Integrating Data into an OWL Knowledge Base via the DBOM Protégé Plug-in

Olivier Curé, Raphaël Squelbut
Université de Marne-la-Vallée, France

9th International Protégé Conference
July 23-26, 2006
Stanford University, USA
Main idea of this presentation

- Two facts
  - The Semantic Web needs ontologies.
  - Databases are everywhere

- Our approach
  - map databases to knowledge bases
  - provide a GUI (integrated into Protégé) to ease the creation of mapping files.
Motivating example

• Implementation of a system that helps patients to self-medicate safely.
• This application requires inferences on drugs and symptoms (contraindications, side-effects, posology, etc.).
• The system exploits the main DL reasoning tasks: ontology consistency, concept subsumption, concept satisfiability and instance checking.
Architecture of the self-medication application

- End-user authentication
- Anamneses
  - Patient Q & A
- Inference Engine
  - Provide drug list
- Consult and maintain SEHR
  - XIMSA ontology (OWL)
  - SEHR file (XML)

- Database (RDBMS)
  - Access to description of selected drug

SEHR: Simplified Electronic Health Record
Problem:
- Need to integrate all drugs sold in France (more than 10,000 drugs) with complete information (Summary of Products Characteristics).
- Most French drug databases are incomplete and are usually not available on-demand.
- Many standards need to be integrated: ATC (Anatomical Therapeutic Chemical classification) and DDD (Defined Daily Dose) from the WHO, EphMRA, etc..
DBOM
DataBase Ontology Mapping

• Objective: design, instantiate and maintain a knowledge base (KB) from multiple relational databases (DBs).
  • Design the TBox using the DBs schemas.
  • Instantiate the ABox with the tuples of the data sources w.r.t. the mapping.
  • Maintain the ABox using the mapping (from DBs to KB), a set of automatically created triggers and Java methods.
DBOM (2)

- DBOM is related to the exchange and integration of data.
- The DBOM system is a triple \((S, O, M)\) with
  - \(S\), a set of sources
  - \(O\), an ontology formalized in a Description Logic (DL) that can be as expressive as SHOIN(D), syntactically equivalent to OWL DL.
  - \(M\) the mapping in a language over \(S\) and \(O\).
Main characteristics of DBOM:

- Mapping exploits the GAV (Global As View) approach: the elements of the target are expressed in terms of the sources (opposed to LAV - Local As View).
- Mapping file is serialized in XML.
- The target is materialized (because on-demand querying may not be possible) and is an OWL document.
Characteristics of DBOM (2)

- Main operations of DBOM
  - Instantiation (at mapping processing time)
  - Maintenance (whenever a tuple of a source is updated)
- both operations adopt the possible answer semantics (opposed to certain answers in data integration and data exchange).
DBOM members

- DL Members = DL concepts + DL roles
- In DBOM, we distinguish abstract to concrete members.
- Approach is similar to Object-Oriented Programming:
  - abstract members serve to design a hierarchy and are not instantiated. They are created with the owlClasses and Properties tabs of Protégé.
  - SQL queries are associated to concrete members to enable instantiation from tuples of the sources. They are created with the DBOM Protégé plug-in.
Dealing with inconsistencies

- Because of the adoption of possible answers with multiple sources, inconsistencies can emerge from redundant data.

<table>
<thead>
<tr>
<th>Source DB1</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>cip</td>
<td>name</td>
<td>price</td>
<td>reimb</td>
<td>atc</td>
<td>rate</td>
</tr>
<tr>
<td>3311692</td>
<td>TUXIUM 30 mg</td>
<td>2.32</td>
<td>0</td>
<td>R05DA09</td>
<td>18</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source DB2</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>cip</td>
<td>name</td>
<td>price</td>
<td>reimb</td>
<td>atc</td>
<td></td>
</tr>
<tr>
<td>3311692</td>
<td>TUXIUM 30 mg</td>
<td>3.32</td>
<td>35</td>
<td>R05DA09</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source DB3</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>cip</td>
<td>name</td>
<td>price</td>
<td>reimb</td>
<td>atc</td>
<td></td>
</tr>
<tr>
<td>3626651</td>
<td>DIMETANE Sirop Sans Sucre</td>
<td>2.94</td>
<td>35</td>
<td>R05DA08</td>
<td></td>
</tr>
<tr>
<td>3311692</td>
<td>TUXIUM 30 mg</td>
<td>2.05</td>
<td>65</td>
<td>R05DA08</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source DB4</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>code</td>
<td>name</td>
<td></td>
</tr>
<tr>
<td>R05DA08</td>
<td>Pholcodin</td>
<td></td>
</tr>
<tr>
<td>R05DA09</td>
<td>Dextromethorphan</td>
<td></td>
</tr>
</tbody>
</table>
Confidence values

- The end-user has the ability to set a confidence value (real value in $[0,1]$) for each member's view. Intuitively defines the reliability of the view from the designer's point of view. [Mendelzon et al, Greco et al, De Giacomo et al].
- In cases of several views for a given member, it defines a partial order on the views.
- Mapping example using conjunctive queries:

\[
\text{Drug} \equiv \{(U,V,W,X,Y,Z) \mid \text{DB1.drug}(U,V,W,X,Y,Z)\} \\
\text{conf}=0.8
\]

\[
\text{Drug} \equiv \{(U,V,W,X,Y) \mid \text{DB2.drug}(U,V,W,X,Y)\} \\
\text{conf}=0.6
\]

\[
\text{Drug} \equiv \{(U,V,W,X,Y) \mid \text{DB3.drug}(U,V,W,X,Y)\} \\
\text{conf}=0.5
\]
DBOM Protégé plug-in (1)

- Loading DB sources
- Visualization of the DB schemas
DBOM Protégé plug-in (2)

- Concept definition
- Association of SQL queries to this concept, with confidence values.
Associate a datatype property to each attribute of the SELECT clause.
DBOM Protégé plug-in (4)

- Visualization of all the queries associated to a Concept.
• Same mechanism for roles but we associate DL concepts to attributes of the SELECT clause (domain and range).
DBOM Protégé plug-in (5)

- Visualization of the concrete members
- Process the serialization of the mapping and creation of the ABox
Serialization of the mapping

<?xml version="1.0" encoding="iso-8859-1"?>
<map xmlns:dbom="http://www.univ-mlv.fr/~ocure/dbom/1.0#">
  <namespaces prefix="owl" namespace="http://www.w3.org/2002/07/owl"/>

  <dbConnect dbDriver="org.postgresql.Driver"
             dbNamePrefix="jdbc:postgresql" dbUser="olive"
             dbPw="***"/>

  <dbom:map xmlns:dbom="http://www.univ-mlv.fr/~ocure/dbom/0.1#">
    <dbom:class className="Man">
      <dbom:instanceUnion>
        <dbom:instance dbSrc="parent1" query="SELECT ssn, name FROM person WHERE idgender=1;" confidence="0.65">
          <dbom:id>
            <dbom:field value="1"/>
          </dbom:id>
          <dbom:data>
            <dbom:field value="2" datatypeProperty="hasPersonName"/>
          </dbom:data>
        </dbom:instance>
        ...
      </dbom:instanceUnion>
    </dbom:class>
  </dbom:map>
  ...
</map>
Benefits of the Protégé plug-in approach

• A user-friendly graphical user interface
• Exploits the end-user's Protégé expertise: use OWL tabs to create abstract members and datatype properties, add restrictions to concepts, etc..
• Possibility to enrich an existing ontology with concrete members.
Future works on DBOM

• Integrate a Query By Example (QBE) approach to facilitate the declaration of SQL queries attached to concrete members.
• Exploit the mapping to enable
  • data synchronization: maintain the ABox according to updates on available data sources.
  • schema synchronization: adapt the TBox according to some modifications on the source schemata.
• Considering XML documents as data sources.
• Propose a mapping methodology.
• In cases of data synchronization, infer on the KB to validate updates at the sources.
Inference example

• Scenario: An authorized end-user logs in the database administration web site and records a new drug: D1 with RINN 'dextromethorphan' and therapeutic class 'antidepressive'.
• The tuple is recorded in the database.
• A trigger fires the ABox synchronization.
Inference example (2)

- Searching the KB graph.
- Result: no relationship exists between the RINN and the therapeutic class.
- A new entry is recorded in the maintenance log file. This record contains:
  - the id of the user
  - the tuple that caused the problem
  - a problem description (RINN-therapeutic class problem).
Inference example (3)

• A solution to this problem can either be:
  • A new relation between the RINN and the therapeutic class can be validated by the end-user.
  • The RINN for that drug is false and the system can propose valid RINN for anti-depressive (for example iproniazide)
  • The therapeutic class is false and valid a therapeutic class will be proposed according the RINN (i.e. Antitussive).
  • All information are false.
Summary

- Using existing databases to design ontologies, instantiate and maintain knowledge bases.
- DBOM is application-independent and can be used when databases are available and covering a domain.
Thank you

Questions?