Automation and Integration of Components for Generalized Semantic Markup of Electronic Medical Texts

Jonathan M. Dugan MS*
Daniel C. Berrios MD, MPH
Xiaole Liu, David K. Kim, MS
Herbert Kaizer, MD, PhD
Lawrence M. Fagan MD, PhD
Stanford Medical Informatics, Department of Medicine, Stanford University School of Medicine, Stanford, California
*Corresponding Author: dugan@smi.stanford.edu

ABSTRACT

Our group has built an information retrieval system based on a complex semantic markup of medical textbooks. We describe the construction of a set of web-based knowledge-acquisition tools that expedites the collection and maintenance of the concepts required for text markup and the search interface required for information retrieval from the marked text.

In the text markup system, domain experts (DEs) identify sections of text that contain one or more elements from a finite set of concepts. End users can then query the text using a predefined set of questions, each of which identifies a subset of complementary concepts. The search process matches that subset of concepts to relevant points in the text.

The current process requires that the DE invest significant time to generate the required concepts and questions. We propose a new system — called ACQUIRE (Acquisition of Concepts and Queries in an Integrated Retrieval Environment) — that assists a DE in two essential tasks in the text-markup process. First, it helps her to develop, edit, and maintain the concept model: the set of concepts with which she marks the text. Second, ACQUIRE helps her to develop a query model: the set of specific questions that end users can later use to search the marked text. The DE incorporates concepts from the concept model when she creates the questions in the query model. The major benefit of the ACQUIRE system is a reduction in the time and effort required for the text-markup process.

We compared the process of concept- and query-model creation using ACQUIRE to process used in previous work by rebuilding two existing models that we previously constructed manually. We observed a significant decrease in the time required to build and maintain the concept and query models.

INTRODUCTION

Health-care professionals in clinical environments need precise information if they are to provide optimal patient care.[1-3] Through analysis of these information requirements, Osheroff and colleagues[4] showed that tertiary references, such as textbooks or edited reviews, could meet the majority of these information needs. Other studies[5, 6] showed a distinct need for more specific search results retrieved from medical databases, supporting the need for well-indexed tertiary sources such as textbooks.

Researchers have developed a variety of systems to improve the indexing and searching of medical text sources; the primary goal is to increase search precision without severely reducing recall.[†] For example, Purcell and Shortliffe[7] developed a system of document templates to delineate context for specific phrases in medical journals. Lenert and Tovar[8] developed a system for coding free-text reports found in patient records. Hersh and collaborators developed the SAPHIRE system to improve biomedical-information retrieval.[9, 10]

Kim and associates built MYCIN II[11], a prototype IR system capable of searching content-based markup in an electronic textbook on infectious disease. The MYCIN II system is a specific implementation of a general web-based **search engine.** Users select from pre-determined set of query templates (the *query model*) a query that is passed to a search engine for processing. Based on the first version by Kim, Liu rebuilt the search engine to use a table driven approach for defining the search method. **Figure 1** shows an example set of query templates through which the user can query the revised search engine. We marked a text on hematopoietic stem-cell transplantation[12] (HSCT) for the examples in this report.

Berrios and colleagues[13] developed a **markup tool** (**Figure 2)** that provided the HTML indexing required for the MYCIN II search engine. A significant amount of manual work was required by DEs to generate the ontology of concepts in the concept model and the set of questions for the search engine in the query model. In addition, we developed the tools as separate projects so there was minimal integration between this markup tool and the search engine. This resulted in the domain expert DE repeating several common tasks when using both tools.

The previous work of Kim, Liu, and Berrios provide the tools used in the current text-markup process. We sought

[†] Precision (or specificity) = fraction of relevant search hits (results) to all hits; Recall (or sensitivity) = fraction of relevant hits to all possible relevant hits in the database.

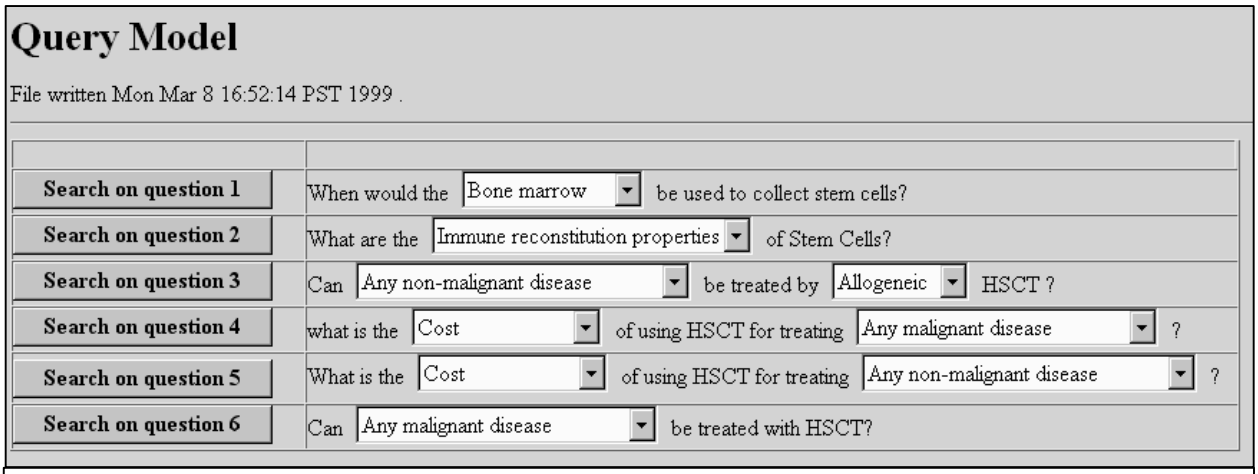

Figure 1: The query model with an example set of query templates in the domain of stem cell transplantation.

to integrate these tools in a common markup and search environment. We developed ACQUIRE (Acquisition of Concepts and Queries in an Integrated Retrieval Environment), a knowledge-acquisition (KA) tool that created and maintained the data resources for both the markup tool and the search engine, and that generated automatically an HTML interface for user queries. We refined and enhanced both the search engine and the markup tool to work with ACQUIRE.

The creation of ACQUIRE is an important step in a longer project to build a single integrated system that incorporates all required tasks for semantic text indexing and retrieval. Our hypothesis in this study was that by integrating the search engine and the markup tool, ACQUIRE will automate the text markup procedure. The measurable result of this automation will be savings in time for the DE who performs the text markup.

DESIGN CONSIDERATIONS

Creation of the concept and query models is a dynamic process. As text is being marked, the domain expert often needs to edit and change the concepts in the concept model. We therefore wanted ACQUIRE to facilitate dynamic and concurrent alteration of the concept model during text markup, and to automate the transfer of those data to the markup tool.

DEs who use ACQUIRE should also be able to develop a domain-specific query model with questions that relate specifically to the text. She should be able to use the text being marked to assist in the creation of the query model. This requirement directed us to design ACQUIRE such that it permits alteration of the query model during text

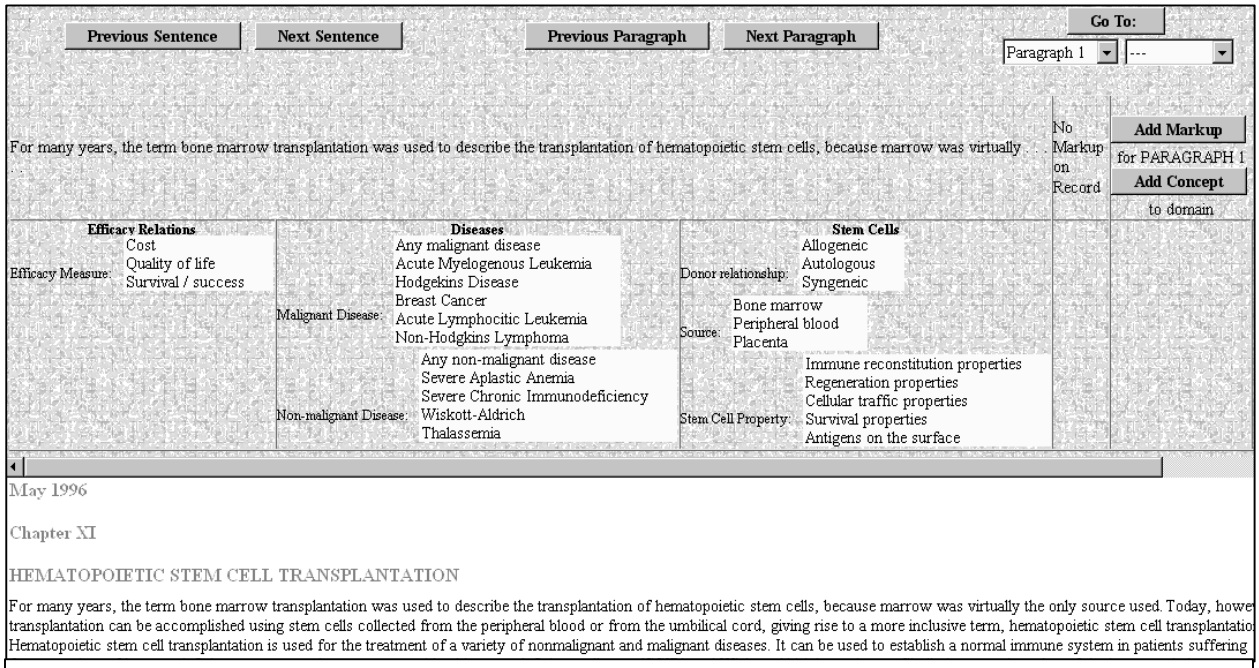

Figure 2: The markup tool used by the DE to provide XML-style indexing of medical texts.[13]

markup. As concepts are added to the concept model for use in markup, they must be immediately available to help the DE construct appropriate questions in the query model. Finally, ACQUIRE would need to generate automatically the interface similar to the one shown in Figure 1 for accessing the search engine.

We chose to make ACQUIRE available over the web to allow easy integration with the existing text markup system, which uses XML-style markup. We chose to implement ACQUIRE in HTML in part because of the stability and availability of development tools. Languages such as XML, although more expressive than HTML, were rejected because of a lack of development tools. Ideally, DEs could use existing knowledge-acquisition (KA) tools, such as Protégé[14], but we deferred this choice until the release of the Protégé JAVA version, which will simplify several of the integration issues.

SYSTEM DESCRIPTION

Figure 3 shows a schematic diagram of how the complete markup system works. We describe the concept model, the use of the markup tool by the DE, the query model, and the actions of the search engine. Then we discuss ACQUIRE and detail its contributions to the markup process.

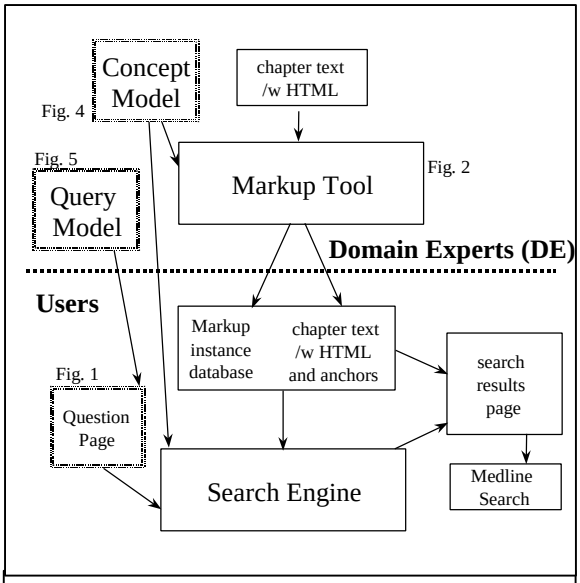

Figure 3: Schematic of the Text Markup system showing those elements altered by ACQUIRE with shaded boxes.

The concept model has three parts: **headings concepts,** and **values**. We use *headings* to organize the *concepts* into coherent groupings, but they do not contribute to the text markup. *Concepts* are variables that take on as their values one of the listed *values* for each concept. There are separate *values* listed for each *concept.*

The DE uses the concept model to mark an electronic text with the markup tool. To create an instance of markup, she chooses one or more *concept–value* pairs to associate with a specific point in the text. The markup tool then associates this markup (the set of concept–value pairs) with a hidden HTML anchor, which is placed in the text. The anchor references an entry in the markup instance database, which stores all the concept–value pairs used in the markup.

The query model is a structured set of questions created by the DE. Each question has one or more replaceable parameters that correspond one-to-one with concepts in the concept model. Users wishing to search the text select from the allowed values for each concept when forming specific queries. In this way, each question becomes a template for many possible questions that are formed by the user as he chooses particular values for each concept.

The search engine receives the set of user-selected concept–values pairs from a given question and matches the set of pairs against the markup-instance database. The search hits occur when markup instances contain all concept–value pairs specified by the question. The search engine then presents the original text corresponding to the associated HTML anchor for each hit.

THE ACQUIRE SYSTEM

The concept-model tool (**Figure 4**) is the first of two interfaces in ACQUIRE. This tool allows the DE to create of the concept model dynamically during the markup process. The concept-model tool is implemented with HTML forms, and is controlled with CGI (Common Gateway Interface) scripts written in PERL. The DE can add entries to or change entries in this interface; such actions change the concept model. These changes in the concept model are reflected in the query model and in the markup tool. Concept models can be saved for later use.

The query-model tool (**Figure 5**) is the second interface in ACQUIRE, and also uses HTML forms. The current query model is displayed in the upper portion of the form. Each question contains one or more pull-down menus, each populated by the allowed values of a single concept in the concept model. The query-model tool updates these values dynamically when the DE changes the concept model using the concept-model tool.

The DE adds new questions to the query model using the bottom half of the query-model tool. To construct a question, she enters free text to the input boxes (e.g., "Can″) and then selects a concept in adjacent pull-down menus where appropriate (e.g. "Malignant Disease"). More free text or pull-down choices complete the question (e.g. "be treated with stem-cell transplantation?"). In this example, the DE wants the user to select a specific malignant disease from among the values associated with the concept "Malignant Disease." In this way, questions integrate data from the concept model. The query-model tool generates automatically the HTML interface (Figure 1) required by the search engine.

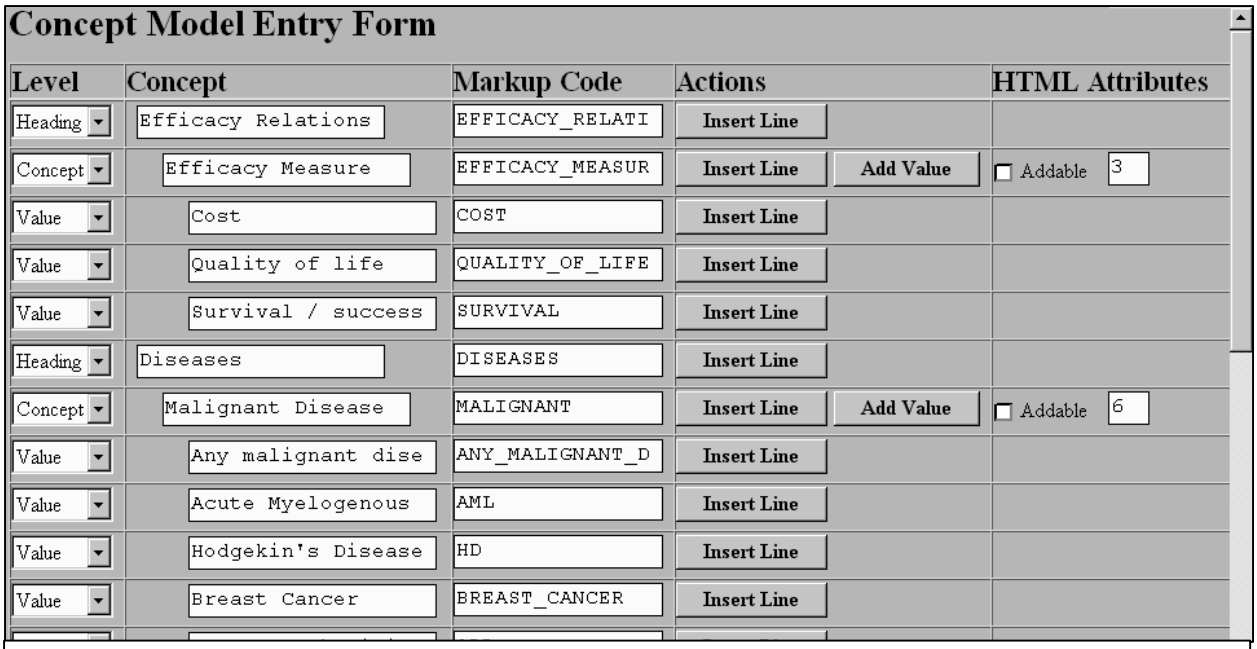

Figure 4: The ACQUIRE concept-model tool used by the domain expert to enter and maintain the concept model.

ACQUIRE uses frames to display both the tools simultaneously in a single browser. The DE can develop the concept model and the query model concurrently because the two interfaces share relevant data in the concept model. The concept model tool automatically creates the data structure required for the markup tool, and points the DE directly to a new URL for a domain-specific version of the markup tool. The query model tool can spawn browser windows containing a search engine interface similar to the one shown in Figure 1.

TESTING THE ACQUIRE SYSTEM

We have worked with three domain areas on this electronic textbook project: infectious disease (ID), medical oncology, and HSCT. We have used ACQUIRE to recreate a portion of the original infectious disease concept and query models and to build the HSCT models.

We verified that we could recreate a representative portion of the ID models using ACQUIRE and determined that there were no concepts or functionality that we could not recreate. We also compared the time required to build concept and query models for two application areas of medicine: one built by hand (medical oncology), and one by ACQUIRE (HSCT).

It took about 3 student-weeks to build the oncology domain by manually linking the tools together and building and testing the concept model. It took about 4

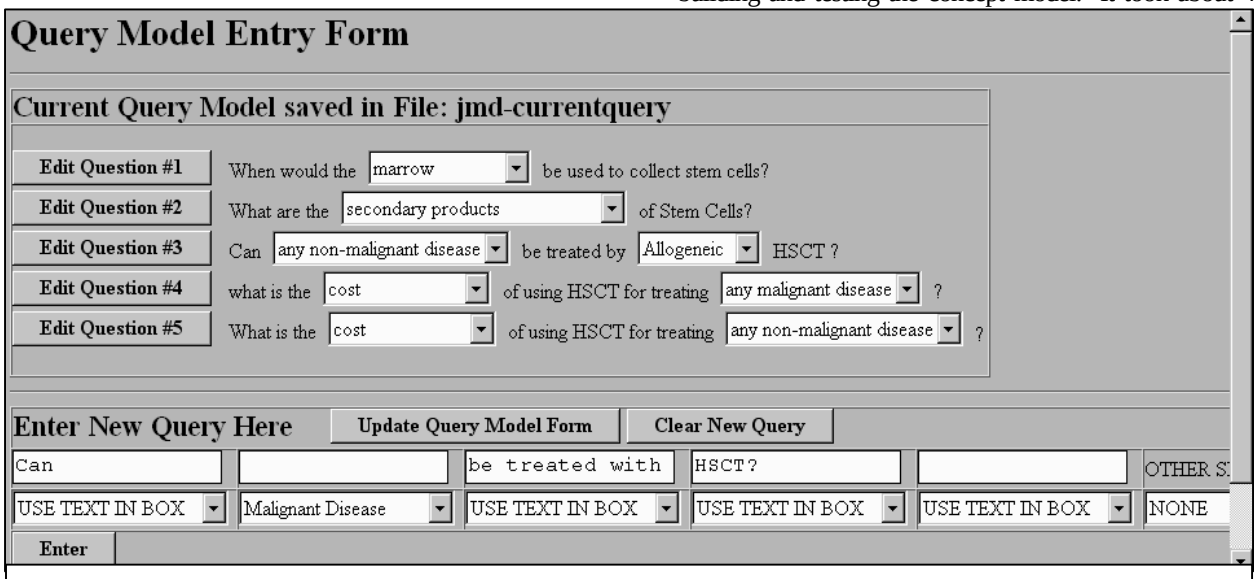

Figure 5: The ACQUIRE query-model tool used by the domain expert to generate the query templates in the query model.

hours using the ACQUIRE system to build the HSCT system with approximately the same level of difficulty as measured by the complexity of concept and query models.

DISCUSSION

We demonstrated that it is possible to automate the process of knowledge acquisition for the creation of a concept model and a query model in a textbook-markup system by integrating several disparate tools. The use of ACQUIRE reduced the time required of domain experts using the markup system. ACQUIRE eliminated manual creation of input files and editing of those files by the domain experts.

One of the constraints of the current markup system is the choice of organization in the concept model. The current model uses sets of concept–value pairs organized into groups under headings. This method is not hierarchical and is therefore not scalable to more complicated concept hierarchies. This constraint was significant in the creation of markup for the domain of stem-cell transplantation. In the future, we plan to organize the concept model to permit multilevel, hierarchical organization of concepts.

Three changes would allow a multilevel hierarchy. Simple changes to the concept model tool would permit the DE to construct a more complicated model. We would need to change the markup tool to use a schema different from the concept–value scheme that it currently uses to identify each markup instance. Instead, the markup tool would point more generally to a place in a concept hierarchy. We would also need to alter significantly the search tool to reflect the more complicated markup scheme. The changes to the markup tool and search engine would require major changes to their function–arguably, these changes might necessitate rebuilding of the tools.

Another area for future work is the expansion of the concept model to include relations. Currently, the context in which a concept is used in the markup is implicit. Explicit representation of the relation between certain concepts could make the markup more expressive and thus lead to more accurate search results. However, using a relational model would further complicate the organization of the concept model and make the markup tools more difficult for the DE to use. We think that a hierarchical concept model with relations may contain such complexity that it requires DEs to use more robust and complex ontology editors, such as Protégé., to effectively manage the concept model.

SUMMARY

We have demonstrated that it is possible to automate the process of knowledge acquisition in the context of semantic markup of electronic medical textbooks. Through the development of ACQUIRE and integration of an existing markup tool and search engine, we achieved a significant time savings in the text-indexing and retrieval process.

ACKNOWLEDGMENTS

We thank Lyn Dupré for her patient copy editing. Victor Yu at the Pittsburgh VA established this project. The work was supported in part by the National Cancer Institute Contract # N44-CO 61025 to Lexical Technologies Inc. and by the National Library of Medicine Training Grant LM 07033.

References

1. Covell DG, Uman GC, Manning PR. Information needs in office practice: are they being met? Ann Intern Med 1985;103(4):596-599.
2. Stinson ER, Mueller DA. Survey of health professionals' information habits and needs. Conducted through personal interviews. Jama 1980;243(2):140-143.
3. Woolf SH, Benson DA. The medical information needs of internists and pediatricians at an academic medical center. Bull Med Libr Assoc 1989;77(4):372-380.
4. Osheroff JA, Forsythe DE, Buchanan BG, Bankowitz RA, Blumenfeld BH, Miller RA. Physicians' information needs: analysis of questions posed during clinical teaching. Ann Intern Med 1991;114(7):576-581.
5. Williamson JW, German PS, Weiss R, Skinner EA, Bowes Fd. Health science information management and continuing education of physicians. A survey of U.S. primary care practitioners and their opinion leaders. Ann Intern Med 1989;110(2):151-160.
6. Forsythe DE, Buchanan BG, Osheroff JA, Miller RA. Expanding the concept of medical information: an observational study of physicians' information needs. Comput Biomed Res 1992;25(2):181-200.
7. Purcell GP, Shortliffe EH. Contextual models of clinical publications for enhancing retrieval from full-text databases. Proc Annu Symp Comput Appl Med Care 1995:851-857.
8. Lenert LA, Tovar M. Automated linkage of free-text descriptions of patients with a practice guideline. Proc Annu Symp Comput Appl Med Care 1993:274-278.
9. Hersh WR, Greenes RA. SAPHIRE--an information retrieval system featuring concept matching, automatic indexing, probabilistic retrieval, and hierarchical relationships. Comput Biomed Res 1990;23(5):410-425.
10. Hersh W, Hickam D. Information retrieval in medicine: the SAPHIRE experience. Medinfo 1995;2:1433-1437.
11. Kim DK, Fagan LM, Jones KT, Berrios DC, Yu VL. MYCIN II: design and implementation of a therapy reference with complex content-based indexing. Proc Amia Symp 1998:175-179.
12. Dale DC, Federman DD. Hematopoitic Stem Cell Transplantation. In: A Comprehensive Knowledge Base of Internal Medicine. Scientific American; CD Ver. 1998.
13. Berrios DC, Kehler A, Kim DK, Yu VL, Fagan LM. Automated Text MArkup for Information Retrieval from an Electronic Textbook of Infectious Disease. Proc Amia Symp 1998:975.
14. Musen MA. Domain ontologies in software engineering: use of Protege with the EON architecture. Methods Inf Med 1998;37(4-5):540-550.