Semantic Web Technologies

Lecture 11: SWT for the Life Sciences 4: BioRDF and Scientifc Workflows

Maria Keet

email: keet -AT- inf.unibz.it

home: http://www.meteck.org

blog:

http://keet.wordpress.com/category/computer-science/72010-semwebtech/

KRDB Research Centre
Free University of Bozen-Bolzano, Italy

11 January 2010

Outline

BioRDF
- Considerations
- One RDF-ised base
- Toward federation

Scientific workflows
- Introduction
- Scientific workflows in the Semantic Web setting

The backdrop

- Data integration again (*still...*), but now with new technologies
- 'bio' having a go at RDF, RDFS, SPARQL and the like
- Using technologies for newly developed RDFizers, ontology development tools, triplestore tools (e.g., Sesame, Virtuoso, AllegroGraph), and visualisation tools
- Mesh- vs. mash-ups

Recollecting the Ruttenberg et al (2007) paper (lecture 8)

- Are (should?) "Tools and strategies to extract or translate from non-RDF data sources to enable their interoperability with data organized as statements." (be) part of the set of SW Technologies?
  - Or: where are (W3C) standardization efforts for RDBMS→RDF, excel→RDF, OBO→OWL, structured flat file → language y mappings?
- "BioRDF has the goal of converting a number of publicly available life sciences data sources into RDF and OWL."
  - Thus: *not using* SW Tech but *preparing for use*

Recollecting the Ruttenberg et al (2007) paper (lecture 8)

- D2RQ “The mappings allow RDF applications to access the contents of relational databases using Semantic Web query languages like SPARQL. Doing such a mapping requires us to choose how tables, columns, and values in the database map to URIs for classes, properties, instances, and data values.”
  - Name the pros and cons of RDF triplestores vs RDBMSs

Data integration *strategies* (lecture 8)

I. Physical schema mappings
   - Global As View (GAV)
   - Local As View (LAV)
   - GLAV

II. Conceptual model-based data integration

III. Data federation

IV. Data warehouses

V. Data marts

VI. Services-mediated integration

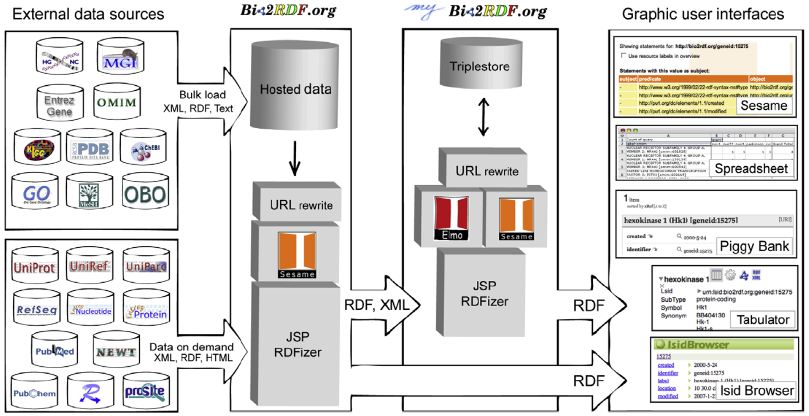
VII. Peer-to-peer data integration

VIII. Ontology-based data integration
   - I or II (possibly in conjunction with the others) through an ontology
   - Linked data by means of an ontology

Overview (Bio2RDF as example)

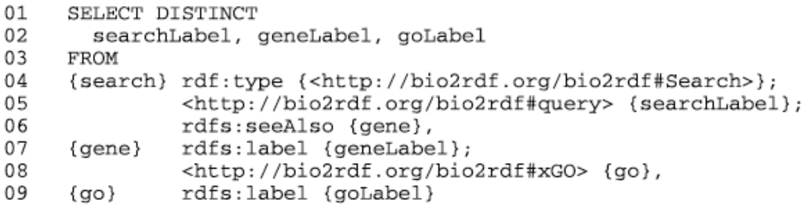
- Data integration with SWT
- What can be achieved with publicly (freely, open source) available tools and data sources
- Main immediate goal of Bio2RDF: convert those data sources to RDF
- How:
  - Bottom-up ontology development of chosen subject domain
  - Decide on architecture: which integration strategy, which data sources, which triplestore, which presentation tools
  - Develop RDF-izer programs (jsp programs HTMLtoRDF, XMLtoRDF, SQLtoRDF, etc)
  - Refinements, a.o.: URI ‘normalization’: multiple URIs denote the same entity, generate new one, link them with owl:sameAs

Bio2RDF architecture



from Belleau et al, 2008

First step in the workflow: SeRQL query to fetch basic data from Entrez and GO



```
SELECT DISTINCT
  searchLabel, geneLabel, goLabel
FROM
{search} rdf:type {<http://bio2rdf.org/bio2rdf#Search>};
         <http://bio2rdf.org/bio2rdf#query> {searchLabel};
         rdfs:seeAlso {gene},
{gene}   rdfs:label {geneLabel};
         <http://bio2rdf.org/bio2rdf#xGO> {go},
{go}     rdfs:label {goLabel}
```

from Belleau et al, 2008

A consideration

- "An ontology belongs to a community who adapts it, uses it and shares it. With the warehouse stored into a triplestore, it is possible to query the local knowledge base with SeRQL queries. However, the semantic web is meant to be distributed. With more RDF resources available on the web and by using the SPARQL [58] language and protocol, a standard defined by the W3C, the data warehousing concept could become obsolete in the future. This is one perspective of the semantic web."

from Belleau et al, 2008

One strategy for a "virtual repository", various architectures and tools[1]

- OWL mappings to provide an integrated list of receptors and executes individual queries against different SPARQL endpoints (receptor explorer)
- AIDA Toolkit for cooperatively search, annotate, interpret, and enrich large collections of heterogeneous documents from diverse locations
- FeDeRate enables a global SPARQL query to be decomposed into subqueries against the remote databases offering either SPARQL or SQL query interfaces.
- Vocabulary of interlinked Datasets (voiD) to create metadata for describing datasets exposed as Linked Data URIs or SPARQL endpoints

[1] Kei-Hoi Cheung, H Robert Frost, M Scott Marshall, Eric Prudhommeaux, Matthias Samwald, Jun Zhao, and Adrian Paschke. A journey to Semantic Web query federation in the life sciences. *BMC Bioinformatics* 2009, 10(Suppl 10):S10

Differences

- The layer at which the virtual repository abstraction is supported:
  - Receptor Explorer: Semantic Service Bus (ESB + Jena + Sesame) API exposed to client applications
  - AIDA toolkit: web services, using the SOA API exposed to client applications by WSDL
  - FeDeRate: at the SPARQL query interface
- Generality of approach:
  - Receptor Explorer is a single scenario with step-wise exploration/navigation
  - AIDA Search interface is general-purpose and can be utilized to explore a wide range of RDF data retrieved from multiple locations (SKOS + Sesame, Virtuoso)
  - FeDeRate also broad range of RDF-based applications that query a range of datasets and repositories

FeDeRate's approach to query management

- Use the common variables in a SPARQL query to decompose it into separate queries for the different data sources; e.g., connecting them by a common human EntrezGene identifier:

```
SELECT ?iupharNm ?type ?label
...
GRAPH <source1> {
     ?iuphar iface:iupharName ?iupharNm .
     ?human iface:iuphar ?iuphar .
     ?human iface:geneName ``GABBR1'' .
     ?human iface:entrezGene ?humanEntrez }
GRAPH <source2> {
     ?gene db:entrezgene ?humanEntrez ;
     ?gene a ?type ;
     ?gene rdfs:label ?label }
```

FeDeRate's approach to query management

- Determines which variables in each GRAPH constraint are present in the SELECT (or are referenced in subsequent graph patterns), then translate the constraint into a subordinate SPARQL query:

```
SELECT ?iupharNm ?humanEntrez
FROM <source1>
WHERE {
     ?iuphar iface:iupharName ?iupharNm .
     ?human iface:iuphar ?iuphar .
     ?human iface:geneName ``GABBR1'' .
     ?human iface:entrezGene ?humanEntrez }
```

FeDeRate's approach to query management

- Subsequent subordinate queries with bindings constraining the variables bound by earlier queries, expressed as standard SPARQL constraints:

```
FROM <source2>
WHERE {
     ?gene db:entrezgene ?humanEntrez ;
     ?gene a ?type ;
     ?gene rdfs:label ?label
FILTER (?humanEntrez = 2550 ||
     ?humanEntrez = 9568)}
```

Consideration

- The billion triple challenge
- Focus on data or on knowledge; how to combine it?
- Scalability
- Date integration strategy

Some recent observations from LODD

- A significant challenge ... is the strong prevalence of terminology conflicts, synonyms, and homonyms. These problems are not addressed by simply making data sets available on the Web using RDF as common syntax but require deeper semantic integration
- For applications that focus on discovery and data navigation, having explicit links between data sources is often already a huge benefit even without semantic integration
- **For other applications that rely on expressive querying or automated reasoning deeper integration is essential** ..., it would be beneficial if more community practices on publishing term and schema mappings would be established

Anja Jentzsch, Bo Andersson, Oktie Hassanzadeh, Susie Stephens, Christian Bizer. Enabling Tailored Therapeutics with Linked Data. LDOW2009, April 20, 2009, Madrid, Spain.

More at: `http://esw.w3.org/topic/HCLSIG/LODD`

Requirements[2]

- Seamless access to resources and service
- Service composition & reuse and workflow design
- Scalability
- Detached execution
- Reliability and fault-tolerance
- User-interaction
- "Smart" re-runs
- "Smart" (semantic) links
- Data provenance

[2] Ludäscher et al. Scientific Workflow Management and the Kepler System.

Some general characteristics of Scientific Workflows[3]

- The ability to handle many and varied analysis tools; not merely database systems that have to be linked up, but the many (custom-made) analysis tools w.r.t. amount of databases
- Interfaces to a diverse range of computational environments (supercomputers, grid, Internet and Semantic Web)
- The ability to handle activity mixes that are different from typical business profiles—and there are, at least initially, few canned and reusable workflows (i.e., design from scratch)
- Need for explicit representation of knowledge at different stages
- Auditability of the computations (when the results are used to make decisions that carry regulatory or legislative implications; e.g., data analysis of clinical trials, climate model predictions)

[3] `http://people.engr.ncsu.edu/mpsingh/papers/databases/workflows/sciworkflows.html`

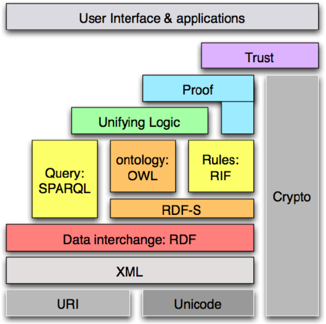
- Do biologists and bioinformaticians need scientific workflows?
- What if we want to run a (scientific) workflow with tools and databases freely available on the Internet?
- What are the problems of (scientific) workflows, if any, and can they be solved with SWT?
- Do scientific workflows need Semantic Web technologies?

Further additions w.r.t. bio and the Semantic Web

- Software version of the "Materials & Methods", i.e., a 'pipeline' of activities that can, is, and has to be, be carried out more than once
- Repeatability of the *in silico* experiment
- Customization of the data sources and methods for each researcher
- 'open' system
- Provenance of the data; or: a need for addressing the **trust** layer in the Semantic Web layer cake

Further additions w.r.t. bio and the Semantic Web

- Software version of the "Materials & Methods", i.e., a 'pipeline' of activities that can, is, and has to be, be carried out more than once
- Repeatability of the *in silico* experiment
- Customization of the data sources and methods for each researcher
- 'open' system
- Provenance of the data; or: a need for addressing the **trust** layer in the Semantic Web layer cake



Where can we plug SWT languages into Scientific Workflows?

- RDF: common data format (for linking and integration)
- SPARQL: querying data
- OWL: representation of the knowledge across the workflow
- Rules: orchestrate the service execution
- Services (e.g., WSDL, OWL-S): to discover useful scripts that can perform a task in the workflow
- The "trust" (provenance) layer: ...... (currently with any of the previous ones)

Examples w.r.t. SWT

- Taverna, on top of ^{my}Grid
  - RDF for data interoperability, for provenance
  - RDF query language for querying the RDF data
  - OWL ontologies for the domain, for the services, for the workflows; for consistency checking, taxonomic classification
  - WSMO's tasks
  - Jena, Sesame, Protégé
- Kepler, on top of Ptolemy
  - services
  - explorations with ontologies
- Wings, on top of Pegasus
  - ontologies (OWL), reasoning (with Jena)

Provenance and trust, system examples

- Taverna: experiment-, workflow-, and knowledge-provenance, representing a mixture of RDF(S) and OWL to represent the overall model, individual provenance graphs of a particular workflow[4]
- PASS experiments, with another provenance ontology for the workflow[5], and Pychinko, a Semantic Web rule engine to orchestrate the service execution[6]

[4] Carole Goble et al. Knowledge Discovery for biology with Taverna. In: *Semantic Web: Revolutionizing knowledge discovery in the life sciences*. 2007, pp355-395.

[5] `http://provenance.mindswap.org/provenance.owl`

[6] Jennifer Golbeck and James Hendler. A Semantic Web Approach to the Provenance Challenge. *Concurrency Computat.: Pract. Exper.*, 2007

Summary

BioRDF

Considerations

One RDF-ised base

Toward federation

Scientific workflows

Introduction

Scientific workflows in the Semantic Web setting