CSC6362 — System/Software Architecture
Research Track — Spring 2005
Project Part III: A Growing, Event-Driven Web Search Engine on a Middleware Platform
Due: Wednesday April 20 – Monday April 25 (during demo)

The design of complex systems must blend the art of architecture with the science of engineering.

— "The Art of Systems Architecting".

I. Summary

As system/software architects of a renowned company, your team is to architect and implement a web search engine, **Info-miner Primer**, using **Info-miner** – a growing web search engine which you implemented as part of Project II (which in turn uses **KWIC** which you implemented as part of Project I).

II. Info-miner Primer - A growing, event-driven web search engine on a middleware platform

Functional Requirements:

Info-miner primer shall accept a list of keywords and return a list of URLs whose descriptions contain any of the given keywords.

Info-miner primer shall be event-driven, instead of the pure object-orientation of **Info-miner**, which uses another software system as a component, **KWIC**, in order to efficiently maintain a database of URLs and the corresponding descriptions.

KWIC shall accept an ordered set of lines, where each line consists of two parts:

- *the URL part*, whose syntax is:

  URL ::= 'http://' identifier '.' identifier '.' ['edu' | 'com' | 'org' | 'net']

  identifier ::= {letter|digit}$^+$

  letter ::= ['a' | 'b' | ... | 'y' | 'z' | 'A' | 'B' | ... | 'Y' | 'Z']
  digit ::= ['1' | '2' | ... | '9' | '0']

- *the descriptor part*, whose syntax is:

  identifier {" " identifier}*.

The descriptor part of any line shall be "circularly shifted" by repeatedly removing the first word and appending it at the end of the line. The **KWIC**$^+$ index system shall output a list of all circular shifts of the descriptor parts of all lines in alphabetically ascending order, together with their corresponding URLs. No line in the output list shall start with any noise word such as "a", "the", and "of".

KWIC shall allow for two modes of operation: i) for building an initial KWIC indices; and ii) for growing the indices with later additions.

As with **Info-miner**, **Info-miner Primer** shall allow for:

- *Case sensitive search*: The system shall store the input as given and retrieve the input also as such;
- *Hyperlink enforcement*: When the user clicks on the URL retrieved as the result of a query, the system shall automatically take the user to the web site that matches the URL;
- *Specifying OR/AND/NOT Search*: A keyword-based search is usually an OR search, i.e., a search on any of the keywords given. The system shall allow the user to specify the mode of search using "OR", "AND" or "NOT";
- *Multiple search engines*: to run concurrently;
- *Deletion of out-of-date URL*: and corresponding description from the database;
- *Listing of the query result in alphabetically ascending order*;
- *Setting the number of results to show per page, and navigation between pages.*

Non-Functional Requirements:

Info-miner primer shall be easily understandable, portable, enhanceable and reusable with good performance. **Info-miner** must also be user-friendly, responsive, and adaptable.

III. The Deliverable

Your description should be elegant and comprehensible. Your deliverable should be available as both on-line and off-line specifications:

1. **Architecture** Describe both pictorially and textually the constituents of the architecture, namely, components/elements, interactions/connections, any constraints, any patterns. Also present a discussion of advantages and disadvantages of your architecture.

   Your desciption shall establish *traceability* for both functional requirements and non-functional requirements.

   Your desciption shall establish *traceability* for both functional requirements and non-functional requirements.

   N.B.: Your desciption shall establish *traceability* for both functional requirements and non-functional requirements.

2. **Implementation** Build a system using a middleware platform, either J2EE or the .NET framework, which shall run with a web browser (preferably with recent versions of both NetScape and IE).

3. **User Manual** Describe the URLs (of course web sites) of all the team members. Describe the typical interactions between the user and the system, e.g., what are the steps the user has to follow in using the system. Also describe any operating requirements. The manual shall use screendumps whenever possible.