CEOS/NASA

Earth Observation (EO)/GEO Web Workshop '97

ZDSR

Z39.50 Profile for Distributed Search and Ranked Retrieval

February 4-6, 1997
Washington, DC, USA

Ray Denenberg, Library of Congress
ray@rden.loc.gov

Z39.50 Profiles

See http://lcweb.loc.gov/z3950/agency

- **ATS**
- **GILS**
- **WAIS**
- **GEO**

- **ZDSR**

- **CIP**

- **Collections**
  - **CIMI**
  - **Digital Library Objects**

- **ZDSR**

- **Collections**

  - **CIMI**
  - **Digital Library Objects**

--> - **CIP**
--> - **GILS**
--> - **Geo**

  - **(music)**
  - **(multimedia)**
  - **(archives)**

Some Historical Background Events Relevant to ZDSR

- **Stanford Agreement for Internet Retrieval and Search (STAIRS/STARTS)**

- **W3C Distributed Indexing/ Distributed Searching Workshop, May 1996.**

- **Z39.50 "Lite" (Zlite)**

Semantic Interoperability Spectrum

Navigation
(Collections Model)

Distributed Searching

Global Index

Problems with the "global Index" model

- **Flat**
- **Inability to distribute**
- **Only web pages**
- **Complexities**
  - **What documents to index?**
  - **How much of a document?**
  - **How often?**
  - **How deep?**
- **Problems with crawlers**

Distributed Search Model

client sends query to *meta-search engine*, who:

- ***Selects*** **one or more real sources.**
- ***Relays*** **query to the real sources.**
- **Retrieves and *integrates* results.**
- **Presents single, logical result set.**

Distributed Searching: Barriers

- ***Economic***

  The advertisement model

- ***Technical***
  - **Referral**
  - **Merging and ranking**

STAIRS

Stanford Agreement for Internet Retrieval and Search

PLS
Infoseek
Microsoft
Fulcrum
Netscape
Hewlett-Packard
Exite
Verity
Xerox

...

STAIRS Requirements

- **Selecting sources**
- **Merging/ranking**
- **Common query language**
- **Search capabilities**
  - **Fielded searching**
  - **Stem/phonetic/RF/thesaural expansion/truncation/case**
  - **stop word control**
- **Client specifies:**
  - **ranking criteria**
  - **Minimum score**
  - **Max documents**
  - **result record composition**
  - **result order**

Requirements (continued)

Servers Provide:

- Ranking algorithm id
- Tokenizer id
- Sample results

  Results for well-known sample document collection and set of well-known queries; allows meta-searcher to calibrate document scores from different sources.
- Metadata
  - Server metadata
  - Database metadata
  - Document metadata
  - Per-term metadata
    - document frequency
    - per-document
      - Term-frequency
      - Term-weight

Zlite	**Global Index**
	V
STAIRS/ ZSTAIRS	**Distributed Searching**
	V
Z39.50 DC Profile	**Navigation (Collections)**

STAIRS

Stanford Agreement for Internet Retrieval and Search

STARTS

Stanford Protocol for Internet Search and Retrieval

ZSTARTS

Z39.50 Profile for ZSTARTS

ZDSR

Z39.50 Profile for Simple Distributed Search and Ranked Retrieval

Zlite

--------> (superceded by)

ZSTAIRS

--------> (changed name to)

ZSTARTS

--------> (changed name to)

ZDSR

ZDSR Features:

- **Documents.**
- **Document *descriptors* (metadata).**
- **Database/ and server level Metadata.**
- **Search by author, title, language, within body of text, url, last-modified.**
- **Miscellaneous auxiliary search features:**
  Relevance feedback, stem, phonetic, Stop-word control.

ZDSR Features (cont):

- **Query re-formulation.**

- **Query and term level metadata.**

- **Ranking features. Including client ranking control.**

- **Sorting.**

- **Encapsulation.**

ZDSR Document Metadata

- title
- abstract
- publication date
- creation date
- date last modified
- size
- score
- frequency and weight per term
- url

Normalizing Rankings

* * Metasearcher requests each server employ specific, public algorithm.

* * Metasearcher retrieves normalization info from each search engine.

* * Client specifies ranking criteria.

* * Server supplies ranking algorithm id.

- metasearcher retrieves documents into pseudo db and executes original query.

Complexity and Religion

- **Size and complexity of the Z39.50 protocol/specification**

- **Encoding. ASN.1/BER vs. ASCII**

- **Record Syntax. GRS-1 vs. SOIF**

- **Stateless vs. stateful**

- **Connection vs. Connectionless (i.e. single round trip)**

- **Modularity vs. overloading**

Z39.50 Operations

Client **Server**

Init Operation

Init request . . . . >

< . . . . Init response

Search Operation

Search request . . . . >

< . . . . Search response

Present Operation

Present request . . . . >

< . . . . Present response

Sort Operation

Sort request . . . . >

< . . . . Sort response

The Essense of a Search

Search request . . . . >

***Database*(s) to search**

new *result set* name

Query

< . . . . Search response

***How may records* were identified by the query**

Present

Present request . . . . >

What *result set*

which *records* from that result set

syntax/composition

Example: Retrieve records 1-10 from result set R; retrieve 'brief' records, in USMARC format

< . . . . Present response

Records

Search request . . . . >

Database(s)
result set
Query

+

- **Additional "piggy back" present parameters**
- **Additional "extensibility" information**

< . . . . Search response

Number of records identified

+

Maybe some of the records

"Additional Extensibility Information" in the ZDSR Search Request

- **The "ranking" component of the query**

- **A ranking algorithm id**

- **Sort criteria**

ZDSR Query

ZDSR Query =

restriction component

+

ranking component

restriction component =

Z39.50 type-1 query

Ranking component =

List of terms. Each assigned a relative weight.

"Additional Extensibility Information" in the ZDSR Search Response

- Actual restriction component executed.
- Recommended restriction component.
- Actual ranking component.
- Recommended ranking component.
- Ranking algorithm id.
- Sort response.

Stateless/Connectionless, via "Encapsulation":

Instead of this:

Search request . . . . >
< . . . . Search response

Sort request . . . . >
< . . . . Sort response

Present request . . . . >
< . . . . Present response

This:

Search request . . . . >
encapsulated Sort request
encapsulated Present request

< . . . . Search response
encapsulated Sort response
encapsulated Present response