Chapter 4 Prototype System Development

In this chapter, we will implement the benchmark model designed for this research on a prototype platform. This prototype illustrates the feasibility and validity of the model of this research. The architecture and design of the prototype is described in the following sections.

4.1 Prototype Development Tool

For this research, the prototype is implemented on a benchmark experiment. The prototype system uses a client/server structure, as shown in Figure 4.1. The client-end interface to operate the workload generator prototype is a simple Web Browser. We used Microsoft Internet Explorer 6.0 SP1 as the Web Browser, and Microsoft Internet Information Services 6.0 as the Web server. The database server we adopted is Microsoft SQL Server 2005. Table 4.1 illustrates the developmental environment of this prototype.

We select the Web Search service of the Yahoo! Web Search APIs to integrate with the prototype system. Web Search APIs allow the application program to search web pages before submitting queries. The Yahoo! Search Web Services are all REST services. That means we need to reconstruct the request URLs into prototype code and give different parameter values to the web searches, after which the search results will be returned. We use the GET method to construct a request URL to be sent to the Yahoo! Web Search APIs. Based on the parameter selection, we specify the corresponding request parameters and generate the search script. Table 4.2 shows the request method of web search services. Table 4.3 shows the request parameters.

Table 4.1: The Description of Prototype Development Tool

Operating System	Windows 2003 Server
Web Server	Microsoft Internet Information Services 6.0
Database Server	Microsoft SQL Server 2005
Client	Microsoft Internet Explorer 6.0 SP1
Application Program Language	ASP.NET 2.0, JavaScript
Web Search Service API	Yahoo! Web Search APIs

Table 4.2: The Request Method and URL of Web Search Services

Request Method	GET
Request URL	http://search.yahooapis.com/WebSearchService/V1/webSearch

Table 4.3: The Request Parameters of Web Search Services

Parameter	Value	Description
appid	An Application ID is a string that uniquely identifies your application. Think of it as a User-Agent string. If you have multiple applications, you must use a different ID for each one.	The application ID.
query	• hostname: use to find all documents from a particular host only. Example: hostname:autos.yahoo.co • link: use to find documents that link to a particular url. Example: link:http://autos.yahoo.com/ • url: use to find a specific document in our index. Example: url:http://edit.autos.yahoo.com/repair/tree/0.html • inurl: use to find a specific keyword as part of indexed urls. Example: inurl:bulgarian • intitle: use to find a specific keyword as part of the indexed titles. Example: intitle:Bulgarian	The query to search for (UTF-8 encoded). This query supports the full search language of Yahoo! Search, including meta keywords.
type	all (default), any or phrase	The kind of search to submit: •all returns results with all query terms. •any returns results with one or more of the query terms. •phrase returns results containing the query terms as a phrase.
results	integer: default 10, max 100	The number of results to return.
similar_ok	no value (default) or 1	Specifies whether to allow multiple

		results with similar content. Enter a 1 to allow similar content.
site	string: default no value	A domain to restrict your searches to (e.g. www.yahoo.com). You may submit up to 30 values (site=www.yahoo.com&site=www.cnn.com).

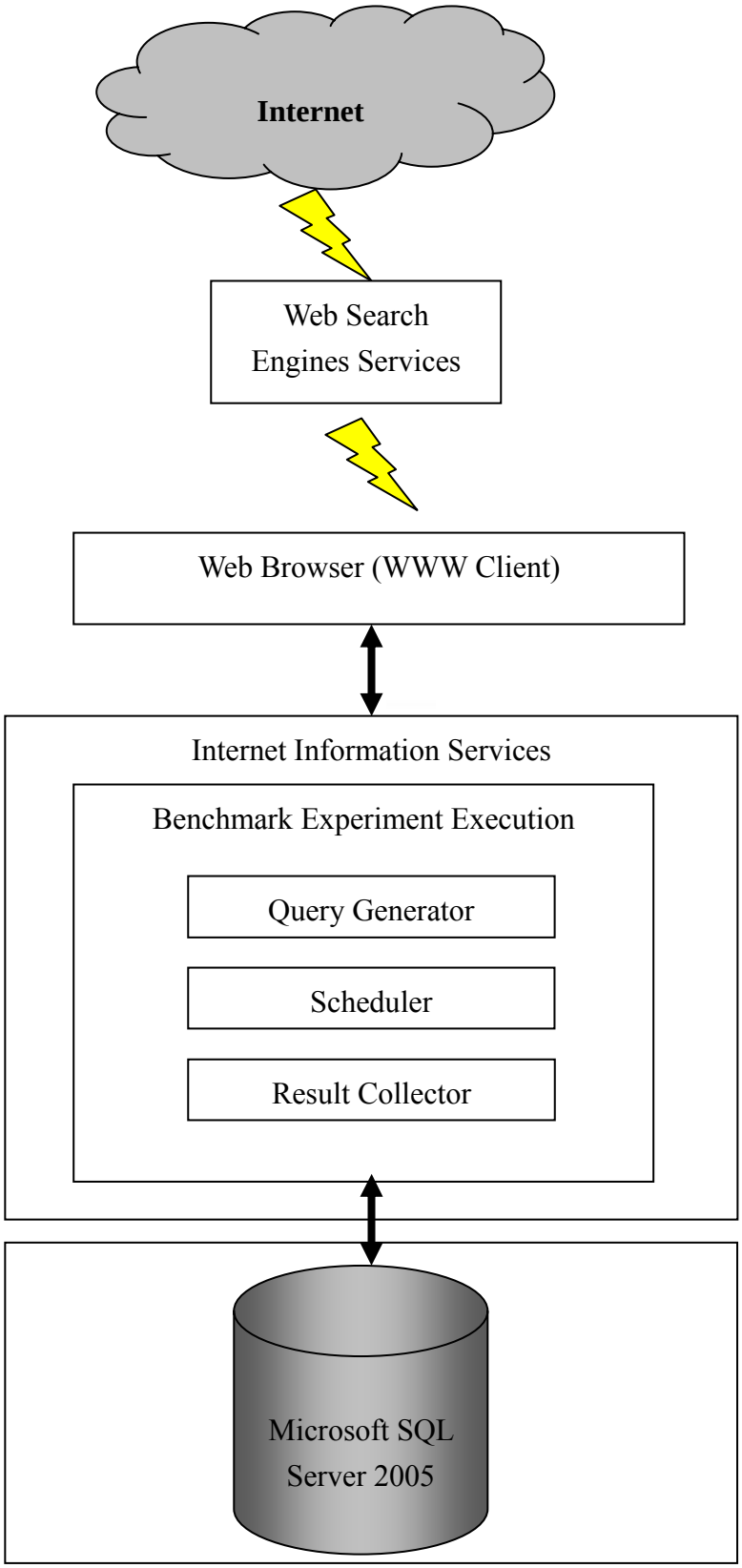

Figure 4.1: The Prototype System Structure

4.2 Prototype System Implementation

The research model must be followed when implementing the prototype system. Therefore, the prototype has several primary engines including the query generator, the scheduler, and the result collector.

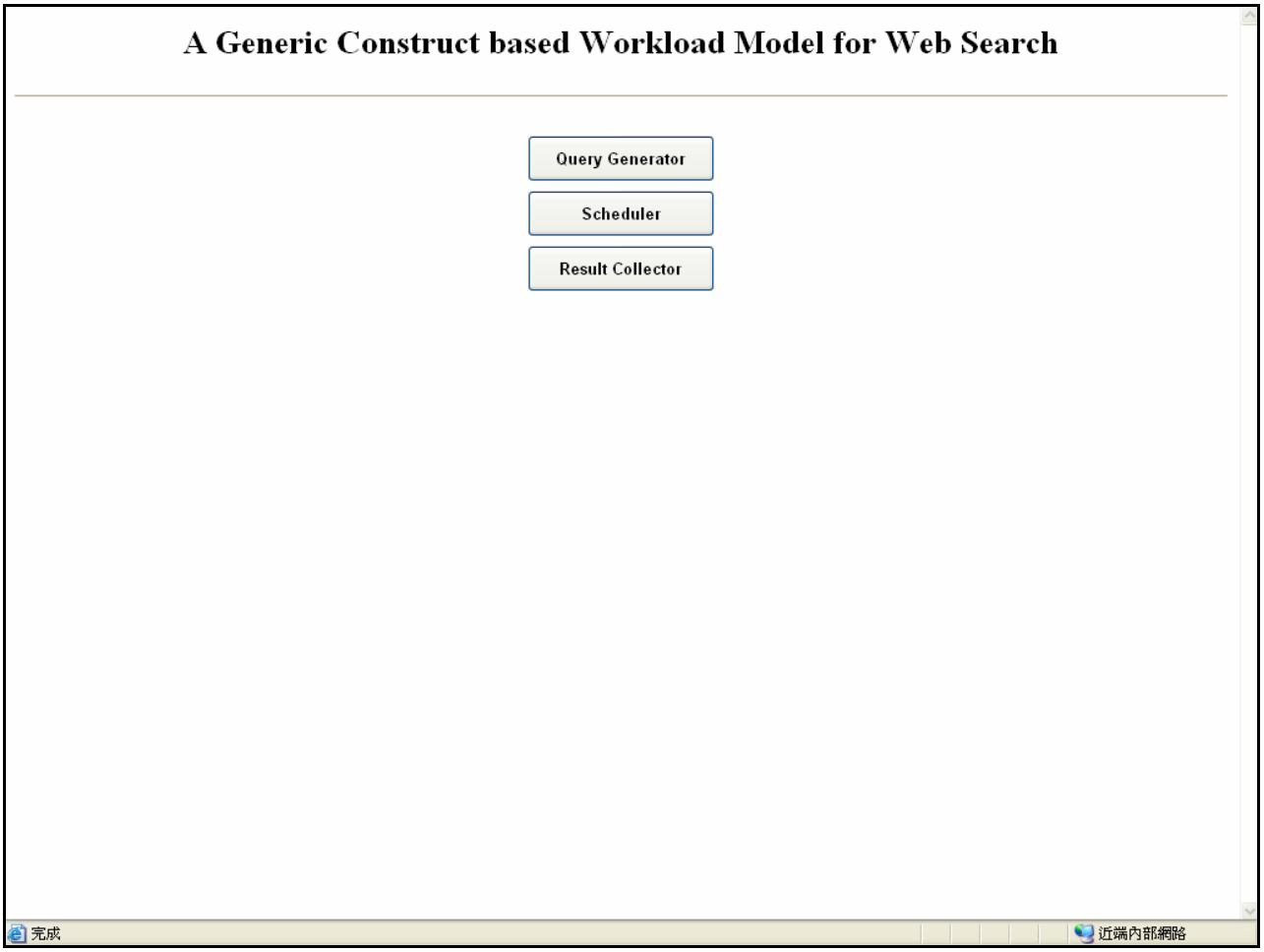

Figure 4.2: The Main Menu of Workload Model

4.2.1 Query Generator

In the query generator (Figure 4.3), there are two parts: page model selection and query model selection. In the page model selection, users are allowed to choose the selection of the page model. In the query model selection, users are allowed to choose the selection of the query model. After finishing the selections of the page model and the query model, to submit the generate script button to generate the query script. We will detail the page model selection and query model selection in the following section.

4.2.1.1 Page Model Selection

According to page model, there are two parts of pages model selections: single-page option and multi-page option. For single page search, users are allowed to select the location of page where the query term will be searched. For multi-page search, users are allowed to specify authoritative pages/websites to perform a search, such as portals, educations and governments. We collect some web pages of these websites as examples for selections (see Figure 4.3).

4.2.1.2 Query Model Selection

According to query model, query model selections composed of query type, link structure, similarity and synonym. Users are allowed to select which kind of query to perform in the selections of query type. And users are allowed to select the ranking method in the options of link structure; users are allowed to select similarity method in the options of similarity. Also, in the options of synonym, users are allowed to have multiple selections. When users have selected query options, they must choose corresponding page options to match the principles of query in this prototype. After users finish their requirements, query generator consolidates the query options to generate query scripts (see Figure 4.4).

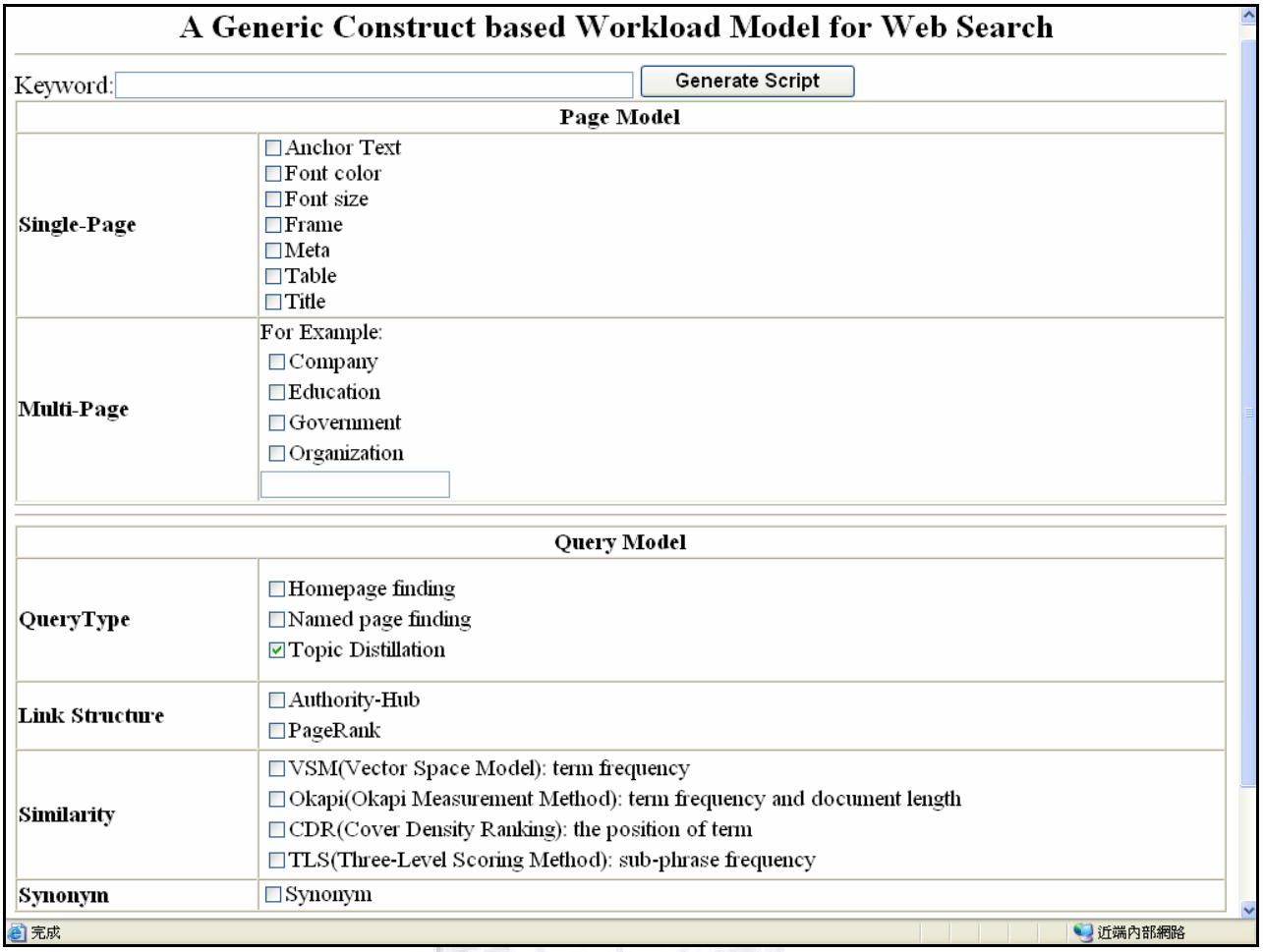

Figure 4.3: Page Model Selection and Query Model Selection

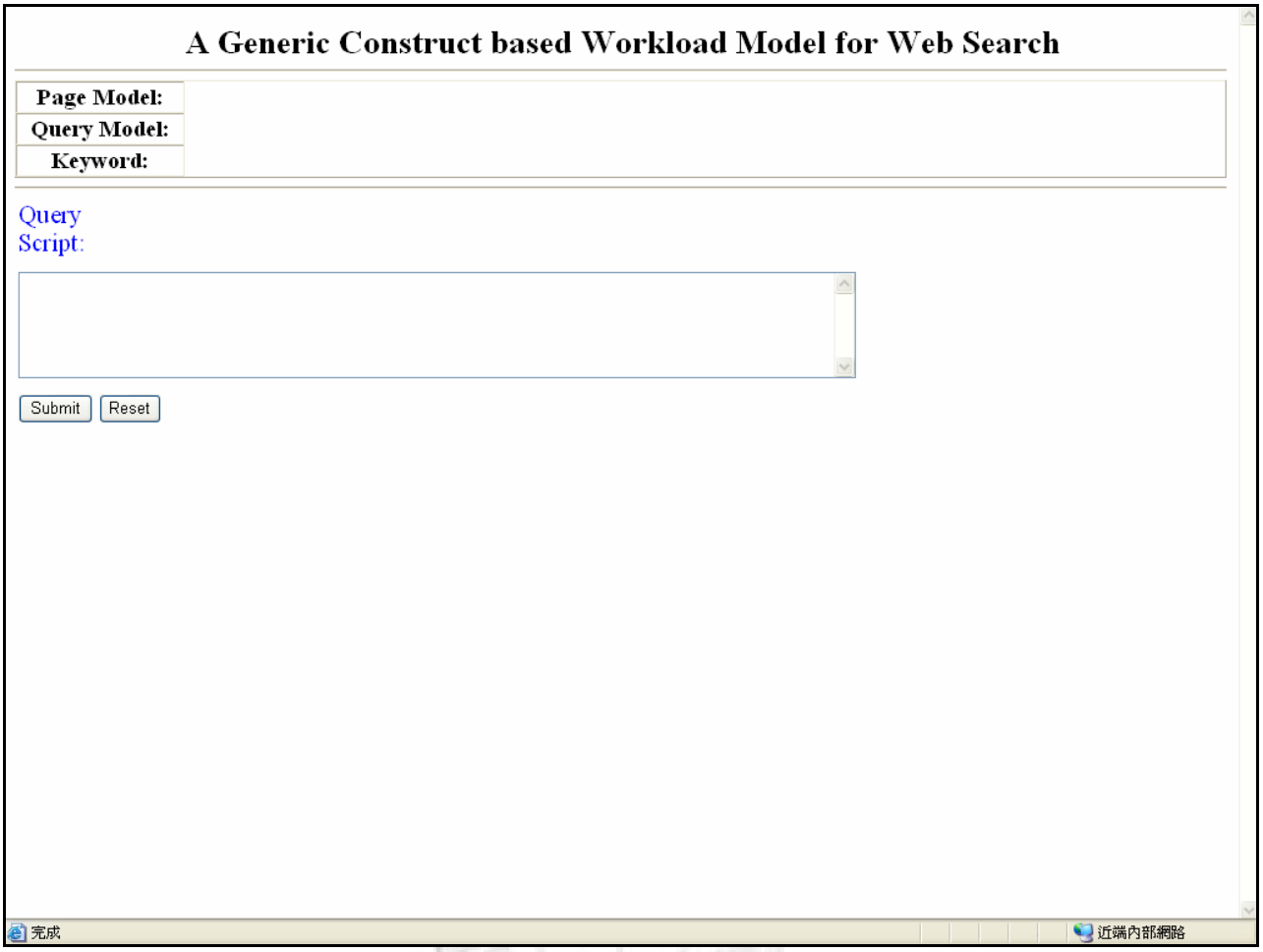

Figure 4.4: Query Script Output of the Workload Model

4.3.2 Scheduler

According to the control model, several parameters should be set to execute the benchmark. The parameters we implemented in the prototype are test sequence and number of repetitions. Both of them are in the open operation or computation input as well as in the operation or computation selector. Once the test set is determined, users can set up the executed sequence and the number of repetitions for each query in the scheduler (see Figure 4.5).

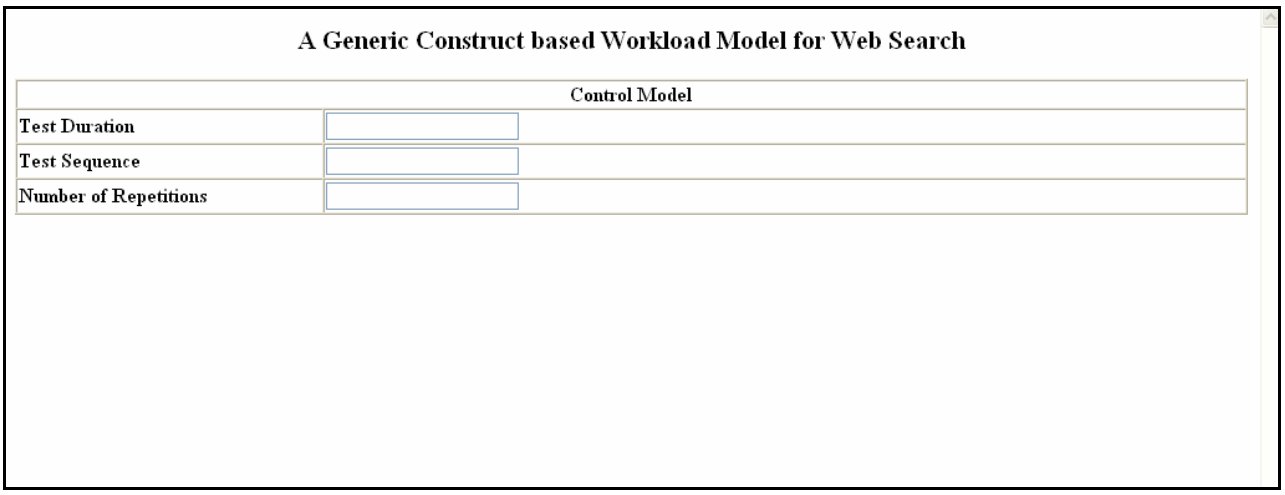

Figure 4.5: Control Model Input of Scheduler

4.3.3 Result Collector

The result collector shows the test result of the query script we specified. The test result can be divided into two parts: performance metrics and the lists of the search result. In the performance metrics, the total response time, throughput, totalResultsAvailbe and totalResutlsReturned are illustrated, and in each of search result lists, title, summary and URL are shown in Figure 4.6.

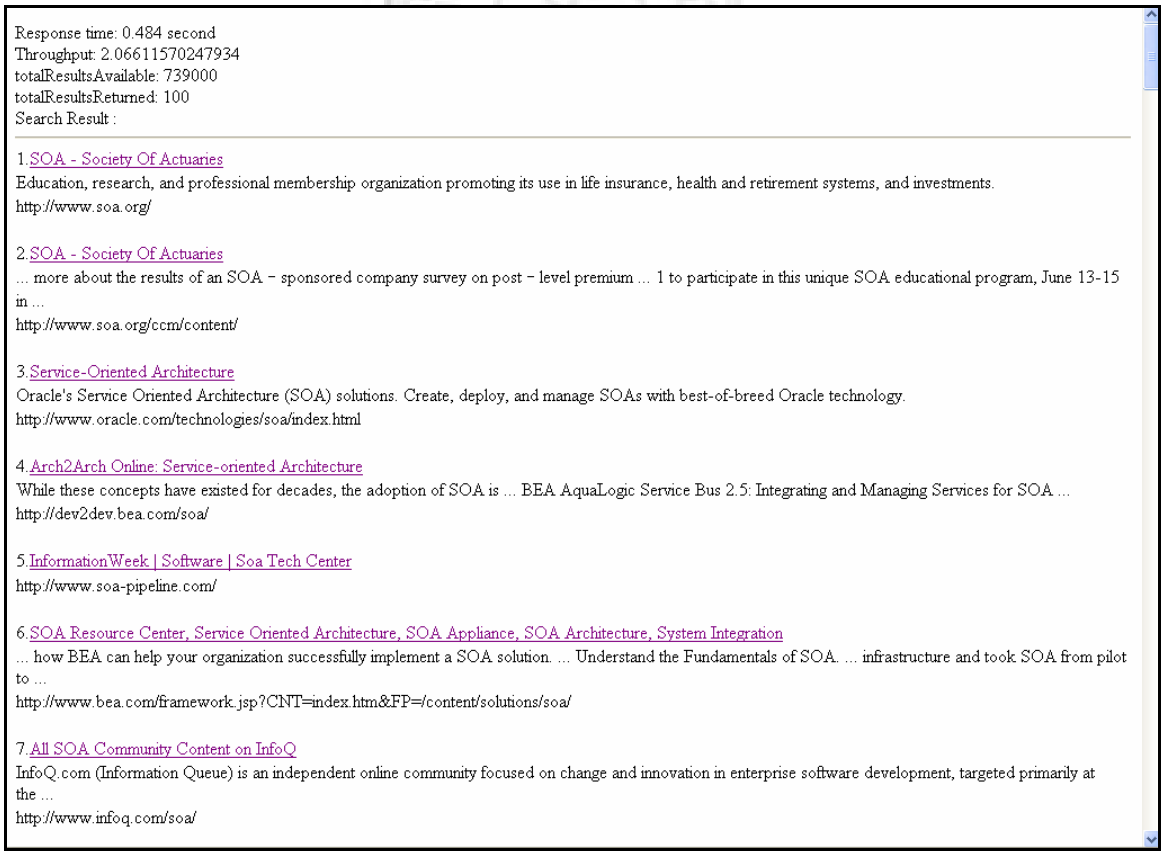

Figure 4.6: The Output of Result Collector