The D2RQ mapping language

Richard Cyganiak

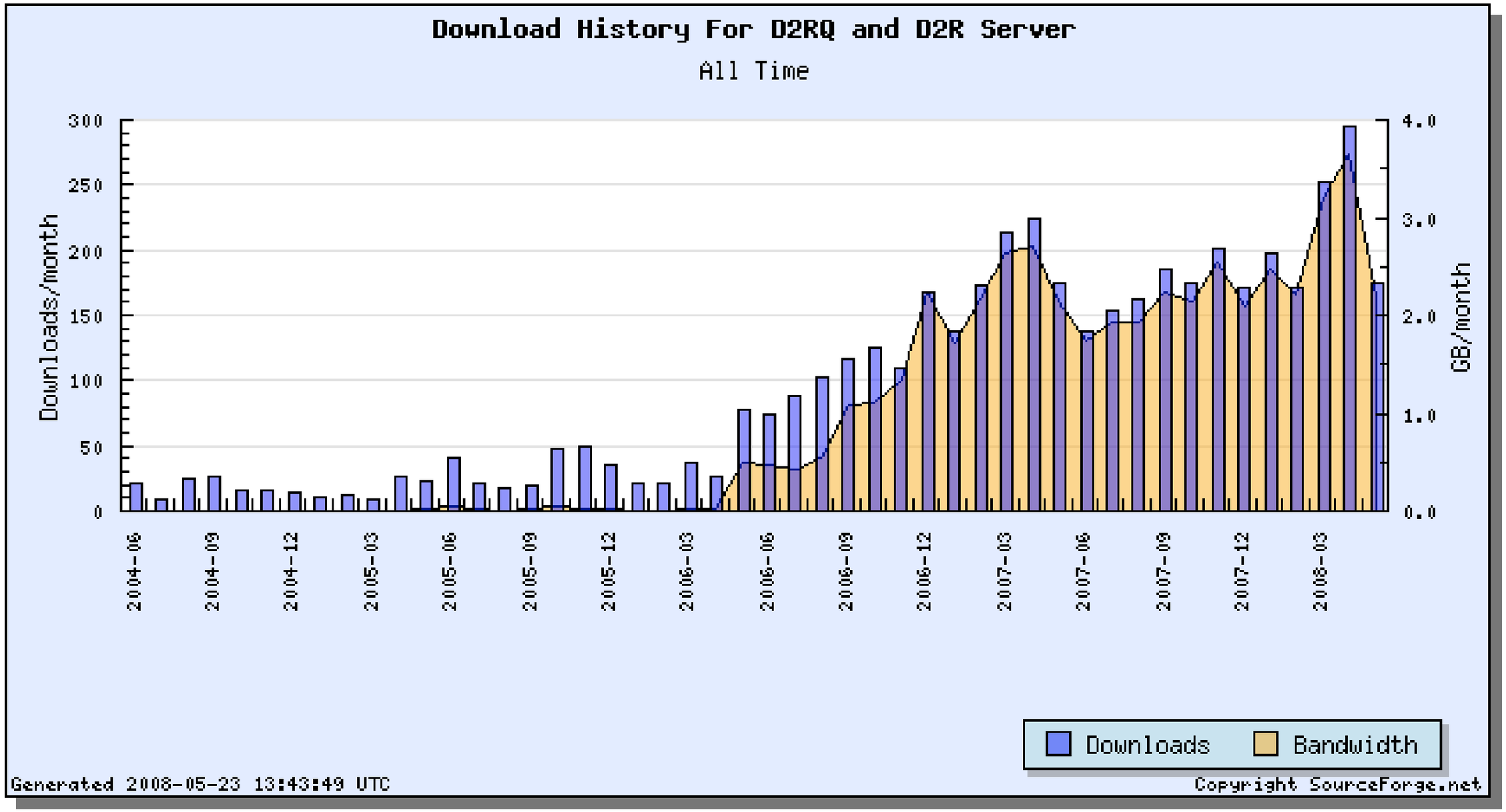
Presentation to W3C RDB2RDF XG, 23 May 2008

D2RQ

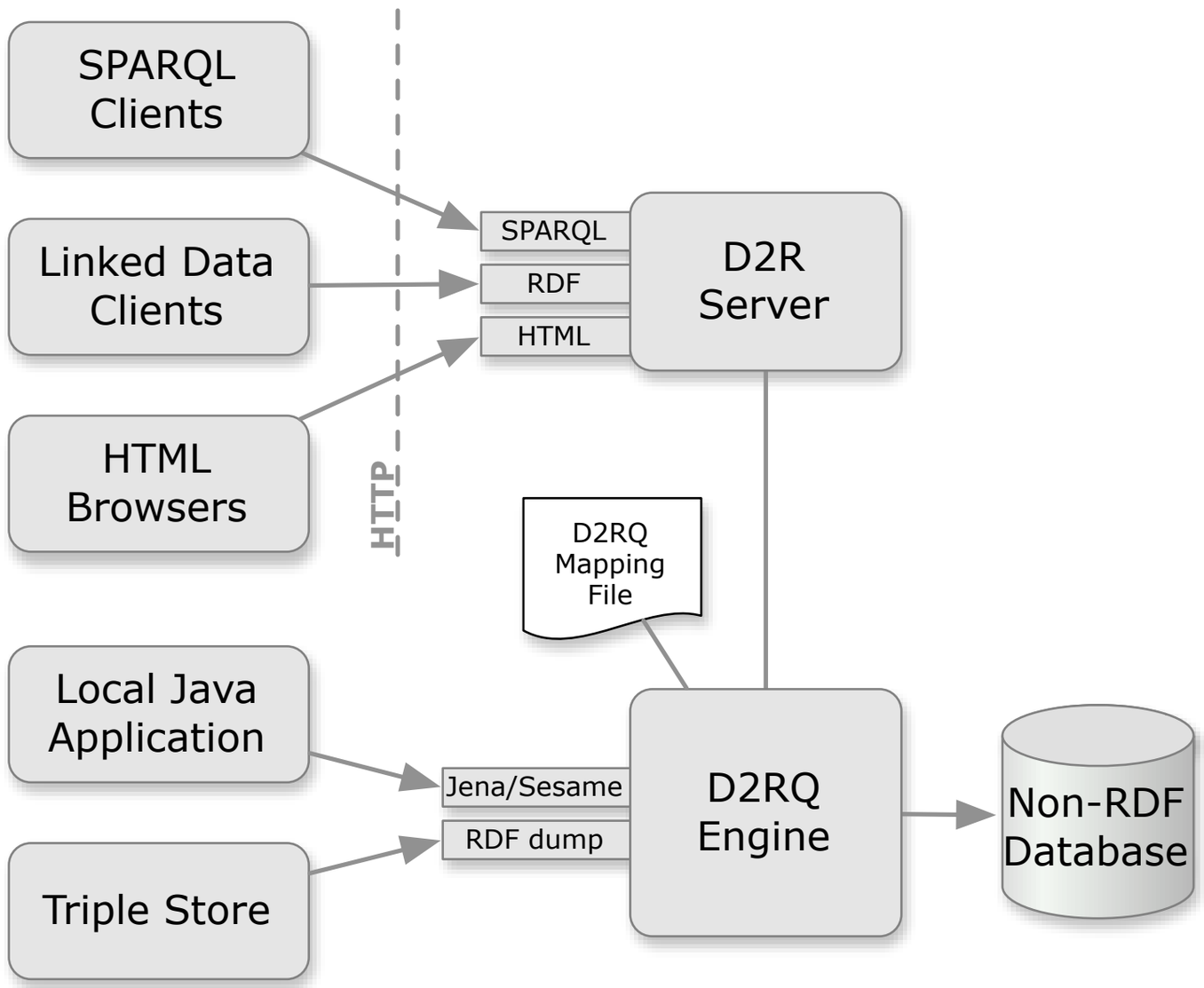
- DB-to-RDF mapper written in Java
- In: any JDBC database
- Out: SPARQL, Linked Data, or Jena API
- GPL, popular, easy to get started
- SPARQL-to-SQL algorithm not state of the art
- Axiom: We never modify the database

The project

- Started 2004 (roots: 2002) by Chris Bizer at FU Berlin; later me at FU and HP Labs
- 200+ downloads/month, 4600+ total
- mailing list at ~20 msgs/month, 700+ total
- In LOD cloud, TopBraid Composer etc


Download History For D2RQ and D2R Server
All Time
Downloads/month
GB/month
300
250
200
150
100
50
0
4.0
3.0
2.0
1.0
0.0
2004-06
2004-09
2004-12
2005-03
2005-06
2005-09
2005-12
2006-03
2006-06
2006-09
2006-12
2007-03
2007-06
2007-09
2007-12
2008-03
Downloads
Bandwidth
Generated 2008-05-23 13:43:49 UTC
Copyright SourceForge.net

Architecture



Architecture (2)

- maps DB to *virtual* RDF graph
- easy to offer arbitrary interfaces to the RDF graph
- most requested: SPARQL and RDF dumps

Mapping language

Mapping language

- N3 based syntax
- Very flexible
- Language is not easy, wish we had a GUI
- Usual workflow: auto-generate mapping from DB schema, then customize

Flexible mappings!

- Properties of one class from multiple tables
- Several classes in the same table
- Value translations, SQL expressions
- Arbitrary joins and SQL conditions
- *This is a MUST HAVE!* Users need it

Still missing

- SQL subqueries in WHERE or FROM
- Determine RDF/OWL property based on type code in a table
- …

To SQL or not to SQL?

- Users want to deal with complexity by using their SQL knowledge
- They want to write arbitrary SQL queries
- We don’t want to parse SQL *(painful!)*
- We force users to decompose their query into small fragments

Mapping process

1. Define your entities
2. Add properties to entites
3. Link entities together
4. Get fancy with conditions, joins, value translations

1. Define your entities

```
map:People a d2rq:ClassMap;
   d2rq:uriPattern "http://.../people/@@User.ID@@";
   d2rq:class foaf:Person;
   d2rq:condition "User.deleted=0".
```

(SQL fragments in red, RDFS/OWL vocabulary in blue)

2. Add properties

```
map:name a d2rq:PropertyBridge;
   d2rq:column "User.name";
   d2rq:property foaf:nick .

map:mbox a d2rq:PropertyBridge;
   d2rq:uriPattern "mailto:@@User.email@@";
   d2rq:property foaf:mbox .
```

3. Link your entities

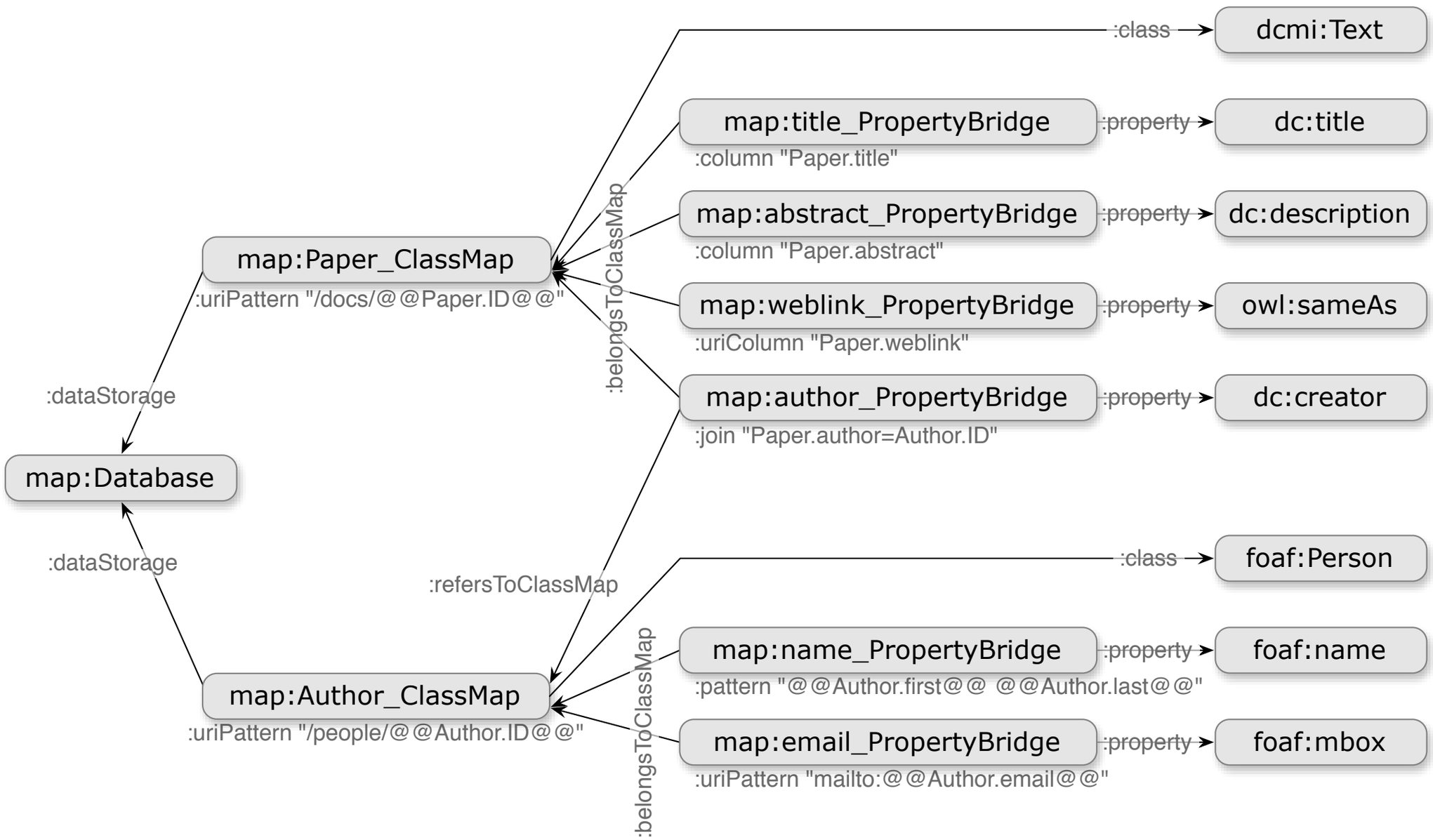
```
map:photo a d2rq:PropertyBridge;
  d2rq:refersToClassMap map:Photos;
  d2rq:property foaf:made;
  d2rq:join "User.ID = Photo.UserID".
```

(also d2rq:alias for self-joins)

4. Get fancy

```
map:mbox_sha1 a d2rq:PropertyBridge;
  d2rq:sqlExpression
            "SHA1(CONCAT('mailto:', User.email))";
  d2rq:property foaf:mbox_sha1sum .
```

Mapping file overview



SQL composition

- Take one or more PropertyBridges with its ClassMap
- SELECT *columns/expressions*
  FROM *all mentioned tables*
  WHERE *joins*
  AND *conditions*
- (and deal with aliases)

Terminology warning!

- A ClassMap doesn't necessarily correspond to an RDFS/OWL class
- Neither necessarily to a single table
- Rather: A set of entities/resources generated in the same way from the DB

Mapping language warts

- Remnants from pre-SPARQL days (constraints)
- Remnants from early lack of JDBC schema introspection (column types)
- Designed for MySQL 3's primitive SQL (e.g. we didn't consider temporary views)

Summary

- D2RQ: four years, 4600+ downloads
- Powerful mapping language is important
- To SQL or not to SQL?
- In the future there will be GUIs

Links

- D2RQ homepage
  http://www4.wiwiss.fu-berlin.de/bizer/d2rq/
- D2RQ manual & language spec
  http://www4.wiwiss.fu-berlin.de/bizer/d2rq/spec/
- Mailing list
  d2rq-map-devel@lists.sourceforge.net