Jena

http://openjena.org/

The Beginning ...

From: McBride, Brian <bwm@hplb.hpl.hp.com>
Date: Mon, 28 Aug 2000 13:40:03 +0100
To: "RDF Interest (E-mail)" <www-rdf-interest@w3.org>

A few weeks ago I posted some suggestions for an improved java RDF API.
I've placed an implementation of these ideas at
http://www-uk.hpl.hp.com/people/bwm/rdf/jena/index.htm
<http://www-uk.hpl.hp.com/people/bwm/rdf/jena/index.htm> .

The code supports in memory models. David Megginson's RDFFilter is hooked in so it can parse RDF serializations. I suggest you use the version of RDFFilter supplied in the distribution as it has some minor bug fixes. Its alpha code - it gets through the regression tests and runs the samples but hasn't had much use other than that. An SQL implementation may follow.

I think better tools will help encourage the adoption of RDF. I'm looking on this as an experiment to see whether this API brings any benefits. So I'd like some feedback.

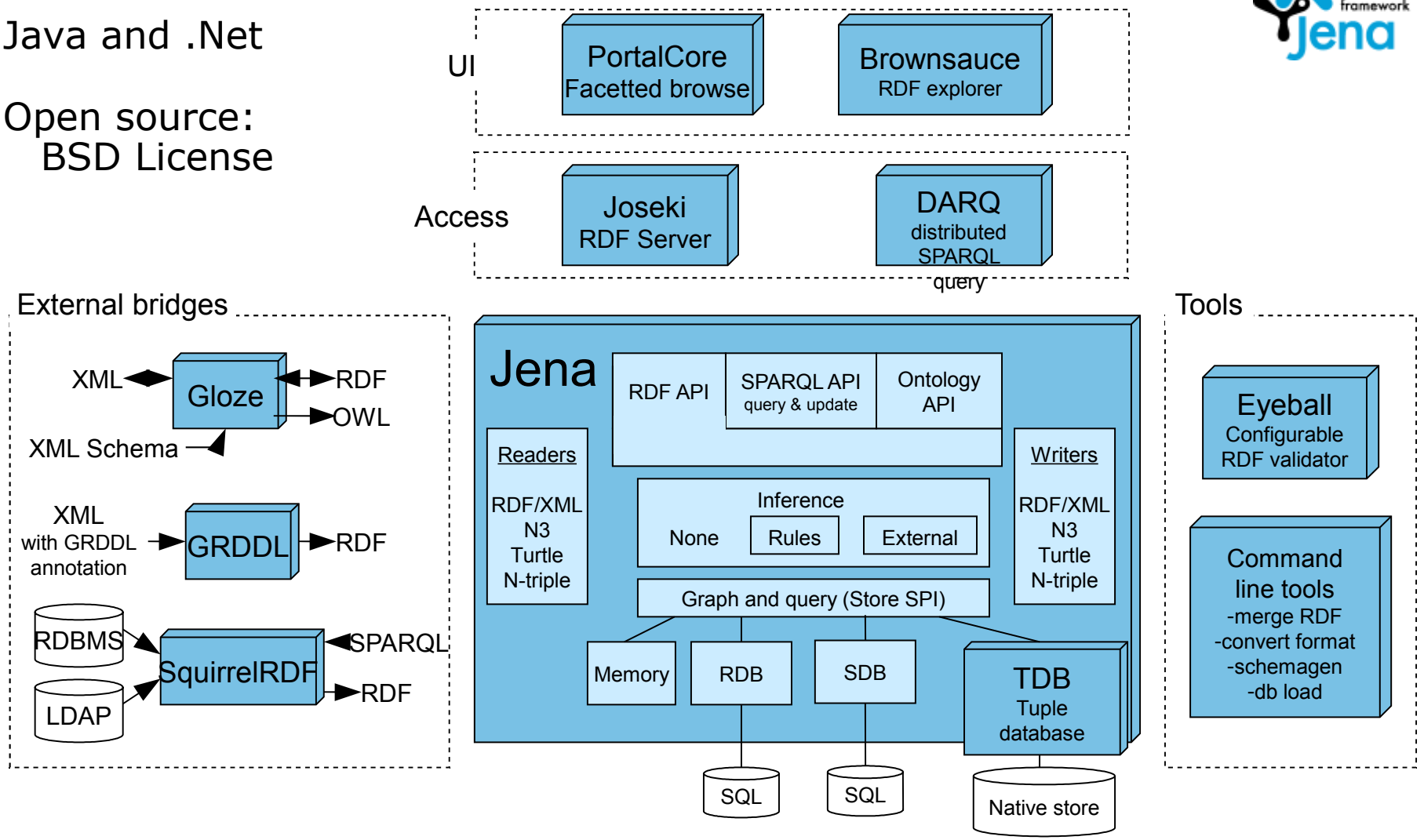
Brian McBride
HPLabs

http://lists.w3.org/Archives/Public/www-rdf-interest/2000Aug/0128.html

Jena

Java and .Net

Open source:
BSD License



SPARQL 1.1

SPARQL 1.1

- Complete query
  - Property paths
  - Aggregatrion, groups, subquery
- Service description
- Federated Query
- Update
  - HTTP update
    - Coarse-grained graph level
  - SPARQL Update Language
    - Fine-grained triple level manipulation
- Entailment

GROUP BY and Aggregation

```
SELECT (COUNT(*) AS ?C) { ?s ?p ?o }
```

```
SELECT ?s (COUNT(*) AS ?C)
{ ?s ?p ?o }
GROUP BY ?s
```

```
PREFIX : <http://people.example/>
PREFIX : <http://people.example/>
SELECT ?y ?minName
WHERE {
  :alice :knows ?y .
  {
    SELECT ?y (MIN(?name) AS ?minName)
    WHERE {
      ?y :name ?name .
    } GROUP BY ?y
  }
}
```

Property Paths

```
ex:me foaf:knows/foaf:name ?name .
```

```
?x rdf:type/rdfs:subClassOf* :SomeClass .
```

```
:aList rdf:next*/rdf:first ?member .
```

Negation

```
PREFIX  foaf:  <http://xmlns.com/foaf/0.1/>
PREFIX  rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

# People with no declared name
SELECT *
{
  ?x rdf:type foaf:Person .
  FILTER NOT EXISTS {?x foaf:name ?name}
}
```

Negation

```
PREFIX  foaf:  <http://xmlns.com/foaf/0.1/>
PREFIX  rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

# People with no declared name
SELECT *
{
  ?x rdf:type foaf:Person .
  MINUS {?x foaf:name ?name}
}
```

Federated Query

```
PREFIX  iuphar:     <http://iuphar.example/ns#>
PREFIX  entrez:     <http://entrez.example/ns#>
PREFIX  void:       <http://rdfs.org/ns/void#>
PREFIX  dcterms:    <http://purl.org/dc/terms/>

SELECT ?service ?id ?iuphar
WHERE {
  # Find the service with the expertise.
  [] dcterms:subject ?gene ;
     void:sparqlEndpoint ?service
  FILTER (?gene IN (entrez:h2550, entrez:h9568) )

  # Query that service for species and iuphar.
  SERVICE ?service {
    ?receptor  iuphar:species    ?species .
    ?species   iuphar:name       ?iuphar .
    ?species   entrez:id         ?id .
  }
}
```

Update Language

```
PREFIX  :       <http://example/>
PREFIX  foaf:   <http://xmlns.com/foaf/0.1/>
PREFIX  rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

DELETE WHERE
{
   :person foaf:name ?x .
}

INSERT DATA
{
   :person foaf:name “Andy” .
}
```

HTTP Update

- GET, PUT, DELETE, POST graphs
- Naming:

```
http://host/store/g1

http://host/store?graph=http://example/g1
http://host/store?graph=http%3A%2F%2Fexample%2Fg1
```

TDB

Tuples Database

http://openjena.org/TDB/

SDB

SPARQL

```
PREFIX  dc:
<http://purl.org/dc/elements/1.1/>

SELECT *
 WHERE {
    ?x dc:title ?title ;
       dc:date ?date
}
```

SQL

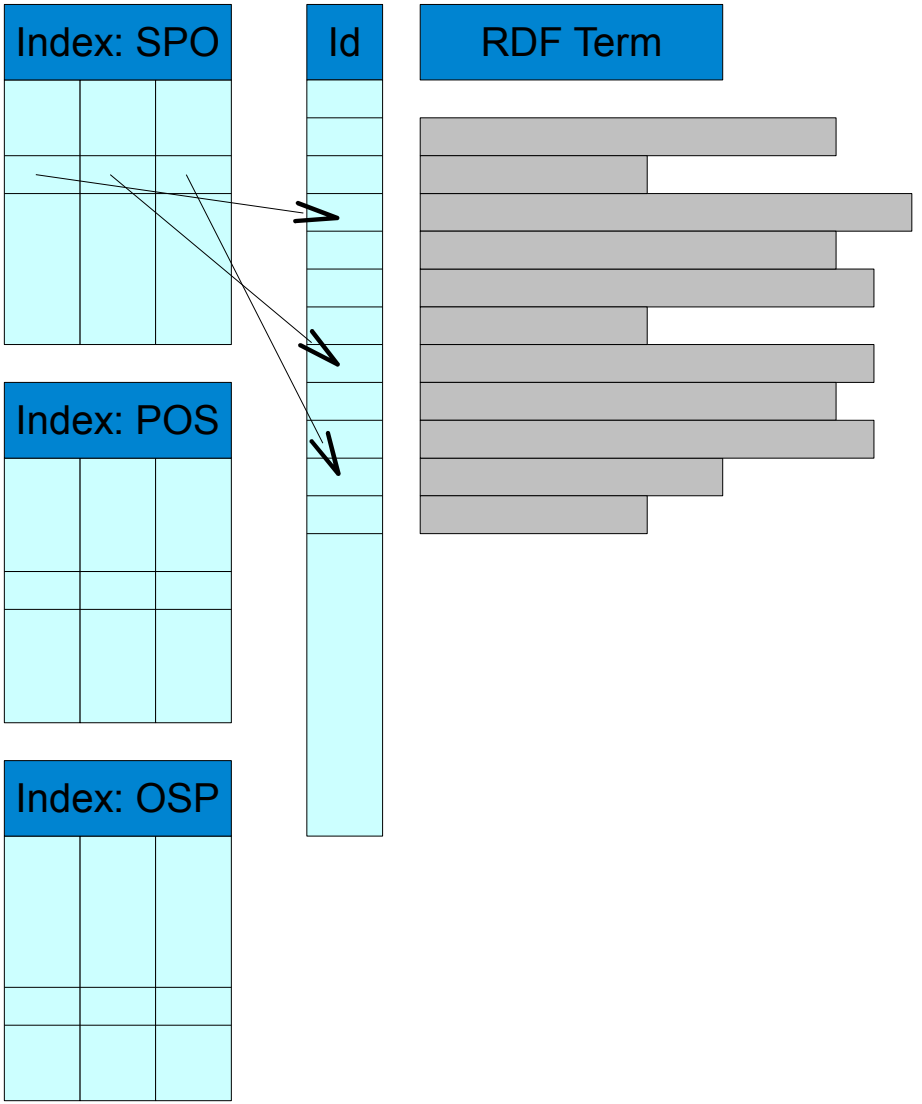
```
SELECT
  R_1.lex AS V_1_lex, R_1.datatype AS V_1_datatype,
  R_1.lang AS V_1_lang, R_1.type AS V_1_type,
  R_2.lex AS V_2_lex, R_2.datatype AS V_2_datatype,
  R_2.lang AS V_2_lang, R_2.type AS V_2_type,
  R_3.lex AS V_3_lex, R_3.datatype AS V_3_datatype,
  R_3.lang AS V_3_lang, R_3.type AS V_3_type
FROM
    Triples AS T_1
  INNER JOIN
    Triples AS T_2
  ON ( T_1.p = -5696047020500897841
   AND T_2.p = -4503196271826130170
   AND T_1.s = T_2.s
   )
  LEFT OUTER JOIN
    Nodes AS R_1
  ON ( T_1.s = R_1.hash )
  LEFT OUTER JOIN
    Nodes AS R_2
  ON ( T_1.o = R_2.hash )
  LEFT OUTER JOIN
    Nodes AS R_3
  ON ( T_2.o = R_3.hash )
```

TDB

- Lessons of the past …
  - SQL databases limited for RDF-style workloads
    - RDF workload Complete indexes; few, very large tables
  - SQL access costs (non-embedded databases) too high
  - SQL transactions can not be used for loading
  - ⇒ Modify the database internally or replace
- TDB
  - UI workloads
  - Non-transactional
  - 64 bit
  - Exploit modern operating system features

TDB

- 3 Index Design for triples
- RDF Terms ids allocated by sequence
  - Inline some xsd datatypes: integers, decimals, dates, datetimes
  - Never loose precision
- B+Tree
  - 100 way
  - Optimized range scans
  - Memory mapped I/O
- Cost-based, rule-based optimizer



32 bit vs 64 bit

- 64 bit: memory mapped files for B+Tree indexes
  - OS in control of paging
  - Segments of 8Mbytes
  - ==> the process size can look like many times the heap size.
- 32 bit: in-JVM caching of index blocks
  - Memory mapped files: JVM limitation of about 1.2Gbytes
  - LRU cache

TDB Bulk loader

- Efficient loading into empty databases
- Uses knowledge of index structure
  - Better use of RAM
  - Speeds: from 50K triples/s, slowing to 10K as total data grows

TDB Optimizer

- High level optimizations: algebra transformations
- Low level optimizations
  - Best order for a BGP
  - Rule based

```
(prefix ((xsd: <http://www.w3.org/2001/XMLSchema#>))
  (stats
    (meta
      (timestamp "2010-06-20T16:35:37.864+01:00"^^xsd:dateTime)
      (run@ "2010/06/20 16:35:37 BST")
      (count 10000))
    (<http://www.w3.org/1999/02/22-rdf-syntax-ns#type> 2101)
    (<http://purl.org/dc/elements/1.1/creator> 1106)
    (<http://musicbrainz.org/mm/mq-1.1#status> 63)
    (<http://musicbrainz.org/mm/mm-2.1#artistType> 124)
    (<http://purl.org/dc/elements/1.1/comment> 12)
    (<http://musicbrainz.org/mm/mm-2.1#trackList> 62)
    (<http://musicbrainz.org/mm/mm-2.1#trmidList> 682)
    (<http://purl.org/dc/elements/1.1/title> 1454)
    (<http://musicbrainz.org/mm/mm-2.1#endDate> 15)
    (<http://musicbrainz.org/mm/mm-2.1#beginDate> 76)
    (<http://musicbrainz.org/mm/mm-2.1#duration> 1009)
    (<http://musicbrainz.org/mm/mm-2.1#sortName> 348)
    (other 0)))
```

RDF Datasets

- A dataset is one un-named graph and zero or more named graphs (defined by SPARQL)
- TDB support RDF datasets
- Dynamic union
  - Default graph as union of all named graphs
  - `<urn:x-arq:UnionGraph>` the union of all named graphs
  - `<urn:x-arq:DefaultGraph>` is the concrete default graph
  - In SPARQL and named graph API

Variations

- TDB-BDB : http://github.com/afs/TDB-BDB
- TDB using Oracle Berkeley DB Java Edition
  - Swaps native B+Tree for BDB indexes
  - Uses BDB storage for RDF terms
  - Higher concurrency possible
  - Enables transactions
  - Contributions welcome!

RIOT

- New high performance parsers for N-Triples, N-Quads, Turtle andTriG
  - Parsers do higher level checking of URIs and literals
  - Warn or skip on bad RDF terms

Bad literals

```
"abc"^^xsd:integer
"1,000"^^xsd:integer
"Tue, 15 Jun 2010"^^xsd:date
"2010-02-30"^^xsd:date
"   123"^^xsd:int
```

Bad IRIs (or at least not recommended)

```
<http://example/foo bar>
<http://EXAMPLE.org/>
<http://example.org:80/
<http://example.org/abc[]def>
```