On Storing Voluminous RDF Descriptions: The case of Web Portal Catalogs
(http://www.ics.forth.gr/proj/iss/RDF)

Sofia Alexaki, Vassilis Christophides
Gregory Karvounarakis, Dimitris Plexousakis
Computer Science Department, University of Crete
and
Institute for Computer Science - FORTH
Heraklion, Crete, Greece

Portalmania!
Internet Portals Example: The Open Directory

Browsing the ODP Topics
Searching the ODP Topics & URLs

- Descriptions in ODP consist of the classification of URLs to topics, a textual description and various administrative information
What is needed?

- Flexible Modeling of Web Portal Catalogs using W3C standards (RDF/S)
  1. Exploit existing forms of domain knowledge
     - Ranging from simple vocabularies to formal ontologies
  2. Describe information resources in various ways
     - Administration, Classification, Content Rating, Channels, ....
- Secondary Storage Management of Portal Metadata
  1. Large Schemas: e.g., 170 Mbytes of ODP Topics (the Art Hierarchy contains 25315 terms)
  2. Voluminous Description Bases: e.g., 700 Mbytes of ODP indexed sites (2,342,978 URLs)
- Declarative Query Languages for Portal Catalogs
  1. Interleave schema with data querying
  2. Optimize access to Portal Catalogs

Our Approach

- Use W3C Standards to describe (RDF/S) & exchange (XML) information
- Our Main Contribution: Declarative Languages for Browsing & Querying
Outline

- The Open Directory Portal: a case study
- The RDF Query Language (RQL)
- RDF Storage Strategies
- Testbed: the ODP RDF dump
- Representative queries
- Performance
- Summary and Outlook

Modeling the ODP Catalog with RDF/S

d: http://www.w3.org/1999/02/22-rdf-syntax-ns#
dsf: http://www.w3.org/2000/01/rdf-schema#

Class

Regional Recreation
Regional Lodging
Regional Vacation-Rentals
Regional Travel
Regional Hotel Directories

related

Paris
Travel
Hotel

related

Lodging
Vacation-Rentals
Tours-de-France

related

Ext.Resource

title

description

title
date

title

title

string

file_size

last_modified

string

string


&rs1: http://www.sunscale.com/france/paris/index.htm
Resource Description Framework (RDF/S)

- **RDF**: Resource Descriptions
  - **Data Model**: Directed Labeled Graphs
    - **Nodes**: Resources (URIs) or Literals
    - **Edges**: Properties – Attributes or Relationships
    - **Labels**: Nodes (Class names) and Edges (Property names)
    - **Statement**: assertion of the form resource, property, value
    - **Description**: collection of statements concerning a resource
  - **XML syntax**
  - **RDF Schema (RDFS)**: Schema Vocabularies
    - **Specialization** of both classes & properties (simple & multiple)
    - **Multiple classification** under several classes
    - **Unordered, optional, and multi-valued properties**
    - **Domain and range polymorphism** of properties

RDF/S vs. Well-Known Formalisms

- **Relational or Object Database Models** (ODMG, SQL)
  - Classes don’t define table or object types
  - Instances may have associated quite different properties
  - Collections with heterogeneous members

- **Semistructured or XML Data Models** (OEM, UnQL, YAT, XML Schema)
  - Labels on both nodes and edges
  - Schema class and property subsumption is not captured
  - Heterogeneous descriptions reminiscent of SGML exceptions

- **Knowledge Representation Languages** (Telos, DL, F-Logic)
  - Absence of complex values and n-ary relationships (bags, sequences)
The RDF Query Language: RQL

- Declarative query language for RDF description bases
  - relies on a typed data model (see SemWeb2001 paper)
  - follows a functional approach (basic queries and filters)
  - adapts the functionality of semistructured or XML query languages to RDF, but also:
    - treats properties as self-existent individuals
    - exploits taxonomies of node and edge labels
    - allows querying of schemas as semistructured data
- Relational interpretation of schemas & resource descriptions

Portal Navigation with RQL

- Browsing large description bases is cumbersome!
- RQL provides powerful path expressions permitting filtering and navigation on both portal schemas and resource descriptions
- E.g., to find (under the Regional ODP hierarchy) URI’s of Hotels in Paris whose title matches “Orsay”

```sql
select Z
from (select $X
    from Regional {: $X}
    where $X I I k e "**Hotel**"
    and $X < Paris) {Y}. {Z}title{T}
where T I I k e "**Orsay**"
```
The ICS-FORTH RDFSuite Architecture

Generic Representation

<table>
<thead>
<tr>
<th>Resources</th>
<th>Triples</th>
</tr>
</thead>
<tbody>
<tr>
<td>id: int</td>
<td>predid: int</td>
</tr>
<tr>
<td>url: text</td>
<td>subid: int</td>
</tr>
<tr>
<td></td>
<td>objid: int</td>
</tr>
<tr>
<td></td>
<td>obivalue: text</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>SunScale</td>
</tr>
<tr>
<td>r1</td>
<td>3</td>
</tr>
</tbody>
</table>
**Specific Representation**

<table>
<thead>
<tr>
<th>Namespace</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>id: int</td>
<td>id: int</td>
</tr>
<tr>
<td>uri: text</td>
<td>superid: int</td>
</tr>
<tr>
<td>1</td>
<td><a href="http://www.w3.org/2000/01/rdf-schema#">http://www.w3.org/2000/01/rdf-schema#</a></td>
</tr>
<tr>
<td>2</td>
<td><a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#">http://www.w3.org/1999/02/22-rdf-syntax-ns#</a></td>
</tr>
<tr>
<td>3</td>
<td><a href="http://www.oclc.org/dublincore.rdfs#">http://www.oclc.org/dublincore.rdfs#</a></td>
</tr>
<tr>
<td>4</td>
<td><a href="http://www.dmoz.org/topics.rdf#">http://www.dmoz.org/topics.rdf#</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>id: int</td>
<td>id: int</td>
</tr>
<tr>
<td>superid: int</td>
<td>domainid: int</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>16</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instances</th>
<th>SubClass</th>
<th>SubProperty</th>
</tr>
</thead>
<tbody>
<tr>
<td>uri: text</td>
<td>id: int</td>
<td>nsid: int</td>
</tr>
<tr>
<td>classid: int</td>
<td>superid: int</td>
<td>domainid: int</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

**DBMS Size vs. Schema Triples**

→ DBMS size scales linearly with the number of schema triples

<table>
<thead>
<tr>
<th>SpecRepr</th>
<th>GenRepr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aver. triple size (with indexes)</td>
<td>0.086 KB (0.1734 KB)</td>
</tr>
<tr>
<td>Aver. triple storage time (with indexes)</td>
<td>0.0021 sec (0.0025 sec)</td>
</tr>
</tbody>
</table>
DBMS Size vs. Data Triples

DBMS size scales linearly with the number of data triples.

<table>
<thead>
<tr>
<th></th>
<th>SpecRepr</th>
<th>GenRepr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aver. triple size</td>
<td>0.123 KB</td>
<td>0.123 KB</td>
</tr>
<tr>
<td>(with indexes)</td>
<td>(0.2566 KB)</td>
<td>(0.2706 KB)</td>
</tr>
<tr>
<td>Aver. triple time</td>
<td>0.0033 sec</td>
<td>0.0039 sec</td>
</tr>
<tr>
<td>(with indexes)</td>
<td>(0.0043) sec</td>
<td>(0.00457 sec)</td>
</tr>
</tbody>
</table>

Query Templates for RDF description bases

**Pure schema queries**

- **Q1**: Find the range (or domain) of a property
- **Q2**: Find the direct subclasses of a class
- **Q3**: Find the transitive subclasses of a class
- **Q4**: Check if a class is a subclass of another class

**Queries on resource descriptions using available schema knowledge**

- **Q5**: Find the direct extent of a class (or property)
- **Q6**: Find the transitive extent of a class (or property)
- **Q7**: Find if a resource is an instance of a class
- **Q8**: Find the resources having a property with a specific (or range of) value(s)
- **Q9**: Find the instances of a class having a given property

**Schema queries for specific resource descriptions**

- **Q10**: Find the properties of a resource and their values
- **Q11**: Find the classes under which a resource is classified
### Execution Time of RDF Benchmark Queries

<table>
<thead>
<tr>
<th>Query</th>
<th>Generic</th>
<th>Specific</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Case 1</td>
<td>Case 2</td>
</tr>
<tr>
<td>Q1</td>
<td>0.0015</td>
<td>0.0017</td>
</tr>
<tr>
<td>Q2</td>
<td>0.0460</td>
<td>0.082</td>
</tr>
<tr>
<td>Q3</td>
<td>0.033</td>
<td>0.0415</td>
</tr>
<tr>
<td>Q4</td>
<td>0.0043</td>
<td>0.008</td>
</tr>
<tr>
<td>Q5</td>
<td>0.0573</td>
<td>0.315</td>
</tr>
<tr>
<td>Q6</td>
<td>0.0034</td>
<td>0.0034</td>
</tr>
<tr>
<td>Q7</td>
<td>124.20</td>
<td>365.73</td>
</tr>
<tr>
<td>Q8</td>
<td>110.58</td>
<td>117.68</td>
</tr>
<tr>
<td>Q9</td>
<td>0.0072</td>
<td>0.0072</td>
</tr>
<tr>
<td>Q10</td>
<td>0.0035</td>
<td>0.0043</td>
</tr>
</tbody>
</table>

### Comparison

- Specific Representation permits the customization of the database representation of RDF metadata.
- Specific Representation outperforms the Generic Representation for all types of queries:
  - Q1, Q2, Q5, Q7, Q10, Q11: by a factor up to 3.73
  - Q3, Q4, Q6: by a factor up to 2.8
  - Q8, Q9: by a factor up to 95,538
- Generic representation pays severe penalty for maintaining large tables (Triples, Resources):
  - e.g., queries Q8, Q9 require (self-) joins of Triples, Resources.
Conclusions and Future Work

- Use DB technology for Web Portals
- RDF/S data model is flexible enough to represent superimposed descriptions of:
  - Information resources
  - E-services
  for various content syndication applications
- RDFSuite addresses the needs of effective and efficient management of RDF descriptions by providing tools for validation, storage and querying
  - First set of benchmark queries for RDF description bases
  - First implementation of an experimental framework for real-scale RDF applications
- Ongoing efforts:
  - appropriate encoding for taxonomic schema relationships to optimize i.e. subclass computation (Q3, Q6)

The ICS-FORTH RDFSuite

The growing number of available information resources and the proliferation of description services in user communities lead nowadays to large volumes of RDF metadata. It becomes evident that browsing such large description bases is a quite cumbersome and time-consuming task. Yet, we want to take benefit from three decades of research in declarative DB and KB languages in order to access voluminous description bases. In this context, ICS-FORTH R&D activities focus on high-level and scalable software tools enabling the realization of the full-potential of the Semantic Web:

- **The Validating RDF Parser (VRP):** The first RDF parser supporting semantic validation of both resource descriptions and schemas.
- **The RDF Schema Specific Database (RSSDB):** The first RDF store that exploits schema knowledge to generate automatically an object-relational representation (SQL3) of RDF metadata and load resource descriptions accordingly.
- **The RDF Query Language (RQL):** The first declarative language for querying RDF resource descriptions and schemes in a uniform manner.
Acknowledgements

- Funding was generously provided by the projects:
  - C-WEB (IST-1999-13479): “A Generic Platform Supporting Community Webs”
  - MESMUSES (IST-2000-26074): “Metaphor for Science Museums”

Summary and Outlook

- RDFSuite addresses the needs of effective RDF metadata management by providing tools for validation, storage and querying
  - validation follows a formal data model and constraints enforcing consistency of RDF schemas
  - incremental loading of voluminous description bases in a persistent store
  - declarative query language for schema and data querying
- Ongoing efforts:
  - RQL query optimization
  - transactional aspects
  - alternative encoding and representation schemes for access optimization