Linking Clinical and Knowledge-Based Information Systems: A Comparative Review of Interoperability Protocols

by

Keith C. Klopfer, MD

CAPSTONE PROJECT

Presented to the Division of Medical Informatics and Outcomes Research
and the Oregon Health & Science University School of Medicine
in partial fulfillment of the requirements for the degree of
Master of Medical Informatics

Spring 2003

<u>Advisor</u>

Judith Logan, MD, MS

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to extend my sincere thanks to those individuals who have helped make this degree a reality. To my advisor, Judith Logan, for her guidance and support over the years, and for her help with the design and write-up of this project. To Bill Hersh for his guidance and inspiration in pursuing this line of research. To Kent and all the fellows, who have made the last 3 years educational and fun. And most importantly, to my family and friends, especially Lara, who have always been there for me.

Abstract

The vast majority of physicians have unmet information needs during their clinical encounters with patients. Studies have shown that these information needs vary, but can be as high as 5 questions per patient encounter. Traditionally, most physicians would seek answers to these questions using paper resources, such as textbooks, systematic reviews, or journal articles. These sources of medical knowledge are commonly referred to as knowledge-based information (KBI). However, the success of the World Wide Web as a convenient source of medical knowledge is starting to change the way clinicians address their information needs. More and more clinicians are starting to view the Internet as a potential source for KBI and decision support.

As the electronic health record (EHR) becomes more commonplace, there is a great opportunity to link these KBI sites with the EHR, and bring relevant KBI to the point of care. However, in order for this connection to occur, there needs to be agreement as to the syntax and semantics of the communication between these two types of systems. This can be accomplished through the standardization and adoption of *interoperability protocols*. Interoperability protocols have been well researched in the digital library community, but have yet to be applied to the problem of integrating KBI with the EHR. This paper examines three promising interoperability protocols for their potential to link KBI with the EHR. The protocols are: the Z39.50 Information Retrieval standard, the Open Archives Initiative – Protocol for Metadata Harvesting, and the Simple Digital Library Interoperability Protocol. This paper will discuss the general issues surrounding the interoperability of KBI in health care and provide a technical comparison of the 3 protocols. Z39.50, OAI, and SDLIP represent significant advances in interoperability and should be researched further as an effective interoperability layer between knowledge-based information systems and the electronic health record.

Introduction

One of the defining characteristics of the field of medical informatics is the application of advances in information technology to the problems of health care. Over the past decade, many new technologies have emerged to enable better solutions to existing informatics problems. One of the most challenging of these problems is the integration of the electronic health record (EHR) with knowledge-based information (KBI) systems. This "connection" needs to be established in order to provide the clinician with relevant decision support information at the point of care, and in turn, to improve the quality of health care in this country.

The success of the World Wide Web (WWW) as a convenient source of medical knowledge has changed the way clinicians address their information needs. Many physicians now turn to the Web, instead of textbooks and other paper resources, to answer the clinical questions that arise during their patient encounters. Hundreds of quality web sites exist to provide decision support to the clinician. One example shown to be effective is Medline, a comprehensive bibliographic database enabling physicians to search and retrieve relevant journal articles from the medical literature [1]. Other sites frequently used by clinicians include the Cochrane Collaboration, the National Guideline Clearinghouse, and a growing number of commercial sites that provide a variety of medical content. Despite the plethora of medical resources available on the Internet, there are many challenges to integrating this content with the EHR.

The majority of KBI repositories[1] on the Internet are designed for interaction with a human user, such as a physician. The user navigates to a web site, logs in, and searches the information. The physician may have to visit several web sites to find the answer to a simple clinical question such as "What is the current best drug therapy for tinea pedis?" This can be a time consuming and frustrating process to a busy clinician. Unfortunately, few of these web sites have the interfaces needed to allow other software applications, such as the EHR, to access this information automatically. This lack of *interoperability* is one of the biggest challenges to integrating KBI with the EHR. Fortunately, promising advances in the past

---

[1] This paper will use the term ***repository*** to denote a collection of digital resources, regardless of the type of resource or the means by which they are stored (RDBMS, File System, etc.).

decade in information and networking technology, as well as efforts in the digital library research community, may lead to a solution.

This paper will outline the challenges associated with integrating the EHR with KBI and focus on the use of *interoperability protocols* to facilitate communication between these types of systems. An interoperability protocol is a standard syntax and communication model for how two computer applications exchange information. In this case, the interoperability protocol defines the syntax of the communication between the EHR and the KBI repository. Several interoperability protocols have emerged in recent years, primarily from research efforts in the digital library community. Three of these protocols will be reviewed for their potential to act as an effective link between the EHR and KBI repositories. The three protocols are: the *Z39.50 Information Retrieval* standard, the *Open Archives Initiative – Protocol for Metadata Harvesting*, and the *Simple Digital Library Interoperability Protocol*. These protocols will be reviewed in the context of providing remote access to KBI at the point of care. The analysis will be generalizable to any type of information, but emphasis will be placed on those resources applicable to the information needs of clinicians.

Background

Information Needs of Clinicians

Medicine is inherently an information-rich discipline. The typical health care provider needs access to a vast and diverse array of information sources in the day-to-day care of her patients. In order to make accurate decisions, she needs access to the latest *clinical data* about her patients, such as recent laboratory tests, vitals signs, and medication lists. This type of information is typically referred to as patient-specific information, or PSI, and can be found in the EHR. Many research efforts have focused on PSI; from representing clinical concepts in a unified manner [2], to messaging standards for the transmission of PSI between clinical information systems [3].

The other major type of information needed by clinicians is knowledge-based information, or KBI. KBI refers to the body of generalized medical knowledge, manifested in textbooks, journals, systematic reviews, and clinical guidelines. These are the resources a clinician turns to when asking questions like "Is there a drug-drug interaction between Cipro and Lovenox?" Access to relevant KBI is becoming increasingly important to health care providers, especially in primary care, where they are required to handle the widest range of ailments. Medical knowledge is expanding too rapidly for physicians to keep abreast on the latest research. As has been pointed out, "some of the information in doctors' heads is out of date and wrong, new information may not have penetrated, and the information may not be there to deal with patients with uncommon problems" [4]. It is therefore not surprising that most physicians have unmet information needs when caring for their patients. Figure 1 illustrates the information needs of clinicians in the clinical setting.

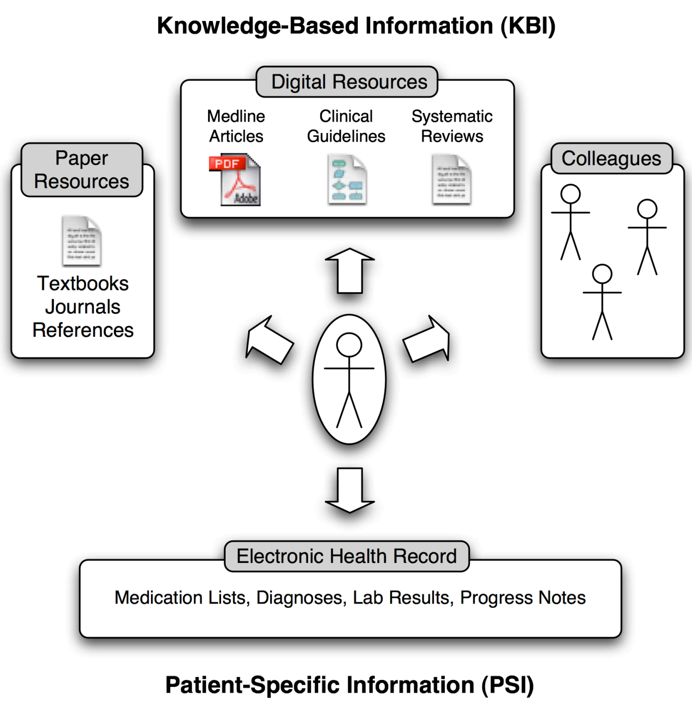

Figure 1. Information needs in health care. Clinicians need access to a diverse array of information sources in the care of their patients. These can be loosely divided into two categories – knowledge-based information (KBI) and patient-specific information (PSI).

The information needs of clinicians are well documented in the medical literature [5,6,7]. Gorman et al. report, in a systematic review, that estimates of clinical information needs vary from less than one question per patient, to as many a 5 questions per patient encounter [6]. These studies not only show that physicians generate a large number of questions while they see patients, but that a significant number of these questions go unanswered. Those questions that are pursued are usually answered using colleagues or printed literature, such as textbooks. Few health care providers have embraced online resources to pursue these questions, possibly because the process of accessing online KBI does not integrate well with the workflow of the busy physician. Until there is better integration of KBI with the EHR, most physicians will likely continue to use other resources or none at all.

Interoperability in Health Care

Background

Despite recent advances in the computing industry, the technical infrastructure in medicine to allow for interoperability of KBI is lacking. This is not the case for all parts of the health care system, however. There has been considerable progress made in the interoperability of PSI. For example, the Health Level 7 (HL7) messaging standard, which is currently being redesigned to support XML[2], has enabled clinical information systems, such as the EHR and laboratory reporting systems, to exchange information seamlessly. DICOM, the Digital Imaging and Communications in Medicine standard, has done the same for the exchange of radiographic images between systems [8]. Both DICOM and HL7 are considered interoperability protocols for exchanging *clinical information*.

In the digital library community, interoperability protocols have been developed for the search and retrieval of library resources, such as electronic versions of books and journal articles. The primary goal of these protocols is to allow for *resource discovery* from library collections located throughout the world. For example, using the Z39.50 protocol (to be discussed later) you can query dozens of digital libraries, including the Library of Congress, for a specific resource. Most major library systems have implemented one or more of these protocols, allowing users to search for books remotely.

---

[2] The eXtensible Markup Language (XML) is a standard for describing the syntax and structure of information in a hierarchical manner. It has quickly become an industry standard as a data representation and messaging format.

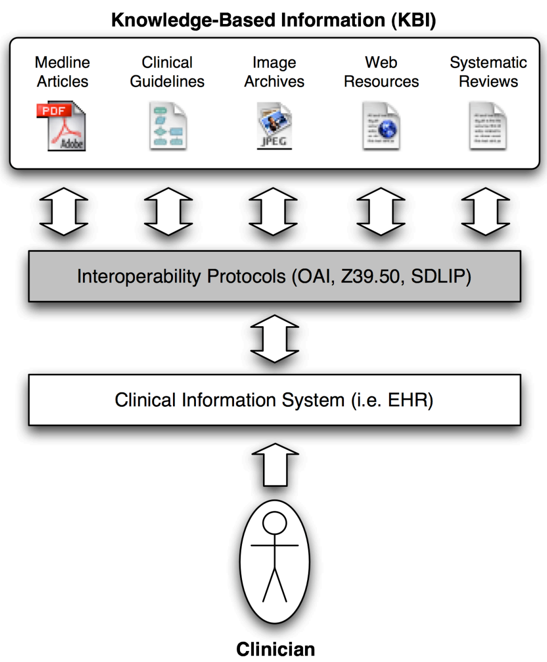

Figure 2. An interoperability layer for KBI. Interoperability protocols, like OAI, Z39.50 and SDLIP, facilitate the communication between KBI repositories and clinical information systems. This model has the potential to bring KBI closer to the point of care.

In recent years, researchers have proposed using these protocols to facilitate the interoperability of KBI in health care [9]. Figure 2 illustrates how these protocols may be used to create an "interoperability layer" between KBI repositories and the EHR. While these protocols were primarily designed for resource discovery of digital library information, it is unclear whether they can be adapted to serve as viable solutions for the interoperability of KBI in health care. In addition, there are challenges beyond the interoperability protocol that play an important role, such as metadata and digital rights management.

Metadata

Resources on the Internet are generally searched in one of two ways. More commonly, the text of a document is parsed and an *index* is created based on the frequency of words in the document. This index is then searched to retrieve those documents containing the terms of interest. This is the most common type of searching on the Internet and the one most search engines, such as Google[3], utilize. One disadvantage to this approach is that only text documents can be indexed. There is no easy way to do content indexing of

3 http://www.google.com/.

non-text resources like images or video clips. This is a significant impediment to effective searching since much of KBI exists in non-textual formats.

The other type of searching uses *metadata*[4] to describe digital resources. There are several metadata formats in use today, but the most promising is the Dublin Core Metadata Element Set (DC). Designed to describe web-based resources, DC is a simple metadata scheme consisting of only 15 elements [10]. Table 1 list the 15 basic elements of the DC standard along with descriptions of each. One of the unique features about DC is the ability to customize, or *qualify*, the 15 basic elements of the set. This allows for different communities to refine DC to accommodate the needs of their domain. One such refinement is the Medical Core Metadata (MCM) -- a syntax for describing health and medicine resources on the Internet [11].

Element	Definition
Title	The name given to the resource.
Creator	An entity primarily responsible for making the content of the resource.
Subject	A topic of the content of the resource.
Description	An account of the content of the resource.
Publisher	An entity responsible for making the resource available.
Date	A date of an event in the lifecycle of the resource.
Contributor	An entity responsible for making contributions to the resource.
Type	The nature or genre of the content of the resource.
Format	The physical or digital manifestation of the resource.
Identifier	An unambiguous reference to the resource in a given context.
Source	A reference to a resource from which the present resource is derived.
Language	A language of the intellectual content of the resource.
Relation	A reference to a related resource.
Coverage	The extent or scope of the content of the resource.
Rights	Information about rights held in and over the resource.

Table 1. The Dublin Core Metadata Element Set. The DC element set specifies 15 basic elements, but allows for qualification of these elements to provide more granularity.

Digital Rights Management

Content providers, such as digital libraries and commercial websites, have a large investment in the content they provide. This investment not only includes the intellectual property of their content, but the cost of maintaining the repository as well. Any potential interoperability protocol will need to take into account the business needs of these stakeholders. Digital Rights Management (DRM) technology is a mechanism for providers to manage access to their content and protect their resources from unauthorized

[4] Metadata is a structured set of descriptive information about a resource. Title, author, subject, and pubication date are examples of common metadata elements used to describe a digital document.

copying and abuse. For example, a repository should be able to specify not only which individuals have access to a resource, but what actions are allowed to be performed on that resource, such as viewing, printing, copying, etc. This is an area that has received much attention in the computing industry, primarily due to the ease at which digital resources, such as MP3 music files, can be copied and exchanged. This has forced the industry to develop DRM strategies [12], some of which can be implemented in conjunction with these protocols.

Current Research Efforts

Only a handful of research studies have attempted to link clinical systems with KBI at the point of care. Cimino et al. have done the most significant work in this area and report, in a series of articles, the successful integration of an EHR and a local KBI repository at the Columbia-Presbyterian Medical Center [13,14,15]. Their "infobutton" approach embeds context-sensitive links from their inpatient EHR system to a proprietary collection of KBI resources. This research focuses on the challenge of predicting what questions clinicians might need answers to while using the EHR. By using clinical data from the EHR, combined with the Medical Entities Dictionary (a concept terminology) [16], they were able to suggest clinical topics to the user and link these topics to specific resources in a local KBI repository.

Price et al. have reported similar success in implementing this type of context-sensitive linkage in their SmartQuery project [17]. Using clinical data such as medication lists and ICD-9 codes, a list of MeSH[5] terms is automatically generated for querying to multiple repositories on the Web. In addition, the user has the ability to add terms by clicking on checkboxes next to relevant pieces of clinical data (such as a lab test). These terms are submitted to five KBI repositories on the Web and the results are presented to the user as hyperlinks to the original documents.

It is important to note, however, that these studies were designed to evaluate the *usability* or *efficacy* of these systems in providing clinicians the ability to answer clinical questions at the point of care. While they represent an important preliminary step, they do not address the requirements, from a technical

[5] MeSH stands for Medical Subject Headings and is a hierarchical terminology used to describe clinical concepts.

or standards viewpoint, for integrating the EHR with KBI. Few technical and design details of these implementations have been reported in the literature. I suspect that few are based on industry standards, which is a necessary step for widespread adoption and access to the plethora of online medical resources.

Review of Interoperability Protocols

The digital library community has performed the most significant research on the interoperability of digital resources. The concept of a digital library has been around for decades and many research groups have developed interoperability protocols for sharing the wealth of information residing in them. It is logical then to examine these protocols for their potential in promoting interoperability of KBI in health care.

Among the interoperability protocols developed by these efforts, the following 3 protocols were chosen for a variety of reasons. First, each protocol approaches interoperability from a slightly different technical perspective, each with its own advantages and disadvantages (to be discussed later). Second, they each provide unique functionalities that may make them attractive to different types of health care organizations. Finally, all 3 are, at the moment, active research projects with significant academic or industry momentum. Of the three protocols to be reviewed, Z39.50 is the most widely implemented and generally considered to be the industry standard in this area, so it is appropriate to begin our discussion with it.

Z39.50 Information Retrieval

Background

Z39.50[6] has a long and checkered history in both the information retrieval and digital library research communities [18]. Although it is currently the industry standard for digital library interoperability, it has gone through many revisions over its 20-year history before arriving at its current state of functionality and acceptance. The first official version was ratified in 1988, and subsequent

[6] The full name of the standard is: International Standard, ISO 23950: "Information Retrieval (Z39.50): Application Service Definition and Protocol Specification" The specification can be found at: http://www.loc.gov/z3950/agency/

versions appeared in 1992 and 1995. The standard is now under the auspices of the Library of Congress (LOC) and as of this writing, a draft specification for a new version (2001) has been released. However, since this release is still in the public feedback stage, our discussion will be limited to the current official version (1995).

Z39.50 is widely used throughout the world as a means of searching and exchanging information between applications and databases residing in different locations. It is a very complex and feature-rich protocol, with a high learning curve. Few organizations are able to implement the full standard due to its size and complexity [19]. While providing a comprehensive technical description of the full Z39.50 specification is beyond the scope of this paper, the following is a review of the most important functionalities as they relate to the interoperability of digital resources.

Technical Overview

Z39.50 is based on a client/server model of communication. A Z39.50 ***client*** represents a software application that initiates a request to one or more Z39.50 servers to search and retrieve information from a repository. A Z39.50 ***server*** is a software application that provides access to one or more repositories using a well-defined interface. Most implementations of Z39.50 are focused on the discovery and retrieval of *bibliographic* information, however, the specification allows for any type of content to be searched and retrieved using the protocol.

The Z39.50 protocol consists of 11 facilities, which are groups of operations that implement a specific functionality (it is common for protocols to group operations into categories). A facility consists of a set of predefined operations for communicating between the client and server. For example, the *Initialization* facility specifies those operations related to the establishment of a connection between the client and server. Table 2 lists the facilities along with descriptions of each. It is important to note that not all of the facilities in Table 2 are required. Some are considered optional, such as the *Extended Services* facility.

Z39.50 Facility	Description
Initialization	Establishes connection with servers and sets resource limits.
Search	Initiates search using query syntax and generates result set.
Retrieval	Retreives a set of records from a given result set.
Result-Set-Delete	Requests deletion of server-side result sets.
Access Control	Server initiated authentication check.
Resource Control	Requests status reports of server resources.
Sort	Specifies how a result set should be sorted.
Browse	Access ordered lists such as title and subject metadata.
Explain	Requests information about server's supported services.
Extended Services	Access optional services such as persistent queries.
Termination	Ends client-server session; can be initiated by client or server.

Table 2. Z39.50 Facilities. Specific operations of the protocol are grouped into facilities. Adapted from [19].

Unlike other protocols, Z39.50 is a session-oriented, or stateful, protocol. Once a client initiates a session, a connection is established between the client and server. This connection remains open until either the client or server terminates it using the syntax specified in the *Termination* facility. At that point the session is closed and the client would have to initiate a new session to perform any additional searching. In general, maintaining a session requires significant processing and resources on the part of the server. This is in stark contrast to the stateless HTTP model of the OAI protocol, to be discussed later. One of the advantages to session-oriented management is that queries are performed and result sets are stored on the server side. The client can sort and browse result sets that reside on the server without the need to transfer the entire result set over the network.

Example

Figure 3 illustrates a typical scenario for Z39.50 using a sequence diagram[7], followed by a description of each step in the scenario.

[7] A sequence diagram is a UML (Unified Modeling Language) notation used to illustrate the interaction between multiple components of a software system. The arrows indicate data flow and time progresses from the top to bottom. Dotted arrows indicate return data.

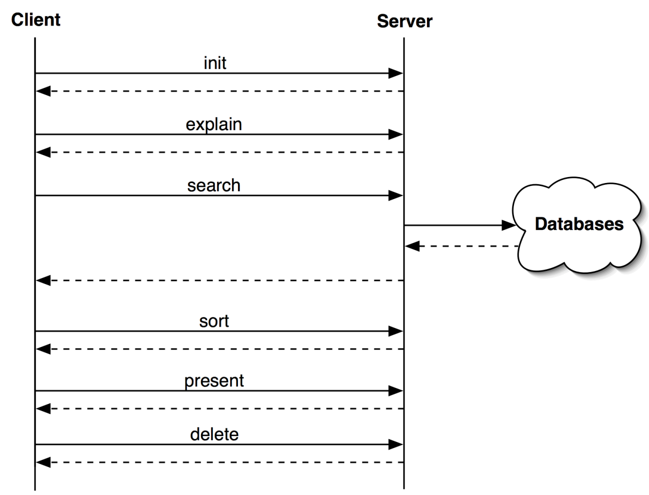

Figure 3. Sequence diagram of Z39.50 client/server interaction. Most of the functionality of the Z39.50 model is provided by the server. Numerous interactions between the client and server are required. (See text for explanation of steps in the scenario.)

The client application sends an ***init*** request to establish a session with the server. The server allocates resources for the session and returns a confirmatory response to the client. The client then sends an ***explain*** request to the server, to obtain information about the server configuration, such as which repositories are available to search. The client then issues a ***search*** request for a specific piece of information, such as "Find all resources with title '*Romeo and Juliet*' and author '*Shakespeare*'". The server would then translate this request into the appropriate database queries and perform the search. The results are then stored on the server as a ***Result Set*** (it is important to note that the results are not returned to the client until the client requests them). The client uses the ***sort*** and ***browse*** facilities to manipulate and examine the result set. It then uses the ***present*** operation to retrieve the desired resources. It is the ***present*** command that actually retrieves the results from the server. Finally, the ***delete*** facility is used to purge the ***Result Set*** from the server's memory.

Summary

Z39.50 is by far the most widely implemented of the interoperability protocols discussed in this paper. Most major library systems have adopted Z39.50 in one form or another, including the Library of Congress (LOC). However, most small organizations may find Z39.50 difficult to implement. One of the protocol's biggest weaknesses is its level of complexity, which creates a high barrier to adoption. Due to the vast number of features and operations provided by the protocol, implementing even a subset of the 11 facilities can be a daunting project. Another weakness is its lack of support for stateless transactions, such as with HTTP. For these reasons, the LOC maintenance agency in charge of Z39.50 recently introduced several initiatives aimed at making Z39.50 more attractive to developers. These initiatives are collectively referred to as ZING (Z39.50 International: Next Generation) [20]. Among the features added are support for Web Services/HTTP models of communication and the use of XML for data representation. The goal of ZING is "to make the intellectual/semantic content of Z39.50 more broadly available and to make Z39.50 more attractive to information providers, developers, vendors, and users, by lowering the barriers to implementation while preserving the existing intellectual contributions of Z39.50 that have accumulated over nearly 20 years" [20]. These initiatives are an important step in lowering the barrier of adoption for Z39.50.

Open Archives Initiative – Protocol for Metadata Harvesting

Background

Unlike Z39.50, the Open Archives Initiative (OAI) is a relative newcomer in the interoperability arena [21]. The initiative was established in 1999 as a result of a desire to extend the successes of the E-Prints initiative[8] to a wider audience and to promote better interoperability between collections of journal articles, research papers, and other forms of scholarly publication. The current paradigm for publishing research papers, particularly in the scientific community, has many weaknesses. Costly subscription rates, slow submission process, publication bias, and inaccessibility to non-subscribers are just a few examples. In light of these weaknesses, the focus of the OAI is to "develop and promote interoperability solutions that aim to facilitate the efficient dissemination of content" [21]. The interoperability protocol developed by the OAI is based on a model of *metadata harvesting* (see figure 4) and is referred to as the *Open Archives Initiative – Protocol for Metadata Harvesting* (OAI-PMH). This paper will focus on Version 2.0 of the OAI-PMH (released June 2002).

---

[8] E-Prints are electronic copies of research papers, both before and after the peer review and journal submission processes. E-Prints are stored in network accessible archives, to promote the timely dissemination of scholarly publications. See http://www.eprints.org for more information.

Technical Overview

In the metadata harvesting model, there are two types of applications: *data providers* and *service providers*. Data providers are the archives of digital content, and will be referred to as repositories. Repositories expose metadata about their content using a simple, well-defined interface specified in the OAI-PMH. Service providers are applications that interface with a repository in order to retrieve metadata about the resources contained in it. Service providers can be any type of application or information system.

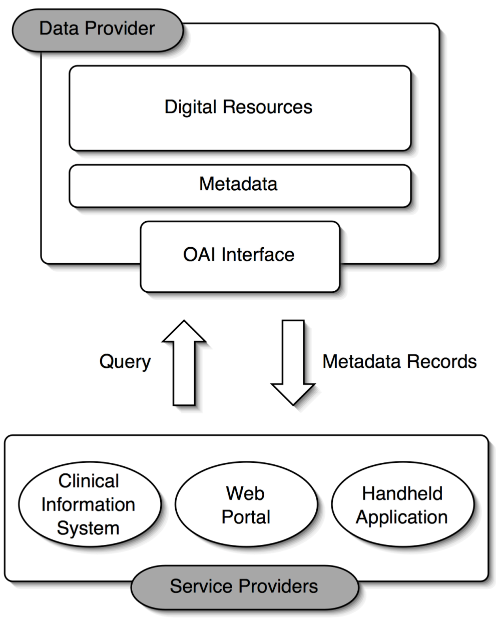

Figure 4. The metadata harvesting model of the OAI. Data providers implement the OAI-PMH interface to expose metadata about their content. Service providers harvest metadata from numerous OAI archives and store the metadata records locally.

Some examples of service providers include web-based digital library portals, handheld applications, and potentially, clinical information systems. In contrast to Z39.50, very little functionality exists in the protocol itself. It is simply a lightweight protocol designed to retrieve metadata records from repositories. Very little processing occurs on the server beyond the searching and returning of metadata records. No sessions are being maintained and no result sets are stored on the server. Because of this simplicity, the protocol is relatively easy to implement with minimal effort and resources.

Metadata

Since the OAI model is metadata-focused, a closer look at the its metadata support is warranted. Each item, or resource, stored in an OAI-compliant repository contains one or more metadata records for each of its digital resources. The purpose of metadata, in general, is to describe the resource with enough detail and structure to allow for efficient and accurate searching of the repository. OAI metadata records are XML-encoded and contain three distinct sections (see figure 5).

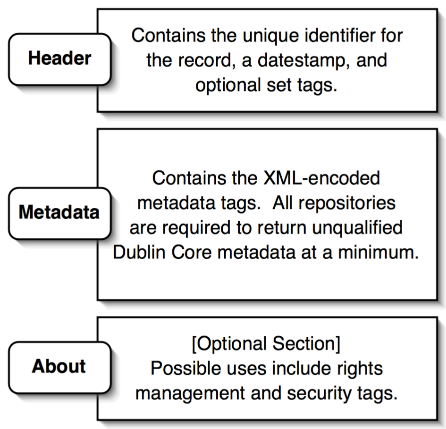

Figure 5. The OAI record structure. The OAI metadata record is an XML file containing three sections.

The *header* portion of the record contains the identifier for the record, a datestamp, and optional tags indicating if this record belongs to any sets (more on sets later). The *metadata* portion of the record contains the descriptive information about the content of the resource (such as title and author). The protocol requires that all repositories support unqualified Dublin Core metadata at a minimum. Repositories are free to support additional metadata standards, such as the MARC standard [22], but they all must support Dublin Core at a minimum. The *about* section of the record is optional and may include information about the record itself, such as rights management or security tags.

Interface

The OAI-PMH contains six verbs to allow communication between the repositories and the service providers. Table 3 describes each of these verbs. OAI verbs are analogous to the operations of the Z39.50 protocol.

OAI Verbs	Description
Identify	Retrieves information about the repository, such as the base URL and the version of the OAI protocol supported.
GetRecord	Retrieves an individual record from the repository given the item identifier and metadata format requested.
ListRecords	Retrieves a collection of records from the repository. Selective harvesting based on sets or dates can be used.
ListIdentifiers	Retrieves a collection of identifiers of records from the repository. Selective harvesting based on sets or dates can be used.
ListSets	Retrieves the set structure of the repository.
ListMetadataFormats	Retrieves the available metadata formats for the repository.

Table 3. OAI Verbs. Six OAI verbs are used to query repositories. Each query returns an XML-encoded byte stream.

Service providers communicate with repositories using standard HTTP POST and GET methods. By choosing a standard like HTTP for submitting queries, the protocol leverages the many toolkits available for creating HTTP-based applications, and reduces the complexity of such applications. An example query for retrieving metadata records from a repository would look something like this:

`http://medir.ohsu.edu/oai-servlet?`**`verb=ListRecords`**`&`**`from=2001-09-26`**

This query will retrieve all records from the specified date in the form of an XML byte stream. The *from* argument in the HTTP query illustrates the protocol's support for *selective harvesting*. The protocol allows for both date-based and set-based queries. What types of sets are used is up to each repository. The protocol does not specify a common set structure, but those repositories implementing sets must return a description of the set hierarchy upon a query with the *ListSets* verb.

Since the protocol is based on HTTP mechanisms of communication, it is inherently stateless. Each interaction between a service provider and repository concludes after each request and response. Unlike other protocols discussed, OAI-PMH has no support for sessions. To accommodate queries with extremely large result sets (imagine issuing a *ListRecords* request to the Library of Congress) the protocol utilizes *resumption tokens* to handle large transactions. For large result sets, the repository may return a resumption token (a unique string value) to the service provider along with a truncated result set. The service provider can then use the resumption token to issue a request for the remainder of the result set. This allows the protocol to handle large amounts of metadata records without the overhead of supporting sessions.

Example

Since OAI has its roots in the digital library community, we will use a web-based digital library portal application as our example. The purpose of this application is to provide users with a single location for searching the contents of multiple off-site library collections. In this case, the portal application acts as a service provider and interfaces with many off-site library collections using the OAI-PMH. Initially, the portal application would issue a *ListRecords* request to each off-site library, obtaining all of the metadata records for the resources at each of these locations. Since each metadata record contains a datestamp, the portal application can make regular queries to these repositories for any new or updated records that have been added or changed since the last query.

The portal application provides a search interface to its locally stored collection of metadata records, allowing the user to review the descriptions of the resources from all of the libraries, without having to make a network connection. Since each metadata record contains a unique identifier for the resource it describes, the user can potentially retrieve a specific digital resource from the repository which houses the document. This functionality, however, is not part of the OAI-PMH and would depend on the access permissions of the resource at the off-site library. A real-world example of this type of application is called my.OAI[9]. My.OAI is a framework for building portal applications based on the OAI protocol that harvest metadata from multiple repositories and present the records to the user in a single unified interface.

Summary

Since Version 2 of the OAI-PMH was released in June of 2002, it has gained significant momentum in the digital library community. Hundreds of repositories have implemented the protocol

[9] http://www.myoai.org

including the California Digital Library (CDL), the Digital Library of Information Science and Technology (DLIST), and many university library collections. With respect to medical literature, BioMedCentral[10], a repository of peer-reviewed biomedical literature, has recently added an OAI interface to its collection. This rapid adoption of the OAI-PMH is mainly due to the simplicity of the protocol and the metadata harvesting model. Developers can implement clients and servers based on OAI in a matter of days. Contrast this to Z39.50, which can take months for development, and even then implementing only a subset of the 11 facilities in the specification.

Simple Digital Library Interoperability Protocol (SDLIP)

Background

The Simple Digital Library Interoperability Protocol (SDLIP, pronounced S-D-Lip) is a collaborative research effort between Stanford University, UC-Berkeley, UC-Santa Barbara, the San Diego Supercomputer Center (SDSC), and the California Digital Library project (CDL) [23]. The development of SDLIP is part of the National Science Foundation's 2nd Digital Library Initiative. Similar in concept to OAI and Z39.50, SDLIP is a protocol for requesting and retrieving information from heterogeneous information sources, such a digital libraries and web databases. However, SDLIP has several unique features that set it apart from the other two. These include:

1) Support for both stateful and stateless transactions.
2) Support for CORBA (Common Object Request Broker Architecture).
3) Server-side load balancing.

SDLIP is not as widely implemented as either Z39.50 or OAI, but there is significant interest in the research community.

Technical Overview

Like OAI and Z39.50, SDLIP is based on a client/server model of operation, where the server component is referred to as a Library Service Proxy (LSP), or SDLIP proxy. The proxy is responsible for interacting with repositories and providing an SDLIP interface to the client applications. Figure 6 illustrates the SDLIP architecture. Since most repositories are currently using a variety of different

[10] http://www.biomedcentral.com

interfaces to their collections, such as Z39.50 or proprietary web interfaces, the purpose of the SDLIP proxy is to translate queries from the native interface to the SDLIP interface. This is unique to the protocols discussed so far, in that the consortium recognizes the investments made in other protocols, and provides proxies to work in conjunction with them.

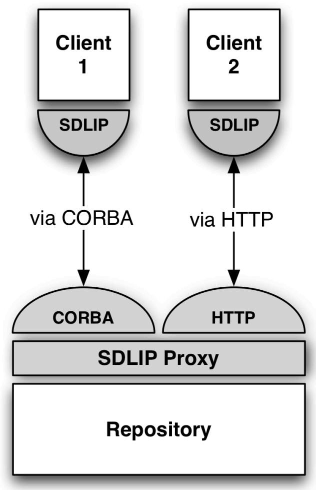

Figure 6. The SDLIP architecture. SDLIP provides a common interface to the repository and supports both CORBA and HTTP modes of communication. Adapted from [23]

SDLIP has built-in support for CORBA (Common Object Request Broker Architecture). CORBA is a language and platform-independent technology framework for allowing computer applications to work together over networks. CORBA is a technology that has gained a lot of momentum in the industry over the past decade, primarily for its robust functionality as a distributed object technology. In addition to CORBA, SDLIP provides support for the HTTP transport model.

The SDLIP architecture is composed of three components, called interfaces, each encapsulating a specific kind of functionality. This is similar to Z39.50's grouping of operations into facilities. Table 4 describes the three interfaces.

SDLIP Interface	Description
Search	This interface contains the operations needed for a client application to submit queries to a repository.
Result Access	This interface allows the client to access any result sets maintained by the Library Service Proxy.
Source Metadata	Allows clients to query a repository for information such as what collections it contains.

Table 4. SDLIP Interfaces. Only the search interface is required in SDLIP.

The Search interface is the only required interface of the three. The Result Access interface is only necessary if the server is going to support sessions, in which case the client will interact with this interface to retrieve result sets stored on the server. The Source Metadata interface is analogous to the *Explain* facility of Z39.50, and the *Identify* verb of OAI. Its function is to provide descriptive information about the repository, such as what collections are available for searching.

Summary

Being primarily an academic research effort, SDLIP has not been widely implemented at the time of this writing. However, the institutions involved with its development have demonstrated several large-scale pilot projects. One such project is the development of a web-based browsing application for the content of dozens of libraries in the state of California. SDLIP's support for CORBA will likely attract further attention from the developer community, especially those sites with an investment in CORBA technology.

Comparison of Interoperability Protocols

Unfortunately, there is no quantitative method for comparing the technical merits of interoperability protocols. Features of a protocol that work well in one environment may be ill suited for another. Also, any comparison of existing applications that utilize these protocols would be more of a comparison of the specific implementations than of the protocols themselves. At best a subjective comparison can be performed. Paepcke at al. have described a set of criteria for the subjective analysis of interoperability protocols [24]. We will draw on these criteria as we summarize the advantages and disadvantages of each protocol with respect to the interoperability of KBI in health care. Table 5 provides a listing of the major functionalities supported by each of the protocols.

Feature	Z39.50	OAI-PMH	SDLIP
Metadata Formats	Flexible	Flexible (DC required)	Flexible
Formal Standard	Yes	No	No
Query Syntax	Common Command Language	Proprietary XML	Flexible
Stateless Transactions	No	Yes	Yes
Support for Sessions	Yes	No	Yes
HTTP Support	Yes (with ZING)	Yes	Yes
SOAP Support	Yes (with ZING)	No	No
CORBA Support	No	No	Yes

Table 5. Feature comparison of Z39.50, OAI-PMH, and SDLIP. This table indicates those features *directly* supported by each protocol.

There are several major differences between these protocols that should be clarified. First, the OAI protocol uses a model of *metadata harvesting*. This is distinctly different from the client/server model of Z39.50 and SDLIP. Whereas most of the search functionality of the Z39.50 and SDLIP protocols is provided by the server component, the OAI protocol returns *metadata records* to the client application. Only limited searching (set-based and date-based) is performed by an OAI repository. More detailed searching needs to be performed by the client. For example, in order to find resources on the topic of "Diabetes Mellitus" using the OAI model, you would first need to retrieve a collection of metadata records from a repository, then search those records on the client side by examining the metadata field representing *subject*.

Second, OAI and SDLIP are designed to be both *scalable* and *portable*. In fact, SDLIP clients exist for a variety of different devices, including Palm and PocketPC handhelds. On the other hand, Z39.50 is not known to be either scalable or portable, although specific implementations will vary. Because Z39.50 applications expends a significant amount of resources to maintain a session, such a multiple result sets, performance drops as the number of simultaneous users increases. SDLIP, which can also be session-oriented, avoids this pitfall by providing support for dynamic load balancing. For example, if the number of simultaneous users reaches a critical level, the SDLIP server can switch from stateful to stateless communication. The LOC is attempting to address some of the shortcomings of Z39.50 in the ZING initiatives.

Likelihood of adoption of these protocols is difficult to determine. The choice of interoperability protocol will depend on a lot of factors, such as an organizations investment in specific technology. For example, if an organization has a significant investment in CORBA technology, in terms of software and developer expertise, SDLIP may be an attractive choice since it supports this technology. For small organizations, the OAI protocol may be attractive due to its simplicity and ease of implementation (it is not uncommon to implement an OAI-compliant repository in a matter of days). Of course, client application developers will likely choose to implement the interoperability protocols use by the KBI repositories of interest.

These are just a few of the technical differences between the protocols. More can be found in the previous sections of this paper. These technical differences, as we will see, are only part of the overall picture. There are many technical issues surrounding interoperability of KBI in health care unrelated to the interoperability protocol, but play an equally important role. In addition, non-technical issues, such as standards-compliance and intellectual property, must be considered as well. These issues will be discussed in the following sections.

Discussion

While this paper has focused on the technical details of interoperability protocols for accessing KBI repositories, these protocols are simply one component of the interoperability problem. In order to seamlessly integrate KBI with the EHR at the point of care, several other technical issues need to be addressed alongside interoperability protocols. In addition to these technical issues, there are many organizational factors that play an important role, such as the business needs of the stakeholders. While these organizational issues are beyond the scope of this paper, we will discuss several of the technical issues that relate to the use of interoperability protocols to access KBI. To put the technical issues into some context, we will discuss them in light of a clinical scenario involving the information needs of a physician at the point of care. This scenario will form the basis for the discussion of the key areas of interoperability, and wherever possible, shed light on the differences between OAI, Z39.50, and SDLIP.

Clinical Scenario

A 35 year old woman visits her primary care doctor because she started experiencing headaches and nausea a few days ago. The patient suffers from hypertension, high cholesterol, and obesity. She is taking many medications for these conditions and was recently started on a new medication, *metoprolol*, for her hypertension. The physical exam and history are unremarkable other than the chief complaints.

The physician has many questions about this case.

- Could her headache and nausea be an adverse reaction to *metoprolol*?
- Could there be a drug-drug interaction between *metoprolol* and one of her other medications?
- If this is a new onset migraine, are there any guidelines available for working this up?

Clinical Context

In order to facilitate the use of KBI at the point of care, it is helpful to present the physician with suggested questions and resources based on the *clinical context* of the patient record. Certain areas of the EHR, such as medication and problem lists, can be used to generate queries automatically and present suggestions to the clinician within the EHR. For example, if a new *problem* is entered into the EHR, the clinician can be presented with KBI resources related to that diagnosis, like systematic reviews or clinical guidelines. This context-sensitive linking has been studied by several researchers and shown to be an effective tool [13,14,15,17].

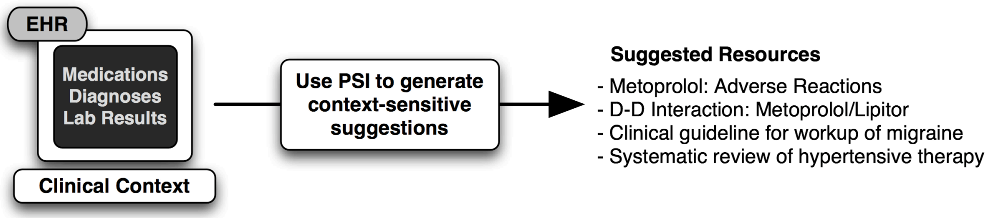

Figure 7. Context-sensitive linking within the EHR.

In our example scenario, suggested resources might include *adverse reactions to metoprolol*, *drug-drug interactions between metoprolol and her other medications*, *clinical guidelines for migraines*, or *systematic reviews of hypertensive therapy*. Figure 7 illustrates this process. The physician is free to choose one of these context-sensitive links or devise her own search strategy using a standard search form. Context-sensitive suggestions would be a function of the EHR, so interoperability protocols would have little to do with this step in the scenario. Only with respect to how the query is packaged and transmitted, do interoperability protocols come into play.

Access to KBI

Once a physician recognizes a need for KBI, and chooses which question to pursue, the query needs to be transmitted to one or more KBI repositories using the appropriate transport mechanism and syntax. Figure 8 illustrates this step in the scenario. If each repository accepts a different syntax for queries, that is, has a different interface, it would be too resource intensive to support many KBI repositories in your application. This is the purpose of the interoperability protocols we have discussed. If every KBI repository adopts a standard interoperability protocol as its interface, it would be trivial to query large numbers of repositories.

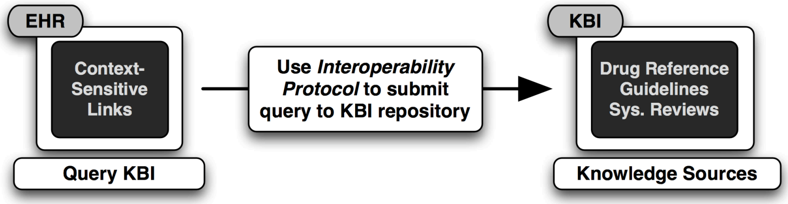

Figure 8. Utilizing interoperability protocols for querying KBI.

Each of the protocols discussed in this paper uses a different syntax and transport model for connecting with KBI repositories. Z39.50 uses the ANSI standard CCL (Common Command Language) and SDLIP uses a flexible scheme. Because OAI uses the metadata harvesting model, queries are usually performed against a local collection of metadata records, not by dynamic querying of a repository. One advantage to this model is that a network connection is not required during the initial phase of the KBI retrieval process. A disadvantage to this approach is the requirement for the EHR application developers to implement searching algorithms against local collections of metadata records. With Z39.50 and SDLIP, it is up to the repository to provide this functionality.

Metadata

One of the most important components of the successful retrieval of KBI is the method by which repositories describe and search their collections. Most use some form of metadata for this purpose, as we described earlier. Figure 9 illustrates the process of searching metadata records from a repository. This is an area that needs considerable customization, beyond what has already been done in the digital library community, to accommodate the information needs of clinicians. Some informatics research has investigated the customization of existing metadata formats to better accommodate the structure and content of the medical KBI [11].

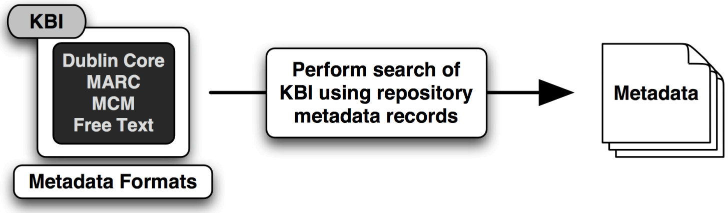

Figure 9. Metadata formats for KBI.

Digital Rights Management

As discussed earlier, DRM is an important component of interoperability because it protects the intellectual property of the KBI vendors. Fortunately, this is an area that has seen considerable progress in the past couple of years. This is in part due to the indiscriminate and illegal copying of music files over the Internet. XML has played an important part as well, and most of the proposed DRM strategies are implemented in XML. Since all of the interoperability protocols support XML, it should be possible to layer these XML-based DRM solutions on top of an interoperability protocol. Of course, this has not been researched extensively.

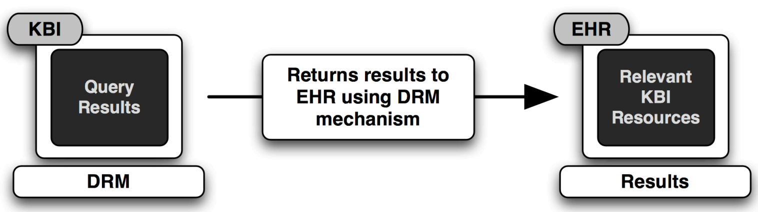

Figure 10. Digital Rights Management.

Figure 10 illustrates the use of DRM in our scenario. The results from our query are returned to the EHR along with DRM information specifying the access privileges of the resources in the result set. Ideally, the DRM technology would not only allow the EHR to limit use of the resource to certain individuals, but prevent certain operations on those resources, such as printing, copying, or even viewing the document more than a specified number of times.

Conclusion

The rapid proliferation of KBI repositories on the Internet, along with recent technological advances such as XML, has provided an opportunity to recognize the long-standing informatics goal of integrating relevant KBI with the EHR at the point of care. While large-scale interoperability projects demonstrating this integration do not yet exist, several researchers have demonstrated small-scale integration of KBI with positive results. All of the necessary technologies exist for this integration to occur. The primary challenge will be overcoming the many organizational and political issues surrounding interoperability – such as intellectual property protection and finding a viable business model for the widespread distribution of medical KBI.

The solution to the interoperability of KBI in health care is fundamentally dependent on the acceptance and adoption of standards by the health care IT industry. Without industry wide support for the interoperability protocols discussed in this paper, integration solutions will remain proprietary and ad hoc. Z39.50, OAI, and SDLIP represent significant advances in digital library interoperability and should be researched further as an effective interoperability layer between knowledge-based information systems and the electronic health record.

References

1. Lindberg DA, Siegel ER, Rapp BA, Wallingford KT, Wilson SR. *Use of MEDLINE by Physicians for Clinical Problem Solving*. JAMA. 1993 Nov 3;270(17):2053-4.

2. Stearns MQ, Price C, Spackman KA, Wang AY. *SNOMED clinical terms: overview of the development process and project status.* Proceedings / AMIA Annual Symposium, 2001: 662-6.

3. Kurtz M. *HL7 version 3.0: a preview for CIOs, managers, and programmers*. Journal of Healthcare Information Management. 16(4):22-3, 2002 Fall.

4. Smith, Richard. *What clinical information do doctors need?* BMJ, 1996. 313:1062-1068.

5. Gorman PN, Helfand M. *Information Seeking in Primary Care: How physicians choose which clinical questions to pursue and which to leave unanswered*. Medical Decision Making 1995;15:113-119.

6. Gorman PN. *Information Needs of Physicians*. Journal of the American Society on Information Science. 1995;46:729-736.

7. 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): p. 181-200.

8. Bidgood WD Jr. *Clinical Importance of the DICOM Structured Reporting Standard*. Int J Card Imaging, 1998 Oct;14(5):307-15.

9. Hersh W, Klopfer K, Bhupatiraju R, Pruzan M (2003). *A Model for Interoperability of Knowledge-Based Information.* [Submitted for Publication, AMIA]

10. Darmoni SJ, Thirion B, Leroy JP, Douyere M. *The use of Dublin Core metadata in a structured health resource guide on the internet.* Bulletin of the Medical Library Association. 89(3):297-301, 2001 July.

11. Malet, G et al. *A Model for Enhancing Internet Medical Document Retrieval with "Medical Core Metadata"*. Journal of the American Medical Informatics Association, 1999. 6:183-208.

12. Erickson, JS. *Fair Use, DRM, and Trusted Computing*. Communications of the ACM. April 2003, 46(4):34-39.

13. Cimino J.J. et al. *Theoretical, empirical and practical approaches to resolving the unmet information needs of clinical information system users*. Proc AMIA Symp, 2002: p. 170-4.

14. Cimino, J.J., G. Elhanan, and Q. Zeng. *Supporting infobuttons with terminological knowledge*. Proc AMIA Annu Fall Symp, 1997: p. 528-32.

15. Cimino, J.J., *Intranet technology in hospital information systems*. Stud Health Technol Inform, 1997. 45: p. 102-9.

16. Cimino JJ. *Terminology Tools: State of the Art and Practical Lessons*. Methods Inf Med. 2001;40(4):298-306.

17. Price, S.L., et al. *SmartQuery: Context-Sensitive Links to Medical Knowledge Sources from the Electronic Patient Record*. Proc AMIA Symp, 2002: p. 627-31.

18. Lynch, CA. *The Z39.50 Information Retrieval Standard, Part I: A Strategic View of Its Past, Present and Future.* D-Lib Magazine. April 1997.

19. Witten, I.H., *How to Build a Digital Library*. Morgan Kaufmann Series in Multimedia Information and Systems. 2003, San Francisco, CA: Morgan Kaufmann Publishers.

20. *Z39.50 International: Next Generation*. Available at http://www.loc.gov/z3950/agency/zing/.

21. Lagoze, Carl and Herbert Van de Sompel, *The Open Archives Initiative: Building a Low-barrier Interoperability Framework*. Available at: http://www.openarchives.org.

22. MARC Standards, Library of Congress. Available at: http://lcweb.loc.gov/marc/marc.html

23. *SDLIP: The Simple Digital Library Interoperability Protocol*. Specification available at: http://www-diglib.stanford.edu/~testbed/doc2/SDLIP/.

24. Paepcke, A., et al., *Interoperability for Digital Libraries Worldwide,* Communications of the ACM. 1998, 41(4): 33-42.