Politecnico di Milano

Technologies for Information Systems

Questions
a.a. 2018-2019

Author
Tommaso Scarlatti

January 15, 2019

1 Data Integration

Describe the main differences between materialized and virtual data integration, explaining which of these two approaches is used in the case of Data Warehousing and why.

Data integration process aims at combining data coming from different data sources, providing the user with a unified vision of the data. Data has four interesting dimensions to be considered: Volume, Variety, Velocity and Veracity. The are two relevant ways of integrating Database Systems:

- **Materialized Integration:** data are merged in a new database. This is the approach adopted in data warehouses, and the software to access, scrape, transform, and load data into warehouses, became known as extract, transform, and load (ETL) systems. ETL is performed periodically, building an history of the enterprise, allowing systematic or ad-hoc data analysis and mining. This approach is not indicated when data needs to be kept up-to-date.
- **Virtual Integration:** this 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.

Describe the conflict analysis step in data integration; describe the most important classes of conflicts and provide a set of examples.

Once the identification of the sources and the reverse engineering of their schemata have been completed, the next step in the data integration process is conflict resolution and reconstructing. It starts with the identification of related concepts across sources: a simple process if manual but very difficult to be automatized. The most important classes of conflicts are the following:

- ***Name:*** such as homonyms and synonyms, for instance price that means standard and discounted, or employee/agent.
- ***Type:*** when the same attribute has different types, for instance M/F or 0/1. It can be also two different entities that represent the same real concept with different attributes.
- ***Data Semantics:*** different currencies or measures or different granularities of the same measure.
- ***Structure:*** when the same concept is represented with an attribute in one source and with an entity in another.
- ***Cardinality:*** same relation between entities but with different cardinalities.
- ***Key:*** the same entity in different sources has different keys.

Describe the GAV (global as view) and LAV (local as view) approaches used to define the mapping between the global logical schema and the single source schemata in the data integration context. Discuss the main differences between the two approaches and describe under which conditions GAV is more appropriate than LAV, and vice versa.

A data integration system can be identified by a tuple (G, S, M) where G is the global schema, S is the source schema and M is a set of mappings (assertions) between G and S. There are two basic approaches used to define the mapping between the global logical schema and the single source schemata in the data integration context:

- **GAV (Global As View):** the global schema is expressed in terms of the data source schemata. For each element g of G is defined a mapping $g \rightarrow q_S$ as a query over the data sources. This mapping tells us exactly how the element g is computed. This approach is well suited when data sources are stable and is difficult to add new sources since the schema will need to be reconsidered.

- **LAV (Local As View):** in this approach the content of each source is characterized in terms of a view q_G over the global schema G: $s \rightarrow q_G$. This approach is applicable when the global schema is stable, for instance when it is based on a domain ontology or an enterprise model. Furthermore, this model favours extendibility but it also leads to a much more complex query processing.

2 Semistructured Data

Define Wrappers and Mediators, explain in which circumstances their use is advised in Data Integration and the way they work. Discuss the various types of Wrappers and Mediators that have been introduced during the course.

Often the Data Integration process has to deal with multiple, distributed and heterogeneous data sources which can include semi-structured or unstructured data. For the former, data have some form of structure, but is not prescriptive, regular or complete as in traditional DBMS.

The final aim is to integrate, query and compare data from multiple sources and with different structure just as if they were all structured. For this purpose, we rely on Wrappers and Mediators:

- **Wrappers:** are modules binded to the sources which are in charge to translate an incoming query in one or more queries which are understandable for the specific source (in this way they can also extend its query possibilities). They also convert back the query's answer in the correct format for the requesting application. Human-based maintenance of an ad-hoc wrapper is very expensive, but for some applications there are automatic wrapper generators.

- **Mediators:** are interfaces specialized in a certain domain which stand between application and wrappers. In one way it accepts queries written in the application's language, decomposes them, and send them to each specific wrapper. On the other hand it is also in charge of gathering all the responses and send them back to the application, providing a unified vision of data.
  The term mediation has a broad meaning: it includes the processing needed to make the interfaces work, the underlying knowledge structure which guides data transformations and also any intermediate storage which is needed.

During the course we have seen the architecture of **TSIMMIS** project, which relies on the usage of both wrappers and mediators. In TSIMMIS, queries are posed to the mediators in a specific object-oriented language called LOREL. The data model adopted is OEM (Object Exchange Model), a graph-based and self-descriptive model, since it represents directly data with no schema at all. The model is completely managed by the mediator.

Describe dynamic data integration: techniques based on mediators and wrappers, techniques based on meta-models. Provide a small example.

For techniques based on mediators and wrappers see the previous answer.

A metamodel is an abstract model for specifying concrete models. It solves the problem of mapping between different models. They are useful when dealing with semistructured data sources; in this way we can use just one model. In particular, there are two types of metamodel:

- Metamodels in which general entities specializes into object of the specific target model. GSMM (General Semistructured Meta-Model) is a metamodel which specifies the concrete model in graph. Constraints define the semantic aspects of the concrete model.

- Entities that describes the objects of the target model. GDF (Geographical Data Files) is the standard model for the description of road networks.

Describe concisely the concept of mashup and its utility in data integration, highlighting its distinctive features w.r.t. using other integration techniques.

A Mashup is an application for light-weight integration of two or more mashup components, in a new way which can provide additional information, functions or visualization. Components may be put into communication with each other. Two are the key elements of this application:

- A **mashup component** is any piece of data, logic or user interface which can be reused locally or even remotely.

- The **mashup logic** is the logic which specifies the invocation of components, the control flow, the data flow, the data transformations, and the UI of the mashup.

We can have different types of mashups, depending on which layer the integration takes place:

- **Data Mashups**: fetch and integrate data from different sources
- **Logic Mashups**: combine functionalities of different components
- **UI Mashups**: combine (and possibily synchronize) UI native components into an integrated UI.
- **Hybrid Mashups**: span multiple layers of the application stack, bringing together different types of components inside one and a same application.

With respect to other integration practices mashups introduce integration at the presentation layer and typically focus on non-mission-critical applications. (Traditional integration focuses on application or data).

Mashup development is non trivial but has several advatanges. Developers can neglect the underlying complexity of each component, working on the "surface". Costs for product evaluation are reduced, final user are involved actively in the creation of the applications.

3 Data Analysis and Exploration

Consider these data mining problems: classification and clustering. Define them, discuss the differences between the two problems and provide an application example for each of them.

Data-Mining is an interdisciplinary field, which draws ideas from machine learning/AI, statistics, and database systems, and which is focused on the extraction of potentially useful information from data. We can distinguish mainly between two classes of methods:

- **Predictive:** Use some variables to predict unknown or future values of other variables.
- **Descriptive:** Find human-interpretable patterns that describe the data.

In **Classification** (supervised and predictive method) we have a collection of records (called training set) and each record has its own set of attributes, one of which is the *class*. The goal is to find a model for class attribute as a function of the values of other attributes, in order to assign a class to previously unseen records as accurately as possible. The accuracy of the model is then evaluated over a set of unseen records, called test set. Given a dataset, the rule of thumb is to split it into 80% training and 20% test. (Examples: fraud detection, customer fidelity).

Clustering is an unsupervised and descriptive method which tries to divide data points into clusters such that data points in one cluster are more similar to one another and data points in separate clusters are less similar to one another. Each data point has a set of attributes and we need to define a similarity measures between each data point. If the attributes are continuous we can use the Euclidean distance, trying to maximize inter-cluster distance and minimize intra-cluster distance. (Examples: market segmentation into distinct subsets of customers, or document clustering).

Describe the Box-Plot method for displaying the distribution of data, using an example to illustrate it clearly.

Box-Plot method is a technique for data visualization, which is often use in the data exploration context, a preliminary study over the data in order to understand characteristics and highlight possible patterns. Box-Plot is used mainly for displaying the distribution of data, comparison of different distribution or even for comparing attributes. Let's start from the definition of **percentile**:

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.

The Box-Plot represents percentiles (10, 25, 50, 75, 90) as shown in figure. We are not interested in data values, just in their distribution. (Example: marks of the class of an exam, or comparing attributes of the Iris dataset).

Define the concept of association rule in the data mining context, and give a small example.

Association Rule Discovery is a descriptive methods used in Data Mining context. It starts from a given set of records, each of which contains some number of items from a given collection, and it tries to produce dependency rules which will predict occurrence of an item based on occurrences of other items. (Examples: marketing and sales promotions).

Consider the following data mining problem: Sequential Pattern Mining. Define and illustrate it clearly with an application example.

Let's start from the formal definition of a sequence. A **sequence** is an ordered list of elements (transactions) $s =< e_1, e_2, e_3, ..., e_n >$. Each element contains a collection of events (items) $e_i = \{i_1, i_2, ..., i_k\}$ and each element is attributed to a specific time or location. A sequence that contains k elements is called k-sequence. A sequence could be, for instance, the purchase history of a given customer, with purchase sessions as elements and bought items as events, or the chronological web history of a user.

A **subsequence** is a sequence contained in another sequence. For instance, the sequence $a =< \{1\}, \{3\} >$ is included in the sequence $b =< \{1, 2\}, \{3, 4\} >$. The support of a subsequence w is defined as the fraction of data sequences that contain w. A **sequential pattern** is a frequent subsequence over a predefined sequence.

Sequential Pattern Mining is a technique which, given a database of sequences and a user-specified minimum support threshold, called *minsup*, tries to find all the subsequence whose support $\geq$ *minsup*. (Example: in a retail store, place the products on shelves based on the order of mined purchasing patterns).

4 Data Warehouses

Define what is a Data Warehouse and describe the dimensional fact model used in the data warehouse context and define its main elements. Provide a small example.

A Data Warehouse is 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 for organizational decision making. This collection of data is is integrated, subject-oriented, non-volatile and time-varying. Usually are very large databases.

In order to build a Data Warehouse, we start from designing its conceptual model. This phase is supported by the dimensional fact model, an intuitive, graphic formalism developed ad-hoc for this phase, which can used also by analysts and non-technical users. A model is a collection of fact schemata. A **fact schema** is composed by the following elements:

- **Fact**: a relevant concept for the decision making problem. It describes an N:M relationship among its dimensions.
- **Dimensions**: fact properties defined on a finite domain. They are the coordinates of the fact.
- **Measures**: numerical properties of the fact.
- **Dimension hierarchy:** a dimension can be structured in a hierachy, for instance: Day -> Month -> Year.

(Example: draw a fact sale, with attributes product, date and shop and measures quantity and income).

Describe flow, level, and unitary measures in the data warehouse context. Provide a set of examples.

In the context of the conceptual design phase of a Data Warehouse, we usually rely on the support of the dimensional fact model. Each fact can have zero or more associated numerical properties called **measures**. It is possible to identify three different categories of measures, depending on their aggregation level:

- **Flow measures:** they are related to a time period; at the end of the period the measures are evaluated in a cumulative way. (SUM, AVG, MIN, MAX over both temporal and non temporal hierarchies.) (Examples: number of sales in a day, total income in a month, number of birthdays in a year).
- **Level measures:** they are evaluated in particular time instants. (AVG, MIN, MAX, but SUM only over non temporal hierarchies.)(Examples: number of product in stocks, or number of citiziens in a city).
- **Unitary measures:** they are evaluated in particular time instants but they are relative measures. (Only AVG, MIN, MAX). (Examples: unitary price for an item in a particular instant. It cannot be aggregated with respect to time, nor category, nor shop).

Define the typical operations necessary in the multidimensional data model that is at the basis of data warehouses.

The multidimensional data model defines, for each combination of the dimensions of a fact, a value for its measures. The following OLAP operations are made available to the final users, which can query the DW with different purposes:

- **Roll-up**: aggregates data at a higher level - e.g. last year's sales volume per product category and per region.
- **Drill-down**: de-aggregates data at the lower level - e.g. for a given product category and a given region, show daily sale.
- **Slice-and-Dice**: applies selections and projections, which reduce data dimensionality.
- **Pivoting**: selects two dimensions to re-aggregate data (cube re-orientation).
- **Ranking**: sorts data according to predefined criteria.
- **Traditional operations**: select, project, join, etc.

There is only one major difference between the functionality of the **ROLLUP** operator and the **CUBE** operator. ROLLUP operator generates aggregated results for the selected columns in a hierarchical way. On the other hand, CUBE generates an aggregated result that contains all the possible combinations for the selected columns.

Define what is a Data Mart in a Data Warehouse and clearly summarize the methodological steps that lead from a collection of datasets to the specification of the logical schemas of one or more Data Marts.

A Data Mart is a structure/access pattern specific to data warehouse environments, used to retrieve client-facing data. The data mart is a subset of the data warehouse and is usually oriented to a specific business area.

In this context, we can distinguish between three different logical models:

- **MOLAP (Multidimensional):** data is natively stored in a multidimensional cube, in proprietary formats. Excellent performance, fast data retrieval, and are optimal for slicing and dicing operations, since all complex calculation have been pre-generated. Unfortunately, for this reason, it is not possible to include a large amount of data in the cube itself.
- **ROLAP (Relational):** relies on data stored in a relational database to givee the appearance of traditional OLAP's slicing and dicing functionality. Can handle large amounts of data, but performance can be slow. Each ROLAP report is essentially a SQL query.

- **HOLAP (Hybrid):** attempts to combine the advantages of MOLAP and ROLAP. For summary-type information, HOLAP leverages cube technology for faster performance. When detail information is needed, HOLAP can "drill through" from the cube into the underlying relational data.

The **logical modelling** process includes a sequence of steps that, starting from the conceptual schema, workload, data volume and system constraints, allows to obtain the logical schema for a specific data mart.

1. Choice of the logical schema (star/snowflake schema)
2. Conceptual schema translation
3. Choice of the materialized views
4. Optimization

In particular, it is useful to materialize a view when: it directly solves a frequent query or it reduce the costs of some queries.

Describe and compare the Star Schema and the Snowflake Schema used in the data warehouse context.

Star Schema and Snowflake schema are both used in the logical design phase of a Data Warehouse. The **Star Schema** consists of one or more fact tables referencing any number of dimension tables. Each dimension table is characterized by a primary key d_i and by a set of attributes describing the analysis dimensions with different aggregation levels. It is a simple schema, which leads to fast and simple queries. Tables are de-normalized, which implies redundancy but fewer joins to perform.

The **Snowflake Schema** reduces the de-normalization of the dimensional tables of a star schema, removing some transitive dependencies. All the attributes of a dimension table directly depends on the key, and there are zero or more external keys that allow to obtain the entire information. This schema reduces the memory space, removing data redundancy. It simplifies data update but queries are made more complex. It is possible to define views to mitigate this problem.

5 Miscellaneous

Define what is a NoSQL database. What does the CAP theorem state?

The classical DBMSs (also distributed) are transactional systems: they provide a mechanism for the definition and execution of transactions, guaranteeing the four ACID properties. It has been realized that it is not always necessary that a system for data management guarantees all transactional characteristics. A non transactional DBMS is commonly called NoSQL DBMS. This definition leads to ambiguity: with NoSQL we can also refer to databases that are not relational.

Main characteristics of NoSQL databases are that they are object-oriented friendly, they provide flexible schemas, they perform updates asynchronously and they make caching easier. The most common data models for a NoSQL databse are:

- **Key-Value**
- **Column-family**
- **Document-based**
- **Graph-based**

In theoretical computer science, the **CAP theorem** states that a data management system shared over the network can guarantee at most two of the following properties:

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

Interesting applications are found in these scenarios: data collected from sensors (append-only) or, in general, datasets which are seldom updated. Some famous implementations are: Amazon DynamoDB, Google BigTable, Cassandra, MongoDB, CouchDB.

Explain the Data Cleaning process in the context of Data Quality.

With the term **Data Quality** we usually refer to the ability of a data collection to meet user requests. Achieving a good quality is not easy, since there are several factors which can have a negative impact on data quality:

- **Historical changes**: the importance of data might change over time.
- **Data usage**: data relevance depends on the process in which data are used.
- **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.

The methodology for obtaining a good data quality is the following:

1. Data Quality requirements definition
2. Data assessment
3. Data analysis
4. Data improvement

All the steps are disposed in a logical ring.

Data Cleaning is a data-oriented improvement method used in the Data improvement phase. It is the process of identifying and eliminating inconsistencies, discrepancies and errors in data in order to improve its quality.

Define the concepts of valid time and transaction time in temporal databases also describing their advantages and disadvantages.

In temporal DBMSs, time is timestamped. This is very useful and has many benefits, like efficiency and performance increasing. There are two main types of semantic:

- **Valid Time** of a fact: it times when the fact is true with respect to the modelled reality. Thus, valid time captures the time-varying states of real world.
  - It can be in the past or in the future and can be change frequently.
  - Although all facts have a valid time, the valid timeof a fact may not necessarily be recorded in the DB (unknown or irrelevant for the application).
  - If a databases models different worlds, database facts might have several valid times, one for each world.

  A valid time table can be updated and it support historical queries.
- **Transaction Time** of a fact: it times when it was recorded in the database. Thus transaction time captures the time-varying states of the database. Applications that demand traceability of DB changes require transaction time. A transaction time table is append only: it keeps the history of the updates made on the database. It supports rollback queries.
  - They could be independently recorded or not and are associated with specific properties.
  - TT, unlike VT, is well-behaved and may be supplied automatically by the DBMS.
  - Time domain may be discrete or continuous.

Describe the problem of personalization and list the various kinds of personalization with a small explanation of each.

Personalization enables a personalized vision of data, taking into account needs, preferences and characteristics of a user (or a group of them). There are three different levels of personalization: presentation, interaction, data.

Considering the data-level personalization, here are listed some possible operations:

- **Re-order items:** e.g. chronological vs popularity order.
- **Focus on the items of interests:** people with different interests may exhibit different interests in data too.
- **Recommend additional options or suggestions:** suggestions of other items based on the previous history.

Briefly define pervasive data management and the main problems that must be solved.

The rapid development of the web, which enabled people to access an ever-growing amount of information and online services and the dominance of hand-held electronic devices, which made information access possible from anywhere and any time, led to a pervasive system.

"Pervasive computing is roughly the opposite of virtual reality. Where virtual reality puts people inside a computergenerated world, pervasive computing forces the computer to live out there in the world with people."

In a pervasive system things "disppear", i.e. we are no more aware of their presence. They become part of the infrastructure. The middleware of a pervasive system hides the heterogeneity of hundreds of devices making them transparent to the application. Devices are both reactive and proactive, so the computer modifies the world where it is built in. There are several problems related to pervasive data management:

- Hiding the devices heterogeneity from the application user.
- Using a high-level declarative language to send queries and commands to the devices.
- The network must be inexpensive to develop, deploy, program, and easy to use and maintain.

Application domains are: innovative applications, social and economic aspects and models, virtual and immersive systems.