Digital Libraries: Interoperability

RAFFAELLA BERNARDI

UNIVERSITÀ DEGLI STUDI DI TRENTO

P.ZZA VENEZIA, ROOM: 2.05, E-MAIL: BERNARDI@DISI.UNITN.IT

Contents

1. Interoperability

Interoperability is the ability of systems, services and organisations to *work together* seamlessly toward common or diverse goals.

In the technical arena it is supported by open *standards* for communication between systems and for description of resources and collections, among others.

Interoperability is of paramount relevance in the context of resource discovery and access.

2. Background: DB, Client and Servers, Protocols

Database Think of a DB as a table.

Professions			
Id	First Name	Surname	Role
1	Raffaella	Bernardi	Teacher
2	Enrico	Bignotti	Student
...	...	...	...

Courses		
Id	Surname	Coursename
1	Bernardi	DL
1	Bernardi	LoLa
2	Bignotti	DL
...	...	...

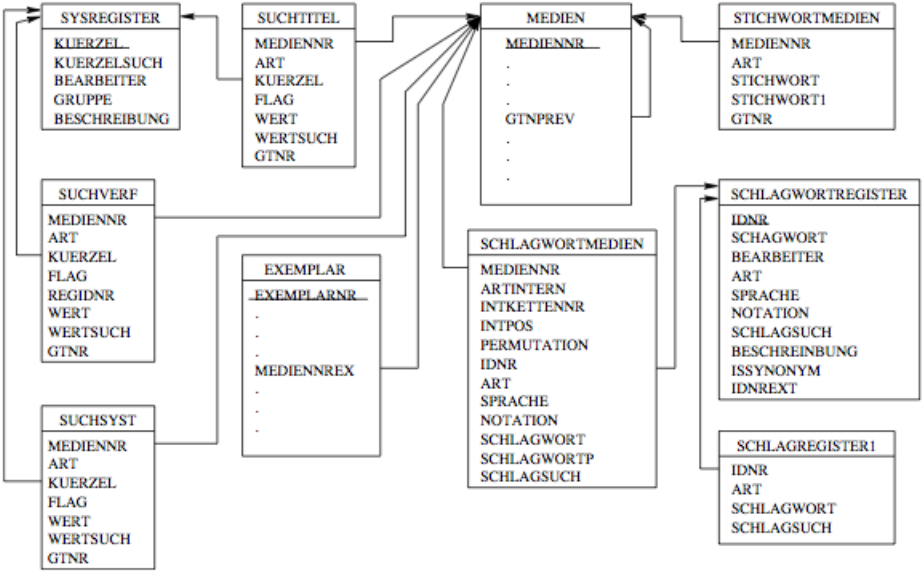
Query: course taught by Raffaella. Or students attending courses taught by Raffaella. Need of merging the info in the two tables.

Query Languages are computer languages used to make queries into DB.

Client-Server model is a distributed application structure that partitions tasks between the *providers of a resource or service, called servers*, and *service requesters, called clients*.

Protocol A communications protocol is a formal description of digital message formats and the rules for exchanging those messages in or between computing systems.

3. OPAC Databse: an example of Bolzano schema



4. Query Languages

The formal language for representing queries to bibliographic catalogues is Common Query Language (CQL). Another standard query language is SQL:

```
Question: Courses taught by Raffaella:
SELECT C.Coursemame
FROM Courses C, Professions P
WHERE C.Id = P.Id AND
     P.FirstName = 'Raffaella' AND
     P.Role = 'Teacher'
```

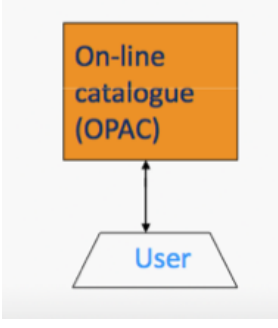
```
Question: Students attending courses taught by Raffaella:
SELECT PS.FirstName, PS.Surname
FROM Courses CS, Courses CT, Professions PS, Professions PT
WHERE CS.Id = PS.Id AND
     PS.Role = 'Student' AND
     CS.Coursename = CT.Coursename AND
     CT.Id = PT.Id AND
     PT.FirstName = 'Raffaella' AND
     PT.Role = 'Teacher'
```

5. Recall: Access to OPAC

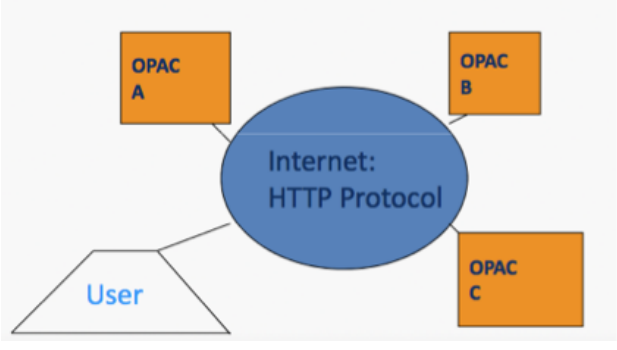
Local access A user had to go to the library, and use a PC where OPAC was installed and search there.

Remote access We can search an OPAC (even more than one) remotely.

Local



Remote



5.1. Client and Server need to speak

The user uses the client (e.g. a Browser – Netscape, Internet Explorer, etc.). The Client needs to send a message to the Server, it has to send him a request.

The Server needs to answer and send him the object required. The communication happens via a *Protocol*.

5.2. Example: HTTP Protocol

Eg. Request and answer for a file about wwwOpac via HTTP Protocol:

```
dream:Lectures bernardi$ telnet pro.unibz.it 80
Trying 46.18.24.42...
Connected to pro.unibz.it.
Escape character is '^]'.
GET /opacuni/index.asp
HTTP/1.1 200 OK
Date: Mon, 02 May 2011 16:24:30 GMT
Server: Microsoft-IIS/6.0
Content-Type: text/html; Charset=iso-8859-1
X-Powered-By: ASP.NET
Pragma: no-cache
cache-control: no-store
Content-Length: 1422
Content-Type: text/html
Expires: Mon, 02 May 2011 16:24:30 GMT
Set-Cookie: ASPSESSIONIDCAQCRDQQ=NALHDLKCJKFBHJDLMKIIFNPH; path=/
Cache-control: private
```

```
<HTML>
<TITLE>wwwOpac - vers.6.1</TITLE>

MARC: per scambiare dati; occupa poco spazio (pochi byte), si
trasferisce piu' velocemente. oggi non serve un tempo si.
....
```

6. Z39.50 Protocol

Z39.50 is a clientserver protocol for searching and retrieving information from remote computer databases. The syntax of the Z39.50 protocol allows for very complex queries.

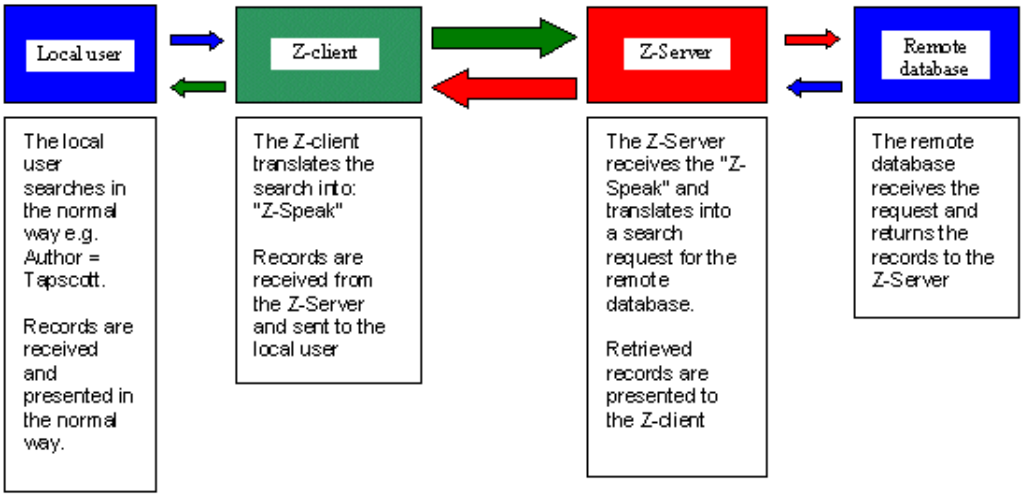
Z-Client The software on the local system translates search query into format of Z39.50 standard; Connects to and sends the query to the system housing the database; presents records/results of query to searcher. The searcher of the client never interacts directly with the server.

Z-Server The Server house the database(s); translates the Z39.50 query to the search logic of database system; obtains info from the database, returns it to the origin system; returns records or reports a result set.

6.1. Client Facilities

Z39.50 Facility	Client-side description
Initialization	Establish connection with server and set/request resource limits.
Search	Initiate search using a registered query syntax, generating a result set server-side.
Retrieval	Retrieve a set of records from a specified result set: a large record may be segmented and transmitted piecemeal.
Result-set-delete	Request deletion of server-side result set or sets.
Access Control	Server initiated authentication check.
Accounting & Resource Control	Request status reports of committed server resources and dictate if server is allowed to contact client when agreed limits are reached.
Sort	Specify how a result set should be sorted.
Browse	Access ordered lists such as title and subject metadata.
Explain	Interrogate server to discover supported services, registries, and so on.
Extended Services	Access services that continue beyond the life of this client-server exchange, such as persistent queries and database update.
Termination	Abruptly end client-server session: initiated by either client or server.

6.2. From Users to DB via the client and server



7. Open Archive Initiative (OAI)

- The roots of OAI lie in the development of eprint archives ((i.e. Institutional Repositories) such as arXiv, CogPrints, NACA (NASA), RePEc, NDLTD, NCSTRL, etc.
- The OAI use of the term "archive" implies very little of what we normally associate with archives. No preservation aspect is implied whatsoever (not what the protocol is about at all.) Archive stands simply for "collection of digital objects".
- Each repository offered a web interface for deposit of articles and for end-user searches
- It was difficult for end-users to work across archives without having to learn multiple different interfaces
- Initial experiments for single search interface to all archives
- Universal Pre-print Service (UPS) renamed OAI at the Santa Fe Convention (1999)

7.1. Z39.50 and OAI PMH

- For “resource discovery” in the “Web age”, the proposed alternative to Z39.50 is the OAI Protocol for Metadata Harvesting (OAI PMH)
- Historical separation from Z39.50: OAI appears about 15 years after Z39.50
- Cultural separation Z39.50: Z39.50 originated in the traditional library community, while OAI originated in the “Web Community”
- Conceptual separation Z39.50: Z39.50 based on solid (but heavy and bulky) foundations, while OAI based on simple and pragmatic ideas

7.2. Searching vs. Harvesting

- Two possible approaches for single search interface to all archives
  1. cross searching multiple archives based on protocol like Z39.50 (possibly lighter)
  2. harvesting metadata into one or more central services
- Problems with cross searching
  1. Not scalable (overall performance determined by slowest server)
  2. Problems of deciding which servers to target (collection descritpions not consistent)
  3. Differences in interfaces and query languages
  4. Problems in the ranked merging of results (different types and size of targets can skew results)
  5. Browse interface very difficult to build

The decision was to go with harvesting.

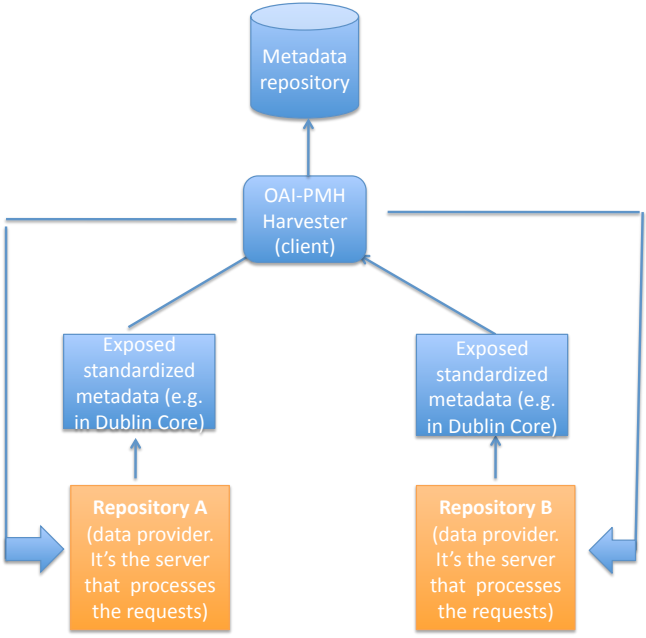
7.3. OAI-PMH

OAI Protocol for Metadata Harvesting:

1. Data providers make metadata available for harvesting
2. Service Providers harvest metadata
3. Metadata can be centrally collected or “aggregated”
4. Data Providers
   - Are creators and keepers of the metadata for objects (repositories) and (possibly but not necessarily) archives of resources
   - Handle deposit and publishing
5. Service Providers: Are harvesters of metadata for the purpose of providing a service such as a search interface, peer-review system, etc.

Example of Digital Library software systems: http://www.dspace.org/ and http://fedoraproject.org/.

7.4. OAI-PMH workflow



7.5. Web Service

Semantic Web view: a networked world with ubiquitous access to a wide variety of both resources and services, a world of Web Services. Different view on how to share holdings.

Web Services should be: modular, self-describing, standards-base, platform and programming-language independent, XML-based.

Examples: annotation services, automatic document alignment, gazetteer lookup, named entity identification, etc...

7.6. From Z39.50 to SRW/U

- Need for a generic Information Retrieval capability more suited to the Web Architecture
- Motivation to create an easy to implement protocol with (more or less) the power of Z39.50
- Use existing off the shelf solutions where possible
- Re-evaluate Z39.50, “a good idea at the time”
- Avoid library-centric perspective

Solution:

- SRU Search/Retrieve via URL
- SRW Search/Retrieve via Web Service (SRW is now called SRU over SOAP)

8. Database vs. IR

	Database	IR
System provides	data item	pointer to data
User's query	specific	general
Retrieval method	deterministic	probabilistic
Success criteria	(correctness) efficiency, user-friendliness ...	utility

Next time Tomorrow, we will study IR. Wednesday 11th of April: 15:00-16:30 (instead of 16:00-18:00.)