POLITECNICO
MILANO 1863

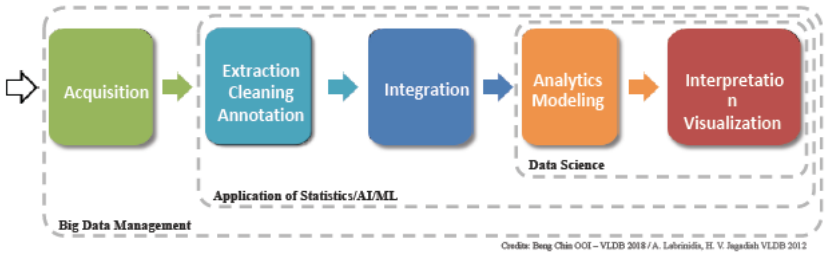
Introduction to data integration

Cinzia Cappiello
A.A. 2023-2024

These slides are based on Prof. Tanca slides.

Motivation:
The reality behind each data analysis task

- The actual implementation of the Data Analysis (ML, statistics, Data Mining…) algorithm is usually less than 5% lines of code in a real, non-trivial application
- The main effort (i.e. those 95% LOC) is spent on:
  - Data cleaning & annotation
  - Data extraction, transformation, loading
  - Data integration & pruning
  - Parameters tuning
  - Model training & deployment
  - …



The Data Integration problem is:

Combining data coming from different data sources, providing the user with a unified vision of the data

→ Detecting correspondences between similar concepts that come from different sources, and conflict solving

3

The four V's of Big data in data integration

- **Volume:** Not only can each data source contain a *huge volume of data*, but also the number of data sources has grown to be in the *millions*.
- **Velocity:** As a direct consequence of the *rate* at which data is being collected and continuously made available, many of the data sources are *very dynamic*.
- **Variety:** Data sources (even in the same domain) are extremely *heterogeneous* both at the schema level, regarding how they structure their data, and at the instance level, regarding how they describe the same real world entity, exhibiting considerable variety even for substantially similar entities.
- **Veracity**: Data sources (even in the same domain) are of widely differing *qualities*, with significant differences in the coverage, accuracy and timeliness of data provided. This is consistent with the observation that "1 in 3 business leaders do not trust the information they use to make decisions."

(Xin Luna Dong, Divesh Srivastava, VLDB2013 tutorial)

4

Information Systems in the Era of Big data

The Variety dimension:

people and enterprises need to integrate data and the systems that handle those data: relational DBMSs and their extensions, legacy data and legacy DBMSs, sensors and user-generated content produce **heterogeneous**, structured or unstructured data

The Veracity dimension:

Data Quality is the most general and used term, and represents a number of quality aspects besides veracity:

- Completeness,
- Validity,
- Consistency,
- Timeliness
- Accuracy
- Ethics and fairness (a new entry!)

5

Heterogeneity derives from various forms of AUTONOMY

- Design (representation) autonomy,
- Communication (querying) autonomy, and
- Execution (algorithmic) autonomy

….. GENERATE HETEROGENEITY,

….while there is a need for interoperability among SW applications, services and information (databases, and others) managed by different organizations that need to reuse legacy applications, existing data repositories, and reconcile the different points of view adopted by the various players using the information.

Data integration approaches interoperability from a Data Perspective

7

(Big) Data Integration Problems

VARIETY (heterogeneity) among several data collections to be used together

1. Different platforms: Technological heterogeneity
2. Different data models at the participating DBMS → Model heterogeneity
3. Different query languages -> Language heterogeneity
4. Different data schemas and different conceptual representations in DBs previously developed → Schema (semantic) heterogeneity
5. Different values for the same info (due to errors or to different knowledge)→ instance (semantic) heterogeneity

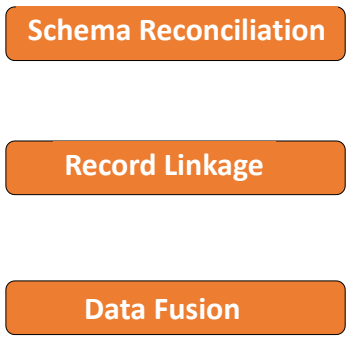
VERACITY: (Main) Data Quality dimensions:

1. Completeness,
2. Validity,
3. Consistency,
4. Timeliness
5. Accuracy

9

9

The steps of Data Integration: how is it done?



Schema reconciliation: mapping the **data structure** (if it exists!)

Record linkage (aka Entity resolution): data matching based on **the same content**

Data fusion: reconciliation of **non-identical content**

10

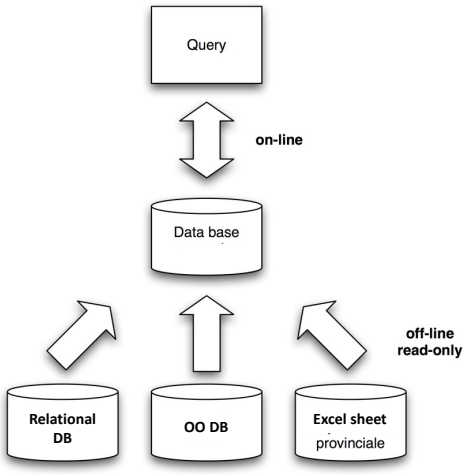
Relevant Ways of

Integrating Database Systems

1. Use a materialized database (data are merged in a new database) → Extract-Transform-Load Systems
   → Data Warehouses: Materialized integrated data sources
2. Use a virtual non-materialized data base (data remain at sources) →
   - Enteprise Information Integration (EII) (or Data Integration) Systems (common front-end to the various datasources)
   - Data Exchange (source-to-target)

11

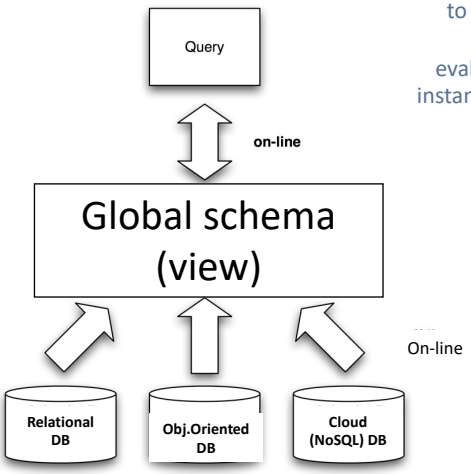
Materialized



12

Virtual

Given a target schema T, a set S of source schemas, a set M of mappings relating T to each element in S, and a query Q(T) against the target schema, use M to evaluate Q(T) over the set I(S) of source instances (i.e., over the data stored in the sources).



13

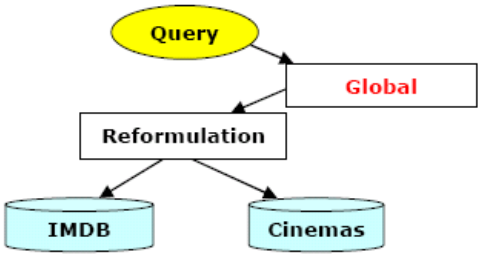
Materialized integration

- Large common stores came to be known as warehouses, and the software to access, scrape, transform, and load data into warehouses, became known as extract, transform, and load (ETL) systems.
- In a dynamic environment, one must perform ETL periodically (say once a day or once a week), thereby building up a history of the enterprise.
- The main purpose of a data warehouse is to allow systematic or ad-hoc data analysis and mining.
- Not appropriate when need to integrate the *operational* systems (keeping data up-to-date

15

Virtual integration

The virtual integration approach **leaves the information requested in the local sources.** The virtual approach will always return *a fresh answer to the query*. The query posted to the global schema is reformulated into the formats of the local information system. The information retrieved needs to be combined to answer the query.



16

Rationale

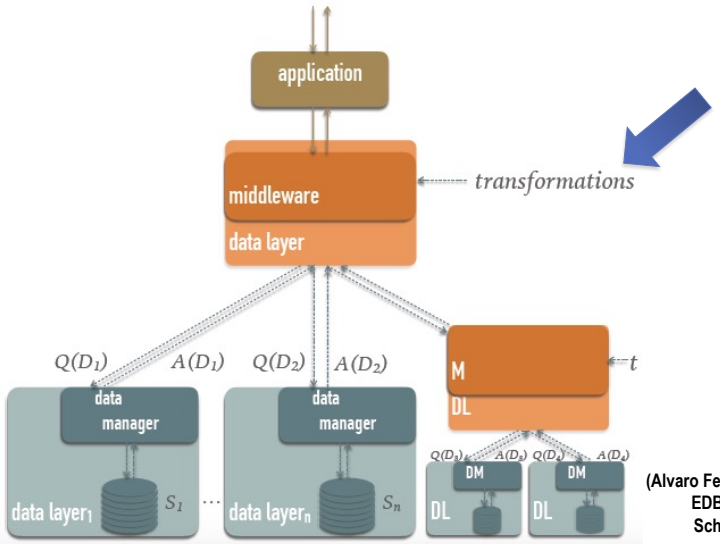
- The conventional wisdom is to use data warehousing and ETL products to perform data integration. However, there is a serious flaw in one aspect of this wisdom.
- Suppose one wants to integrate current (operational) data rather than historical information. Consider, for example, an e-commerce web site which wishes to sell hotel rooms over the web. The actual inventory of available hotel rooms exists in 100 or so information systems since Hilton, Hyatt and Marriott all run their own reservation systems.
- Applying ETL and warehousing to this problem will create a copy of hotel availability data, which is quickly out of date. BUT, if a web site sells a hotel room, based on this data, it has no way of guaranteeing delivery of the room, because somebody else may have sold the room in the meantime.

HENCE:

when you need the integrated data to be always up-to-date, USE VIRTUAL INTEGRATION !!!

17

A general framework for Data Integration



(Alvaro Fernandez, EDBT Summ. School 2017)

Query execution in the integration system

When a query is submitted, the integration system has to decompose it into queries against the component datasets.

1. determine first which parts of the query refer to which dataset,
2. which parts apply to only a single dataset and
3. which parts apply to data from different datasets.

The latter ones can only be evaluated over the integrated data (view), whereas the former ones can be evaluated within the component datasets.

The possible situations

- Data integration problems arise even in the simplest situation: unique, centralized DB…

…and become more and more complex

…up to the extreme case of transient, dynamic, initially unknown data sources…

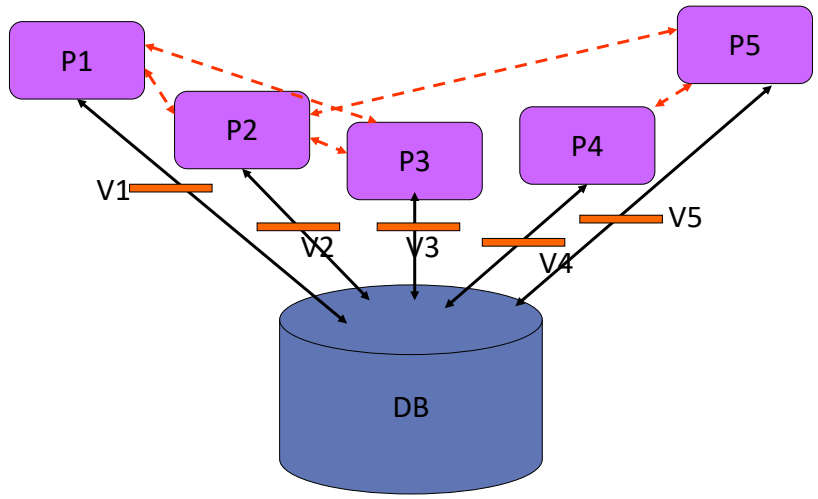
22

Why data integration even in a UNIQUE DB ?

- Each area of the company will ask the designer to design their (part of) database (DB)
- However, a lot of this data are common to some of the areas
- If the global company DB is just the collection of these partial DBs, there will be many redundancies (useless memory occupation)
- Worst of all, when updating one instance of these duplicates, maybe the other one will remain as before (inconsistencies)

The correct answer:

within a single company, an integrated, centralized DB



25

Why data integration in a UNIQUE DB ?

So that each datum, whatever application uses it, only appears once. This ELIMINATES useless REDUNDANCIES which would cause:

- Inconsistencies (i.e. contradictions)
- Useless memory occupation

Each functionality in the company will have its own personalized view

This is achieved by using view definitions

Views allow personalization as well as access control

26

Let's now reason on multiple data sources.

For the first part of our Data Integration study,

we concentrate on Relational Databases

27

Here and later we will use the concept of VIEW:
a conceptual tool to express the transformations

- Example:

  Create view *CourseCampus* **as**
  Select *CourseName, CampusName*
  From *Courses, Rooms*
  Where *Courses.RoomName=Rooms.RoomName*

  ```
  COURSES(CourseName, Teacher, RoomName)
  ROOMS(RoomName, BuildingNo, CampusName)
  ```

- View schema:

  ```
  CourseCampus(CourseName, CampusName)
  ```

28

Non-centralized DB's: Distributed DB (*)

- It is the simplest case of non-centralized DB
- Often, data for the same organization
- Integrated a-priori: same design pattern as in the centralized situation, with homogeneous technology and data model
- Once the distributed database has been designed as if it were centralized, at *distribution design time* the design decisions concern:
  - Fragmentation:
    - Vertical
    - Horizontal
  - Allocation
  - Replication
- NOT a matter of data integration, therefore not our concern here: see DB2 course

29

In general, the Data Integration problem is (recall):

Combining data coming from different data sources, providing the user with a unified vision of the data

→ Detecting correspondences between similar concepts that come from different sources, and conflict solving

33

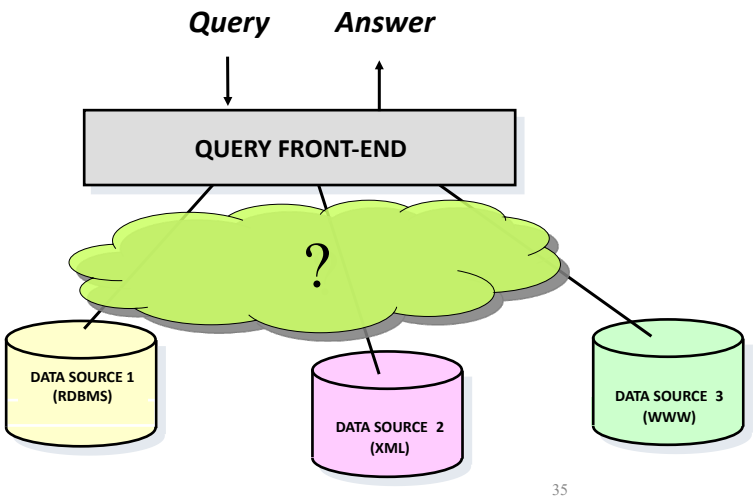
33

We study data integration for:

- Federated DBs
  - Homogeneous data: ***same*** **data model**
  - Heterogeneous data: ***different*** **data models: OO, XML, relational, RDF, Web data etc. (Semi-structured data**), even **free text (unstructured data)**
- Transient, initially unknown data sources

We propose techniques and methods which should be applied, adding complication as the situation becomes more complex

Data Integration with various data sources



The steps of Data Integration: how is it done?

✓

Schema Reconciliation

(If the sources have a schema) **Schema reconciliation**: mapping the **data structure** as in the centralized case

Record Linkage

Record linkage (aka Entity resolution): data matching based on **the same content**

Data Fusion

Data fusion: reconciliation of **non-identical content**

37

Approaches (recall)

- Data conversion (materialization): data are converted and stored into a unique system, e.g. a Data Warehouse
  - Multiple copies: untimeliness, redundancy, inconsistency
  - Application rewriting if one system is discarded
- Data exchange (virtual): creation of gateways between system pairs
  - Only appropriate when only two systems, no support for queries to data coming from multiple systems (e.g. peer-to-peer)

✓ Multidatabase (virtual): creation of a global schema

39

39

Data integration in the Multidatabase

We must build a system that:

- Supports access to different data sources
- "Knows" the contents of these data sources
- Integrates the different data sources by means of a unifying, global schema
- Receives queries expressed in the language of the global schema
- Distributes "rewritten" queries to the sources
- Combines the answers received from the sources to build the final answer

40

Many heterogeneities (recall):

- Same data model, different systems e.g. relational (Oracle, Sybase, DB2…) → *technological heterogeneity*
- Different data models, e.g. hierarchical or network (IMS, Codasyl…), relational, OO → *model heterogeneity*
- Same data model, different query languages (SQL, Quel) → *language heterogeneity*
- Semi- or unstructured data (HTML, XML, multimedia…) → *again model heterogeneity*

42

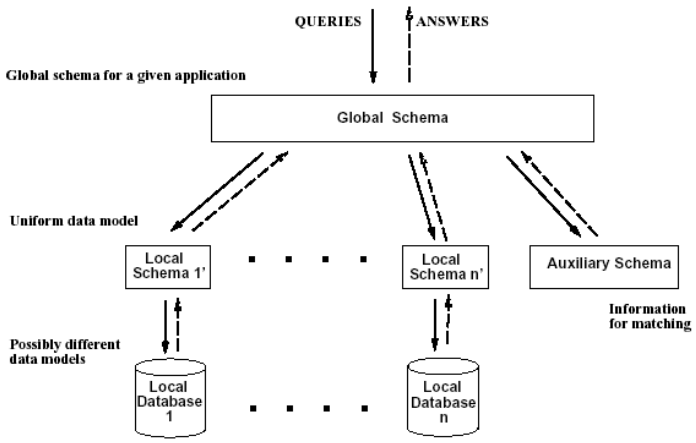
42

A first case

- The data sources have the same data model
- Adoption of a *global schema*
- The global schema will provide a
  - Reconciled
  - Integrated

view of the data sources

Architecture with a uniform data model



Designing data integration in the Multidatabase (with the same data model)

1. Source schema identification (when present)
2. Source schema reverse engineering (data source conceptual schemata)
3. Conceptual schemata integration and restructuring: conflict resolution and restructuring
4. Conceptual to logical translation (of the obtained global schema)
5. Mapping between the global logical schema and the single schemata (logical view definition)
6. After integration: query-answering through data views

45

Example (points 1, 2)

PoliRobots:

REPORT (ID, dateTime, #ofFaults)

REPORTFAULT (reportID, faultID)

FAULT (ID, description, solution, responsibleOperatorID)

UniRobots:

MESSAGE (robotID, dateTime, errorCode) *//robotIDs are R001, R002, ..., R100*

ERROR (code, description, solution, personInCharge, urgency)

Additional note: PoliRobots has only one robot

46

46

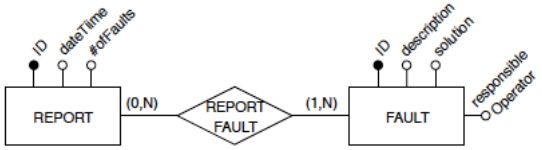
Example (points 1, 2)

PoliRobots:

REPORT (ID, dateTime, #ofFaults)

REPORTFAULT (reportID, faultID)

FAULT (ID, description, solution, responsibleOperatorID)

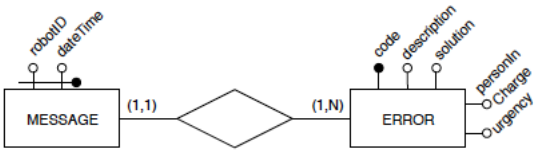


Example (points 1, 2)

UniRobots:

MESSAGE (robotID, dateTime, errorCode) *//robotIDs are R001, R002, ..., R100*

ERROR (code, description, solution, personInCharge, urgency)



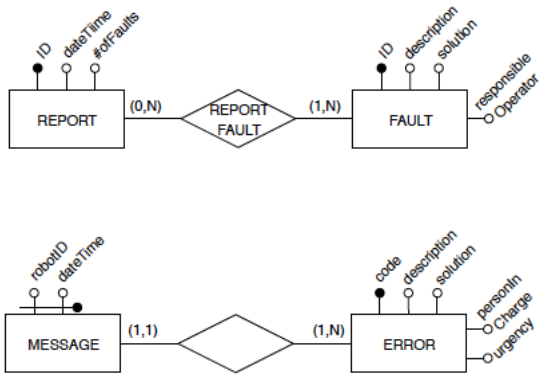
Conflict resolution and restructuring (point 3)

a. Related concept identification
b. Conflict analysis and resolution
c. Conceptual Schema integration

Related Concepts' identification

- Ex:
  - employee, clerk
  - exam, course
  - code, num
- Not too difficult if manual
- Much more difficult if automatic – we will see this later on

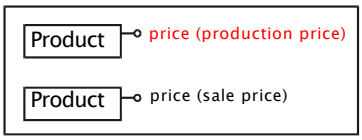
Related concept identification



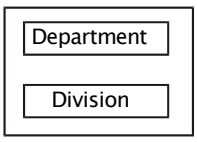
Conflict analysis

NAME CONFLICTS

- HOMONYMS



- SYNONIMS



Conflict analysis

TYPE CONFLICTS

- in a single attribute **(e.g. NUMERIC, ALPHANUMERIC, …)**

  e.g. the attribute "gender":
  - **Male/Female**
  - **M/F**
  - **0/1**
  - **In Italy, it is implicit in the "codice fiscale" (SSN)**
- in an entity type

  different abstractions of the same real world concept produce different sets of properties (attributes)

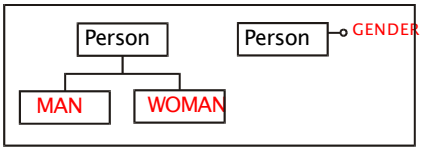
Conflict analysis

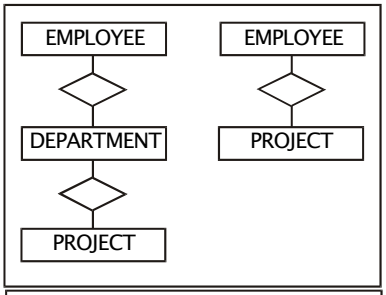
DATA SEMANTICS

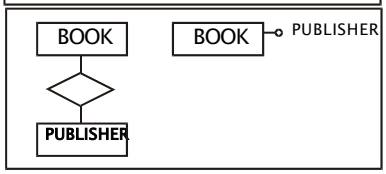
- different currencies (euros, US dollars, etc.)
- different measure systems (kilos vs pounds, centigrades vs. Farhenheit.)
- different granularities (grams, kilos, etc.)

Conflict analysis

STRUCTURE CONFLICTS

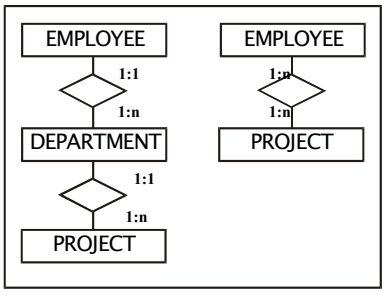






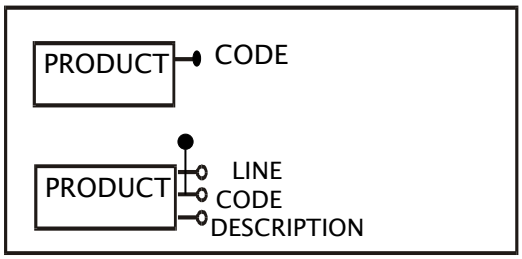
Conflict analysis

- DEPENDENCY (OR CARDINALITY) CONFLICTS

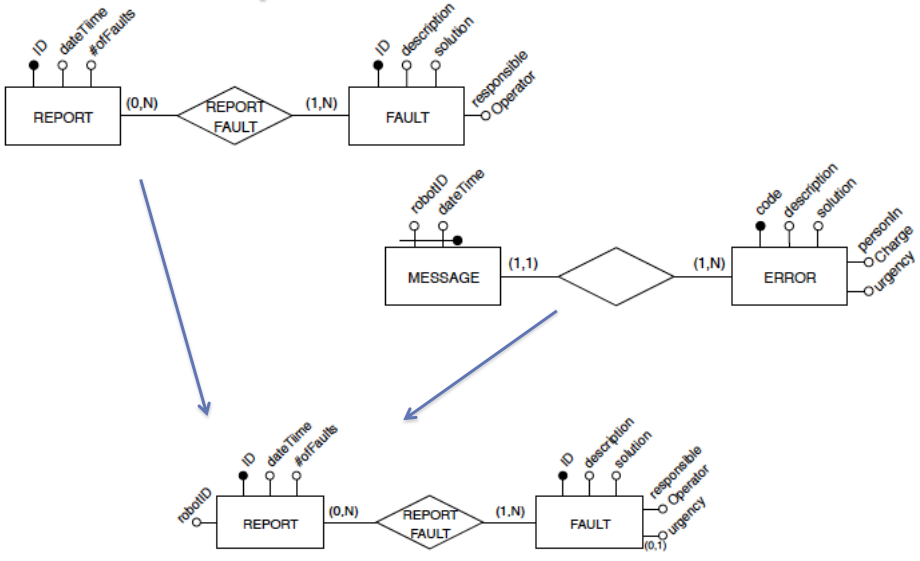


Conflict analysis

- KEY CONFLICTS



Example: UniPoliRobots



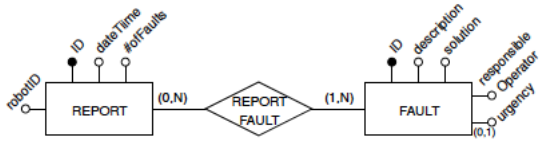
Designing data integration in the Multidatabase (with the same data model)

1. Source schema identification (when present)
2. Source schema reverse engineering (data source conceptual schemata)
3. Conceptual schemata integration and restructuring: conflict resolution and restructuring

NOW:

1. Conceptual to logical translation (of the obtained global schema)
5. Mapping between the global logical schema and the single schemata (logical view definition)
6. After integration: query-answering through data views

UniPoliRobots



REPORT(ID, RobotID, dateTime, #ofFaults)

FAULT(ID, description, solution, resp_operator, urgency*)

REPORT_FAULT(RID, FID)

```
CREATE VIEW UniPoliRobots.Report (ID, RobotID, dateTime, #ofFaults) AS
(
    SELECT KeyGenReport(ID, 'PoliRobots'), 'R101', dateTime, #of-
    Faults
    FROM PoliRobots.Report

    UNION

    SELECT KeyGenReport(robotID || dateTime, 'UniRobots'), robotID,
    dateTime, 1
    FROM UniRobots.Message
)
```

Additional note: R101 is the new ID for the PoliRobots robot

60