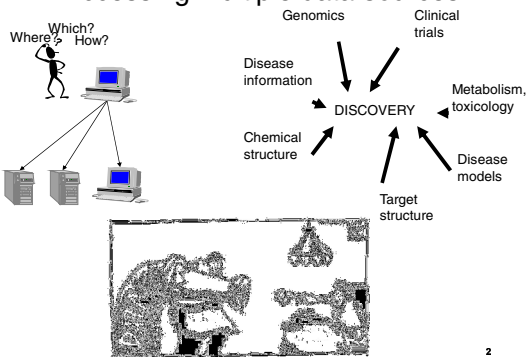
Integration of data sources

Patrick Lambrix
Department of Computer and Information Science
Linköpings universitet

Accessing multiple data sources

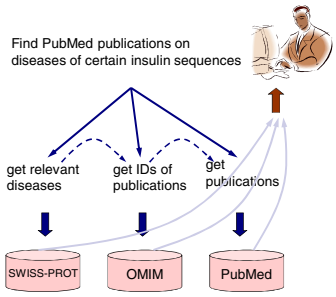


Access to multiple data sources-Problems

- Users need good knowledge on where the required information is stored and how it can be accessed
- Representation of an entity in different data sources can be different.

  Same name in different data sources can refer to different entities.

Queries over multiple data sources

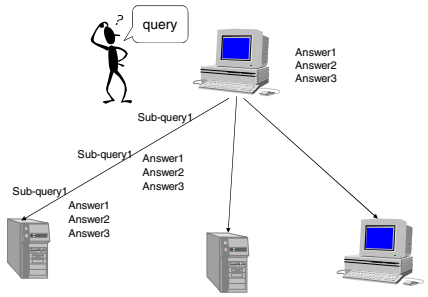


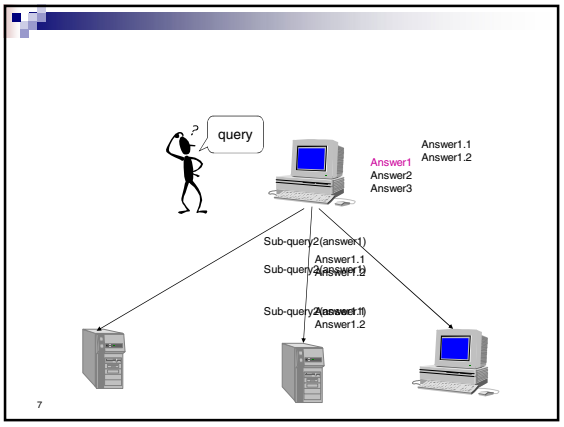
- Find
- Divide & order
- Execute
- Combine

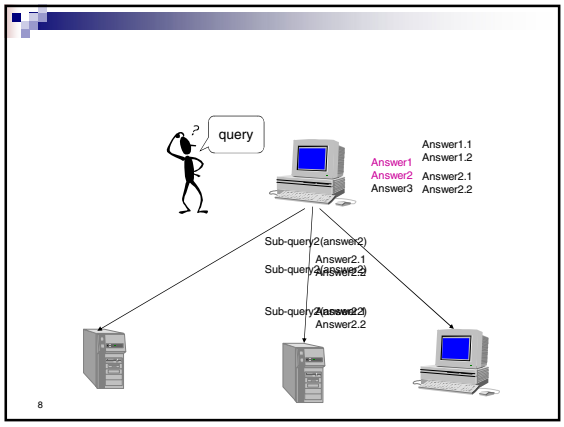
Access to multiple data sources - steps

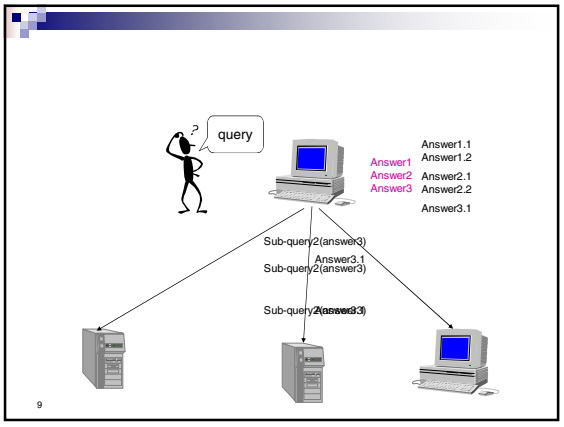
- Decide which data sources should be used
- Divide query into sub-queries to the data sources
- Decide in which order to send sub-queries to the data sources
- Send sub-queries to the data sources - use the terminology of the data sources
- Merge results from the data sources to an answer for the original query

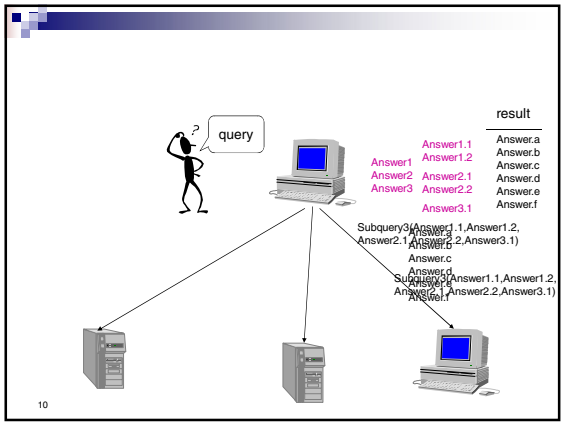
→ mistake in any step can lead to inefficient processing of the query or failure to get a result











Problem formulation

- Data source properties
  - Autonomous data sources
  - Different data models
  - Differences in terminology
  - Overlapping, redundant data
- Integration aims to provide transparent access to multiple heterogenous data sources
  - uniform query language
  - uniform representation of results

Data source 1
Protein(name, authors, date, organism)
Article(authors, title, year)
date>1995

Data source 2
Structure(name, structure, organism)
date>2000

11

Problem formulation

Protein(name, date, organism)
ProteinStructure(name, structure)

- Data source properties
  - Autonomous data sources
  - Different data models
  - Differences in terminology
  - Overlapping, redundant data
- Integration aims to provide transparent access to multiple heterogenous data sources
  - uniform query language
  - uniform representation of results

Data source 1
Protein(name, authors, date, organism)
Article(authors, title, year)
date>1995

Data source 2
Structure(name, structure, organism)
date>2000

12

Methods for integration

- Link driven federations
  - Explicit links between data sources.
- Warehousing
  - Data is downloaded, filtered, integrated and stored in a warehouse. Answers to queries are taken from the warehouse.
- Mediation or View integration
  - A global schema is defined over all data sources.

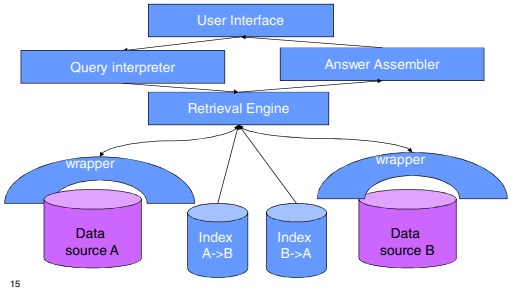
13

Link driven federations

- Creates explicit links between data sources
- query: get interesting results and use web links to reach related data in other data sources

14

Link driven federations



15

SRS

- Integrates more than 300 resources
- Possible to add own resources
- interface: SRSWWW, getz
- http://srs.ebi.ac.uk/

16

SRS – query language

- text search

[swissprot-des:kinase]

documents in swissprot that contain 'kinase' in the 'description'-field

[swissprot-des:kin*]

documents in swissprot that contain a word that starts with 'kin' in the 'description'-field

17

SRS – query language

- boolean operators:

and (&), or (|), andnot (!)

[swissprot-des:(adrenergic & receptor) ! (alpha1A)]

documents in swissprot that contain 'adrenergic' and 'receptor' in the 'description'-field, but not 'alpha1A'

18

SRS – query language

- boolean operators:
  and (&), or (|), andnot (!)

[swissprot-des:kinase] & [swissprot-org:human]

documents in swissprot that contain 'kinase' in the 'description'-field and 'human' in the 'organism'-field

19

SRS – query language

- links

[swissprot-des:kinase] > PDB

documents in PDB that are referred to from documents in swissprot that contain 'kinase' in the 'description'-field

20

SRS – query language

- links

[swissprot-id: acha_human] > prosite > swissprot

documents in swissprot that are referred to from documents in prosite that are referred to from documents in swissprot that contain 'acha_human' in the 'id'- field

21

SRS – query language

- links

[swissprot-org:human] >
[swissprot-features:transmem]

documents in swissprot that contain 'transmem' in the 'features'-field and that are referred to from documents in swissprot that contain 'human' in the 'organism'-field

22

SRS – query language

- multiple sources

[{swissprot sptremb}-des:kinase]

[dbs={swissprot sptremb}-des:kinase]
& [dbs-org:human]

23

Link driven federations

- Advantages
  - complex queries
  - fast
- Disadvantages
  - require good knowledge
  - syntax based
  - terminology problem not solved

24

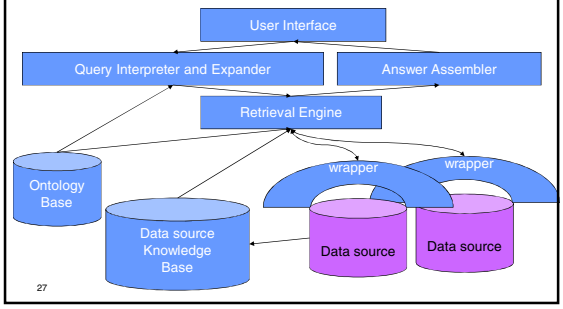
25

Mediation

- Define a global schema over the data sources
- high level query language

26

Mediation



27

Mediation

- Advantages
  - complex queries
  - requires less knowledge
  - solution for terminology problem
  - semantics based

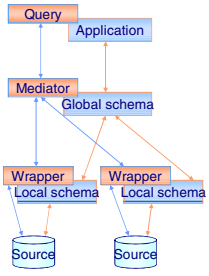
28

Mediation

- Disadvantages
  - more computation
  - view maintenance

29

Mediation

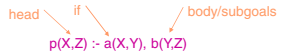


- Query problem
  How to answer queries expressed using the global schema.
- Modeling problem
  How to model the global schema, data sources and mappings.

30

Queries

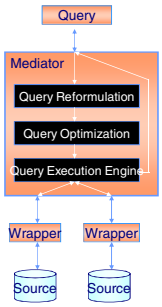
- Queries use the global schema
- Conjunctive queries
  - select-project-join queries



q(name, structure) :- Protein(name, 2001, 'human'),
ProteinStructure(name, structure)

- Mediator reformulates queries in terms of a set of queries that use the local schemas. Equivalence and containment of queries needs to be preserved.
  - Q1 is contained in Q2 if the result of Q1 is a subset of the result of Q2.

Mediator

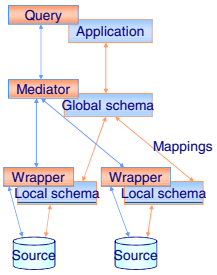


- Mediator is responsible for query processing
  - reformulation of queries, decide query plan
  - query optimization
  - execution of query plan, assemble results into final answer

Issues:

- Semantically correct reformulation
- Access only relevant data sources

Knowledge



- Description of data source content
  - global schema (domain model/ontology)
  - local schema (data source model)
- Information for integration
  - mapping
- Capabilities
  - attributes and constraints
  - processing capabilities
  - completeness
  - cost of query answering
  - reliability
- Used for
  - selection of relevant data sources
  - query plan formulation
  - query plan optimization

Mapping

- Relation between domain and data source content

Global schema:
Protein(name, date, organism)
ProteinStructure(name, structure)

Data source local schema:
DS1(name, authors, date, organism)
DS2(name, structure, organism)

- Global as view
  The global schema is defined in terms of source terminology

Protein(name, date, organism) :- DS1(name, authors, date, organism)
ProteinStructure(name, structure) :- DS2(name, structure, organism)

Mapping

- Relation between domain and data source content

Global schema:
Protein(name, date, organism)
ProteinStructure(name, structure)

Data source local schema:
DS1(name, authors, date, organism)
DS2(name, structure, organism)

- Local as view
  The sources are defined in terms of the global schema.

DS1(name, authors, date, organism) -:
Protein(name, date, organism), date >1995
DS2(name, structure, organism) -: Protein(name, date, organism),
ProteinStructure(name, structure), date >2000

Query processing in GAV

Query: give name and structure for human proteins with date '2001'.
q(name, structure) :- Protein(name, 2001, 'human'),
ProteinStructure(name, structure)

GAV: Protein(name, date, organism) :- DS1(name, authors, date, organism)
ProteinStructure(name, structure) :- DS2(name, structure, organism)

- No explicit representation of data source content
- Mapping gives direct information about which data satisfies the global schema.
- Query is processed by expanding the query atoms according to their definitions.

New query: q(name, structure) :-
DS1(name, authors, 2001, 'human'), DS2(name, structure, organism)

Query processing in LAV

Query: give name and structure for human proteins with date '2001'.
q(name, structure) :- Protein(name, 2001, 'human'),
ProteinStructure(name, structure)

LAV: DS1(name, authors, date, organism) -:
Protein(name, date, organism), date >1995
DS2(name, structure, organism) -: Protein(name, date, organism),
ProteinStructure(name, structure), date >2000

- Mapping does not give direct information about which data satisfies the global schema.
- To answer the query it needs to be inferred how the mappings should be used.

Query processing in LAV

Query: give name and structure for human proteins with date '2001'.
q(name, structure) :- Protein(name, 2001, 'human'),
ProteinStructure(name, structure)

LAV: DS1(name, authors, date, organism) -:
Protein(name, date, organism), date >1995
DS2(name, structure, organism) -: Protein(name, date, organism),
ProteinStructure(name, structure), date >2000

- Bucket algoritm (Information Manifold)
  - For each sub-goal in query create bucket of relevant views.
  - Define rewritings of query. Each rewriting consists of one conjunct from every bucket. Check whether the resulting conjunction is contained in the query.
  - The result is the union of the rewritings.

New query: q(name, structure) :-
DS1(name, authors, 2001, 'human'), DS2(name, structure, organism)

Comparison GAV - LAV

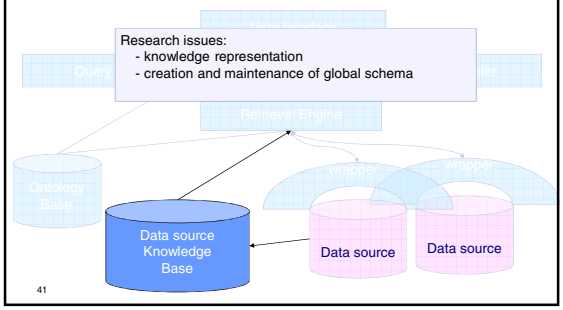
- Global as view
  - Clear how data sources interact
  - When a data source is added, the global schema can change
  - Query processing is easy
- Local as view
  - Each data source is specified in isolation
  - Easy to add data sources
  - Easier to specify constraints on the contents of sources
  - Query processing requires reasoning

Capabilities

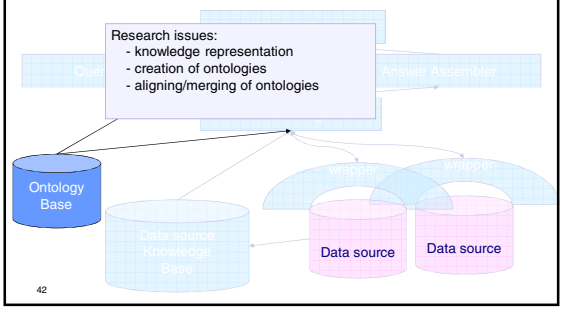
- Most common capabilities describe attributes
  - f - free, attribute can be specified or not
  - b - bound, a value must be specified for the attribute, all values are permitted
  - u - unspecified, not permitted to specify a value for the attribute
  - c[S] - value should be one of the values in finite set S
  - o[S] - value is not specified or one of the values in finite set S

DS1: (name, authors, date, organism) f b c[human mouse]

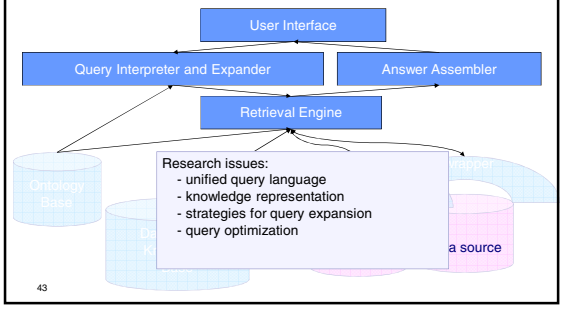
Mediation



Mediation

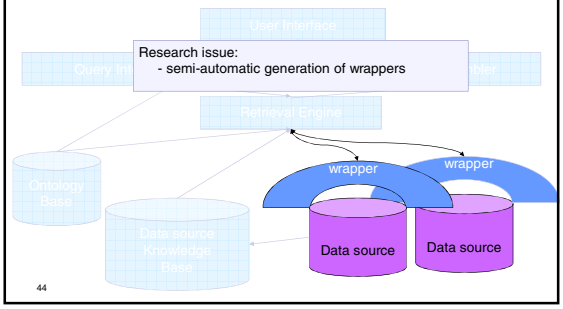


Mediation



43

Mediation



44

45