DartGrid

Sai Phalgun Tatavarthy
Vijetha Shivarudraiah

Course Name: Databases and the Web
Course Number: CSc 8711
Instructor: Dr. Raj Sunderraman
Term: Spring 2011

Department of Computer Science
Georgia State University
Atlanta, Georgia

Contents

1 Introduction

1.1 Semantic Web

The goal of the semantic web is to be “a web talking to machines”, i.e. in which machines can provide a better help to people because they can take advantage of the content of the Web. The information on the web should thus be expressed in a meaningful way accessible to computers. This definition is easily related to what already exists on the web: wrappers for extracting data from regularly structured pages, natural language analysis for extracting web page contents, indexing schemes, syndication facilities for broadcasting identified web resources. Much of this is painful and fragile: the semantic web should make it smart and robust.

1.2 The Grid

Grids are a form of distributed computing whereby a “super virtual computer” is composed of many networked loosely coupled computers acting together to perform very large tasks. “The Grid” is a vision of “…flexible, secure, coordinated resource-sharing among dynamic collections of individuals, institutions, and resources—what we refer to as *virtual organisations*”. The resources that are shared on the grid largely include data. One major challenge of the grid is to be able to store and process the huge volumes of diversity of content efficiently. The grid should be able to combine the content from multiple sources in unpredictable ways depending on the users’ needs. The users’ should also be able to discover, transparently access and process relevant content wherever it is located on the grid

2 Semantic Grid

The *Semantic Grid* integrates the work on grid architecture from grid community and the work on web semantics from semantic web area, aiming to provide an interconnection environment that can effectively organize, share, cluster, fuse, and manage globally distributed versatile resources based on the interconnection semantics. The use of semantics makes it easier to deal with the data heterogeneity in the grid. A major challenge facing semantic grid is to develop a framework which can collaborate the data from various sources. Integrating relational databases is recently acknowledged as an important vision in this regard; however there are not many well implemented tools and not many applications that are in large-scale real use either. In order to realize this vision, the Semantic web should be able to:

(i) interconnect distributed located legacy databases using richer semantics,

(ii) provide ontology-based query, search and navigation as one huge distributed database, and

(iii) add additional deductive capabilities on the top to increase the usability and reusability of data.

Figure 1 illustrates the basic idea of Semantic-based data integration. With this approach, the users and applications now only need to interact with the semantic layer. The semantic interconnections allow for searching, querying, and navigating around an extensible set of databases without the awareness of boundaries.

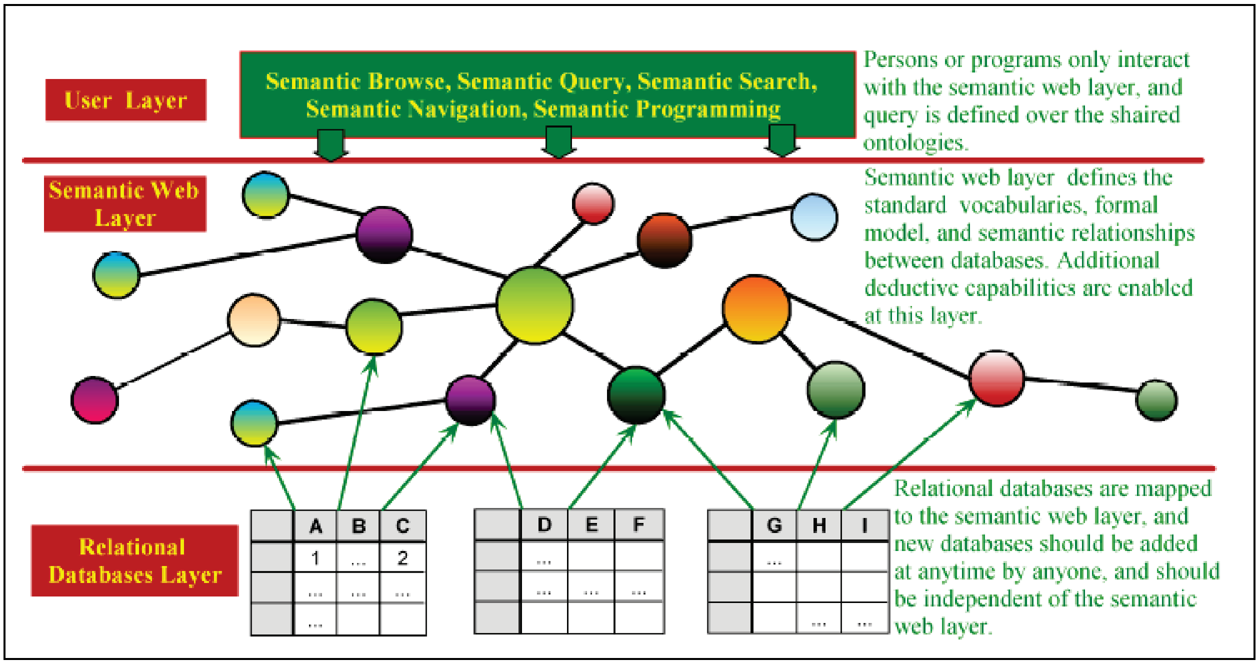

Figure 1. Towards a semantic web of relational databases

3 DartGrid

DartGrid is a semantic grid toolkit for data integration using technologies from both semantic web and grid. It is an application development framework coupled with a set of semantic tools to facilitate the integration of heterogeneous relational databases using semantic web technologies.

3.1 System Architecture of DartGrid

As depicted in Figure 4, there are four key components in the core of DartGrid.

1. **Ontology Service** is used to expose the shared ontologies that are defined using web ontology languages. Typically, the ontology is specified by a domain expert who is also in charge of the publishing, revision, extension of the ontology.
2. **Semantic Registration Service** maintains the semantic mapping information. Typically, database providers define the mappings from relational schema to domain ontology, and submit the registration entry to this service.

3. **Semantic Query Service** is used to process SPARQL semantic queries. Firstly, it gets mapping information from semantic registration service. Afterward, it translates the semantic queries into a set of SQL queries and dispatches them into specific databases. Finally, the results of SQL queries will be merged and transformed back to semantically-enriched format.
4. **Search Service** supports full-text search in all databases. The search results will be statistically calculated to yield a *concepts ranking*, which help user to get more appropriate and accurate results.

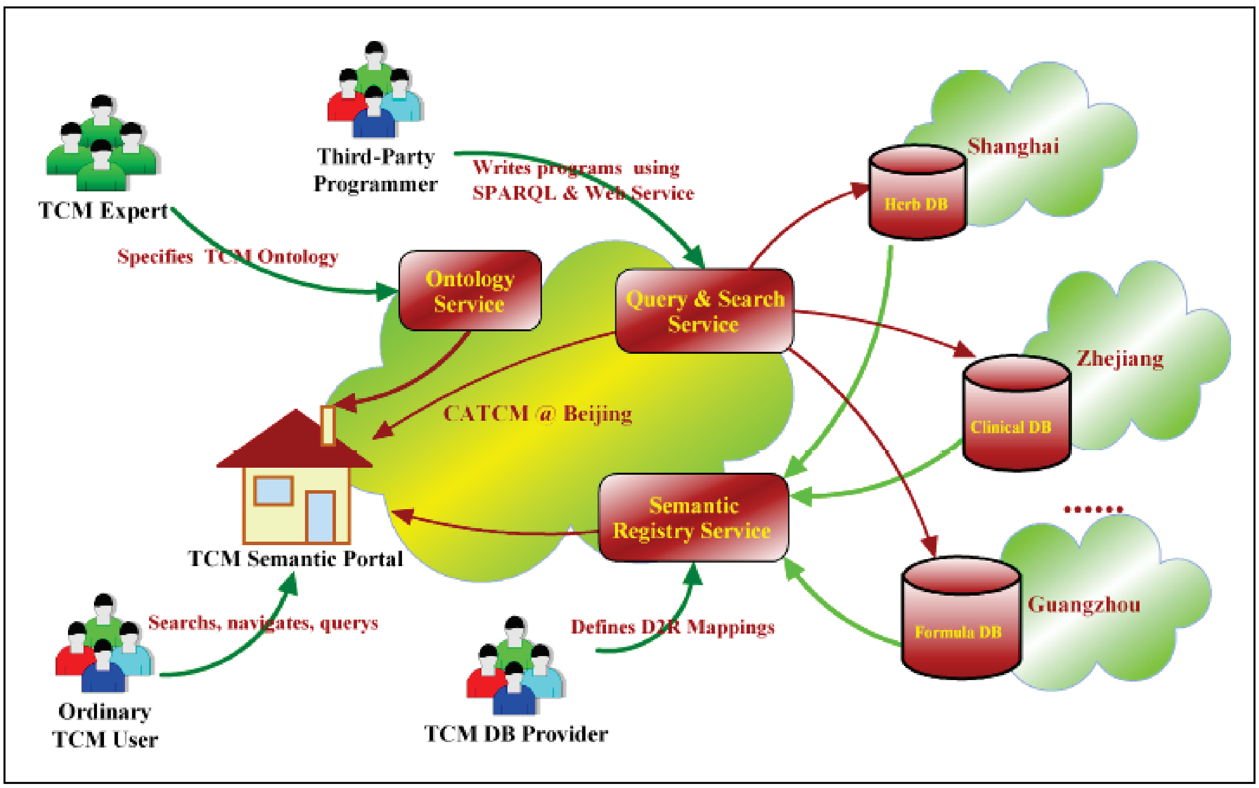

Figure 2. DartGrid System Architecture

DartGrid also includes tools to implement these services, like

- DartMapping - A visual mapping tool to help data provider to define semantic mapping from relational schema to shared ontology.
- DartQuery - A visual query tool which can automatically generate query interface based on ontology definitions, to specify complex semantic queries.
- DartSearch - A search-engine-like interface to enable quick search and to semantically navigate through data from one database to another database.

4 Semantic Mapping

DartGrid takes a view-based approach to define the semantic mapping from source relational schema to mediated RDF ontologies, and offer a visual mapping tool to help define mappings.

4.1 Semantic Mapping and RDF Views

Consider a simple example: suppose both *W3C* and *Zhejiang University* (abbreviated as ZJU) have a legacy relational database about their employees and projects, and we would like to integrate them by the FOAF ontology, so that we can query these relational databases by formulating RDF queries upon the FOAF ontology. The mapping scenario in Figure 3 illustrates two source relational schemas (W3C, and ZJU), a target RDF schema (a part of the foaf ontology), and two mappings between them. Graphically, the mappings are described by the arrows that go between the mapped schema elements.

The lower part of Figure 3 illustrates how to represent the semantic mappings as RDF views. A typical RDF view consists of two parts. The left part is called the view head, and is often a relational predicate. The right part is called the view body, and is often a set of RDF triples. In general, the body can be viewed as a RDF query over the target RDF ontology, and it defines the semantics of the relational predicate from the perspective of the RDF ontology.

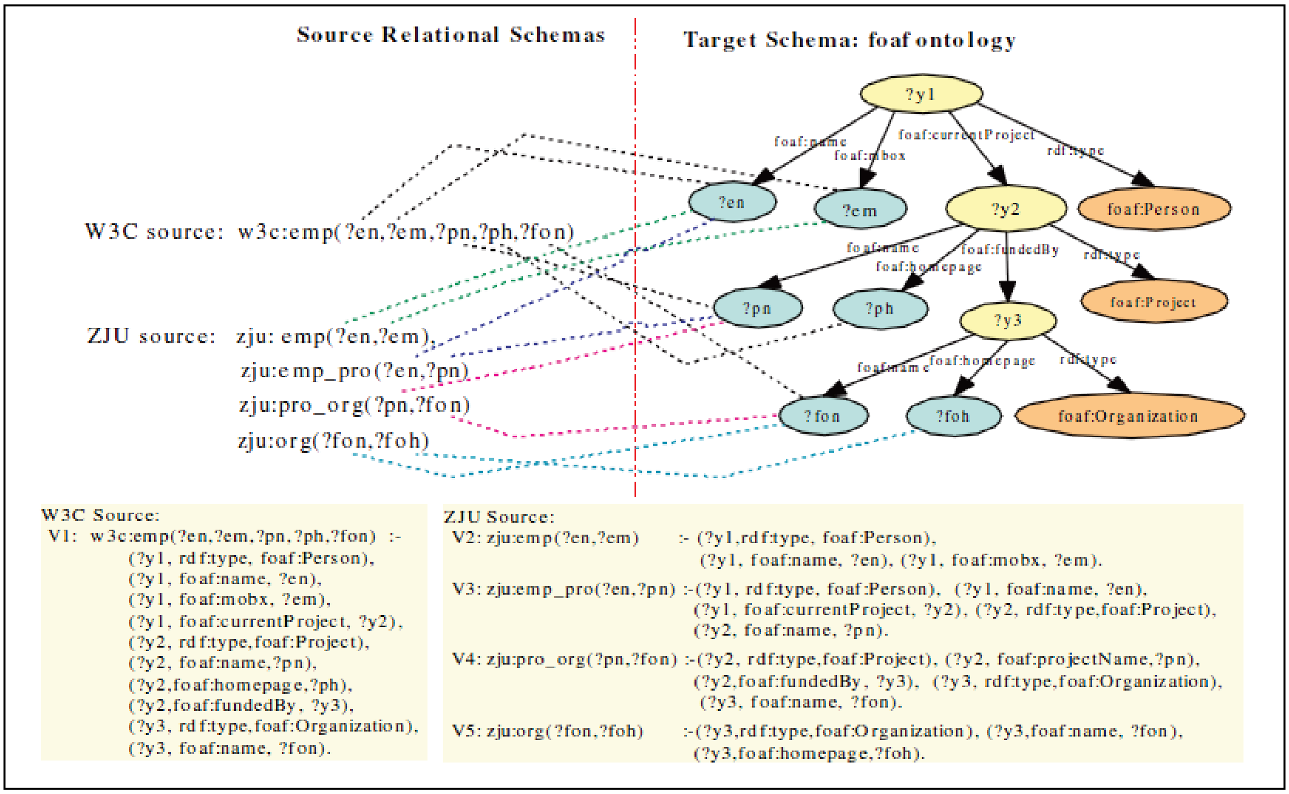

Figure 3. Semantic Mapping Example

An instance of the Target RDF based on semantic mappings using RDF views is shown in figure 4. It illustrates the resulting RDF triples got by applying V1 (see figure 3) on a given relational tuple. The key notion is the newly generated blank node ID in the RDF triples. As can be seen, corresponding to each existential variable $?y$ in the view, we generate a new blank node ID. For examples, *:bn1, :bn2* are both newly generated blank node IDs corresponding to the variables $?y1$, $?y2$ in $V1$. This treatment of existential variable is in accordance with the RDF semantics, since blank nodes can be viewed as existential variables.

```
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".
```

Figure 4. Target RDF Instance based on semantic mapping using RDF views

4.2 DartMapping: Visual Mapping Tool

To speed up the process of defining RDF views, a *Visual Semantic Mapping* tool is developed, shown in figure 5. It has five panels. The *DBRes* panel displays the relational schemas, and the *OntoSchem* panel displays the shared ontology. The *Mapping Panel* visually displays the mappings from relational schemas to ontologies. Typically, user drag tables or columns from DBRes panel, and drag classes or properties from OntoSchem panel, then drop them into the mapping panel to establish the mappings. By simple drag-and-drop operations, users could easily specify which classes should be mapped into a table and which property should be mapped into a table column. After these operations, the tool automatically generates a registration entry, which is submitted to the semantic registration service. Besides, user could use the *Outline* panel to browse and query previously defined mapping information, and use the *Properties* panel to specify some global information, such as namespace, or view the meta-information about the table.

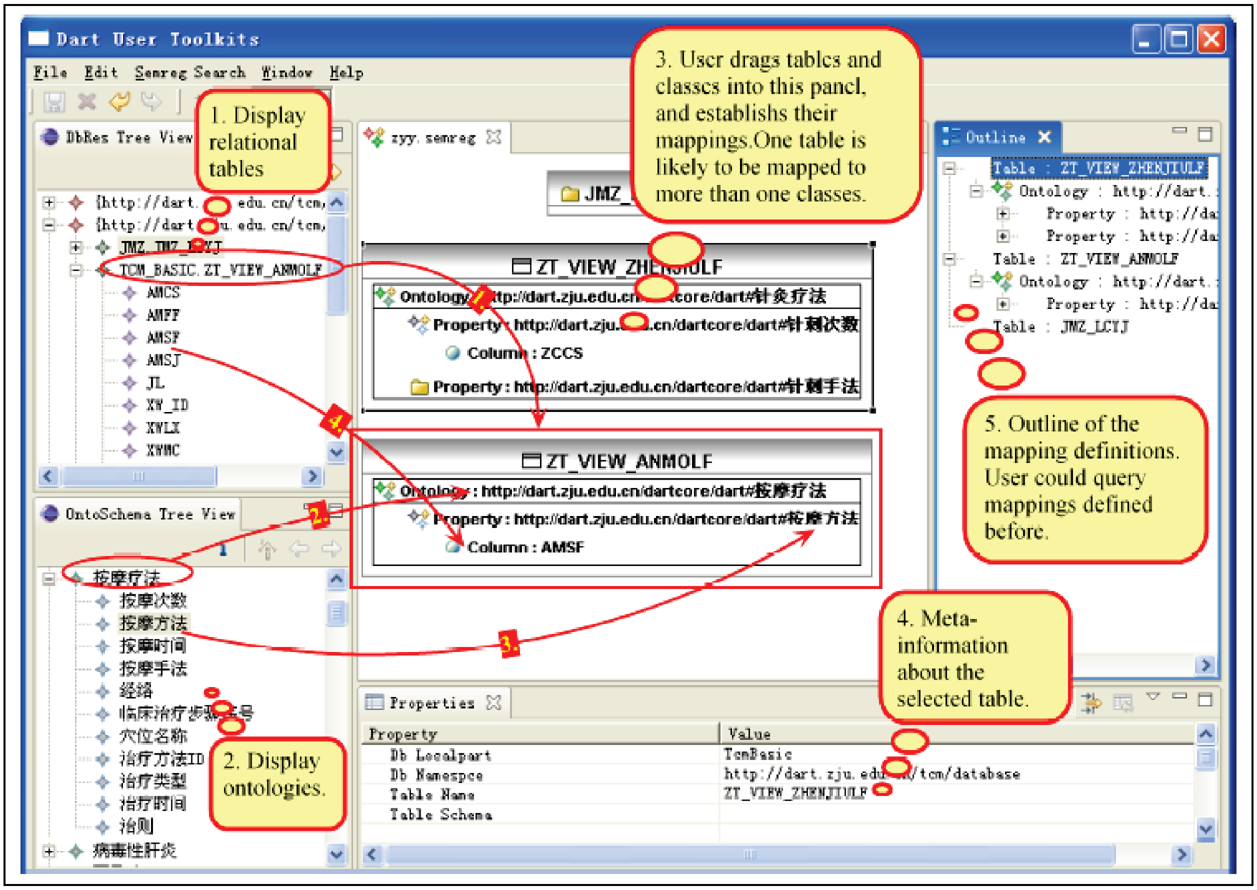

Figure 5. Semantic Mapping Tool

DartGrid also offers two different kinds of user interface to support query and search services.

5 DartQuery: Ontology-based Semantic Query Interface

This form-like query interface is intended to facilitate users in constructing semantic queries such as SPARQL. The query form is automatically generated according to RDF class definitions. This design provides the extensibility of the whole system i.e. when ontology is extended or updated, the interface could dynamically adapt to the updated shared ontology. Figure 6 shows the situation of how a user constructs a semantic query. Starting from the ontology view panel on the left, user can browse the ontology tree and select the classes of interest. A query form corresponding to the property definitions of the selected class will be automatically generated and displayed in the middle. Then user can check and select the properties of interests or input query constraints into the text boxes. Accordingly, a semantic query is constructed and will be submit to the semantic query service, where the query will be rewritten into a set of SQL queries using mapping views contained in the semantic registration service.

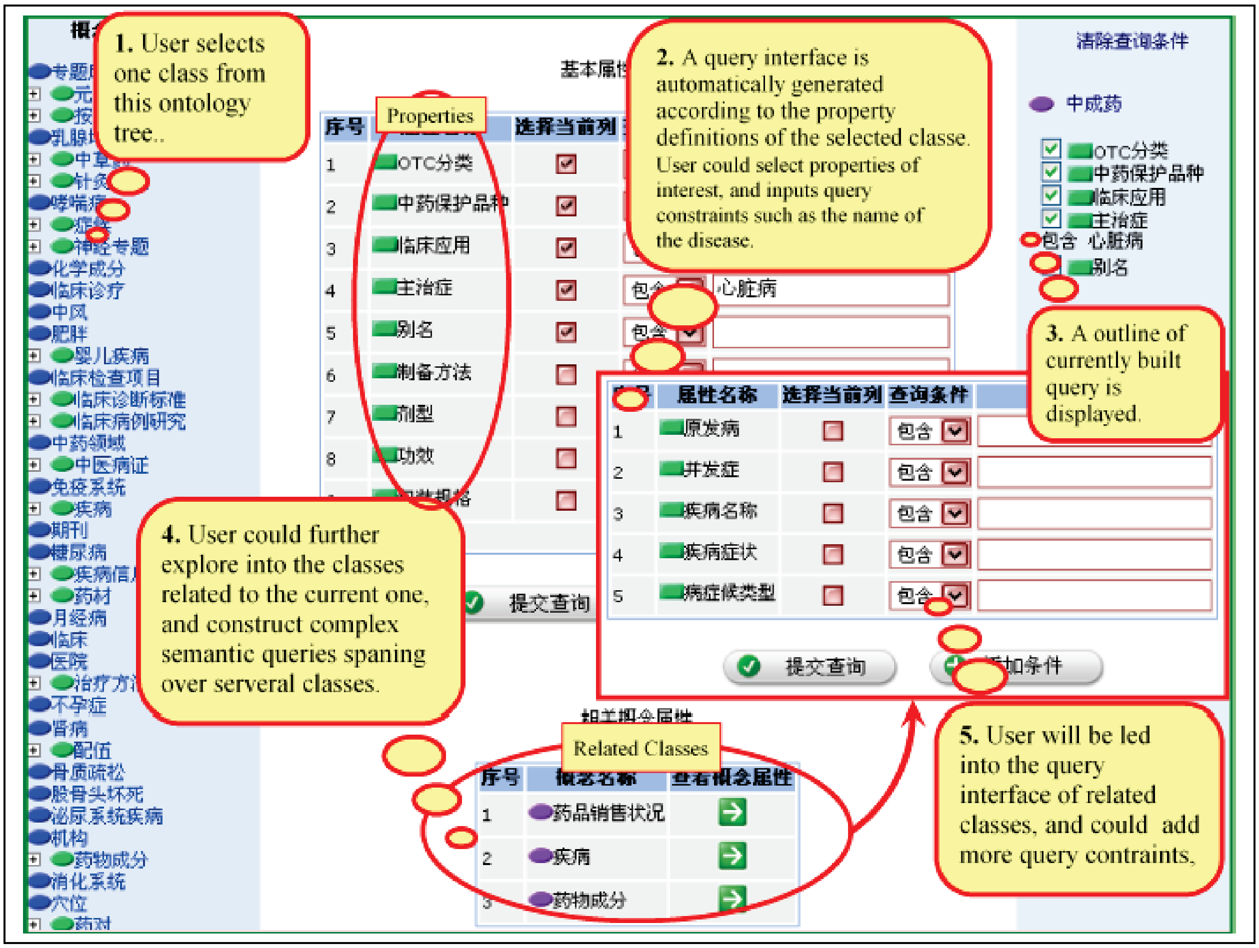

Figure 6. Dynamic Semantic Query Portal

Figure 7 shows the situation in which a user is navigating the query results. When a keyword is submitted, all of the relevant database entries are retrieved and displayed as semantically enriched format, i.e., RDF format. For each entry, the RDF classes that the data entry belongs to are listed below the data. More importantly, all of the names of those data entries relating to the current data entry are also listed as hyper links, so that user could navigate into them to view all of the related information. Because the relationship is established at a semantic level, we call those links as semantic link and the navigation as semantic navigation.

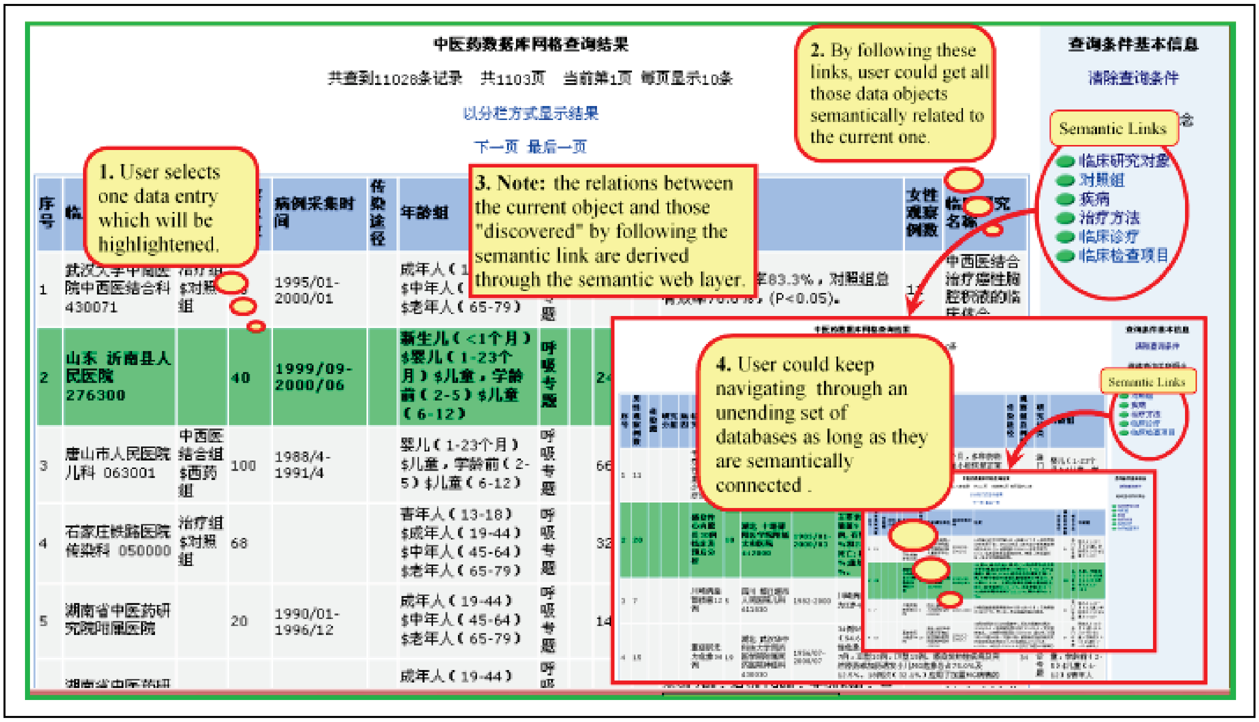

Figure 7. Semantic Navigation through the query results

6 DartSearch: Ontology-based Search Interface with Concepts Ranking and Semantic Navigation

Unlike the semantic query interface, this Google-like search interface accepts one or more keywords and makes a complete full-text search in all databases. Figure 8 shows the situation where a user performs some search operations. Starting from inputting a keyword, the user can retrieve all of those data entries containing one or more hits of that keyword. Being similar to the case of the query interface, user could also semantically navigate the search results by following the semantic links listed with each entry. Meanwhile, the search system generates a list of suggested concepts which are displayed on the right part of the portal. They are ranked based on their relevance to the keywords. These concept links will lead the users to the semantic query interface. Thereafter, users could specify a semantic query to get more accurate and appropriate information.

Figure 8. Search Portal with Concept Ranking and Semantic Navigation

7 Applications

DartGrid has been used to develop a semantic web application for China Academy of Traditional Chinese Medicine (CATCM). It semantically interconnects over 70 legacy TCM databases by a formal TCM ontology with over 70 classes and 800 properties. In this application, the TCM ontology acts as a separate semantic layer to fill up the gaps among legacy databases with heterogeneous structures. Users and machines only need to interact with the semantic layer, and the semantic interconnections allow them to start in one database, and then move around an extendable set of databases.

Other applications where the DartGrid can be used are:-

1. E-learning – Semantic Grid for E-learning based on DartGrid can prove to be a useful and extensible infrastructure for E-learning. RDF semantics can be used for e-learning resource sharing.
2. Data Mining – Data Mining is the computer-assisted process of digging through and analyzing enormous sets of data and then extracting the meaning of the data. Employing DartGrid in data mining applications not only makes analysis of data across heterogeneous data sources easier, it also makes it scalable.
3. Intelligent Transport System (ITS) – ITS plays an increasingly important role in modern transportation systems. It is typically abundant in all kinds of information sources such as sensors, monitoring video cameras which need to be collaborated to increase traffic safety. DartGrid can be a useful part of the distributed infrastructure in support of information resource management and coordinated resource sharing.

8 Issues in DartGrid

The authors state that there are still unsolved issues on mapping relational database schema into RDF/OWL semantic and lists three of them.

1) Redundancy among different database schemas,

2) Inconsistence between two database schemas,

3) Alternative ways to map n-ary (n>2) relation into RDF/OWL model.

9 References

1. Dartgrid: a Semantic Web Toolkit for Integrating Heterogeneous Relational Databases. Zhaohui Wu, Huajun Chen, Heng Wang, Yimin Wang, Yuxin Mao, Jinmin Tang, and Cunyin Zhou. College of Computer Science, Zhejiang University.
2. Towards a Semantic Web of Relational Databases: a Practical Semantic Toolkit and an In-Use Case from Traditional Chinese Medicine. Huajun Chen, Yimin Wang, Heng Wang, Yuxin Mao, Jinmin Tang, Cunyin Zhou, Ainin Yin, and Zhaohui Wu. College of Computer Science, Zhejiang University.
3. DartGrid III: A Semantic Grid Toolkit for Data Integration. Huajun Chen, Zhaohui Wu. College of Computer Science, Zhejiang University.
4. Research challenges and perspectives of the Semantic Web. Report of the EU-NSF strategic workshop.