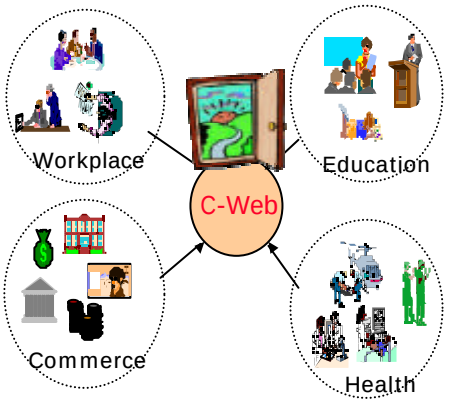
Managing RDF metadata for Community Webs

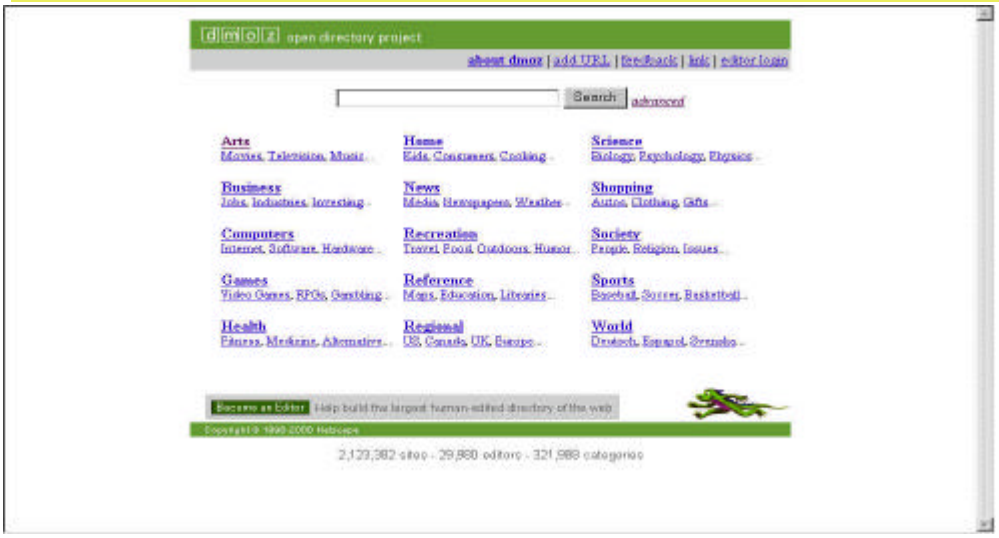
S. Alexaki, V. Christophides, G. Karvounarakis, D. Plexousakis
University of Crete & Institute of Computer Science – FORTH, Greece
Karsten Tolle
University of Frankfurt, Germany
B. Amann, I. Fundulaki, Michel Scholl, A.M. Vercoustre
INRIA Rocquencourt, France

What is a Community Web ?

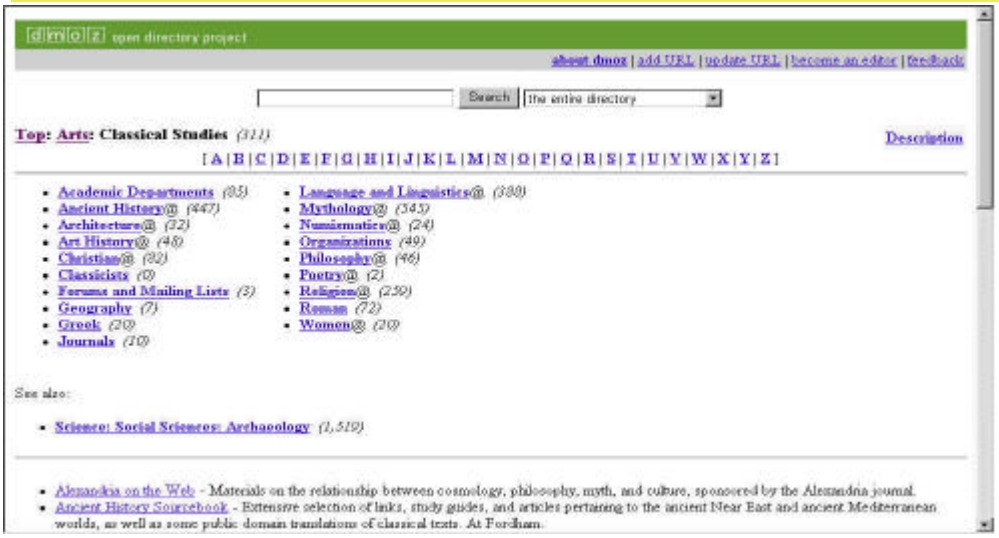
- A group of people sharing a domain of discourse and a set of information resources (e.g., data, documents, services) and having some common interests
  - Commerce, Education, Health
- The main requirement is to provide a single point of useful, ubiquitous comprehensive, and integrated access to various information resources
  - Community Web Portals



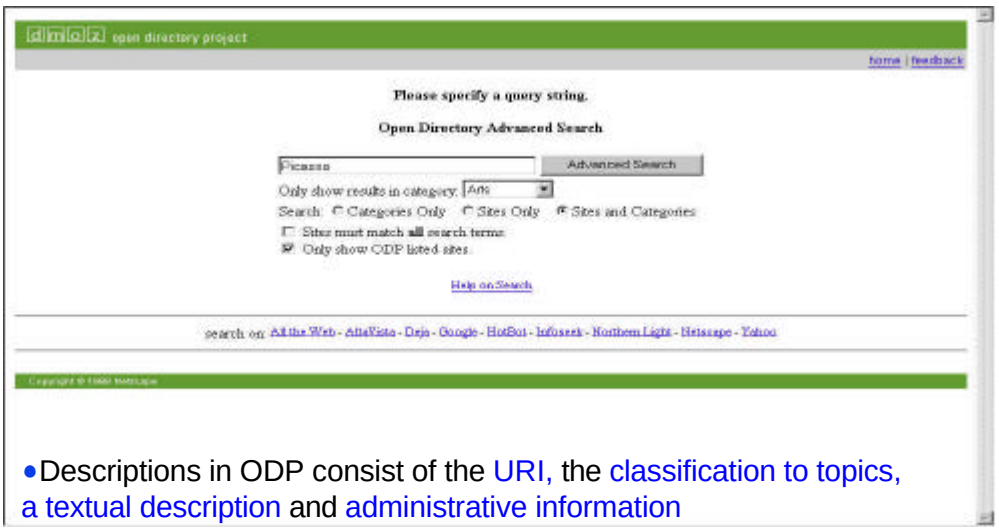
Internet Portals Example: The Open Directory



Browsing the ODP Topics

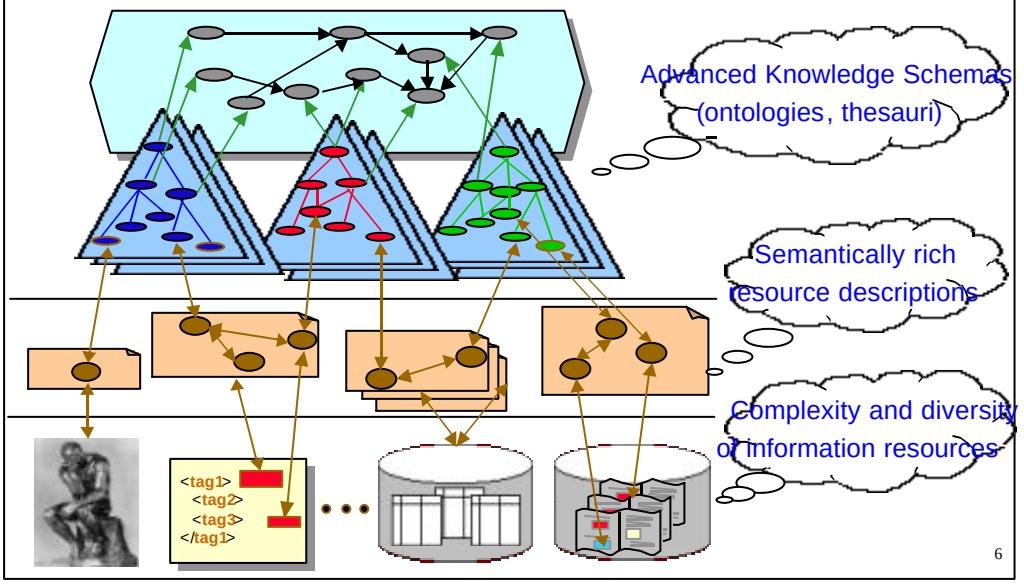


Searching the ODP Topics & URLs

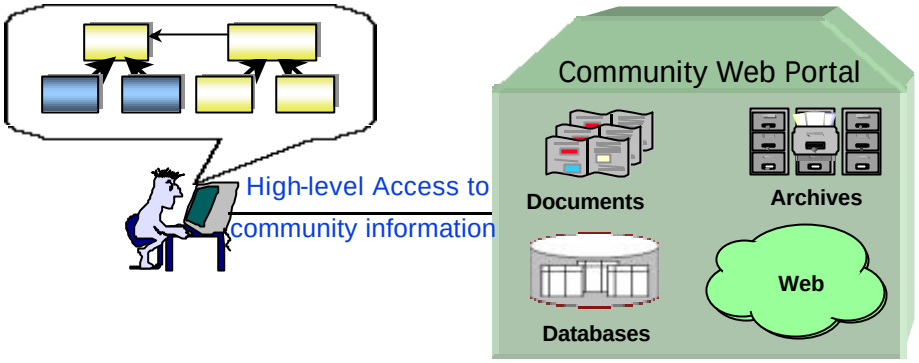


- Descriptions in ODP consist of the URI, the classification to topics, a textual description and administrative information

Corporate Community Web Portals: Main Features



Our Approach



- Use W3C Metadata Standards (RDF/S)
  1. Exploit existing forms of community knowledge
  2. Describe semantically various information resources
- Our Contribution: Declarative Languages for Conceptual Browsing Querying

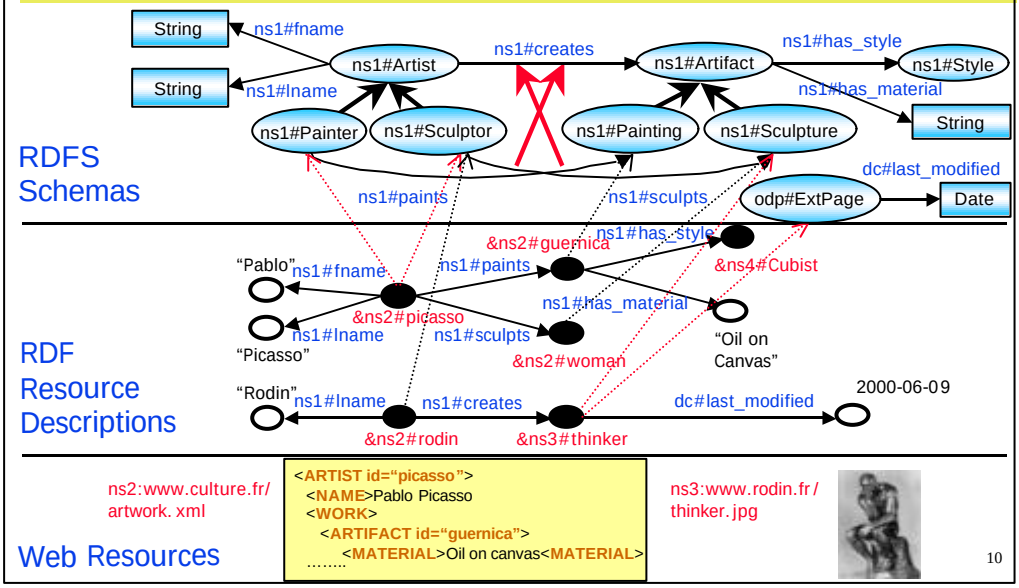
Presentation Outline

- Describing and Querying Community Resources
  - A Data Model for RDF
  - The RDF Query Language (RQL)
- Architecture
  - Metadata Warehouse
  - RQL Interpreter
  - ICS-VRP Validating RDF Parser
- Summary and Future Work

A Data Model for RDF

- Basic Notions:
  - Resource: anything that can be identified by a URI
  - Property: attribute characteristics, roles, relations to resources
  - Class: a container of resources' URIs
  - Statement: assertion consisting of a resource, a named property and a property value
  - Description: a set of statements describing a resource
- Semistructured data model with schema structuring primitives:
  - do not impose a strict typing on data (multiple classification under several classes, optional and multi-valued properties)
  - can be easily extended (specialization of both classes & properties)
  - partial or overlapped data interpretation (several, eventually incomplete schemas for the same data)

A Data Model for RDF: Example



Various RDF/XML Serializations

```
<rdf:RDF xml:lang="en"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
      xmlns="">
....
<Painter rdf:about="http://www.museum.gr/picasso.htm">
    <paints>
      <Painting rdf:about="http://www.museum.gr/guernica.gif">
        <has_style>
          <rdf:description rdf:about="http://www.museum.gr/artstyles.xml#cubism"/>
        </has_style>
        <has_material>oil on canvas</has_material>
      </Painting>
    </paints>
    <sculpts>
      <Sculpture rdf:about="http://www.museum.gr/woman.gif"/>
    </sculpts>
    <fname>Pablo</fname>
    <lname>Picasso</lname>
</Painter>
<Sculptor rdf:about="http://www.museum.gr/picasso.htm"/>
<Sculptor rdf:about="http://www.museum.gr/rodin.htm" lname="Rodin">
    <creates rdf:resource="http://www.museum.gr/thinker.gif"/>
</Sculptor>
<Sculpture rdf:about="http://www.museum.gr/thiker.gif"/>
</rdf:RDF>
```


The RDF Query Language: RQL

- Declarative query language for semantic resource descriptions
  - relies on a formal model (directed, labeled graphs)
  - offers similar functionality to semistructured or XML query languages 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 community schemas & resource descriptions
  - Classes (unary relations)
  - Properties (binary relations)
  - Containers (n-ary relations)

RQL: Examples

- Simple set queries on class and property extents:
  - *Find the resources in the extent of the property creates*

    creates
    - {{ [ns2#picasso, ns2#guernica],
      [ns2#picasso, ns2#woman],
      [ns2#rodin, ns3#thinker] }}

    The creates extent includes paints & sculpts
  - *Find the resources of type painter and sculptor*

    Painter intersect Sculptor
    - {{ ns2#picasso }}

    Resources may be classified under different classes
- Schema constructs used as query terms & support for automatic term expansion (similar to thesauri-based IRS)
  - Useful to query resources with minimal schema knowledge

RQL: Examples

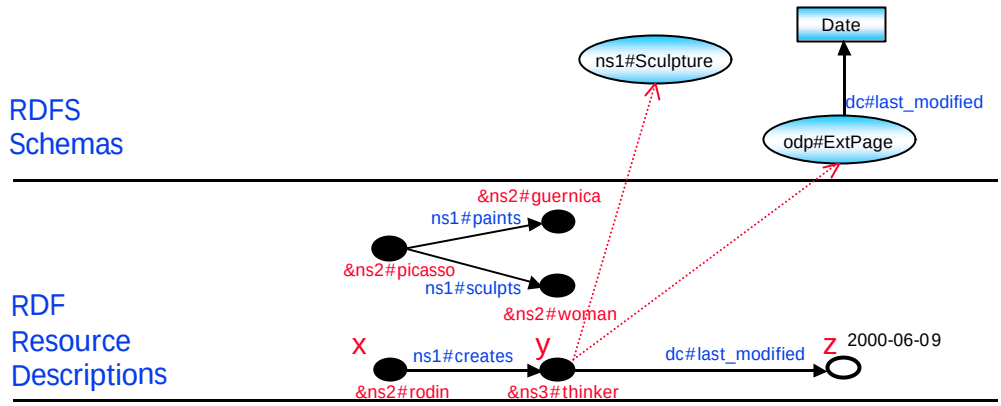
- Navigational queries on semistructured resource descriptions
  - *Find the created resources that have been modified in year 2000.*

```
select y
from {x}creates{y}.last_modified{z}
where z >= 2000-01-01
```

    - {{ns2#thinker}}

    Data paths not foreseen in the schema

RQL: Examples



- Similar functionality to semi-structured or XML query languages (Lorel, UnQL, XQL, XML-QL, XML-GL)
  - Useful in the absence of schema information or when multiple schemas are used to describe resources

RQL: Examples

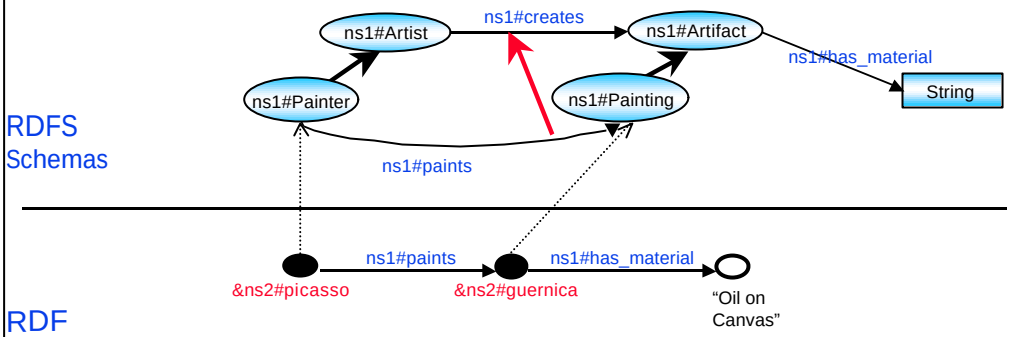
- Filtering resource descriptions with schema knowledge
  - *Find the paintings created by a painter having as material "oil on canvas"*

```
select y
from {:Painter}creates{y:Painting}.has_material{z}
where z = "oil on canvas"
```

Filtering data with schema information

➢ {{ns2#guernica}}

RQL: Examples



- Similar functionality to object query languages (OQL, SQL3)

RQL: Examples

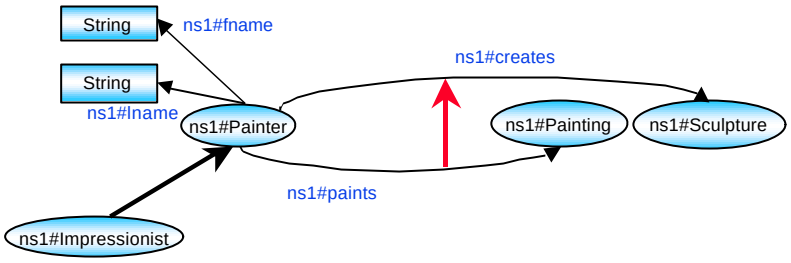
- Pure schema queries
  - *Find the properties and their corresponding range classes that can be used to describe a resource classified between Painter and Impressionist*

```
select $P, $Y
from {$X}$P{$Y}
where $X <= Painter and $X >= Impressionist
```

➢ {{[creates, Artifact],
[creates, Painting],
[creates, Sculpture],
[paints, Painting],
[fname, String],
[lname, String] }}

Schema filtering on class/property hierarchies

RQL: Examples



- Similar functionality to DBMS schema QLs (SchemaSQL, XSQL)
  - Useful for large schemas (integrating ontologies and thesauri)

RQL: Examples

- Schema and data queries
  - *Find all related metadata about the resources of the site www.culture.fr*

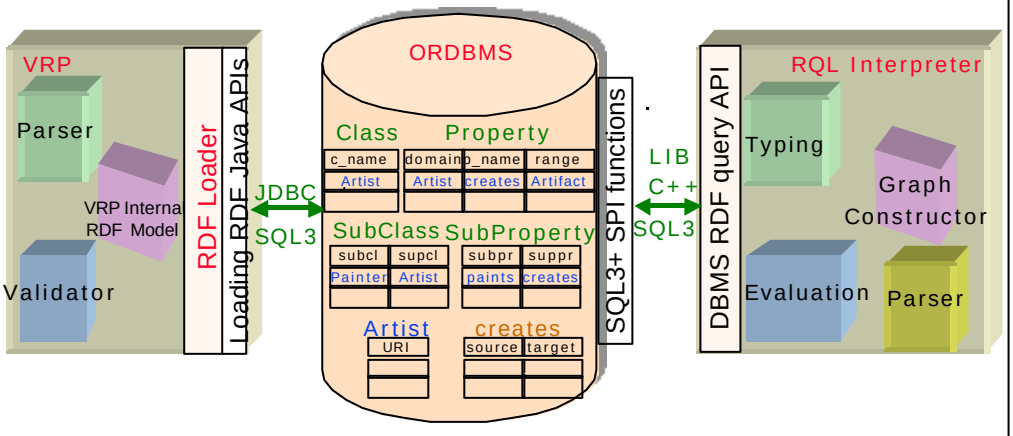
```
select  x,$Y,$P,z,$W
from    {x:$Y}$P{z:$W}
where x like "*www.culture.fr*" or y like "*www.culture.fr*"
```

URLs' pattern matching

➢ {{ [ns2#rodin, Sculptor, creates, ns3#thinker, Sculpture],
[ns2#picasso, Painter, paints, ns2#guernica, Painting],
[ns2#picasso, Sculptor, sculpts, ns2#woman, Sculpture],
[ns2#picasso, Painter, fname, "Pablo", String],
[ns2#picasso, Sculptor, fname, "Pablo", String],
[ns2#picasso, Painter, lname, "Picasso", String],
[ns2#picasso, Sculptor, lname, "Picasso", String],
[ns2#guernica, Painting, has_style, ns4#cubist, Style] ….}}

- Sub-communities may be served by separate metadata servers

The RDF Metadata Warehouse



- Class, Property, subClass, subProperty serve as DB catalog
- Relational storage schemas created automatically at loading time