Standardization Approach for Information Retrieval in WAN

Krasimir Trichkov

Abstract: *This paper aims to present an ANSI/NISO Z39.50 protocol for distributed searching in library applications. It specifies Z39.50 client and Z39.50 server behavior for search and retrieval across online library catalogues. The paper examines the potential of Z39.50 to enable new methods of data creation and exchange in library networks. The essence of SRW, the Search/Retrieve Webservice, XML oriented protocol for business communication is explained.*

Key words: *Z39.50, SRW, Internet, Database, Zebra, Zap, Php/Yaz.*

INTRODUCTION

A lot of activities in today's dynamic world concern effective information exchange. A large amount of information in an organization leads to the need of making possible to extract the necessary data and to access them everywhere and any time. To facilitate information retrieval across the diverse collections of data resources now available, a non-proprietary standards-based communications protocol for distributed searching which is independent of database and computer environment. Z39.50 is a mature standard, widely implemented in the library community. It is beginning to solve real problems, not just for libraries, but also for other collecting agencies such as art galleries, museums and archives. And like the dynamic network environment in which it is used, the standard is evolving to meet the changing needs of information creators, providers, and users.

DESCRIPTION OF THE PROTOCOL

Z39.50 or ISO23950 is a protocol enabling search of and retrieval from remote databases. Its full name is ANSI Z39.50-1995, *Information Retrieval (Z39.50) Application Service Definition and Protocol Specification* [1]. The standard defines specifications for protocols (rules and procedures) to promote communication between different systems. Z39.50 is one of many NISO standards that address the application of both traditional and new technologies to information management, retrieval, and storage. The goal in developing and using technical standards in information services, libraries, and publishing is to make information systems easier to use and less expensive to operate. Z39.50 is a computer-to-computer communications protocol designed to support searching and retrieval of information (full-text documents, bibliographic data, images, multimedia) in a distributed network environment. Based on client/server architecture and operating over the Internet, the Z39.50 protocol is supporting an increasing number of applications. Z39.50 supports open systems, which means it is nonproprietary, or vendor independent. Also next generation Z39.50 protocol called SRW (Search and Retrieve Web Service), building on Z39.50 along with web technologies, recognizes the importance of Z39.50 (as currently defined and deployed) is available for business communication now. In this way can be developed so called network of e-services [2]. The body responsible for Z39.50 is ANSI/NISO [3].

A. Basics

The core functions of Z39.50 relate to searching and retrieving information from databases stored on multiple host sites. The protocol "specifies data structures and interchange rules that allow a client machine (called an 'origin' in the standard) to search databases on a server machine (called a 'target' in the standard) and retrieve records that are identified as a result of such a search [4].

The protocol confines itself to interactions between the client and server machines, and does not address interaction between a human user and the client machine or between the target machine and its databases. The standard is designed to facilitate

interoperability between computer systems. The communication described in the standard is connection-oriented and stateful: that is, the origin initiates a session with the target and the connection is maintained until the association is terminated.

In an implementation, the origin and target convert their local forms of messages and responses to and from Z39.50 'language'. This means an origin can maintain a consistent user interface for searching targets which support Z39.50, because the client machine's searching syntax can be mapped into Z39.50 queries. In this way, the origin extends the local interface to search external targets. On the target or server side, this requires considerable conversion because the incoming Z39.50 query must be mapped to retrieval mechanisms and vice versa (Figure 1).

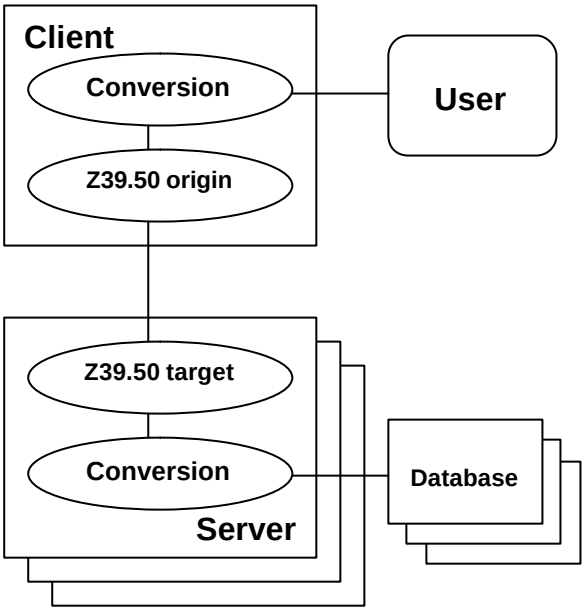

Figure 1 Z39.50 client architecture

The standard does not directly support the broadcasting of searches to multiple servers, but a client can open Z39.50 sessions with multiple servers either sequentially or simultaneously. Manipulating multiple result sets to remove duplicates and ensure a uniform presentation to the user also falls outside the scope of the protocol.

Web-based search and retrieval applications need Z39.50 for the same reason as proprietary applications - to avoid the proliferation of interfaces to the target databases. The Web is a static collection of html documents stored on http servers. Special programs using scripting languages and compiled modules are needed to deliver search and retrieval functionality. In server-based implementations, the HTTP/ Z39.50 gateway resides on an HTTP server as in the diagram below [5]. Browser-based implementations also exist which require Java or Active X applets to be downloaded to the user's machine.

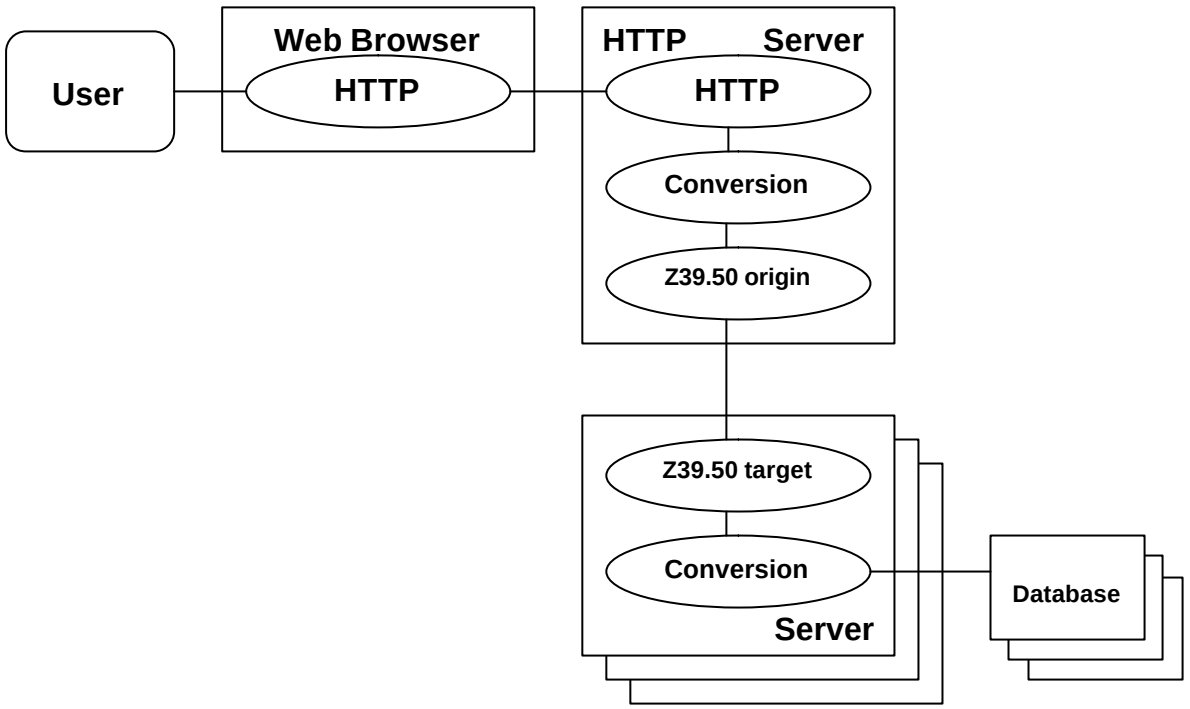

Figure 2 Web/ Z39.50 gateway architecture

As databases differ considerably in structure and indexing methods, the protocol employs a common, abstract model for describing databases. The model requires a "schema" or abstract record structure to be defined for each database, composed of "elements" such as author, title, and date last modified. Access points are also defined for each searchable element or group of elements. However, Z39.50 should not be interpreted as a database indexing standard. In each implementation, the target databases must be mapped to the Z39.50 database model to enable communication between origin and target.

B. Search and Retrieval Facilities

The Search and Retrieval facilities are the core functions of the standard. A search request can be made to one or more databases at a target system and must contain a query. The group of records retrieved as the result of a query is called a *result set*.

When a database is searched, the client passes a query to the server. The query contains search terms (e.g., terms that the user has identified to be matched against access points in the database) and attributes of those search terms (e.g., specifying the terms as an "author" or "title," specifying if the terms are to be "truncated," etc.). Queries can include different attribute types. For example, if a user wants to search for an author's name, a "use" attribute specifies the search term as "author." If the user wants to search for all books published after a certain date, a "use" attribute specifies the search term is a "date of publication" and a "relation" attribute specifies that the user wants all dates of publication "greater than" a particular date. ANSI/NISO Z39.50 enumerates these attribute types and their values in registered attribute sets. Standardized and mutually recognized attribute sets allow implementers a common basis for intersystem communication. After the server executes a search of a database, it creates a result set consisting of those records that match the criteria of the query. Clients can request that servers return those records from a result set, or they can issue additional searches that further qualify a result set or use result sets as arguments in subsequent searches.

Z39.50 registers standardized element set names and record syntaxes to support client/server communication for this aspect of information retrieval.

Based on the requirements of Z39.50 implementers and users, the current draft contains a number of new features and enhancements:

- Sort: allows the client to request that the server sort and order a result set according to client-supplied criteria.
- Scan: provides the client with the ability to scan lists of terms (i.e., access point values) available from a database or a group of databases.
- Extended Services: defines a set of tasks or operations that the client may request the server to perform, such as: saving a result set for later use, executing search queries on a periodic schedule, exporting the records in a result set, ordering documents, and requesting printing.
- Explain: allows the client to obtain information about the implementation of a server system including the databases available for searching, restrictions on the use of the server, hours of operation and availability and a broad range other important information the client can use to facilitate effective information retrieval with a particular server.
- Segmentation: provides for the effective transfer of parts of a record when the entire record exceeds the transfer size negotiated between the client and server; this is especially critical for image databases and multimedia services.
- Proximity Searching: a query type that enables a client to specify proximity searching.

C. Records Information

Records pass through three different states during processing in the system (Zebra system).

- When records are accessed by the system, they are represented in their local or native format. This might be SGML or HTML files, News or Mail archives, MARC records [6]. If the system doesn't already know how to read the type of data you need to store, you can set up an input filter by preparing conversion rules based on regular expressions and possibly augmented by a flexible scripting language.
- When records are processed by the system, they are represented in a tree-structure, constructed by tagged data elements hanging off a root node. The tagged elements may contain data or yet more tagged elements in a recursive structure.
- Before transmitting records to the client, they are first converted from the internal structure to a form suitable for exchange over the network - according to the Z39.50 standard.

The Zebra system currently supports two fundamental types of records: structured and simple text (sgml, html, text, etc..). It is designed to support a wide range of data management applications. The system can be configured to handle virtually any kind of structured data. Each record in the system is associated with a *record schema*, which lends context to the data elements of the record. Any number of record schemas can coexist in the system. Although it may be wise to use only a single schema within one database, the system poses no such restrictions.

Raw text is just that, and it is selected by providing the argument *text* to Zebra. Structured records are all handled internally using the basic mechanisms described in the subsequent sections. Zebra can read structured records in many different formats.

Although input data can take any form, it is sometimes useful to describe the record processing capabilities of the system in terms of a single, canonical input format that gives access to the full spectrum of structure and flexibility in the system. In Zebra, this canonical format is an "SGML-like" syntax.

The keywords surrounded by <...> are *tags*, while the sections of text in between are the *data elements*. A data element is characterized by its location in the tree that is made up by the nested elements. Each element is terminated by a closing tag - beginning with </, and containing the same symbolic tag-name as the corresponding opening tag. The

general closing tag - </> - terminates the element started by the last opening tag. The structuring of elements is significant.

The first tag in a record describes the root node of the tree that makes up the total record. In the canonical input format, the root tag should contain the name of the schema that lends context to the elements of the record. Zebra allows providing individual data elements in a number of *variant forms*. Examples of variant forms are textual data elements, which might appear in different languages, and images, which may appear in different formats or layouts. Next ?able shows variant of record based on GILS (Government Information Locator Service) [7], Amico [8] and Dublincore [9] standards.

TABLE 1. Example of record

```
<gils>
<Title> Dream </Title>
<Creator> Angelina Stancheva </Creator>
<Contributor> K.Trichkov </Contributor>
<Date> 10.10.2004 </Date>
<Description> Dream, 1998 </Description>
<Identifier> D000 </Identifier>
<Type> Image </Type>
<Language > Bg </Language >
<Subject> Tapestry </Subject>
<Publisher> ICCS </Publisher>
<Format> TXT, JPEG </Format>
<Source> http://www3.iccs.bas.bg </Source>
<Relation> http://www3.iccs.bas.bg </Relation>
<Coverage>Contemporary Bulgarian Art </Coverage>
<Rights> UBA </Rights>
</gils>
```

Converting records from the internal structure to exchange format is largely an automatic process. Currently, the following exchange formats are supported: GRS-1; XML; SUTRS; ISO2709; Explain; Summary; SOIF [10].

D. Software components

Zebra - Zebra is a fielded free-text indexing and retrieval engine with a Z39.50 fronted. Zebra is a high-performance, general-purpose structured text indexing and retrieval engine. It reads structured records in a variety of input formats (eg. email, XML, MARC. Zebra supports large databases (more than ten gigabytes of data, tens of millions of records). It supports incremental, safe database updates on live systems.

ZAP - ZAP is a module (to Web servers), which allows you to build simple WWW interfaces to Z39.50 servers. ZAP hides most of the complexity of session management, parallel searching. The integration of system into the popular Web servers offers several advantages to the operators and users of the software, including simplified maintenance of the Module, and improved performance.

PHP/YAZ - This extension offers a PHP interface to the YAZ toolkit that implements the Z39.50 protocol for Information Retrieval. With this extension its easily to implement a Z39.50 origin (client) that searches or scans Z39.50 targets (servers) in parallel.

This is free software [10] that can work on various operating systems and various Web Servers.

SRW - NEXT GENERATION Z39.50

SRW is "an XML-based protocol designed to be a low-barrier-to-entry solution for searching and other information retrieval operations across the internet [11]. It uses existing, well tested, and easily available technologies, such as URI, XML, SOAP, HTTP, and XPath. All SRW records are transfered in XML [12]. Record schemas used in SRW

include Dublin Core, Onix, MODS, and MarcXml. Support for Dublin Core is strongly encouraged; other schemas can be defined locally. The protocol has two ways that it can be carried, either via SOAP (Search and Retrieve Web Service) or as parameters in a URL (Search and Retrieve URL Service) [13]. Other transports would also be possible, for example simple XML over HTTP, but these are not defined by the current standard. As SRU does not carry the request in an XML form, we talk about request parameters rather than elements within a request XML schema.

The SRW Initiative, building on Z39.50 along with web technologies, recognizes the importance of Z39.50 (as currently defined and deployed) for business communication. While SRW focuses on getting information to the user, building on Z39.50 semantics enables the creation of gateways to existing Z39.50 systems. SRW defines a web service combining several Z39.50 features, most notably, the Search, Present, Sort and Scan Services.

CONCLUSION AND FUTURE WORK

The essence and functional possibilities on communication protocols Z39.50 and Z39.50 Next Generation were presented. Definite are special futures of the protocol and its application for information search in distributed databases. Definitely are software components of the protocol. Proposed the decision for works with heterogeneous databases (architecture for searching in distributed databases) using Z39.50 protocol. The protocol is platform and software independent. As a future work is the problem for optimization of developed searching services.

REFERENCE

[1] http://www.ansi.org

[2] Stoilov T. and K.Stoilov (2003). Network of e-services. Proceedings of the International Conference on Computer Systems and Technologies, COMPSYSTECH'2003, Sofia, Bulgaria, p.IIIA16.1-IIIA16.6.

[3] http://www.niso.org

[4] http://sunsite.anu.edu.au

[5] http://www.nla.gov.au

[6] http://www.loc.gov

[7] http://www.gils.net

[8] http://www.amico.org

[9] http://dublincore.org

[10] http://www.indexdata.dk

[11] Stoilov ?. (2002). XML technology in Internet applications. "Automatica and informatics", ? 4, Sofia, Bulgaria, p.25-28.

[12] Tsenov M. (2003). Internet databases using SOAP protocol and XML standard. XXXVIII International Scientific Conference on Information, Communication and Energy Systems and Technologies, Technical University-Sofia, Bulgaria, p.299.

[13] Ivanova E. (2003). Application of Distributed Search in Databases for Web Services, International conference ICEST'03, p.291-294.

ABOUT THE AUTHOR

Assistant Prof. M.Sc. Eng. Krasimir Trichkov, Institute of Computer and Communication Systems – Bulgarian Academy of Sciences, Acad. G. Bonchev bl.2, 1113 Sofia, Bulgaria, E-mail: krasi@hsi.iccs.bas.bg