Data Integration 1

Giuseppe De Giacomo e Antonella Poggi

Dipartimento di Informatica e Sistemistica "Antonio Ruberti"

Università di Roma "La Sapienza"

Seminari di Ingegneria Informatica: Integrazione di Dati e Servizi

A.A. 2005/06

Prof. Giuseppe De Giacomo

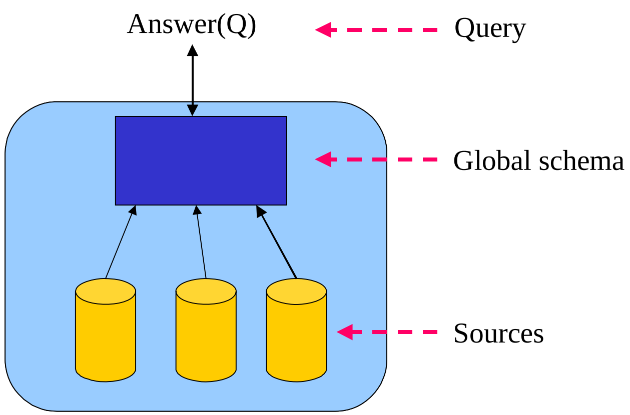
Laurea Specialistica in Ingegneria Informatica

Università di Roma "La Sapienza"

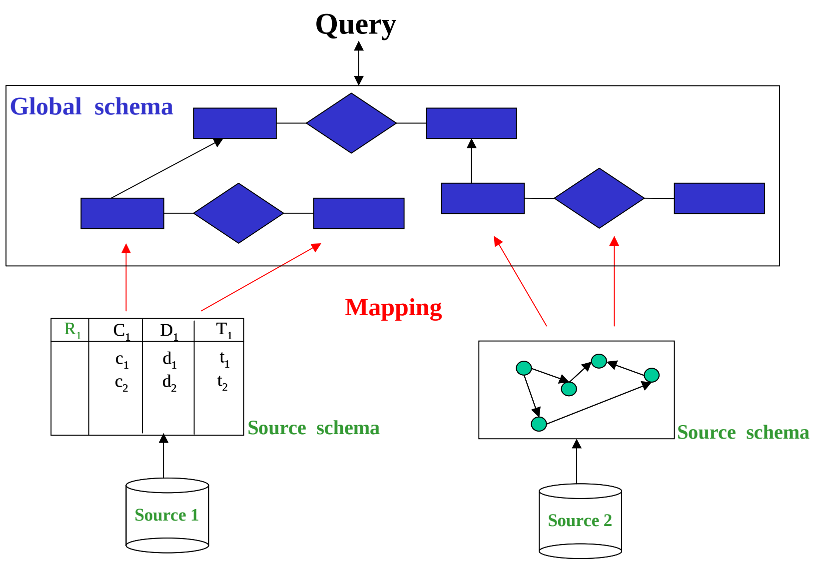
Data integration 1: outline

- Introduction to data integration
- Data integration: logical formalization

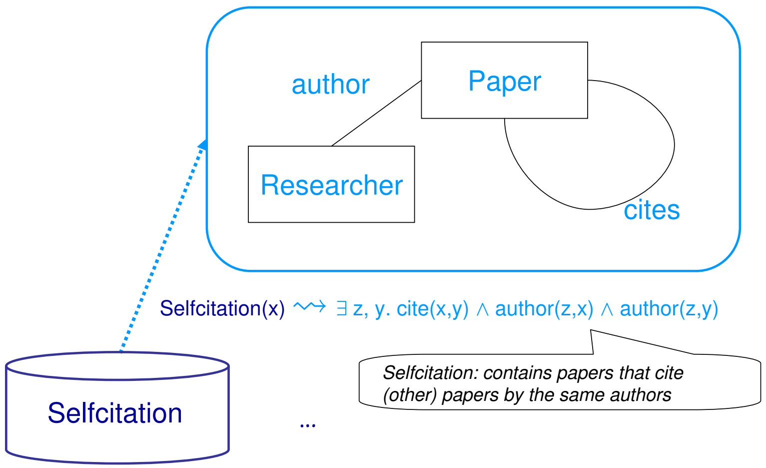
Data integration



Data integration



An example



IT hype

The current trend in IT industry is operating in on-demand environments. Operating on-demand is based on three key elements:

- **Integration**: *"Integration creates the necessary business flexibility to optimize operations across and beyond the enterprise".*
- **Automation**: *"Automation reduces the complexity and cost of IT management and improves the availability and the resilience".*
- **Virtualization**: *"Virtualization provides a single consolidated view of (and easy access to) all available resources, which improves working capital and asset utilization".*

Data integration

Data integration is the problem of providing unified and transparent access to a set of autonomous and heterogeneous sources

- Growing market
- One of the major challenges for the future of IT
- At least two contexts
  - Intra-organization data integration (e.g., EIS)
  - Inter-organization data integration (e.g. integration on the Web)

Data integration: available industrial efforts

- Distributed database systems
- Information on demand
- Tools for source wrapping
- Tools based on database federation, e.g., DB2 Information Integrator
- Distributed query optimization

Integrated access to distributed data

Different approaches/architectures:

- **distributed databases**

  data sources are homogeneous databases under the control of the distributed database management system

- **multidatabase or federated databases**

  data sources are autonomous, heterogeneous databases; procedural specification

- **(mediator-based) data integration**

  access through a global schema mapped to autonomous and heterogeneous data sources; declarative specification

- **peer-to-peer data integration**

  network of autonomous systems mapped one to each other, without a global schema; declarative specification

Database federation tools: characteristics

- Physical transparency (masking from the user the physical characteristics of the sources)
- Heterogeinity (federating highly diverse types of sources)
- Extensibility
- Autonomy of data sources
- Performance (distributed query optimization)

However, current tools do not (directly) support logical or conceptual transparency

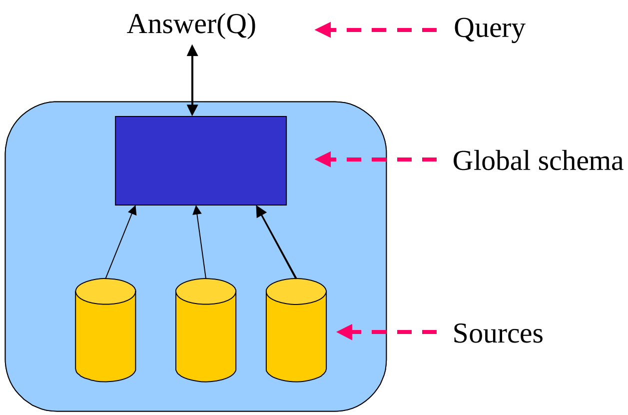
Logical transparency

Basic ingredients for achieving logical transparency:

- The global schema (ontology) provides a conceptual view that is independent from the sources
- The global schema is described with a semantically rich formalism
- The mappings are the crucial tools for realizing the independence of the global schema from the sources
- Obviously, the formalism for specifying the mapping is also a crucial point

All the above aspects are not appropriately dealt with by current tools. This means that data integration cannot be simply addressed on a tool basis.

Data integration



Main problems in data integration

1. How to construct the global schema
2. (Automatic) source wrapping
3. How to discover mappings between the sources and the global schema
4. Limitations in the mechanisms for accessing the sources
5. Data extraction, cleaning and reconciliation
6. How to process updates expressed on the global schema, and updates expressed on the sources ("read/write" vs "read-only" data integration)
7. The modeling problem: How to model the mappings between the sources and the global schema
8. The querying problem: How to answer queries expressed on the global schema
   This is view-based query answering!
9. Query optimization

Data integration 1: outline

- Introduction to data integration
- Data integration: logical formalization

Formal framework for data integration

A **data integration system** $\mathcal{I}$ is a triple $\langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$, where

- $\mathcal{G}$ is the global schema
  The global schema is a logical theory over an alphabet $\mathcal{A}_{\mathcal{G}}$
- $\mathcal{S}$ is the source schema
  The source schema is constituted simply by an alphabet $\mathcal{A}_{\mathcal{S}}$ *disjoint from* $\mathcal{A}_{\mathcal{G}}$
- $\mathcal{M}$ is the mapping between $\mathcal{S}$ and $\mathcal{G}$
  Different approaches to the specification of mapping

Semantics of a data integration system

Which are the databases that satisfy $\mathcal{I}$, i.e., which are the logical models of $\mathcal{I}$?

The databases that satisfy $\mathcal{I}$ are logical interpretations for $\mathcal{A}_{\mathcal{G}}$ (called global databases). We refer only to databases over a fixed infinite domain Γ of constants.

Let $\mathcal{C}$ be **a source database** over Γ (also called source model), fixing the extension of the predicates of $\mathcal{A}_{\mathcal{S}}$ (thus modeling the data present in the sources).

The set of models of (i.e., databases for $\mathcal{A}_{\mathcal{G}}$ that satisfy) $\mathcal{I}$ relative to $\mathcal{C}$ is:

$$sem^{\mathcal{C}}(\mathcal{I}) = \{ \mathcal{B} \mid \mathcal{B} \text{ is a } \mathcal{G}\text{-model (i.e., a global database that is legal wrt } \mathcal{G}) \text{ and is an } \mathcal{M}\text{-model wrt } \mathcal{C} \text{ (i.e., satisfies } \mathcal{M} \text{ wrt } \mathcal{C}) \}$$

What it means to satisfy $\mathcal{M}$ wrt $\mathcal{C}$ depends on the nature of the mapping $\mathcal{M}$.

Semantics of queries to $\mathcal{I}$

A **query** q of arity n is a formula with n free variables.

If $\mathcal{D}$ is a database, then $q^{\mathcal{D}}$ denotes the extension of q in $\mathcal{D}$ (i.e., the set of n-tuples that are valuations in Γ for the free variables of q that make q true in $\mathcal{D}$).

If q is a query of arity n posed to a data integration system $\mathcal{I}$ (i.e., a formula over $\mathcal{A}_{\mathcal{G}}$ with n free variables), then the set of **certain answers to q wrt $\mathcal{I}$ and $\mathcal{C}$** is

$$cert(q, \mathcal{I}, \mathcal{C}) = \{(c_1, \ldots, c_n) \in q^{\mathcal{B}} \mid \forall \mathcal{B} \in sem^{\mathcal{C}}(\mathcal{I})\}.$$

Note: query answering is logical implication.

Note: complexity will be mainly measured wrt the size of the source database $\mathcal{C}$, and will refer to the problem of deciding whether $\vec{\mathbf{c}} \in cert(q, \mathcal{I}, \mathcal{C})$, for a given $\vec{\mathbf{c}}$.

Databases with incomplete information, or Knowledge Bases

- Traditional database: one model of a first-order theory

  Query answering means evaluating a formula in the model
- Database with incomplete information, or Knowledge Base: set of models (specified, for example, as a restricted first-order theory)

  Query answering means computing the tuples that satisfy the query in all the models in the set

There is a strong connection between query answering in data integration and query answering in databases with incomplete information under constraints (or, query answering in knowledge bases).

Query answering with incomplete information

- [Reiter '84]: relational setting, databases with incomplete information modeled as a first order theory
- [Vardi '86]: relational setting, complexity of reasoning in closed world databases with unknown values
- Several approaches both from the DB and the KR community
- [van der Meyden '98]: survey on logical approaches to incomplete information in databases

The mapping

How is the mapping $\mathcal{M}$ between $\mathcal{S}$ and $\mathcal{G}$ specified?

- Are the sources defined in terms of the global schema?

  Approach called **source-centric**, or **local-as-view**, or **LAV**
- Is the global schema defined in terms of the sources?

  Approach called **global-schema-centric**, or **global-as-view**, or **GAV**
- A mixed approach?

  Approach called **GLAV**

GAV vs LAV – example

Global schema: movie($Title, Year, Director$)

european($Director$)

review($Title, Critique$)

Source 1: $r_1(Title, Year, Director)$ since 1960, European directors

Source 2: $r_2(Title, Critique)$ since 1990

Query: Title and critique of movies in 1998

$\exists D.\ \text{movie}(T, 1998, D) \land \text{review}(T, R)$, written

$\{\ (T, R) \mid \text{movie}(T, 1998, D) \land \text{review}(T, R)\ \}$

Formalization of LAV

In LAV the mapping $\mathcal{M}$ is constituted by a set of assertions:

$$s \ \rightsquigarrow\ \phi_{\mathcal{G}}$$

one for each source element s in $\mathcal{A}_{\mathcal{S}}$, where $\phi_{\mathcal{G}}$ is a **query** over $\mathcal{G}$ of the arity of s.

Given source database $\mathcal{C}$, a database $\mathcal{B}$ for $\mathcal{G}$ satisfies $\mathcal{M}$ wrt $\mathcal{C}$ if for each $s \in \mathcal{S}$:

$$s^{\mathcal{C}} \ \subseteq\ \phi_{\mathcal{G}}{}^{\mathcal{B}}$$

In other words, the assertion means $\forall \vec{\mathbf{x}}\ (s(\vec{\mathbf{x}}) \rightarrow \phi_{\mathcal{G}}(\vec{\mathbf{x}}))$.

The mapping $\mathcal{M}$ and the source database $\mathcal{C}$ do **not** provide direct information about which data satisfy the global schema. Sources are views, and we have to answer queries on the basis of the available data in the views.

LAV – example

Global schema: movie($Title, Year, Director$)
european($Director$)
review($Title, Critique$)

LAV: associated to source relations we have views over the global schema

$r_1(T, Y, D) \ \rightsquigarrow\ \{\ (T, Y, D) \mid \text{movie}(T, Y, D) \land \text{european}(D) \land Y \geq 1960\ \}$

$r_2(T, R) \ \rightsquigarrow\ \{\ (T, R) \mid \text{movie}(T, Y, D) \land \text{review}(T, R) \land Y \geq 1990\ \}$

The query $\{\ (T, R) \mid \text{movie}(T, 1998, D) \land \text{review}(T, R)\ \}$ is processed by means of an inference mechanism that aims at re-expressing the atoms of the global schema in terms of atoms at the sources. In this case:

$$\{\ (T, R)\ \mid\ r_2(T, R) \land r_1(T, 1998, D)\ \}$$

Formalization of GAV

In GAV the mapping $\mathcal{M}$ is constituted by a set of assertions:

$$g \ \rightsquigarrow\ \phi_{\mathcal{S}}$$

one for each element g in $\mathcal{A}_{\mathcal{G}}$, where $\phi_{\mathcal{S}}$ is a **query** over $\mathcal{S}$ of the arity of g.

Given source database $\mathcal{C}$, a database $\mathcal{B}$ for $\mathcal{G}$ satisfies $\mathcal{M}$ wrt $\mathcal{C}$ if for each $g \in \mathcal{G}$:

$$g^{\mathcal{B}} \ \supseteq\ \phi_{\mathcal{S}}{}^{\mathcal{C}}$$

In other words, the assertion means $\forall \vec{\mathbf{x}}\ (\phi_{\mathcal{S}}(\vec{\mathbf{x}}) \rightarrow g(\vec{\mathbf{x}}))$.

Given a source database, $\mathcal{M}$ provides direct information about which data satisfy the elements of the global schema. Relations in $\mathcal{G}$ are views, and queries are expressed over the views. Thus, it seems that we can simply evaluate the query over the data satisfying the global relations (as if we had a single database at hand).

GAV – example

Global schema: movie($Title, Year, Director$)

european($Director$)

review($Title, Critique$)

GAV: associated to relations in the global schema we have views over the sources

movie$(T, Y, D) \leadsto \{ (T, Y, D) \mid r_1(T, Y, D) \}$

european$(D) \leadsto \{ (D) \mid r_1(T, Y, D) \}$

review$(T, R) \leadsto \{ (T, R) \mid r_2(T, R) \}$

GAV – example (constraints) – *see more later*

Global schema containing constraints:

movie($Title, Year, Director$) european($Director$) review($Title, Critique$)

european_movie_60s($Title, Year, Director$)

$\forall T, Y, D.$ european_movie_60s$(T, Y, D) \supset$ movie(T, Y, D)

$\forall D. \exists T, Y.$ european_movie_60s$(T, Y, D) \supset$ european(D).

GAV mappings:

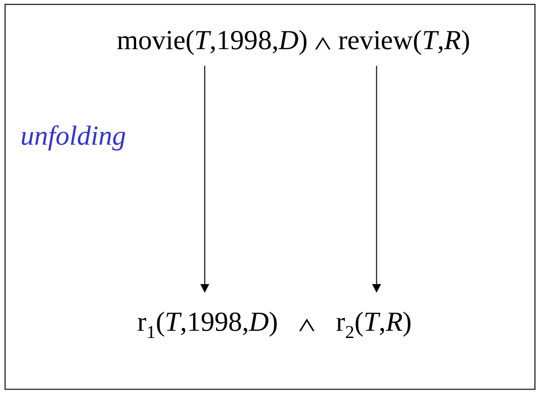
european_movie_60s$(T, Y, D) \leadsto \{ (T, Y, D) \mid r_1(T, Y, D) \}$

european$(D) \leadsto \{ (D) \mid r_1(T, Y, D) \}$

review$(T, R) \leadsto \{ (T, R) \mid r_2(T, R) \}$

GAV – example of query processing

The query $\{ (T, R) \mid$ movie$(T, 1998, D) \wedge$ review$(T, R) \}$ is processed by means of unfolding, i.e., by expanding each atom according to its associated definition in $\mathcal{M}$, so as to come up with source relations. In this case:



GAV and LAV – comparison

LAV: (Information Manifold, DWQ)

- Quality depends on how well we have characterized the sources
- High modularity and extensibility (if the global schema is well designed, when a source changes, only its definition is affected)
- Query processing needs reasoning (query answering complex)

GAV: (Carnot, SIMS, Tsimmis, IBIS, Momis, DisAtDis, . . .)

- Quality depends on how well we have compiled the sources into the global schema through the mapping
- Whenever a source changes or a new one is added, the global schema needs to be reconsidered
- Query processing can be based on some sort of unfolding (query answering looks easier – without constraints)

Beyond GAV and LAV: GLAV

In GLAV, the mapping $\mathcal{M}$ is constituted by a set of assertions:

$$\phi_{\mathcal{S}} \leadsto \phi_{\mathcal{G}}$$

where $\phi_{\mathcal{S}}$ is a **query** over $\mathcal{S}$, and $\phi_{\mathcal{G}}$ is a **query** over $\mathcal{G}$ of the arity $\phi_{\mathcal{S}}$.

Given source database $\mathcal{C}$, a database $\mathcal{B}$ that is legal wrt $\mathcal{G}$ satisfies $\mathcal{M}$ wrt $\mathcal{C}$ if for each assertion in $\mathcal{M}$:

$$\phi_{\mathcal{S}}{}^{\mathcal{C}} \subseteq \phi_{\mathcal{G}}{}^{\mathcal{B}}$$

In other words, the assertion means $\forall \vec{\mathbf{x}}\,(\phi_{\mathcal{S}}(\vec{\mathbf{x}}) \rightarrow \phi_{\mathcal{G}}(\vec{\mathbf{x}}))$.

As for LAV, the mapping $\mathcal{M}$ does **not** provide direct information about which data satisfy the global schema: to answer a query q over $\mathcal{G}$, we have to **infer** how to use $\mathcal{M}$ in order to access the source database $\mathcal{C}$.

Example of GLAV

Global schema: $Work(Person, Project)$, $Area(Project, Field)$

Source 1: $HasJob(Person, Field)$

Source 2: $Teach(Professor, Course)$, $In(Course, Field)$

Source 3: $Get(Researcher, Grant)$, $For(Grant, Project)$

GLAV mapping:

$$\{ (r,f) \mid HasJob(r,f) \} \leadsto \{ (r,f) \mid Work(r,p) \wedge Area(p,f) \}$$
$$\{ (r,f) \mid Teach(r,c) \wedge In(c,f) \} \leadsto \{ (r,f) \mid Work(r,p) \wedge Area(p,f) \}$$
$$\{ (r,p) \mid Get(r,g) \wedge For(g,p) \} \leadsto \{ (r,p) \mid Work(r,p) \}$$

GLAV: a technical observation

In GLAV, the mapping $\mathcal{M}$ is constituted by a set of assertions:

$$\phi_{\mathcal{S}} \leadsto \phi_{\mathcal{G}}$$

Each such assertion can be rewritten wlog by introducing a **new predicate** r (not to be used in the queries) of the same arity as the two queries and replace the assertion with the following two:

$$\phi_{\mathcal{S}} \leadsto r \qquad r \leadsto \phi_{\mathcal{G}}$$

In other words, we replace $\forall \vec{\mathbf{x}}\,(\phi_{\mathcal{S}}(\vec{\mathbf{x}}) \rightarrow \phi_{\mathcal{G}}(\vec{\mathbf{x}}))$ with $\forall \vec{\mathbf{x}}\,(\phi_{\mathcal{S}}(\vec{\mathbf{x}}) \rightarrow r(\vec{\mathbf{x}}))$ and $\forall \vec{\mathbf{x}}\,(r(\vec{\mathbf{x}}) \rightarrow \phi_{\mathcal{G}}(\vec{\mathbf{x}}))$.