Peer-to-Peer Data Integration

Alan Davoust

28.02.2008
Presentation outline

- What is Data Integration?
- From Conventional to Peer-to-Peer Data Integration
- Some issues with semantics and inconsistencies
- Overview of some P2P Data Integration applications
- UP2P, our local project
Data Integration : example

>> A Data Integration System for movies

Query over Global Schema

IMDB
- Film:
  - Title: ?1
  - Director: Woody Allen
  - Playing: 10 pm
- Cinema:
  - city: Paris
  - name: ?2

Query Engine

Pariscope
- Cinema:
  - Name:
  - Address:
- Film:
  - Title:
  - Hours:

Q(?1, ?2) :- IMDB(?1, "woody allen", - ) \ Pariscope(?2, - , ?1, "10pm")
Data Integration

- Mediator
  - Combines data from different sources
  - Translates to a common data model

- Wrapper
- Data source A
- Data source B
- Data source C
Data Integration: views, semantics

- **GAV view definition**: \( G_1(x) \leftarrow LA_1(x,y) \land LB_2(y) \)
- **LAV**: \( LA_1(x,y) \leftarrow G_1(x) \land G_2(x,y) \)
- **GLAV**: \( G_1(x) \land G_2(x,y) \leftarrow LA_1(x,y) \land LC_1(y) \)

**Logical Semantics of “:-”**
- \( G_1(x) \land G_2(x,y) \leftarrow LA_1(x,y) \land LC_1(y) \) (sound)
- \( G_1(x) \land G_2(x,y) \rightarrow LA_1(x,y) \land LC_1(y) \) (complete)
- \( G_1(x) \land G_2(x,y) \leftrightarrow LA_1(x,y) \land LC_1(y) \) (exact)
Towards P2P architecture

Peer A
- P2P Mediator
- Wrapper
  - Data source A

Peer B
- P2P Mediator
- Wrapper
  - Data source B

Peer C
- P2P Mediator
- Wrapper
  - Data source C
P2P architecture : logical view

Mapping example:

Peer_A.ArticleTitle :- Peer_B.PublicationTitle
Peer_A.Author :- concat(Peer_B.AuthorLastName, Peer_B.AuthorFirstName)
Why get data from other sources?

- Get more answers (Open World Assumption)
- Other sources more authoritative
- Combine information from several sources (relational join)

Imply "redundant" sources

→ Will define the semantics of the mappings
Peer-to-peer networks

• Advantages
  ➢ Decentralized and dynamic
  ➢ Scalable

• Disadvantages
  ➢ No control = no guarantee that sources are reliable
  ➢ Sometimes very long paths to reach other nodes
Some issues in P2P Data Integration

- Complexity / Expressiveness trade-off
  - Language for queries and views/schema determines complexity
  - Query answering is undecidable in general
  - In practice in a distributed setting delays can be high

- Inconsistencies
  - Data inconsistencies
  - Incorrect mappings
  - Most studies assume the data is consistent
Inconsistencies

• How do inconsistencies arise?
  ➢ Negation
  ➢ Primary / foreign keys
  ➢ CWA
  ➢ Explicit constraints

• Resolving Inconsistencies
  ➢ We may trust all peers equally
  ➢ We may have one or several more trusted peers

• Inconsistency tolerance: theoretical studies
  ➢ Calvanese et al. using modal logic
  ➢ Bertossi et al. using answer set programming
Inconsistencies : Calvanese approach

Mappings $Q_i \rightarrow Q_j$ from $P_i$ to $P_j$
$Q_k \rightarrow Q_j$ from $P_k$ to $P_j$

• Case 1: $P_i$ is inconsistent
  ➢ Ignore it entirely

• Case 2: data from $P_i$ inconsistent with data from $P_j$
  ➢ Ignore imported data
  ➢ (implicitly a peer trusts itself more)

• Case 3: data from $P_i$ inconsistent with data from $P_k$
  ➢ Ignore both pieces of imported data
  ➢ (no preference between other peers)
Inconsistencies : Calvanese approach

• Summary :
  ➢ Ignore inconsistent peers
  ➢ Transfer knowledge only if it does not create inconsistencies
  ➢ Implicit trust relation : each peer trusts itself more than the others
  ➢ Compatible with OWA and CWA

• Formalization with multi-modal logic:
  Mapping $Q_i \rightarrow Q_j$ from $P_i$ to $P_j$ produces semantic rule:

$$ (K_i(Q_i) \land \neg A_i(\bot_i) \land \neg A_j(\neg Q_j)) \Rightarrow K_j(Q_j) $$
Inconsistencies: Bertossi approach

- Explicit trust relations between peers
- Closed-world assumption
- Propagation of queries is only for consistency checking

- Peer consistent answers based on the notion of peer solution

- A solution is an DB instance which respects all the constraints for the peers and stays as close as possible to original data
Inconsistencies : Bertossi approach

Computing the solutions for a peer:

- Import all relevant data from peers
- Depending on data exchange constraints:
  - Virtually add, remove tuples from relations to resolve inconsistencies
- Data imported from more trusted peers does not change (even virtually)

→ Peer Consistent Answers to a query are those which hold in every solution.
Inconsistencies : Example

- Schemas and data at the peers

<table>
<thead>
<tr>
<th>Peer</th>
<th>Person</th>
<th>Birthdate</th>
<th>Birthplace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pj</td>
<td>Alice</td>
<td>01/04/75</td>
<td>Paris</td>
</tr>
<tr>
<td></td>
<td>Bob</td>
<td>04/05/80</td>
<td>Orleans</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Peer</th>
<th>Person</th>
<th>PlaceOfBirth</th>
<th>Citizenship</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pi</td>
<td>Alice</td>
<td>Ottawa</td>
<td>French</td>
</tr>
<tr>
<td></td>
<td>Bob</td>
<td>Orleans</td>
<td>French</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Peer</th>
<th>Person</th>
<th>PlaceOfBirth</th>
<th>Citizenship</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pk</td>
<td>Albert</td>
<td>Ottawa</td>
<td>Canadian</td>
</tr>
<tr>
<td></td>
<td>Bob</td>
<td>Orleans</td>
<td>Canadian</td>
</tr>
</tbody>
</table>

Queries to Pj:
1: People born in Ottawa?
2: Citizenship of Bob?
3: Birthplace of Bob?
## A few P2P data integration applications

- Piazza
- Edutella
- SomeWhere / SomeRDFS
- UP2P
PIAZZA (U of Washington, Seattle)

• Peer Data Management System

• XML, Relational DBs

• Recursive rewriting and propagation of queries

• Allows different semantics for mappings

• Nodes may contribute data and / or mappings
PIAZZA: example 2

Worker(SID, first, last)
Ambulance(VID, hosp, GPS, dest)
EMT(SID, hosp, VID, start, end)
Doctor(SID, hosp, loc, start, end)
EmergBed(bed, hosp, room)
CritBed(bed, hosp, room)
GenBed(bed, hosp, room)
Patient(PID, bed, status)

SkilledPerson(PID, skill)
Located(PID, where)
Hours(PID, start, stop)
TreatedVictim(PID, BID, state)
UntreatedVictim(loc, state)
Vehicle(VID, type, cap, GPS, dest)
Bed(BID, loc, class)
Site(GPS, status)

911 Dispatch Center (9DC)

First Hospital (FH)
Lakeview Hospital (LH)

Hospitals (H)

Fire Services (FS)

Portland Fire District (PFD)
Vancouver Fire District (VFD)

Station 3 Station 19 Station 12 Station 32

Engine(VID, cap, status, station, loc, dest)
FirstResponse(VID, station, loc, dest)
Skills(SID, skill)
Firefighter(SID, station, first, last)
Schedule(SID, VID, start, stop)

Earthquake Command Center (ECC)

Medical Aid (MA)
Search & Rescue (SR)

Emergency Workers (EW)

National Guard Washington State

Legend
Peer
Set of Stored Relations
Edutella (Stockholm, Hannover, Kassel)

- Distributed ontology of educational resources
- Super-peer topology
- Single schema (meta-model) for super-peers
- Queries independently executed on each node (including “join” type queries)
- Own query language (an extension of datalog)
Framework for distributed ontologies
Any peer can extend the ontology by declaring relations with classes defined by other peers
DRAGO built roughly on the same idea
U-P2P (Carleton U :))

- Our local project!!
- Share metadata and files

Data Integration

Integration via P2P mappings across schemas

P2P file sharing apps

P2P file sharing with searchable metadata

Each client supports several schemas

P2P mappings within peer connect schemas
U-P2P

- Multiple schemas (Communities)
- Each community is a (P2P) distributed database
U-P2P

- Query: \( \{ \text{doc} \mid \text{Science}\_\text{papers}.\text{Author}(\text{“Einstein”}, \text{doc}) \} ? \)
- Query propagated only within community: no network flooding
U-P2P : Data Integration perspective

- Data Integration is between heterogeneous communities
- Semantic mappings defined as “bridge”
- Queries in one community can be propagated to other communities
- Meta-model allows for mappings to be shared in a specific community
U-P2P : Challenges

- On-going refactoring of application
- Integrate formal semantics for mappings

- ... Any suggestions?
P2P Data Integration : Purpose

• File sharing : get \{ doc | Author(“Einstein”, doc)\}
  - Expected response: Metadata about the doc? The doc itself? (search or retrieve)
  - Usually, each peer can provide answers autonomously
  - Challenge is to **propagate** query and possibly **translate** it

• Process a query over a distributed knowledge base :
  get \{film, theatre | director(“Spielberg”, film) and PlaysIn(film, theatre, “28.02.2008”)\}
  - Expected response: a set of atomic pieces of data (DB entries)
  - A response may result from relational join operation on several peers
  - Challenge is to **rewrite** the query
Issues with Query processing

- Do queries terminate?
  - Most studies assume limited network of peers
  - Issues with topology (cycles)

- Certain answers to a query:
  - True in every possible consistent interpretation
  - Open-World vs. Closed-World Assumption

- Inconsistencies depend on:
  - Redundancy in DBs
  - Integrity constraints
  - Uniqueness of names
P2P data integration

- Metadata is logical predicate about some file or object

- Conjunctive query: get x such that...
  - x can be object (file, URL)
  - x can be atom (logical atom)

- We have the logical view where we process logical queries in a distributed knowledge base