Resource sharing protocol Z39.50: A Bird's eye view

By
S. Siva Chidambaram
NCAER LIBRARY, New Delhi

[Abstract : This article describes what is Z39.50?, and its genesis and development, structure. It highlights the basic function with diagrammatic representation. Brief about record syntaxes and definition. Finally I conclude with advantage and its disadvantages of this tool]

INTRODUCTION

Resource sharing development in the Library and Information Science field we had come across Inter Library Loan, Union Catalogue. OPAC (Online Public Access catalogue), Local Library Networks and its Associations Etc. Then we reach the Universal formats like Common Communication Format (CCF) and MARC. But in the Internet era Online is not a Profound word or work in the Library. World Wide Web has scanned photos and museum documents and manipulated heterogeneous data through HTML and JAVA Scripts. In this Networked environment, Since the early 1980s libraries have been working to implement the basic component layers of the OSI (Open system interconnection) Communication protocols. OSI Comprises a complex set of communication protocols designed to enable different systems to communicate across the network. The Libraries thinks to design software that enables a user from Library A to formulate a search using the familiar commands of its own online catalog excute the search across the network on Library B's online catalog. That has derived the protocol suit Z39.50. Carson and Frivalde (1993) describe one of the first such Z39.50 implementation. Then Michael and Hemebush (1995) provide an excellent analysis of the development of Z39.50 protocol and its implementation in Libraries.

This article highlights the development of Z39.50 structure, function and how effectively implement in library functions. The aim of this article is to disseminate the Technology know how to our professionals through NACLIN 99.

WHAT IS Z39.50?

ANSI/NISO Z39.50-1995 (ISO 23950), is one of a set of standards produced to facilitate the interconnection of computer systems. The standard specifies formats and procedures governing the exchange of messages between a client and server, enabling the user to search remote databases, identify records which meet specified criteria, and to retrieve some or all of the identified records and is concerned, in particular, with the search and retrieval of information in databases. One of the major advantages of using Z39.50 is that it enables uniform access to a large number of diverse and heterogeneous information sources. [Pierre 1997].

GENESIS AND DEVELOPMENT OF Z39.50

Computers and telecommunications networks offer the potential to share data in electronic format. The challenge has been to develop a mechanism that would standardize communication between existing computer systems. NISO responded to this challenge by establishing a Standards Committee in 1979 to work on an information retrieval protocol. This work culminated in Z39.50-1988. Also beginning in the late 1970s funding by the Council on Library Resources (CLR) supported the development of an experimental protocol as part of the Linked System Project (LSP) for searching bibliographic databases and transferring records among the Library of Congress, the Online Computer Library Center (OCLC), the Research Libraries Group (RLG), and the Washington (now Western) Library Network (WLN). This Linked Systems Protocol laid the groundwork for the Z39.50 protocol.

The name "Z39.50" come from the fact that the NISO, The ANSI accredited standard development organiztion serving Libraries, publishing and Information services, was once the Z39 Committee of ANSI, NISO standards are numbered sequentially and Z39.50 is the fiftieth standard developed by NISO. The current version of Z39.50 was adopted in 1995 version 3.

Z39.50 had its roots in the OSI efforts of the 1980s. Within the OSI model it is an application layer protocol. So that it requires a full-duplex byte stream transport such as TCP. A TCP port number for Z39.50 is registered and there is a request for command (RFC) that specified hoe to use Z39.50 over TCP(Cliffor A.Zynch) Abstract Syntax Notation one (ANS.1) is used to specify the contents of the protocol data units that are the Basic encoding Rules (BER) are used to serialize the ANS 1 structures.

Work in the ISO on information retrieval standards resulted in ISO approval in 1991 of two International Standards that are companions to Z39.50: ISO 10162, Search and Retrieve Application Service Definition and ISO 10163-1, Search and Retrieve Protocol Specification. ANSI/NISO Z39.50-1992 harmonized the U.S. standard and the international standards so that the U.S. standard became a compatible superset of the International Standards. In early 1994, international standards developers made a crucial decision to begin the process of adopting Z39.50 as the International Standard. No longer will there be different national and international standards that must be harmonized. Rather, Z39.50 will provide the basis for the International Standard for information retrieval.

The NISO Standards Committee that originally created and wrote the first version of Z39.50 was disbanded after the standard was approved in 1988. In 1990, a group of Z39.50 implementors formed the Z39.50 Implementers Group (ZIG). This voluntary group of implementers meet in open sessions approximately three times a year to discuss implementation issues, bring requirements for changes to the standard, work out the detailed specifications, and come to agreements that will go into the draft standard. The ZIG uses a public electronic discussion list (instructions to subscribe to the listserv are described below) to discuss implementation and standards development issues between meetings.

When the draft standard is completed, it is NISO's responsibility to assure that ANSI procedures for balloting, consensus, due process, and other requirements are followed, and that the final approved American National Standard is available for use.

Working closely with the ZIG and providing continuity for Z39.50 development is the Z39.50 Maintenance Agency. In 1988 NISO assigned the Network Development and MARC Standards Office at the Library of Congress to be the Maintenance Agency for Z39.50. The Maintenance Agency provides technical coordination of Z39.50 development, registration of implementers, the registration of objects such as attribute sets and record syntaxes used by the standard, and editorial work on the standard. The Maintenance Agency has provided publicly available electronic drafts of Version 3 (address information for access to the drafts is given below); this electronic access to the drafts has broadened participation in revising the standard.

BASIC STRUCTURE OF Z39.50

Now let's move our focus to that of the Z39.50 structure. The aim is to bring out the functionality of the standard for the library professionals - not to provide a reference for the programmer.

It describes the different facilities of Z39.50 but concentrating on the implications of the standard for LIS. It aims to equip the LIS with enough knowledge to understand what is possible so that he/she can manage the set-up of Z39.50 servers or talk to vendors about their product's capabilities. It is presented in sections to for ease and speed of retrieval and reading.

Basic unit structure of Z39.50 and it process related to Library search is as follows:

a) OPAC user selects Target library (Z-server) from an OPAC menu.

b) OPAC user enters search terms

c) OPAC software sends search terms and Target library details to a "Z-client" a piece of software usually running as part of the library system.

d) Z-client translates the search terms into "Z-speak" and contacts the Target library's Z-server software. There is a preliminary negotiation between the Z-client and Z-server to establish the rules for the "Z-Association" between the two systems.

e) Z-server translates the "Z-speak" into a search request for the Target library's database and receives a response about numbers of matches etc.

f) Z-client receives records

g) Records are presented to the OPAC interface for the user.

During the Z-Association, the Z-client (origin) and Z-server (target) exchange a series of messages. Each message is identified as a technical "service". The services are grouped into what the standard calls "facilities". There are eleven facilities that are as follows:

1. ***Initialisation:*** Setting up the Z-Association, negotiating levels of service
2. ***Search:*** Sending a search string at a database and getting back a result set and the first few records.
3. ***Retrieval:*** Retrieval of records from the result set as specified by the Z-client.
4. ***Result-set-delete:*** Deleting a set of search results held on the Z-server
5. ***Access Control:*** Allowing the Z-server to ask for passwords etc.
6. ***Accounting/Resource Control:*** Allowing accounting, credit control etc.
7. ***Sort:*** Sorting a result set in a defined order on the Z-server
8. ***Browse:*** Scanning an index on the Z-server

9. ***Extended Services: Allowing*** Z-client to start a "task packages" e.g. ILL on Z-server

10. ***Explain:*** Allowing the Z-client to query a database of implementation details on the Z-server and

11. ***Termination:***

A typical Z-39.50 session goes through facilities 1-3 and then terminates when the client has the records it wants. The other facilities add account control and enhanced features like Sort, Browse, ILL etc. These first three facilities are called Core services.

"Core" services that are as follows:

Initialisation :	the preliminary negotiation of what services can be provided.
Search:	the creation and transmission of a query and the receiving of the first results.
Retrieve:	the selection and retrieval of records from a result set.

INITIALISATION

The initialisation is, as the name suggests, a precursor to the real work. The Z-client (origin) contacts the Z-server (target) and suggests some basic parameters for the session. The Z-server can modify these and if both ends agree, the Z-Association begins. Note that the Z-server is in control. Whatever may or may not occur between the two computers is up to the Z-server - thus the library can decide with whom it "deals" and what services it shall provide right at the beginning. Other facilities like access control and resource control give more detailed control.

The parameters that are discussed during the initialisation are:

- **Version of Z39.50 to be used -** thus a version 3 client and a version 2 Z-server (target) can agree to talk - using the highest common version.

- **ID / authentication -** usually a password style of control.

- **Options -** a string of yes / no responses to whether certain capabilities will be used. Search and Retrieve - the basic operations are normally allowed but, because the Z-server (target) is in control, it may be that only record counts for successful searches should be returned - not the records themselves - unlikely perhaps, but possible. The other options that may be allowed / disallowed at this stage of the game are: delete, resource-report, scan, sort, extended services, resource-control and access control.

SEARCH

This is where the basic power of the standard can be assessed. Z39.50 allows complex searches to be constructed with great flexibility. The search parameters allowed for bibliographic records are defined in the Bib-1 attribute set. If you look at this set then you can see that it will work with demanding information retrieval applications as well as standard OPAC type bibliographical queries. The implication is that you could use the same search tool for simple bibliographic queries and then switch to "Power mode" and conduct complex searches against text databases. A well designed Z39.50 tool should provide not only the ability to search across many library catalogues using the same interface but also the ability to search a databases at an appropriate level of complexity if it can support it.

It could be argued that a search tool cannot be all things to all men but Z39.50 has the power to support the various levels and even some capabilities to "dumb down" a search if the target cannot handle the search statement.

Here is how a typical Z39.50 search statement could be packaged up - showing the key parameters of the message passed to the Z-server.

The Query is a string of data and associated parameters that define the records being sought, the database to search and so on.

The query that is sent to the Z-server (target) is obviously a key part of the whole process. Internally, it is represented in "Reverse Polish Notation" or RPN. The query can use the Boolean operators AND, OR, AND-NOT and also the Proximity operator PROX depending on the version and query type. The Boolean structure can be as complex as required with multiple nesting of search phrases e.g. shown in algebraic format (a AND (b OR d)) OR c. Also allowed in the query structure are operands for Restriction and Proximity.

When Restriction is allowed, a result-set can be reduced by applying a restriction such as "Author" to restrict the result set to just those records where the query term occurs in the specified field.

Proximity allows the specifying of terms to be near to each other and thus allows for "Full text retrieval". The "nearness" of terms can be expressed in terms of their distance - expressed in units of text e.g. words, sentence, paragraphs. E.g. 0 + sentence means in the same sentence. 2 + word means not more than 2 words apart. Relation can be expressed e.g. > 2 words would mean must be more than 2 words apart.

THE RESPONSE

When the Z-server processes the query, it creates a "result set" - basically a set of pointers to the records in the database - a set of control numbers.

The initial response from the Z-Server is part of the search service and is basically information about the results of the search plus the first few real records as defined by the search parameters. The possible parts of the first response to the query are as follows:

RETRIEVAL

The retrieval facility comprises two services - Present and Segment. Present is a request to the Z-server to send certain records and Segment is the process of breaking a large number of records into smaller numbers for ease of transmission. Segment can also be applied to breaking very large records into parts. The Present service is more interesting to the librarian since it defines how records can be requested. Segment is primarily of interest for optimising network transmission.

PRESENT

After the Search service has retrieved the initial response of records, the user may wish to see more records and a Present message is sent to the Z-server (target). The parameters of this message deal with how many records, how they should be shown etc.
Z-server (target) Response

When the target gets the "Present" request it responds with the records and some associated information.

SEGMENT

As mentioned, the Segment service is not of much interest to librarians and so I am not going to cover this aspect of the standard in any detail. It concerns how records are packaged to transmission across the network. However if network speed is a problem, or you are asking routinely for hundreds or thousands of records to be downloaded, or your records are very large, then segmentation issues are likely to be important.

RESULT-SET DELETE

This service enables a result set to be deleted by the Z-client (origin). The Z-server (target) responds accordingly.

ACCESS CONTROL, ACCOUNT, BROWSE AND SORT

Access control has far reaching implications for the services that libraries might wish to operate via Z39.50. There have been debates about what should be allowed to happen over a Z39.50 session - should libraries download MARC records or not - who has copyright? The Access control built into Z39.50 allows the Z-server (target) to disallow virtually any operation that might be asked of it by the Z-client (origin). If access control fails, then the Z-Association can either continue with those operations that it is allowed to use or terminate. An access control "challenge" can be made in respect of individual records if desired. Thus a library might want certain records only to be provided on a paid for basis or, for security reasons, not to be viewed without authorization.

The format of the messages is defined in the standard or can be agreed separately by libraries and information resources.

ACCOUNTING/RESOURCE CONTROL

The Z-server (target) as part of a specific operation or the whole Z-association initiates this service. It can optionally include a resource report, which could say how current or predicted expenditure was looking, and whether it would exceed agreed limits. Consent to continue an operation can be sought from the Z-client - "can I spend some more of your money?"

Resource Control

The Z-server (target) as part of a specific operation or the whole Z-Association initiates this service. It can optionally include a resource report, which could say how current or predicted expenditure was looking, and whether it would exceed agreed limits. Consent to continue an operation can be sought from the Z-client - "can I spend some more of your money?"

Trigger Resource Control

This service is triggered by the Z-client (origin) and basically asks the Z-server (target) "how much is there left in the account?" The response is a resource control message with a resource report.

Resource Report

The resource report, which is passed to the Z-client (origin), can contain a lot of detailed information including figures on:

- Numbers of records in current set so far
- Estimated complete set size
- Processing time so far and estimated
- Cost so far for this operation
- Estimated cost if completed
- Cost for the Z-Association

SORT

The sort facility is a single service of Sort carried out on the Z-server (target). Some Z-clients have sorting built in so that they can sort (and filter) the results once they have the records retrieved. This way of working is easier, faster and more flexible. Currently not many Z-servers have the sort facility operational so it is the sensible way to approach this function. It is also the most efficient way to handle the vagaries of international sorting sequences since different languages sort in different orders.

The standard defines parameters for the naming of the sorted result set, the sort sequence and statuses to say whether they have been successfully completed or not.

Browse (Scan)

Browse is another facility that is not currently (early 1998) supported on many Z-servers. The single service is called Scan and allows an ordered list e.g. alphabetical list of subject headings or authors to be scanned or browsed. This is in contrast to a typical Z39.50 search and retrieve where a Result-Set of records is retrieved in response to a query. Scan is a request to bring back a list of headings in order from a specific start point and allow browsing. Browsing is an important part of research and the adoption and improvement of this feature will provide much-improved features when searching remote catalogues.

The Scan request is issued from the Z-client (origin) much as a Query is issued. The parameter list shows the controls that can be set for a scan operation. A good Z-Client can also use the returned information as a specific Query without re-entry so that on selection of a personal author for example, the user can then request books by that author.

EXTENDED SERVICES

The extended service facility is a major enhancement to the standard's original "Search and Retrieve" objectives. Extended services allow the Z-client (origin) to start-up specific "task packages" on the Z-server (target) and to control how they operate.

Much of the interest in the extended services lies in the set of currently defined task packages as shown in the package type parameter below. Z39.50 defines how to start and control these packages via parameters in the various messages in the table - and it also defines task packages. The definition of the task packages is the key to appreciating the implications of the extended services of Z39.50.

When an extended service task package is started up, details are kept in an extended service database. These may be searched (via Z39.50 of course) in order to see what is running on the Z-server (target).

Extended Services: definitions of task packages

This section covers the extended services so far defined by the standard and gives a description of the main features and implications for library and information service provision.

Persistent Result Set

This Service allows the Z-client (origin) to save a result set during a normal search session and then specifies that result set during another session to retrieve the records. The persistent result set can be added to or deleted as required. This service could be used if building up a large result set that is a result of several different searches on the Z-server (target) before they are downloaded. Another use might be the creation of a set of records that could then be accessed by several users or actively sent to several users as part of a "Push" type service. New book records from suppliers as a potential requirements or "desiderata" list could be handled this way.

Persistent Query

A persistent query can be stored on the Z-server (target) for use in subsequent sessions - as a search profile for a SDI service for example - especially when combined with a Periodic Query Schedule Service. Note that the Bib-1 attribute set includes the data added to the database as a searchable parameter so that a query asking for books with Subject ="Library Automation" added since "1 Jan 1998" is possible.

Persistent Query Schedule

This service has been specified to allow a query (either a new one or a previously defined Persistent Query) to be run at intervals automatically - this is the basic requirement for an SDI service. Furthermore, the results can be "exported" to a specific destination via a specified mechanism such as Fax, e-mail, X400 address etc. The important parameters for this service are given below:

Item Order

This service allows a user to place an order for an item. The item can be either an item from a result set or a request - possibly coming from an ISO 10161 Inter Library Loan system. The name of the requester with details of address etc. can be included in the order. Billing information e.g. credit card number and purchase order number can be included.

Database update

This service allows the Z-client (origin) to update records on the Z-server (target). Records can be modified, added or deleted. Note that the standard does not address the problem of two users trying to modify the same record - this has to be controlled outside the standard. It does allow for a client-server model for the basic cataloguing function and has been used by both Geac in their successful GeoCAT product. It also points the way to a possible new class of Z39.50 product, which would work against any Z39.50 database. Thus an LMS (Library Management System) without a MARC cataloguing data entry screen could bundle in a third party MARC editor to work against its catalogue database.

Export Specification

This service allows the Z-client (origin) to define the composition of records to be exported to a destination. The specification includes the fields, syntax e.g. USMARC and the address e.g. e-mail address, printer address, fax number etc.

Export Invocation
This service allows the target to "invoke" an export specification i.e. use the export specification to send a file of records in the required format to the given address. The parameters of this service include:

Explain

Explain is a version 3 facility designed to maintain a searchable database of all the features of a Z39.50 implementation on a Z-server (target). The Z-client (origin) can interrogate this database and find out exactly what services and their basic characteristics are available at the target. For instance the query types supported, Attribute sets, record syntaxes, Term lists (for browsing) extended services etc. There is also provision for a text description of the databases available, when available, number of records, copyright information etc. The Explain facility should allow the Origin to understand exactly what the Z-server (target) can offer. This information can then be presented to the user and/or used to adjust the configuration of the Z-client (origin). The intelligent Z39.50 client of the future will be able to query the database to be used and suggest the most appropriate way of searching it or automatically adjust the way that it presents a query.

Termination

The termination facility allows the Z-client (origin) or Z-server (target) to close down a Z-Association a reason for the close is given e.g. system problem, cost limits, security violation etc. The Z-client (origin) can also ask for a resource report as part of the close request i.e. how much money have I got left in the account?

FUTURE

There have been many criticisms of Z39.50 in technical terms - the very extensibility of Z39.50 has meant that it could become hampered by an ever widening group of potential interested user communities. It is considered old-fashioned and complex because it is implemented at a fairly low level - requiring specific binary bits to be set by the program etc. More modern protocol methodologies are available and would be used if it were re-written.

The most important thing about Z39.50 right now is that it exists as a standard and is affecting the way that libraries trade with each other and the services they offer their clients. Should technical considerations force a re-think on how it should operate, from the librarians point of view the most important thing is that it should retain at least the functionality currently implemented. What is required is an open standard allowing seamless information retrieval and associated services from different databases.

IMPLICATIONS

Just as the Web created a new force within the IT industry - even a new industry, so the Z39.50 standard and associated extended services could revolutionise the library industry by creating a class of software products that can talk to any library system. In turn, the use of such products by individual users will mean libraries doing business directly with researchers. Libraries will be competing with each other for this business as it grows - and much of it will probably still be book based...We are already seeing stand alone products with Z39.50 and ILL facilities coming on to the market.

The record syntaxes are the definitions that Z-clients and Z-servers work with when exchanging records. Both ends need to understand the records they are dealing with. This section describes the main types currently defined by the standard.

Z39.50 RECORD SYNTAXES

MARC formats

Z39.50 supports most of the main MARC formats - assigning them an "object identifier number" so that the Z-client and Z-server can easily understand which favour of MARC they are dealing with. This is the list of supported MARC formats:
UNIMARC, CCF, USMARC, etc. These formats are all exchanged in the ISO2709 format (MARC exchange format).

SUTRS

SUTRS is the Simple Unstructured Text Record Syntax - designed to allow ASCII text records to be transferred during a Z39.50 session. The syntax does not have any field structures - the data is supplied as text data with "line feeds" every 72 characters.

OPAC

The OPAC syntax is designed to simulate a catalogue card or similar bibliographic reference format. The OPAC syntax includes additional fields for the display of holdings and circulation data e.g. date due if on loan.

SUMMARY

The summary record syntax is often used to present brief records to the user as the first response during the Search operation. Fields include, title, author, call number place, date.

GRS-1

Generic Record Syntax - a complex tree like structure designed to represent a database record. Although complex, it can be used to represent any type of database records.

Explain

The syntax used for querying the Explain database where information about available services is stored. E.g. databases available, record syntaxes supported (versions of MARC), lists of terms available for the Browse function, Payment terms, languages etc.

Extended Services

When a user starts an extended service e.g. an ILL package, the details of the "task package" are held in the Extended Services database. These details may be queried via Z39.50 and information is presented sign the extended services record syntax. Information held is fairly brief - package name, user ID, retention time permissions etc.

Z39.50: BIB-1 ATTRIBUTES

The Bib-1 Attribute Set is used in the formulation of the query by the client to specify the search. This page gives the full set of attributes comprising fields, operator's etc. The potential power of the standard can be gauged from the comprehensive nature of these values.

Use attributes: (fields for searching) relation attributes, position attributes, structure attribute, truncation attributes, completeness attributes.

This is the complete list of the fields that maybe specified when making a search request. These will be mapped by the Z-server (target) to the database. It is this processes that is key to deciding what is retrieved in answer to a specific request. The level of detail shows how precise it is possible to be. As well as individual fields like author-name personal, there are combinations of fields defined so that more generic search requirement or the limitations of Z-servers can be managed. For example name, author, any, author-title-subject. There are individual specific concepts like Title uniform as well as the more general title. When setting up a Z-client (origin) and Z-server (target), these fields will be mapped to the index choices for the user and the database indexes at the Z-server. Poor set-up of these can make significant differences to the success of searches. Note that the standalone PC Z-clients like BookWhere 2000 allow the user full access to these parameters for constructing searches.

- Personal name
- Corporate name
- Conference name
- Title series
- Title uniform
- ISBN
- ISSN
- LC card number
- BNB card no.
- BGF number
- Local number
- Dewey classification
- UDC classification
- Bliss classification
- LC call number
- NLM call number
- NAL call number
- MOS call number
- Local classification
- Subject heading
- Subject Rameau
- BDI index subject
- INSPEC subject
- MESH subject
- PA subject
- LC subject heading
- RVM subject heading
- Local subject index
- Date
- Date of publication
- Date of acquisition
- Title key
-

- Title collective
- Title parallel
- Title cover
- Title added title page
- Title caption
- Title running
- Title spine
- Title other variant
- Title former
- Title abbreviated
- Title expanded
- Subject precis
- Subject rswk
- Subject subdivision
- No. nat'l biblio.
- No. legal deposit
- No. govt pub.
- No. music publisher
- Number db
- Number local call
- Code--language
- Code--geographic area
- Code--institution
- Name and title
- Name geographic
- Place publication
- CODEN
- Microform generation
- Abstract
- Note
- Author-title
- Record type
- Name

- Author
- Author-name personal
- Author-name corporate
- Author-name conference
- Identifier--standard
- Subject--LC children's
- SubjectName--personal
- Body of text
- Date/time added to db
- Date/time last modified
- Authority/format id
- Concept-text
- Concept-reference
- Any
- Server-choice
- Publisher
- Record-source
- Editor
- Bib-level
- Geographic-class
- Indexed-by
- Map-scale
- Music-key
- Related-periodical
- Report-number
- Stock-number
- Thematic-number
- Material-type
- Doc-id
- Host-item
- Content-type
- Anywhere
- Author-Title-Subject

Relation attributes

- less than
- lessthan or equal
- equal
- greater or equal
- GreaterThan not equal
- phonetic
- stem
- relevance
- AlwaysMatches

Truncation Attributes

- right Truncation
- left truncation
- left and right
- do not truncate
- Process # in search term
- regExpr-1
- regExpr-2
-
-

Structure Attribute

-
- phrase
- word
- key
- year
- date (normalized)
- word list
- date (un-normalized)
- name (normalized)
- name (un-normalized)
- structure
- urx
- free-form-text
- document-text
- local number
- string
- numeric string

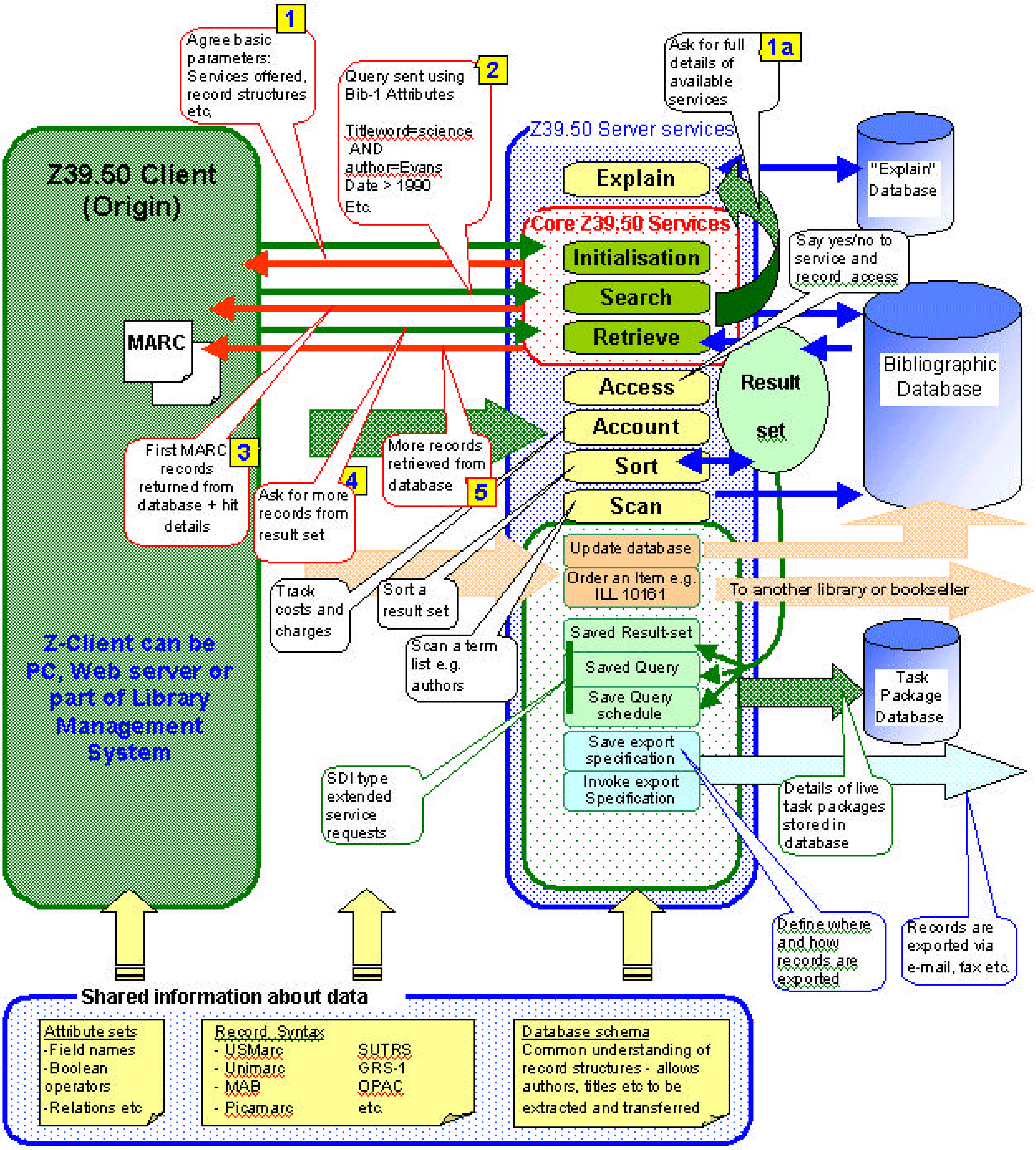
Completeness Attributes

- incomplete subfield
- complete subfield
- complete field

Position Attributes

- first in field
- first in subfield
- any position in field

Z39.50 Diagram


1
Agree basic parameters: Services offered, record structures etc.
2
Query sent using Bib-1 Attributes
Titleword=science AND author=Evans Date > 1990 Etc.
1a
Ask for full details of available services
Z39.50 Client (Origin)
MARC
Z39.50 Server services
Explain
"Explain" Database
Core Z39.50 Services
Initialisation
Search
Retrieve
Say yes/no to service and record access
Access
Account
Sort
Scan
Result set
Bibliographic Database
3
First MARC records returned from database + hit details
4
Ask for more records from result set
More records retrieved from database
5
Track costs and charges
Sort a result set
Scan a term list e.g. authors
Update database
Order an Item e.g. ILL 10161
To another library or bookseller
Saved Result-set
Saved Query
Save Query schedule
Task Package Database
Save export specification
Invoke export Specification
Details of live task packages stored in database
Z-Client can be PC, Web server or part of Library Management System
SDI type extended service requests
Define where and how records are exported
Records are exported via e-mail, fax etc.
Shared information about data
Attribute sets
-Field names
-Boolean operators
-Relations etc
Record Syntax
- USMarc
- Unimarc
- MAB
- Picamarc
SUTRS
GRS-1
OPAC
etc.
Database schema
Common understanding of record structures - allows authors, titles etc to be extracted and transferred

CONCLUSION

Z39.50 does offer one TRUE interface to a variety of databases. Changing the interface is not frequent, but the beauty of this one is that it still remains the same for all servers! Some of these products are starting to offer UNICODE support, which will become more and more important as we move into multi- lingual record displays. They are functionally rich because this kind of product can support simultaneous searching of multiple databases. This is a very valuable feature in that it greatly compresses the amount of time required to sequentially query multiple databases. Support of virtual union catalogs is possible, removing the need to duplicate data to create union catalogs. Search filtering on the client PC is also possible. At the same time, it is higher cost than browsers are. Configurations are bit complex for our professional levels. Mostly it is server dependent Because of that certain types of search may not be success if the field is not configured etc. the list goes on and on.... . These types of products typically require fairly new and powerful systems upon which to run. Most require at least a 486 with 8 MB of memory and a fair amount of disk. Anyhow these is the better solution for the current information environment in three ways: Common User Interface, Common and database specific Interface and Multi database searching facility.

FURTHER INFORMATION:

ANSI/NISO Z39.50-1992 (version 2), Information Retrieval Service and Protocol: American National Standard, Information Retrieval Application Service Definition and Protocol Specification for Open Systems Interconnection, 1992.

ANSI/NISO Z39.50-1995 (version 3), ANSI Z39.50: Information Retrieval Service and Protocol, 1995.

Clifford A.Lynch. Using the Z39.50 information retrieval protocol in the Internet environment: result for Comments: 1726 (Dec.1994)

LC Z39.50 server configuration Guidelines. LCWWW/Z39.50GATEWAY

RFC 1729, Using the Z39.50 Information Retrieval Protocol in the Internet Environment, December 1994. Z39.50 Resources - a Pointer Page, by Robert Waldstein of AT&T (includes pointers to various freely available Z39.50 software packages).

Register of Z39.50 Implementers, available from the Z39.50 Maintenance Agency.

Sara L.Randall (Ed.), Up and Running: Implementing Z39.50. Bethesda, MD: NISO Press,64p.