </td> </tr> <tr> <td> </td> <td>
160:</td> <td> <div>
161:<br> <br>
162:<br>
163:</div> </td> </tr> </table> </tr> </table> <hr> </td> </tr> <tr> <td> <table>
<tr> <td>
164:</td> <td>
165:<strong>
166:</strong>
167:<a>
168:</a>
169:<a>
170:</a>
171:<a>
172:</a>
173:<a>
174:</a>
175:<a>
176:</a>
177:<a>
178:</a>
179:<a>
180:</a>
181:<a>
182:</a>
183:<a>
184:</a>
185:<a>
186:</a>
187:<a>
188:</a> </td> </tr> </table> <br> <br> <div> <br>
189:<br> <a>
190:</a>
191:<a>
192:</a>
193:<a>
194:</a> </div> </td> </tr> </table> </div> </body> </html>
```

Then, below are the distinct HTML tags from the result web page

<table> <input> <title> <html> <!doctype> <form> </head> </title> <col> <tr> <td> <a>   <img> </form> <hr> <body> </body> </table> </html> </td> <link> </tr>   <head>

When the Collection of tag-String and distinct tag-String are processed, the tag-Set is produced. A tag-Set is representation of tag-String in form of the number which shows how many times a tag occurs in the tag-String with reference to the list of distinct tag-String. Below is the list tag-String produced when the list of tag-String and distinct tag-Strings are processed.

```
0: 0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,1,0,0
1: 3,0,0,0,0,0,1,1,0,2,0,3,1,1,0,0,1,0,0,1,0,0,0,1,0,0,0,0,0,0,1
2: 0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
3: 0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
4: 1,2,0,0,0,1,0,0,0,2,0,3,0,0,0,0,1,0,0,0,0,1,0,3,0,2,0,0,0,0,0
5: 0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0
6: 0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,1,1
7: 0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
8: 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0
9: 2,0,0,0,0,0,0,0,0,3,0,3,0,0,0,0,0,1,0,0,0,2,0,2,0,2,0,0,0,1
10: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0
11: 0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
12: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0
13: 0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
14: 0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
15: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0
16: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
17: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0
18: 1,0,0,0,0,0,0,0,0,2,0,2,0,0,1,0,0,0,0,0,0,1,0,2,0,2,0,0,0,0,0
19: 0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0
20: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0
21: 0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
22: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
23: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
24: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
25: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
26: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
27: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
28: 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
29: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
30: 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
31: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
32: 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
33: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
34: 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

```
35: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
36: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
37: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
38: 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
39: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
40: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
41: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
42: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
43: 1,0,0,0,0,0,0,0,0,0,2,0,2,0,0,0,0,0,0,1,0,0,1,0,2,0,2,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1
44: 1,0,0,0,0,0,0,0,0,3,1,1,2,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
45: 0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
46: 0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
47: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
48: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
49: 0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
50: 1,0,0,0,0,0,0,0,0,3,1,1,2,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
51: 0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
52: 0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
53: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
54: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
55: 0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,1,0,1,0,2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
56: 1,0,0,0,0,0,0,0,0,3,1,1,2,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
57: 0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
58: 0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
59: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
60: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
61: 0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,1,0,1,0,2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
62: 1,0,0,0,0,0,0,0,0,3,1,1,2,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
63: 0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
64: 0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
65: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
66: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
67: 0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,1,0,1,0,2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
68: 1,0,0,0,0,0,0,0,0,3,1,1,2,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
69: 0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
70: 0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
71: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
72: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
73: 0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,1,0,1,0,2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
74: 1,0,0,0,0,0,0,0,0,3,1,1,2,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
75: 0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
76: 0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
77: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
78: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
79: 0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,1,0,1,0,2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
80: 1,0,0,0,0,0,0,0,0,3,1,1,2,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
81: 0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
82: 0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
83: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
84: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
85: 0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,1,0,1,0,2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
86: 1,0,0,0,0,0,0,0,0,3,1,1,2,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
87: 0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
88: 0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
89: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
90: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
91: 0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,1,0,1,0,2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
92: 1,0,0,0,0,0,0,0,0,3,1,1,2,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
93: 0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
94: 0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
95: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

```
96: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0
97: 0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,0,1,0,1,0,1,0,1,0,0,0
98: 1,0,0,0,0,0,0,0,0,3,1,1,2,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
99: 0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,1
100: 0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0
101: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0
102: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0
103: 0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,0,1,0,1,0,2,0,1,0,0,0
104: 1,0,0,0,0,0,0,0,0,3,1,1,2,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
105: 0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,1
106: 0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
107: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0
108: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0
109: 0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,1,0,1,0,2,0,1,0,0,0
110: 1,0,0,0,0,0,0,0,0,3,1,1,2,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
111: 0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,1
112: 0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0
113: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0
114: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0
115: 0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,1,0,1,0,2,0,1,0,0,0
116: 1,0,0,0,0,0,0,0,0,3,1,1,2,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
117: 0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,1
118: 0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
119: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0
120: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0
121: 0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,1,0,1,0,2,0,1,0,0,0
122: 1,0,0,0,0,0,0,0,0,3,1,1,2,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
123: 0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,1
124: 0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
125: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0
126: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0
127: 0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,1,0,1,0,2,0,1,0,0,0
128: 1,0,0,0,0,0,0,0,0,3,1,1,2,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
129: 0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,1
130: 0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
131: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0
132: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0
133: 0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,1,0,1,0,2,0,1,0,0,0
134: 1,0,0,0,0,0,0,0,0,3,1,1,2,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
135: 0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,1
136: 0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
137: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0
138: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0
139: 0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,1,0,1,0,2,0,1,0,0,0
140: 1,0,0,0,0,0,0,0,0,3,1,1,2,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
141: 0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,1
142: 0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
143: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0
144: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0
145: 0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,1,0,1,0,2,0,1,0,0,0
146: 1,0,0,0,0,0,0,0,0,3,1,1,2,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
147: 0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,1
148: 0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
149: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0
150: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0
151: 0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,1,0,1,0,2,0,1,0,0,0
152: 1,0,0,0,0,0,0,0,0,3,1,1,2,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
153: 0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,1
154: 0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
155: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0
156: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0
```

```
157: 0,0,0,0,0,0,0,0,0,1,0,1,0,1, ...
158: 1,0,0,0,0,0,0,0,3,1,1,2,0,1, ...
159: 0,0,0,0,0,0,0,0,0,1,0,2,0,0, ...
160: 0,0,0,0,0,0,0,0,0,0,0,1,1,0, ...
161: 0,0,0,0,0,0,0,0,0,0,0,0,0,0, ...
162: 0,0,0,0,0,0,0,0,0,0,0,0,0,0, ...
163: 1,0,0,0,0,0,0,0,0,2,0,2,0,0, ...
164: 0,0,0,0,0,0,0,0,0,0,0,1,0,0, ...
165: 0,0,0,0,0,0,0,0,0,0,0,0,0,0, ...
166: 0,0,0,0,0,0,0,0,0,0,0,1,0,0, ...
167: 0,0,0,0,0,0,0,0,0,0,0,0,0,1, ...
168: 0,0,0,0,0,0,0,0,0,0,0,0,0,0, ...
169: 0,0,0,0,0,0,0,0,0,0,0,0,0,1, ...
170: 0,0,0,0,0,0,0,0,0,0,0,0,0,0, ...
171: 0,0,0,0,0,0,0,0,0,0,0,0,0,1, ...
172: 0,0,0,0,0,0,0,0,0,0,0,0,0,0, ...
173: 0,0,0,0,0,0,0,0,0,0,0,0,0,1, ...
174: 0,0,0,0,0,0,0,0,0,0,0,0,0,0, ...
175: 0,0,0,0,0,0,0,0,0,0,0,0,0,1, ...
176: 0,0,0,0,0,0,0,0,0,0,0,0,0,0, ...
177: 0,0,0,0,0,0,0,0,0,0,0,0,0,1, ...
178: 0,0,0,0,0,0,0,0,0,0,0,0,0,0, ...
179: 0,0,0,0,0,0,0,0,0,0,0,0,0,1, ...
180: 0,0,0,0,0,0,0,0,0,0,0,0,0,0, ...
181: 0,0,0,0,0,0,0,0,0,0,0,0,0,1, ...
182: 0,0,0,0,0,0,0,0,0,0,0,0,0,0, ...
183: 0,0,0,0,0,0,0,0,0,0,0,0,0,1, ...
184: 0,0,0,0,0,0,0,0,0,0,0,0,0,0, ...
185: 0,0,0,0,0,0,0,0,0,0,0,0,0,1, ...
186: 0,0,0,0,0,0,0,0,0,0,0,0,0,0, ...
187: 0,0,0,0,0,0,0,0,0,0,0,0,0,1, ...
188: 0,0,0,0,0,0,0,0,0,0,0,0,1,0, ...
189: 0,0,0,0,0,0,0,0,0,0,0,0,0,1, ...
190: 0,0,0,0,0,0,0,0,0,0,0,0,0,0, ...
191: 0,0,0,0,0,0,0,0,0,0,0,0,0,1, ...
192: 0,0,0,0,0,0,0,0,0,0,0,0,0,0, ...
193: 0,0,0,0,0,0,0,0,0,0,0,0,0,1, ...
194: 0,0,0,0,0,0,0,0,0,0,0,0,0,0, ...
```

It should be noted that, the number of tag-Strings (hence the number tag-Sets) are 194 while those of text-String is 193. When the list of distinct tag-Set is produced the number of distinct tag-Set is found to be less than that of tag-String. This shows that some tag-String have been used more than once. This marks the existence of data clusters. Below is the typical list of distinct tag-Set.

```
0: 0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,1,0,0
1: 3,0,0,0,0,0,1,1,0,2,0,3,1,1,0,0,1,0,0,1,0,0,0,1,0,0,0,0,0,1
2: 0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0
3: 1,2,0,0,0,1,0,0,0,2,0,3,0,0,0,0,1,0,0,0,0,1,0,3,0,2,0,0,0,0
4: 0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0
5: 0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,1,0,1
6: 0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
```

```
7: 0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0
8: 2,0,0,0,0,0,0,0,0,3,0,3,0,0,0,0,0,1,0,0,0,2,0,2,0,2,0,0,0,1
9: [illegible]
10: 0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,[illegible]
11: 0,0,0,0,0,0,0,0,0,0,0,0,[illegible]
12: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,[illegible]
13: 1,0,0,0,0,0,0,0,0,2,0,2,0,0,1,0,0,0,0,0,1,0,2,0,2,0,0,0,0,0
14: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
15: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
16: 1,0,0,0,0,0,0,0,0,2,0,2,0,0,0,0,0,0,1,0,0,1,0,2,0,2,0,1,0,0,1
17: 1,0,0,0,0,0,0,0,0,3,1,1,2,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
18: 0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,1
19: 0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
20: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0
21: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0
22: 0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,1,0,1,0,2,0,1,0,0,0
23: 1,0,0,0,0,0,0,0,0,0,2,0,2,0,0,0,1,0,0,1,0,0,2,0,2,0,0,0,0,0,0
24: 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,1,0,1,0,0,0,3,1
25: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0
26: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,1,1,1,1,0,1,0,0,0,0,1
```

There are 26 distinct tag-Sets out of 194 tag-Sets. When the similar tag-set numbers are grouped together the item-Set is produced. The item set represents the numbers which have the same tag-String. Below is the typical list of item sets for each of the distinct tag-Set.

```
0: 0
1: 1
2: 2 3 14 19 164
3: 4
4: 5
5: 6
6: 7
7: 8
8: 9
9: 10 12 20 165
10: 11 13 21 166
11: 15 17
12: 16
13: 18
14: 22 24 26 28 30 32 34 36 38 40 42 167 169 171 173 175 177 179 181 183 185 187
191 193
15: 23 25 27 29 31 33 35 37 39 41 168 170 172 174 176 178 180 182 184 186 190
192
16: 43
17: 44 50 56 62 68 74 80 86 92 98 104 110 116 122 128 134 140 146 152 158
18: 45 51 57 63 69 75 81 87 93 99 105 111 117 123 129 135 141 147 153 159
19: 46 52 58 64 70 76 82 88 94 100 106 112 118 124 130 136 142 148 154 160
```

```
20: 47 53 59 65 71 77 83 89 95 101 107 113 119 125 131 137 143 149 155 161
21: 48 54 60 66 72 78 84 90 96 102 108 114 120 126 132 138 144 150 156 162
22: 49 55 61 67 73 79 85 91 97 103 109 115 121 127 133 139 145 151 157
23: 163
24: 188
25: 189
26: 194
```

It can be observed that, the tag-Set numbers 0, 1, 4, 5, 6,7,8,9 occur only once. While 2 3 14 19 164 have the same tag-Set. In this form the data cluster can be easily identified by observing the long rows.

The list of item-Set is refined such that, the similar tag-Set numbers are grouped to only those which belongs to the same cluster. Below is the typical semi-refined list of item-Set. In this form the analyser can easily sort out the clusters and hence simplify analysis process.

```
0: 0
1: 1
2: 2 3
3: 4
4: 5
5: 6
6: 7
7: 8
8: 9
9: 10 12
10: 11 13
11: 14
12: 15
13: 16
14: 17
15: 18
16: 19
17: 20
18: 21
19: 22 24 26 28 30 32 34 36 38 40 42
20: 23 25 27 29 31 33 35 37 39 41
21: 43
22: 44 50 56 62 68 74 80 86 92 98 104 110 116 122 128 134 140 146 152 158
23: 45 51 57 63 69 75 81 87 93 99 105 111 117 123 129 135 141 147 153 159
24: 46 52 58 64 70 76 82 88 94 100 106 112 118 124 130 136 142 148 154 160
25: 47 53 59 65 71 77 83 89 95 101 107 113 119 125 131 137 143 149 155 161
26: 48 54 60 66 72 78 84 90 96 102 108 114 120 126 132 138 144 150 156 162
27: 49 55 61 67 73 79 85 91 97 103 109 115 121 127 133 139 145 151 157
```

```
28: 163
29: 164
30: 165
31: 166
32: 167 169 171 173 175 177 179 181 183 185 187
33: 168 170 172 174 176 178 180 182 184 186
34: 188
35: 189
36: 190 192
37: 191 193
38: 194
```

Note that the items in the rows 32 and 33 are separated from those in the rows 19 and 20 regardless that they have similar tag-Set respectively.

The analyser can automatically discover the data cluster by locating the cycles which can be formed when the trail of tag-Set numbers is followed and tracking the row numbers of the list of item-Set. Below is the typical row succession graph produced from the list of the item-Sets (not the refined one). Where the first column is the serial number, the second column is the start row, the third column is the next row tracked when the trail is followed and the fourth column is the count which shows the number of transitions from one row to the next.

```
0:    0   1   1
1:    1   2   1
2:    2   2   1
3:    2   3   1
4:    2  11   1
5:    2   9   2
6:    3   4   1
7:    4   5   1
8:    5   6   1
9:    6   7   1
10:   7   8   1
11:   8   9   1
12:   9  10   4
13:  10   9   1
14:  10   2   1
15:  10  14   2
16:  11  12   1
17:  11  13   1
18:  12  11   1
19:  13   2   1
20:  14  15  21
```

```
21:  14  16   1
22:  14  24   1
23:  14  26   1
24:  15  14  22
25:  16  17   1
26:  17  18  20
27:  18  19  20
28:  19  20  20
29:  20  21  20
30:  21  22  19
31:  21  23   1
32:  22  17  19
33:  23   2   1
34:  24  25   1
35:  25  15   1
36:  26  27   1
```

It can be noted that, the clusters are located on the cycles where the number of transitions is greater than one. Observing the entries 26, 27, 28, 29, 30 and 32 it can be seen that the path row-17 → row-18 → row-19 → row-20 → row-21 → row 22 forms a cycle. This is one of the data cluster identified. This cluster has six nodes and hence 6 fields per record. After the row cluster has been identified, the tag-Set number with these rows are analysed by reading the column entries on the list of item-Set and hence create the page descriptor. A typical item numbers representing the data record is shown below. These are the first items as extracted from the rows which form data cluster

```
[44  45  46  47  48  49]
```

The investigations show that, first field of the first record in the data cluster has tag-Set which is unique. This is because it is associated with other tags which are not part of data cluster. After the item-Set numbers of the data cluster are refined, below is the exact item-Set numbers than represents the data clusters. And hence will be used to locate the data from the list of text String.

```
0:   43   44   45   46   47   48
1:   49   50   51   52   53   54
2:   55   56   57   58   59   60
3:   61   62   63   64   65   66
4:   67   68   69   70   71   72
5:   73   74   75   76   77   78
```

```
6:    79   80   81   82   83   84
7:    85   86   87   88   89   90
8:    91   92   93   94   95   96
9:    97   98   99  100  101  102
10:  103  104  105  106  107  108
11:  109  110  111  112  113  114
12:  115  116  117  118  119  120
13:  121  122  123  124  125  126
14:  127  128  129  130  131  132
15:  133  134  135  136  137  138
16:  139  140  141  142  143  144
17:  145  146  147  148  149  150
18:  151  152  153  154  155  156
19:  157  158  159  160  161  162
```

The page descriptors are

i. Start of data cluster 43

ii. End of data cluster 162

iii. Record size 6.

Appendix B – Investigations

The part of query results as displayed on the web domain www.ibm.com. The search query keyword used was "data extraction".

1 - 10 of 262,953 results

1. eShopmonitor: A comprehensive **data extraction** tool for monitoring ... discuss techniques for **data extraction** from the Web. IEPAD
URL:

2. eShopmonitor: A comprehensive **data extraction** tool for monitoring ... Roadrunner: Towards Automatic **Data Extraction** from Large Web Sites
URL:

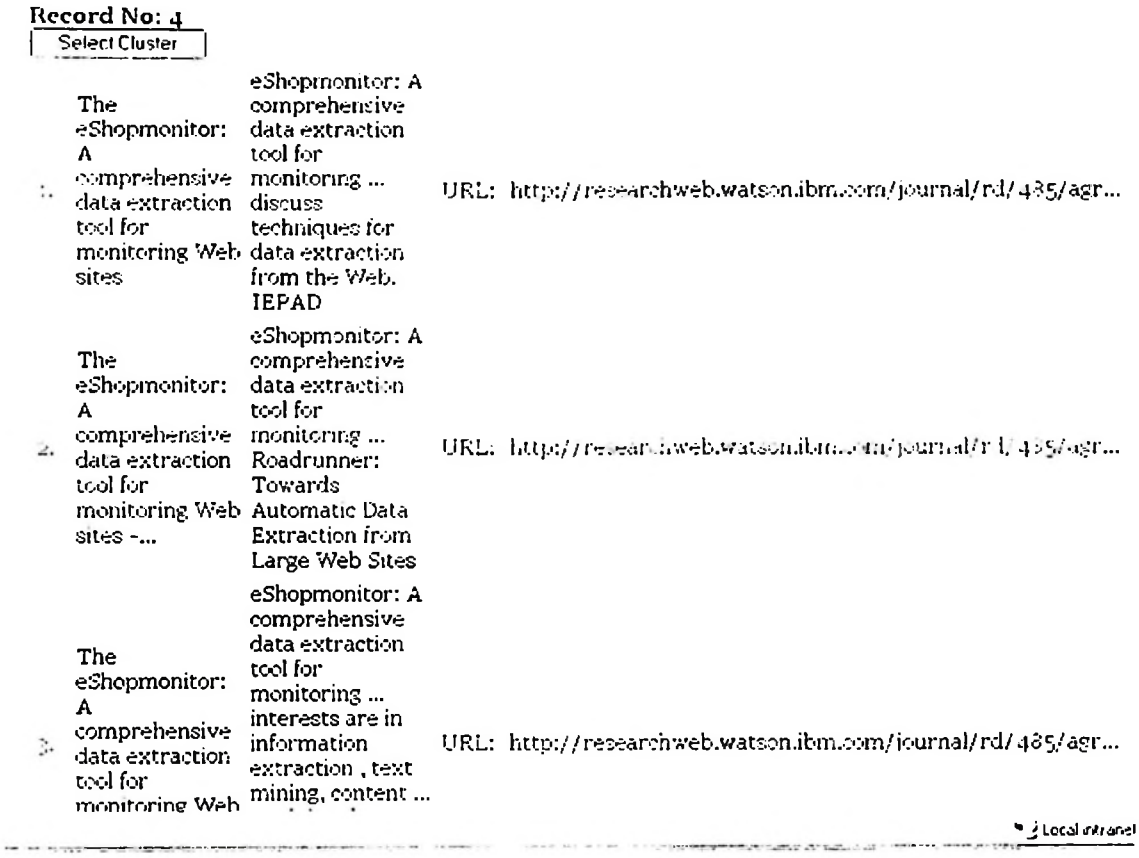
3. eShopmonitor: A comprehensive **data extraction** tool for monitoring ... interests are in information **extraction**, text mining, content ... work involves automatic **data extraction** from semi-structured
URL:

4. 2000 to prepare the **data extraction** process to run automatically ... user. Prepare the **data extraction** process to run automatically
URL:

5. 2000 to prepare the **data extraction** process to run automatically ... 2000 to prepare the **data extraction** process to run automatically
URL:

6. A quick question about **data extraction** Date : Mon, 18 Sep ... for the two sets of **data** prior to AutoGlyph or ... A quick question about **data extraction** Hi Llovd I had one

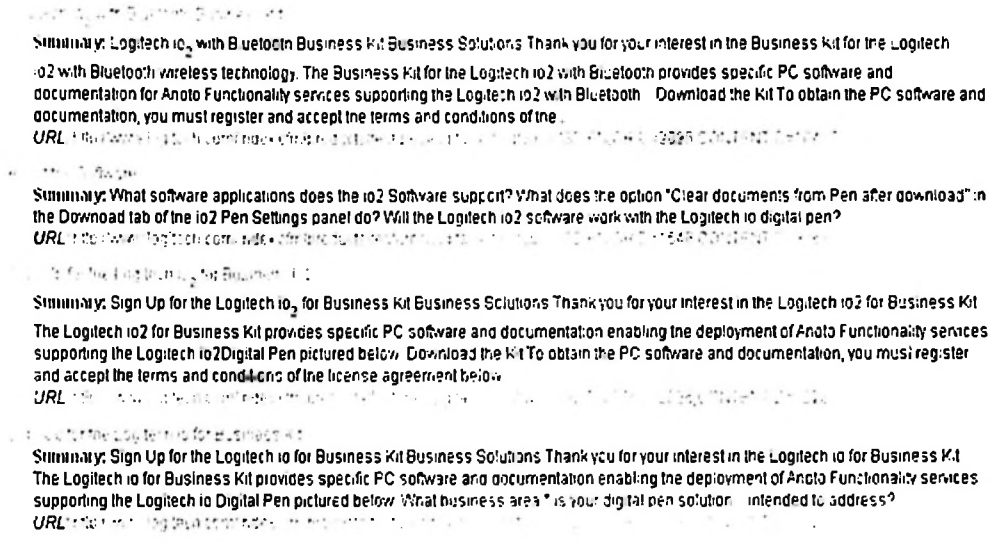
The cluster for the data was not identified as there was bold tag within the data field that was used for putting emphasis on the search keywords. When the bold HTML tag was ignored the good results was produced. Below is the figure that shows the part of the data extract from this web domain.



The other web domain which uses HTML bold tag in the query results are shown below.

1. http://www.google.com
2. http://www.yahoo.com
3. http://www.altavista.com
4. www.exite.com

The figure below is the results from the web domain www.logitech.com which shows that HTML subscript tag that was used in the data field. This has the effect on the structure of the records.



The word io_2 was presented using HTML subscript tag. The analyser treated 2 as a data field which was no the case. After the HTML subscript tag was ignored by the page analyser, the results were good as shown on the figure below after the data cluster have been scrubbed.

Logitech io 2 with Bluetooth Business Kit	Logitech io 2 with Bluetooth Business Kit Business Solutions Thank you for your interest in the Business Kit for the Logitech io2 with Bluetooth wireless technology. The Business Kit for the Logitech io2 with Bluetooth provides specific PC software and documentation for Anoto Functionality services supporting the Logitech io2 with Bluetooth. Download the Kit To obtain the PC software and documentation, you must register and accept the terms and conditions of the ...
About the Software	What software applications does the io2 Software support? What does the option "Clear documents from Pen after download" in the Downoad tab of the io2 Pen Settings panel do? Will the Logitech io2 software work with the Logitech io digital pen?
Sign Up for the Logitech io 2 for Business Kit	Sign Up for the Logitech io 2 for Business Kit Business Solutions Thank you for your interest in the Logitech io2 for Business Kit. The Logitech io2 for Business Kit provides specific PC software and documentation enabling the deployment of Anoto Functionality services supporting the Logitech io2Digital Pen pictured below. Download the Kit To obtain the PC software and documentation, you must register and accept the terms and conditions of the license agreement below.
Sign Up for the Logitech io for Business Kit	Sign Up for the Logitech io for Business Kit Business Solutions Thank you for your interest in the Logitech io for Business Kit. The Logitech io for Business Kit provides specific PC software and documentation enabling the deployment of Anoto Functionality services supporting the Logitech io Digital Pen pictured below. What business area * is your digital pen solution intended to address?

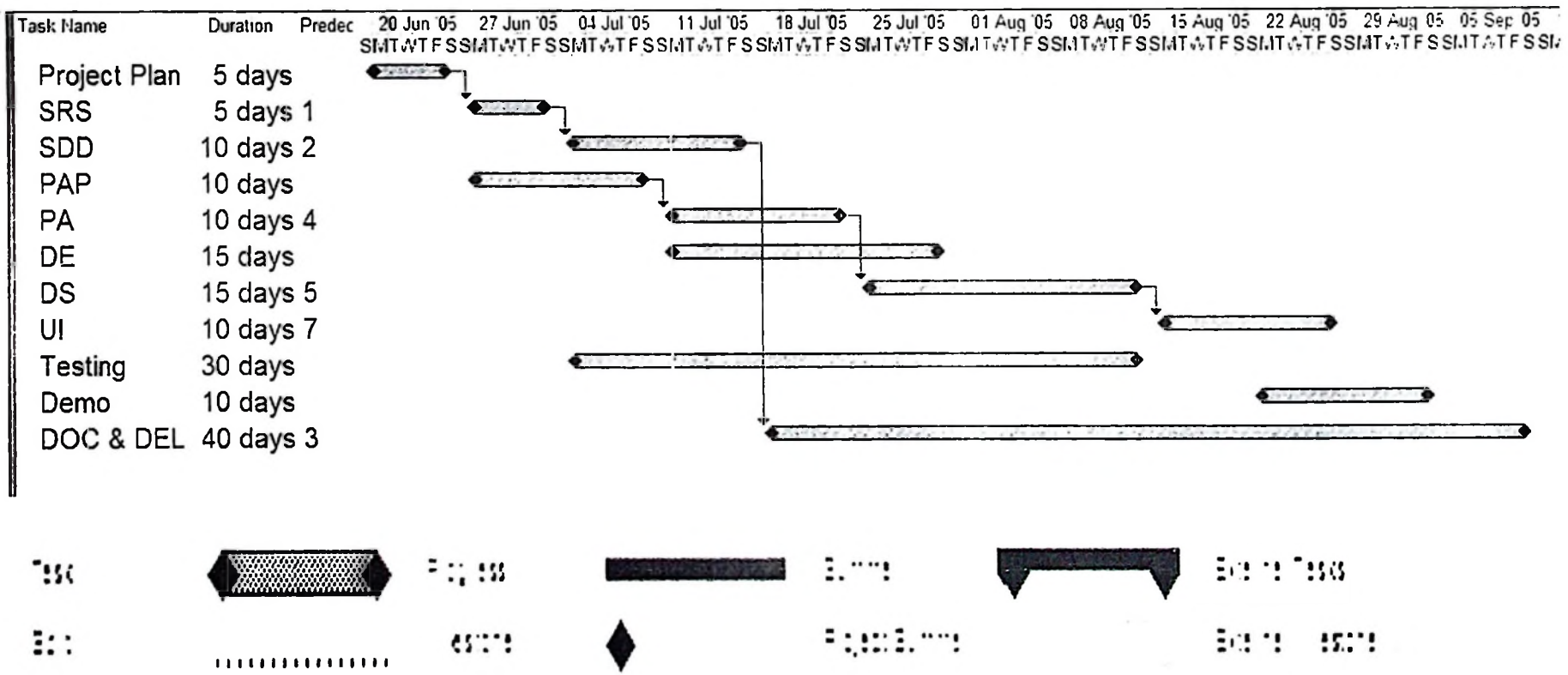
Pages with optional field

The example of the web page with optional field is www.kelkoo.co.uk. Below is the part of the result web page where the data "Merchant rating" is optional field as it occurs only to some records. See the figure below.

When the method of using the trail for sorting the records and align them with template the structure result was produced below is a typical output as produced by the software system.

Appendix C – Project Management

The Gantt chart showing the project schedule and dependencies of tasks



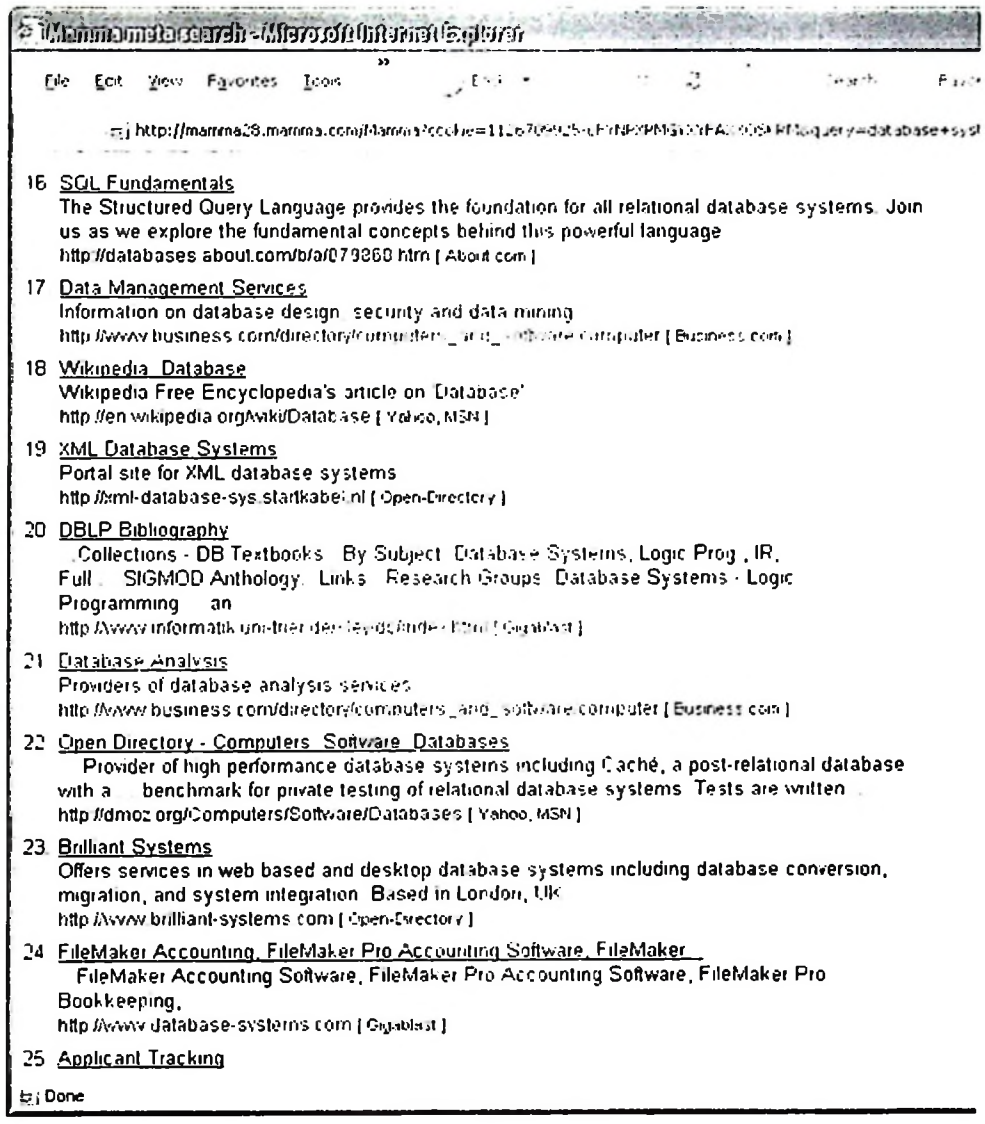
Appendix D – System User Manual

Appendix E – Software Source Code

Appendix F – Software System Testing

This part shows the part of the web page which contains the search result data and the data which has been extracted by the wlcwd system.

1. The typical data extracted from http://www.mamma.com when the search keyword "database systems" was used for data extraction. No tag was ignored but there were some optional fields.



16 SQL Fundamentals
The Structured Query Language provides the foundation for all relational database systems. Join us as we explore the fundamental concepts behind this powerful language
http://databases.about.com/b/a/079860.htm [About.com]

17 Data Management Services
Information on database design, security and data mining
http://www.business.com/directory/computers_and_software/computer [Business.com]

18 Wikipedia: Database
Wikipedia Free Encyclopedia's article on 'Database'
http://en.wikipedia.org/wiki/Database [Yahoo, MSN]

19 XML Database Systems
Portal site for XML database systems
http://xml-database-sys.startkabel.nl [Open-Directory]

20 DBLP Bibliography
... Collections - DB Textbooks - By Subject - Database Systems, Logic Prog., IR, Full ... SIGMOD Anthology. Links - Research Groups - Database Systems - Logic Programming - an
http://www.informatik.uni-trier.de/~ley/db/index.html [GigaBlast]

21 Database Analysis
Providers of database analysis services
http://www.business.com/directory/computers_and_software/computer [Business.com]

22 Open Directory - Computers: Software: Databases
... Provider of high performance database systems including Caché, a post-relational database with a ... benchmark for private testing of relational database systems. Tests are written ...
http://dmoz.org/Computers/Software/Databases [Yahoo, MSN]

23 Brilliant Systems
Offers services in web based and desktop database systems including database conversion, migration, and system integration. Based in London, UK
http://www.brilliant-systems.com [Open-Directory]

24 FileMaker Accounting, FileMaker Pro Accounting Software, FileMaker ...
... FileMaker Accounting Software, FileMaker Pro Accounting Software, FileMaker Pro Bookkeeping, ...
http://www.database-systems.com [GigaBlast]

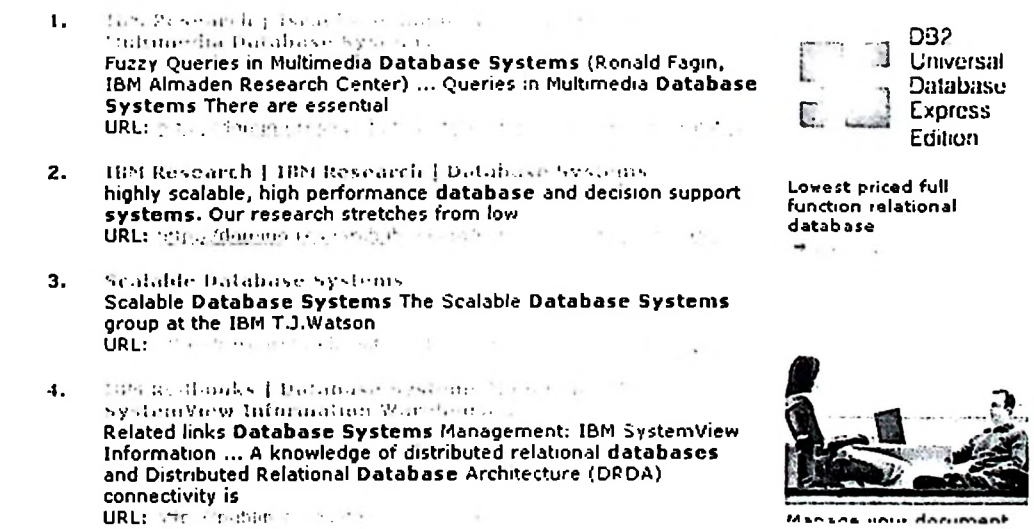
25 Applicant Tracking

Done

The correspond data extracted by the system. Note that other columns have been stripped.

16.	SQL Fundamentals	The Structured Query Language provides the foundation for all relational database systems. Join us as we explore the fundamental concepts behind this powerful language....	[About.com]
17.	Data Management Services	Information on database design, security and data mining.	[Business.com]
18.	Wikipedia: Database	Wikipedia Free Encyclopedia's article on 'Database'	[Yahoo, MSN]
19.	XML Database Systems	Portal site for XML database systems.	[Open-Directory]
20.	DBLP Bibliography	...Collections - DB Textbooks ..By Subject: Database Systems, Logic Prog., IR, ... Full.....SIGMOD Anthology.. Links.. Research Groups: Database Systems - Logic Programming.....an ...	[Gigablast]
21.	Database Analysis	Providers of database analysis services.	[Business.com]
22.	Open Directory - Computers: Software: Databases	... Provider of high performance database systems including Caché, a post-relational database with a ... benchmark for private testing of relational database systems. Tests are written ...	[Yahoo, MSN]
23.	Brilliant Systems	Offers services in web based and desktop database systems including database conversion, migration, and system integration. Based in London, UK.	[Open-Directory]
24.	FileMaker Accounting, FileMaker Pro Accounting Software, FileMaker...	...FileMaker Accounting Software, FileMaker Pro Accounting Software, FileMaker Pro Bookkeeping,..	[Gigablast]
25.	Applicant Tracking	Systems for resume tracking and candidate management.	[Business.com]
26.	Bell Labs - Database Systems	Outlines the lab's major research initiatives, including work on constraint-based and object-based database systems.	[Looksmart]
		Database Administrators! Thousands of quality	

2. The typical data extracted from http://www.ibm.com when the search keyword database systems was used for data extraction. Bold <b> tag was ignored but there were some optional fields. Note that other columns have been stripped.



The corresponding data extracted from this page is shown below

1.	IBM Research | Israel | Seminars | Fuzzy Queries in Multimedia Database Systems	Fuzzy Queries in Multimedia Database Systems [illegible] IBM Almaden Research Center) ... Queries in Multimedia Database Systems There are essentia
2.	IBM Research | IBM Research | Database Systems	highly scalable, high performance database and decision support systems . Our research stretches from low
3.	Scalable Database Systems	Scalable Database Systems The Scalable Database Systems group at the IBM T.J.Watson
4.	IBM Redbooks | Database Systems Management: IBM SystemView Information Warehouse...	Related links Database Systems Management: IBM SystemView Information ... A knowledge of distributed relational databases and Distributed Relational Database Architecture (DRDA) connectivity is
5.	XML programming with SQL/XML and XQuery	stored in relational database systems , and SQL (Structured ... stored in relational database systems must be publishable ... between XML and databases , messaging systems , and Web servers
6.	Usability and design considerations for an autonomic relational database management...	performance. Many systems require complex extra - database operations for data ... today's relational database systems , automatically determining
7.	Database integration with DB2 Relational Connect	DB2 Relational Connect in federated database systems . It describes a design approach using database views for integrating multiple heterogeneous databases together into a single synchronous
8.	Choosing a database management system	development [illegible] actually ask what database [illegible] database . Because database technologies [illegible] important
9.	Usability and design considerations for an autonomic relational database management...	for an autonomic relational database management system - Author ... including autonomic computing, database systems , voice encoding, image
10.	IBM Systems Journal - Vol. 33, No.	large enterprise databases . Seven papers

3. The typical data extracted from http://www.kelkoo.co.uk when the search keyword database systems was used for data extraction. No tag was ignored but there were some optional fields.

Database Systems_A Practical Approach to Design, Implementation and Management — £39.59 [illegible]

Books

[illegible]

Database Systems A Practical Approach to Design, Implementation and Management WITH Success in Your Project: A Guide to Student System Development Projects AND Corporate Computer and Network Security — £111.99 [illegible]

Books

[illegible]

Database Systems_A Practical Approach to Design, Implementation and Management — £43.99 [illegible]

Books

[illegible]

The correspond data extracted by the system. Note that other columns have been stripped

Database Systems: A Practical Approach to Design, Implementation and Management	Author: Connolly, Thomas M.; Begg, Carolyn E.	Delivery Cost: £2.75 - Department:	£39.59	optional	optional	From Computer Books
Database Systems A Practical Approach to Design, Implementation and Management WITH Success in Your Project, a Guide to Student System Development Projects AND Corporate Computer and Network Security	Author: Connolly, Thomas; Weaver, Philip; Panko, Raymond R.	Delivery Cost: Free - Department:	£111.99	optional	optional	From WHSmith
Database Systems: A Practical Approach to Design, Implementation and	Author: Connolly, Thomas M.; Begg, Carolyn E.	Delivery Cost: £3.25 - Department:	£43.99	optional	optional	From The Book Shop at Queens
Database System Concepts	Author: Korth, Henry, Silberschatz, Abraham, Sudarshan, S Sudarshan, S	Delivery Cost: Free - Department:	£39.99	optional	optional	From Blackwells
DATABASE MANAGEMENT SYSTEMS	Author: RAGHU RAMAKRISHNAN	Delivery Cost: £1.25 - Department:	£39.59	optional	optional	From The Book Pl@ce
DATABASE SYSTEM CONCEPTS	Author: HENRY F. KORTH, ABRAHAM SILBERSCHATZ	Delivery Cost: £1.25 - Department:	£31.48	optional	optional	From The Book Pl@ce
Database System Concepts	Author: Korth ; Silberschatz	Delivery Cost: Free - Department:	£41.99	optional	optional	From Play.com - Books
DATABASE MANAGEMENT SYSTEMS	Author: RAMAKRISHNAN, RAGHU	Delivery Cost: £1.50 - Department:	£40.47	optional	optional	From Swotbooks.com

4. The typical data extracted from http://www.tesco.co.uk When the search keyword database systems was used for data extraction

Database and Expert Systems Applications
Author: International Workshop on Database and Expert Systems Applications 199
Format: Paperback
Availability: Special Order, normally available within 4-6 weeks
£75.05
RRP £79.00
Save £3.95 (5%)
add
more info

10th International Workshop on Database and Expert Systems Applications (Dexa '99)
Format: Paperback
Published: 29 November 1999
Availability: Special Order, normally available within 4-6 weeks
£141.55
RRP £149.00
Save £7.45 (5%)
add
more info

2000 Database & Expert Systems Applications /Dexa
Format: Paperback
Availability: Special Order, normally available within 4-6 weeks
£190.00
RRP £200.00
Save £10.00 (5%)
add

7th International Conference on Database Systems for Advanced Applications (Dasfaa 2001)
Format: Paperback
Published: 31 May 2001
Availability: Special Order, normally available within 4-6 weeks
£119.70
RRP £126.00
Save £6.30 (5%)
add
more info

Active Rules in Database Systems
Author: N. Paton
Format: Hardback
Published: 31 January 1999
Availability: Special Order, normally available within 4-6 weeks
£47.50
RRP £50.00
Save £2.50 (5%)
add

The correspond data extracted by the system. Note that other columns have been stripped

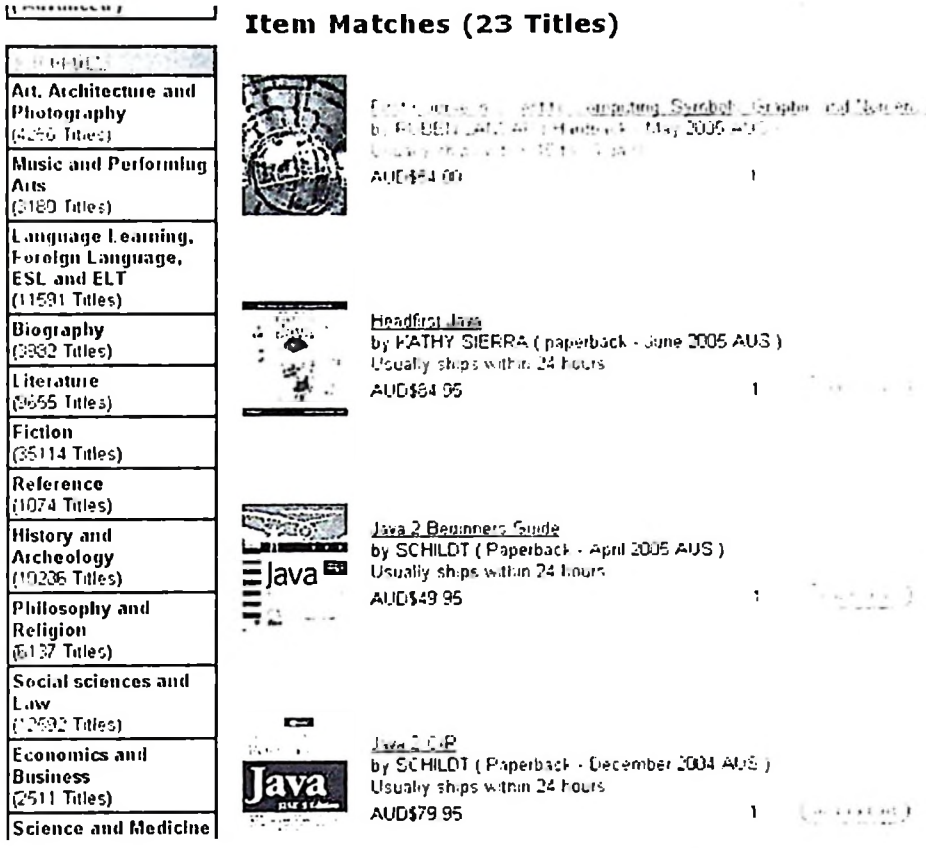
£75.05	RRP £79.00	£3.95 (5%)	Title:	Database and Expert Systems Applications	Author:	on Database and Expert Systems Applications 199
£141.55	RRP £149.00	Save £7.45 (5%)	Title:	10th International Workshop on Database and Expert Systems Applications (Dexa '99)	Format:	Paperback
£190.00	RRP £200.00	Save £10.00 (5%)	Title:	2000 Database & Expert Systems Applications / Dexa	Format:	optional
£119.70	RRP £126.00	Save £6.30 (5%)	Title:	7th International Conference on Database Systems for Advanced Applications (Dasfaa 2001)	Format:	Paperback
£47.50	RRP £50.00	Save £2.50 (5%)	Title:	Active Rules in Database Systems	Author:	N. Paton
£36.58	RRP £38.50	Save £1.92 (5%)	Title:	Active, Real-time, and Temporal Database Systems	Author:	Sten F. Andler
£34.19	RRP £35.99	Save £1.80 (5%)	Title:	An Advanced Course in Databases Systems	Author:	Susan Urban

5. The typical data extracted from http://www.pcworld.co.uk when the search keyword "Toshiba laptop" was used for data extraction. No tag was ignored but there were some optional fields.

The correspond data extracted by the system. Note that other columns have been stripped

EQUIUM M40X-230	INTEL PENTIUM M 750 PROCESSOR 1.86GHz LAPTOP	£999.99
TABLET R10-101	INTEL PENTIUM M 735 PROCESSOR 1.70GHz LAPTOP	£999.99
P30-145	INTEL PENTIUM 4 538 HTT PROCESSOR 3.20GHz LAPTOP	£899.99
M50-164 BUNDLE	INTEL PENTIUM M 740 PROCESSOR 1.73GHz	£799.99
EQUIUM L10-300	INTEL CELERON M 380 PROCESSOR 1.60GHz LAPTOP	£649.99
G20-110	INTEL PENTIUM M 750 QUIET COOL PROCESSOR 1.86GHz LAPTOP	£1,599.99 was £1,699.99 save £100
ENVOY	ENVOY CASUAL BRIEFCASE	£34.99
G20-115	INTEL PENTIUM M 770 QUIET COOL PROCESSOR 2.13GHz LAPTOP	£1,999.99
EQUIUM L10-273	INTEL CELERON M 360 PROCESSOR 1.40GHz LAPTOP	£499.99
EQUIUM M50-164	INTEL PENTIUM M 740 PROCESSOR 1.73GHz LAPTOP	£799.99

6. The typical data extracted from http://www.abbevs.com.au/ when the search keyword "java" was used for data extraction. No tag was ignored but there were some optional fields.



The correspond data extracted by the system. Note that other columns have been stripped

First Course in Scientific Computing: Symbolic Graphic and Numeric Modeling Using Maple Java Mathematica and Fortran90	by RUBEN LANDAU (Hardback - May 2005 AUS)	Usually ships within 15 to 20 days	AUD$84.00
Headfirst Java	by KATHY SIERRA (paperback - June 2005 AUS)	Usually ships within 24 hours	AUD$84.95
Java 2 Beginners Guide	by SCHILDT (Paperback - April 2005 AUS)	Usually ships within 24 hours	AUD$49.95
Java 2 C/R	by SCHILDT (Paperback - December 2004 AUS)	Usually ships within 24 hours	AUD$79.95
Java 2 for Dummies	by BARRY BURD (Paperback - July 2004 AUS)	Usually ships within 7 to 10 days	AUD$49.95
Java Best Practices	by ROBERT ET AL ECKSTEIN (Paperback - March 2003 AUS)	Coming Soon. Pre order now.	AUD$79.95
Java Cookbook	by IAN F DARWIN (Paperback - June 2004 AUS)	Usually ships within 15 to 20 days	AUD$89.95

Appendix G – Glossary

HTML	Stands for HyperText Markup Language
URL	Stands for Universal Resource Locator
Data Store	Means a data warehouse. local cache or database.
DBMS	Database Management System
Text-String	Any string of characters which is not within the angle brackets < and > in the HTML source code.
Tag-String	(In the HTML source code). is any String of characters which proceeds Text-String.
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.
JSP	Java Server pages
XML	eXtended Markup Language
XHTML	Extensible HyperText Markup Language
WLCWD	Warehouse or Local Cache for Web Data
UML	Unified Modelling Language
SQL	Structured Query Language