Using/Extending RIF and SPARQL to navigate Linked Open Data

Axel Polleres, DERI, NUI Galway
“Keynote Checklist”

☐ Funny
☐ Educating
☐ Controversial
☐ Inspiring
☐ Credits to the hosts
☐ Advertising
☐ Steal from famous people
Where is the Semantic Web?
more and more RDF available on the Web…
…thanks to GRDDL, Linked Open Data, etc.
… vocabularies (RDFS+OWL) becoming established
… Now: What can we do with it?
Example: Finding experts/reviewers?

Tim Berners-Lee, Dan Connolly, Lalana Kagal, Yosi Scharf, Jim Hendler: **N3Logic: A logical framework for the World Wide Web.** Theory and Practice of Logic Programming (TPLP), Volume 8, p249-269

- Who are the right reviewers? Who has the right expertise?
- Which reviewers are in conflict?
- Observation: Most of the necessary data already on the Web!

- More and more of it follows the Linked Data principles, i.e.:
  1. Use URIs as names for things
  2. Use HTTP dereferenceable URIs so that people can look up those names.
  3. When someone looks up a URI, provide useful information.
  4. Include links to other URIs so that they can discover more things.
RDF on the Web

- (i) directly by the publishers
- (ii) by GRDDL transformations, or
- (iii) by 3rd-party wrapper

FOAF/RDF linked from a home page: personal data (foaf:name, foaf:phone, etc.), relationships foaf:knows, rdfs:seeAlso)
RDF on the Web

- (i) directly by the publishers
- (ii) by GRDDL transformations, or
- (iii) by 3rd-party wrapper

GRDDL (Gleaning Resource Descriptions from Dialects of Languages.), W3C Rec. 2007

Simple principle:
- extract RDF directly from HTML or XML files
- typically using XSLT transformations (other languages: XQuery, XSPARQL, etc.)
- useful for common Microformats, e.g. hCard, hCal:

```xml
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix vc: <http://www.w3.org/2006/vcard/ns#> .

[ vc:VCard; 
  vc:org [ vc:organization-name "Data Access Technologies" ]; 
  vc:fn "Cory B. Casanave"; 
  vc:n [ vc:name; 
    vc:given-name "Cory"; 
    vc:family-name "Casanave"; 
    vc:additional-name "B." ]; 
  vc:title "President & CEO"; 
  vc:adr [ vc:Address; 
    vc:extended-address "Suite 505"; 
    ... ] ]
```
RDF on the Web

- (i) directly by the publishers
- (ii) by GRDDL transformations, or
- (iii) by 3rd-party wrapper

e.g. L3S’ RDF export of the DBLP citation index, using FUB’s D2R (http://dblp.l3s.de/d2r/)

Gives unique URIs to authors, documents, etc. on DBLP! E.g.,
- http://dblp.l3s.de/d2r/resource/authors/Tim_Berners-Lee,
- http://dblp.l3s.de/d2r/resource/publications/journals/tplp/Berners-LeeCKSH08

Provides RDF version of all DBLP data + query interface!
Excellent tutorial here: http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/
How can I find RDF Data?

- Datawarehouse approach, e.g. SWSE
  - crawling, harvesting, SPARQL interface, RDFS+restricted OWL reasoning

- Search/Lookup indeces for the Semantic Web, e.g. Sindice
  - Indexing RDF sources on the Web, go there and query yourself

http://swse.deri.org
http://sindice.com
How can I query that data? SPARQL

- SPARQL – W3C approved standardized query language for RDF:
  - look-and-feel of “SQL for the Web”
  - allows to ask queries like
    - “All documents by Tim Berners-Lee”
    - “All documents published in TPLP that have ‘Web’ in the title”
    - “Names of all persons who co-authored with authors of http://dblp.l3s.de/d2r/…/Berners-LeeCKSH08 or known by co-authors”
    - “All people who have published in TPLP but have not co-authored with any of the authors of http://dblp.l3s.de/d2r/…/Berners-LeeCKSH08”

Example:

SELECT ?D
FROM <http://dblp.l3s.de/…/authors/Tim_Berners-Lee>
WHERE {?D dc:creator <http://dblp.l3s.de/…/authors/Tim_Berners-Lee>}
What can/can’t be done with SPARQL?

- **Can:**
  - FILTERs
  - UNION
  - traversing GRAPHs
  - OPTIONAL, set difference
  - CONSTRUCT new graphs from existing ones

- **Can’t:**
  - Full support of RDFS+OWL
  - recursive “views”, dynamic datasets
  - Aggregates, built-ins
  - Generate XML, HTML
SPARQL “recipes”: FILTERs

“All documents published in TPLP that have ‘Web’ in the title”

PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX swrc: <http://swrc.ontoware.org/ontology#>

SELECT ?T WHERE
  ?D dc:title ?T.
  FILTER ( RegEx(?T, "Web") )
}


SPARQL “recipes”: UNIONs

- “Names of all persons who co-authored with authors of http://dblp.l3s.de/d2r/.../Berners-LeeCKSH08 or known by co-authors”

```sparql
SELECT ?Name WHERE
{
  <http://dblp.l3s.de/d2r/resource/publication/journals/tplp/Berners-LeeCKSH08>
    dc:creator ?Author.
  ?CoAuthor foaf:name ?Name
}
```
SPARQL “recipes”: UNIONs

“Names of all persons who co-authored with authors of http://dblp.l3s.de/d2r/…/Berners-LeeCKSH08 or known by co-authors”

SELECT ?Name WHERE
{
  <http://dblp.l3s.de/d2r/resource/publications/journals/tplp/Berners-LeeCKSH08>
    dc:creator ?Author.
  {  ?CoAuthor foaf:name ?Name . }
}

UNION
{
  ?Person foaf:name ?Name }

Doesn’t work… no foaf:knows relations in DBLP 😞
“Names of all persons who **co-authored** with authors of [http://dblp.l3s.de/d2r ..//Berners-LeeCKSH08](http://dblp.l3s.de/d2r ..//Berners-LeeCKSH08) or known by co-authors, according to FOAF information found on their homepages”

```
SELECT ?Name WHERE {
  { ?CoAuthor foaf:name ?Name . }
  UNION
  { ?CoAuthor foaf:homepage ?Homepage.
    GRAPH ?Homepage
    { ?CoAuthor foaf:knows ?Person.
      ?Person foaf:name ?Name .
    } }
}
```

```
Doesn’t work either… would need dynamic datasets + GRDDL support… at least not (yet) in standard SPARQL engines.
```
SPARQL “recipes”: OPTIONALs

- “Select names of people who co-authored with authors of http://dblp.l3s.de/d2r/.../Berners-LeeCKSH08 and OPTIONALLY their home page” (not all people in DBLP have a homepage)

SELECT ?Name ?Homepage WHERE
{
  ?CoAuthor foaf:name ?Name
  OPTIONAL {?CoAuthor foaf:homepage ?Homepage}
}

- Oops, that is non-monotonic! Closed world! Handle with care on the Web!
“Select names of people who co-authored with authors of http://dblp.l3s.de/d2r/…/Berners-LeeCKSH08 and WHO DON’T HAVE a home page” (not all people in DBLP have a homepage)

SELECT ?Name ?Homepage WHERE

{ <http://dblp.l3s.de/d2r/resource/publication/journals/tplp/Berners-LeeCKSH08> 
  dc:creator ?Author.
  ?CoAuthor foaf:name ?Name
  OPTIONAL {?CoAuthor foaf:homepage ?Homepage}
  FILTER ( ! Bound (?Homepage ) )
}

This emulates “NOT EXISTS” from SQL.

BTW: This can be done even WITHOUT using FILTERS!!! Next slide!
“Select names of people who co-authored with authors of http://dblp.l3s.de/d2r/…/Berners-LeeCKSH08 and WHO DON’T HAVE a home page” (not all people in DBLP have a homepage)

SELECT ?Name ?Homepage WHERE

{ <http://dblp.l3s.de/d2r/resource/publication/journals/tplp/Berners-LeeCKSH08>  
    dc:creator ?Author.  
    ?CoAuthor foaf:name ?Name  
    OPTIONAL {?CoAuthor foaf:homepage ?Homepage}  
    GRAPH <boundchecker.ttl> { ?Homepage ex:is ex:unbound } .  
}

Alternative encoding of set difference:

- boundchecker.ttl contains the single triple _:b ex:is ex:unbound.
- _:b is a blank node - a node without a name (URI) that can only be referenced within the graph in which it is declared.
“I know all people who are my coauthors according to DBLP”

CONSTRUCT
{ <http://www.polleres.net/foaf.rdf#me> foaf:knows ?CoAuthor}
WHERE
{ GRAPH <http://dblp.l3s.de/d2r/resource/authors/Axel_Polleres>
  ?D dc:creator
    <http://www.polleres.net/foaf.rdf#me>.
}

Doesn’t work…DBLP uses a different identifier for me…
“I know all people who are my coauthors according to DBLP”

CONSTRUCT
{ <http://www.polleres.net/foaf.rdf#me> foaf:knows ?CoAuthor }
WHERE
{ GRAPH <http://dblp.l3s.de/d2r/resource/authors/Axel_Polleres>
  ?D dc:creator
  <http://www.polleres.net/foaf.rdf#me>.
}

A FOAF and OWL aware SPARQL engine should be able to do this...

DBLP has: <http://dblp.l3s.de/d2r/resource/authors/Axel_Polleres>
foaf:homepage <http://www.polleres.net>.

My FOAF file has: <http://www.polleres.net/foaf.rdf#me>
foaf:homepage <http://www.polleres.net>.

The FOAF ontology has:
foaf:homepage rdf:type owl:InverseFunctionalProperty.

THUS: <http://dblp.l3s.de/d2r/resource/authors/Axel_Polleres>
owl:sameAs <http://www.polleres.net/foaf.rdf#me>
SPARQL “recipes”: CONSTRUCTING new graphs

- “I know all people who are my coauthors according to DBLP”

CONSTRUCT

{ <http://www.polleres.net/foaf.rdf#me> foaf:knows ?CoAuthor}

WHERE

{ GRAPH <http://dblp.l3s.de/d2r/resource/authors/Axel_Polleres>
  ?D dc:creator
  <http://dblp.l3s.de/d2r/resource/authors/Axel_Polleres>.
}

- … Ok, better, but… how do I integrate this in my FOAF file?
I would actually want to write in my FOAF file:

```
:me a foaf:Person;
:me foaf:name "Axel Polleres";
:me foaf:homepage <http://www.polleres.net>.

CONSTRUCT
  { <http://www.polleres.net/foaf.rdf#me> foaf:knows ?CoAuthor }

WHERE
  { GRAPH <http://dblp.l3s.de/d2r/resource/authors/Axel_Polleres>
    ?D dc:creator
    <http://www.polleres.net/foaf.rdf#me>.
  }

:me foaf:knows <http://dblp.l3s.de/d2r/resource/authors/Robert_Tolksdorf>.
:me foaf:knows <http://dblp.l3s.de/d2r/resource/authors/Adrian_Paschke>.
```

Arbitrary mix of SPARQL and RDF not (yet) standard, but

- Proposals out there: [Schenk+Staab, 2008], [Polleres, et al. 2007]
- should become possible with Rules (RIF)!
Open Linked data with extended RDF Graphs:

- Ultimately:

  ![Diagram]

  - Publish extended graphs “linked” via CONSTRUCTs

  Semantic Web = RDF + CONSTRUCT

- Linked data on steroids!
What can/can’t be done with SPARQL?

- **Can:**
  - FILTERs
  - UNION
  - traversing GRAPHs
  - OPTIONAL, set difference
  - CONSTRUCT new graphs from existing ones

- **Can’t:**
  - Full support of RDFS+OWL
  - recursive “views”, dynamic datasets
  - Aggregates, built-ins
  - Generate XML, HTML
RIF – The Rule Interchange Format

http://www.w3.org/2005/rules/wg

- W3C working group, established December 2005
- like all W3C WGs: industrial and academic participants
- not only rules for RDF, but also production rules, business rules, policies, etc.
- recent “last call working drafts”, 30 July 2008:
  - RIF Basic Logic Dialect (BLD) [Kifer, Boley (eds.), 2008]
  - RIF RDF and OWL Compatibility [de Bruijn (ed.), 2008]
- We only use RIF’s presentation syntax here, more details on RIF, cf. [Boley et al., 2007] as well as the latest RIF drafts.
What can RIF do for you?

- Import RDF similar to “Extended Graphs”
- Custom rules on top of RDF, e.g. RDFS and OWL entailment rules
- More flexible use of built-ins supported than in SPARQL, e.g. for ontology mappings
:me a foaf:Person.
:me foaf:name "Axel Polleres".
:me foaf:homepage <http://www.polleres.net>.
CONSTRUCT

{ <http://www.polleres.net/foaf.rdf#me> foaf:knows ?CoAuthor}
WHERE
{ GRAPH <http://dblp.l3s.de/d2r/resource/authors/Axel_Polleres>
  ?D dc:creator
  <http://www.polleres.net/foaf.rdf#me>.
}

:me foaf:knows <http://dblp.l3s.de/d2r/resource/authors/Robert_Tolksdorf>.
:me foaf:knows <http://dblp.l3s.de/d2r/resource/authors/Adrian_Paschke>.
Import RDF: “Extended Graphs” use case in RIF

Import( <http://dblp.l3s.de/d2r/resource/authors/Axel_Polleres> )

:me#foaf:Person.
:me [ foaf:name ->"Axel Polleres"] .

Forall ?D ?CoAuthor (  
<http://www.polleres.net/foaf.rdf#me> [foaf:knows -> ?CoAuthor] :-
    And (  
        ?D [ dc:creator -> <http://www.polleres.net/foaf.rdf#me> ]
    )

:me [ foaf:knows -> <http://dblp.l3s.de/d2r/resource/authors/Adrian_Paschke> ] .

- Rules Syntax inspired by F-Logic, RDF Triples displayed as FRAMES
- Note: RIF is not meant as an alternative RDF syntax!
E.g. inverseFunctionalProperty:


Ontology mapping

- Mapping names from vCard to FOAF, needs concatenation…
- Not expressible in SPARQL…

\[
\text{CONSTRUCT}\{ ?X \text{ foaf:name } ?FN \}
\]

\[
\text{WHERE } \{ ?X \text{ VCard:Given } ?N. ?X \text{ VCard:Family } ?F.
\quad \text{FILTER}(?FN = \text{fn:concat}(?N," ",?F)) \}
\]

Doesn’t work…
- FILTERs only bind variables, can’t create newbindings
- Not all XPath/Xquery functions available.
Ontology mapping

- Mapping names from vCard to FOAF, needs concatenation...
- not expressible in SPARQL...

CONSTRUCT{?X foaf:name fn:concat(?N," ",?F) }
WHERE {?X VCard:Given ?N. ?X VCard:Family ?F. }

Some non-standard extensions of SPARQL support that...
  - E.g. XSPARQL, see later on.
Ontology mapping

- Mapping names from vCard to FOAF, needs concatenation...
- No problem in RIF:

```prolog
FORALL ?X ?F ?N (  
  ?X [ foaf:name -> External( func:concat( ?N " " ?F) ) :-  
  )
```

Externally defined functions, e.g. Xpath/Xquery built-ins, are allowed!
How do rules (RIF) and SPARQL interplay?

- SPARQL is translateable to Rules! [Polleres, 2007]
  + custom RIF Rules
  + OWL/RDFS entailment Rules in RIF

= Semantic Web toolbox implementable in standard Rules engines!
STARTING POINTS for Advanced SW TOOLS:

- DLVHEX: SPARQL+Rules engine
- Semantic Web Pipes
- XSPARQL
Prototype SPARQL++ engine available:

- dlvhex-semweb: http://sourceforge.net/projects/dlvhex-semweb/

- Based on dlvhex Logic Programming engine, translates SPARQL to Rules in a fully spec compliant way

- Supports:
  - Extended Graphs
  - Adds Built-ins, aggregates to SPARQL
  - Persistent storage (database backend)
  - RIF support being worked on

- Easy integration with other languages, flexible plugin-system:
  - Web service, various output formats, …

- Joint work with TU Vienna, Univ. Calabria
Example: “Author names and optionally their Homepages.”

```
SELECT ?Name ?Homepage WHERE
FROM <http://dblp.l3s.de/d2r/data/publications/journals/tplp/Berners-LeeCKSH08>
{ <http://dblp.l3s.de/d2r/resource/publications/journals/tplp/Berners-LeeCKSH08>
   dc:creator ?Author.
   ?Author foaf:name ?Name.
   OPTIONAL {?Author foaf:homepage ?Homepage}
}
```
Translating SPARQL to Rules in dlvhex:

Example: “Author names and optionally their Homepages.

```prolog
namespace("foaf","http://xmlns.com/foaf/0.1/");
namespace("dc","http://purl.org/dc/elements/1.1/");
namespace("swrc","http://swrc.ontoware.org/ontology#");
answerVars("", "Homepage", "Name").
answer_1(X_Author,default) :-
"triple"("<http://dblp.l3s.de/d2r/resource/publications/journals/tplp/Berners-LeeCKSH08>",
"dc:creator",X_Author,default).
answer_2(X_Author,X_Name,default) :- "triple"(X_Author,"foaf:name",X_Name,default).
answer_3(X_Author,X_Homepage,default) :- "triple"(X_Author,"foaf:homepage",X_Homepage,default).
answer_bjoin_1(X_Author,0,X_Name,default) :-
    answer_1(X_Author,default),answer_2(X_Author,X_Name,default),
    not answer_3_prime(X_Author,X_Name,default).
answer_3_prime(X_Author,X_Name,default) :-
    answer_1(X_Author,default),answer_2(X_Author,X_Name,default),answer_3(X_Author,X_Homepage,default).
answer_bjoin_1(X_Author,X_Homepage,X_Name,default) :-
    answer_1(X_Author,default),answer_2(X_Author,X_Name,default),answer_3(X_Author,X_Homepage,default).
answer(X_Author,X_Homepage,X_Name) :- answer_bjoin_1(X_Author,X_Homepage,X_Name,default).
```

Don’t read the code, important issues are:

- Each subpattern of the query yields a PROLOG-like rule
- Some trickery necessary to “emulate” SPARQL semantics: multisets, OPTIONALS, UNIONs
- E.g. use of negation as failure (not) for expressing OPTIONAL
You are welcome to try/extend/help:

http://sourceforge.net/projects/dlvhex-semweb/
Transformations between XML/HTML and RDF cumbersome:
Until recently, XSL the only standard means

*Combining* SPARQL and XQuery (standardized beginning of 2007) offer new possibilities

Their combination: http://xsparql.deri.org

Joint work with Thomas Krennwallner, Waseem Akhtar, Jacek Kopecky, Nuno Lopes
Live data processing with Semantic Web Pipes

- Live, reusable, composable transformation of Semantic Web Sources
- Piping: connecting processing components
- Web Piping: connecting components using web technologies
  - Data has URLs
  - Components have URLs
- Data mashups, reuse of components
- http://pipes.deri.org/

Joint work with Danh Le Phuoc, Giovanni Tummarello, Christian Morbidoni.
Semantic Web Pipes

- Allow users to build mashups from a set of base operators
- Can be stored and republished under a persistent URI and executed dynamically

Example:

http://pipes.deri.org/
Challenges/What’s missing? The small picture:

- **Theoretical challenges:**
  - Aggregates in SPARQL: What do they mean?
  - SPARQL + OWL, RIF+ OWL: What does it mean?

- **Technical challenges:**
  - Semantic data endpoint description
  - Scalability, Query distribution (e.g. DARQ good starting point)
  - Good tools, easy enough to use for your grandparents

- **Integration with other paradigms:**
  - RDB2RDF (D2R good starting point), RDF2XML (XSPARQL good starting point)
Challenges/What’s missing? The BIG picture:

- Our use case: Finding Experts/reviewers:
  - What are the right metrics for expertise?
  - Trust? Change?

- Needs:
  - Temporal, Provenance, Policy extensions to SPARQL, RDF, OWL
  - Dealing with contradictions, uncertainty, incompleteness
  - Understanding the Web’s structure
  - Understanding social networks
  - Understanding the Web’s economics

How much information resides in the “long tail”?

In: 44 million pages
Out: 44 million pages
Central core: 56 million pages

Inspiring? ✓
A lot of challenges waiting ahead!

Can you take them?

- Thanks to:
  - Renaud Delbru, Andreas Harth, Aidan Hogan, Thomas Krennwallner, Nuno Lopes, Alessandra Martello, Danh Le Phuoc, Giovanni Tummarello, Roman Schindlauer, etc., members of the RIF working group, members of the SPARQL working group (esp. Andy Seaborne, Eric Prud’hommeaux, Lee Feigenbaum), etc.