Data Integration

Work by Many People

Presented by Alon Levy

What is Data Integration?

The problem of providing

uniform (sources transparent to user)

access to (query, and eventually updates too)

multiple (even 2 is a problem!)

autonomous (not affect the behavior of sources)

heterogeneous (different data models, schemas)

structured (at least semistructured)

data sources (not only databases)

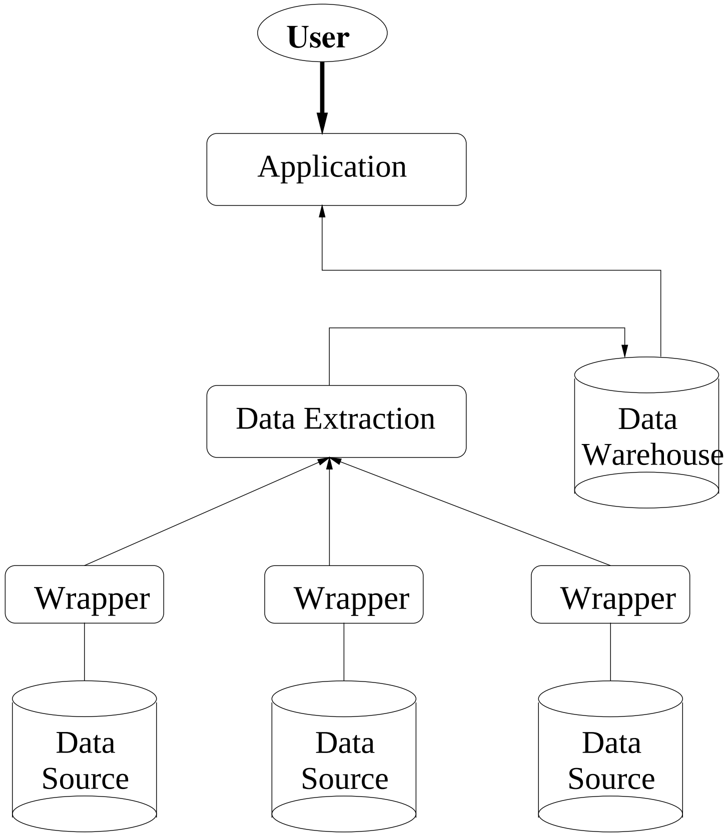
Examples of Data Integration

Provide uniform access to the 17 databases in our company.

Tool for uniformly manipulating biological data produced by different centers and experiments.

Provide a uniform access to all the cultural databases produced by countries in Europe.

Provide a query interface to arbitrary structured sources on the WWW.



Virtual vs. materialized architecture

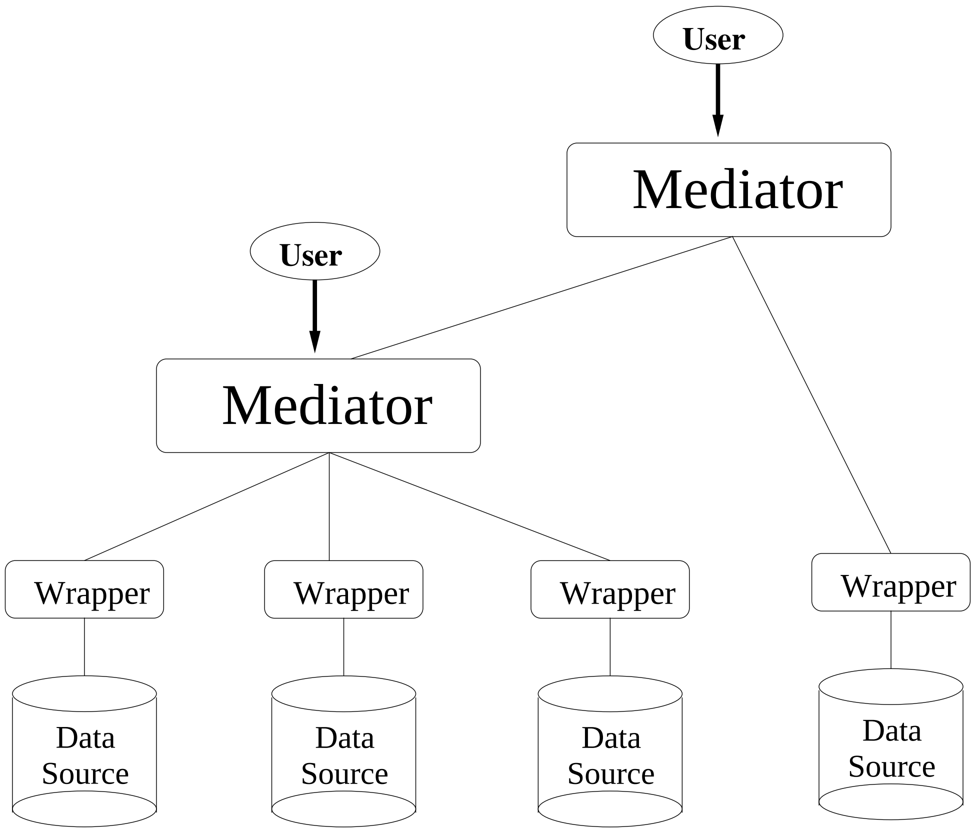
Principle Dimensions of Data Integration II

Access: query only or query&update?

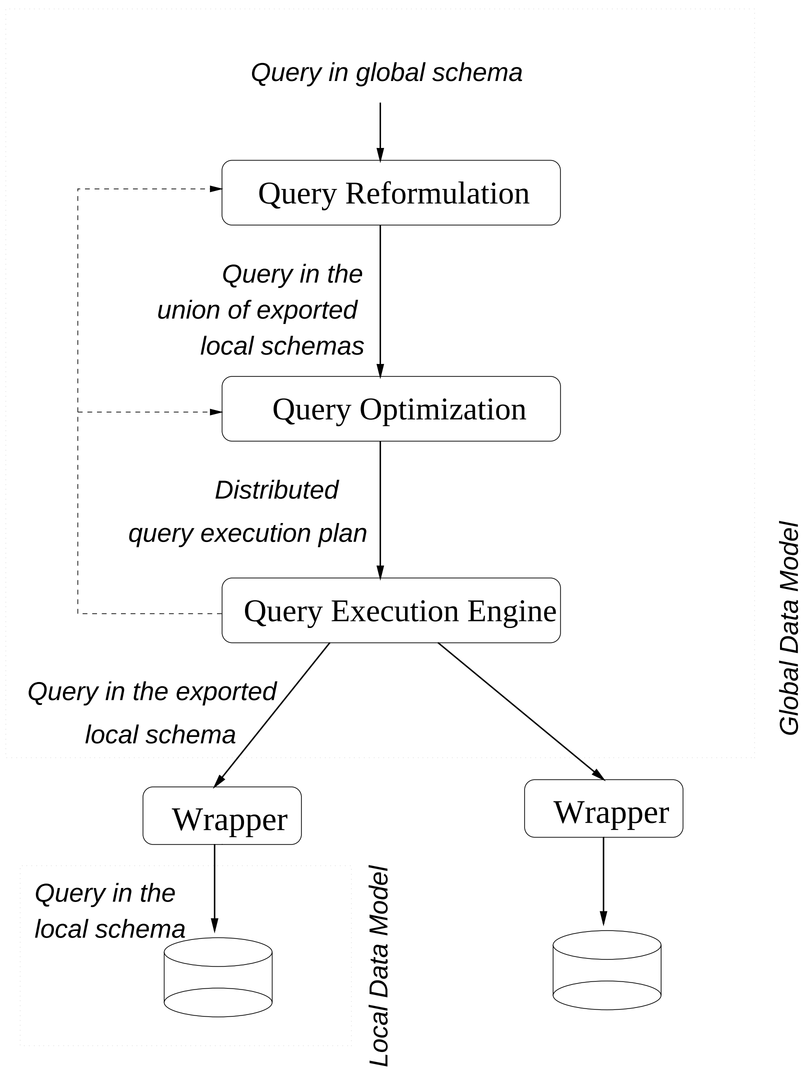
- problem similar to updating through views
- need distributed transactional services.

Mediated schema: yes or no?

- Mediated schema requires schema integration and then query reformulation.
- Without mediated schema, we lose some of the advantages of data integration.



Generic Architecture for Data Integration


Query in global schema
Query Reformulation
Query in the
union of exported
local schemas
Query Optimization
Distributed
query execution plan
Query Execution Engine
Query in the exported
local schema
Global Data Model
Wrapper
Wrapper
Query in the
local schema
Local Data Model

Additional Dimensions of Data Integration

1. How many sources are we accessing?
2. How autonomous are the sources?
3. How much knowledge do we have about sources?
4. How structured are the data in the sources?
5. Requirements from responses:
    - accuracy
    - completeness
    - machine readable vs. human readable.
    - handling inconsistencies
    - speed
6. Closed world assumption vs. open world assumption.

Related Technologies / Problems

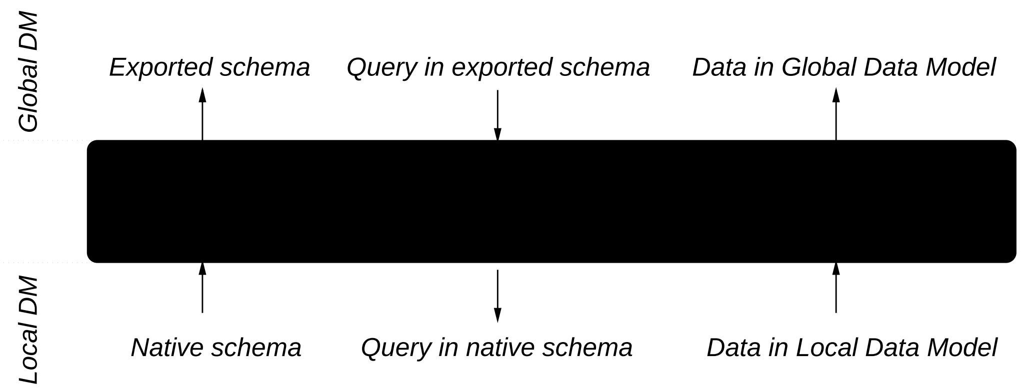
Distributed databases:

- sources are homogeneous,
- data is distributed a priori,
- sources are not autonomous.

Similarities at the optimization and execution level.

Information retrieval: Keyword search, no semantics.

Data mining: Discovering properties and patterns in data.



Picture of Wrappers

Where is the wrapper?

Built w.r.t. a mediator?

Thin vs. thick?

How intelligent is the wrapper?

Catalogs

Catalogs contain descriptions of:

- logical source contents
- source capabilities
- source completeness
- physical properities of the source and network
- statistics about the data
- source reliability
- mirror sources

(some) Research Prototypes

- DISCO (INRIA)
- Garlic (IBM)
- HERMES (U. of Maryland)
- InfoMaster (Stanford)
- Information Manifold (AT&T)
- IRO-DB (Versailles)
- SIMS (USC/ISI)
- The Internet Softbot / Occam / Razor (UW)
- TSIMMIS (Stanford)

Describing Information Sources

The mediated schema is the schema in which the user poses queries.

Each information source contains one or more relations.

Content descriptions provide the mappings between the two sets of relations.

Descriptions of Information Sources

Elements of source descriptions:

- Contents: source contains movies, their directors, cast.
- Constraints: all movies produced after 1965.
- Completeness: source contains all American movies,
- Capabilities: source must receive movie title or director as input, source can perform selections.

Desiderada from source descriptions

Distinguish between sources with closely related data: so we can prune access to irrelevant sources.

Enable easy addition of new information sources: because sources are dynamically being added and removed.

Be able to find sources relevant to a query: reformulate queries such that we obtain guarantees on which sources we access.

Query Reformulation Problem

Problem: reformulate the user query as a query on the information sources.

Given

- a query Q in terms of the mediated-schema relations,
- descriptions of the information sources,

Find a query Q' that uses only the information source relations, such that

- $Q' \models Q$ and
- Q' provides all the possible answers to Q using the sources.

Approches to Specification of Source Descriptions

Mediated schema relations defined as views over the source relations: TSIMMIS (Stanford), HERMES (Maryland), DISCO (INRIA, France).

Source relations defined as views over mediated-schema relations: Information Manifold (AT&T), Occam, Razor (UW), InfoMaster (Stanford).

Sources described as concepts in a description logic: SIMS (ISI/USC), Catarci and Lenzerini (Rome).

The Global As View Approach

The mediated-schema relations are described in terms of the source relations.

Movies and their years can be obtained from either DB_1 or DB_2:

$$MovieYear(title, year) :- DB_1(title, director, year)$$
$$MovieYear(title, year) :- DB_2(title, director, year)$$

Movie reviews can be obtaind by joining DB_1 and DB_3:

$$\begin{aligned} MovieRev(title, director, review) :- \ & DB_1(title, director, year)\ \& \\ & DB_3(title, review). \end{aligned}$$

Query Reformulation in GAV

Query reformulation is done by rule unfolding.

Query: Find reviews for 1997 movies:

$$
\begin{aligned}
q(title, review) :- \; & MovieYear(title, 1997) \,\&\\
& MovieRev(title, director, review).
\end{aligned}
$$

Reformulated query on the sources:

$$
\begin{aligned}
q(title, review) :- \; & DB_1(title, director, year) \,\&\, DB_3(title, review)\\
q(title, review) :- \; & DB_1(title, director, year) \,\&\\
& DB_2(title, director, year) \,\&\, DB_3(title, review)
\end{aligned}
$$

A containment check shows that the second rule is redundant.

The Local As View Approach

Every data source is described as a query expression over mediated-schema relations:

$$S_1 : V_1(title, year, director) \subseteq Movie(title, year, director, genre) \& \\ American(director) \& \\ year \geq 1960 \& genre = Comedy.$$

$$S_2 : V_2(title, review) \subseteq Movie(title, year, director, genre) \\ year \geq 1990 \& Review(title, review).$$

Query Reformulation

Find reviews for comedies produced after 1950:

$$q(title, review) :- Movie(title, year, director, Comedy) \;\&\; year \geq 1950 \;\&\; Review(title, review).$$

$$V_1(title, year, director) \subseteq Movie(title, year, director, genre) \;\&\; American(director) \;\&\; year \geq 1960 \;\&\; genre = Comedy.$$

$$V_2(title, review) \subseteq Movie(title, year, director, genre) \; year \geq 1990 \;\&\; Review(title, review).$$

The reformulated query on the sources:

$$q'(title, review) :- V_1(title, year, director) \;\&\; V_2(title, review).$$

Comparison of the Approaches

See [Ullman, ICDT-97] for a detailed comparison.

Local as View approach:

Easier to add sources: specify the query expression.

Easier to specify constraints on contents of the sources: they're part of the query expression describing them.

Global as view:

Query reformulation is straightforward.

More natural to have several levels of mediation.

Does presence of semistructured data make a difference?

Reformulation Algorithm

How do we translate the query on the mediated-schema to a query on the sources?

Translate the query:

$$q(title, review) :- Movie(title, year, director, Comedy)\ \&\ year \geq 1950\ \&\ Review(title, review).$$

into a query on the sources:

$$q'(title, review) :- V_1(title, year, director)\ \&\ V_2(title, review).$$

Conceptually: we have a set of precomputed queries and we want to use them to answer a new query.

Rewriting Queries Using Views

Given a set of view definitions $V_1, V_2, \ldots, V_n$ and a query Q, find a query Q' that:

- uses only the views $V_1, V_2, \ldots, V_n$, and
- is contained in Q and yields as many answers of Q as possible from the views.

The query rewriting problem is also important for:

- query optimization (Surajit), GMAP,
- data warehousing,
- some AI problems...

See Rachel for more details on this problem.

Algorithms for Query Rewriting

Algorithms and complexity depend on:

- the language used in the query and view definitions,
- equivalent rewriting or contained rewriting?

Approaches to the rewriting problem:

- Apply a set of rewriting rules until the views are explicit in the query.
- Establish a bound on the size of the possible rewritings, and search the resulting space.
- Write rules to reconstruct the base relations from the rules.

Views with Binding Patterns

$CitationDB^{bf}(X,Y) \subseteq Cites(X,Y)$
$CitingPapers^{f}(X) \subseteq Cites(X,Y)$
Query: $Q(X) :- Cites(X,a)$

There is no equivalent rewriting with 1 subgoal.

However, there is a rewriting with two subgoals:

$q(X) :- CitingPapers(Y) \,\&\, CitationDB(Y,a).$

Rajaraman, Sagiv & Ullman, 95: the bound on the size an equivalent rewriting is $\#atoms + \#variables.$

Contained Rewritings for Views with Binding Patterns

Kwok and Weld, 96: for contained rewritings, there may be no bound on the size of the rewriting:

$AttDB(X) \subseteq AttPapers(X)$
$CitationDB^{bf}(X, Y) \subseteq Cites(X, Y)$
$AwardDB^{b}(X) \subseteq AwardPaper(X)$
Query: $Q(X) :- AwardPaper(X)$

There is no bound on the size of contained rewritings:
$Q'(X) :- AttDB(X) \,\&\, AwardDB(X)$
$Q'(X) :- AttDB(V) \,\&\, CitationDB(V, X_1) \ldots$
$CitationDB(X_n, X) \,\&\, AwardDB(X)$

Recursive Rewritings

Duschka & Levy, 97: we can find a maximal recursive rewriting:

$papers(X) :- AttDB(X)$
$papers(X) :- papers(Y) \,\&\, CitationDB(Y, X)$
$Q'(X) :- papers(X) \,\&\, AwardDB(X).$

Recursive rewritings also arise when the views and queries:

- are recursive,
- mention relations satisfying functional dependencies,
- are expressed using description logics.

The Query Optimization Problem

The goal of a query optimizer:

> Translate a declarative query into an equivalent imperative program of *minimal* cost.

The imperative program is a query execution plan (QEP): an operator tree in some algebra.

Basic notions in optimization:

- search space,
- search strategy,
- cost model.

Current Research Directions

- Query optimization and execution (see Marc and Zack for details...)
- Combining structured and unstructured sources.
- Exploiting capabilities of sources.
- Combining querying and browsing.
- Obtaining maximal answers.
- Adding probabilistic information to descriptions (see AnHai for details).
- Consistency?
- Matching objects across sources.

Final Conclusions

Data integration at a crossroads:

- We need a more refined classification of the field.
- Test current systems on real applications.
- Benchmarks!
- Industry: service only, or also products?