Ken Nguyen, Software Engineer
Martin Walther, Software Engineer
Hugh Garsden, Senior Software Engineer
Prof. Enrico Coiera, Director Centre for Health Informatics
A/Prof. Nigel Lovell, Deputy Director Centre for Health Informatics

Centre for Health Informatics
Level 2, Samuels Building (F25)
UNSW
Sydney NSW 2052
Telephone: +61 2 9385 1270
FAX: +61 2 9385 1813

Topic: Research
Stream: Academic

Presentation Learning Outcomes:
- Methodology of a decision support system used in clinical settings.
- Software architecture for the decision support system.
- Results of the trial of the decision support system used in clinical settings.

Peer review: Yes
Presentation type: Oral
Architecture for an on-line and just-in-time decision-support for General Practice

Introduction

Those involved in the provision of health care need to keep up-to-date with respect to the latest knowledge in their field. The problem for these professionals is that the amount of available literature is exponentially increasing. Instead of reading all the literature, they need to be able to search it for specific items of interest or be alerted to things they should know about.

A proposed solution to this problem is a system that allows intelligent searches of current guidelines, textbooks, journals, etc. and presents the information in a way that is easily digested. The system would have similarities to a meta-search engine and include access to selected peer-reviewed information sources, intelligence to understand users’ requests and the capacity to present results in an easy-to-understand way, tailored for health care professionals.

Based on extensive study of the needs of the General Practitioner (GP) [1], we designed and implemented the “Quick Clinical” system to provide just-in-time information for decision support. Quick Clinical (QC) gathers the clinical context and the specific question from the GP and searches through numerous sources to find documents that best answer the question.

Traditional information retrieval systems are based on the library model where one can search for a specific title or document and then browse through the chapters or sections to find the required information. However, in a clinical just-in-time environment this model is inadequate on two accounts. Firstly, one needs to know where to search, i.e. in which of the various information systems (searching all sources is not an option because it returns far too many results). Secondly, one needs to know how to search the selected source or sources. In QC, the search context is used to tailor the search request, for example, a “prescribing” request will only search particular knowledge sources.

System Architecture

The system’s main components are the User Interface (UI), the Mediator and a number of Wrappers [2] and data sources. The UI is a Web page that supports the user in composing the search query. A complete search query is defined by a search context and a set of search keywords. This information is forwarded to the Mediator. It is the Mediator’s task to dissect the query and pass it on to different wrappers (one per source). While the wrappers execute the search request with the actual data source, the Mediator waits to collect the results. Once all results are collected the Mediator merges them together and passes the combined result on to the UI.
To make the system more versatile it requires a number of different data sources. However, every data source is unique in its structure and search capabilities. Thus, the system has to deal with many heterogeneous sources through their individual search interface. It is this heterogeneous nature of the data sources that complicates the search process. Depending on which data sources are used in a search, it is necessary to rewrite the search query accordingly. To simplify the search process we built wrappers that encapsulate the data sources and provide a unified search interface. In other words, the wrappers translate search queries from our unified search language (USL) into a source specific language. Once the result from the data source is available, the wrapper translates the result back into our USL. The wrappers become especially valuable when a data source changes its search interface (e.g. the Web page is upgraded). Only the corresponding wrapper will have to be adapted, the other components of the system remain unchanged.

The system has been constructed using Java and Cold Fusion (Macromedia Inc.), running on Red Hat 7.1 Linux and Windows 2000. The UI is deployed through an Apache Web server connection to a Tomcat servlet engine. The Apache-Tomcat platform incorporates load balancing and fail-over and is suitable for scalability and large-scale deployment.

Conclusion

QC has undergone limited technical evaluation to assess usability and functionality in the GP environment (N=21). Using twelve data sources, the average response time for a search was 13.1s with 75% of GPs rating the speed of search as good or excellent. The majority of GPs expressed the opinion that QC has the potential to improve patient care (67%).

References