

LINK AFFILIATES



Search Web Services (SWS)

Judith Pearce

Link Affiliates
jpearce@gmail.com

Linking research & learning technologies through standards

Why I'm here

- To outline the current status of the work being done by the OASIS Search Web Service Technical Committee

Background

- The TC was set up in June 2007 to define a set of search web services that would address the growth in number of search protocols and proprietary and abstract search APIs in recent years
- The plan was to build on existing work, using SRU and CQL as starting points

-
- SRU is a search web service developed to replace ISO 23950 (ANSI/NISO Z39.50)
 - CQL is the abstract query language that it uses
 - Both are highly-functional standards but relatively unknown outside the library community

OASIS SWS TC Members

- .∴ Ray Denenberg—Library of Congress—Co-Chair
- .∴ Matthew Dovey—JISC Executive—Co-Chair
- .∴ Kerry Blinco - Link Affiliates - Voting Member
- .∴ Larry Dixon—Library of Congress—Voting Member
- .∴ Janifer Gatenby—OCLC—Voting Member
- .∴ Ralph LeVan—OCLC—Voting Member
- .∴ Ashley Sanders—Univ. of Manchester—Voting Member
- .∴ Robert Sanderson—Univ. of Liverpool—Voting Member
- .∴ Robin Cover - OASIS - Member
- .∴ Sri Gopalan—Booz Allen Hamilton—Member
- .∴ Carl Read - Open Geospatial Consortium, Inc. (OGC) - Member
- .∴ MacKenzie Smith—M.I.T. —Member



- Most of the people on the technical committee, including the joint chairs had been involved in the development of SRU and CQL.
- Using them as the starting point for the new search web service was a deliberate strategy to expose this work to mainstream communities
- SRU 1.2 also needed updating and it was hoped that the new protocol, enriched by new usage scenarios, would be SRU 2

My credentials

- Helped design and build a range of federated discovery services at the NLA
- Influenced NLA to move to a single business, service-oriented approach using SRU as the preferred search protocol
- Commissioned Indexdata to build a Z39.50-SRU Gateway (released as Simple2ZOOM)

Why we need search protocols

- Search as a “Service”
 - SOA, Mashups, workflows...
- Federated Search / Metasearch
- Dynamic vs static data sharing
 - Search is dynamic, data can reside (and be updated) elsewhere
 - Harvest is static, data is gathered for local storage and use and has to be re-gathered when it is updated

Features of Search Protocols

- A way to express queries (query language)
- A way to express results
- A method of transporting queries and results
- Description of the service itself

Lessons learnt

- Developers prefer to implement a simple low-functionality protocol like Opensearch
- There is still be a need for high-functionality interfaces to achieve interoperability
- High-functionality interfaces are complex but there is already a lot of invested IP in SRU and CQL
- A development path is needed which allows this shared experience to be exploited when it is required

Proposed solution

- Abstract Protocol Definition
 - Set of application Protocol Bindings
 - Binding to HTTP Get
 - Binding to SRU 1.2
 - Binding to OpenSearch
- (Drafts released for comment on 30 June 2008)

Abstract Protocol Definition

•• Data model

- A server exposes a datastore for access by a remote client. The datastore is a collection of units of data or *items* with associated formats or *item types*

•• Processing model

- A client sends a request to a server with a search query and may also include desired number of items and desired formats for the result set and matching items
- The server responds with a result set containing a set of items and may also include diagnostic information and a result set identifier

•• Result set model

- The result set is a set of abstract items that may or may not be persistent



Request parameters

- ⋮ responseType
 - e.g. 'text/html', 'application/atom+xml' , application/x+sru
- ⋮ query
- ⋮ startPosition (or group)
- ⋮ maximumItems
- ⋮ itemType
- ⋮ sortOrder

Response parameters

- .: numberofItems
- .: numberOfGroups
- .: resultSetId
- .: **items**
- .: nextPosition / nextGroup
- .: diagnostics
- .: echoedRequest

Bindings

| Request parameter | OpenSearch 1.0 | SRU 1.2 |
|-------------------|---|--|
| responseType | (none) | (fixed) |
| query | searchTerms | query |
| startPosition | startIndex | startRecord |
| maximumItems | count | maximumRecords |
| Group | startPage | (none) |
| itemType | (none) | recordSchema |
| sortOrder | (none) | (specified in query) |
| Extensions | Language InputCoding OutputCoding | Operation, Version recordPacking resultSetTTL, styleSheet extraRequestData |

Bindings

| Response parameter | OpenSearch 1.0 | SRU 1.2 |
|--------------------|-------------------------------------|---|
| numberOfItems | totalResults | numberOfRecords |
| numberOfGroups | (none) | (none) |
| resultSetId | (none) | resultSetId |
| items | per response schema | record |
| nextPosition | derivable | nextRecordPosition |
| nextGroup | (none) | (none) |
| diagnostics | (none) | diagnostics |
| echoedRequest | derivable | echoedRequest |
| Extensions | startIndex itemsPerPage query | version resultSetIdleTime extraResponseData |

Types of bindings

- Static (human-readable)
 - Protocol specifications (e.g. SRU 1.2, SRU 2, Opensearch)
 - Profiles (community agreements about level of conformance, item types, controlled vocabularies and query semantics)
- Dynamic (machine-readable)
 - Implementation-level bindings
- A static binding may need dynamic bindings for a complete description (e.g. Opensearch)
 - Called Intermediate bindings in the Abstract Protocol Definition

SWS Description Language

- a client should be able to use a dynamic binding to formulate a request and interpret the response
- This is dependent on being able to read and interpret the description
- This needs a standard description language

Next steps for the TC

- Standardisation of CQL 1.2
- SRU 2
- CQL 2
- SWS Description Language

More information

- OASIS SWS TC website

<http://www.oasis-open.org/committees/search-ws>

- SRU/CQL Standardization in OASIS

<http://www.loc.gov/standards/sru/oasis.html>