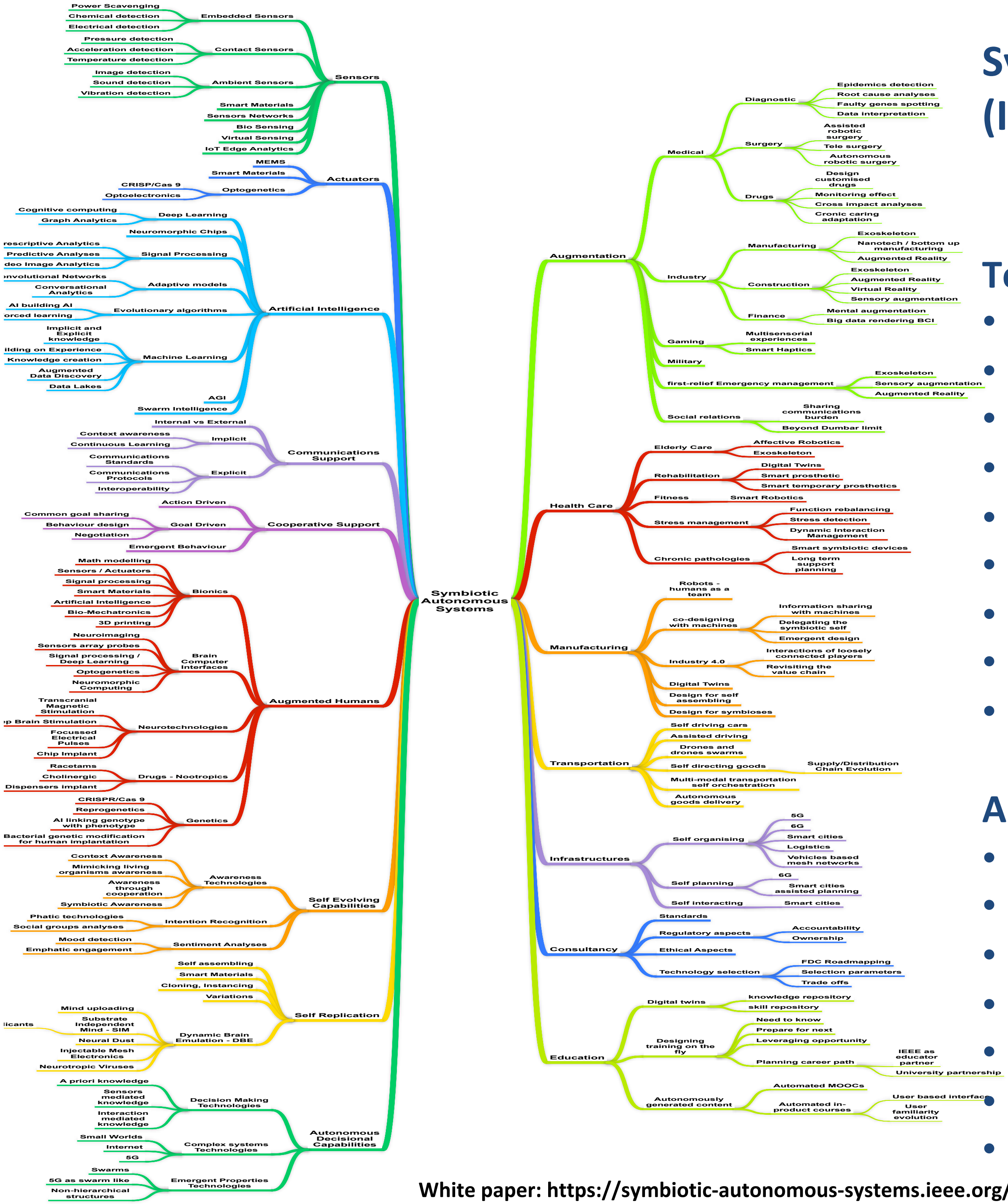
Industrial automation, communication and data management

An Overview of Data Management in 4.0

Prof. Letizia Tanca

Politecnico di Milano

Dipartimento di Elettronica, Informazione e Bioingegneria



Symbiotic Autonomous Systems (IEEE SAS initiative

Technologies (left):

- Sensors
- Actuators
- AI
- Communication Support
- Cooperative Support
- Augmented Humans
- Self-evolving Capabilities
- Self-replication
- Autonomous Decision Cap.

Applications (right):

- Augmentation
- Health care
- Manufacturing
- Transportation
- Infrastructures
- Consulting
- Education

White paper: https://symbiotic-autonomous-systems.ieee.org/images/files/pdf/sas-white-paper-final-nov12-2017.pdf

Where are the data?

Technologies:

- Existing Information Systems
- Sensors
- Actuators
- AI (Machine Learning and Deductive Systems)
- Communication Support
- Cooperative Support
- Augmented Humans
- Self-evolving Capabilities
- Self-replication
- Autonomous Decision Capabilities

Applications:

- Augmentation
- Health care
- Manufacturing
- Transportation
- Infrastructures
- Consulting
- Education

- Data are everywhere, and the information systems that already exist must be included in the loop
- Transforming DATA into KNOWLEDGE for humans and into BEHAVIOURS for machines

Program of the course (3rd part)

- **Introduction to the architectures of modern data management systems**

- **Basics of data integration:**
  - Model heterogeneity, semantic heterogeneity at the schema level, heterogeneity at the data level.

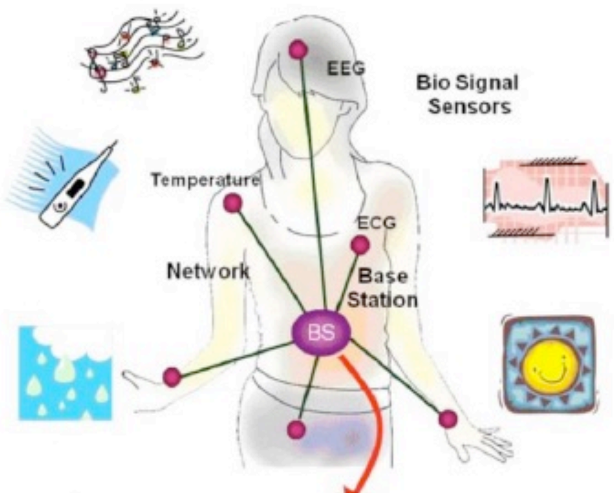
- **Dynamic data integration:**
  - The use of wrappers, mediators, meta-models, ontologies, , etc.

- **Introduction to data analysis and exploration**

- **Exercises and practical examples (Dr. Davide Azzalini)**

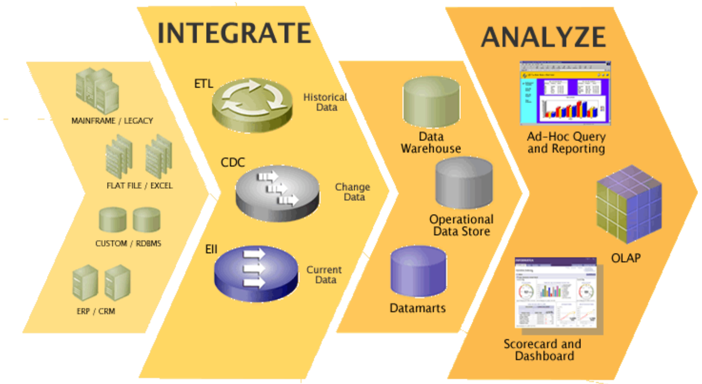
Data-related challenges in the I4.0 ecosystems

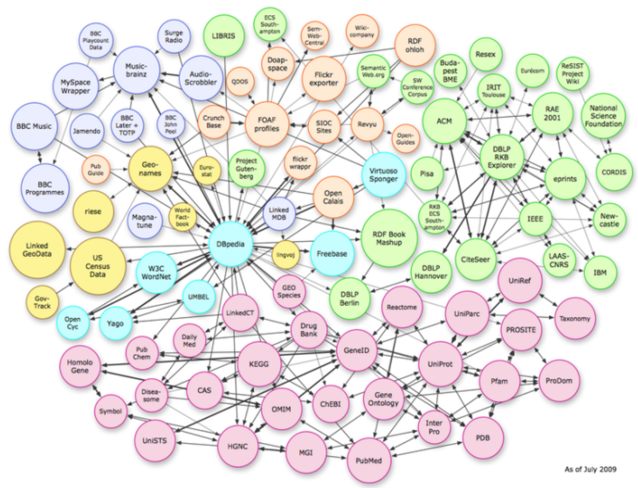
- ✓ Pre-history: focused on challenges that occur *within enterprises → Information Systems*
- ✓ The Web era:
  - scaling to a much larger number of semi- and un-structured data sources
- ✓ Nowadays:
  - Sensors and actuators form the Internet of Things and support production
  - Large scientific experiments rely on data management for progress.
  - With the massive use of social media and smart devices, people create data fragments (breadcrumbs) by interacting with services on the Web
  - User-generated content merges with the Internet of Things: users *as sensors and actuators*



The four V's of Big Data

- Volume
- Velocity
- Variety
- Veracity
- We should be able to govern:
  - data abundance
  - data and user dynamicity and mobility
  - heterogeneity, data semantics
  - incompleteness/uncertainty
  - interaction with the real-world





making sense of all this data:

→ Extract useful knowledge

VOLUME and VELOCITY

- The classical DBMSs (also distributed) are <u>transactional systems</u>: they provide a mechanism for the definition and execution of transactions
- In the execution of a transaction the ACID properties must be guaranteed
- A transaction represents the typical elementary unit of work of a Database Server, performed by an application
- New DBMS have been proposed that <u>are not transactional systems</u>

Recall: the Relational Model

attributes
(or columns)

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

tuples
(or rows)

Data Manipulation Language (DML)

- Language for accessing and manipulating the data organized by the appropriate data model
  - DML also known as Query Language - SQL is the most widely used query language
- A typical SQL query has the form:

  select $A_1, A_2, ..., A_n$
  from $r_1, r_2, ..., r_m$
  where P

  - ★ A_i represents an attribute
  - ★ R_i represents a relation
  - ★ P is a predicate.
- The result of an SQL query is *a relation*.

Transaction Management

- A *transaction* is a collection of operations that performs *a single logical function* in a database application
- The *transaction-management component* ensures that the database remains in a consistent (correct) state despite system failures (e.g., power failures and operating system crashes) and transaction failures.
- The *concurrency-control manager* controls the interaction among the concurrent transactions, to ensure the consistency of the database.

ACID

- Atomicity: A transaction is an indivisible unit of execution
- Consistency: the execution of a transaction must not violate the integrity constraints defined on the database
- Isolation: the execution of a transaction is not affected by the execution of other concurrent transactions
- Persistence (Durability): The effects of a successful transaction must be permanent

BIG DATA and the Cloud

DATA CLOUDS: ON DEMAND STORAGE SERVICES, offered on the Internet with easy access to a virtually infinite number of storage resources, computing and network

Cloud databases: support BIG DATA by means of load sharing and data partitioning

- It has been realized that it is not always necessary that a system for data management guarantees all transactional characteristics
- The non-transactional DBMS, typically offered on the Cloud, are commonly called NoSQL DBMS
- This really is not correct because the facts that a system is relational (and uses the SQL language) and that it has transactional characteristics are independent

NoSQL databases

- Provide flexible schemas
- The updates are performed asynchronously (no explicit support for concurrency)
- Potential inconsistencies in the data must be solved directly by users
- Scalability: no joins, ease of clustering
- Evolution to a “simpler” schema: key/value-based, semi/non- structured
- Object-oriented friendly
- Caching easier (often embedded)
- Easily evolved to live replicas and node addition, made possible by the simplicity of repartitioning of data
- DO NOT support all the ACID properties

DATA MODELS

3 main categories:

- Key –Value
- Document –based
- Column –family

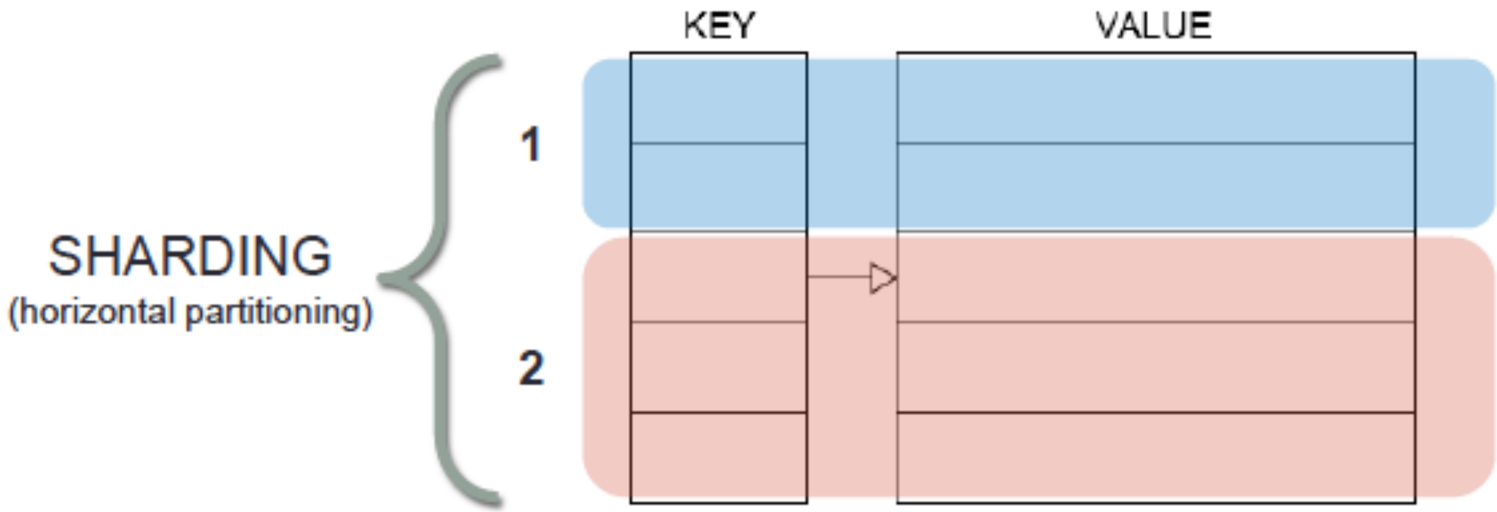
Another category, *graph-based* (not treated here), a separate evolutionary path from the other categories.They are mainly oriented on modeling relationships

Key -Value

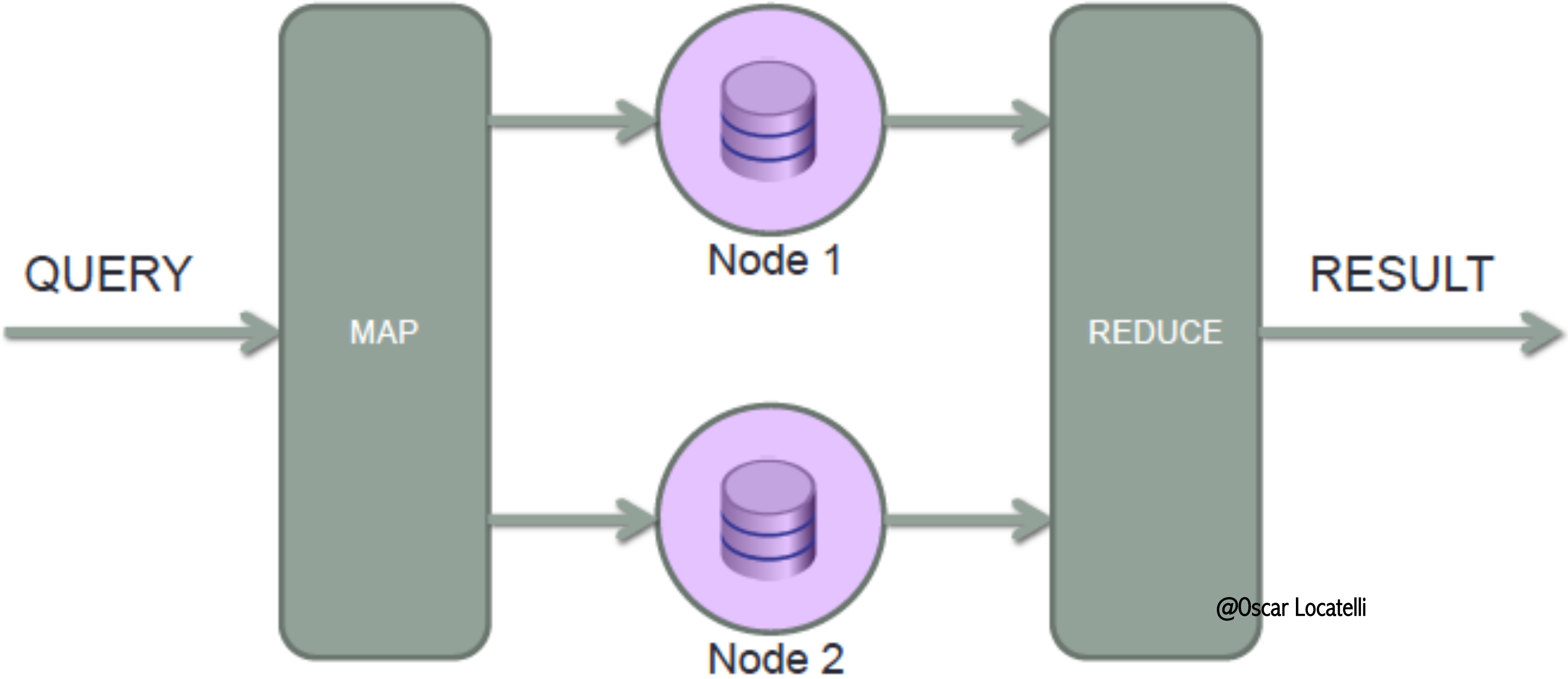
- Classical reference model of NoSQL systems
- Key: single or compound
- Value: blob, " opaque”
- Querying = find by key
- No schema (a dictionary)
- Standard APIs: get / put / delete

Key -Value : Scaling on multiple nodes

- Joins limit scalability
- No relationships in the database → easier to scale!
- Decoupled and denormalized entities are ‘self-contained' .
- We can move them to different machines without having to worry about “neighbourhood”!
- Sharding (horizontal partitioning)

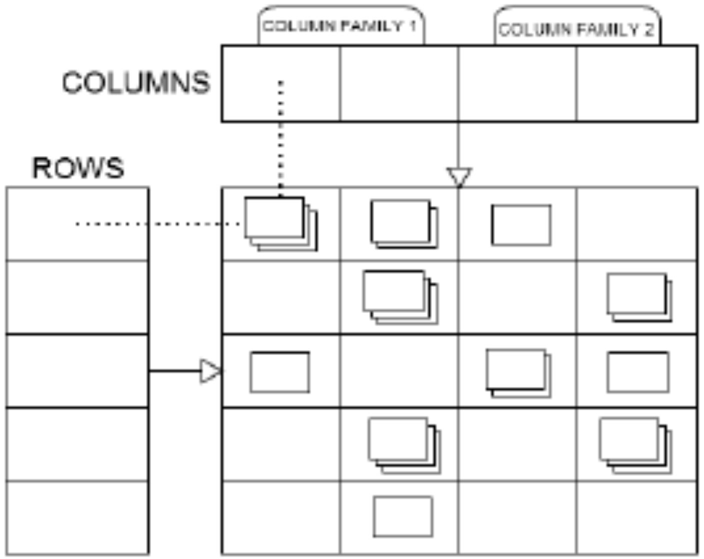


Map Reduce

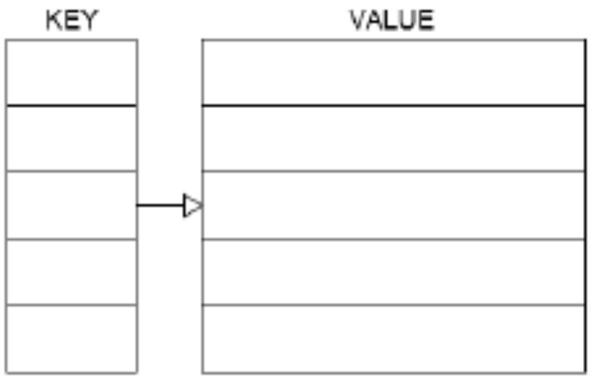


Further Models

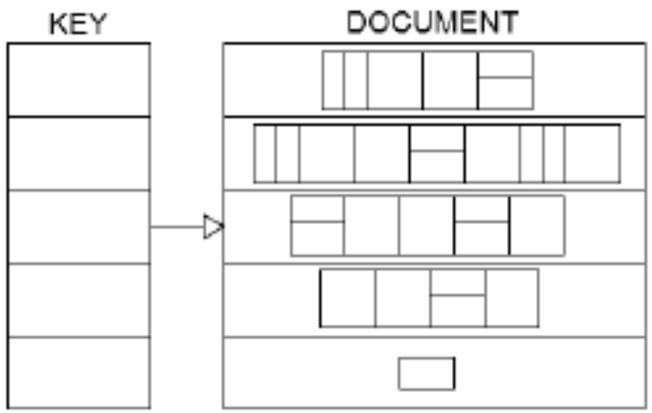
Column oriented



Key-Value

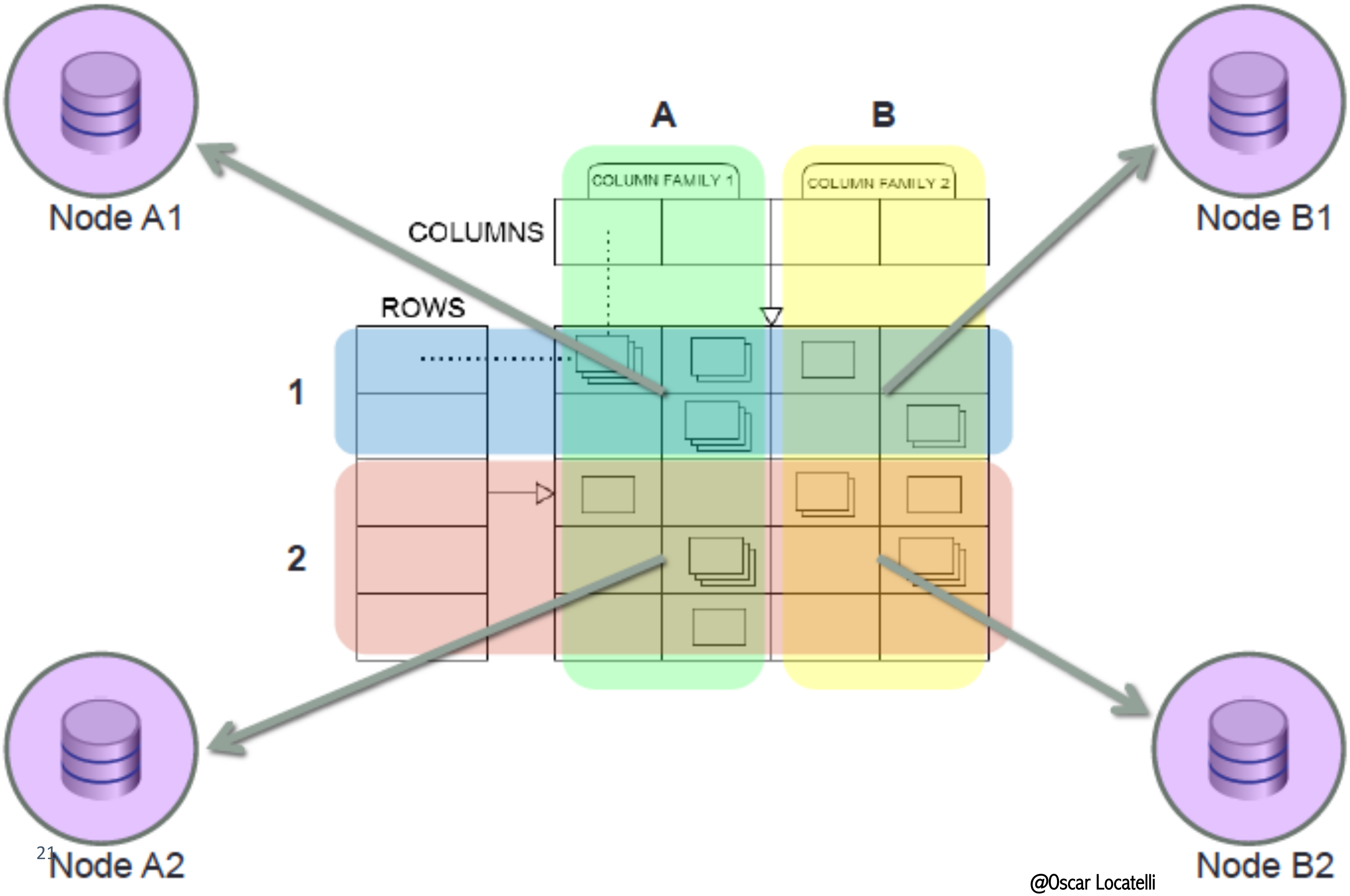


Document based



@Oscar Locatelli

Column-family: maximum scaling



@Oscar Locatelli

The CAP theorem

A data management system shared over the network (cloud, networked shared-data system) can guarantee at most two of the following properties:

- Consistency (C): all nodes see the same data at the same time
- Availability (A): a guarantee that every request receives a response about whether it was successful or failed
- (tolerance to) Partitions (P): the system continues to operate despite arbitrary message loss or failure of part of the system

NOTES:

- In ACID, the C means that a transaction preserves all the database constraints, while the C in CAP refers only to copy consistency, a strict subset of ACID consistency.
- From this we can understand why in more traditional applications, such as banking or accounting, or bookings etc. these systems may be catastrophic

Different products for different objectives

- Column-family for Enterprise Big Data and possibly sensor data: maximum performance and scalability - Amazon DynamoDB , Google BigTable and all their derivatives (Voldemort , Cassandra , HBase , Hypertable)
- Document Stores for Document Management or RDBMS replace (with caution!!!): MongoDB (write heavy) RavenDB (read -heavy), BigCouch (both not discussed here)

SOME FAMOUS IMPLEMENTATIONS

- Amazon DynamoDB
  - Key – value
  - CAP : AP - guarantees Availability and Partition tolerance, relaxing Consistency
  - auto – sharding
  - P2P networks
  - It aims to eliminate the job of the database administrator
  - Project Voldemort, SimpleDB
- Google BigTable
  - Column- oriented, on the Google BigTable paper serves as the foundation to the NoSQL Column- based data –model
  - CAP: CP - if there is a network partition Availability is lost, but ‘strict' consistency may be required
  - auto - sharding , conflict resolution manual, no P2P

SOME FAMOUS IMPLEMENTATIONS (II)

- Hypertable and Hbase
  - Implementations of BigTable (built on Google File System)
  - Both Apache Hadoop (framework for distributed applications based on map - reduce)
  - Interface Thrift , REST and APIs for various languages
  - HBase extensible (coprocessors) , Hypertable most powerful
- Cassandra
  - Free from the Apache Foundation, Unix -like and Windows
  - Super -Column –family
  - CAP : AP consistency with configurable auto - sharding , automatic conflict resolution
  - Combines the P2P with the data -model of BigTable
  - Transactions lock- free
  - Key names only on rows and columns
  - Also on Hadoop

SOME FAMOUS IMPLEMENTATIONS (III)

- MongoDB
  - Embeddable only in a C++ process, with LGPL license
  - Document –based
  - CAP: CP
  - auto - sharding with configurable strategy
  - static and automatic Indices created synchronously with the write
  - No transactions but atomic operations, and patterns for creating 2-phase commit or other politics
  - Query Task executed in Map-Reduce or otherwise distributed systems
  - In-place update of document attributes
  - Optimized for write- heavy (updates on index update and maintain consistency)
  - APIs for various languages ORM-like

→ It is the most widely used and known, excellent performance and excellent documentation

SOME FAMOUS IMPLEMENTATIONS (IV)

- CouchDB
  - Document
  - CAP: AP, Multi-Version Concurrency Control , strict consistency of master , slave where appropriate
  - View materialized on the first read and updated with map -reduce algorithm written in Javascript, projection , sort and calculations
  - Transactions lock free
  - Write- heavy, Read-heavy
  - CouchDB is also a WebServer , can do application hosting HTML5 + JavaScript that are treated as documents (then synchronized between multiple databases, easy load-balancing)
  - Couchbase, CouchDB offers Memcached + GeoIndex
  - CouchDB is the basis of the synchronization service Ubuntu One

Bibliography on NoSQL

BOOK :

Tamer Ötzsu M., Valduriez P. – Principles of *Distributed Database Systems: 3rd ed.* - Springer, 2011

More references

•Abadi Daniel J. - *Data Management in the Cloud: Limitations and Opportunities* - IEEE Data Engineering Bulletin, Vol. 32 No. 1, March 2009

http://sites.computer.org/debull/A09mar/A09MAR-CD.pdf#page=5

•Dean J., Ghemawhat S. – *MapReduce: A Flexible Data Processing Tool* - CACM, Vol.53, n. 1, pp. 72-77, 2010

•Foster I., Yong Zhao, Raicu I., Lu S - *Cloud Computing and Grid Computing 360-Degree Compared* - Grid Computing Environments Workshop 2008, pp. 1-10, 2008

http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4738445

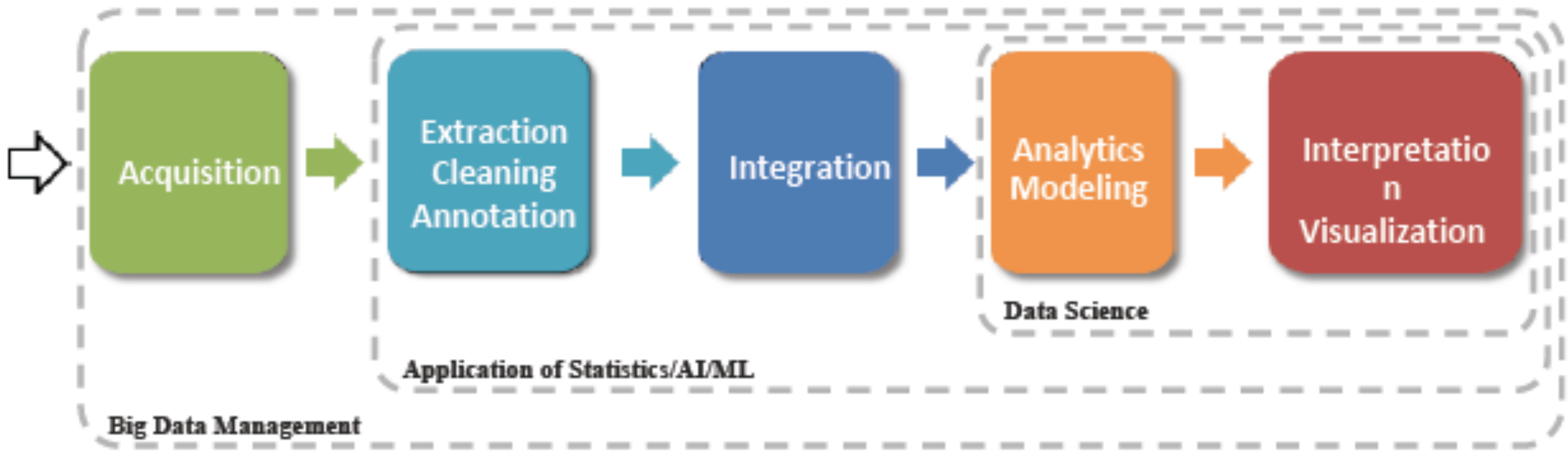
More than Volume and Velocity

Extraction of synthetic and useful knowledge from the data:

- ***Massive data integration:*** People and enterprises need to integrate data and the systems that handle those data: Relational DBMSs and their extensions, legacy data and legacy DBMSs, structured or unstructured data from people and devices
- ***Massive data analysis and exploration:*** data analysis and data mining research focuses on studying algorithms and techniques to find interesting patterns representing implicit knowledge stored in massive data repositories, useful to generate concise models of the analyzed data.
- ***Data warehousing:*** A single, complete and consistent store of data obtained from a variety of different sources for analysis in a business context.[Barry Devlin]
- **Knowledge representation and reasoning:** using conceptual models and ontologies, formal specifications allows for use of a common vocabulary for automatic knowledge sharing; using reasoning services, which allow some forms of deduction and inference.

Motivation:
the reality behind each data extraction (analysis) task

- **The actual implementation of the Data Analysis (ML, statistics, Data Mining, and obviously querying…) 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**



Credits: Beng Chin OOI – VLDB 2018 / A. Labrinidis, H. V. Jagadish VLDB 2012

The Data Integration problem

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

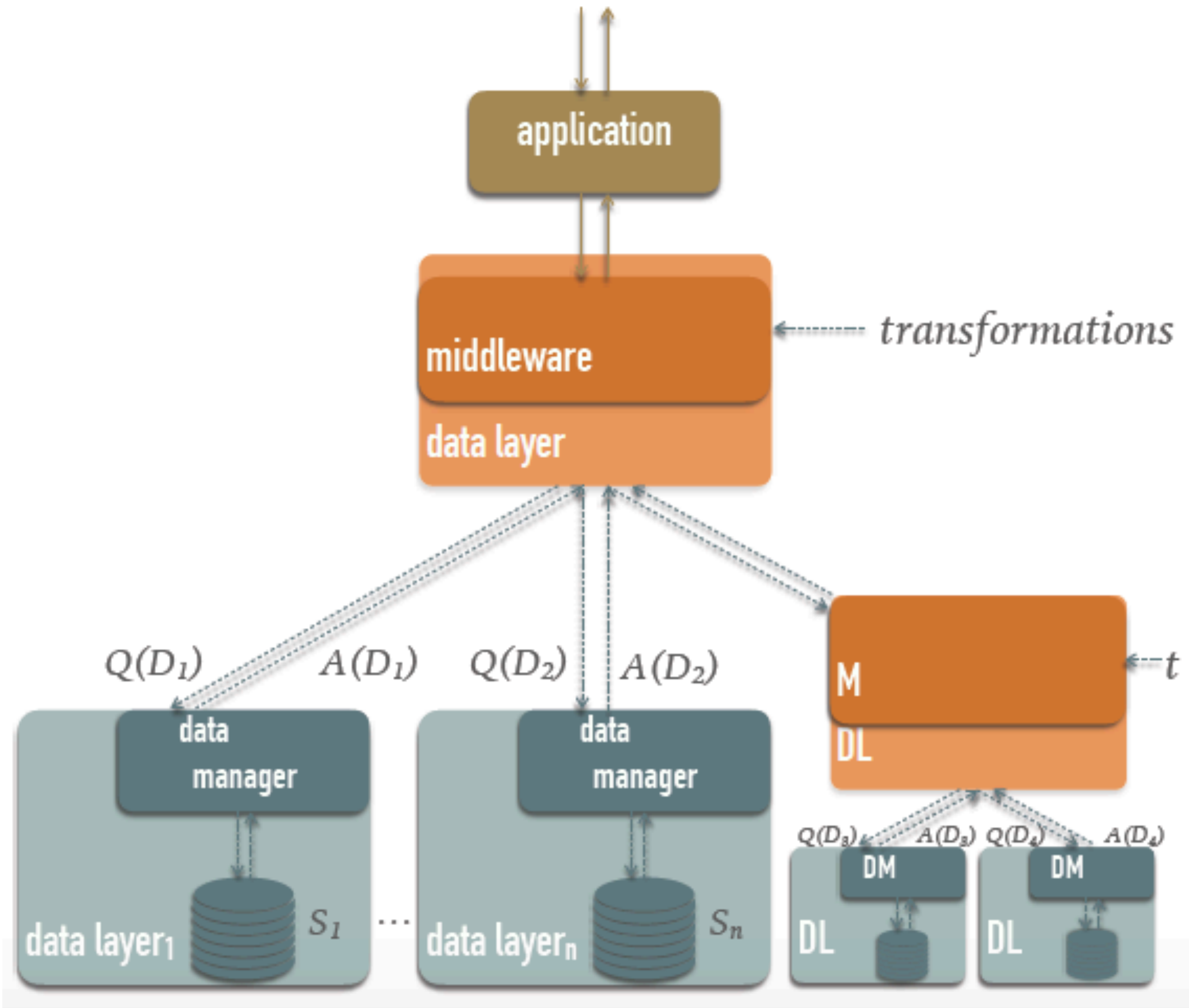
→ Detecting correspondences between similar concepts that come from different sources, and conflict solving

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)

A general framework for Data Integration



(Alvaro Fernandez, EDBT Summ. School 2017)

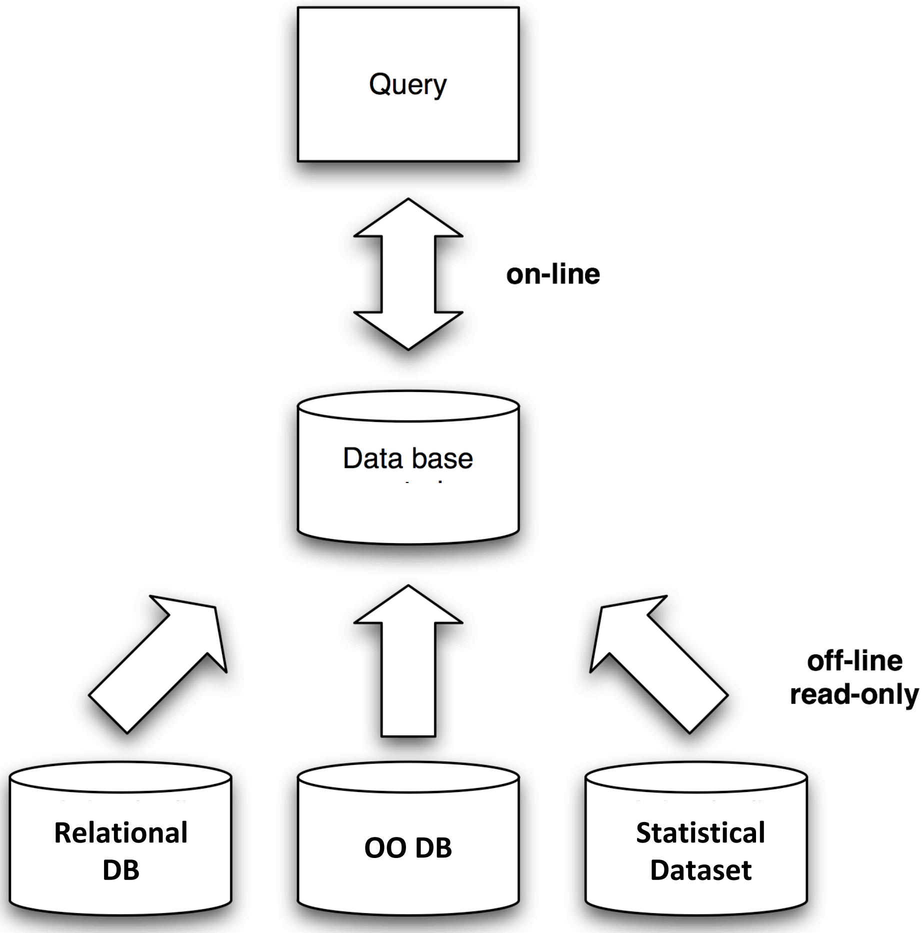
Relevant Ways of Integrating Database Systems

1. Use a materialized data base (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)

Materialized

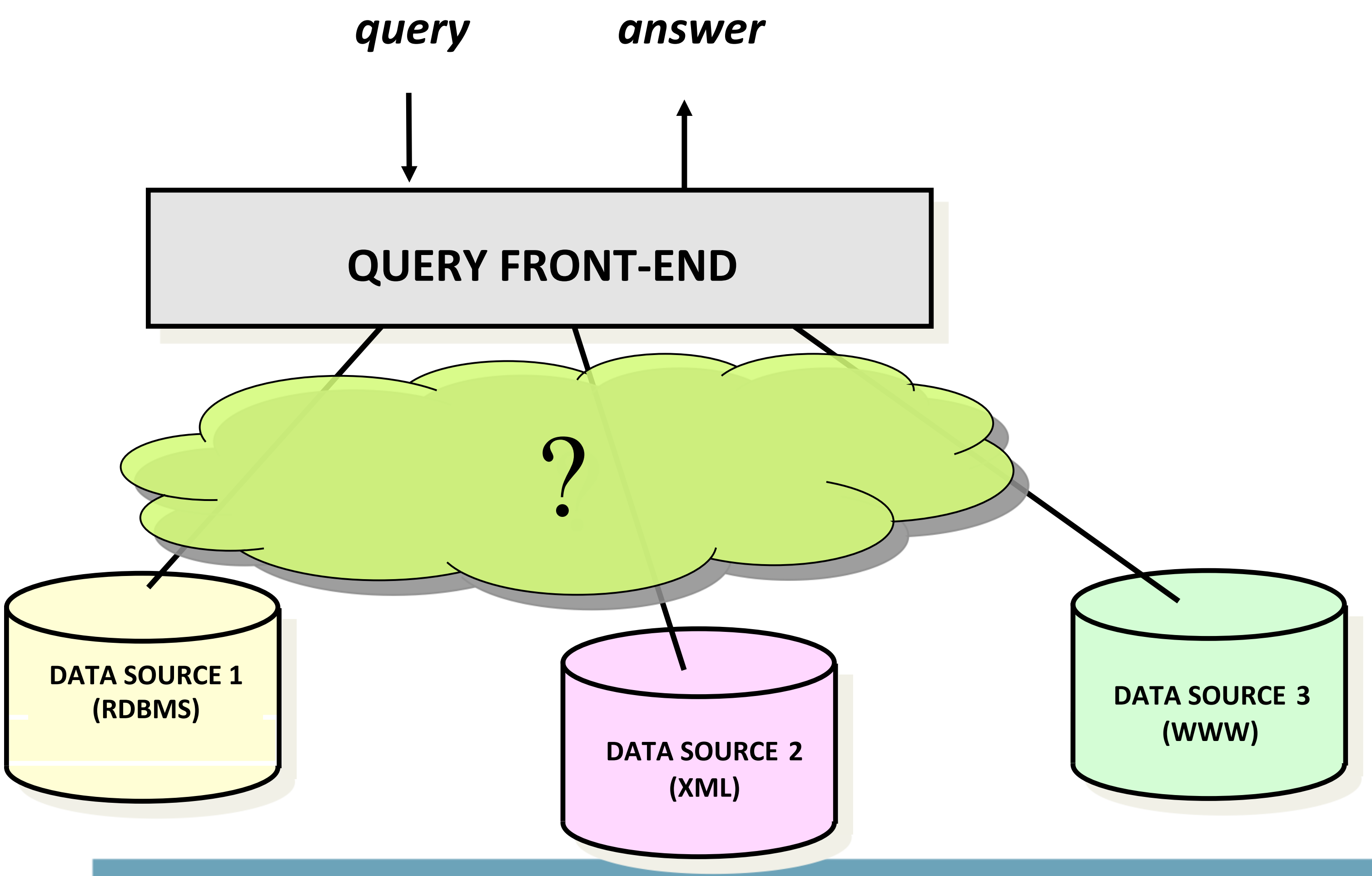


Materialized integration is typically adopted for Data Warehouses

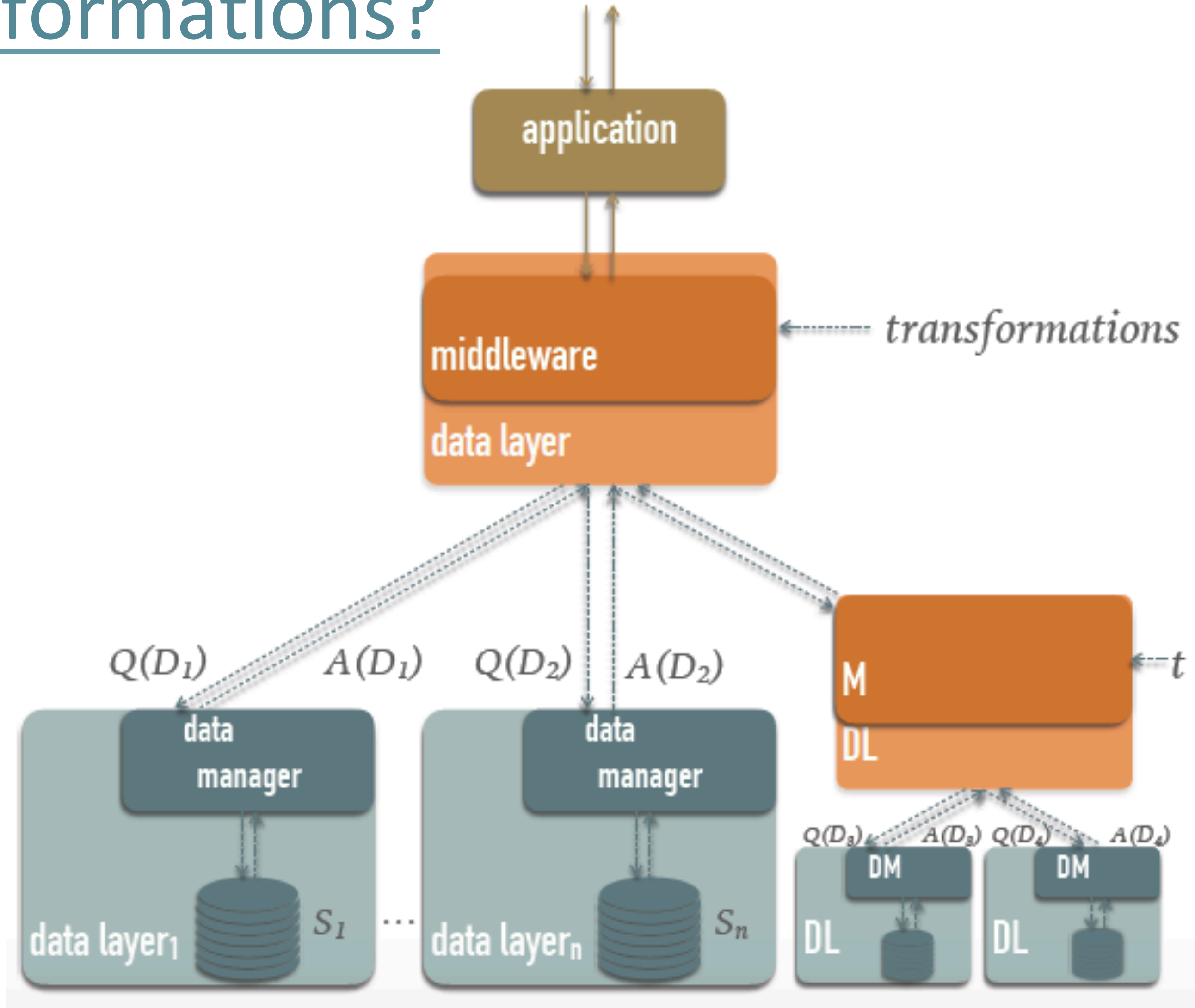
- Large common systems known as warehouses, and the software to access, scrape, transform, and load data into warehouses, became known as extract, transform, and load (ETL) systems.
- In a dynamic environment, one must perform ETL periodically (say once a day or once a week), thereby building up a history 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)

Virtual

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.



Be it virtual or materialized, the fundamental problem to solve is: <u>what transformations?</u>

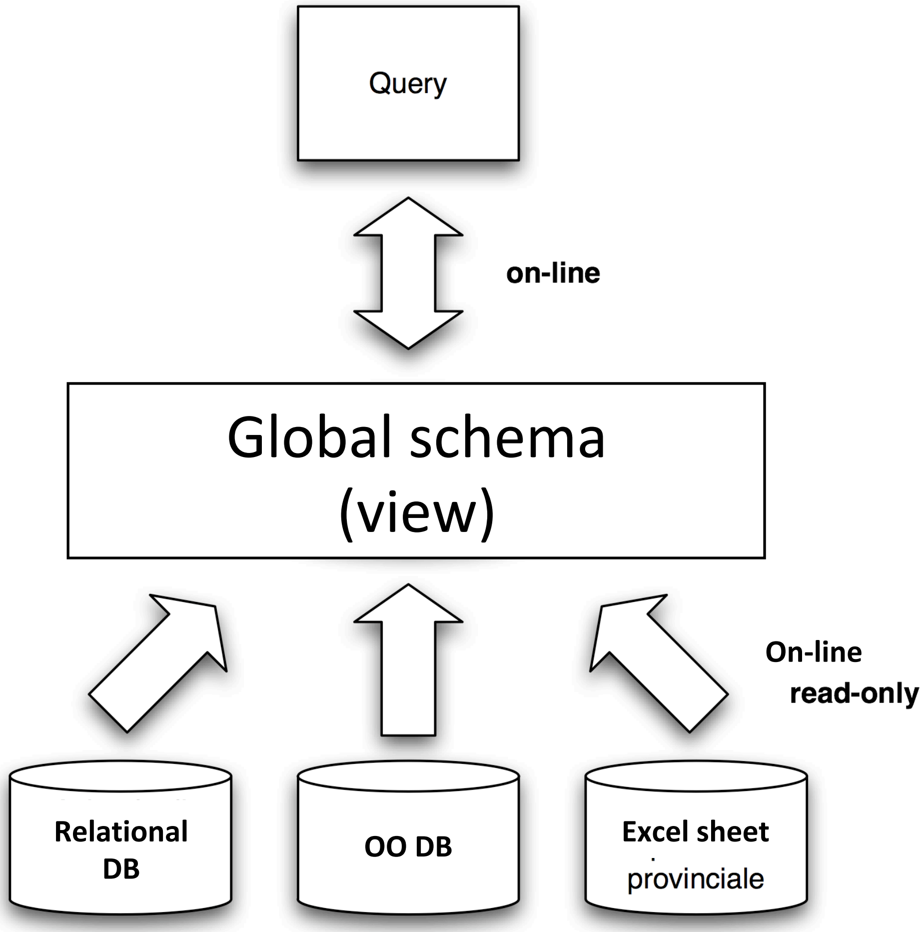


(Alvaro Fernandez, EDBT Summ. School 2017)

The steps of Data Integration

Schema Reconciliation	(If the sources have a schema) **mapping the data structure**
Record Linkage	**data matching based on the same real-world entity**
Data Fusion	**reconciliation of non-identical content**

The simplest case of virtual integration



Given a target schema T, a set S of source schemas, define a set M of mappings relating T to the elements in S, so that and a query Q(T) against the target schema can be evaluated using M to transform it into queries to the sources

Views

- Also called external schemata

- Sintax:

  create view *ViewName* [(*AttList*)] **as** *SQLquery*

  [**with** [**local** | **cascaded**] **check option**]

Schema-level integration

a. Related concept identification
b. Conflict analysis and resolution
c. Conceptual Schema integration and restructuring
d. Translation into the logical model (tables, ...)

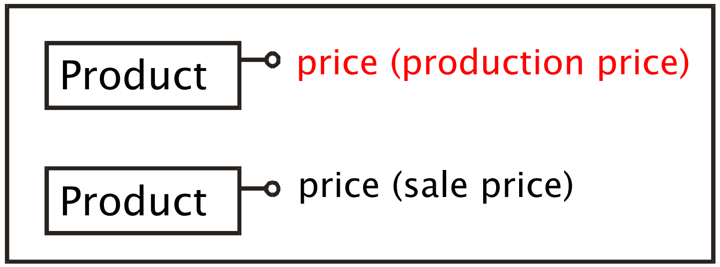
Related Concepts' identification

- Ex:
  - employee, clerk
  - exam, course
  - code, num
- Not too difficult if manual
- Very difficult if automatic – this is the extreme case
- Manual: translate all source schemas into a single conceptual representation model, e.g. Entity-Relationship

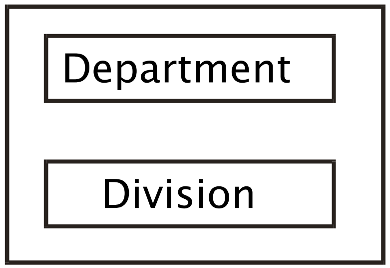
Conflict analysis

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)

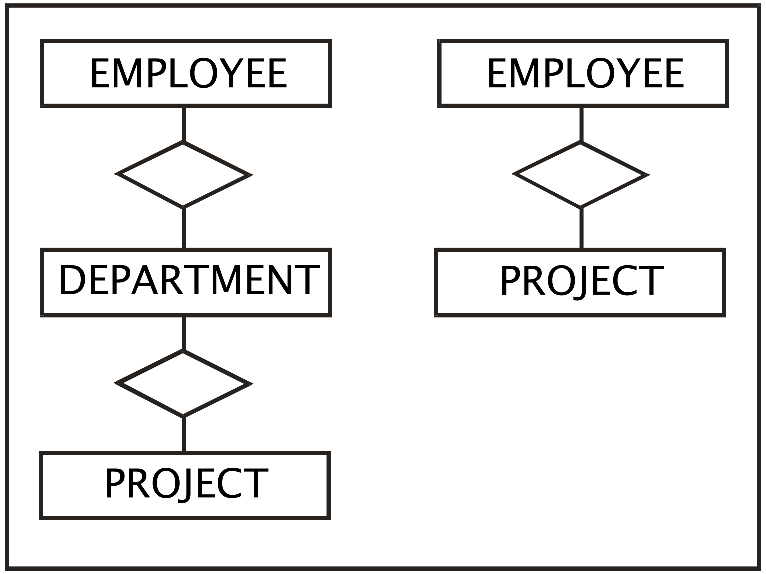
Conflict analysis

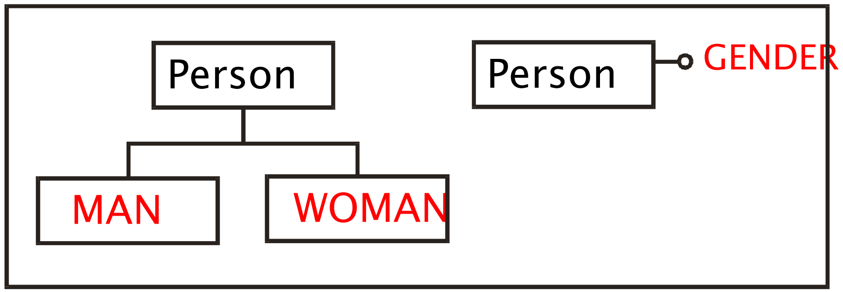
DATA SEMANTICS

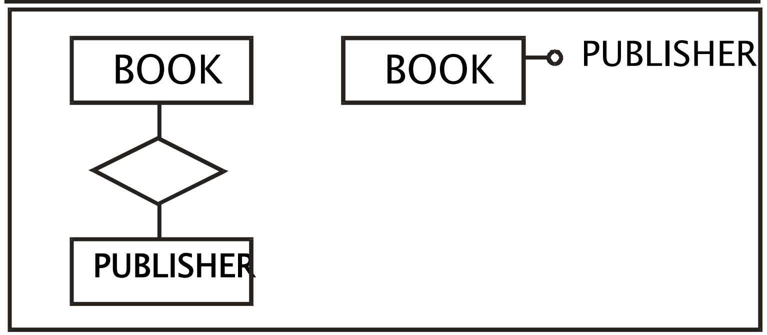
- different currencies (euros, US dollars, etc.)
- different measure systems (kilos vs pounds, centigrades vs. Farhenheit.)
- different granularities (grams, kilos, etc.)

Conflict analysis

STRUCTURE
CONFLICTS

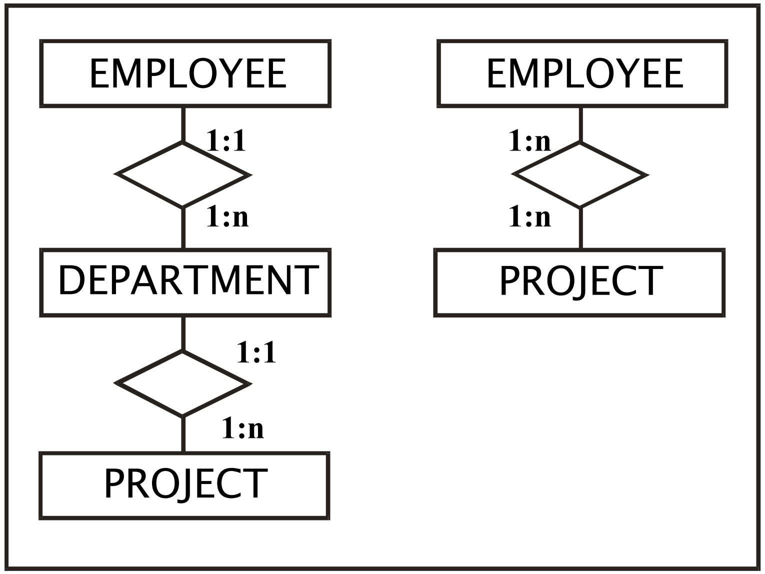






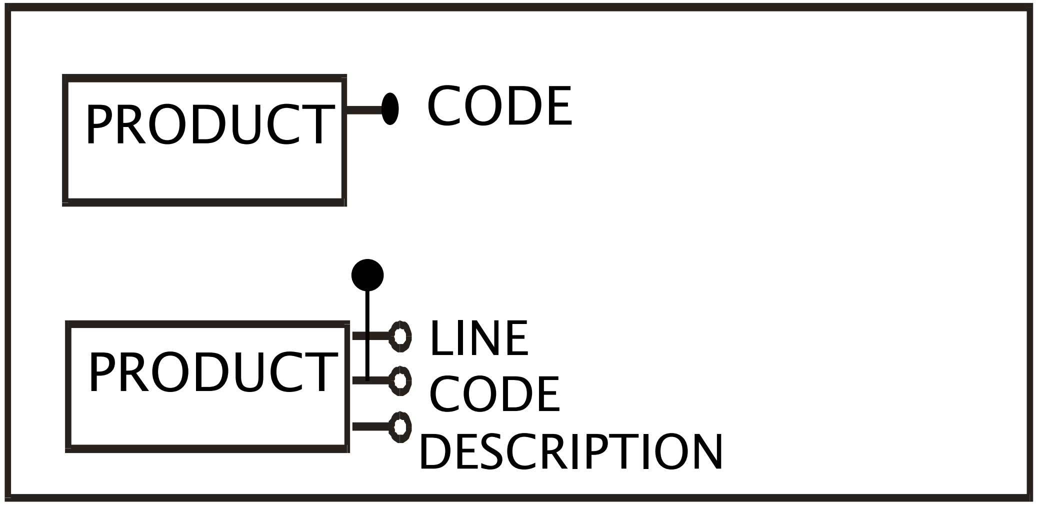
Conflict analysis

- DEPENDENCY (OR CARDINALITY) CONFLICTS



Conflict analysis

- KEY CONFLICTS



The steps of Data Integration

Schema Reconciliation

(If the sources have a schema)
mapping the data structure

Record Linkage

data matching based on the same real-world entity

Data Fusion

reconciliation of non-identical content

Schema Integration

- Global schema conceptual design:
  - conflict resolution ✓
  - restructuring ✓
  - production of a new DB schema which expresses (as much as possible) the same semantics as the schemata we wanted to integrate ✓
- Production of the transformations (views) between the original schemata and the integrated one: $V_1(DB)$, $V_2(DB)$,… $V_3(DB)$

A first, easy case

- The data sources have <u>the same data model</u>
- Adoption of a *global schema*
- The global schema will provide a
    - Reconciled
    - Integrated
    - Virtual

view of the data sources

Schema integration example (GAV)

SOURCE 1

Product(<u>Code</u>, Name, Description, Warnings, Notes, CatID)
Category(<u>ID</u>, Name, Description)
Version(<u>ProductCode</u>, <u>VersionCode</u>, Size, Color, Name, Description, Stock, Price)

SOURCE 2

Product(<u>Code</u>, Name, Size, Color, Description, Type, Price, Q.ty)
Type(<u>TypeCode</u>, Name, Description)

note: here we do not care about data types…

SOURCE 1

Product(Code, Name, Description, Warnings, Notes, CatID)

Version(ProductCode, VersionCode, Size, Color, Name, Description, Stock, Price)

SOURCE 2

Product(Code, Name, Size, Color, Description, Type, Price, Q.ty)

GLOBAL SCHEMA

```
CREATE VIEW GLOB-PROD AS
SELECT Code AS PCode, VersionCode
    as VCode, Version.Name AS Name,
    Size, Color, Version.Description as
    Description, CatID, Version.Price,
    Stock
FROM SOURCE1.Product,
    SOURCE1.Version
WHERE Code = ProductCode
UNION
SELECT Code AS PCode, null as
    VCode, Name, Size, Color,
    Description,Type as CatID, Price,
    Q.ty AS Stock
FROM SOURCE2.Product
```

Query processing in GAV

QUERY OVER THE GLOBAL SCHEMA

```
SELECT PCode, VCode, Price, Stock
FROM GLOB-PROD
WHERE Size = “V” AND Color = “Red”
```

The transformation is easy, since the combination operator is a UNION → **push selections through union!!**

```
SELECT Code, VersionCode, Version.Price, Stock
FROM SOURCE1.Product, SOURCE1.Version
WHERE Code = ProductCode AND Size = “V” AND Color = “Red”
UNION
SELECT Code, null, Price, Q.ty
FROM SOURCE2.Product
WHERE Size = “V” AND Color = “Red”
```

GAV method

- The global schema is formed of *views over the data sources*
- Mapping 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

The other possible ways
(not studied here)

LAV (Local As View)

- The global schema has been designed independently of the data source schemata
- The relationship (mapping) between sources and global schema is obtained by defining each data source as a view over the global schema

GLAV (Global and Local As View)

- Mixing GAV with LAV

Next problem: various kinds of heterogeneity

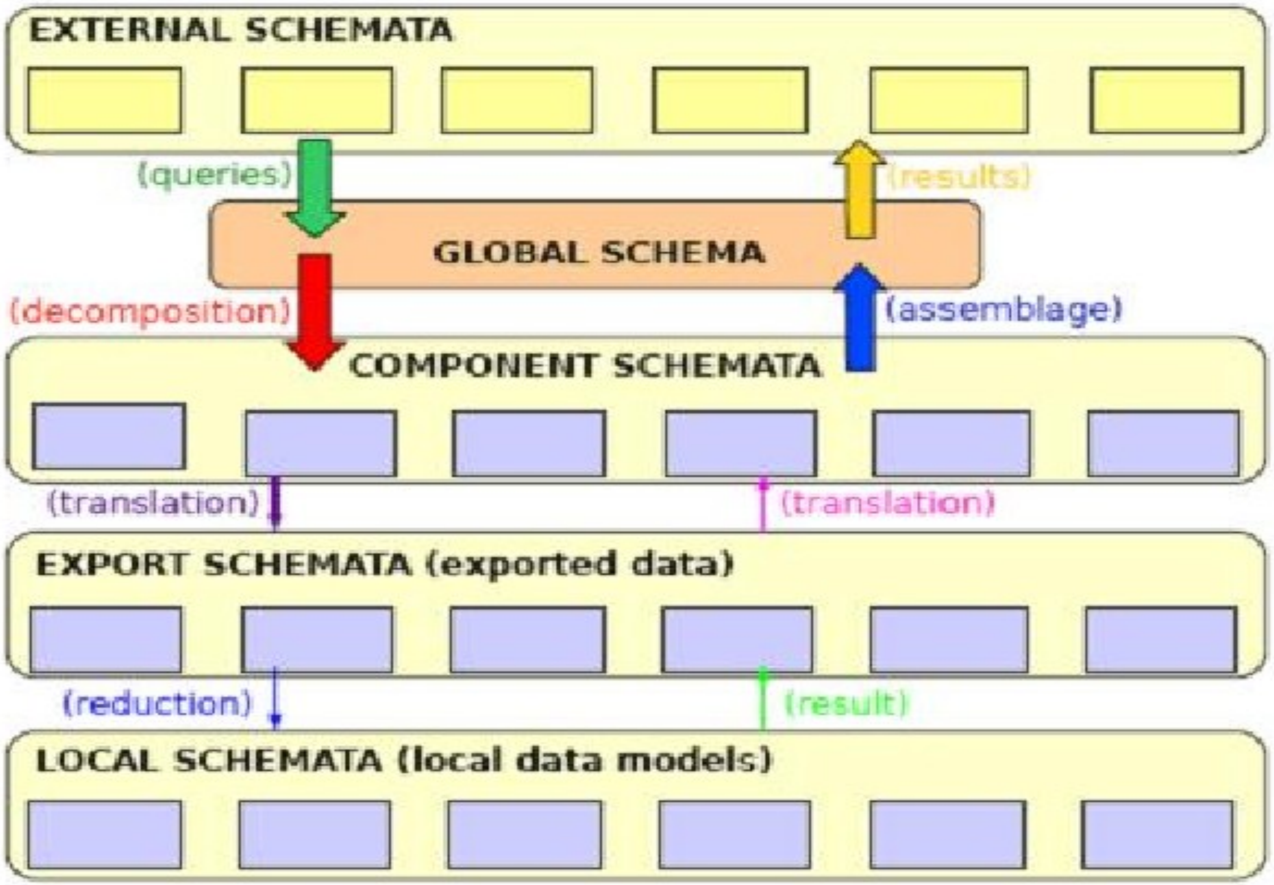
- Same data model, different systems e.g. relational (Oracle, Sybase, DB2…) → *technological heterogeneity*
- Different data models, e.g. relational, Obj.Oriented → *model and language heterogeneity*
- Semi- or unstructured data (HTML, NoSQL, XML, multimedia, sensors…) → *again model heterogeneity, but including non-structured data models*

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

Data integration in the MULTIDATABASE



A new element in this figure: WRAPPERS (translators)

- They convert queries into queries/commands which are understandable for the specific data source
  - they can even extend the query possibilities of a data source
- They convert query results from the source format to a format which is understandable for the application

Design steps

1. Reverse engineering (i.e. production of the conceptual schema)
2. Conceptual schemata integration
3. Choice of the target logical data model and translation of the global conceptual schema
4. Definition of the language translation (wrapping)
5. Definition of the data views (as usual)

Recall the steps of Data Integration

Schema Reconciliation — (If the sources have a schema) **mapping the data structure**

Record Linkage — **data matching based on the same real-world entity, a.k.a. entity resolution**

Data Fusion — **reconciliation of non-identical content**

Record linkage: detect the data referring to the same real entity

- Be there a schema or not, we may have inconsistencies in the data
- At query processing time, when a real world object is represented by instances in different databases, they may have different values

SSN	NAME	AGE	SALARY
234567891	Ketty	48	18k

SSN	NAME	AGE	SALARY
234567891	Ketty	48	25k

EXAMPLE

SSN	NAME	AGE	SALARY	POSITION
123456789	JOHN	34	30K	ENGINEER
234567891	KETTY	27	25K	ENGINEER
345678912	WANG	39	32K	MANAGER

SSN	NAME	AGE	SALARY	PHONE
234567891	KETTY	25	20K	1234567
345678912	WANG	38	22K	2345678
456789123	MARY	42	34K	3456789

Some data in these two tables clearly represent the same people

Data Fusion, aka Entity Resolution

Inconsistency may depend on different reasons:

- One (or both) of the sources are incorrect
- Each source has a correct but partial view, e.g. databases from different workplaces → the full salary is the sum of the two
- For example, the correct value may be obtained as a resolution function of the original ones

  (maybe: $1*value_1 + 0*value_2$)

RESOLUTION FUNCTION: EXAMPLE

SSN	NAME	AGE	SALARY	POSITION
123456789	JOHN	34	30K	ENGINEER
234567891	KETTY	27	25K	ENGINEER
345678912	WANG	39	32K	MANAGER

SSN	NAME	AGE	SALARY	PHONE
234567891	KETTY	25	20K	1234567
345678912	WANG	38	22K	2345678
456789123	MARY	42	34K	3456789

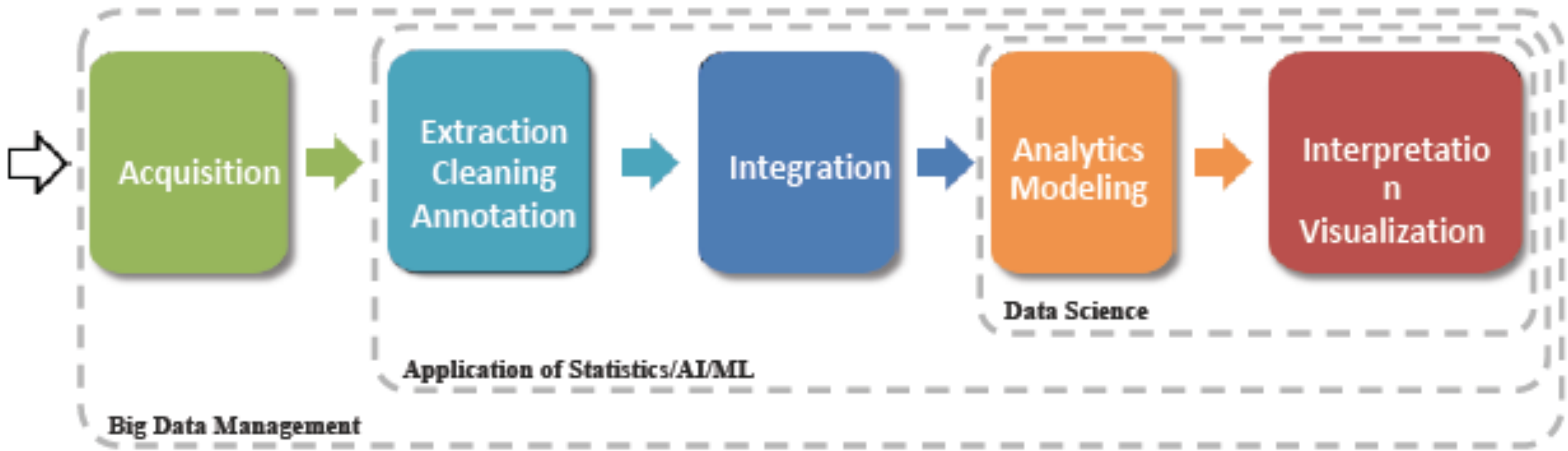
SSN	NAME	AGE	SALARY	POSITION	PHONE
123456789	JOHN	34	30K	ENGINEER	NULL
234567891	KETTY	27	45K	ENGINEER	1234567
345678912	WANG	39	54K	MANAGER	2345678
456789123	MARY	42	34K	NULL	3456789

R=MAX_AGE, SUM_SALARY (R1 OuterJoin R2)

Now we are ready for a digression
on Data Quality

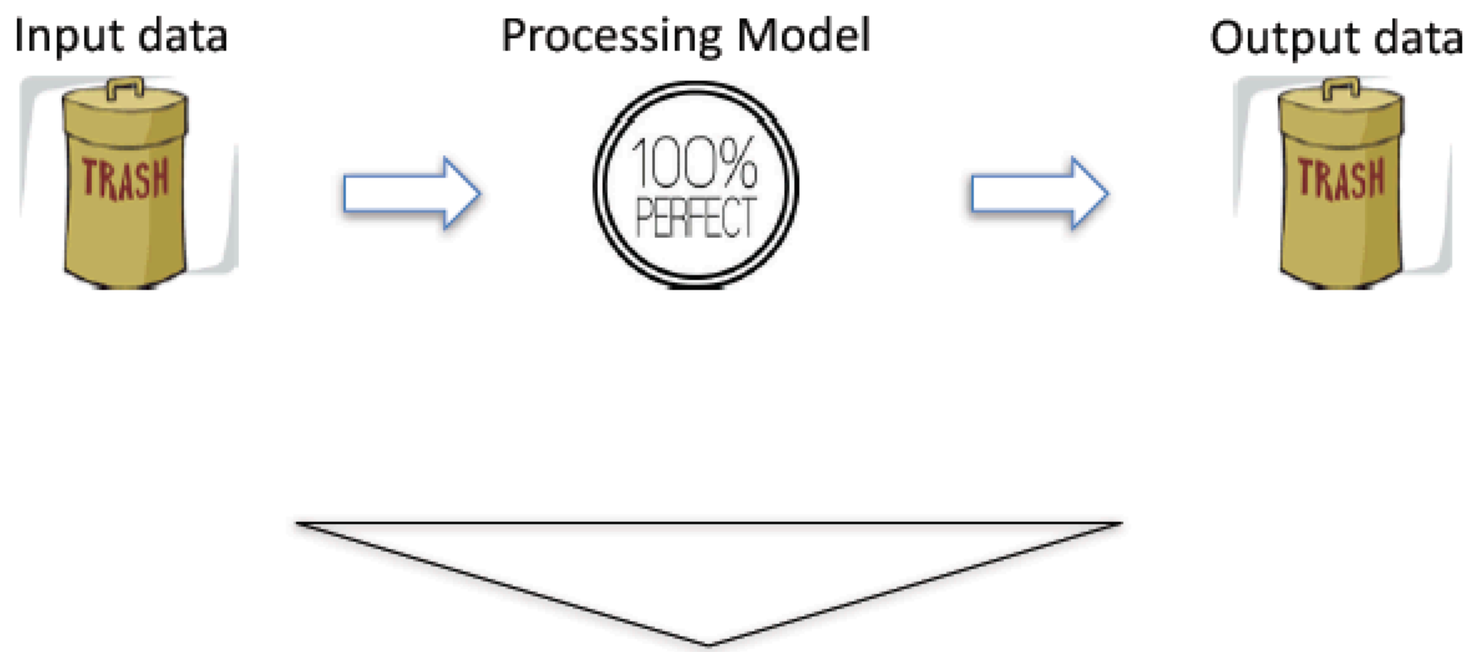
Recall the motivation:
the reality behind each data extraction (analysis) task

- The actual implementation of the Data Analysis (ML, statistics, Data Mining, and obviously querying…) 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



Credits: Beng Chin OOI – VLDB 2018 / A. Labrinidis, H. V. Jagadish VLDB 2012

The GIGO (Garbage In – Garbage Out) phenomenon



Errors (and not only) affect decisions

"Fast is fine but accuracy is everything"

...even if it is a small error...you can have the snowball effect

Causes for a poor quality

- *Historical changes*: the importance of data might change over time
  - Example: the birthdates of customers for a financial institution was not relevant in the past
- *Data usage*: data relevance depends on the process in which data are used
  - Example: operational and decisional process
  - Example: purchase of stocks by a financial institution. Purchase price and number of stocks must be correct, personal data can be affected by some errors while the customer job is not relevant from an operational perspective (but useful in decisional processes)
- *Corporate Mergers*: data integration might cause some difficulties
- *Privacy*: data are protected by privacy rules and thus it is difficult to find data to correct and its own db.
- *Data enrichment*: it might be dangerous to enrich internal data with external sources.

Recall:

VARIETY: Various types of heterogeneity among several data collections to be used together

1. Different platforms, data models at the participating datasets, interaction languages
2. Different conceptual representations (schemas) and different values for the same info (instance) due to errors or to different knowledge
3. Dependencies exist among datasets, data and applications

VERACITY: Data Quality is the most general term to represent:

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

SEMISTRUCTURED DATA

For these data there is some form of structure, but it is not as

- Prescriptive
- Regular
- Complete

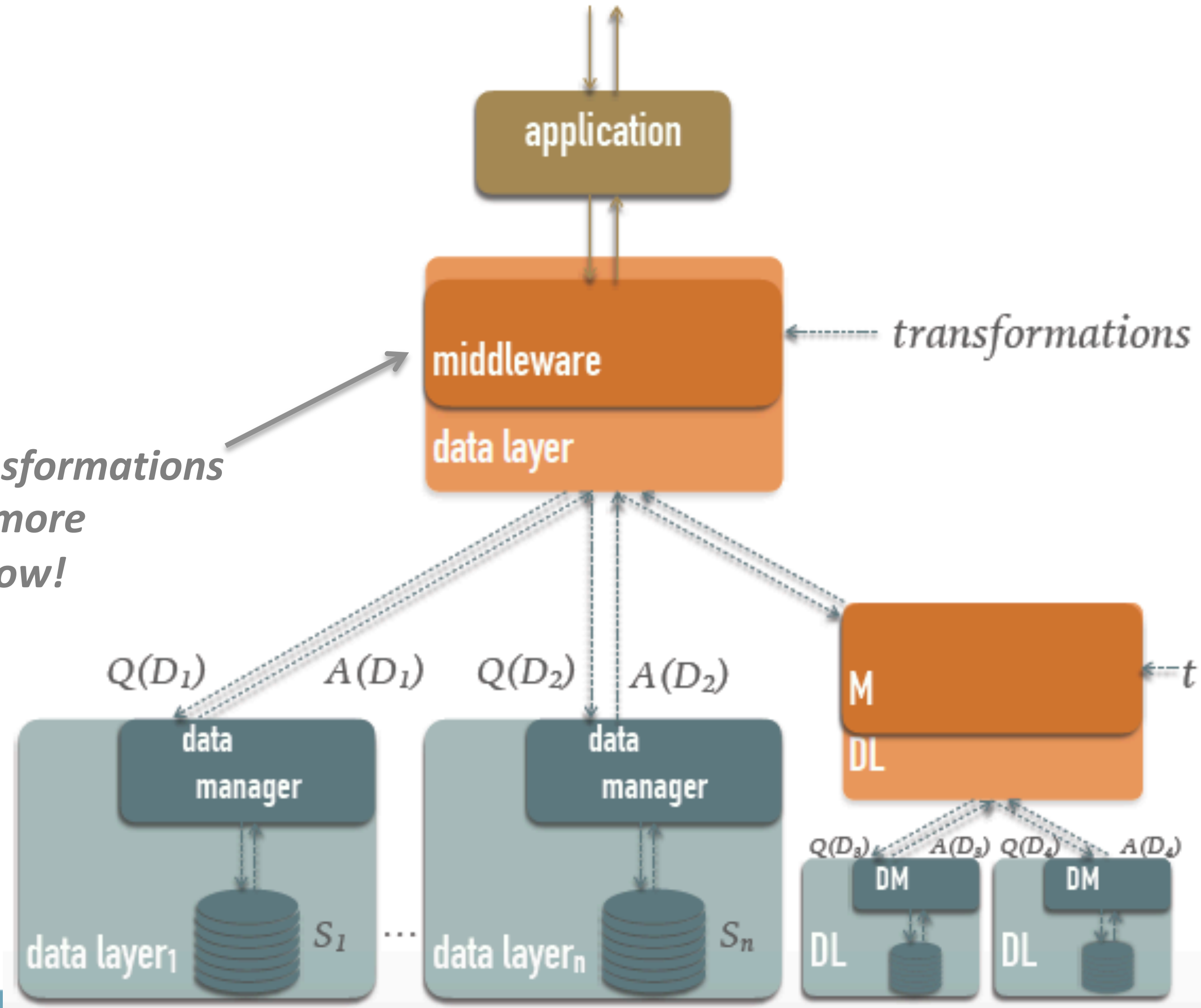
as in traditional DBMSs

EXAMPLES

- Web data
- XML data
- Json
- (Sensor data)

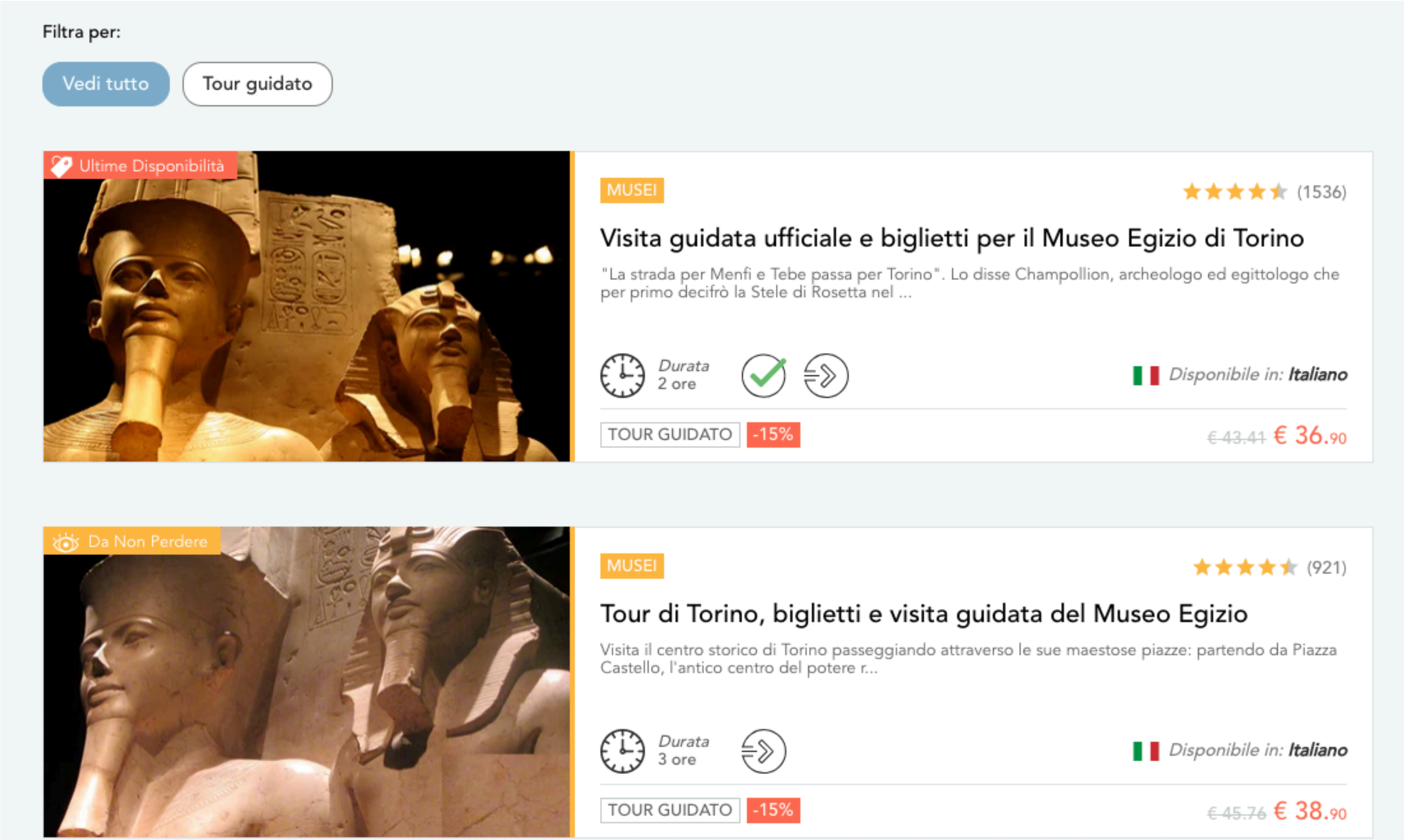
Recall the general framework for Data Integration

These transformations are much more complex now!

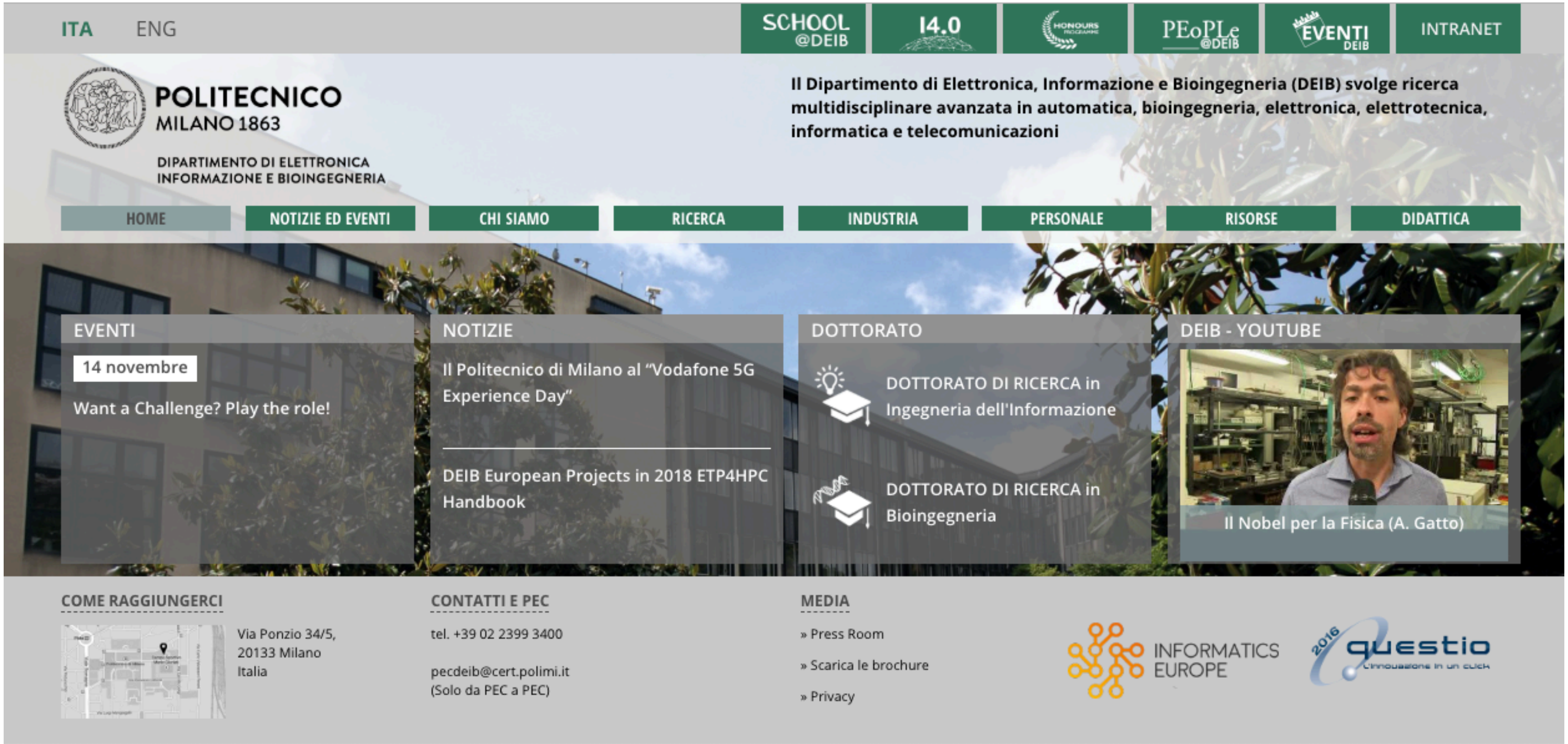


(Alvaro Fernandez, EDBT Summ. School 2017)

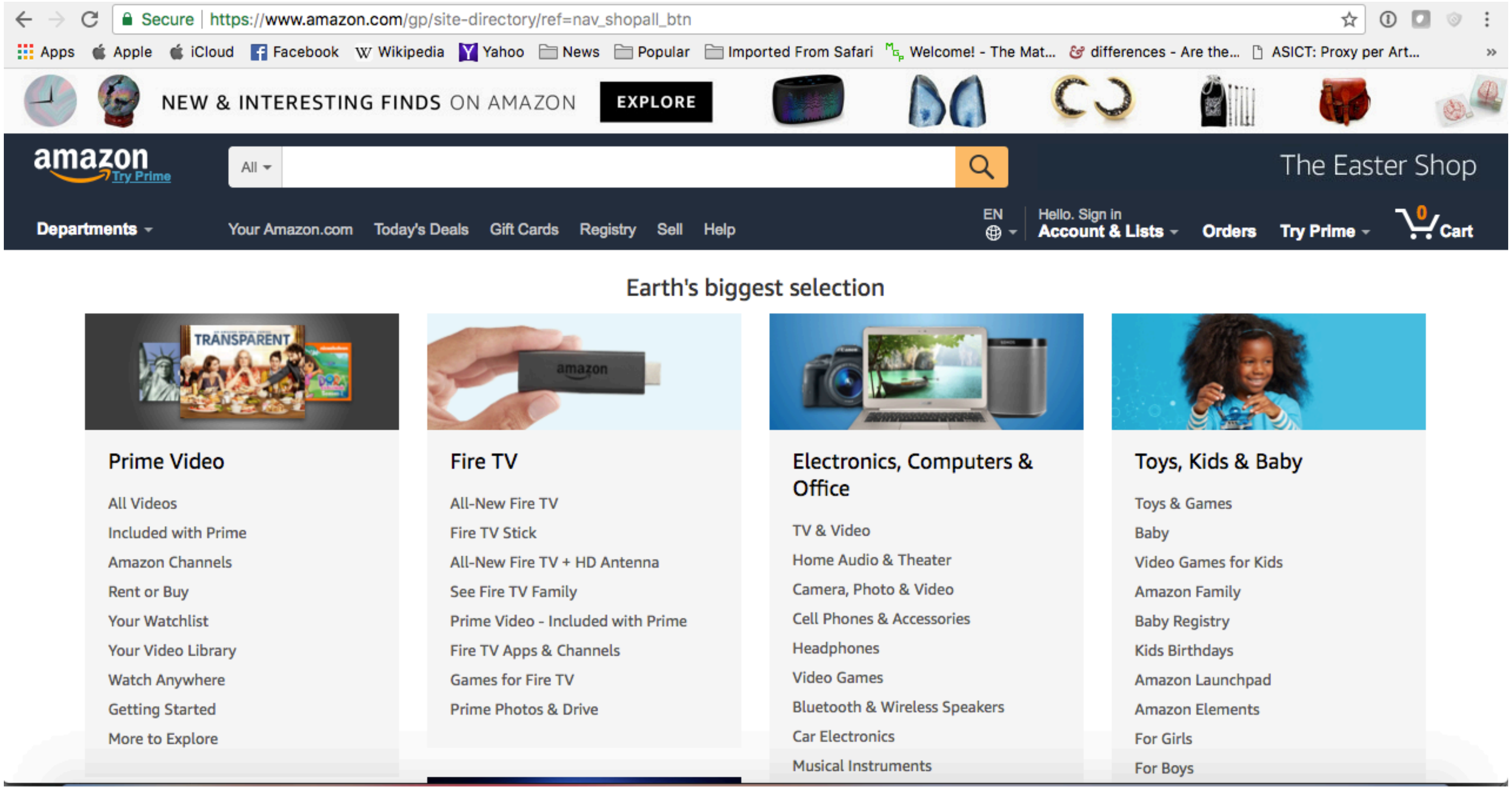
EXAMPLE OF SEMISTRUCTURED DATA



EXAMPLE OF SEMISTRUCTURED DATA



EXAMPLE OF SEMISTRUCTURED DATA



INFORMATION SEARCH IN SEMISTRUCTURED DATABASES

- WE WOULD LIKE TO:
  - INTEGRATE
  - QUERY
  - COMPARE

  DATA WITH DIFFERENT STRUCTURES ALSO WITH SEMISTRUCTURED DATA, JUST AS IF THEY WERE ALL STRUCTURED
- AN OVERALL DATA REPRESENTATION SHOULD BE PROGRESSIVELY BUILT, AS WE DISCOVER AND EXPLORE NEW INFORMATION SOURCES

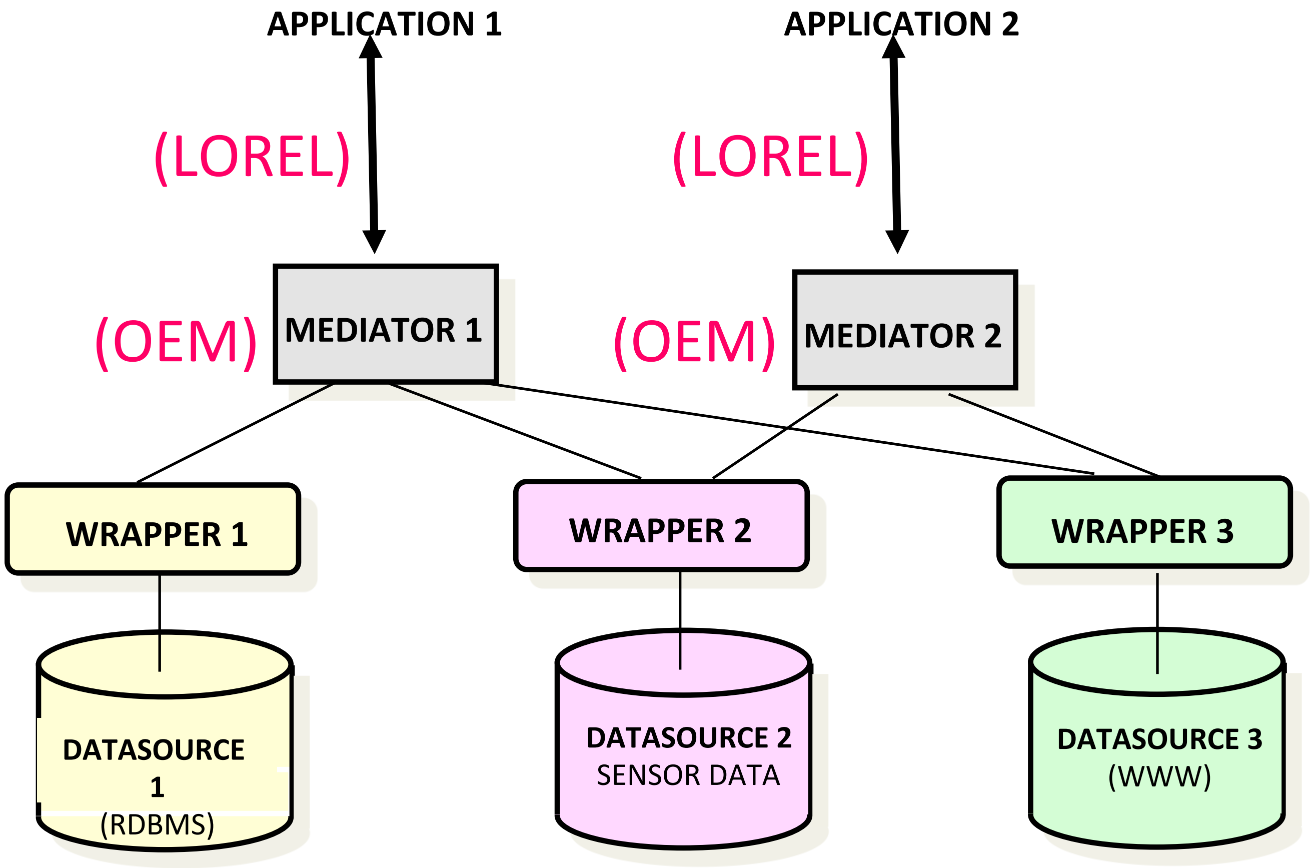
MEDIATORS

The term mediation includes:

- the processing needed to make the interfaces work
- the knowledge structures that drive the transformations needed to transform data into information
- any intermediate storage that is needed (Wiederhold)

First introduction of the mediation concept: TSIMMIS (1990's)

- QUERY POSED TO THE MEDIATOR
- MEDIATOR "KNOWS" THE SEMANTICS OF THE APPLICATION DOMAIN
- UNIQUE, GRAPH-BASED DATA MODEL
- DATA MANAGED BY THE MEDIATOR
- WRAPPERS FOR THE MODEL-TO-MODEL TRANSLATIONS



Integrating semistructured or unstructured data

Mediators:

- Each mediator is specialized in a certain domain (e.g. weather forecast), thus...
- Each mediator must know domain metadata , which convey the data semantics
- The mediator has to solve on-line duplicate recognition and removal (no designer to solve conflicts at design time here)

Wrappers (translators):

- Wrappers convert queries into queries/commands which are understandable for the specific data source
- Wrappers can even extend the query possibilities of a data source
- Wrappers convert query results from the source format to a format which is understandable for the application

Wrappers: estraction of information from HTML docs (e.g. Web pages)

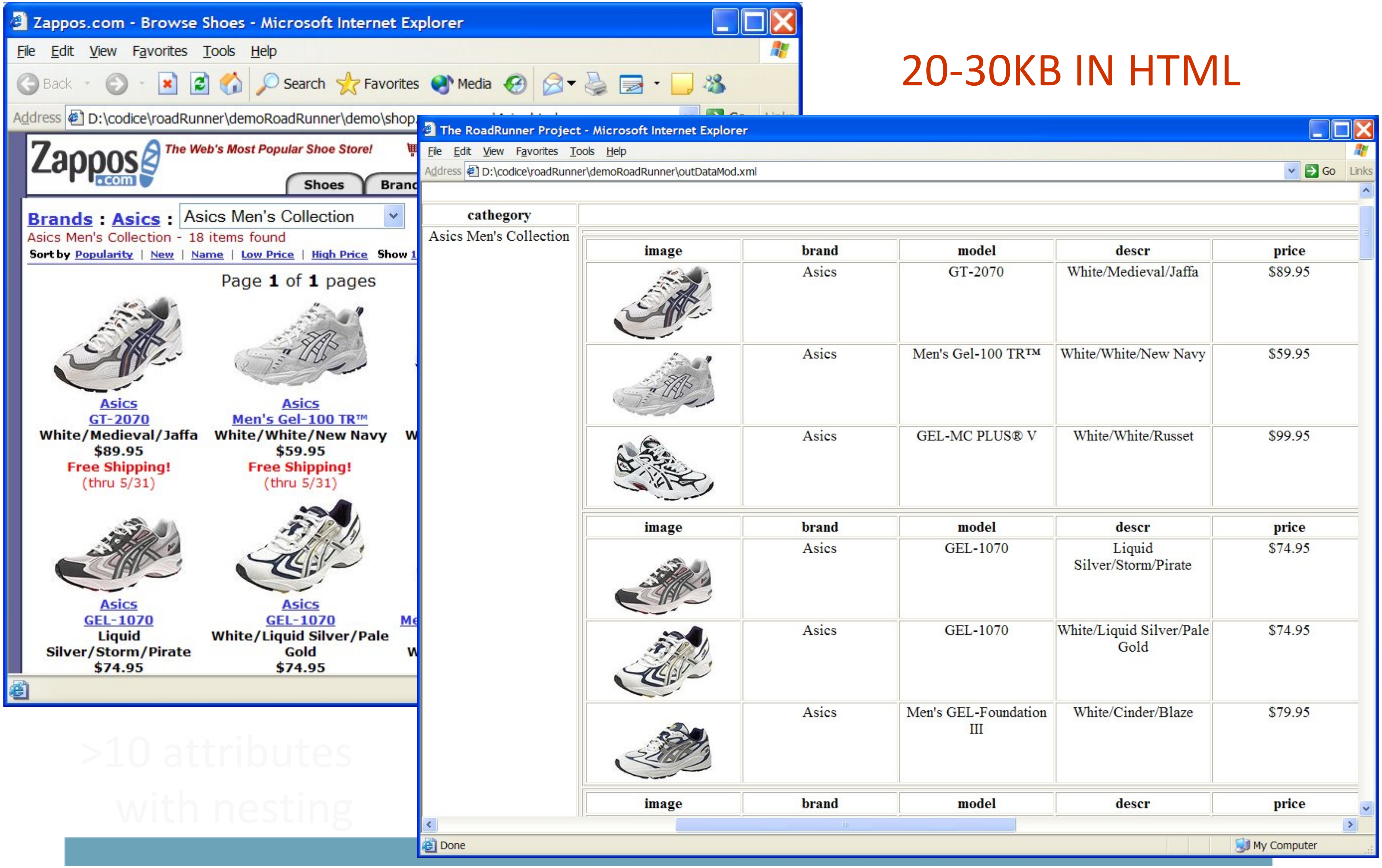
- Information extraction
  - Source Format: plain text with HTML tags (no semantics)
  - Target Format: e.g. relational table (we add *structure*, i.e. *semantics*)
- Wrapper
  - Software module that performs an *extraction step*
  - Intuition: use extraction rules which exploit the *marking tags*

Problems

- Web sites change very frequently
- A layout change may affect the extraction rules
- Human-based maintenance of an ad-hoc wrapper is very expensive
- Better: ***automatic wrapper generation***

Tim Weninger, Rodrigo Palácios, Valter Crescenzi, Thomas Gottron, Paolo Merialdo: ***Web Content Extraction: a MetaAnalysis of its Past and Thoughts on its Future*** SIGKDD Explorations 17(2): 17-23 (2015)

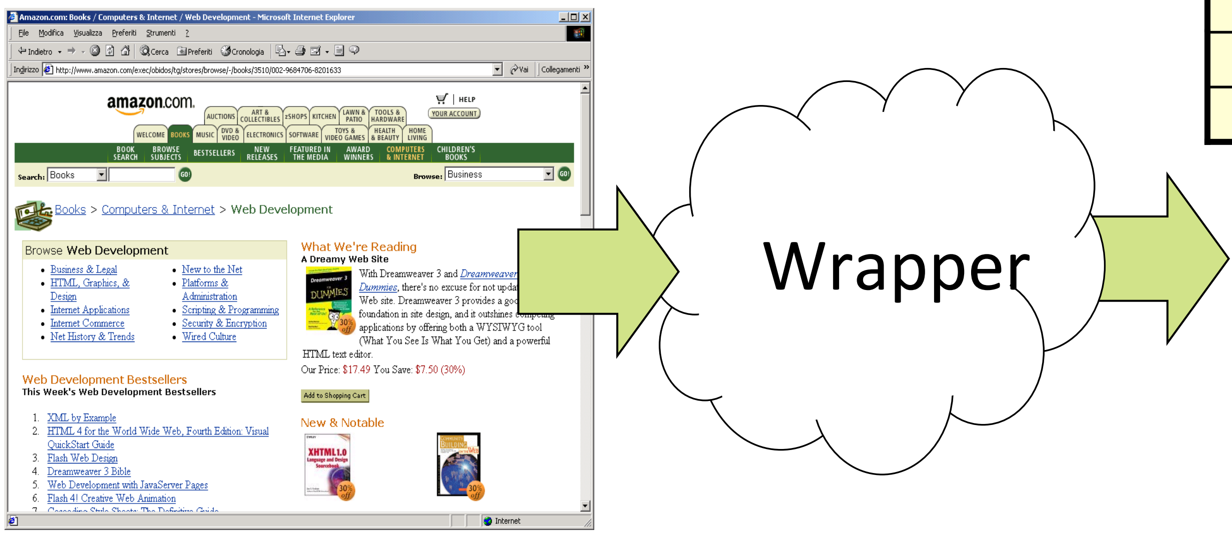
What is behind a commercial Web Site



20-30KB IN HTML

>10 attributes with nesting

WRAPPERS for (data intensive) Web Pages



HTML page

BookTitle	Author	Editor
The HTML Sourcebook	J. Graham	...
Computer Networks	A. Tannenbaum	...
Database Systems	R. Elmasri, S. Navathe	...
Data on the Web	S. Abiteboul, P.Buneman, D.Suciu	...

database table(s)
(or XML docs)

Ontologies, a possible support way to mediation: they *represent knowledge*

ONTOLOGY: a *formal* and shared definition of a vocabulary of terms and of their inter-relationships

- Predefined relations:
  - *synonimy*
  - *omonimy*
  - *hyponimy*
  - *etc..*
- More complex, designer-defined relationships, whose semantics depends on the domain: enrolled(*student,course)*

→ then an ER diagram, a UML class diagram, any conceptual schema *might be an ontology !*

→ right, but here we are interested in automatically explorable and "querable" (*i.e. formal*) representations

Definitions

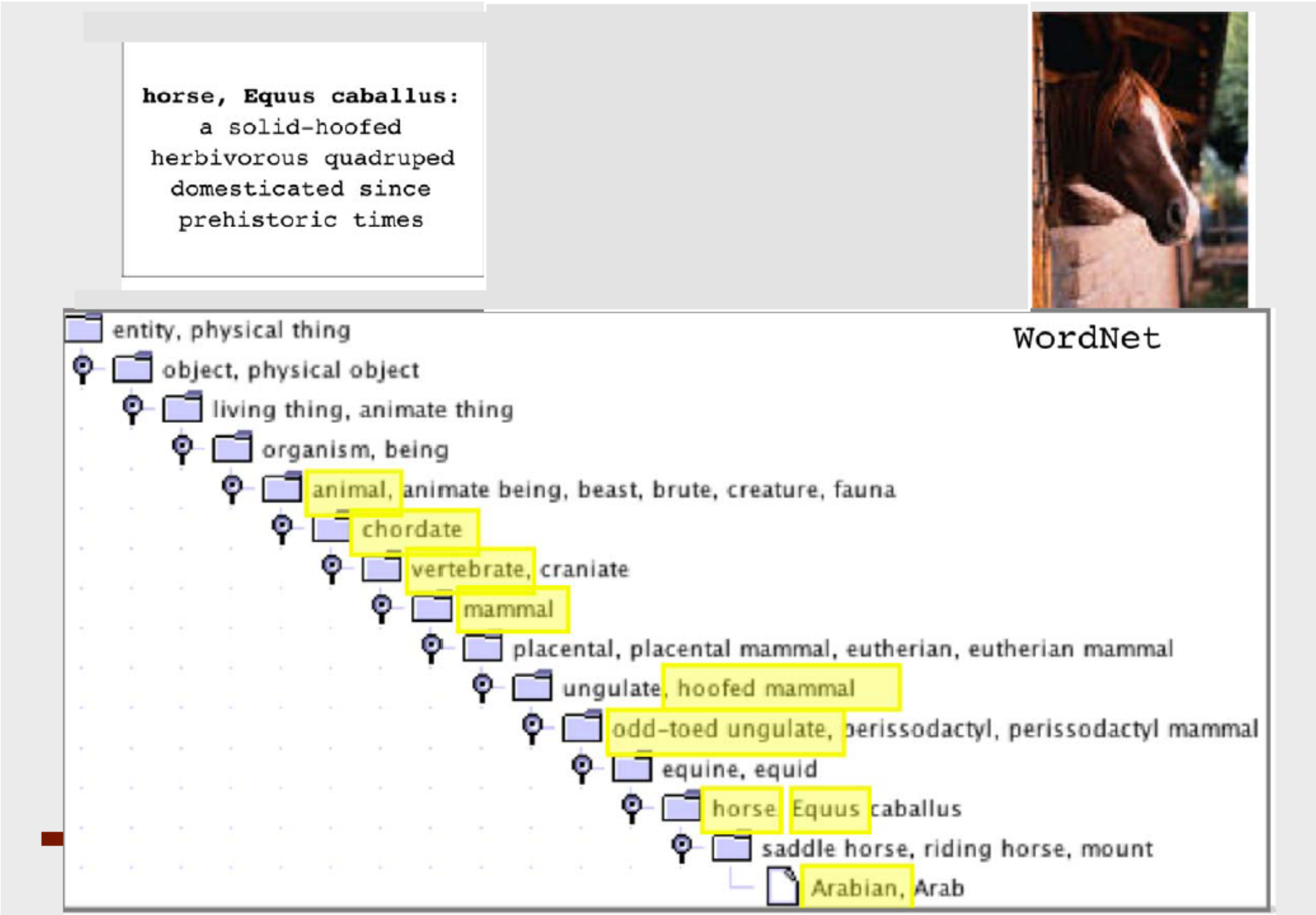
- Ontology = formal specification of a conceptualization of a shared knowledge domain.
- An ontology is a controlled vocabulary that describes objects and the relationships between them in a formal way
- It has a grammar for using the terms to express something meaningful within a specified domain of interest.
- The vocabulary is used to express queries and assertions.
- Ontological commitments are agreements to use the vocabulary in a consistent way for knowledge sharing

semantic interoperability → semantic Web

Ontology types

- **Taxonomic ontologies**
  - Definition of concepts through terms, their hierarchical organization, and additional (pre-defined) relationships (synonymy,composition,…)
  - To provide a reference vocabulary
- **Descriptive ontologies**
  - Definition of concepts through data structures and their interrelationships
  - Provide information for “aligning” existing data structures or to design new, specialized ontologies (***domain ontologies***)
  - Closer to the database area techniques

Wordnet: a taxonomic ontology



An ontology consists of…

- Concepts:
  - Generic concepts, they express general world categories
  - Specific concepts, they describe a particular application domain (***domain ontologies***)
- Concept Definition
  - Via a formal language
  - In natural language
- Relationships between concepts:
  - Taxonomies (IS_A),
  - Meronymies (PART_OF),
  - Synonymies, homonymies, ...
  - User-defined associations,

Formal Definitions

$O = (C, R, I, A)$

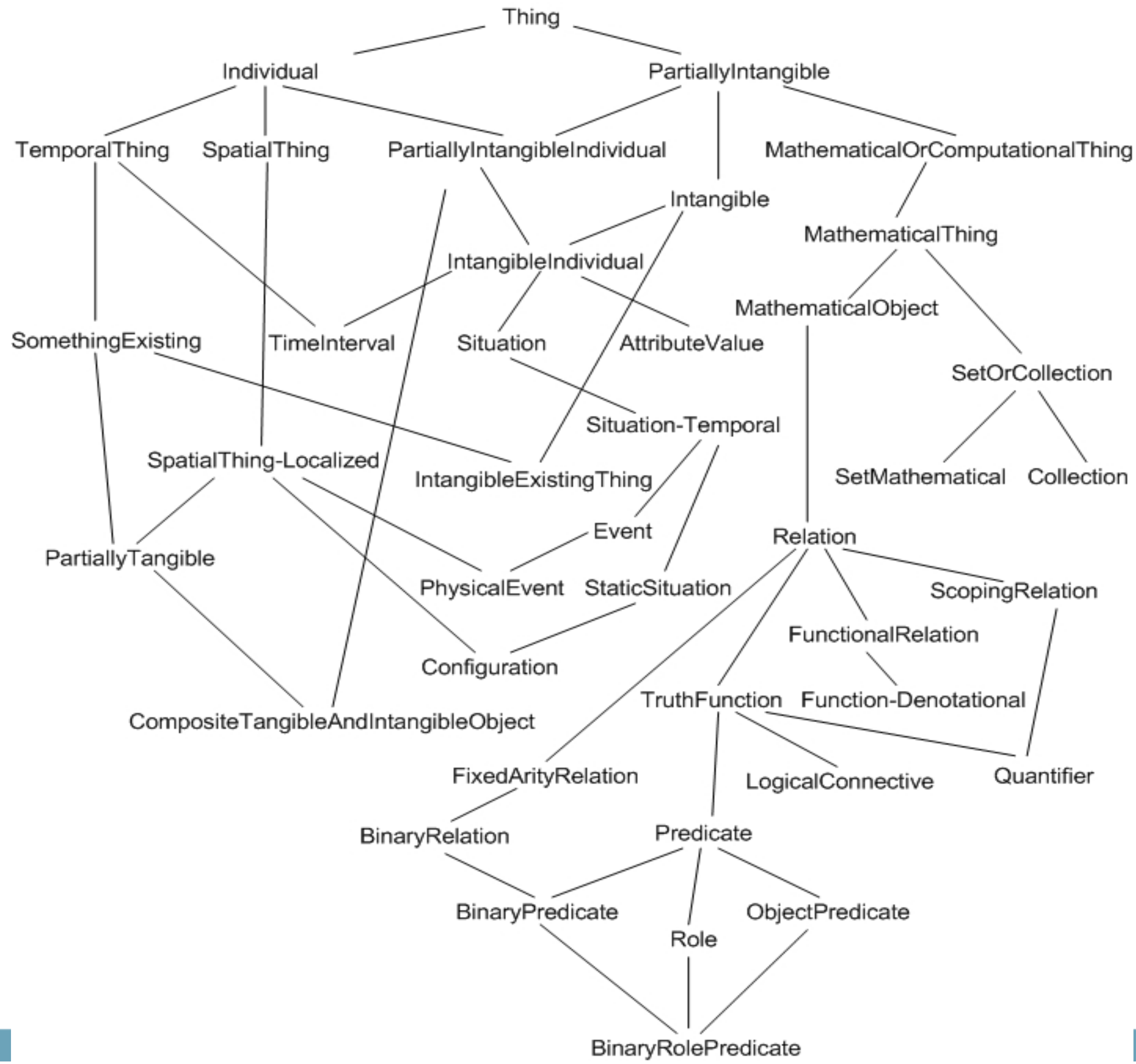
O ontology, *C* concepts, *R* relations, I Instances, *A* axioms

- Specified in some logic-based language
- Organized in a generalization hierarchy
- *I* = instance collection, stored in the information source (e.g., "John", "Politecnico di Milano",...)
- A = set of axioms describing the reality of interest – e.g.
  - "a FATHER is a PERSON"
  - "John is a FATHER"
  - "Annie is daughter of John"

OpenCyc: a descriptive ontology

- The open source version of the Cyc technology, started in 1984 at MCC.
- Available until early 2017 as OpenCyc under an open source (Apache) license.
- More recently, Cyc has been made available to AI researchers under a research-purpose license as ResearchCyc.
- The entire Cyc ontology containing hundreds of thousands of terms, along with millions of assertions relating the terms to each other, forming an ontology whose domain is all of human consensus reality.

Top level concepts of Cyc



Some famous datasets

- CKAN – registry of open data and content packages provided by the Open Knowledge Foundation
- DBpedia – a dataset containing data extracted from Wikipedia; it contains about 4 million concepts described by some billion triples, including abstracts in 11 different languages
- GeoNames provides RDF descriptions of more than 7,500,000 geographical features worldwide.
- YAGO (Yet Another Great Ontology) is an ever-growing open source knowledge base developed at the Max Planck Institute for Computer Science in Saarbrücken. It is automatically extracted from Wikipedia and other sources.
- UMBEL – a lightweight reference structure of 20,000 subject concept classes and their relationships derived from OpenCyc, which can act as binding classes to external data; also has links to 1.5 million named entities from DBpedia and YAGO
- FOAF – a dataset describing persons, their properties and relationships

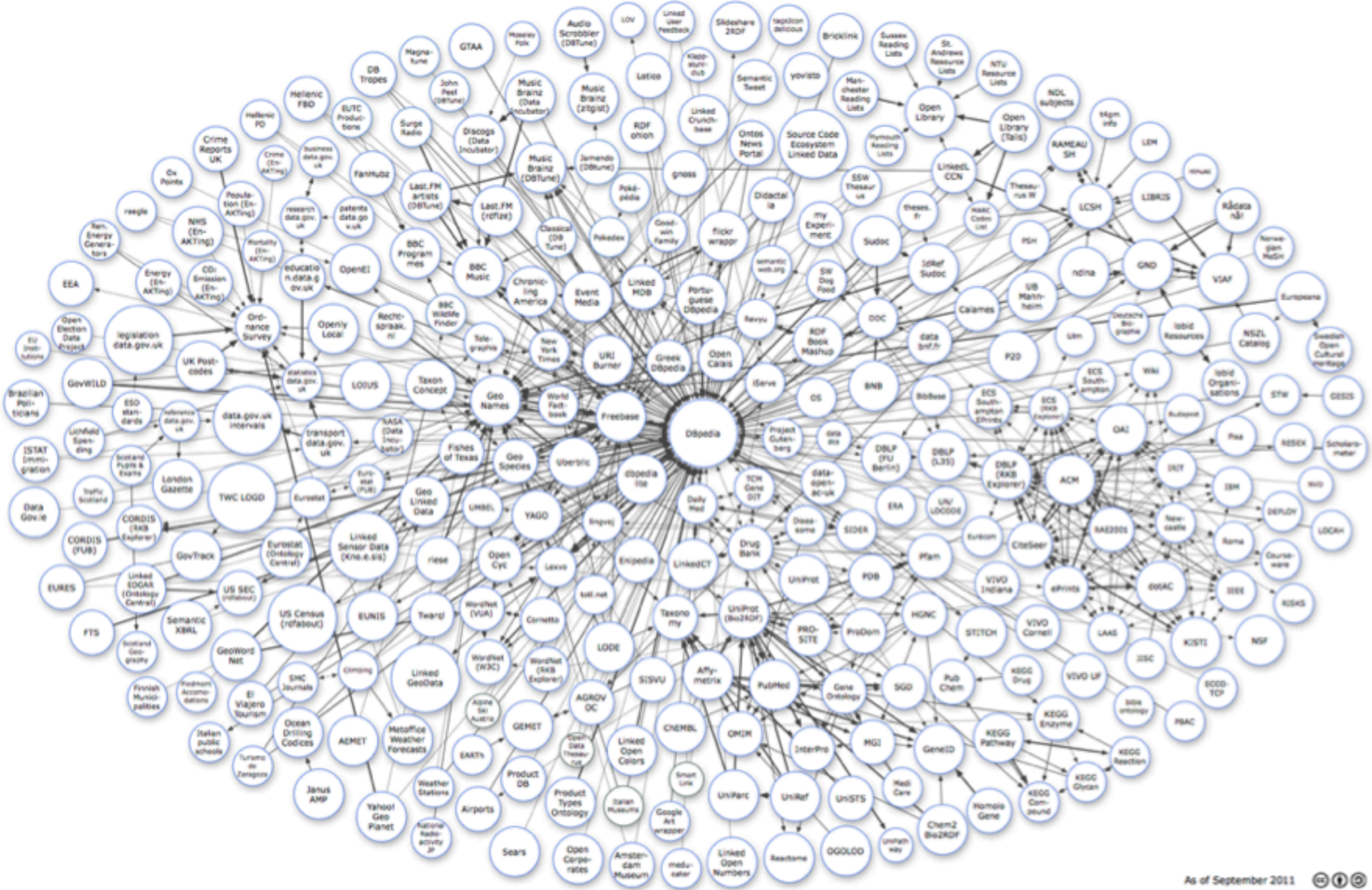
The Semantic Web

- A vision for the future of the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web.
- builds on XML's ability to define customized tagging schemes and RDF's flexible approach to representing data.
- the first level above RDF: OWL, an ontology language what can formally describe the meaning of terminology used in Web documents → beyond the basic semantics of RDF Schema.

Linked Data

- Linked Data is a W3C-backed movement about connecting data sets across the Web. It describes a method of publishing structured data so that it can be interlinked and become more useful.
- It builds upon standard Web technologies such as HTTP, RDF and URIs, but extends them to share information in a way that can be read automatically by computers, enabling data from different sources to be connected and queried.
- A subset of the wider Semantic Web movement, which is about adding meaning to the Web
- Open Data describes data that has been uploaded to the Web and is accessible to all
- Linked Open Data: extend the Web with a data commons by publishing various open datasets as RDF on the Web and by setting RDF links among them

Linked Open Data Cloud Diagram



https://www.w3.org/2010/Talks/0928-socialcloud-tbl/ - (4)

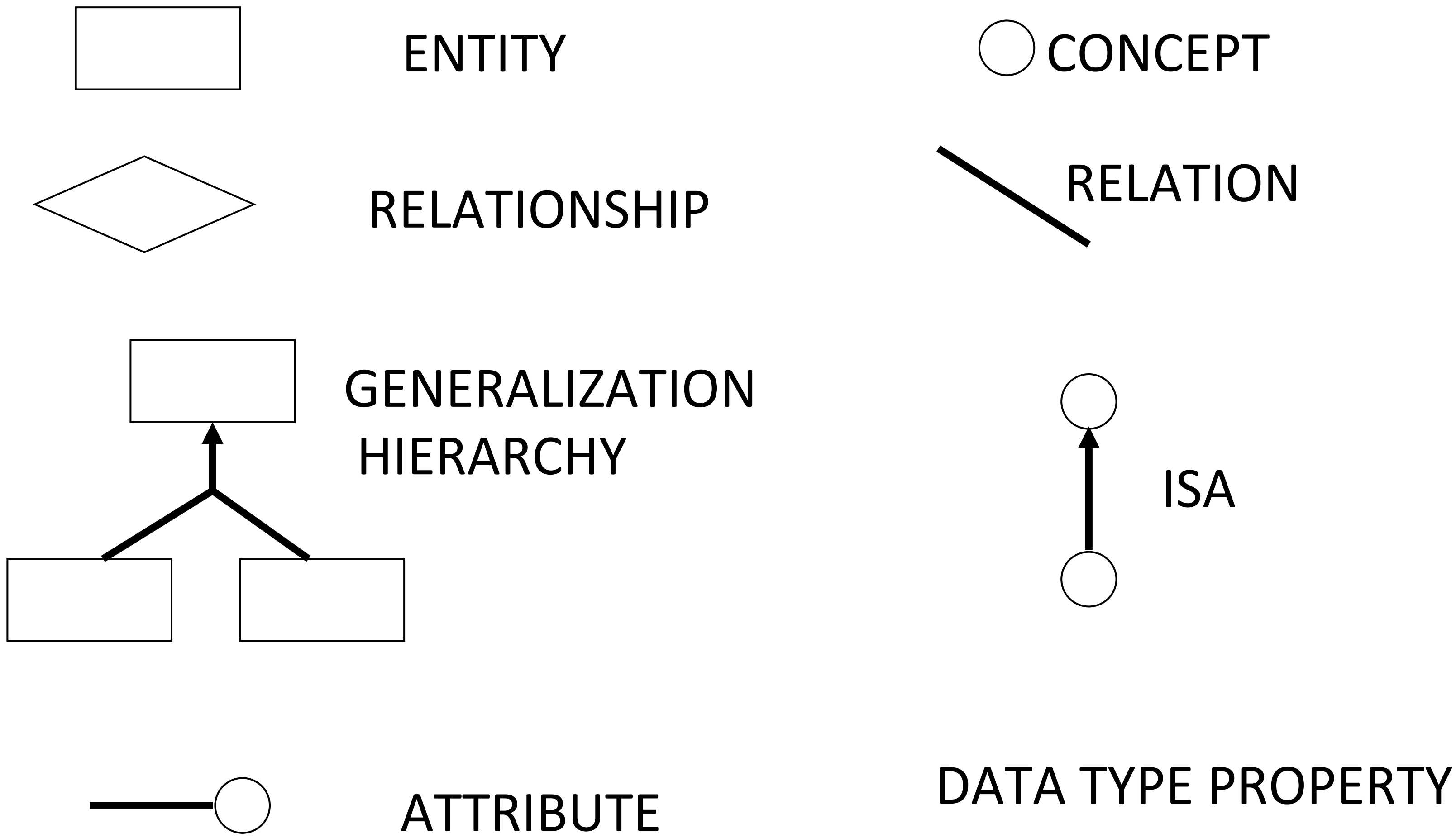
RDF and OWL

- Designed to meet the need for a Web Ontology Language, OWL is part of the growing stack of W3C recommendations related to the Semantic Web.
- XML provides a surface syntax for structured documents, but imposes no semantic constraints on the meaning of these documents.
- XML Schema is a language for restricting the structure of XML documents and also extends XML with data types.
- RDF is a data model for objects ("resources") and relations between them, provides a simple semantics for this data model, and can be represented in an XML syntax.
- RDF Schema is a vocabulary for describing properties and classes of RDF resources, with a semantics for generalization-hierarchies of such properties and classes.
- OWL adds more vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes.

A fragment of an RDF (XML) document, describing an ontology. The language is OWL http://www.w3.org/TR/owl-ref/

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns="http://eng.it/ontology/tourism#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
  xml:base="http://eng.it/ontology/tourism">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Church">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Definition: Edificio sacro in cui si svolgono pubblicamente gli atti
di culto delle religioni cristiane.</rdfs:comment>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#PlaceOfWorship"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Theatre">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Definition: a building where theatrical performances or motion-
picture shows can be presented.</rdfs:comment>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#SocialAttraction"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="DailyCityTransportationTicket">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#CityTransportationTicket"/>
    </rdfs:subClassOf>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Definition: Biglietto che consente di usufruire di un numero
illimitato di viaggi sui mezzi pubblici (autobus e metropolitana)
all'interno del centro urbano (o della regione, con un costo maggiore) per
un periodo di 24 ore.</rdfs:comment>
  </owl:Class>
```

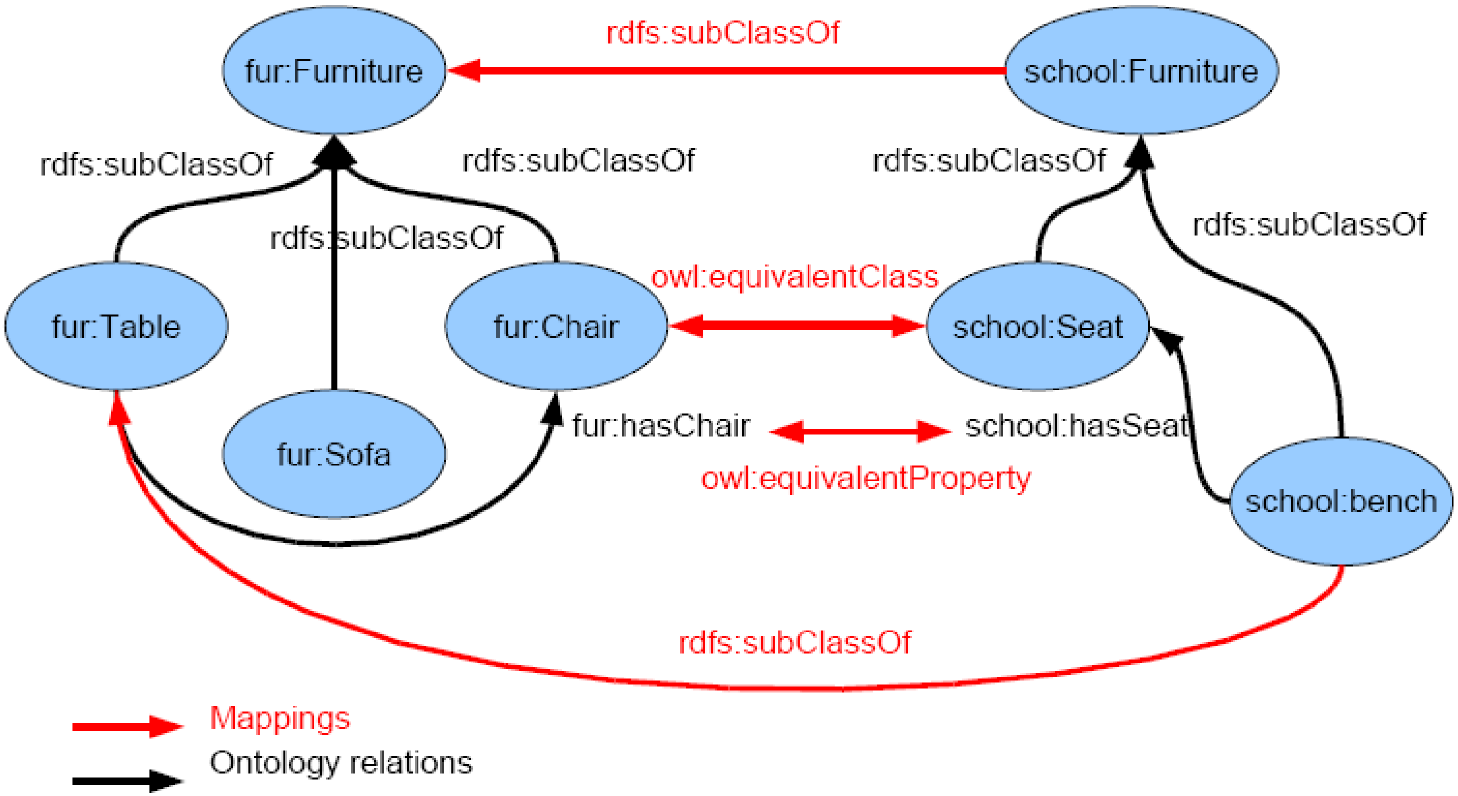
ER vs.ontology



Automatic Ontology *matching*

- The process of finding pairs of resources coming from different ontologies which can be considered equal in meaning – *matching operators*
- The similarity value is usually a number in the interval [0,1]
- It is an input to the different approaches to integration, described below
- Mediation may be done without integrating the ontologies, but using the matchings in different ways

Ontology mapping



How can ontologies support automatic integration?

An ontology as a schema integration support tool

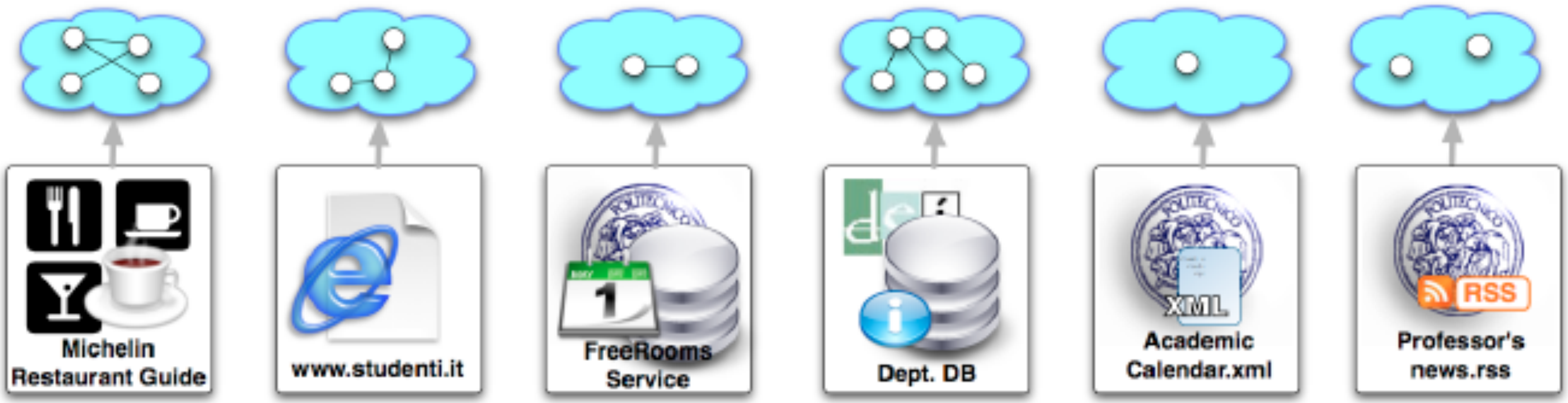
- Ontologies used to represent the semantics of schema elements (if the schema exists)
- Similarities between the source ontologies guide conflict resolution
  - At the schema level (if the schemata exist)
  - At the instance level (record linkage)

An ontology instead of a global schema:

- Schema-level representation only in terms of ontologies
- Ontology mapping, merging, etc. instead of schema integration
- Integrated ontology used as a schema for querying

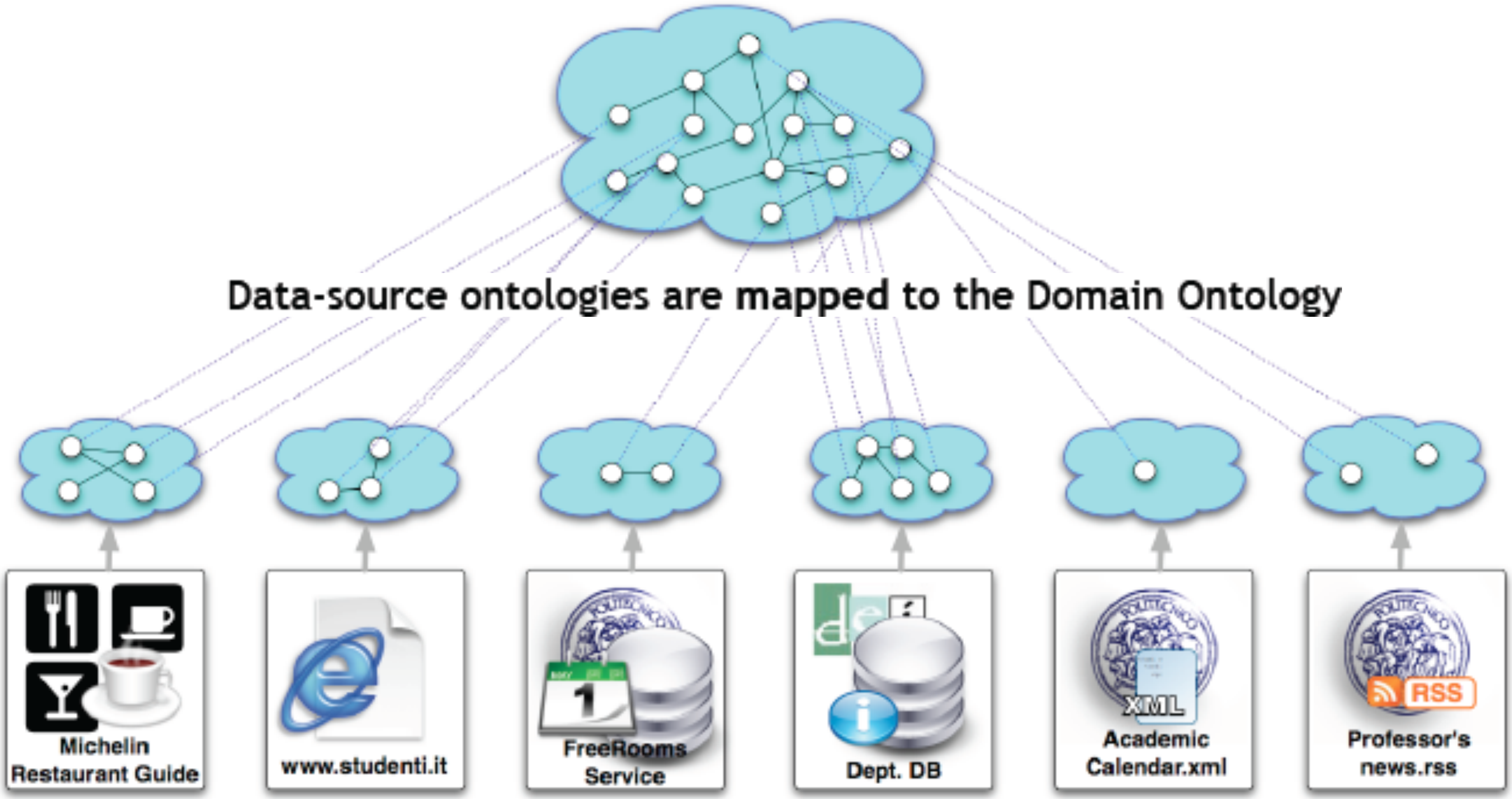
An ontology instead of a global schema

- Data-source heterogeneity is solved by extracting the semantics in an ontological format (potentially *at run-time*)
- Automatic Wrapper generation + Query translation will bridge among two models.
- Not an easy task:
  - several issues, e.g., impedance mismatch
  - unstructured data sources



An ontology
instead of a global schema

Global Schema : Domain Ontology (*at design-time*)



Ontologies to support integrated data querying

Ontologies provide query languages allowing

- Schema exploration
- Reasoning on the schema
- Instance querying
- E.g. SPARQL (W3C)

More ways for ontologies to support automatic integration

- An ontology as a support tool for content interpretation and wrapping (e.g. HTML pages)

- An ontology as a support tool for content inconsistency detection and resolution (record linkage and data fusion)

Lightweight Integration

Many data integration tasks are transient:

- We may need to integrate data from multiple sources to answer a question asked once or twice. The integration needs to be done quickly and by people without technical expertise (e.g. a disaster response situation in which reports are coming from multiple data sources in the field, and the goal is to corroborate them and quickly share them with the affected public)

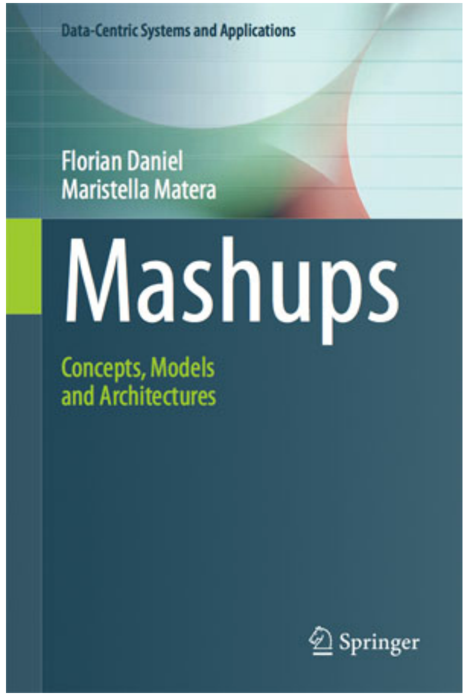
- Problems typical of lightweight data integration:
  - locating relevant data sources
  - assessing source quality
  - helping the user understand the semantics
  - supporting the process of integration.

- Ideally, machine learning and other techniques can be used to amplify the effects of human input, through semi-supervised learning, where small amounts of human data classification, plus large amounts of additional raw (“unlabeled”) data, are used to train the system.

→ Mash-up is an example of lightweight integration

Mashups: a paradigm for lightweight integration

(from a tutorial by:
Cinzia Cappiello, Florian Daniel, Maristella Matera)

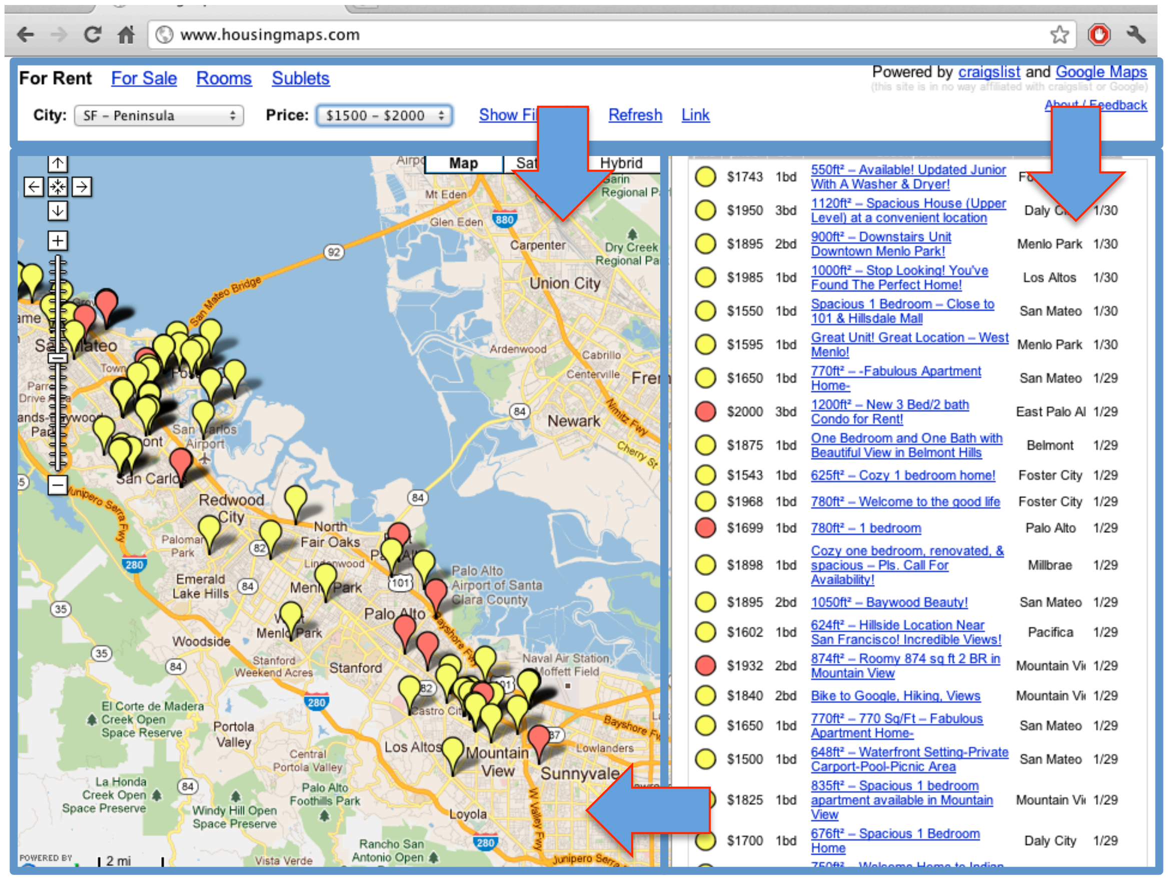


The term mashup is widely used today:

A mashup is *an application that integrates two or more mashup components at any of the application layers* (data, application logic, presentation layer) possibly putting them into communication with each other

Own application logic/UI

GoogleMaps



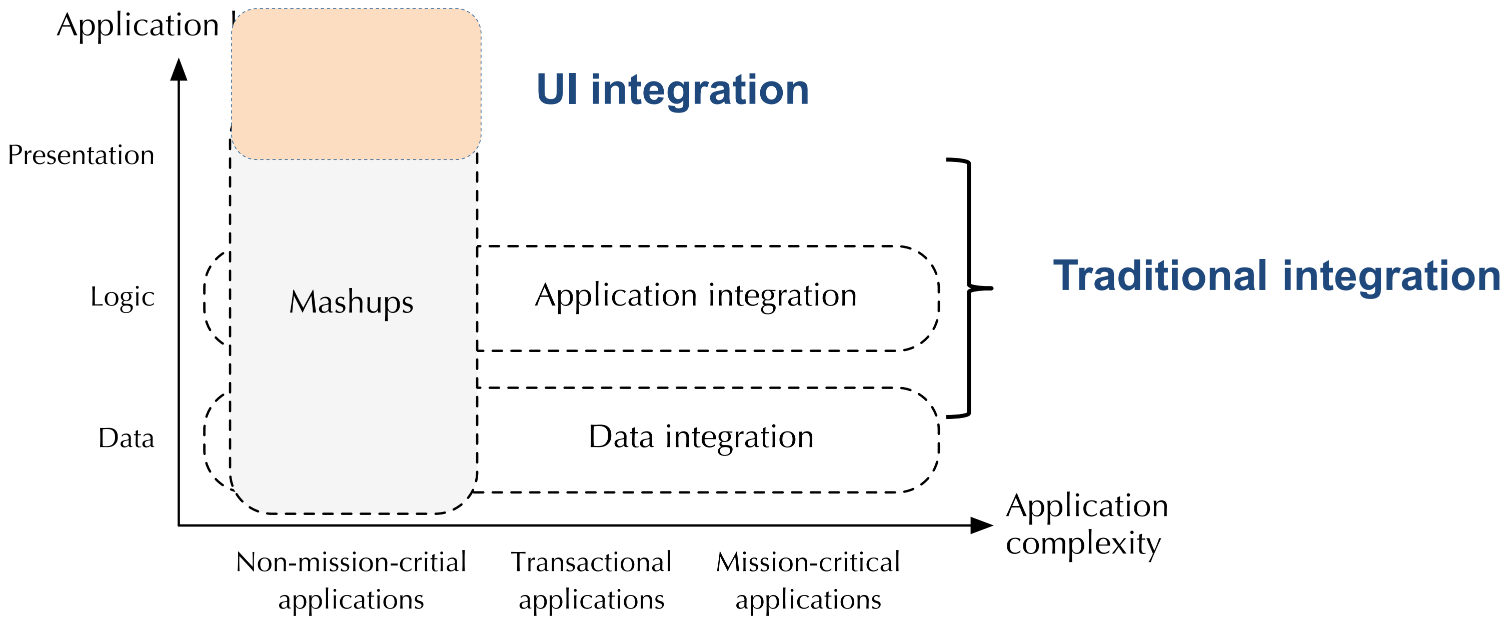
Craiglist

The housingmaps.com mashup

Provides for the synchronized exploration of housing offers from *craigslist.com* and maps by *Google Maps*

Integration is the added value provided by the mashup

Mashup positioning
w.r.t. other integration practices



Mashups introduce integration at the presentation layer and typically focus **on non-mission-critical applications**

Data mashup vs. data integration…

- Data mashups are a lighweight form of data integration, intended to solve different problems
- Covering the “long tail” of data integration requirements
  - Very specific reports or ad-hoc data analyses
  - Simple, ad-hoc data integrations providing “situational data” that meet short term needs
  - Non-mission-critical integration requests
  - On-the-fly data integration

Data spaces: another example of lightweight integration

Two basic principles:

- keyword search over a collection of data coupled with effective data visualization. This can be enriched with some of the techniques for automatic schema or instance matching, automatic extraction of metadata.

- Improving the metadata in the system to the end of supporting and validating schema mappings, instance and reference reconciliation, or improve the results of information extraction.

Data Lakes: a fashionable term in ICT companies, one possible concretization of Data Spaces

- (Gartner) A Data Lake is a concept consisting of a collection of storage instances of various data assets. These assets are stored in a near-exact, or even exact, copy of the source format and are in addition to the originating data stores.
- A Data Lake is a storage repository that holds a vast amount of raw data in its native format until it is needed.
- A Data Lake contains all data, both raw sources over extended periods of time as well as any processed data. The purpose of a Data Lake is to enable users across multiple business units to refine, explore and enrich data on their terms

Current problems and ideas in (Big) Data Integration

reference:

Principles of Data Integration
by A. Doan, A. Halevy, and Z. Ives
Morgan Kaufmann

Uncertainty in Data Integration

- Data itself may be uncertain (e.g. extracted from an unreliable source)
- Mappings might be approximate (e.g. created by relying on automatic ontology matching)
- Reconciliation is approximate
- Approximate mediated schema
- Imprecise queries, such as keyword-search queries, are approximate

Data Provenance

- Also called data lineage or data pedigree. Sometimes knowing where the data have come from and how they were produced is critical.
- Provenance of a data item records “where it comes from”:
  – Who created it
  – When it was created
  – How it was created - as a value in a database, as the result of a computation, coming from a sensor, etc...
- E.g. an information extractor might be unreliable, or one data source is more authoritative than others

- The database community models provenance in terms of how the datum was derived from the original source databases, the rest is left to the application (it is assumed to be domain dependent)

Uses of provenance information

- Explanations
- Scoring of sources and data quality
- Influence of sources on one another
- Utilize data usage, provenance and data quality info to assess uncertainty and automate cleaning

Crowdsourcing

- Some checks are very simple for humans but hard for a computer
  - image contents
  - Web content extraction

  ....
- Amazon Mechanical Turk

- Wikipedia is also a kind of crowdsourcing, collecting information from "unknown" humans

- Can provide powerful solutions to traditionally hard data integration problems (e.g. wrapping, as above, check correcteness of schema mappings, etc.)

Bibliography

- A. Doan, A. Halevy and Z. Ives, Principles of Data Integration, Morgan Kaufmann, 2012
- L. Dong, D. Srivastava, Big Data Integration, Morgan & Claypool Publishers, 2015
- Roberto De Virgilio, Fausto Giunchiglia, Letizia Tanca (Eds.): Semantic Web Information Management – A Model-Based Perspective. Springer 2009, ISBN 978-3-642-04328-4
- M. Lenzerini, Data Integration: A Theoretical Perspective, Proceedings of ACM PODS, pp. 233-246, ACM, 2002, ISBN: 1-58113-507-6
- Clement T. Yu, Weiyi Meng, Principles of Database Query Processing for Advanced Applications , Morgan Kaufmann, 1998, ISBN: 1558604340

Data Management in Pervasive Systems

(Courtesy of prof. Fabio A. Schreiber)

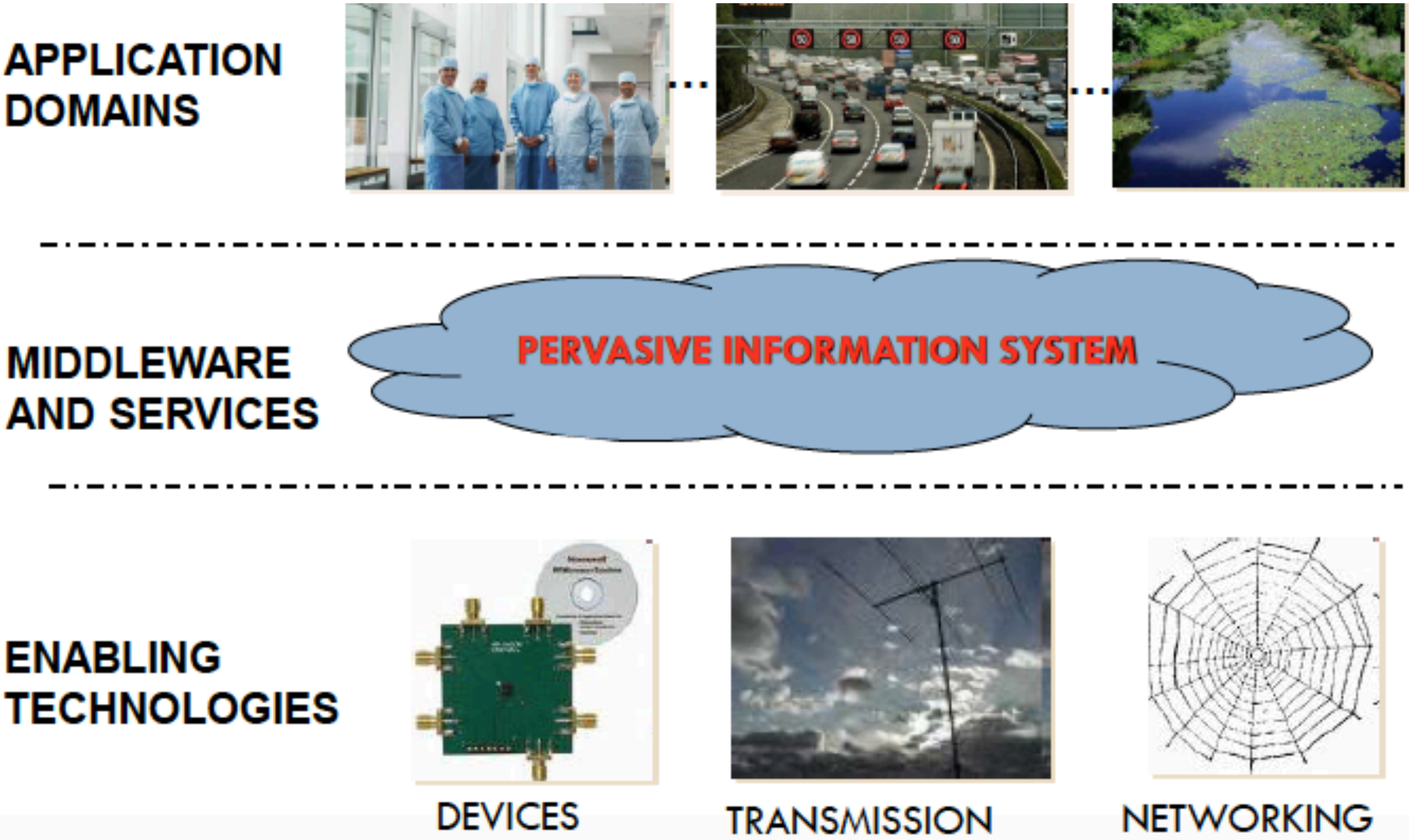
PERVASIVE SYSTEMS

PERVASIVE → PER – VADERE

THROUGH **TO GO**

- The middleware of a pervasive system hides the heterogeneity of hundreds of devices making them transparent to the application
- the perception of the environment makes the system autonomic and proactive
  - ✓ context-aware
  - ✓ reactive
  - ✓ self-adapting

Pervasive System Components



Pervasive Systems and WSNs

WIRELESS SENSOR NETWORKS (WSN) CONSTITUTE THE BACKBONE OF PERVASIVE SYSTEMS

COMPONENTS

- THOUSANDS OF TINY LOW POWER DEVICES SPREAD OVER (POSSIBLY LARGE) PHYSICAL AREAS
- THE DEVICES MUST BE SMALL, UNOBTRUSIVE, AND CHEAP

NETWORK

- THE NETWORK MUST BE UNEXPENSIVE TO DEVELOP, DEPLOY, PROGRAM, AND EASY TO UTILIZE AND MAINTAIN
- COMPRISE A NUMBER OF SENSOR NODES AND A BASE STATION

A real-life sensor Data Sheet

TC-Link®-1CH-LXRS™

Data Sheet

1 Channel Wireless Thermocouple Node

Introduction

The TC-Link® -1CH-LXRS® 1 Channel Wireless Thermocouple Node features a standard thermocouple input connector with an embedded cold junction temperature compensation sensor. On-board linearization algorithms are software programmable to support a wide range of thermocouple types including J, K, N, R, S, T, E and B. Its internal rechargeable battery allows remote, long term deployment.

Features & Benefits

High Performance

- Scalable, ultra-long-range wireless sensor network
- High-speed, synchronized platform accepts most analog sensors
- Reliable wireless data collection
- Low-power for extended battery life
- SensorCloud – integrated web solution

Ease of Integration

- Small, easy to integrate wireless form factor
- SDK for quick custom app development
- Rapidly deployed wireless solution

Cost Effective

- Significantly reduced development cost
- Competitive OEM and volume discount schedule

Applications

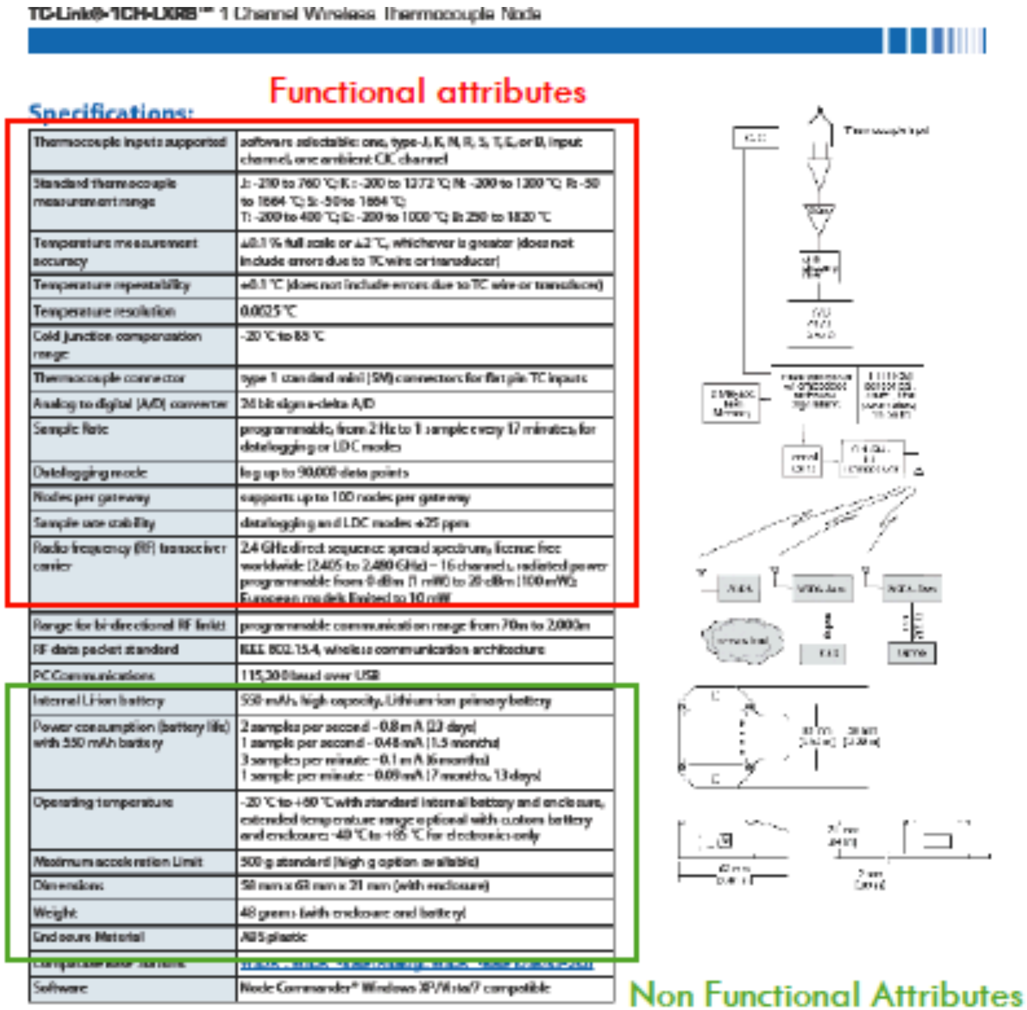
- civil structure sensing, concrete maturation
- industrial sensing networks, machine thermal management
- food and transportation systems, refrigeration, freezer performance monitoring
- advanced manufacturing, plastic processing, composite cure monitoring
- cryogenic applications

System Overview

At the heart of MicroStrain's® LXRS™ Lossless Data Wireless Sensor Networks are WSDA® gateways, which use our exclusive beaconing protocols to synchronize precision timekeepers within each sensor node in the network. The WSDA® also coordinates data collection from all sensor nodes. Users can easily program each node on the scalable network for simultaneous, periodic, burst, or data logging mode sampling with our Node Commander® software, which automatically configures network radio communication to maximize the aggregate sample rate. Optional SensorCloud® enabled WSDA® support autonomous web-based data aggregation.

MicroStrain® Little Sensors, Big Ideas.®

www.microstrain.com



TC-Link®-1CH-LXRS™ 1 Channel Wireless Thermocouple Node

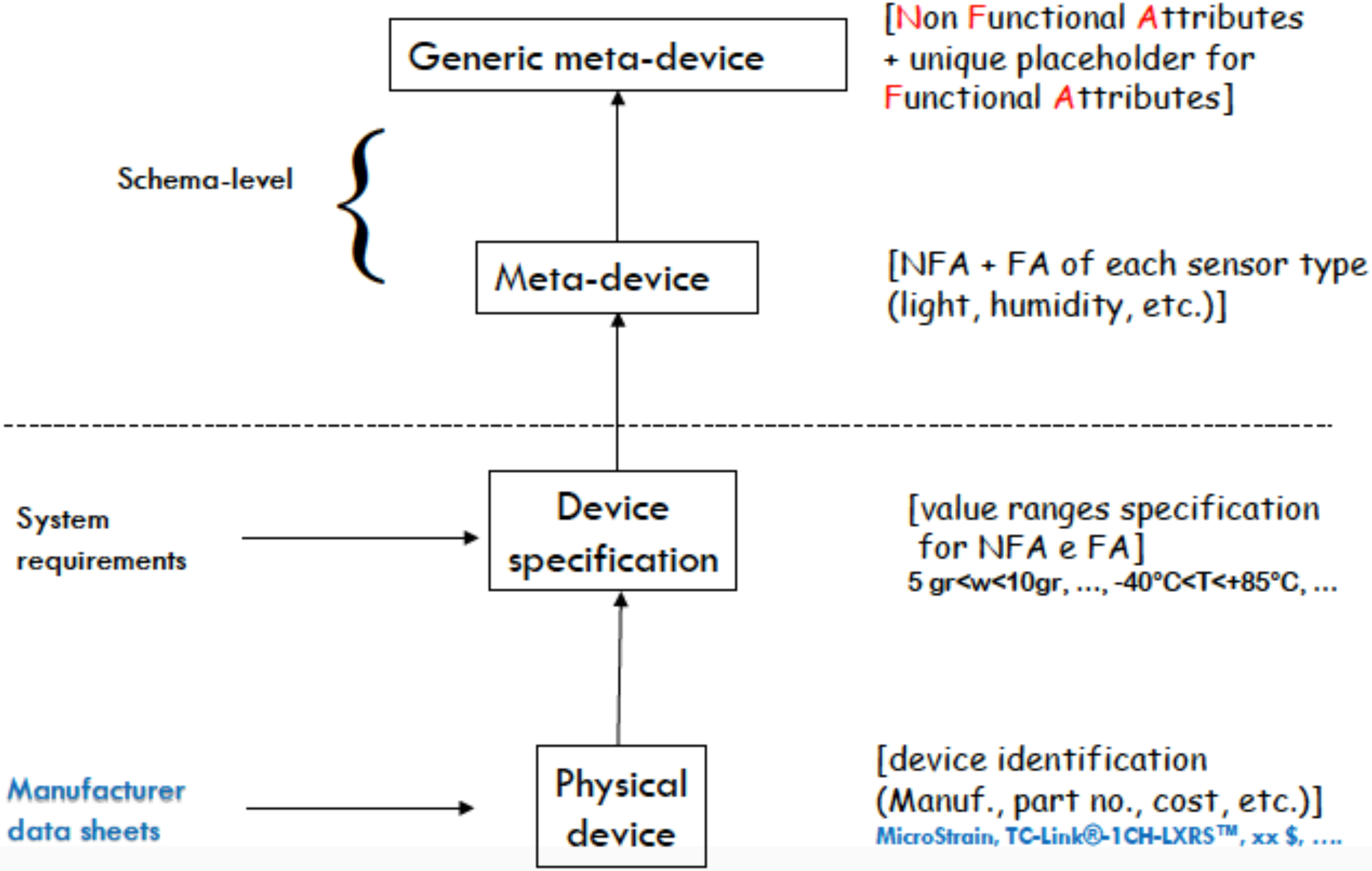
Functional attributes

Specifications:

Thermocouple inputs supported	software selectable: one, type-J, K, N, R, S, T, E, or B, input channel, one ambient CJC channel
Standard thermocouple measurement range	J: -210 to 760 °C; K: -200 to 1372 °C; N: -200 to 1300 °C; R: -50 to 1664 °C; S: -50 to 1664 °C; T: -200 to 400 °C; E: -200 to 1000 °C; B: 250 to 1820 °C
Temperature measurement accuracy	±0.1 % full scale or ±2 °C, whichever is greater (does not include errors due to TC wire or transducer)
Temperature repeatability	±0.1 °C (does not include errors due to TC wire or transducer)
Temperature resolution	0.0025 °C
Cold junction compensation range	-20 °C to 85 °C
Thermocouple connector	type 1 standard mini (SM) connectors for flat pin TC inputs
Analog to digital (A/D) converter	24 bit sigma-delta A/D
Sample Rate	programmable, from 2 Hz to 1 sample every 17 minutes, for datalogging or LDC modes
Datalogging mode	log up to 90,000 data points
Nodes per gateway	supports up to 100 nodes per gateway
Sample rate stability	datalogging and LDC modes ±25 ppm
Radio frequency (RF) transceiver carrier	2.4 GHz direct sequence spread spectrum, license free worldwide (2.405 to 2.480 GHz) – 16 channels, radiated power programmable from 0 dBm (1 mW) to 20 dBm (100 mW); European models limited to 10 mW
Range for bi-directional RF link	programmable communication range from 70m to 2,000m
RF data packet standard	IEEE 802.15.4, wireless communication architecture
PC Communications	115,200 baud over USB
Internal Li-ion battery	550 mAh, high capacity, Lithium-ion primary battery
Power consumption (battery life) with 550 mAh battery	2 samples per second - 0.8 mA (23 days) 1 sample per second - 0.48 mA (1.5 months) 3 samples per minute - 0.1 mA (6 months) 1 sample per minute - 0.09 mA (7 months, 13 days)
Operating temperature	-20 °C to +60 °C with standard internal battery and enclosure, extended temperature range optional with custom battery and enclosure; -40 °C to +85 °C for electronics only
Maximum acceleration Limit	500 g standard (high g option available)
Dimensions	58 mm x 68 mm x 21 mm (with enclosure)
Weight	48 grams (with enclosure and battery)
Enclosure Material	ABS plastic
[illegible]	[illegible]
Software	Node Commander® Windows XP/Vista/7 compatible

Non Functional Attributes

ABSTRACTING THE PHYSICAL DEVICES



A DB VIEW OF SENSOR NETWORKS

WSN TRADITIONAL

- PROCEDURAL ADDRESSING OF INDIVIDUAL SENSOR NODES
  - THE USER SPECIFIES HOW THE TASK IS EXECUTED, DATA IS PROCESSED CENTRALLY

DB-STYLE APPROACH

- DECLARATIVE QUERYING
  - THE USER IS NOT CONCERNED ABOUT “HOW THE NETWORK WORKS” → IN-NETWORK DISTRIBUTED PROCESSING

HOW DIFFERENT ARE THE QUERIES IN PERVASIVE SYSTEMS FROM DB QUERIES?

SENSOR DATA

- TIME STAMPED
- SENSORS DELIVER DATA IN STREAMS
  - CONTINUOUS DATA PRODUCTION
  - OFTEN AT WELL DEFINED TIME INTERVALS
  - NO EXPLICIT REQUEST FOR THAT DATA.
- QUERIES NEED BE PROCESSED IN NEAR- REAL-TIME
  - EXPENSIVE TO SAVE ALL DATA TO DISK
  - DATA STREAMS REPRESENT REAL-WORLD EVENTS WHICH NEED TO BE RESPONDED TO (e.g., traffic accidents and attempted network break-ins),
- NOT ALL SENSOR READINGS ARE OF INTEREST
- UNCERTAIN, INCOMPLETE INFORMATION

QUERY PROCESSING IN WSNs

- WHEN SHOULD SAMPLES FOR A PARTICULAR QUERY BE TAKEN (acquisitional issue)
- WHICH SENSOR NODES HAVE DATA RELEVANT TO A PARTICULAR QUERY (indexing/optimization)
- IN WHAT ORDER SHOULD SAMPLES BE TAKEN AND HOW SHOULD THIS BE INTERLEAVED WITH OTHER OPERATIONS (indexing/optimization)
- IT IS WORTH CONSUMING COMPUTATIONAL POWER and BANDWIDTH TO PROCESS A SAMPLE (stream processing/ approximate answering)

DATA STREAMS AND DSMSs

Data Stream Management Systems (DSMS) are designed to process *unbounded, rapid, time-varying, continuously flowing* streams of data elements, when a store-now- and-process-later approach will not work due to:

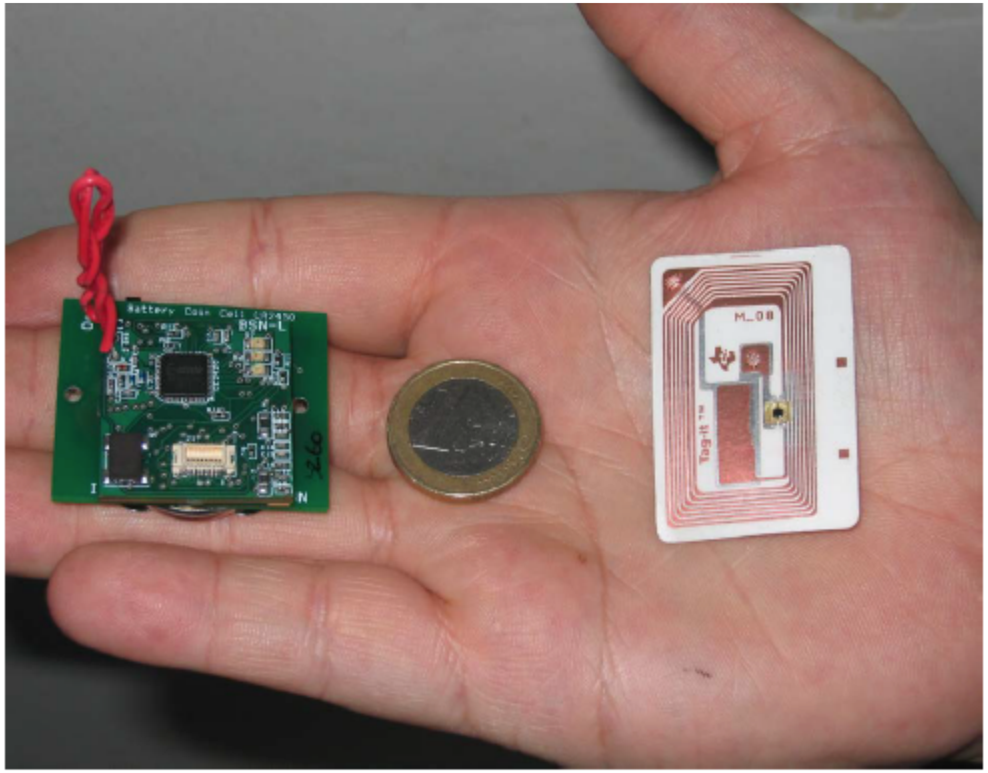
- Response requirements: real time or quasi real-time
- Streams are massive, and also bursty.

However:

- many applications are similar to those of DBMSs.
- most DSMS use some form of SQL
- computing environments quite different
- persistent queries on transient data, vs. transient queries on persistent data
- online filtering of interesting data for later in-depth analysis.

An interesting example: THE PerLa LANGUAGE

A declarative *SQL-like* language



- Data representation and abstraction
- Physical device management

http://perlawsn.sourceforge.net/index.php

PerLa LANGUAGE AND MIDDLEWARE

High Level Interface

LLQ/HLQ/AQ
analyzer and executors

Low Level Interface

SELECT temperature, humidity
WHERE temp>20
SAMPLING EVERY 1h
EXECUTE IF device_id > 2

LLQs: define the behaviour of every device
HLQs: perform SQL operations on sensors streams
AQs: allow to define the behaviour of actuators

The LLI allows for a runtime plug and play integration of heterogeneous sensors. Each device type is wrapped by a customized component built accordingly a device description XML file.

Data analytics: challenges and cautions

Letizia Tanca

Many of these slides are from the Lecture Notes of the book:
Introduction to Data Mining, by Tan,Steinbach and Kumar, pdf at:
https://www.academia.edu/37588575/Introduction-to-Data-Mining.pdf

Making sense of the data: Data Analysis (from Wikipedia)

Data analysis is a process of *inspecting, cleaning, transforming, and modeling* data with the goal of highlighting useful information, suggesting conclusions, and supporting decision making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, in different business, science, and social science domains.

- Data exploration: a preliminary exploration of the data to better understand its characteristics. Now developing into richer paradigms.
- Data mining is a particular data analysis technique that focuses on modeling and knowledge discovery for predictive or descriptive purposes.
- Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is a subset of Artificial Intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so.
- Business intelligence covers data analysis that relies heavily on aggregation, focusing on business information – Data Warehouses.

Analyzing Large Data Sets - Motivation

- Often information is “hidden” in the data but not readily evident
- Human analysts may take weeks to discover useful information
- Much of the data is never analyzed at all

Data exploration

A preliminary exploration of the data
to better understand its characteristics

- Key motivations of data exploration include
  - Helping to select the right tool for preprocessing or analysis
  - Making use of humans' abilities to recognize patterns
    - People can recognize patterns not captured by data analysis tools

- Related to the area of Exploratory Data Analysis (EDA)
  - Created by statistician John Tukey
  - Seminal book is Exploratory Data Analysis by Tukey
  - A nice online introduction can be found in Chapter 1 of the NIST Engineering Statistics Handbook

http://www.itl.nist.gov/div898/handbook/index.htm

Techniques Used In Data Exploration

- In EDA, as originally defined by Tukey
  - The focus was on visualization
  - Clustering and anomaly detection were viewed as exploratory techniques
  - In data mining, clustering and anomaly detection are major areas of interest, and not thought of as just exploratory

- Basic traditional techniques of data exploration
  - Summary statistics
  - Visualization
  - Online Analytical Processing (OLAP)

Summary Statistics

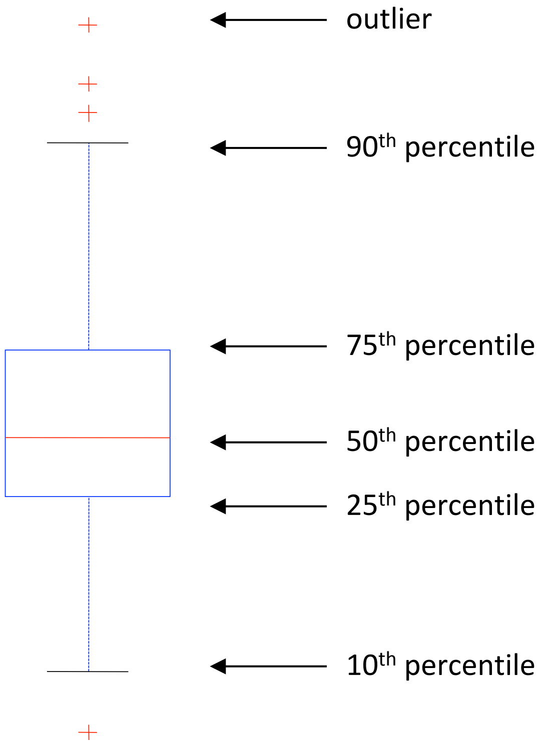
Summary statistics are numbers that summarize properties of the data

- Summarized properties include frequency, location and spread. Examples:
  - location – mean
  - spread - standard deviation
- Most summary statistics can be calculated in a single pass through the data

Frequency and Mode

- The frequency of an attribute value is the percentage of times the value occurs in the data set
  - For example, given the attribute 'gender' and a representative population of people, the gender 'female' occurs about 50% of the times.
- The *mode* of an attribute is the *most frequent attribute value*
- The notions of frequency and mode are typically used with *categorical data*

Percentiles



- For continuous data (and in general for *ordered* data), the notion of a *percentile* is more useful.
- Given an ordinal or continuous attribute *x* and a number *p* between 0 and 100, the *p-th percentile* is a value x_p of x such that *p%* of the observed values of x are less than x_p
- For instance, the 50th percentile $x_{50\%}$ is the value *such that 50% of all values of x are less than* $x_{50\%}$

Given an ordinal or continuous attribute *x* and a number *p* between 0 and 100, the *p-th percentile* is a value Xp of x such that *p%* of the observed values of x are less than Xp

Measures of Location: Mean and Median

- The *mean* is the most common measure of the location of an ordered set of points.
- However, the mean is very sensitive to *outliers.*

$$\text{mean}(x) = \overline{x} = \frac{1}{m}\sum_{i=1}^{m} x_i$$

- Thus, the *median* or a *trimmed mean* is also commonly used.

$$\text{median}(x) = \begin{cases} x_{(r+1)} & \text{if } m \text{ is odd, i.e., } m = 2r+1 \\ \frac{1}{2}(x_{(r)} + x_{(r+1)}) & \text{if } m \text{ is even, i.e., } m = 2r \end{cases}$$

Measures of Spread: Range and Variance

- *Range* is the difference between the max and min
- The *variance* or the *standard deviation* are the most common measures of the spread of a set of points

$$\text{variance}(x) = s_x^2 = \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \overline{x})^2$$

- The *standard deviation* s_x is the square root of the variance.
- However, this is also sensitive to outliers, so that other measures are often used.

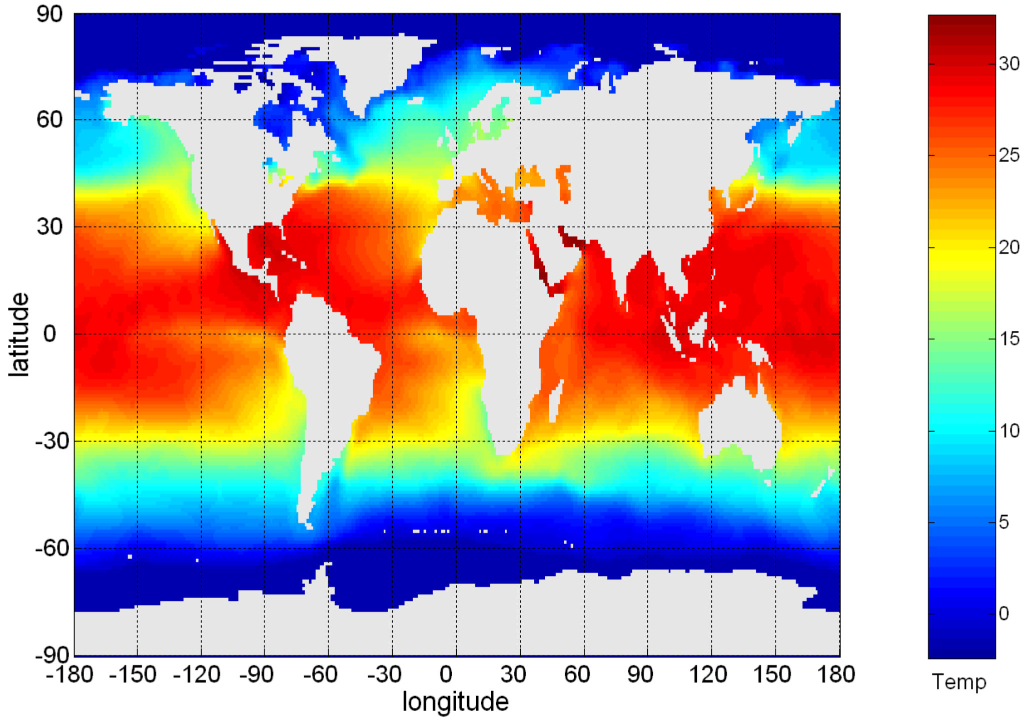
Visualization

- Visualization is the conversion of data into a visual or tabular format so that the characteristics of the data and the relationships among data items or attributes can be analyzed or reported.

- Visualization of data is one of the most powerful and appealing techniques for data exploration.
  - Humans have a well developed ability to analyze large amounts of information that is presented visually
  - Can detect general patterns and trends
  - Can detect outliers and unusual patterns

Example: Sea Surface Temperature

- The following shows the Sea Surface Temperature (SST) for July 1982
- Tens of thousands of data points are summarized in a single figure

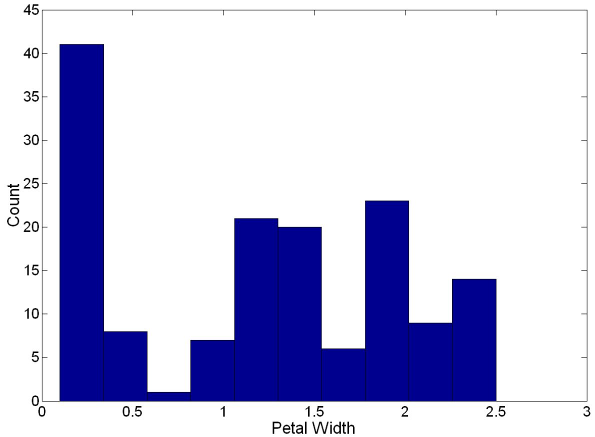


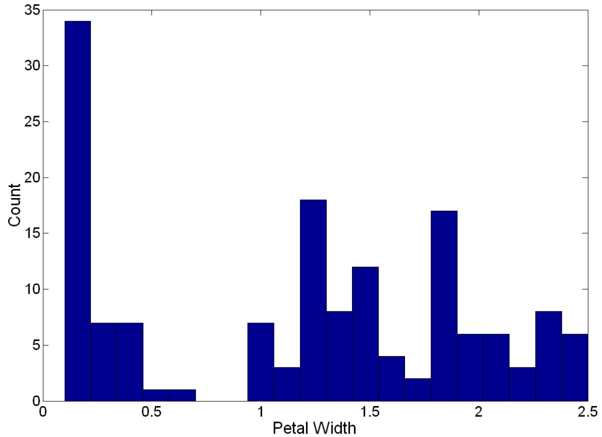
Selection

- Is the elimination or the de-emphasis of certain objects and attributes
- Selection may involve choosing a subset of attributes
  - Dimensionality reduction is often used to reduce the number of dimensions to two or three
  - Alternatively, pairs of attributes can be considered
- Selection may also involve choosing a subset of objects
  - A region of the screen can only show so many points
  - Can sample, but want to preserve points in sparse areas

Visualization Techniques: Histograms

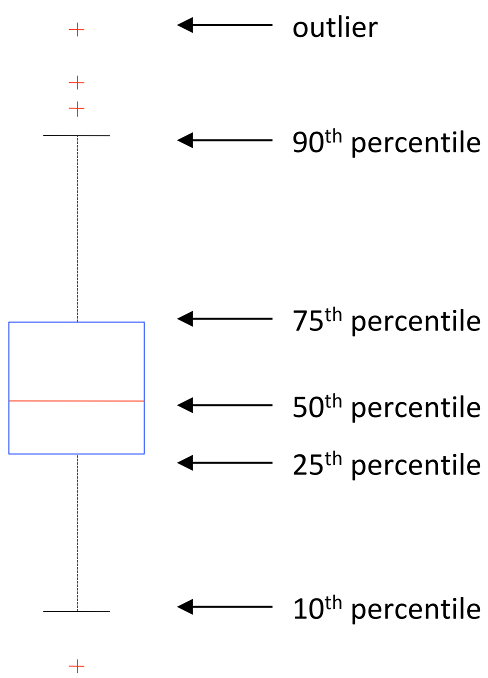
- Histogram
  - Usually shows the distribution of values of a single variable
  - Divide the values into bins and show a bar plot of the number of objects in each bin.
  - The height of each bar indicates the number of objects
  - Shape of histogram depends on the number of bins
- Example: Petal Width (10 and 20 bins, respectively)





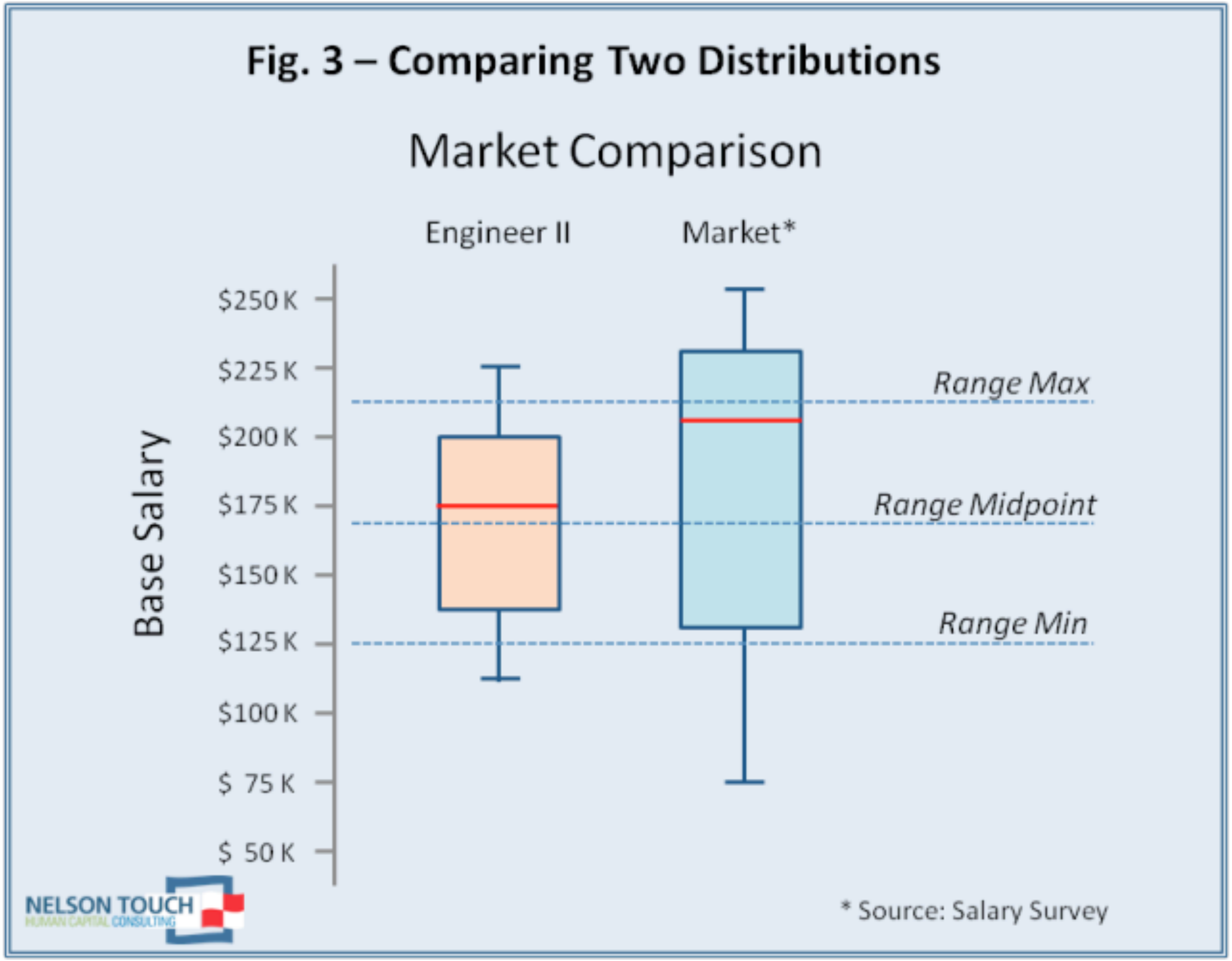
Visualization Techniques: Box Plots

- Box Plots
  - Invented by J. Tukey
  - Another way of displaying the distribution of data and the percentiles



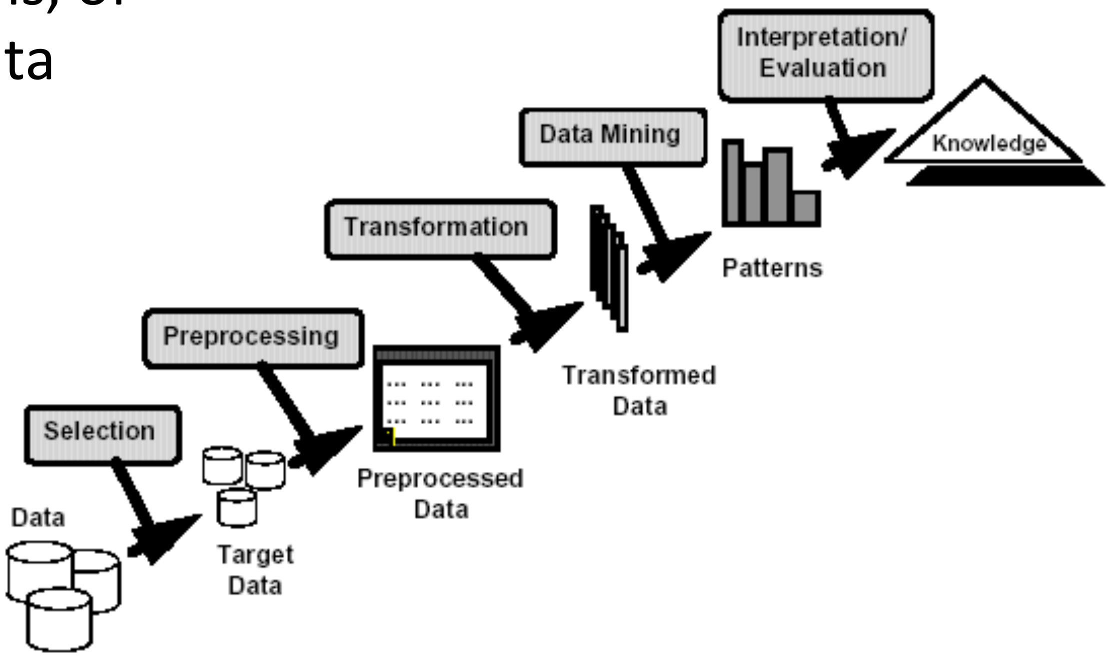
Given an ordinal or continuous attribute *x* and a number *p* between 0 and 100, the *p-th percentile* is a value Xp of x such that *p*% of the observed values of x are less than Xp

Example of Box Plots



Data Mining

- Many Definitions
  - Non-trivial extraction of implicit, previously unknown and potentially useful information from data
  - Exploration & analysis, by automatic or semi-automatic means, of large quantities of data in order to discover meaningful patterns



What is (not) Data Mining?

What is not Data Mining?

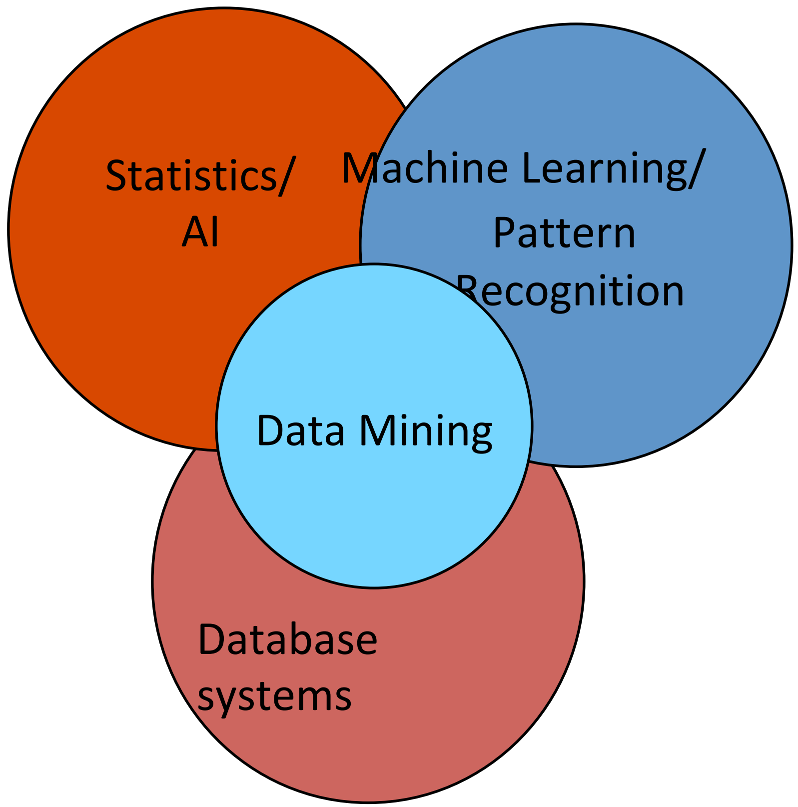
- Look up phone number in phone directory
- Query a Web search engine for information about "Amazon"

What is Data Mining?

- Certain names are more prevalent in certain US locations (O'Brien, O'Rurke, O'Reilly… in Boston area)
- Group together similar documents returned by search engine according to their content (e.g. all financial newspaper articles)

Origins of Data Mining

- Draws ideas from machine learning/AI, pattern recognition, statistics, and database systems
- Traditional Techniques may be unsuitable due to
  - Enormity of data
  - High dimensionality of data
  - Heterogeneous, distributed nature of data



Data Mining Tasks

- Prediction Methods
  - Use some variables to predict unknown or future values of other variables.
- Description Methods
  - Find human-interpretable patterns that describe the data.

We give examples of some data mining tasks

From [Fayyad, et.al.] Advances in Knowledge Discovery and Data Mining, 1996

Methods

- Classification [Predictive]
- Clustering [Descriptive]
- Itemset Discovery [Descriptive]
- Association Rule Discovery [Descriptive]
- Anomaly Detection [Predictive]

More tasks (not described here):

- Sequential Pattern Discovery [Descriptive]
- Regression [Predictive]

Classification: Definition

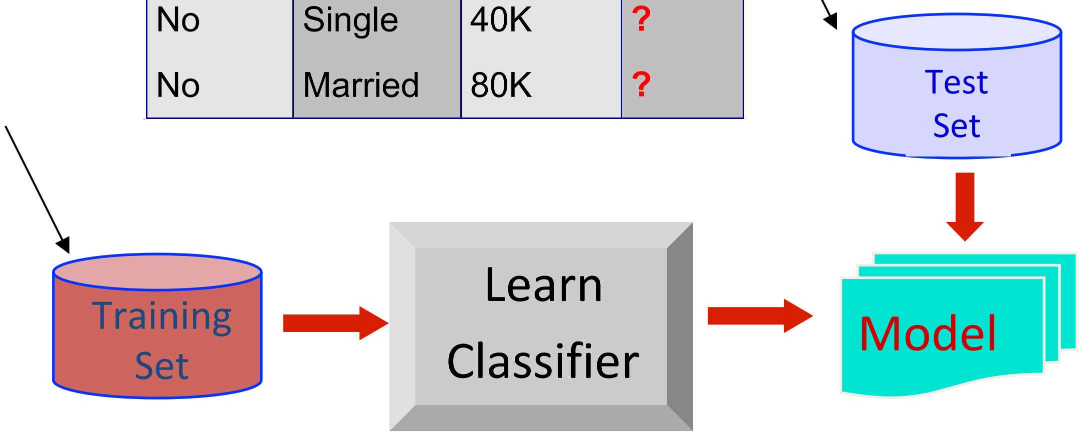
- Given a collection of records (*training set*)
  - Each record contains a set of *attributes*, one of the attributes is the *class*.
- Find a *model* for class attribute as a function of the values of other attributes.
- Goal: previously unseen records should be assigned a class as accurately as possible.
  - A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

Classification Example

categorical categorical continuous class

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Refund	Marital Status	Taxable Income	Cheat
No	Single	75K	?
Yes	Married	50K	?
No	Married	150K	?
Yes	Divorced	90K	?
No	Single	40K	?
No	Married	80K	?



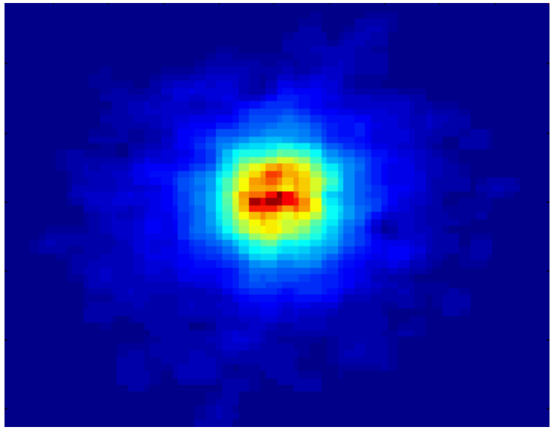
Classification: Fraud Detection

- Goal: Predict fraudulent cases in credit card transactions.
- Approach:
  - Use credit card transactions and the information on its account-holder as attributes.
    - When does a customer buy, what does he buy, how often he pays on time, etc
  - Label past transactions as fraud or fair transactions. This forms the class attribute.
  - Learn a model for the class of the transactions.
  - Use this model to detect fraud by observing credit card transactions on an account.

Classifying Galaxies

Courtesy: http://aps.umn.edu

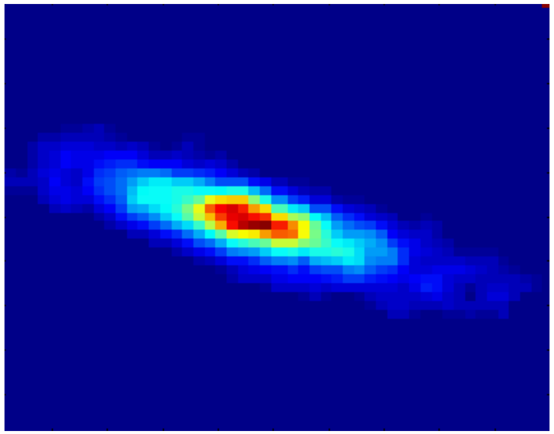
Early



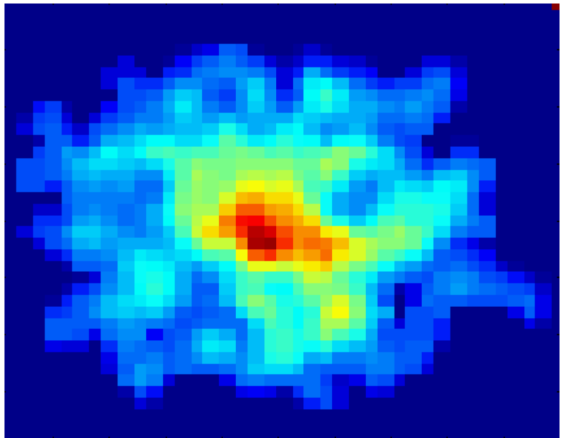
Class:
- Stages of Formation

Attributes:
- Image features,
- Characteristics of light waves received, etc.

Intermediate



Late



Data Size:
- 72 million stars, 20 million galaxies
- Object Catalog: 9 GB
- Image Database: 150 GB

Clustering Definition

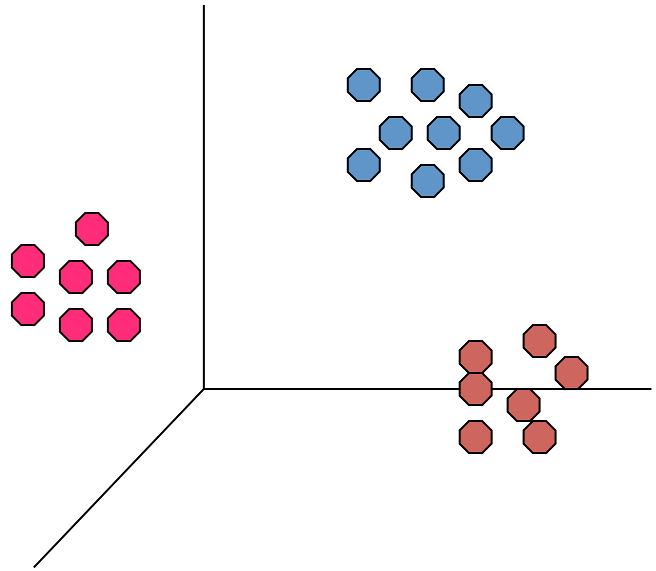
- Given a set of data points, each having a set of attributes, and a similarity measure among them, find clusters such that
  - Data points in one cluster are more similar to one another.
  - Data points in separate clusters are less similar to one another.
- Similarity Measures:
  - Euclidean Distance if attributes are continuous.
  - Other, Problem-specific Measures.

Illustrating Clustering

Euclidean Distance Based Clustering in 3-D space.

Intracluster distances are minimized

Intercluster distances are maximized



Clustering:
Market Segmentation

- Goal: subdivide a market into distinct subsets of customers so that a subset may be selected as a market target to be reached *with a distinct marketing mix*.
- Approach:
  - Collect different attributes of customers based on their geographical and lifestyle related information.
  - Find clusters of similar customers.
  - Measure the clustering quality by observing buying patterns of customers in same cluster vs. those from different clusters.

Document Clustering

- Goal: To find groups of documents that are similar to each other based on the important terms appearing in them.
- Approach: To identify *frequently occurring terms in each document*. Form a similarity measure based on the frequencies of different terms. Use it to cluster.
- Gain: Information Retrieval can utilize the clusters to relate a new document or search term to clustered documents.

Document Clustering

- Clustering Points: 3204 Articles of Los Angeles Times.
- Similarity Measure: How many words are common in these documents (after some word filtering).

Category	Total Articles	Correctly Placed
Financial	555	364
Foreign	341	260
National	273	36
Metro	943	746
Sports	738	573
Entertainment	354	278

Association Rule Discovery

- Given a set of records each of which contains some number of items from a given collection;
  - Produce dependency rules which will predict occurrence of an item based on occurrences of other items.

TID	*Items*
1	**Bread, Coke, Milk**
2	**Beer, Bread**
3	**Beer, Coke, Diaper, Milk**
4	**Beer, Bread, Diaper, Milk**
5	**Coke, Diaper, Milk**

Rules Discovered:
{Milk} --> {Coke}
{Diaper, Milk} --> {Beer}

Frequent Itemsets

- **Itemset**
  - A collection of one or more items
    - Example: {Milk, Bread, Diaper}
  - k-itemset
    - An itemset that contains k items
- **Support**
  - Fraction of transactions that contain an itemset
  - E.g. s({Milk, Bread, Diaper}) = 2/5
- **Frequent Itemset**
  - An itemset whose support is greater than or equal to a *minsup* threshold

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Caution: here the word *transaction* has a different meaning w.r.t. a *database transaction*

Association Rule

- Association Rule
  - An expression of the form X → Y, where X and Y are itemsets (!!! The arrow does NOT represent logical implication !!!)
  - Example:
    {Milk, Diaper} → {Beer}

- Rule Evaluation Metrics
  - Support (s)
    - Fraction of transactions that contain both X and Y
  - Confidence (c)
    - Measures how often items in Y appear in transactions that contain X

TID	*Items*
1	**Bread, Milk**
2	**Bread, Diaper, Beer, Eggs**
3	**Milk, Diaper, Beer, Coke**
4	**Bread, Milk, Diaper, Beer**
5	**Bread, Milk, Diaper, Coke**

Example:

$$\{\text{Milk}, \text{Diaper}\} \Rightarrow \text{Beer}$$

$$s = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{|\text{T}|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{\sigma(\text{Milk}, \text{Diaper})} = \frac{2}{3} = 0.67$$

Association Rule Discovery
Marketing and Sales Promotion

Let the rule discovered be

{Bagels, … } --> {Potato Chips}

- <u>Potato Chips as consequent</u> => Can be used to determine what should be done to boost its sales.
- <u>Bagels in the antecedent</u> => Can be used to see which products would be affected if the store discontinues selling bagels.
- <u>Bagels in antecedent *and* Potato chips in consequent</u> => Can be used to see what products should be sold with Bagels to promote sale of Potato chips!

Association Rule Discovery
Supermarket shelf management

- Goal: To identify items that are bought together by sufficiently many customers.
- Approach: Process the point-of-sale data collected with barcode scanners to find dependencies among items.
- A classical rule --
  - If a customer buys diaper and milk, then he is very likely to buy beer.
  - So, don't be surprised if you find six-packs stacked next to diapers!

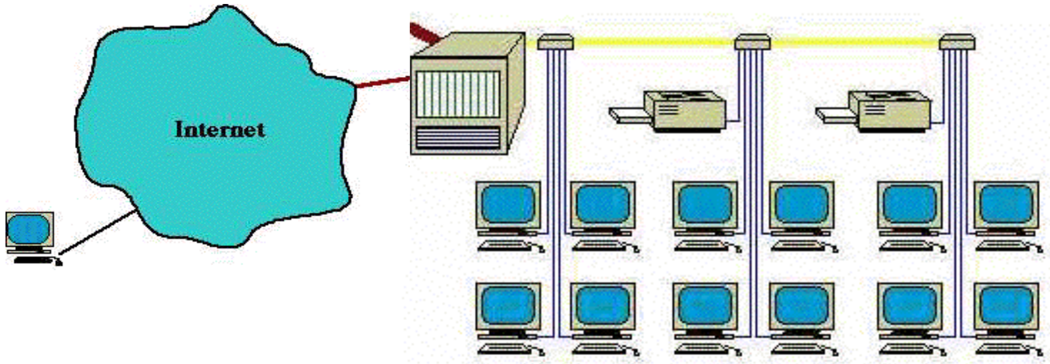
Deviation/Anomaly Detection

What are anomalies/ outliers?

→ The set of data points that are "considerably different" from the remainder of the data

Application: detect significant deviations from normal behavior, e.g.

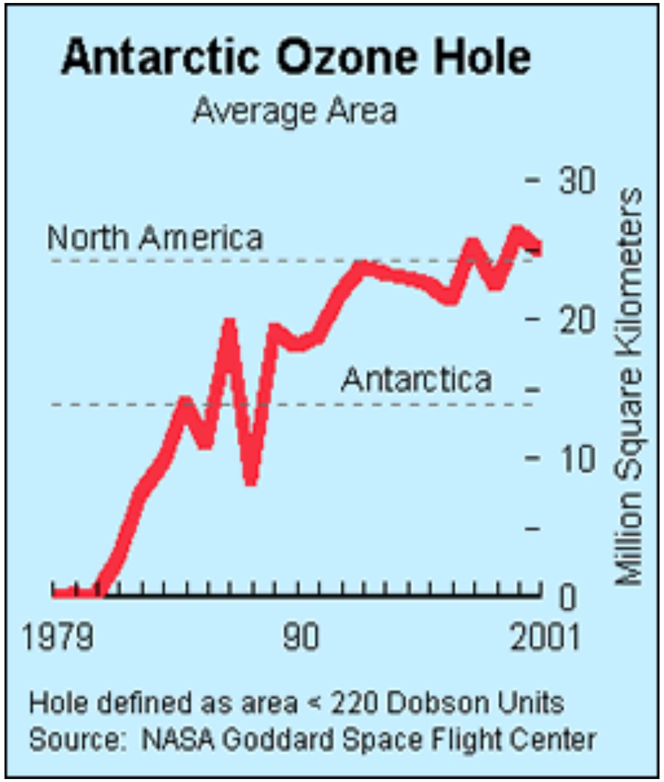
- Credit Card Fraud Detection
- Network Intrusion Detection



Importance of Anomaly Detection

Ozone Depletion History

- In 1985 three researchers (Farman, Gardinar and Shanklin) were puzzled by data gathered by the British Antarctic Survey showing that ozone levels for Antarctica had dropped 10% below normal levels
- Why did the Nimbus 7 satellite, which had instruments aboard for recording ozone levels, not record similarly low ozone concentrations?
- The ozone concentrations recorded by the satellite were so low they were being treated as outliers by a computer program and discarded!



Sources:
http://exploringdata.cqu.edu.au/ozone.html
http://www.epa.gov/ozone/science/hole/size.html

Challenges of Data Mining

- Scalability
- Dimensionality
- Complex and Heterogeneous Data
- Data Quality
- Data Ownership and Distribution
- Privacy Preservation
- Streaming Data

Knowledge discovery and reasoning

- The other typical discipline for knowledge discovery and data analysis is Machine Learning (AI) – we do not deal with it in this course
- Machine learning is the part of Artificial Intelligence that deals with inductive reasoning, i.e. the possibility *to derive conclusions by induction*
- The other great area of AI is deductive reasoning, that is, *deriving conclusions by logical proofs*
- In fact, acquiring knowledge involves more than analytics, that is, a mix of the two activities of induction and deduction that ultimately mimics human reasoning (e.g. using ontologies)

OLAP and Data Warehouses

- On-Line Analytical Processing (OLAP) was proposed by E. F. Codd, the father of the relational database
- Relational databases put data into tables, while OLAP uses a multidimensional array representation
- Such representations of data previously existed in statistics and other fields
- There are a number of data analysis and data exploration operations that are easier with such a data representation.
- However the typical representation used in today's systems is a "simulation" of multidimensionality in relational databases: ROLAP

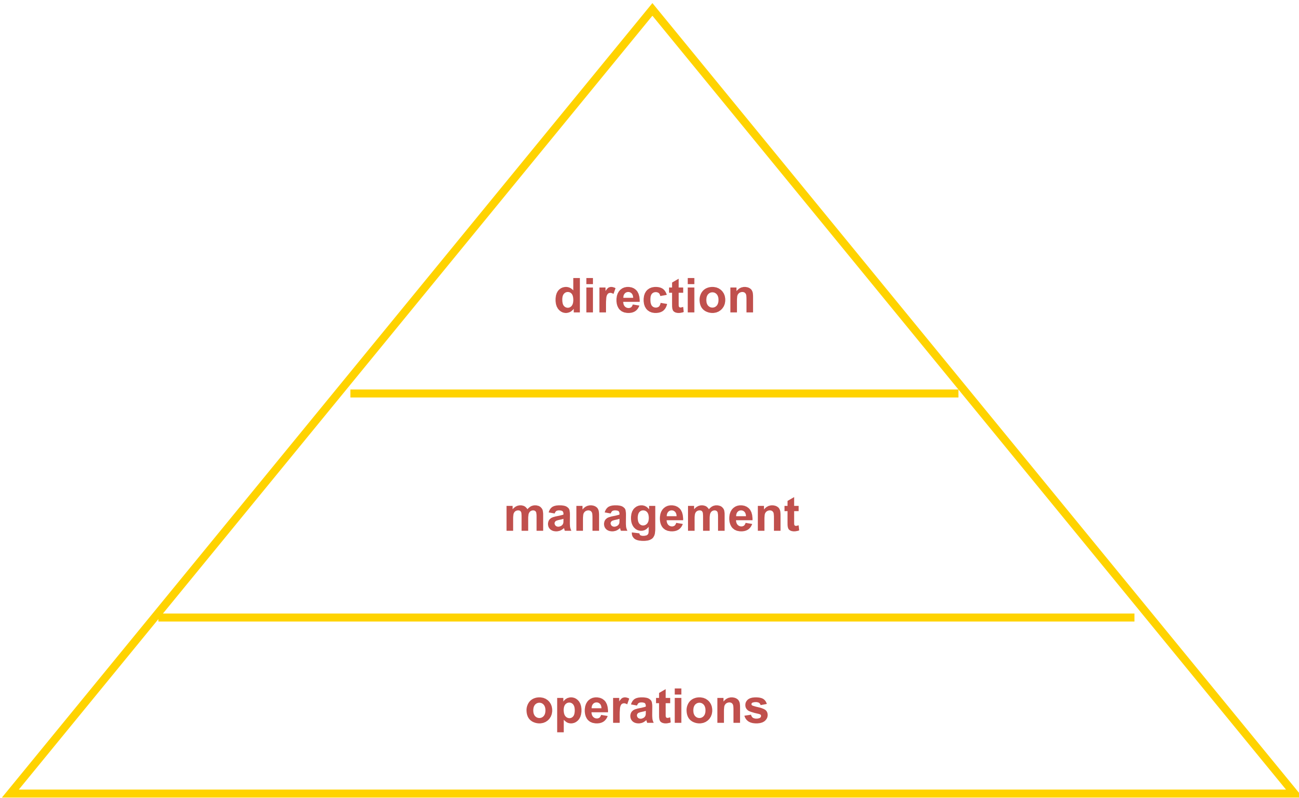
What is a Data Warehouse

- Data should be integrated across the enterprise(s)
- Summary data provide real value to the organization
- Historical data hold the key to understanding data over time
- What-if capabilities are required

A single, complete and consistent store of data obtained from a variety of different sources made available to end users, *so that they can understand and use it in a business context.*

[Barry Devlin]

Business Processes' Pyramid



Data Warehouse

- A Data Warehouse is a
  - subject-oriented,
  - integrated,
  - time-varying,
  - non-volatile

collection of data that is used primarily in organizational decision making.

[Bill Inmon, Building the Data Warehouse, 1996]

DW is a specialized DB

Standard (Transactional) DB (OLTP)

- Mostly updates
- Many small transactions
- Gb – Tb (10^9- 10^{12} bytes) of data
- Current snapshot
- Index/hash on p.k.
- Raw data
- Thousands of users (e.g., clerical users)

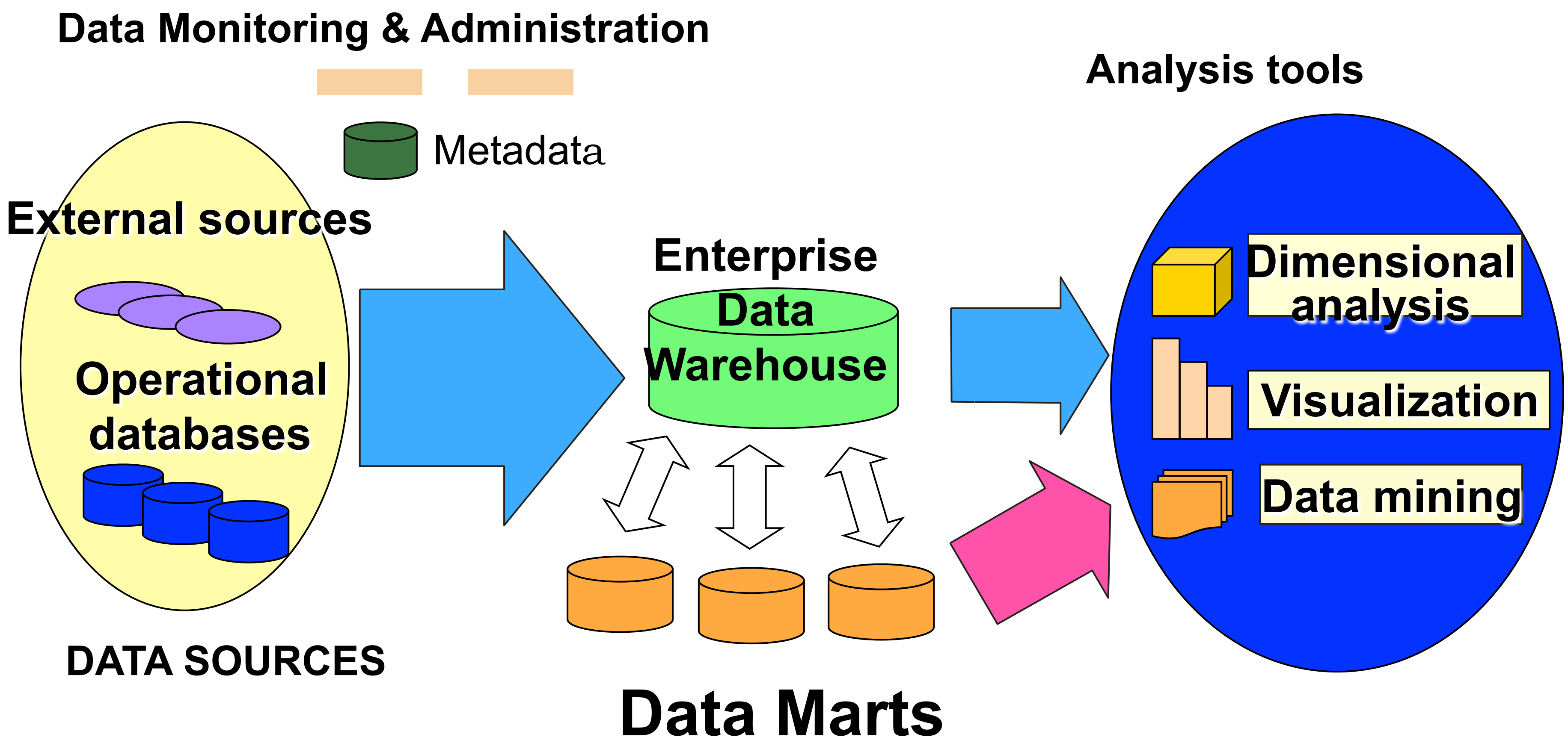
Warehouse (OLAP)

- Mostly reads
- Queries are long and complex
- Tb – Pb (10^{12}- 10^{15} bytes) of data
- History
- Lots of scans
- Summarized, reconciled data
- Hundreds of users

Where is a DW useful

- Commerce: sales and complaints analysis, client fidelization, shipping and stock control
- Manufacturing plants: production cost control, provision and order support
- Financial services: risk and credit card analysis, fraud detection
- Telecommunications: call flow analysis, subscribers' profiles
- Healthcare structures: patients' ingoing and outgoing flows, cost analysis

Architecture for a data warehouse



By contrast: recall Data Lakes

- (Gartner) A Data Lake is a concept consisting of a collection of storage instances of various data assets. These assets are stored in a near-exact, or even exact, copy of the source format and are in addition to the originating data stores.
- A Data Lake is a storage repository that holds a vast amount of raw data in its native format until it is needed.
- A Data Lake contains all data, both raw sources over extended periods of time as well as any processed data. The purpose of a Data Lake is to enable users across multiple business units to refine, explore and enrich data on their terms

Examples of data warehouse queries

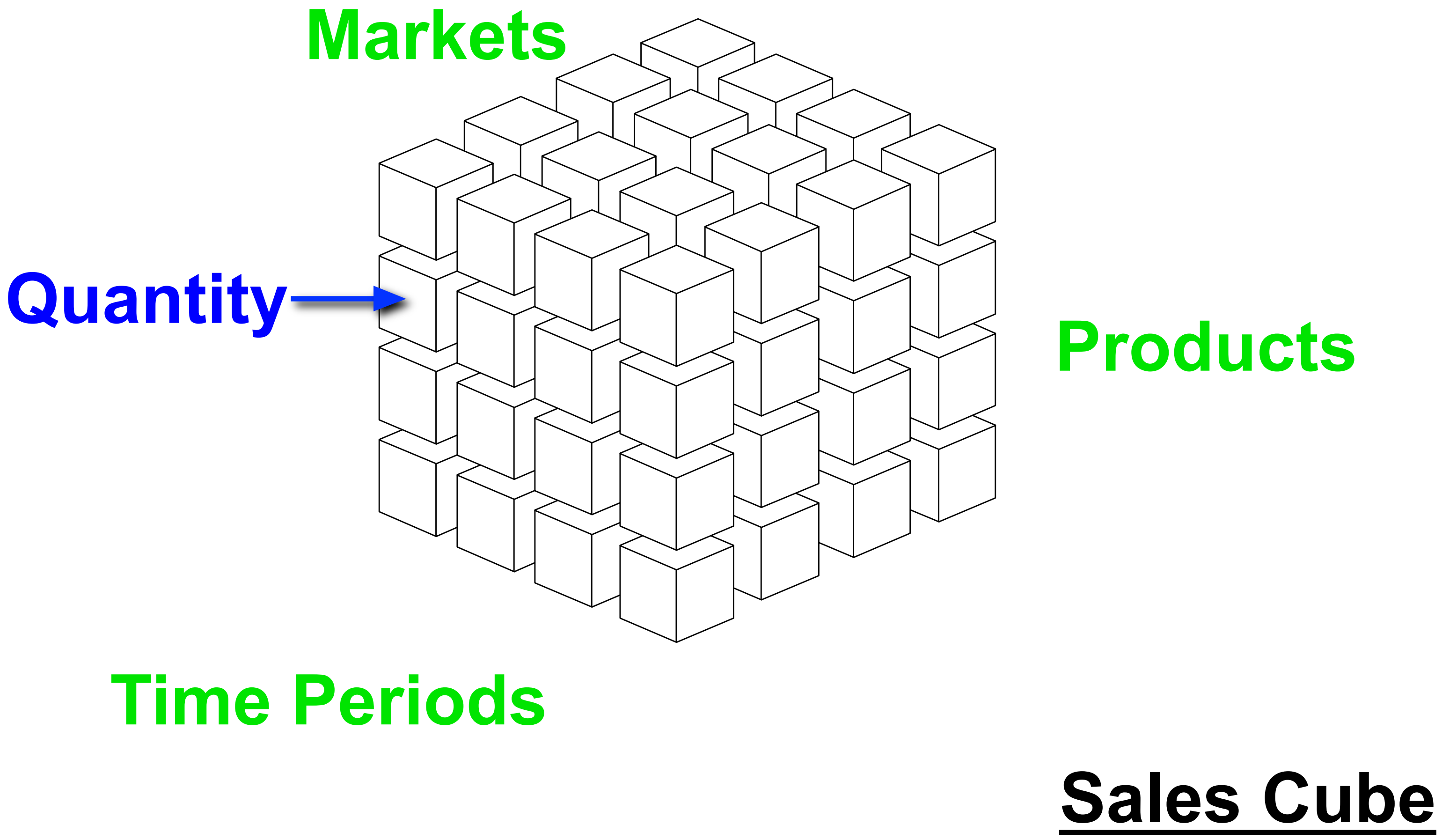
- Show total sales across all products at increasing aggregation levels for a geography dimension, from state to country to region, for 1999 and 2000.
- Create a cross-tabular analysis of our operations showing expenses by territory in South America for 1999 and 2000. Include all possible subtotals.
- List the top 10 sales representatives in Asia according to sales revenue for automotive products in year 2000, and rank their commissions.

OLAP-oriented metaphor

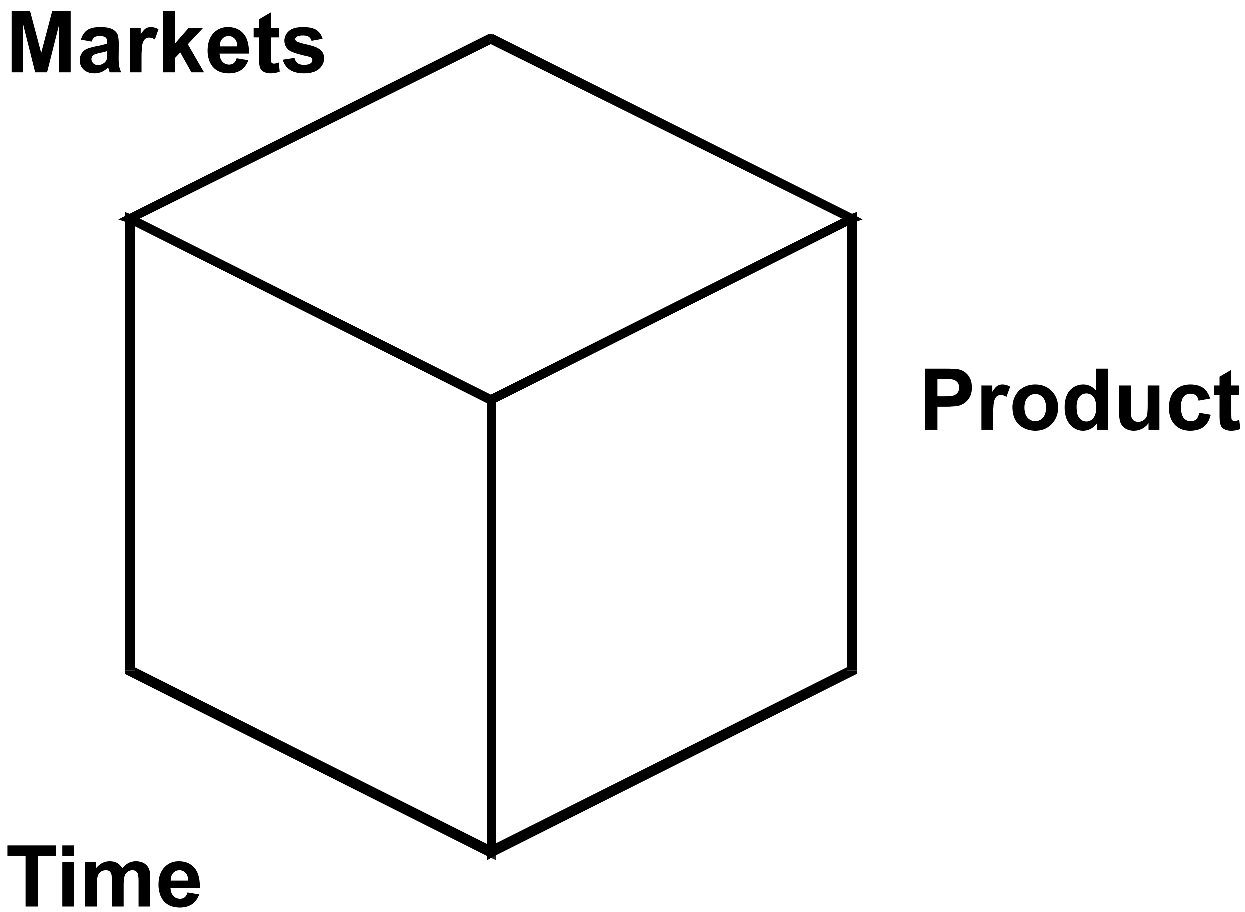
- Must support sophisticated analyses and computations over different dimensions and hierarchies
- Most appropriate data model: data cube
- Cube dimensions are the search keys
- Each dimension may be hierarchical
  - DATE {DAY-MONTH-TRIMESTER-YEAR}
  - PRODUCT {BRAND - TYPE - CATEGORY}

    (e.g. LAND ROVER - CARS - VEHICLES)
- Cube cells contain metric values

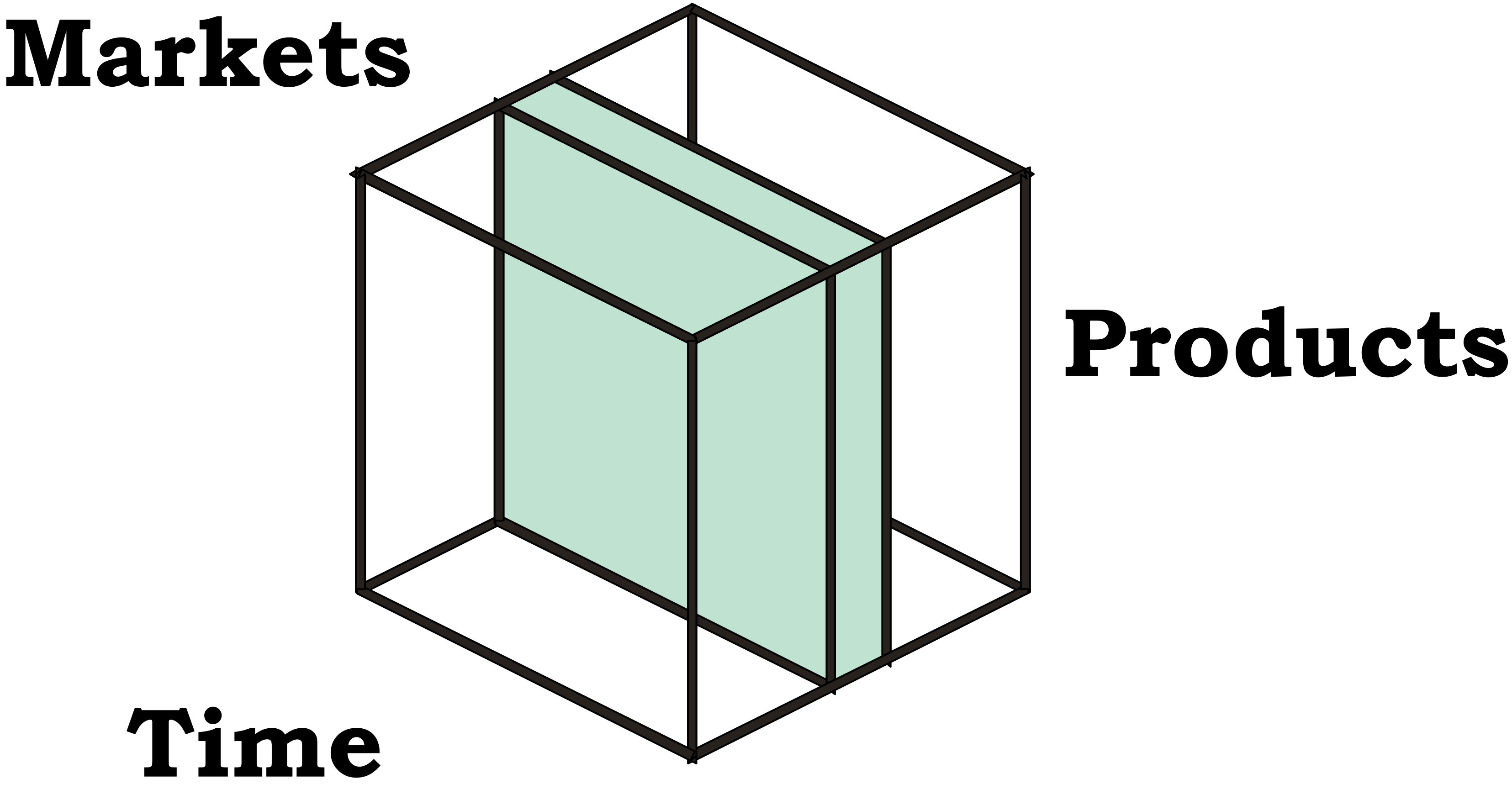
Multidimensional Representation: a LOGICAL MODEL for OLAP



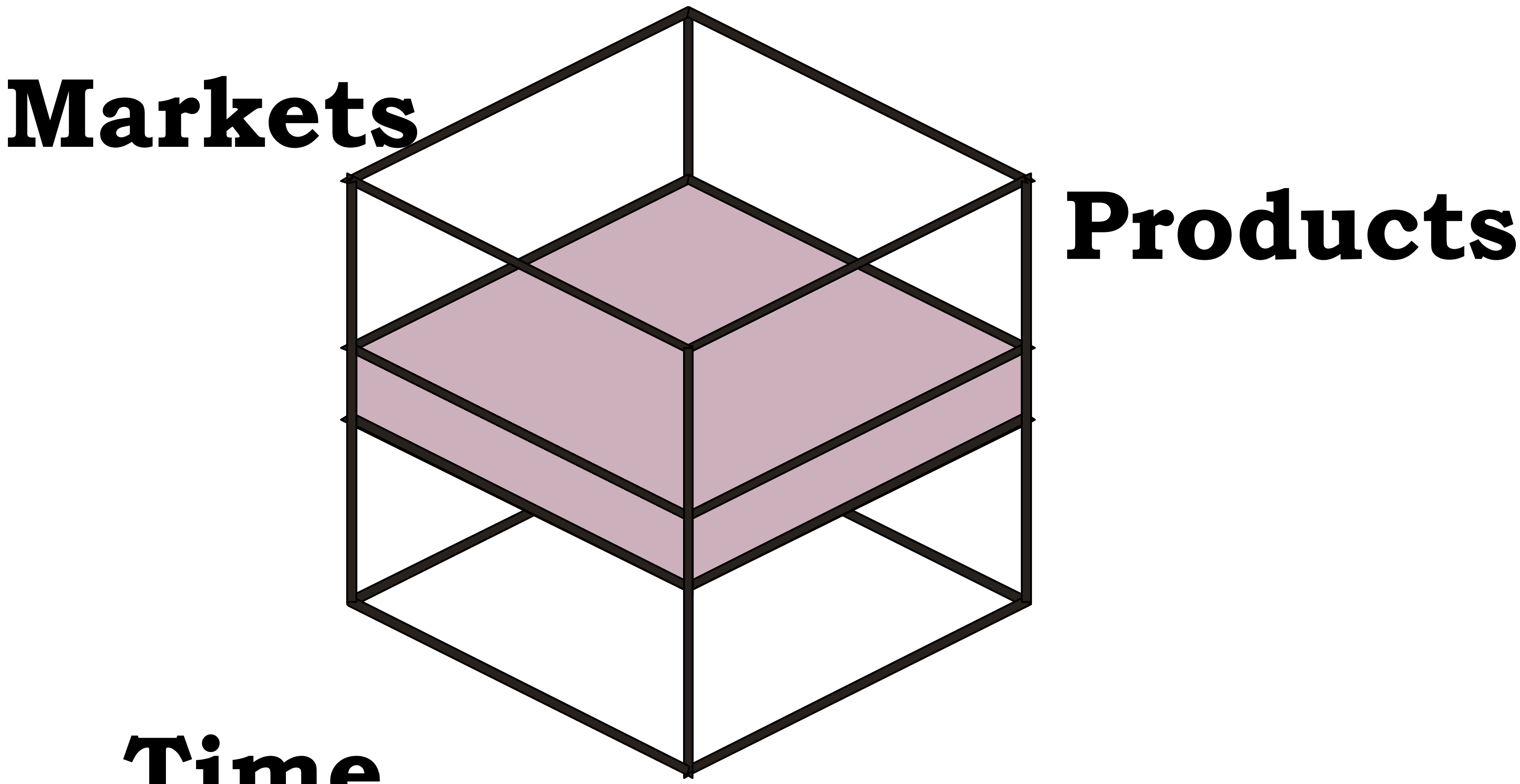
Multidimensional data views



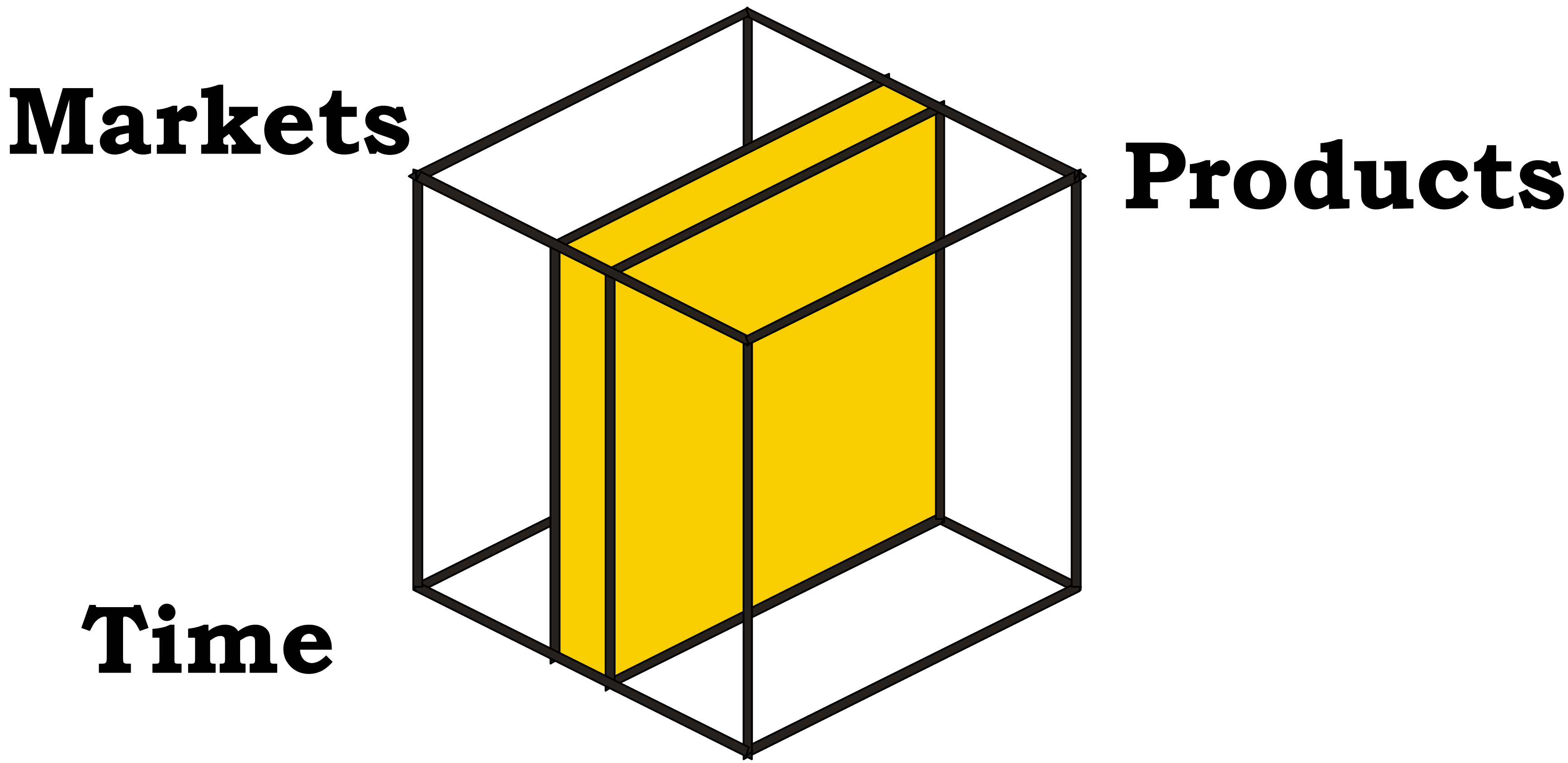
The area manager examines product sales
of his/her own markets



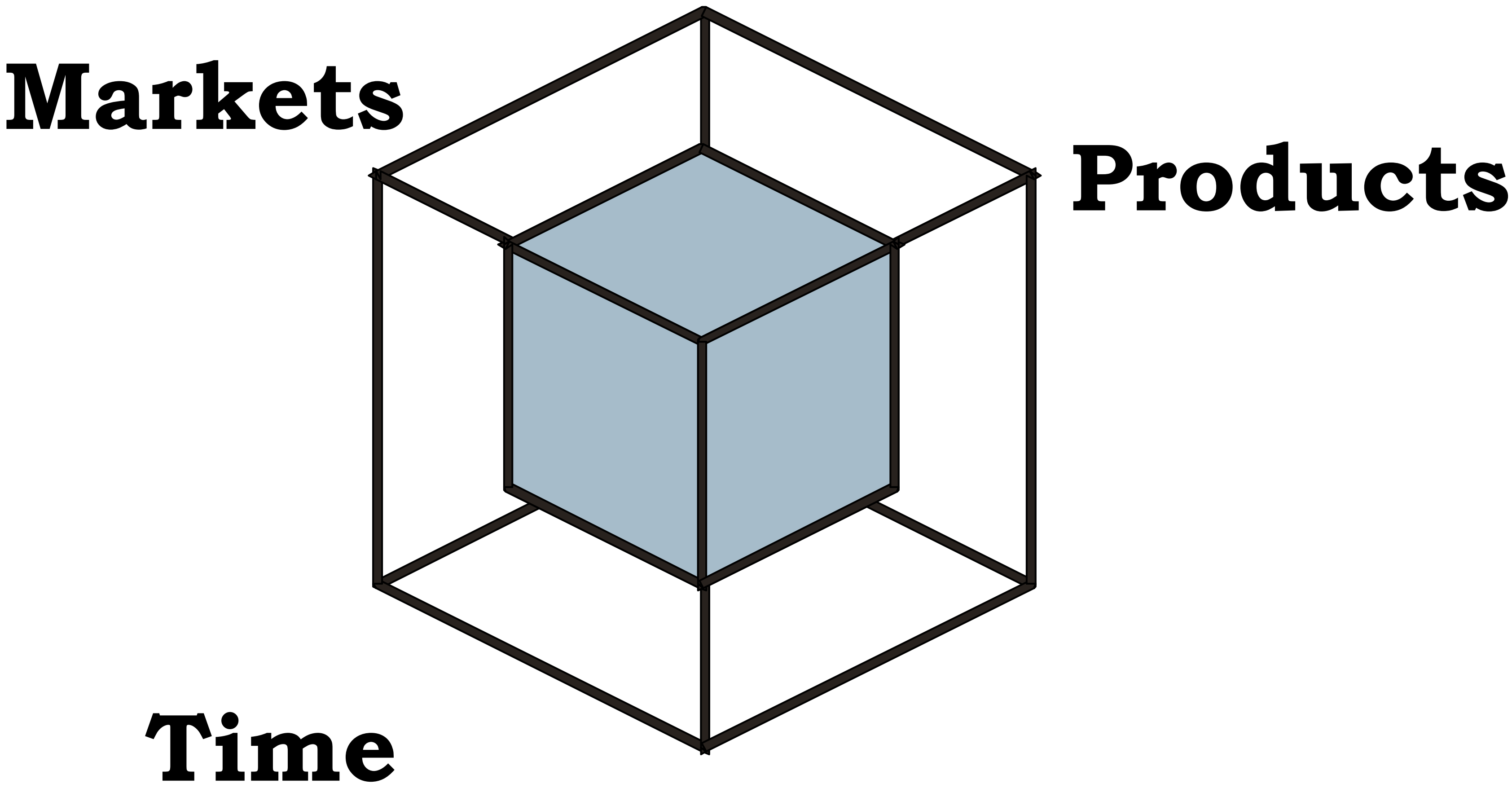
Product manager examines the sales of a specific product in all periods and in all markets



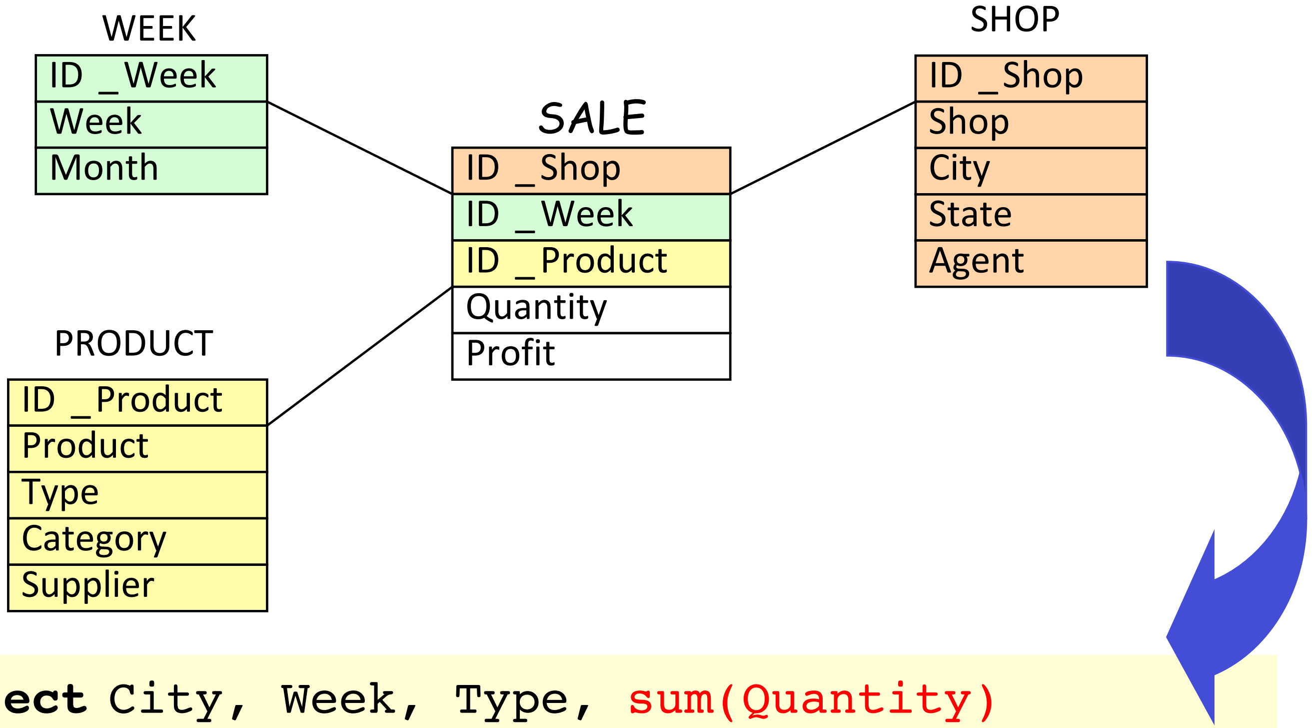
Financial manager examines product sales in all markets, for the current period and the previous one



The strategic manager concentrates on a category of products, a specific region and a medium time span



DW queries on a Data Cube




```
select City, Week, Type, sum(Quantity)
from   Week, Shop, Product, Sale
where  Week.ID_Week=Sale.ID_Week and
    Shop.ID_Shop=Sale.ID_Shop and
    Product.ID_Product=Sale.ID_Product and
    Product.Category = 'FoodStuff'
group by City,Week,Type
```

References on DWs

- M. Golfarelli, S. Rizzi: Data Warehouse Design: Modern Principles and Methodologies By Matteo Golfarelli, Stefano Rizzi, pdf at https://aahmedia.press/med_94898/0071610391

- Ralph Kimball: The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses John Wiley 1996.

Ethics in Data Management

- As data have an impact on almost every aspect of our lives, it is more and more important to understand the nature of this effect
- With search and recommendation engines, the web can influences our lives to a great extent, e.g. by recommending interesting jobs only to white males, discriminating as an effect of biased data or algorithms
- With statistics used everywhere, it may happen that very critical decisions be taken without taking their ethical consequences into account

Ethics
in Data Management

It is up to the data scientists to

- identify which datasets can genuinely help answering some given question
- understand their contents
- choose the most appropriate *knowledge extraction technique* (search, query, or data analysis methods) to obtain a fair result

This sequence of choices may strongly influence the process, and biased results might be obtained.

Something is happening already

- Traditional knowledge extraction systems, including database systems, search engines, data warehouses, etc., hardly pay specific attention to ethically sensitive aspects of knowledge extraction processes and their outcomes.
- Such aspects are now becoming prominent, especially with regard to the protection of human rights and their consequences in normative ethics. These demands are broadly reflected into codes of ethics for companies and computer professionals, and also in legally binding regulations such as the EU General Data Protection Regulation (GDPR):
  - https://www.eugdpr.org/.
- GDPR unifies data protection laws across all European Union members, defining a comprehensive set of rights for EU citizens, describing the requirements for companies and organizations for collecting, storing, processing and managing personal data.
- Following a 2-year post-adoption grace period, the GDPR has become fully enforceable throughout the European Union in May 2018.

Conclusion

- Transforming (all sorts of) data into knowledge: a set of ever-challenging topics
- On the side of the V's of Big Data, let us consider a double one, for WISDOM (*): not only we want to make sense of the data, but we should extract from them a worth that makes us "wiser", doubling their Value.
- Lots of work still to be done
- More questions?

(*) In Italian the letter W is called "double V", instead of "double U" as in English