WEB SPIDER: A SEARCH ENGINE CRAWLER FOR SPECIFIC DOMAIN

By

Ahmad Nasri Alias (7872)

Dissertation submitted in partial fulfillment
of the requirements for the
Bachelor of Technology (Hons)
(Business Information Systems)

JANUARY 2009

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

WEB SPIDER: A SEARCH ENGINE CRAWLER FOR SPECIFIC DOMAIN

by

Ahmad Nasri Bin Alias

A project dissertation submitted to the
Business Information System Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
Bachelor of Technology (Hons)
(Business Information System)

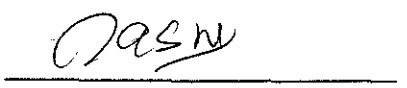
Approved by,

(Ms SHAKIRAH MD TAIB)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
January 2009

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



AHMAD NASRI BIN ALIAS

ABSTRACT

This report presents the idea of enhancing the existing Internet search engine for student's usage based on the research that challenges in the new era of technology and the increasing demand for availability and affordability on Internet in the world. Search engine play a major role at the moment and act as 'must-have' requirement parallel to the growth of virtual world nowadays. Intelligent Internet search engine has entered the research field some 10 years ago and in recent years many intelligent applications have been launched. Web Spider: A Search Engine Crawler for Specifics Domain is a platform that was enhanced from current search engine to allow student to get results based on their preference in an informative and effective way in exploring the advantages of new technologies on the specific domain. The enhancement of the platform is focusing more on ontology-based Web crawler to improve search query and information retrieval by the search engine so that the objective of this project can be achieved. The enhancement also will test the user's acceptance during system testing phase. The improvement that will be made using incremental prototyping method is expected to meet user requirements and enhanced from the existing one. Creating new environment and improving the availability and reliability from current search engine could be beneficial for students as it provide better result in information retrieval process.

ACKNOWLEDGEMENTS

First of all, I would like to bless the Al-Mighty for giving me the strength, patience and guidance during my study period in UTP.

I would like to take this opportunity to express my sincere thanks and appreciation to the following person and others that have directly or indirectly given generous contributions towards the success of this project.

My appreciation and gratitude is extended to my Final Year Project supervisor, Ms. Shakirah Md Taib for all her support, encouragement and guidance throughout the duration of completing this project. Thank you for your advices and comments that always boost my spirit to work harder.

I also would like to thank my family, all my friends and course mates that help a lot and gave valuable advices and tips when I encountered problems during the completion of this project. Without their constant and encouragement and support, I would not have the total concentration on this project.

Lastly, I also like to express my gratefulness and credit to UTP for having such a complete and resourceful library such as Information Resource Center and Online Resources. Without those resources, I would not be able to complete the project in time as much of my literature reviews are based on the sources and information gathered from the library.

Thank you so much for everything.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.0 INTRODUCTION

1.1 Overview

Crawler *a program that searches the Internet in order to create an index of data.*

- Oxford Dictionary Press 1999, 2001 (Tenth Edition)

A lot of effort has been put into developing intelligent search engine which an enhancement from existing search engine that can know what a user want. There were an increasing number of intelligent search engines over the past ten years. New technology could provide a solution for the entire problem that used to occur in virtual world.

The term 'search engine' often used to describe both crawler-based search engines and human-powered directories. The name 'Web Spider: A Search Engine Crawler for Specifics Domain' suggests that it can crawls the Web and user be able to search through what they have found. And if any webpage is changed, crawler-based search engines eventually find the changes. In another word, it is capable in knowing what the user wants, identifying the keyword(s), retrieved feedback from the Web, locating the result and display the outcome to the user.

Research on intelligent search engine has deepened over the past few years as the amount of information available via networks and databases has rapidly increased and continues to increase.

Existing search and retrieval engines provide limited assistance to users in locating the relevant information they need.

Autonomous, enhancement of current search engine may prove to be the needed item in transforming passive search and retrieval engine into active, personal assistants (James Jansen, 1996).

1.2 History

People all around the world including rural area, as long they have Internet connection, are able to use search engine. Search engine act as a complement to Internet and Internet users especially student, use search engine to find materials and information related to their studies. Historically, Internet was developed to improve the military's use of computer technology in 1962. As the Internet grew through the 1980s and early 1990s, many people realized the increasing need to be able to find and organize files and information (Wikipedia, 2009).

Even before the World Wide Web, there were search engines that attempted to organize the Internet. The first of these was the Archie search engine from McGill University in 1990, followed in 1991 by WAIS and Gopher (Wikipedia, 2009). As the Web grew, search engines and Web directories were created to track pages on the Web and allow people to find things. Nowadays, disadvantages of search engine are providing unlimited result to the user due to swift growth of web-based application. Thus, Web Spider created to develop a search engine that can organize the search result in specific domain based on user keywords.

The history lines of the search engine are initiated from very different fields; computer science, science fiction, speculation and philosophy. One of the motivations for enhancement of current search engine comes from the need for searching tools to access the Web as the Web continues its explosive growth. The explosive growth of information is not only occurring on the Web but also with on-line databases.

The number of on-line databases increased from 5000 in 1994 to 5800 in 1996. This number is in addition to the 4600 batch databases that are available via computer networks (James Jansen, 1996).

1.3 Background of Study

The motivation for this study on developing search engine crawler for specifics domain looms from author's interest in the area of crawler-based search engine technology. The main reason is crawler-based search engine for specific domain can perform better than other search engine that crawl over Web into many other domains, that is time consuming and users can be shown irrelevant results that is out of users interest. Instead of using traditional human-powered directories search engine that depends on human for its listings, crawler-based search engine create the listings automatically. Thus, users can now save their time in information searching and retrieval activities.

Basically, the idea of this study came from the need of new technology to cope up with booming of the virtual world; increasing in on-line database, Web and many more. Research was carried out on how existing search engine can be improved to give advantage to the students in Universiti Teknologi PETRONAS (UTP), Tronoh, Perak. The outcome of the research is to develop a search engine that can organize the search result based on user preference on a specific domain. According to Google founders (Larry Page, Sergey Brin, 1996), relationship between websites would produce better ranking of results than existing techniques, which ranked results according to the number of times the search term appeared on a page. Thus, crawler search engine for specific domain could counter the existed problems because the system just crawl over Web pages on the specific domain only and will prevent user from being shown unrelated results.

The integration of the Web Spider: A Search Engine Crawler for Specifics Domain will ease users as it can search right through on specifics domain and organize the search result.

1.4 Problem Statements

1.4.1 Information Abundance

The Internet is very dynamic and has almost uncontrollable growth. Anyone who has a computer, even a small personal computer and a modem telephone line, can access to the Internet. The World Wide Web (Web) is one of the largest publicly available databases of documents, and it is a good testing ground for most retrieval techniques. The Web organizes information by employing a hypertext paradigm. Users can explore information by selecting hypertext links to other information.

James Jansen (1996) claimed that as the Web continues its explosive growth, the need for searching tools to access the Web is increasing. Thus, users might face a problem that they had been shown many unrelated results when they look for materials and information related to their studies if the search engine has to crawl over many links and domains.

1.4.2 Search Engines

Current users also use different search engine that create vary of result compared to other. This will lead to intricate steps as users have to review results by results before go to the relevant result. For example, Google.com results page are different from Yahoo.com results page. So, if the destination page is $Result_1$, user might found it in $Result_2$ in Google.com and $Result_4$ in Yahoo.com. It is practical to review the result manually but the task is ineffective when it's come to hundreds of results.

There is adaptation search engine combining the two search engine like PolyCola.com to make user realize the differences and to search related result promptly. Another search engine; Digg.com uses another method that is displaying result by number of count material searched by user.

1.4.3 Significant of the project

The project main goal is to develop a search engine to cater the need of UTP students to obtain information without any difficulties. It should assist students in information searching and retrieving. This will further avoid students from spending too much time on a particular irrelevant result without getting what they wanted. Apart from that, it may help inexperienced students to reach relevant page without being detoured to any irrelevant page. It should be useful for a lengthy worth of time and expandable through the technology evolvement.

1.5 Objectives and Scope of Study

The main objective of this project is to improve search engine's information retrieval performance from Web based on student's preference on a specifics domain. The end product uses both student's preferences and information content of the document and query. This combination will result in improved information retrieval performance for the students.

The scope of study will be generally about the use of Web Spider in higher education learning and will be narrow down to the use within UTP range that relies on Internet as a medium or searching. The Web Spider will focus on academic domain.

The objectives of the project are:

- To apply ontology-based web crawler technique instead usual crawler-based search engine
- To analyze the explicit knowledge capture through search engine
- To test on users acceptance during system testing

1.5.1 Relevancy of the Project

The idea of this project is to enhance the existing search engine because in the eyes of a search engine, the Web is a body of words on billions of pages, along with the hyperlinks that connect the words.

As a rule, search engine does not understand the words as it is being programmed to match the keywords that are more significant on a page or linked more often from other pages. For example, when user types in "chicken types", he or she sends the search engine for this word, not the animal. As a result, search engine miss the nuance of human language as search engine. Google crawl the pages that include the words "chicken" and "types" and many different results occur such as "chicken recipes types" and "types of chicken hutch" and many more (Stefanie Olsen, 21 August 2006).

CHAPTER 2

LITERATURE REVIEW

2.0 LITERATURE REVIEW

2.1 Search Engine System

Search engine *Computing a program for the retrieval of data, files, or documents from a database or network, especially the Internet.*

- Oxford Dictionary Press 1999, 2001 (Tenth Edition)

There are differences in the ways various search engines work, but they all perform three basic tasks:

- They search the Internet -- or select pieces of the Internet -- based on important words.
- They keep an index of the words they find, and where they find them.
- They allow users to look for words or combinations of words found in that index.

According to Stan Lovic, Meiliu Lu and Du Zhang (2006)

Search engines of today do a great job of sifting through billions of pages of Internet content and returning search results highly relevant to user queries. However, in localized implementations (a local university search or an intranet search of a private company), the same search engine technology usually has less than satisfactory performance. The technology that works well on billions of pages of general content doesn't work well on a much smaller scale of closely related content.

If the search engine has close relationship between websites; indexing websites on specific domain only, the search query would produce better results as the system do not have to crawl over other existed domains.

In the last decade, search engines have improved their performance to the point of becoming a tool of everyday use for most Internet users (Search Engine Watch, 2009).

Currently, search engine leading companies and many smaller players continue to invest in improving performance of their engines and inventing new methods of indexing and searching for content.

Stan Lovic, Meiliu Lu and Du Zhang (2006) point out that localized search technology is still developing and has room for improvement regardless the web search is very saturated. There are many public and private localized searches on the Internet and on corporate intranets, and most of them use the same or similar search engine technology that is used in global web search.

2.1.1 Full Text Search

A very large number of text digital libraries were developed during the last decade. Thus, some technique required to make sure keywords supplied by user return with a relevant result. Full text search refers to a technique for searching a computer-stored document or database.

In a full text search, the search engine examines all of the words in every stored document as it tries to match search words supplied by the user. This means search engine will match all possible results for a search query despite its form or spelling mistakes presence no matter what part of word they will be in. Some Internet search engines; Google and AltaVista, employ full text search techniques, while others index only a portion of the Web pages examines by its indexing system.

Free text searching is likely to retrieve many documents that are not relevant to the intended search question. Such documents are called false positives. Certain clustering techniques based on Bayesian algorithms can help reduce the false positive errors. So if the search term is "chair", these techniques can categorize the document/data universe into "chairman", "plastic chair" etc. Depending on the occurrences of words in a document, it can fall into one of the categories or more. These techniques are being extensively deployed in the e-discovery domain (Wikipedia, 2009).

2.1.1.1 Indexing

It is impossible to search the Web using full text search. Thus, full text search is divided into two tasks; indexing and searching. Indexing is more to a database created by search engine, by scanning whole text in certain Web and updated by successful search words from previous searching, often called an index, but more correctly names a concordance; alphabetical list of all principal words that occur in a text with their immediate context.

Usually the indexer will ignore stop words; the, which are too common and carry too little meaning to be useful for searching like 'the', 'a', 'and' and many more (Wikipedia, 2009).

The indexer also can be preventing from accessing all or part of the websites which might be privately viewable but is necessarily publicly available known as robot exclusion standard or robot.txt protocol and its content is easily checked by anyone with a web browser.

Robot.txt protocol also commonly used to prevent heavy traffic cause by crawler because the crawler send request to $Link_1$ to extracts any available links in the webpage and continue to request next links; $Link_2$, $Link_3$, and so on that exist in the previous page before indexing the pages.

Several major crawlers support a crawl-delay parameter, set to the number of seconds to wait between successive requests to the same server.

2.1.2 Type I and Type II Errors

Type I error, also known as an "error of the first kind", an α error, or a "false positive": the error of rejecting a null hypothesis when it is actually true. Type II error, also known as an "error of the second kind", a β error, or a "false negative": the error of failed to reject a null hypothesis when it is in fact not true. In database searching, documents are assumed to be relevant by default.

Thus, false positives are documents that are rejected by a search despite their relevance to the search question. False Negatives are documents that are retrieved by a search despite their irrelevance to the search question. False negatives are common in full text searching, in which the search algorithm examines all of the text in all of the stored documents and tries to match one or more of the search terms that have been supplied by the user.

Most false positives can be attributed to the deficiencies of natural language, which is often ambiguous: e.g., the term "home" may mean "a person's house" or "the main or top-level page in a Web site" (Wikipedia, 2009).

2.2 Ontology

Most ontology is built by hand. This is a tedious job and its accuracy and maintainability cannot be guaranteed. Because of the great amount and various subjects for the information on Internet, it is of great significance to research how to construct ontology (semi-)automatically. At present, people have developed several ontology learning systems, such as TextToOnto, OntoLearn, the ASIUM system, OntoLT and SOAT, etc.

But some tool systems developed before, e.g. the ASIUM system and the Mo'kWorkbench, have some deficiencies. They often lay particular stress on the exploration and system implementation of some steps in the process of ontology learning.

The Workbench system developed by Mikheev and Finch in University of Edinburgh includes a series of computing tools to explore the inherent structure from natural language texts. Of the current ontology learning systems, the most typical are Text2Onto designed by the AIFB institute in Universität Karlsruhe and the OntoLearn developed by R. Navigli in University of Rome.

Text2Onto has the most powerful function. Its initial version, the ontology learning system TextToOnto developed in 2000, achieved multiple ontology learning algorithms. Based on diversified data sources, it achieved concepts extraction, instance extraction and conceptual relation learning, and it can also prune, construct by sorts, extend and compare ontology. In 2005, TextToOnto was updated to Text2Onto. The new system adds some formal description of ontology representation language, e.g. OWL, F-logic.

Besides, Text2Onto adopts data discovery method driven by data, and it only deals with the changed dataset, which avoids processing afresh the whole corpus. Extractable conceptual relation includes level relation (class and subclass), part-whole relation, synonymy relation and concept instance. The domestic research for ontology learning is still in an initial stage.

At present, some system tools for ontology learning have been built, such as, Cheng Yong et.al developed OntoSphere, which includes corpus analysis, ontology learning, ontology edit and ontology mapping function. OntoSphere is the core component of the knowledge management system KMSphere. The stress of current ontology learning research is the automatic extraction of conceptual terms and conceptual relations, especially extraction algorithms about conceptual relations.

According to the evaluation report of ontology learning system, the accuracy rate of extracting conceptual terms is about 80%~90%. But the accuracy rate of extracting conceptual relations is different given the different types of extracting relations. The rates are usually from 20% - 30% to 70% - 80%. So the tool systems of ontology learning are often integrated into a typical ontology construction tool system, and acquire a draft ontology using ontology learning to help knowledge engineers build ontology. For example, Text2Onto was integrated in the ontology construction tool system OntoEdit and the semantic web suite tool KAON. OntoLT is developed as a plug-in of the ontology construction system Protégé (Ming Xiao, Jinzhu Hu, 2007).

Ontology is used because it is one of the increasingly popular ways to structure information, help users and computers to access information they need, and effectively communicate with each other. It plays diverse roles in developing intelligent systems, for example, knowledge sharing and reusing. By considering domain to be crawled, user expects to spend short time in retrieving really useful information rather than spending plenty of time and ending up with irrelevant information.

In general, current search engines face two fundamental problems. Firstly, the index structures are usually very different from what the user speculates about his problems. Secondly, the classification/clustering mechanisms for data hardly reflect the physical meanings of the domain concepts. These problems stem from a more fundamental problem: lack of semantic understanding of Web documents. New standards for representing website documents, including XML, RDF, DOM, Dublin metatag, and WOM can help cross-reference of Web documents; they alone, however, cannot help the user in any semantic level during the searching of website information. OIL, DAML, DAML+OIL and the concept of ontology stand for a possible rescue to the attribution of information semantics.

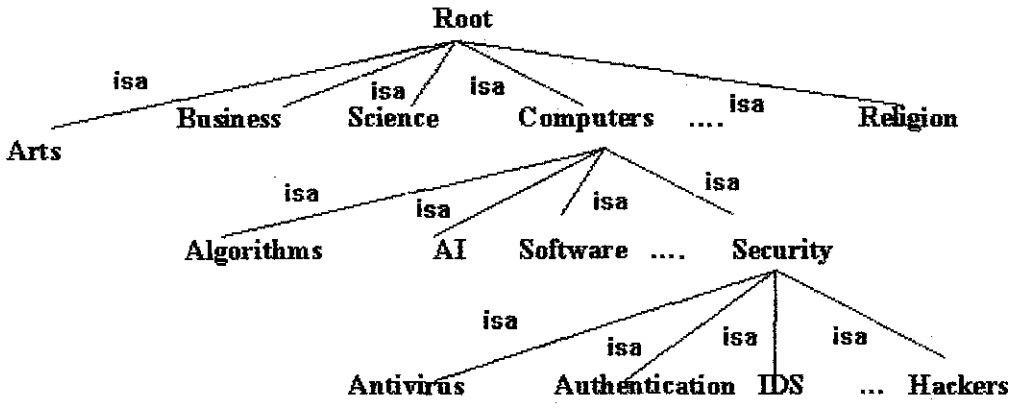


Figure 2.1 Ontology Tree (Source: Ontology-based Web Crawler, S. Ganesh, M. Jayaraj, V. Kalyan, 2004)

Figure 2.1 shows the part of Google domain's ontology taxonomy tree. Google domain is chosen for an example because author is familiar and frequently uses Google as medium of searching. The taxonomy represents relevant Google concepts as classes and their parent-child relationships as *isa* links, which allow inheritance of features from parent classes to child classes.

In the figure, the uppermost node uses various fields to define the semantics of the Google sub domain, each field representing an attribute of "Google", e.g., *arts, business, science, etc. The nodes at the bottom level represent various* Google instances that capture real world data. The arrow line with term "isa" means the instance of relationship.

2.2.1 Artificial Intelligence

Intelligent search engine is an effective tool for solving many bottleneck problems in network information retrieval. It involves acquiring, preprocessing, representing and integrating data and information available at different levels of services (e.g. HTML/XML/RDF/OWL etc) and eventually converts them into useful intelligent semantic information of each domain (Ming Xiao, Jinzhu Hu, 2007).

With the popularization of Internet, the amount and type of web information increase rapidly, which results in more and more difficulties for the users to find useful information. How to search the needed information quickly and effectively is an urgent problem in information retrieval.

But there still exist some bottleneck problems, e.g.: (1) Fuzzy user-expressions. It is difficult for users to express accurately their true meaning simply by keywords or word strings.

Also, for the same concept, different users may use different but similar or relevant keywords to search. (2) The stiff retrieval process and incapability of understanding semantics. The retrieval techniques based on keywords only compare the requirement of users in the form of keywords with the words in full text, but not match search requirements to semantics, which results in including amount of irrelevant information in the output or leaving out much information having the same meaning as the keywords. (3) Information islands. All kinds of connections exist among concepts. When searching a concept, system only treats it as an isolated keyword and ignores connections among concepts. (4) Lack of knowledge in the search outcomes. The related information about the same subject often distributes on multiple websites, but the existent retrieval techniques can only return the list of some URLs, but cannot integrate the relevant information into knowledge to serve the users.

Ontology is the explicit and formal explanation for the concept model. It is the premise to implement the intelligent search. To implement the intelligent search, we need construct a lot of semantics to satisfy the demands. For the mass information on Internet, there are only some ontology's built by hand, e.g. WordNet and Cyc. However, ontology constructed by hand costs much labor and time and contains much less domain concepts. At the same time, how to maintain the existent ontology and update knowledge is still a question, because more and more new concepts, instances, and attributes of old concepts come forth (Ming Xiao, Jinzhu Hu, 2007).

2.2.2 Example of Search Engine That Use Ontology

One example of search engine that use ontology is Simple HTML Ontology Extensions Search Engine (SHOE) that uses XML-like tags and advanced artificial intelligence technology but it is no longer being actively maintained (SHOE).

The basic idea is that all pages indexed by the search engine must use a special set of tags, sort of meta-tags on steroids if you will. These tags provide more than keywords and descriptions, they describe content and relationships. Someone that familiar with XML will aware that more and more web pages are created using content-oriented tags as opposed to the presentation-oriented tags provided by HTML.

However, XML alone cannot solve the search problem because there is no machine understandable meaning associated with ordinary XML tags, for example a furniture store might use the tag <CHAIR> to mean something you sit on while a university might use the same tag to mean a person that heads a department. As a result, traditional search engines will be just as confused with XML as they are now with HTML. The SHOE solution, however, solves this problem by associating a context with a web page; this context can be used to disambiguate terms and provide background knowledge that might help in interpreting content.

Quoted from SHOE's frequently asked questions (FAQs), the admin give an explanation on disadvantage of current search engine and advantages of using ontology.

> Now suppose that you are searching the web for the home pages of a Mr. and Mrs. Cook, whom you met at a computer conference last year. You don't remember their first names, but you *do* recall that both work for an employer associated with the massive ARPA funding initiative 123-4567 (this initiative doesn't really exist, but you get the idea). Now, if you had a *database* with all of the relevant facts stored in it, and a

reasonably decent query language, it'd be pretty easy to construct a query that asks for exactly what you want. Here it is in a pseudo-logic form.

> *Find web pages for all x, y, and z such that*
> *x is a person, y is a person, z is an organization where:*
> *lastName(x,"Cook") and lastName(y,"Cook") and*
> *employee(z,x) and employee(z,y) and*
> *marriedTo(x,y) and involvedIn(z,"ARPA 123-4567")*

So you start the web search. Using an existing man-made web catalog (like Yahoo), you can find ARPA's home page but learn that hundreds of subcontractors and research groups are working on initiative 123-4567. Searching existing web indices (AltaVista, for example) for "Cook" yields thousands of pages about cooking (in fact, AltaVista returns over *200,000* responses). Searching for "ARPA" and "123-4567" provides you with hundreds and hundreds of hits about the popular initiative.

Unfortunately, searching for "Cook" *and* the initiative yields nothing: apparently neither person lists the initiative on his or her web page. Wandering the web on your own is fruitless.

The problem with word indices is that they associate the *syntax* of a word with its *meaning*; there's no way in general for a word index to look at the word "Cook" on a web page and realize that it's about Cook County, or about cooking, or about a person named Cook.

The problem with hand-made web catalogs like Yahoo is that the web is growing so fast, and so much information is out there, that the humans at Yahoo can't possibly keep up.

2.3 Existing intelligent search engine

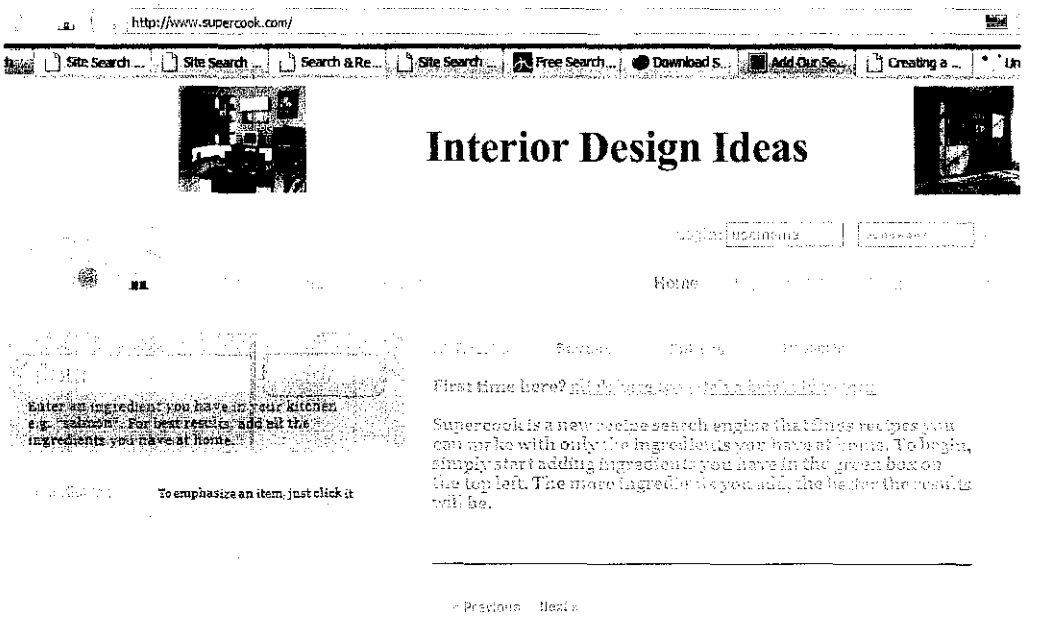


Figure 2.2 Example of existing intelligent search engine

Figure 2.2 shows that example of one of the existing intelligent search engine specifically in cooking field. When user add the word 'flour', the system automatically generate recipes that use flour. The system will narrow down to the most possible recipes that can be use using ingredient that added by user from time to time.

CHAPTER 3

METHODOLOGY / PROJECT WORK

3.0 METHODOLOGY / PROJECT WORK

3.1 Procedure Identification

A methodology is a body of methods used in a particular activities or fields that define the process and order of how the objective is to be achieved (Oxford Dictionary Press, 1999, 2001). Proper planning is the key success for this project. There are six main procedures that are identified in the project development of the incremental prototyping method. Table 3.1 and Figure 3.1 show the methodology that will be use in developing this project.

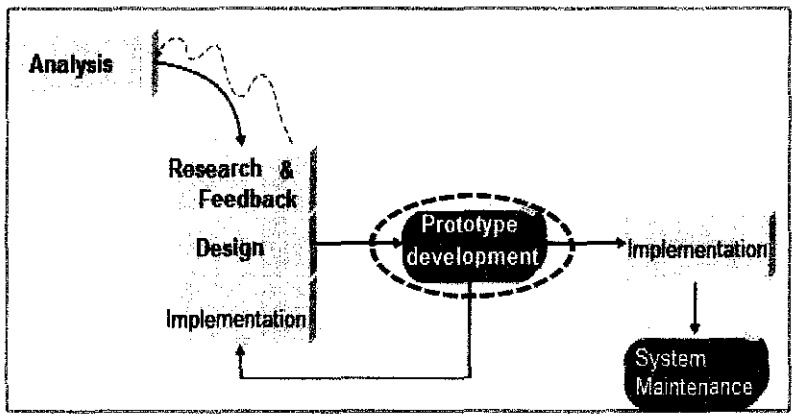


Figure 3.1 Prototyping System Development Life Cycle (Source: System Analysis and Design: An Object-Oriented Approach with UML, John Wiley & Sons, Inc., 2002)

Requirement Analysis and Design	Research and Feedback Analysis	System Design	Construction and Development	Implementation	Maintenance
Objective defined	Research	Proper Design	Develop prototype	Feedback from User	System Release
Scope of project	User Observation	Paper Prototyping	User Testing	Add User Requirement	Enhancement
Functional Requirement			Evaluate Prototype		
Non-Functional Requirement					

Table 3.1 Methodology Used

3.2 Method

3.2.1 Requirement Analysis and Design

During this phase, studies will be conducted extensively to gather information on the requirement and defining the project scope. This includes technical and theoretical aspect of the proposed project.

The requirement analysis is very important as it will be affecting the whole development process. The requirement analysis will be gathered from the supervisor. Objective of the project and the requirement specifications needed are initially defined. Hardware and software requirement are also required to be clarified. In addition, any problem that might occur is recorded so that the developer will be aware of it and therefore assures the efficiency in developing the project. Hence, it reduces the risk of failure during the development phase.

The scope of project is clearly states during this phase. Functional and non-functional requirements are identified in the phase to further explicate the requirements of the proposed project. The functional requirement will be emphasized more towards the technical functionalities needed for the project. Proper classifications are also identified to make sure no redundancy activities will occur. On the other hand, non-functional requirements will accentuate the quality and performance desired.

Functional requirements

1. Internet connection
2. Indexing

Non-functional requirements

1. User friendly interface
2. System is able to display result based on user preference

3.2.2 Research and Feedback Analysis

Research will be conducted in-line with the development of the proposed project. The idea of having concurrent activities is to minimize the time frame of development process and at the same time being able to refer and to revise the content of the research.

The method of research and feedback analysis will be based on findings from the Internet, published journals and questionnaires. The questionnaires are being distributed among people in UTP as the focus group. The questionnaire will be used to gather information in regard to the proposed project.

3.2.3 Design

Proper design is made before constructing the proposed project. The design will facilitate the construction phase by providing the framework for the proposed project. A good design will determine the effectiveness of phase followed to ensure that there will be no redundancy in the following stage.

3.2.4 Construction and Development

This phase is the most crucial phase in making this project and takes up most of the time to put the system into reality. Hardware and software have to be integrated in a way where all systems can communicate effectively and efficiently. The prototype of this project is implemented in author's computer. Nevertheless, it may be implemented in UTP for wider usage.

User testing plays a vital role in the project development. During this phase, the system is being tested before it is delivered to the end-user. Various tests are conducted to make sure the system has fulfilled all requirements.

3.2.5 Implementation

Results from user testing are taken into consideration. Therefore enhancements will be done. Web Spider version 1.0 being releases to be tested to retrieve a feedback from users. Some requirements will be added in order to meet the expectation from the users.

3.2.6 Maintenance

For this phase, the final report will be prepared and end product will be presented to the examiners. Web Spider version 2.0 being release and will be update from time to time to meet the expected future requirements and this phase is the last phase in the methodology for this project.

3.3 Tools

3.3.1 Developer's Specification

Software

The table below shows the list of software that is used for the development of the Web Spider and the usage.

Software	Usage
Mozilla Firefox	Internet browsing
Microsoft Office	Documents, slide presentations, image editing and schedule planning preparation
Macromedia Dreamweaver	HTML Editor for designing, coding, and developing search engine
XAMPP	Local host server for search engine

Table 3.2 Software and the usage

Development and Construction Hardware

Table 3.3 below shows hardware requirements of the computer for the development of Web Spider.

Hardware	Requirement
Operating System	Microsoft Windows XP
Processor	Intel ® Pentium ® M Processor, 1.73GHz
Memory	256MB of memory
Disk Space	1GB of free space
Others	Keyboards, mouse

Table 3.3 Minimum Hardware Requirements

3.3.2 User's specification

Software

- Internet Explorer 6.0 or above or
- Mozilla Firefox 2.0.x or above

Hardware

Table 3.4 below shows hardware requirements of the computer for the development of Intelligent Internet Search Engine.

Hardware	Requirement
Operating System	Microsoft Windows 98 or higher
Processor	Intel ® Pentium ® Celeron or higher
Memory	128MB of memory or higher
Disk Space	2GB of free space or higher
Others	Keyboards, mouse

Table 3.4 User's Hardware Specification

***Note:** The list reflects the minimum requirement for personal computer*

3.4 Project Milestone

			August					September				October				November				December			
ID		Task Name	3	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23	30	2	14	21	28
1	Planning																							
2		Determine The Project Title																						
3		Identify the estimate Time																						
4		Make an Initial Report																						
5																								
6	Analysis																							
7		Gather All the Information Required																						
8		Develop Function Modelling																						
9		Develop Structural Modelling																						
10		Devolop Behavior Modelling																						
11																								
12	Design																							
13		Design Interface																						
14		Design Database																						
15		Design Objects																						
16																								
17	Implementation																							
18		Test the System																						
19		Do modification for the error																						
22																								
23		Time Estimation																						
24		Complete																						
25		Incomplete																						

			January				February				March					April			
ID		Task Name	4	11	18	25	1	8	15	22	1	8	15	22	29	5	12	19	26
1		Planning																	
2		Determine The Project Title																	
3		Identify the estimate Time																	
4		Make an Initial Report																	
5																			
6		Analysis																	
7		Gather All the Information Required																	
8		Develop Function Modelling																	
9		Develop Structural Modelling																	
10		Devolop Behavior Modelling																	
11																			
12		Design																	
13		Design Interface																	
14		Design Database																	
15		Design Objects																	
16																			
17		Implementation																	
18		Test the System																	
19		Do modification for the error																	
		Time Estimation																	
		Complete																	
		Incomplete																	

Figure 3.2 Project Gantt chart

CHAPTER 4

RESULT AND DISCUSSION

4.0 RESULT AND DISCUSSION

4.1 *Results*

The author manage to get a source code from an open source web and in process to modify and integrate the concept of the search engine based on the literature review and research made at present (Ando Saabas, 2007).

As usual search engine, there is a textbox for users and button to trigger the searching. At present, author does not modify anything for the crawler technique but for the future, author will do some modification; to be able to cope with future use *(Refer Figure 4.1)*. User will key in keywords and the search engine will attempt a keyword matching in the indexing page.

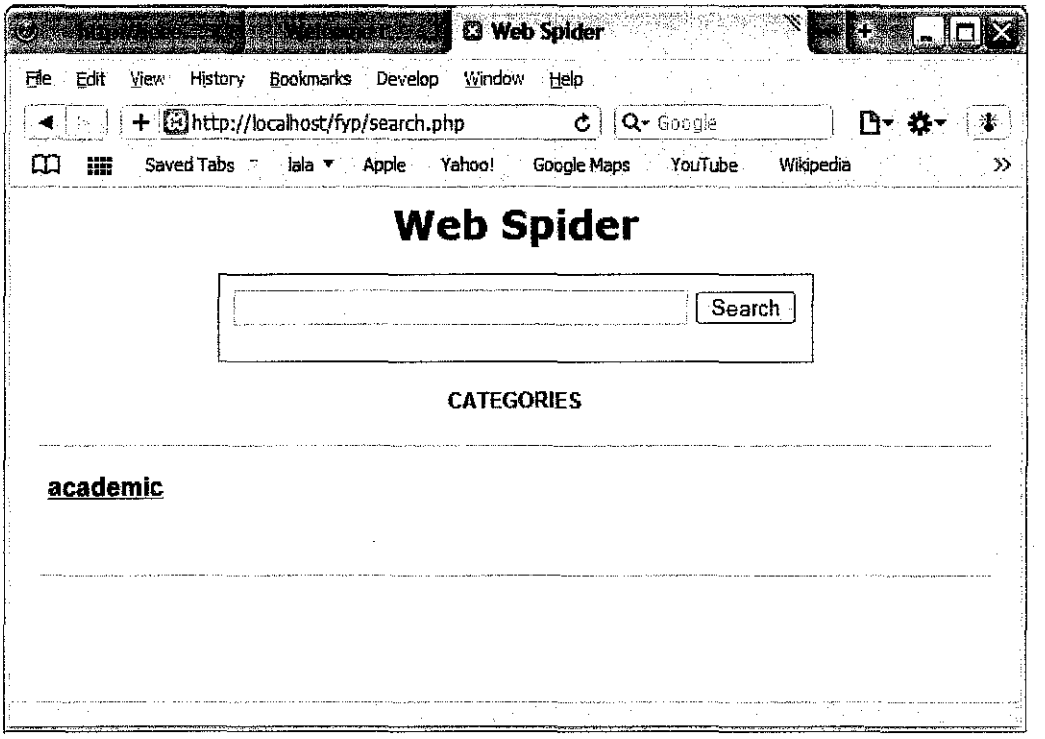


Figure 4.1: Web Spider front page

The search engine will searching and in the same time, indexing before displaying to the user the result that match for the keyword supplied as available in the concordance database. *(Refer Figure 4.2)*.

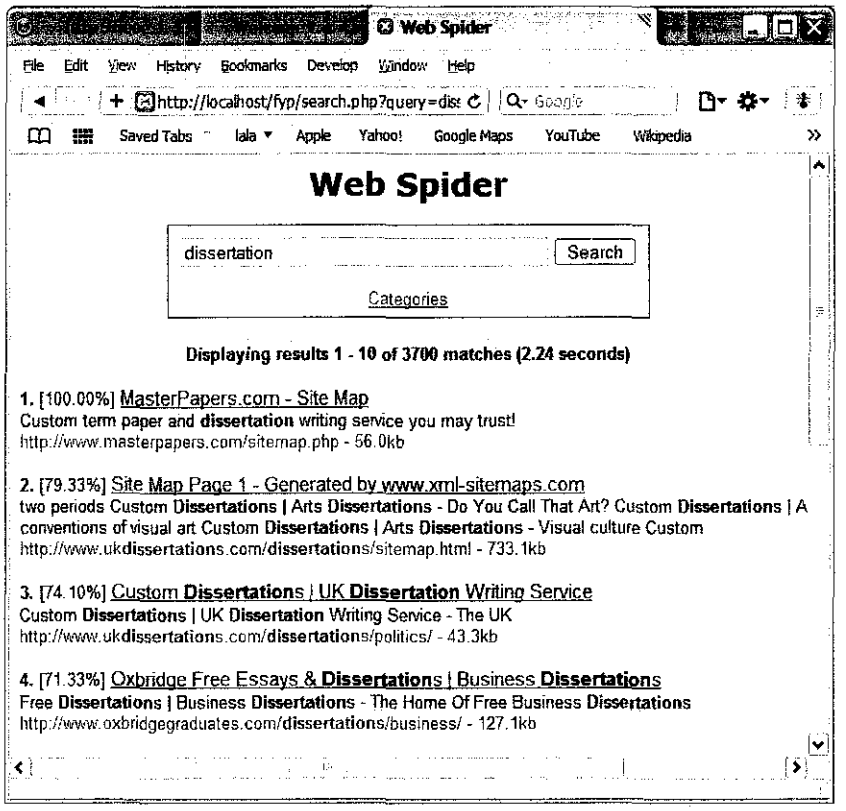


Figure 4.2: Web Spider search result

However, if there is no matching keyword, there will be no result displayed to the user. That might be no exact keyword in the concordance database because of less indexing by the admin. Only administrator can add URL site to be indexed by the search engine to make search result more relevant as spider will crawl from top to the bottom of the URL; including all links and documents, in admin page. A crawler will starts off with the URL for an initial page $Page_0$. It retrieves $Page_0$, extract any URLs in it and add them to a queue of URL to be index. Administrator must login in order to make changes to the search engine to prevent unauthorized changes *(Refer Figure 4.3 and Figure 4.4)*.

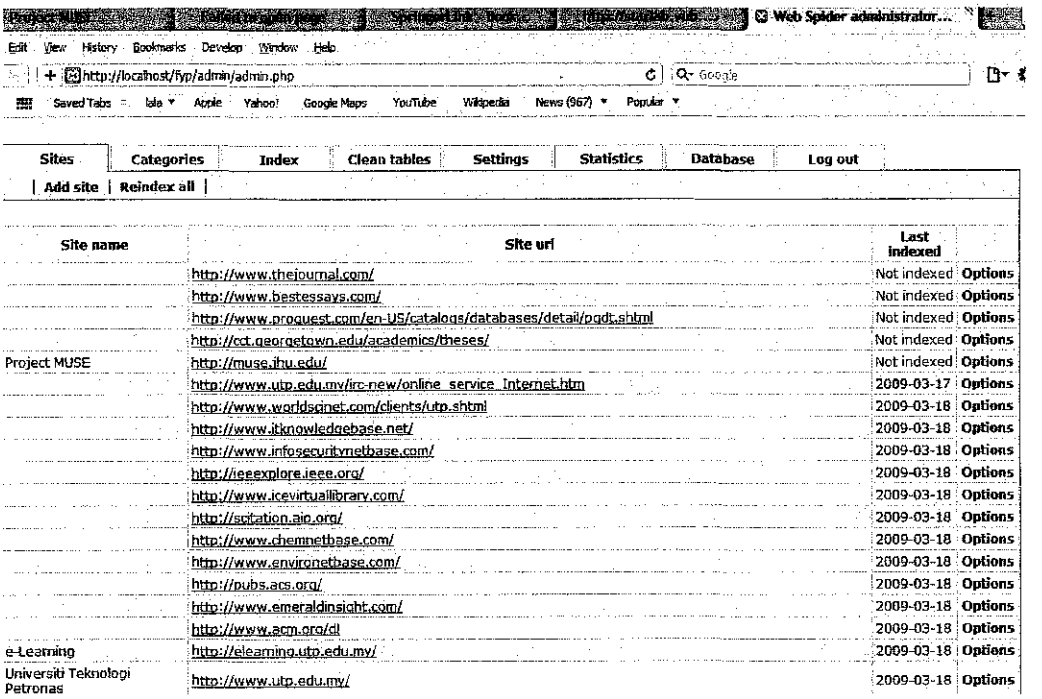


Figure 4.3: Web Spider admin page

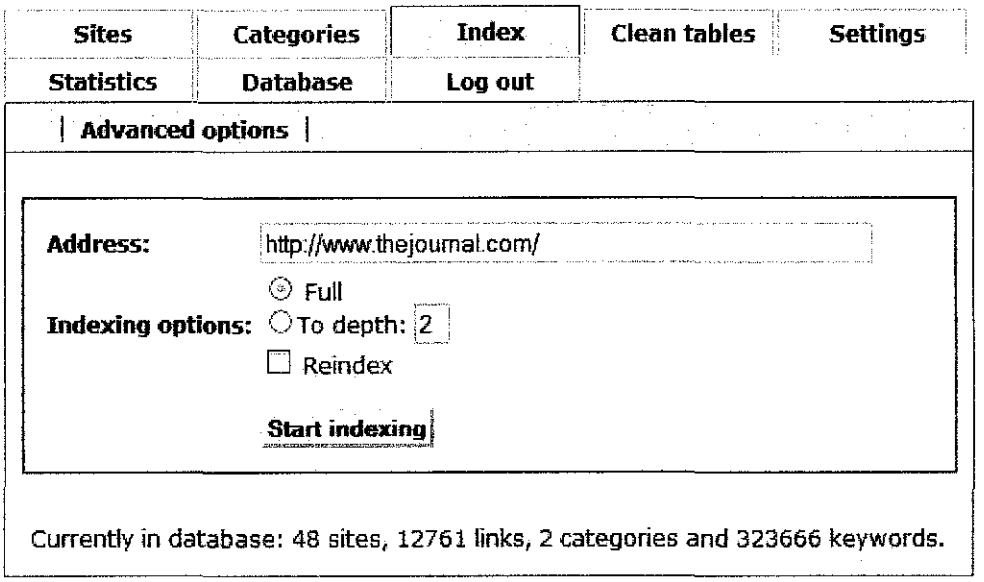


Figure 4.4: Web Spider indexing page

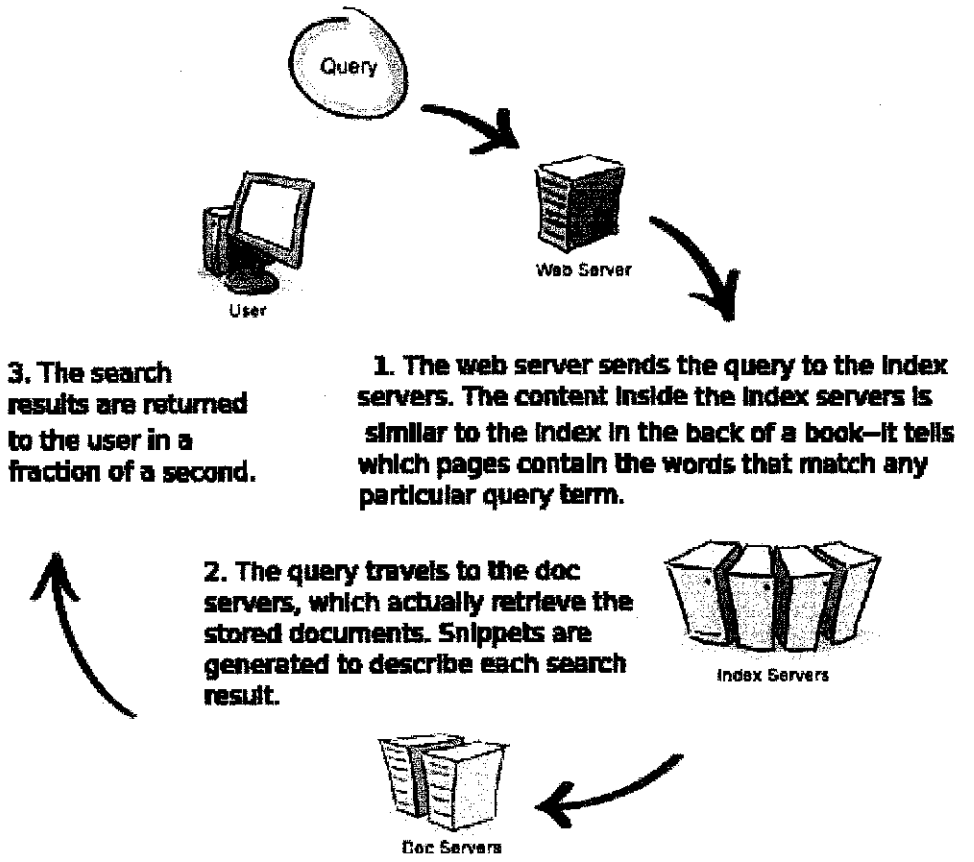


Figure 4.5 Query Diagram (Source: Google Inc. 2003)

Figure 4.5 shows how crawler-based technique works. User will enter keywords in the search box and the system will send the query to the index server; the concordance database. After the query is match by some of the content in the index server, the crawler retrieve the stored URL and a part of the Web page's information being generated to describe each result. Then, the results are displayed to the user.

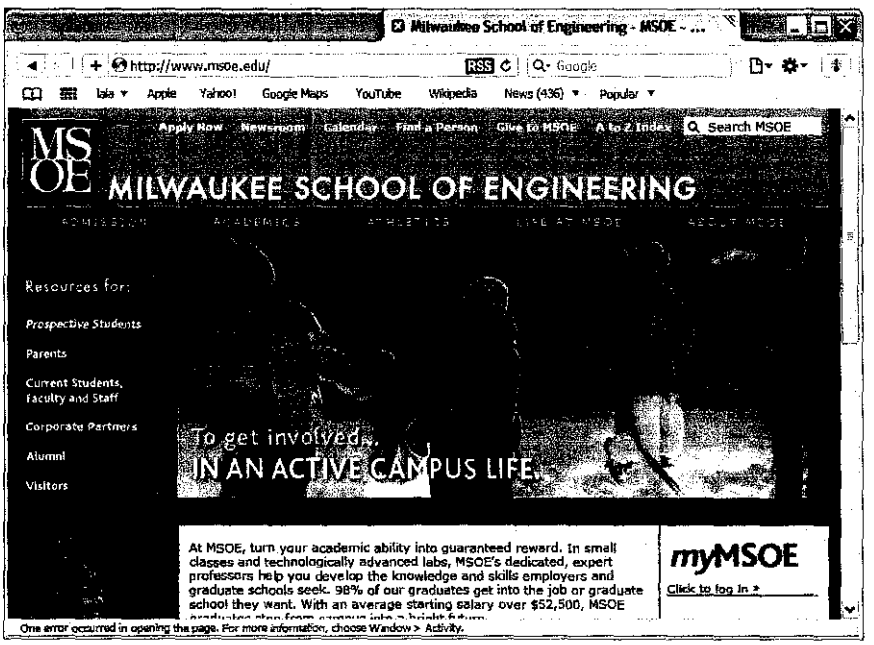


Figure 4.6: Example of Web using crawler search engine

Figure 4.6 shows that one of many examples that used spider search engine. There are also many global search engine companies that already researching about intelligent search engine to counter spamming, and to provide more relevant result to the users.

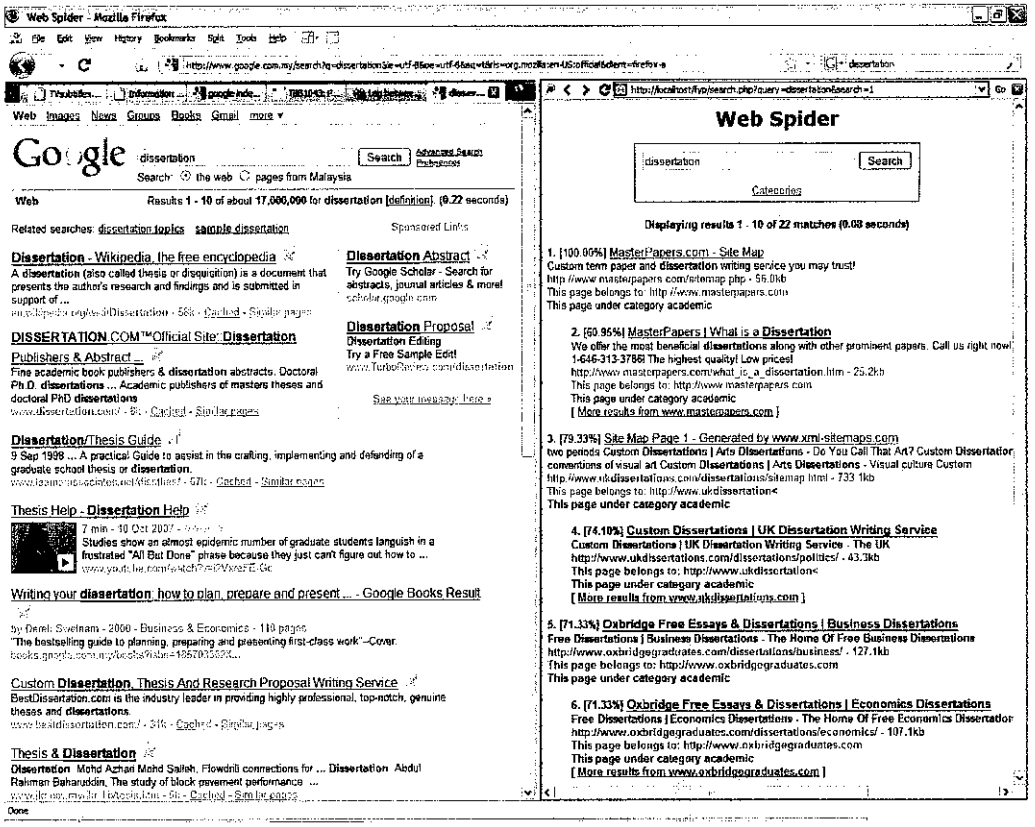


Figure 4.7: Different Google and system's search result

Figure 4.7 shows the different from Google and the system search result. Google displayed about 17,000,000 results while the system only displays 22 results. This lead by a lot of site available in the Google index database, thus student being showed unrelated result because there might only two or three results that relevant to the student's query.

4.2 Discussion

Looking back on some of the results, it shows that Web Spider will help users in retrieving information in a shorter period of time as occurrence of the concordance database on a specifics domain instead using other search engine that going over large volumes of data and domains. The author hopes that the usage of Web Spider in the future will show a positive feedback from the users.

As the system gives opportunities to student that act as alternative to Google and Yahoo, the system is suppose to have less problem exists like Google. For example, Google has a lag time from a time a visit to a new site made by Googlebot about three to five week before the new site being put into index where something will show up in a search (dak, February 2004).

The system takes two to three days to index a new site, depend on the site's depth and number of links occurred in the site instead of Google that take three to five week to new site to be indexed. Thus, it is a disadvantage to Google actually in the side of being too good in the indexing and crawling site field because Google has to index many site; especially new including 5.5 billion spurious pages in June 2006, that came from various domains.

Meanwhile for the system, it will crawl only in the academic domain that will assist student in their studies and the site also key in manually instead to indexing billion of bogus sites that is very tedious job to remove the site from its index as the time taken to manually set the site to be crawled is more efficient compared to time taken to remove the false site from its index.

In April 2006, Google CEO Eric Schmidt told The New York Times that Google's indexing servers are running near capacity (Steve Bryant, 2006).

"Those machines are full. We have a huge machine crisis"

Eric Lander stated that Google updated most on a monthly cycle back in the day and the site take days to complete the indexing process.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

As a conclusion, the author hopes that this project will success and achieve its objective as technology nowadays is fast changing and the Internet has almost uncontrollable growth. Life also getting more complex each day and for that reason, technology which eases the task of people in completing their work is highly demanded.

Thus, implementing Web Spider would be a prudent decision for Internet users especially UTP students because Web Spider really help users in improving the search query and information retrieval from the Web for user's study-related activities. Web Spider will produce less entries of data that will not overwhelms user because domain-specific search engines do help users to narrow down the search scope.

The system also can act as an alternative to Google and Yahoo because Google update their indexing database once a month and create a heavy traffic to the search engine. The system also answer 'why using Web Crawler if I have Google?' question by some of the search engine users.

Looking back at the history of search engines, we know that Google Company was first incorporated as a private company on 1998 and the initial public offering (IPO) on 2004. Now the same question is asked, 'why using Google if you have Yahoo?' because 'Yet another Hierarchical Officious Oracle (Yahoo) create in 1994 and the domain was registered in January 1995 (Wikipedia, 2009).

Thus, it is not wrong to create new search engine to serve Internet users especially student to assist their studies because the Web Crawler only crawl on the specific domain that will create less competition to other well establish search engine like Google and Yahoo.

5.2 Recommendations

The system can be improved in a lot more ways in order to provide better service to users. Some of author recommendations are:

- Improve the user interface so that can attract many users to use the system. Google search engine attracted a loyal following among the growing number of Internet users, who liked its simple design
- 'I Feel Lucky' search function instead only search function; a la Google
- Embedded screenshot thumbnail next to the results to differ the system from other crawler search engines and create a trademark. For example, Google use 'Google' trademark for their search engine
- Embedded in UTP Website for UTP's community use

REFERENCES

James Jansen, 1996, "Using an intelligent agent to enhance search engine performance".Online at <http://www.firstmonday.org/issues/issue2_3/jansen/>.

Wikipedia, 2009, "History of the Internet".
Online at <http://en.wikipedia.org/wiki/History_of_the_Internet>.
"Full Text Search". Online at <http://en.wikipedia.org/Full_text_search>.
"Type I and Type II Errors". Online at <http://en.wikipedia.org/Type_I_and_type_II_errors>.

Stefanie Olsen, 21st August 2006, "Spying an intelligent search engine".
Online at <http://www.builderau.com.au/news/soa/Spying-an-intelligent-search-engine/0,339028227,339269043,00.htm>.

Stan Lovic, Meiliu Lu, Du Zhang, 2006, "Enhancing Search Engine Performance Using Expert System". Online at<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4018553&isnumber=4018443>.

Search Engine Watch, the source for search engine marketing.
Online at <http://searchenginewatch.com/>.

Ming Xiao, Jinzhu Hu, 2007, "A Study on Ontology Learning for the Intelligent Search Engine". Online at <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4341090&isnumber=4339775>.

Ando Saabas, 2007, "Sphider – a PHP spider and search engine". Online at <http://www.sphider.eu/download.php>.

Lefteris Kozanidis, 2008, "An Ontology-Based Focused Crawler". Online at <http://www.springerlink.com/content/16725651q76764n8/>.

Dak, 2004, "Google indexing lag time?". Online at < http://forums.seochat.com/search-engine-spiders-27/google-indexing-lag-time-8511.html>.

Steve Bryant, 2006, "Google, Methinks Thou Doth Search Too Much". Online at < http://googlewatch.eweek.com/content/archive/google_methinks_thou_doth_search_too_much.html>.

John Wiley & Sons Inc, 2002, "Prototyping System Development Life Cycle".

Google Inc, 2003, "Query Diagram".