DartGrid

Vijetha Shivarudraiah
Sai Phalgun Tatavarthy

CSc 8711
Georgia State University
Semantic Web

- Focused on machines
- “a web talking to machines”

The Grid

- Super virtual computer
- Many networked loosely coupled computers
- Work together to perform humongous tasks
Semantic Grid

- Data heterogeneity – semantic web technologies
- Challenge – to design framework to collaborate data from various sources

- Data in big organizations – stored in relational databases - heterogeneous
- support integration of heterogeneous relational databases using semantic web technologies
How Ontology helps?

Without the Semantic Web

Database A
Database B
Database C
Database D
Database E
Database F

24 integration points required!

With the Semantic Web

Database A
Database B
Database C
Database D
Database E
Database F

6 integration points required
Towards a Semantic Web of RD

User Layer

Semantic Web Layer

Relational Databases Layer

Persons or programs only interact with the semantic web layer, and query is defined over the shared ontologies.

Semantic web layer defines the standard vocabularies, formal model, and semantic relationships between databases. Additional deductive capabilities are enabled at this layer.

Relational databases are mapped to the semantic web layer, and new databases should be added at anytime by anyone, and should be independent of the semantic web layer.
DartGrid

- implementation of Semantic Grid
- application development framework
- together with a set of practical semantic tools
- to facilitate integration of heterogeneous relational databases
- leverages upon technologies from both the Semantic web and the Grid
System Architecture

- Domain Expert
- Third-Party Programmer: Writes programs using SPARQL or Web Service
- Publish and Manage Ontology
- Semantic Portal: Searches, navigates, queries
- Ordinary User
- DB Provider: Defines D2R Mappings
- Semantic Registry Service
- Query & Search Service
- Ontology Service
- ......
Key Components in DartGrid

- **Ontology Service**
  - expose the shared ontologies

- **Semantic Registration Service**
  - maintains the semantic mapping information

- **Semantic Query Service**
  - to process SPARQL semantic queries

- **Search Service**
  - supports full-text search in all databases
Semantic Tools

- **DartMapping**
  - Visualized mapping tool
  - heterogeneous relational schemas to RDF/OWL ontologies.

- **DartQuery**
  - Ontology based query interface
  - SPARQL semantic queries rewrite SQL queries

- **DartSearch**
  - ontology-based search engine
  - to make full-text search over all databases
Semantic Mapping

- two legacy relational databases
  - W3C and ZJU (Zhejiang University)
  - about their employees and projects
- integrate them by the FOAF ontology

Target Scheme: foaf Ontology

- Person
- Project
- Organization
Source Relational Schemes

W3C source: w3c:emp(?en,?em,?pn,?ph,?fon)

ZJU source: zju: emp(?en,?em).
           zju: emp_pro(?en,?pn)
           zju: pro_org(?pn,?fon)
           zju: org(?fon,?foh)

- emp (empName, empMail, projectName, projectHomePage, fundingOrganization)
- emp (empName, empMail)
- emp_org (empName, projectName)
- pro_org (projectName, fundingOrganization)
- org (fundingOrganization, fundingOrgHomePage)
RDF Views

- Define each relational table in the source as a view over the RDF ontologies.
- Such views are called RDF Views – 2 parts
  - left part -- view head -- a relational predicate.
  - right part -- view body -- a set of RDF triples

W3C Source:
  (?y1, rdf:type, foaf:Person),
  (?y1, foaf:name, ?en),
  (?y1, foaf: mbox, ?em),
  (?y1, foaf: currentProject, ?y2),
  (?y2, rdf:type, foaf:Project),
  (?y2, foaf: name, ?pn),
  (?y2, foaf: homepage, ?ph),
  (?y2, foaf: fundedBy, ?y3),
  (?y3, rdf:type, foaf: Organization),
  (?y3, foaf: name, ?fon).
Mapping Example

Relational Tuple:
-------------------

w3c:emp("DanBrickley","danbri@w3.org", "SWAD","http://swad.org","EU");

Yielded RDF triples by Applying V1:
-----------------------------------

_:bn1 rdf:type foaf:Person;
foaf:name "Dan Brickley";
foaf:mbox "danbri@w3.org";
foaf:currentProject _:bn2.
_:bn2 rdf:type foaf:Project;
foaf:name "SWAD";
foaf:homepage "http://swad.org";
foaf:fundedBy _:bn3.
_:bn3 rdf:type foaf:Organization;
foaf:name "EU".

DartMapping

- *DBRes panel displays the relational schemas*
- *OntoSchem panel displays the shared ontology*
- *Mapping Panel visually displays the mappings from relational schemas to ontologies*

- Generates registration entry – submit to Semantic Registration Service
- *Outline panel to browse and query mapping information*
1. Display relational tables.
2. Display ontologies.
3. User drags tables and classes into this panel, and establishes their mappings. One table is likely to be mapped to more than one class.
4. Metadata information about the selected table.
5. Outline of the mapping definitions. User could query mappings defined before.
DartQuery

- Ontology based **semantic query user interface**
- Browse ontology tree – select classes
- Query form of properties of the classes generated
- User can select the properties and input constraints

- Semantic query generated – submitted to Semantic Query Service
- Query rewritten into set of SQL queries – using mapping views – Semantic Registration Service
1. User selects one class from this ontology tree.

2. A query interface is automatically generated according to the property definitions of the selected class. User could select properties of interest, and inputs query constraints such as the name of the disease.

3. A outline of currently built query is displayed.

4. User could further explore into the classes related to the current one, and construct complex semantic queries spanning over several classes.

5. User will be led into the query interface of related classes, and could add more query constraints,
Results Interface

- User can navigate through all the related entries by following the semantic links.
- The relations between search results and those discovered by the semantic links are derived from the semantic layer.
Results Interface

1. User selects one data entry which will be highlighted.

2. By following these links, user could get all those data objects semantically related to the current one.

3. Note: the relations between the current object and those "discovered" by following the semantic link are derived through the semantic web layer.

4. User could keep navigating through an unending set of databases as long as they are semantically connected.
DartSearch

- Google-like search interface
- accepts one or more keywords
- makes a complete full-text search in all databases

- Navigate the search results by following the semantic links – as in Query Interface
- Ranking – based on relevance of keywords
- Links lead to the semantic query interface – get more accurate results
1. 用户输入一个关键词，并触发全文检索，覆盖所有数据库。

2. 类似于查询的，通过这些链接，用户可以获取与当前查询相关的所有数据对象。

3. 用户可以永久浏览和导航一个无限的数据库集，只要它们是语义上相关的。

4. 搜索系统可以生成一个具有概念排名的建议列表，根据关键词的相关性进行排名。

5. 用户可以进一步探索查询界面中的这些概念，并指定一个更准确和合适的查询来获取更多信息。
About DartGrid

- developed by Zhejiang University of China
- toolkit was first introduced in 2004
- used to build VO for Traditional Chinese Medicine (TCM)
- 70 legacy TCM databases by a formal TCM ontology with over 70 classes and 800 properties
Pros

- It greatly facilitate developers to interconnect distributed located legacy databases using richer semantics,
- To provide ontology-based query, search and navigation services as one huge distributed database,
- To add additional deductive capabilities on the top to increase the usability and reusability of data
Unsolved Issues on Mapping

1) Redundancy among different database schemas,
2) Inconsistency between two database schemas,
3) Alternative ways to map n-ary (n>2) relation into RDF/OWL model.
Thank You!