

D2RQ

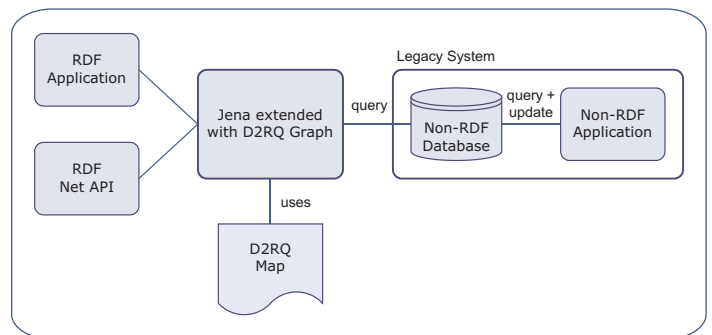
Treating Non-RDF Databases as Virtual RDF Graphs

It will be crucial for many real-world Semantic Web applications to be able to access the content of non-RDF relational databases, used by most legacy systems, without having to replicate the whole database into RDF. This poster presents D2RQ, a declarative language to describe mappings between application-specific relational database schemata and RDF-S/OWL ontologies. Using D2RQ, Semantic Web applications can:

- query a non-RDF database using the RDQL query language
- publish the content of a non-RDF database on the Semantic Web using the RDF Net API
- access information in a non-RDF database using the Jena model API
- do RDFS and OWL inferencing over the content of a non-RDF database using the Jena ontology API

Architecture

D2RQ is implemented as a Jena graph, the basic information representation object within the Jena Semantic Web framework. A D2RQ graph wraps one or more local relational databases into a virtual, read-only RDF graph. D2RQ rewrites RDQL queries and Jena API calls into application-datamodel-specific SQL queries. The result sets of these SQL queries are transformed into RDF triples which are passed up to the higher layers of the Jena framework.



Example

RDQL Query

```
SELECT ?x
WHERE (<http://www.papers.org/3465>,
ex:author, ?x)
```

SQL Query

```
SELECT Persons.Email
FROM PersonPaper, Persons
WHERE PersonPaper.PersonID =
Persons.PersonID
AND PersonPaper.PaperID = 3465;
```

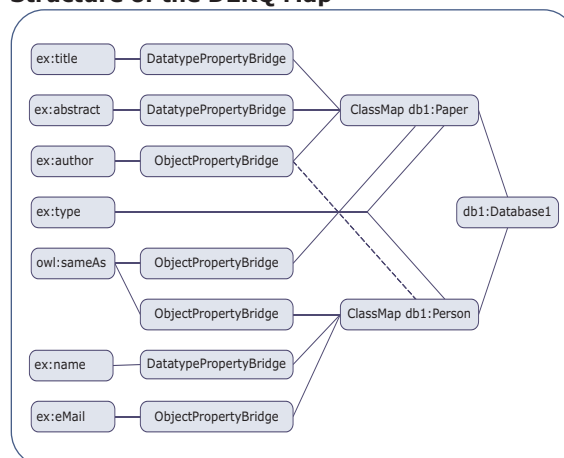
Databases Tables

Persons	PersonID	Name	Email
1	Andy Seaborne	andy.seaborne@hp.com	
2	Christian Bizer	chris@bizer.de	

Papers	PaperID	Title
3465	D2RQ-Treating Non-RDF Databases as Virtual RDF Graphs	
5728	The Impact of Semantic Web Technologies on JDB APIs	

PersonPaper	PaperID	PersonID
1	3465	2
2	5728	2

Structure of the D2RQ Map



Part of the D2RQ Map in N3

```
ex:Paper rdf:type rdfs:Class ;
d2rq:classMap db1:Paper .
db1:Paper rdf:type d2rq:ClassMap ;
d2rq:uriPattern
"http://www.papers.org/
@@Papers.PaperID@" ;
d2rq:dataStorage db1:Database1 .
ex:author rdf:type rdf:Property ;
d2rq:propertyBridge db1:author .
db1:author rdf:type
d2rq:ObjectPropertyBridge;
d2rq:belongsToClassMap db1:Paper ;
d2rq:refersToClassMap db1:Person ;
d2rq:join "Papers.PaperID =
PersonPaper.PaperID" ;
d2rq:join "PersonPaper.PersonID =
Persons.PersonID" .
ex:Person rdf:type rdfs:Class ;
d2rq:classMap db1:Person .
db1:Person rdf:type d2rq:ClassMap ;
d2rq:uriPattern
"mailto:@@Person.Email@" ;
d2rq:dataStorage db1:Database1 .
```

Benchmark Results

We performed a series of benchmarks comparing the performance of D2RQ to the performance of the Jena2 database backend. As benchmarking dataset we used the descriptions of 200,000 papers from the DBLP Computer Science Bibliography. The data was stored using both an application-specific relational data model and the Jena database backend (1.6M triples).

	Jena2 DB	D2RQ
1. find (s ? ?)	1.83 ms	0.01 ms
2. find (? p o)	1.94 ms	0.97 ms
3. find (? p ?)	42431 ms	72 ms
4. find (? ? o)	1.72 ms	3.23 ms