SPARQLing SERVICES

Leigh Dodds
Engineering Manager, Ingenta
XTech, May 2006
The Three Things

- That Web 2.0 Needs a Query Language
  - And SPARQL fits very nicely
- How to Implement the SPARQL Protocol
  - And some useful extensions
- Implementation Issues
  - And other Things to Consider
- ...plus show a few demos
API Design Pressures

• Pressure for Uniformity
  – Reduce Learning Curves
  – Eliminate Switching Costs

• Managing the Surface Area
  – New features; services; data
  – Impacts on existing users

• Performance Optimisation
  – Usage monitoring; adapting storage models
  – Issues of granularity
How a Query Language Helps

- Skills Transfer
- Substitutability
- “Micro protocols”
  - Clients construct their own APIs
  - “Surface Area” remains constant
- Increased Granularity
  - Potentially fewer requests for same results
  - Optimise for data not resource usage?
SPARQL in a Nutshell

- W3C Data Access Working Group
- RDF Query Language
  - No updates
- Three Specifications
  - Query, Results Format, Protocol
  - All Candidate Recommendations
- Four Query Types
  - ASK, SELECT, CONSTRUCT, DESCRIBE
  - Targeted at different use cases
Example SPARQL Query

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?weblog
FROM <http://www.ldodds.com/ldodds-knows.rdf>
WHERE
{
  ?x foaf:name ?name.
}
The SPARQL Results Format

- XML Format for Query Results
  - For SELECT and ASK queries
- Used to transmit results across the Web
- *Very* Simple and Regular Format
  - 10 elements in a single namespace
- Easy to process
  - E.g. with XSLT and other XML tools
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?weblog
FROM <http://www.ldodds.com/ldodds-knows.rdf>
WHERE
{
  ?x foaf:name ?name.
}
<sparql
 xmlns="http://www.w3.org/2005/sparql-results#">
<sparql
xmlns="http://www.w3.org/2005/sparql-results#">

<head>

   <variable name="name"/>

   <variable name="weblog"/>

</head>

</sparql>
<sparql
xmlns="http://www.w3.org/2005/sparql-results#">
<head>
    <variable name="name"/>
    <variable name="weblog"/>
</head>
<results ordered="false" distinct="false">

</results>
</sparql>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="name"/>
    <variable name="weblog"/>
  </head>
  <results ordered="false" distinct="false">
    <result>
      <binding name="name">
        <literal>Libby Miller</literal>
      </binding>
      <binding name="weblog">
        <uri>http://planb.nicecupoftea.org</uri>
      </binding>
    </result>
    ...more results...
  </results>
</sparql>
The SPARQL Protocol

- How to ship a query and results across the web
  - HTTP and SOAP Bindings
- Very Simple Protocol
  - 1 Required Parameter (the query)
  - GET or POST
- HTTP Status Codes for Errors
  - Malformed Queries (400 Bad Request)
  - Server Errors/Refusals (500 Server Error)
- Allows Content-Negotiation
Example Protocol Request

GET /sparql

?query
  =PREFIX+foaf%3A+%3Chttp%3A%2F%2Fxmlns.com...
  com...

&default-graph-uri=
  http://www.ldodds.com/ldodds-knows.rdf
Data Set Ambiguities

- 3 ways to specify data set in protocol
  - Query, Request, or a default
- Disambiguation Rules
- Order of Precedence
  1. Protocol
  2. Query
  3. Processor
- Processor can always refuse a request
  - Either because no data set, or its unacceptable
Types of SPARQL Service

- Spectrum of Service Types
- Fixed Data Set
  - Fixed set of available data sources
  - GovTrack.us, Opera Community
- Arbitrary Data Set
  - Query over any arbitrary data source(s)
  - SPARQL.org; Rasqal Demonstrator
  - XML ArmyKnife
GovTrack.us is a nexus of information about the United States Congress, following the status of federal legislation and the activities of your senators and representatives.

Since 2004, GovTrack has been an independent website run by a graduate student in his spare time. Data is collected via automated processes daily, and you should always confirm information found here using official sources.

GovTrack.us is in session today.

In the News

**Immigration**
Congress is debating immigration. View the two bills being considered: Senate version (being debated in the Senate) and House version (already passed in the House).

**GET INVOLVED**
Be a part of the political process by knowing your representatives, and then tracking legislation that interests you. Once you monitor your representatives, you will be able to see their voting records.

**BE INFORMED**
Stay up-to-date with Congress by using “monitors”. Your customized tracked events page will show the happenings in congress related to your interests. Sign-up for free email updates or use the many RSS/Atom feeds throughout the site.

**RESEARCH**
Search legislation before Congress. You can also browse and search the Congressional Record. Or view Yesterday’s Activity.

Hot Legislation in the Blogosphere

<table>
<thead>
<tr>
<th>Category</th>
<th>Bill Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Veterans</td>
<td>H.R. 5037: Respect for America's Fallen Heroes Act</td>
</tr>
<tr>
<td>Health policy</td>
<td>S. 1955: Health Insurance Marketplace Modernization and Affordability Act of 2005</td>
</tr>
<tr>
<td>Intelligence activities</td>
<td>H.R. 5020: Intelligence Authorization Act for Fiscal Year 2007</td>
</tr>
</tbody>
</table>
SPARQL

You can browse the underlying data here to see what types of things you can query. There are about 25 million triples in the store at the time I'm updating this. (See the source data page.)

The SPARQL engine's base URL is http://www.govtrack.us/sparql, following (or trying to follow) the SPARQL Protocol spec. The SPARQL engine is Ryan Levering's engine for SESAME, plus my SemWeb library for C#. The data store is persisted in MySQL. Responses are limited to 1000 rows in the hopes that having this public won't break anything.

Enter a SPARQL query below:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT * WHERE {
}
```

Run Query  Display AS: SPARQL XML

Notable Entities

Here are some notable URIs in the data:

- The United States: <tag:govshare.info,2005:data/us>
- Sen. John McCain:
World Cup Fever is here...

By noel.

It's a World Cup year and I feel it in me... the anticipation grows and I can't wait for June...

This is a feeling shared by million around the World and specially here at Opera where we have people from over 30 countries.

To celebrate the "Beautiful Game" we decided to enjoy the World Cup Opera style.
What does this mean you ask? Well... I can't answer just yet 😈

The Goal 06 group page is the first step and more exciting stuff will follow next week. Do you have the fever also? Then join the group, start commenting, post in the Forum, and private msg me if you want to blog or if you have any ideas.

Fred seems to be taking the lead in the office and is walking around in full gear to support Sweden.

RS: For the curious ones, I leave you with some clues: World Cup, Opera Community, Opera Technology.
Opera SPARQL Query Engine

This experimental server supports the SPARQL Query language for RDF. It has a partial SPARQL Protocol implementation, i.e. it supports the HTTP access method. On the backend, it uses the latest release of Redland/Rascal and so, inherits the capabilities and limitations of that library.

The model is built on the Opera Community public data and consists of 7289990 triples.

SPARQL Query
Enter your query:

Note that too complex queries will currently result in a "Proxy Error", while malformed queries will result in a pure-text error message.

Examples

SPARQL is a language still under construction by the W3C and not many are familiar with it. Therefore, let's have a look at
XMLArmyKnife -- SPARQL

SPARQL Query Service

- Introduction
- Base URL
- Request Methods
- Request Parameters
- Response Codes
- Response Format
- Additional Response Formats -- SELECT
- Additional Response Formats -- CONSTRUCT, DESCRIBE
- Examples
- Implementation Notes

Introduction

The XAK SPARQL query service implements the SPARQL Protocol for RDF (14th September 2005) providing SPARQL query processing for RDF data available on the open internet.

The query processor extends the standard protocol to provide support for multiple output formats. At present this uses additional query parameters, although Content Negotiation will eventually be supported.

Base URL

The Base URL of the query service is: http://xmlarmyknife.org/api/rdf/sparql/query

Request Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Supported?</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELETE</td>
<td>NO</td>
<td>A GET of the Base URL without any parameters will return an HTML form suitable for experimenting with the</td>
</tr>
</tbody>
</table>
SPARQL Query Form

This query form is a simple user interface for testing out the XAK SPARQL Query Service. Consult the documentation for details of supported output options.

Data URL

http://www.ldodds.com/ldodds.rdf

Enter SPARQL Query

SELECT * WHERE {?x ?y ?z.}

Display Results As: HTML

Run Query  Reset
Protocol Extension: Query by Ref

- **query-uri**
  - Indicate query to run via URI

- Allows easier sharing of queries

- Simplifies protocol requests
  - Reduces need for POST
  - Enables caching

- Useful level of indirection
  - Dynamically generated queries
  - Parameterised (e.g. Default values)
Protocol Extension: XSLT

- **xslt-uri**
  - XSLT post-processing of Query Results
- **Expose non-protocol parameters to XSLT engine**
  - Allows for parameterised stylesheets
- **Configurable mime-type for response**
  - `<xsl:output media-type="..."/>
  - and/or use a request parameter
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX rss: <http://purl.org/rss/1.0/>
SELECT ?rsstitle ?rsslink
FROM <http://del.icio.us/rss/gbilder>
FROM <http://del.icio.us/rss/ldodds>
WHERE {
    ?rsslink a rss:item
    ; rss:title ?rsstitle
    ; dc:date ?date.
}
ORDER BY DESC(?date)
LIMIT 10
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="rsstitle"/>
    <variable name="rsslink"/>
  </head>
  <results ordered="true" distinct="false">
    <result>
      <binding name="rsstitle">
        <literal>geobloggers:"Network Link"...</literal>
      </binding>
      <binding name="rsslink">
        <uri>http://geobloggers.blogspot.com...</uri>
      </binding>
    </result>
    ...more results...
  </results>
</sparql>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="rsstitle"/>
    <variable name="rsslink"/>
  </head>
  <results ordered="true" distinct="false">
    <result>
      <binding name="rsstitle">
        <literal>geobloggers:"Network Link"...</literal>
      </binding>
      <binding name="rsslink">
        <uri>http://geobloggers.blogspot.com...</uri>
      </binding>
    </result>
    ...more results...
  </results>
</sparql>
<rss:item rdf:about=""...">
  <rss:link>http://geobloggers.blogspot.com...</rss:link>
  <rss:title>geobloggers: "Network Link"...</rss:title>
</rss:item>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX rss: <http://purl.org/rss/1.0/>
SELECT ?title ?description ?lat ?long
WHERE {
}
PREFIX myfn: <java:com.ldodds.sparql.>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX rss: <http://purl.org/rss/1.0/>
SELECT ?title ?description ?lat ?long
WHERE {
  ?point a geo:SpatialThing
  ; dc:title ?title
  ; rss:link ?description
  ; geo:lat ?lat
  ; geo:long ?long.
FILTER (myfn:Distance2(?lat, ?long, "51.510025", 
"-0.126171") < 2) 
}
Protocol Extension: JSON

- JSON format for SPARQL Results
  - Re-expression of XML format as JSON
- Not formal standard, but DAWG produced
  - Based on several existing implementations
- Already well supported
  - ARQ, Redland, ARC (SPARQL for PHP)
Leigh Dodds

British; biology graduate; software engineer; geek; father; semweb enthusiast; former XML-Deviant

Friends and Family

- Brian Kelly
- Dan Brickley
- Danny Ayers
- Dave Beckett
- Edd Dumbill
- Jim Ley
- Libby Miller
- Matt Biddulph
- Morten Frederiksen
- Pete Johnston
- Simon St. Laurent
Leigh Dodds

British; biology graduate...
Friends and Family

- Aaron Swartz
- Amy van der Hiel
- Christoph Bussler
- Dan Connolly
- Daniel J Weitzner
- Eric Miller
- Ivan Herman
- Karl Dubost
- Norman Walsh
Timothy Berners-Lee

Made Following Programmes

- BBC BREAKFAST NEWS - 15 April 1998
- BREAKFAST - 05 October 2000
- DESIGNING OUR LIVES - THE WORD
- GREAT BRITONS - THE TOP 100
- HORIZON - COMPUTERS DON'T BITE, INSIDE THE INTERNET
- INTERNET DREAMERS - INVENTING THE DREAM (1/3)
- MARK LAWSON TALKS TO - SIR TIM BERNERS-LEE
- NEWS - 01 December 2004
- NEWSNIGHT - 09 August 2005
- OU - M206AIP/01
- THE CULTURE SHOW - 16 February 2006
- THE WORLD AT ONE - 21 December 1999
- WORKING LUNCH - 15 October 2003
Category Museums

Museums in London

- Canal Museum
- Dali Universe
- Dana Centre
- Dana Centre, SW7,5HE
- London Dungeon
- London Transport Depot Museum
- London Transport Museum
- Natural History Museum
- RAF Museum
- Science Museum
- Sherlock Holmes Museum
- Tate Modern
- Tea And Coffee Museum
- The British Museum
- The Saatchi Gallery
- Victoria And Albert Museum
A listing of the episodes in the BBC Series, Dr Who

All Dr Who Episodes

- AN UNEARTHLY CHILD, 1
- AN UNEARTHLY CHILD, 2
- AN UNEARTHLY CHILD, 3
- AN UNEARTHLY CHILD, 4
- THE DALEKS, 1
- THE DALEKS, 2
- THE DALEKS, 3
- THE DALEKS, 4
- THE DALEKS, 5
- THE DALEKS, 6
- THE DALEKS, 7
- THE EDGE OF DESTRUCTION, 1
- THE EDGE OF DESTRUCTION, 2
- MARCO POLO, 1
- MARCO POLO, 2
- MARCO POLO, 3
- MARCO POLO, 4
- MARCO POLO, 5
- MARCO POLO, 6
- MARCO POLO, 7
- THE KEYS OF MARINUS, 1
- THE KEYS OF MARINUS, 2
- THE KEYS OF MARINUS, 3
- THE KEYS OF MARINUS, 4
- THE KEYS OF MARINUS, 5
- THE KEYS OF MARINUS, 6
- THE AZTECS, 1
- THE AZTECS, 2
- THE AZTECS, 3
- THE AZTECS, 4
- THE AZTECS, 5
- THE AZTECS, 6
- THE AZTECS, 7
- THE WARRIORS OF DEATH
- THE BRIDE OF SACRIFICE
Implementation Issues

- What Kind of Service?
  - FDS, ADS, or somewhere in-between?

- Mapping Data to RDF
  - D2R, Squirrel RDF, Jena Property Tables, etc.
  - XSLT transformation from existing data

- Privacy
  - Exposing personal data
Implementation Issues

- Security
  - Abuse detection (e.g. Rate limiting; Usage tracking)
  - DOS Attacks
- Query Performance
  - Placing LIMITs on results
  - Restricting processing times
- Caching
  - For Fetched Resources
The Three Things (again)

- That Web 2.0 Needs a Query Language
  - And SPARQL fits very nicely
- How to Implement the SPARQL Protocol
  - And some useful extensions
- Implementation Issues
  - And other Things to Consider...
Questions?
Attributions

http://www.flickr.com/photos/leginmat/72022530/
http://www.flickr.com/photos/tico24/96375501/