

# Search Web Services

*Ralph LeVan  
Senior Research Scientist*

*NISO Discovery Tools Forum 2008*

# OASIS Search Web Services Technical Committee



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

To define Search and Retrieval Web Services, combining various current and ongoing web service activities.

# OASIS Search Web Services



## TC

Ray Denenberg—Library of Congress—Co-Chair

Matthew Dovey—JISC Executive—Co-Chair

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

Sri Gopalan—Booz Allen Hamilton—Member

MacKenzie Smith—M.I.T. —Member

## Who is OASIS



OASIS is a non-profit, international consortium that creates interoperable industry specifications based on public standards such as XML and SGML.

The ebXML suite of standards is probably their most famous product

<http://www.oasis-open.org>

## Why are we there?



We were hoping to reach a broader audience than we normally see in NISO

We were hoping that there would be synergies with the other XML-based standards groups. After all, most of them have searching requirements.

# Where We've Come From



## Pros, Cons and What We've Learned

- Z39.50
- SRW/U
- OpenSearch

## Pros

- High Functionality
- High Interoperability

## Cons

- Complicated
- Binary encoding over raw tcp/ip

## Lesson Learned

- There's a need for a high functionality interface
- If people are desperate enough, they'll do anything

## Pros

- XML-based web service
- High Interoperability

## Cons

- Still complicated (but much less than Z39.50!)
- Unheard of outside the library community

## Lesson Learned

- There's still a need for a high functionality interface
- If people aren't desperate, they'll live with what they've got

## Pros

- Simple
- Moderate Interoperability

## Cons

- Low Functionality

## Lesson Learned

- There's a need for a simple low functionality interface
- Developers prefer to do as little as possible

# What We're Doing



- CQL 1.2
- SRU 2.0
- Abstract Protocol Definition
  - Binding to HTTP Get
  - Binding to SRU 1.2
  - Binding to OpenSearch
- SWS Description Language

## CQL 1.2



This is the path to actually standardize CQL

Enhances a couple of features (sort and proximity and the CQL Context Set)

## SRU 2.0?



I wish I had something to say here, but it's mostly on the todo list and the SWS Description Language has more traction in the committee.

# Abstract Protocol Description (APD)



This document is an abstract protocol definition for the Search Web Services (SWS) searchRetrieve operation. It presents the model for the SearchRetrieve operation and is also intended to serve as a guideline for the development of *application protocol bindings* (hereafter *bindings*, see [definitional note](#)).

A binding describes the capabilities and general characteristic of a server or search engine, and how it is to be accessed. A binding may describe a class of servers via a human-readable document or a binding may be a machine-readable file describing a single server, provided by that server, according to the description language described at xxx, which is a fundamental component of the SWS standard

# APD Data Model



A server exposes a datastore for access by a remote client for purposes of search and retrieval. The datastore is a collection of units of data. Such a unit is referred to as an *item* in this model. For purposes of this model there is a single datastore at any given server.

Associated with a datastore are one or more formats that may be used for the transfer of items from the server to the client. Such a format is referred to as an *item type* in this model. An item type represents a common understanding shared by the client and server of the information contained in the items of the datastore, to allow the transfer of that information. The item type identifies an abstract representation of the information. It does not represent nor does it constrain the internal representation or storage of that information at the server

# APD Processing Model



A client sends a searchRetrieve request to a server, which responds with a searchRetrieve response. The request includes a search query to be matched against the items at the server's datastore. The server processes the query, creating a result set (see [Result Set Model](#)) of items that match the query.

The request also indicates the desired number of items to be included in the response and includes information about how the individual items in the response, as well as the response at large, are to be formatted.

The response includes items from the result set, diagnostic information, and a result set identifier that the client may use in a subsequent request to retrieve additional items.

## APD Result Set Model



This is a logical model; support of result sets is not assumed nor required by this standard

From the client's point of view, the result set is a set of items each referenced by an ordinal number, beginning with 1. The client may request a given item from a result set according to a specific format. For example the client may request item 1 in Dublin Core, and subsequently request item 1 in MODS. The format in which items are supplied is not a property of the result set, nor is it a property of the requested items as a member of the result set; the result set is simply the ordered list of items.

# APD Request Parameters



Abstract Parameter Name	Description
<a href="#"><u>responseType</u></a>	e.g. 'text/html', 'application/atom+xml' , application/x+sru
<a href="#"><u>query</u></a>	The search query of the request.
<a href="#"><u>startPosition</u></a>	The position within the result set of the first item to be returned.
<a href="#"><u>maximumItems</u></a>	The number of items requested to be returned.
<a href="#"><u>itemType</u></a>	e.g. string, jpeg, dc, iso2709. From list provided by server.
<a href="#"><u>sortOrder</u></a>	The requested order of the result set.

# APD Response Parameters



Abstract Element Name	Description
<a href="#"><u>numberOfItems</u></a>	The number of items matched by the query.
<a href="#"><u>resultSetId</u></a>	The identifier for the result set created by the query.
<a href="#"><u>items</u></a>	a sequence of items.
<a href="#"><u>nextPosition</u></a>	The next position within the result set following the final returned item.
<a href="#"><u>Diagnostics</u></a>	Error message and/or diagnostics.
<a href="#"><u>echoedSearchRetrieveRequest</u></a>	The server may echo the request back to the client.

# HTTP Get Binding



## Syntax

The client sends a request via the HTTP GET method  
Specifically it is an HTTP URL of the form:

<base URL>?<searchpart>

## Encoding

Convert the value to UTF-8. Percent-encode characters as necessary within the value. Construct a URI from the parameter names and encoded values.

## SRU 1.2 Binding



The APD + the HTTP Get Binding + new request parameters (operation, version, recordPacking resultSetTTL stylesheet extraRequestData) – unused base parameters (responseType, sortOrder) + new response elements (version, resultSetIdleTime, extraResponseData) and a shiny XML encoding.

What do we think we've learned?

***Developers are tired of being told how to do their business!***

Unless they have a business reason to worry about interoperability, they won't. Third party interoperability needs to be something they can add on when they do discover they need it. Better yet, let someone else add it on.

## Prescriptive vs Descriptive Standards



A *prescriptive* standard (Z39.50, SRU and the response part of OpenSearch) causes interoperability by telling you how to construct your interface, allowing for simple clients that know how to talk to you. The hard work of interface is done by the server.

A descriptive standard (WSDL and the request part of OpenSearch) causes interoperability by allowing you to describe your interface in such a way that clients can be created dynamically to talk to you. The hard work of interface is done by the client.

## Who Wants This?



- Anyone who wants access to content that doesn't adhere to any search standards: Web 2.0 and NISO Metasearch!
- Anyone with content to provide who doesn't know what clients might want to search that content

## Essentially OpenSearch...



```
<op>  
  
  <request type="template"  
    href="http://copac.ac.uk/wzgw?rsn={resultSetname}&  
    format=XML+-+MODS&id={sessionId}&  
    fs=Download+records"/>  
  
  <response type="XML" schema="AtomResponse"/>  
</op>
```

## ... On Steroids!



```
<request href="http://copac.ac.uk/">  
  <form action="/wzgw" method="get"  
    name="Copac Quick Search">  
    <param name="au" semantics="au"/>  
    <param name="ti" semantics="ti"/>  
    <param name="any" semantics="kw"/>  
    <param name="form" value="qs"/>  
    <param name="fs" value="Search"/>  
  </form>  
</request>
```

## ... On Steriods (cont.)



```
<response>  
  <set name="numberOfItems">  
    <regexp regexp="&lt;span  
      id="&quot;num_hits&quot;&gt;([0-9]+)&lt;"/>  
    </set>  
  </response>
```

## P.S., Bibliographic Context Set Anyone?



SRU depends on context sets. The SRU Editorial Board recognizes the need for context set for bibliographic searching (equivalent to Bib-1 in the Z39.50 universe). But, they don't feel that they are the appropriate body. Anyone in the NISO community interested?

# Questions?



<http://staff.oclc.org/~levan/docs/SearchWebService.ppt>

[levan@oclc.org](mailto:levan@oclc.org)