Proceedings of the 16th Annual NACCQ, Palmerston North New Zealand July, 2003 (eds) Mann, S. and Williamson, A. www.naccq.ac.nz

Developing a Search Engine Design for Research and Development in a Polytechnic Teaching Environment

Dr Jan Pajak
School of Information Technology
Wellington Institute of Technology,
Wellington, NZ
Jan.Pajak@weltec.ac.nz

1. INTRODUCTION

The working thesis of the research project developed in 2002 and described here reads:

"**It is feasible to involve students in research on search engines in a user-paid education system, and this involvement provides academic benefits that outweigh academic costs**".

The goal of this paper is to present a report testing this thesis. To accomplish this goal I (a) explain the design of the search engine under test, and (b) share my findings and conclusions.

2. SEARCH ENGINE

The research project, carried out with the involvement of students, is a search engine. The research objective was to design and build a universal type of search engine, implement it on a server, and research it's behaviour. In order to involve students in this research, the search engine must be characterised by the following attributes:

1.It must have a modular structure of several independent modules, issued to students as separate assignment projects.

2.It must be simple, incorporating only programming features within student subject areas.

3.It must run on the server used by the students involved (my students use a MS WEB SERVER by Dell, "Power Edge 500 SC").

The following design of search engine meets the above criteria. It is composed of 7 modules, taking the form of separate web pages, including a database file. The modules are subdivided into two cascades, namely the search engine cascade, and the search engine submission cascade. The former is composed of four modules: engine.htm, DB_search.asp, web_search.asp, and outcome.htm, and the latter composed of two modules, namely: URL_submit.htm, and URL_confirm.asp. The web pages are linked together into a search engine working system. This linkage can be illustrated graphically as follows – see Figure 1:

Each web page constituting the skeleton of the search engine fulfils the functions outlined below, working together in series. As each completes, it passes control to the next linked web page, or does as the user directs. Below are the functions of subsequent modules (web pages):

- Default.asp. The default/starting page (server-side), running the homepage of the (client-side) search engine named engine.htm.

- **Engine.htm**. A typical search engine main window, with three user input objects, namely "language", "keywords", and "search option". The information from the first two of these input objects is validated (for "not empty", and for "length not less than 2 characters"). A check box for setting the search option is set ON or OFF. The input from these three objects is then submitted to the DB_search.asp page, which takes control after the user clicks the "search" button. The engine.htm also has an option/button "Submit a new URL" (which will display the form/page "URL_submit.htm", previously completed for PR515 assignment). Engine.htm contains a menu, allowing a client to choose and to run any page of the search engine, i.e.: engine.htm, DB_search.asp, web_search.asp, outcome.htm, URL_DB.txt display, or found.txt display.

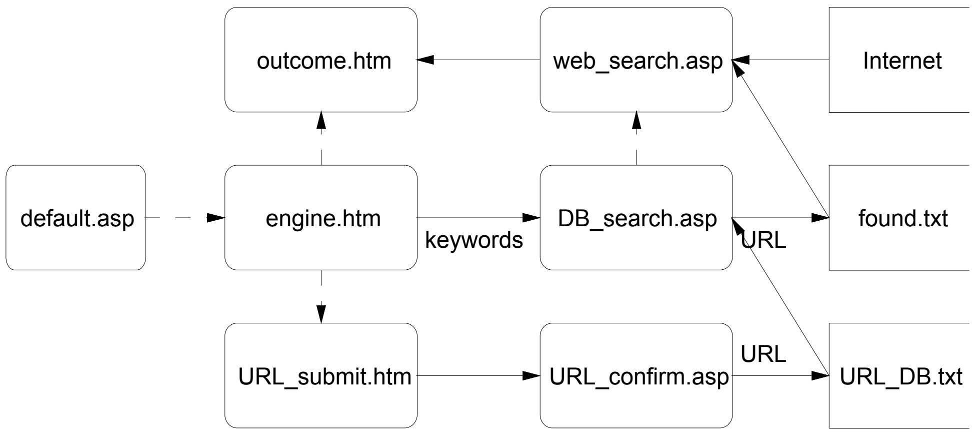

Figure 1. Modular structure of the Internet search engine as discussed, illustrating the basic components of the engine and mutual linking between components.

- **DB_search.asp**. This is a server-side, web page searching, database. In the first stage of this research project, it performs the following functions: (1) it confirms the receipt of information posted by engine.htm, and (2) it runs another web page, web_search.asp. In the second stage, DB_search.asp searches through URL_DB to find web sites containing key words listed by the user. Both stages (first and second), are issued to students as separate assignments.

- **Web_search.asp**. Also a server-side web page for eventually "spidering" (searching) through the Internet. At completion of the first stage of the student project, it performs the following functions: (1) analyses the content of the **found.txt** file, (2) inserts the information, in the order as contained in the found.txt file, into the outcome.htm page (to forward to the user), and (3) runs the outcome.htm page. In the second stage (i.e. for a next assignment) it is now extended with the capability to spider/search through the Internet. Note that in order to forward the required information to the user, web_search.asp needs to generate the outcome.htm page.

- **Outcome.htm**. This is a client-side web page, very similar to engine.htm (actually it should be almost a duplicate of engine.htm). It is however supplemented with the additional information representing the reply of the search engine to user's inquiry. This additional reply information includes a list of items typically provided to clients of search engines (e.g. URLs of web sites containing given key words, samples of web sites' text with these key words highlighted, vital statistics of given web sites, etc.).

Apart from the web pages, the search engine also includes a database file, named **URL_DB**, storing user submissions, and a transitory object named "**found.txt**". For testing purposes this object was presented to students as a disk file, but in operational versions it will be a transitory (hidden) web page.

In addition to this "search engine" cascade of web pages, there is also an "URL submission" cascade. The main function of this cascade is to create, via user submissions, the URL_DB from which the search engine later draws its replies. The "URL submission" cascade is composed of two web pages. The first of these is the **URL_submit.htm** page, which allows users to submit their web sites to this particular search engine. The second is **URL_confirm.asp**, which confirms the completion of the submission procedure. At the initial stage of development, the URL submission cascade does not include web spidering facilities (although these are already included in the search engine itself – see the **web_search.asp** page).

3.PROCEDURE OF SEARCH ENGINE COMPLETION

After the search engine was designed and described in course materials available via two web pages [Pajak, 1999], it was issued to students for completion in two courses on Cyber-Technology (PR515 and PR655). The introductory programming course, PR515 (formerly PR115) was to prepare the URL submission cascade of web pages. The submission cascade was subdivided into two assignments, each of which was to complete a single page over 4 weeks. In fact, during this introductory course, students of PR515 (PR115) familiarised themselves with the problem area of search engines, grasped initial concepts of search engines, and produced two web pages for the final search engine. In the advanced stage, carried out within PR655 (formerly PR255), they completed the remaining four web pages, each also subdivided into two stages/ assignments. During the second semester of the 2002 academic year, several prototypes of this search engine were completed by students.

4.CONCLUSIONS

A wide range of conclusions can be derived by testing the working thesis quoted in the introduction of this paper. A comprehensive discussion of these is provided in the "2003 unofficial report from search engines research" available via [Pajak, 1999] web pages. The essence of the most important conclusions directly relating to this thesis, can be expressed as follows:

(1) **Gains** of involving students into research on search engines. The completion of several search engines by students of my 2002 courses indicate that involving students provides gains of various types. The most important of these are.

(1a) A large **variety of results** were obtained. If the researcher relied solely on his/her own research of search engines as described, he/she would accomplish one outcome only, namely a single search engine – which would not allow for any comparisons. The same project given to numerous students produces several different search engines for later comparison.

(1b) Higher **objectivity** of testing. In the majority of research completed on software, the design and production stages are initial steps leading to further research. This further research usually involves destructive testing, objective assessment, comparisons, etc. But it is difficult psychologically to be destructive (and objective) about software prepared by oneself – especially when it comes to revealing disadvantages of this software. It is therefore more objective to assess software produced by students.

(1c) The capability of **raising standards**. If we repeat the same research project with the next group of students, making the best outcomes from previous groups available to them, then each such subsequent repetition must raise standards of quality.

(1d) **Universal character**. This research indicates that all projects involving students, independently from their topic, have the same process and require the same preparation procedure. This in practice means that experience gathered during the completion of search engines as described here, for example, is valid for numerous other computing research projects. If therefore someone completes one such research project, this completion provides him/ her with the experience to attempt further such projects. This also provides examples for anyone wishing to apply a similar approach to different software projects. The repetitiveness of this approach is also facilitated by the Internet nature of this project on search engines. For example; all materials concerning this research project, all handouts, all assessments and even samples of outcomes, are available through web sites listed below in the reference section of this paper. Thus anyone wishing to explore this approach may start by repeating this path and drawing from materials available on the web sites [Pajak, 1999]. It also means that such research could be extended into a student-based inter-institutional research project.

(2) **Disadvantages** of involving students into research. The experience gathered during this project reveals that involving students into research under user-paid conditions also has disadvantages. The most important out of these are:

(2a) The **labour** required. The preparation of research support for students to carry out such projects takes much more time than would doing this research ourselves. For example, every key problem to complete such a research project must be checked, solved, and described in course materials prior to issuing it to students.

(2b) **Instability** of teaching load. This manifests itself in the lack of certainty a given staff member will teach the same subject again. This instability actually discourages any long-term planning of research, and also raises psychological difficulties with investing our time in involving students in research.

Despite these advantages and disadvantages being of somewhat different nature, it still seems empirically sure that the advantages outweigh the disadvantages. This in turn allows us to draw a final conclusion from this research project that **the working thesis stated in introduction to this paper is validated**. Thus, despite the many unfavourable circumstances academic staff members face in user-paid educational systems, it is possible to carry out successful research with the participation of students in such a manner that the academic gains of this research outweigh the academic losses.

5. REFERENCES

Pajak J. (1988) "Concept of a user-tuned information system for natural language processing", ANZAME Conference, The University of Western Australia, Perth, 29 November 1988 - 2 December 1988; Abstracts, page 73. (Text of this paper is available from within web sites: <http://pajak.20m.com/search_engine.htm> or <http://pajak.20fr.com/search_engine.htm>.)

Pajak J. (1999) Unofficial lecture notes, and research publications. Web sites <http://pajak.20m.com/search_engine.htm> and <http://pajak.20fr.com/search_engine.htm>. Accessed April 2003.