# Managing Data Quality in Business Intelligence Applications

Florian Daniel, Fabio Casati, Themis Palpanas, Oleksiy Chayka

University of Trento
Via Sommarive 14
38100 Povo (TN) - Italy

{daniel, casati, themis, chayka}@disi.unitn.it

## ABSTRACT

Business Intelligence (BI) solutions commonly aim at assisting decision-making processes by providing a comprehensive view over a company's core business data and suitable abstractions thereof. Decision-making based on BI solutions therefore builds on the assumption that providing users with targeted, problem-specific fact data enables them to make informed and, hence, better decisions in their everyday businesses. In order to really provide users with all the necessary details to make informed decisions, we however believe that – in addition to conventional reports – it is essential to also provide users with information about the quality, i.e. with quality metadata, regarding the data from which reports are generated. Identifying a lack of support for quality metadata management in conventional BI solutions, in this paper we propose the idea of quality-aware reports and a possible architecture for quality-aware BI, able to involve the users themselves into the quality metadata management process, by explicitly soliciting and exploiting user feedback.

## 1. INTRODUCTION

Over the last years we have been witnessing an increasing use of *Business Intelligence* (BI) solutions, i.e., solutions such as data warehouses, reporting and data mining tools that allow business people to query, understand, and analyze their business data in order to make better decisions. As it is well known, the quality of the BI solutions is at most as good as the quality of the data in input. Bad or low-quality data may lead to bad business decisions. Imagine, for example, that the Department of Health wants to predict the quantity of flu drugs that is expected to be used in winter 2008/2009, to prepare for outbreaks or simply to negotiate discount rates with drug manufacturers. If the prediction is based on low quality data, e.g., data that are old, incomplete or incorrect, an insufficient quantity of drugs may be predicted and negotiated. Also, while purchasing additional quantities of drugs at higher prices might be acceptable, there still remains the danger that additional drugs cannot be delivered timely, as the manufacturer might not be able to quickly respond to late orders. Analogous problems may occur when logistics departments take goods-routing and warehousing decisions based on wrong sales or shipment data.

Data quality problems in data warehousing and BI applications are more and more common (and more and more impacting the everyday business) due to the fact that warehouses are becoming tentacular, reaching to a larger and larger number of source systems, also due to the recent trend towards enterprise-wide data warehouses. Different source systems typically provide data at different levels of quality, and the ETL process also becomes complex with the risk of errors in the cleaning procedures.

The above scenario underlines two different kinds of problems, whose combined effect leads to wrong business decisions. First, the low quality of the data. Second, the lack of awareness by the analysts that the data is of low quality and therefore that the reports they see and based on which they take their decisions are, in fact, inaccurate.

The latter problem and an attempt towards its resolution or mitigation is the focus of this paper. In particular, we propose the notion of *quality-aware reports* in BI applications, where reports explicitly expose the quality of the data underlying the generated results and, most importantly, their effect on the quality of the report. If the Department of Health were aware of the low quality of the data in input, it could for instance have done some further investigation to refine the quality of data in input and the prediction, thus saving money and assuring on-time delivery.

From an IT perspective, the above problem implies the ability to i) associate quality metadata with a report, ii) compute this metadata based on quality information on the base data, and iii) display such information to users in an easily comprehensible and "actionable" way, so that the viewers can identify the quality problems, understand their extent, and decide how relevant/severe they are and what to do about those. An interesting challenge here is represented by the fact that there are many different potential quality problems (from late arrival of the data, to potentially incorrect information at the source, to inconsistent use of terms by the persons doing data entry, to entity duplication issues, and many more), and it is important that users are aware of *why* a report is considered of low quality and which parts of the report have problems. For example, a report on the total number of surgeries may be inaccurate because it lacks data for the month of May from St. John's Hospital. The report analyst may decide that this is a serious issue as St. John's is large and has a highly variable number of procedures, or may decide that it is irrelevant as the number of procedures is very low or it is fairly stable month over month (or, very pragmatically, the analyst may make a call to St.


*VLDB '08*, August 24-30, 2008, Auckland, New Zealand.

John's and ask an estimate for the data value). Hence, it is not sufficient to merely show quality information; we need to make the user aware of the provenance of the quality information and of the reasons for certain assumptions on the quality of a report.

The latter observation also underlines that quality is *subjective* in two ways: first, the quality issues may or may not be significant or impacting a certain decision. Second, analysts may have (and, in our experience, very often do) personal knowledge or opinions on the quality of the data. For example, a health analyst may know or anyway believe that data from St. John's is often inaccurate, or that doctors use terms inconsistently when they enter data, or that the data collection process is entirely manual and therefore subject to frequent errors. From an IT perspective and with respect to the goal of building a quality-aware report solution, this observation has two implications.

First, we need to allow users to define personalized *quality-aware views* on reports or in general on the data (in contrast to *quality views* introduced in [22] and [23], where quality views express a users' quality processing requirements in terms of workflows). These quality profiles would embody any knowledge the user may want to express over the data. Such knowledge may not always be structural (as in the example of the lack of confidence in St. John's data), but also *situational*. For example, the user may see a detailed report on surgical procedures and detect that two different entries correspond in fact to the same procedure and therefore should be merged. For the situational case, this means that the definition of the quality profile can involve report-specific information, and can be interactive, that is the user "plays" with the report quality to correct the information when needed or to have the quality metadata take into account the user's personal beliefs on the data quality. Allowing the user now to interactively include/exclude data or to merge/unmerge records and to recompute reports on the fly would allow him/her to understand the importance of including/excluding such data into/from the report and to act accordingly.

Second, the opportunity arises for capturing user feedback and personalized quality-aware views and for using this information for re-defining the way quality metadata is computed. For example, if several users note that St. John's data is not to be trusted, then this information may be considered to be accurate by the BI applications (perhaps after review by an authorized user) and used to refine the quality metadata for reports that use St John's data.

As a final observation, we point out that the problem of data quality awareness in BI applications is not restricted to reports, but also applies to data mining models that mine data to discover information. The challenge here is how mining models can take into account data quality when computing their results, and how mining algorithms should be modified in this respect.

In this paper we present an architecture for quality-aware BI solutions and discuss some of the fundamental issues behind it, and specifically i) which are the main ingredients of such a solution and the related challenges; ii) which are the quality dimensions relevant in BI and how to model quality metadata for reports; iii) how quality in the base (warehouse) data affects quality in reports, and hence how to map base quality metadata into report quality metadata; iv) how to model quality-aware views (also called quality profiles); v) how to structure user interaction with the reports.

As the problem space is huge, and as the research is still in the early stages – consistently with the spirit of a workshop, this paper presents issues and solutions we experienced, rather than complete solutions – we focus on some of these issues, specifically quality for BI applications and the relation between data quality and report quality.

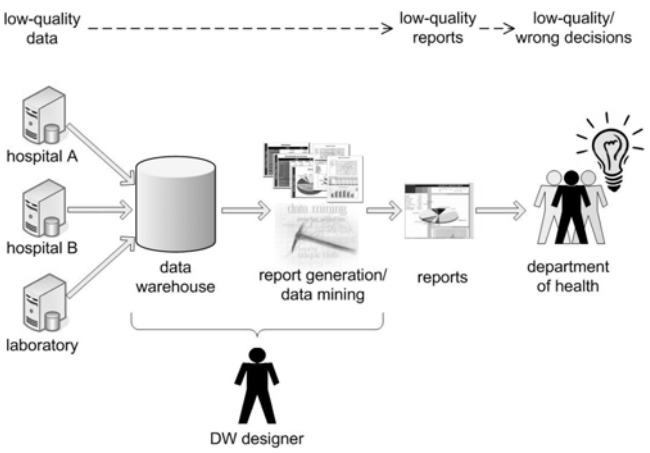


**Figure 1: The risk of low-quality data in healthcare BI.**

# 2. TOWARDS QUALITY-AWARE BI

## 2.1 Reference Scenario

Throughout this paper we will be using the healthcare example as our reference scenario, in order to exemplify and better explain our ideas. Specifically, we are interested in assisting the Italian Department of Health in forecasting the quantity of flu pharmaceuticals (e.g., aspirin) for the upcoming winter 2008/2009, in order to enable the Department of Health to agree with manufacturers on nation-wide stable and fare prices for its citizens.

The typical levers in the hands of the Department of Health to control pharmaceutical prices are dedicated tax regulations (e.g., lowering the value added tax for individual pharmaceuticals) or participation in the production cost (e.g., the state may take over part of the manufacturing cost of a pharmaceutical, in order to keep its customer price low). Obviously, each intervention by the Department of Health is associated with a cost for the State: either there is a missing income in terms of taxes that are not levied, or there is an expense in integrating manufacturing costs. Either way, the expected cost for the state needs to be predicted after summer, when the Government prepares the budget for the following year.

For the prediction of the pharmaceutical demands, the Department of Health adopts a BI solution that sources data from each of the country's 20 Regions (Italy is politically and geographically structured into Regions, at a higher level, and Provinces at a lower level), which aggregate the necessary healthcare data from their local hospitals, laboratories, emergency rooms, and the like. Regional data are collected in a centralized data warehouse, which enables the Department of Health to view National health reports, to analyze and mine collected data, and to predict pharmaceutical demands. Figure 1depicts the described scenario.

Figure 1 also highlights the core problem the Department of Health has to deal with: low-quality data in input unavoidably lead to low-quality reports in output. Low-quality reports might lead to wrong estimates and unwanted budget problems. Unfortunately, low-quality data is a reality, and it is hard (if not infeasible) to eliminate all possible quality problems via data cleaning during the ETL process. For instance, Figure 2 exemplifies some

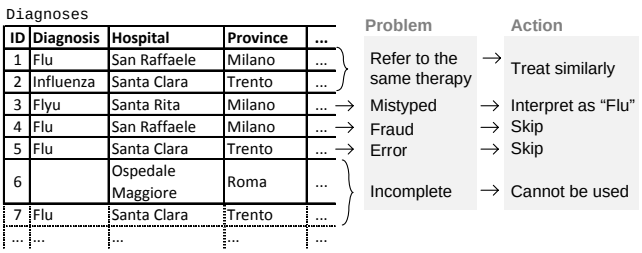


**Figure 2: Typical data quality problems. Dashed lines represent expected but missing data.**

typical quality problems we might encounter when looking for example at the *Diagnoses* table containing information about the diagnoses made in different hospitals.

The first two rows refer to the same diagnosis, with the only difference that in the first row the diagnosis is "Flu", in the second row the diagnosis is "Influenza"; it is important to understand that both rows actually refer to the same diagnosis. The diagnosis in row number 3 is mistyped, which makes it algorithmically hard to understand that row 3 refers to flu, as well. The previous cases represent inconsistencies in the data, which might lead to too low a prediction of necessary flu pharmaceutical, if not identified as such. But even if we are able to infer that the three rows might refer to the same diagnosis, we typically will not be entirely sure of this finding, and it might be good to keep track of our level of confidence when further processing the aligned data. In row number 4 we have assumed that it simply does not correspond to the truth, in the sense that a doctor could simply have declared a fake diagnosis in order to get money for a not provided treatment; the row should actually not be considered in the computation of reports. This kind of fraud is very hard to identify in practice and, hence, might lead to an incorrect overestimation of the drug demand. Row number 5 represents another possible problem: a simple error. Errors happen, but we should be able to identify them, in order to skip the respective row. An erroneous tuple such as the one in the figure might for example be due to a test of the source system where test tuples have not been eliminated correctly. Identifying such kinds of errors is however practically impossible and they might lead to an overestimation in the prediction. Finally, row number 6 lacks the value for the diagnosis attribute, and row number 7 is simply missing (represented by the dotted borders); hence, those rows cannot be used, even though they might correspond to a flu diagnosis (but we don't know). Incomplete data might lead to an underestimation of the drug quantity.

Although the above examples highlight only few of the typical quality problems in databases, they are still enough to show how low-quality data in input to the BI solution might negatively affect the quality of the output of the BI solution, i.e., of the reports and the data mining results.

## 2.2 Research Challenges and Contributions

Generalizing the described healthcare scenario allows us to identify the research challenges that characterize the data quality problem of most business intelligence scenarios:

- How to *define quality* and identify measurable *quality properties* that appropriately characterize the specific case of data warehousing and data mining.
- How to model *quality metadata* associated with warehouse data and reports.
- How to *map* warehouse quality metadata into report quality metadata (given a report definition as a query over warehouse data).
- How to *improve* the quality of data in the warehouse, e.g., via data cleaning techniques.
- How to effectively *expose quality metadata* and assumptions (e.g. about ETL procedures or data cleaning decisions) to end users. Interactive, quality-aware reports could for instance allow dynamic what-if scenarios based on data quality properties.
- How to collect and manage *quality-related user feedback*. While the automated data cleaning process might help mitigate quality problems, in many cases the best evaluator of quality is still the user. Effectively collecting explicitly-provided user feedback might for instance help fine-tune the data cleaning process and improve the quality of outputs.
- How to compute *customized reports* based on individual user feedback. A user might provide customization instructions for his/her reports, expressing personal preferences or knowledge about the quality of data underlying the reports.
- How to *propagate collective user feedbacks* into the warehouse and ETL procedures. If a specific quality problem reaches a predefined threshold of aligned user feedbacks, the feedback might be transformed into proper quality metadata to be used globally in the warehouse.
- How to *build quality-aware mining models* from quality-labeled data. It might for instance be interesting to reconsider known mining models, however considering quality in the training and validation datasets.

As a first step toward quality-aware BI and in particular focusing on the problem of understanding how to inform and involve the end user in report quality management, in this paper we provide the following contributions:

- We define *quality in the context of data warehousing* by identifying relevant quality properties.
- We define the *concept of quality-aware report* as a means to provide users with an awareness of the quality of the data underlying the reports they are inspecting.
- We propose a *quality-aware data warehouse architecture* that aims at i) managing quality metadata, ii) enabling the computation of quality-aware reports, and iii) taking into account user-provided feedback.
- We discuss the *modeling* of warehouse and report metadata and the mapping between the two in report computation.
- We discuss *related works* in light of the requirements identified for the development of the envisioned quality-aware data warehouse and position our work accordingly, highlighting still *open research challenges*.

# 3. DATA QUALITY IN BI

## 3.1 The Notion of Data Quality

To assess the quality of data, the research community has identified various dimensions. A common set of quality dimensions and their definitions (proposed by [41]) is listed in Table 1.

**Table 1: Commonly accepted data quality measures [41].**

| Dimension | Definition |
|---|---|
| Accessibility | the extent to which data is available, or easily and quickly retrievable |
| Appropriate amount of data | the extent to which the volume of data is appropriate for the task at hand |
| Believability | the extent to which data is regarded as true and credible |
| Completeness | the extent to which data is not missing and is of sufficient breadth and depth for the task at hand |
| Concise representation | the extent to which data is compactly represented |
| Consistent representation | the extent to which data is presented in the same format |
| Ease of manipulation | the extent to which is easy to manipulate and apply to different tasks |
| Free of error | the extent to which data is correct and reliable |
| Interpretability | the extent to which data is in appropriate languages, symbols and units, and the definitions are clear |
| Objectivity | the extent to which data is unbiased, unprejudiced and impartial |
| Relevancy | the extent to which data is applicable and relevant for the task at hand |
| Reputation | the extent to which data is highly regarded in terms of its source or content |
| Security | the extent to which access to data is restricted appropriately to maintain its security |
| Timeliness | the extent to which data is sufficiently up-to-date for the task at hand |
| Understandability | the extent to which data is easily comprehended |
| Value-added | the extent to which data is beneficial and provides advantages from its use |

The dimensions above, and analogous proposals arising from the data quality community, are oriented toward evaluating the quality of a generic dataset. In this paper we focus on the problem of representing data quality to end users in BI applications, with the specific goal and challenge of helping users understand the quality of BI results presented to them and to avoid making wrong assumptions on the data presented and therefore running the risk of making wrong decisions. We are particularly interested in exposing non-obvious quality problems to end users, rather than in quality issues that are presented by design in the BI applications.

For example, we are not interested in discussing timeliness (freshness) of the data in the warehouse, or in trying to understand if a warehouse with data loaded monthly is "good" or "bad". Similarly, we are not interested in assessing completeness of sources in the sense of ensuring that we have deployed our ETL application to extract data from all possible hospitals or social care structures. These are conscious *design* decisions, which are well known to the end user (or which anyway can be easily communicated). Such dimensions as appropriate amount of data, concise representation, ease of manipulation, relevancy, security, understandability and value-added are also irrelevant for our goals.

Instead, we are interested in spotting situations where data is loaded monthly but for a given batch load one source did not make the data available, or the data was not loaded due to ETL errors. Similarly, we are interested in data incompleteness problems caused by a source not logging (or the ETL not extracting) some of the surgical procedure data for certain patients or class of surgical procedures. The above situations may lead users to view aggregated data based on certain assumptions (all surgical procedures data is there) which may turn out to be incorrect in the specific report they are viewing.

In summary, we focus on quality dimensions relevant in multi-source BI applications for the purpose of communicating to the end users transient properties of the data sources and of the data extracted from them and loaded into the warehouse.

## 3.2 A BI-Specific Definition of Quality

For the purpose of developing quality-aware reports in BI applications, we propose the following quality dimensions as the relevant ones:

- Completeness
- Consistency
- Confidence

**Completeness** measures to which extent data that according to the warehouse specifications should have been recorded in a table are effectively present. We refer to *vertical incompleteness* when we measure completeness of data in a column, that is, the quantity or percentage of values in the column that are null (or where codes such as “9999” are inserted in place of missing information; for an example, see Figure 3) when the information is instead supposed to be there. Null values might in general be allowed and represent meaningful information, e.g., for persons that undergo their first-ever surgery, the fact that the date of previous surgery is null is acceptable and not a sign of incompleteness. *Horizontal incompleteness* refers instead to the quantity or percentage of entire tuples (e.g., tuples representing surgical procedures), typically entire facts or dimension entries that have not been recorded. Figure 3 exemplifies this dimension, showing for example that row number 8 which should have been logged is not present in the recorded dataset. Note that in [25] vertical and horizontal completeness are called *density* and *coverage*, respectively.

There are many reasons why incompleteness may occur, such as errors or omissions in the data entry at the source, or errors in the ETL process that fails to record some of the tuples in the warehouse. An important and relatively frequent problem that leads to incompleteness is *batch unavailability*, that is, delays in loading batch data in the warehouse. In addition to vertical and horizontal completeness (like [25] and [13]), we therefore also consider batch availability as completeness property. Sources are supposed to make data available at specified time intervals, which is when the data load into the warehouse occurs. A batch is unavailable if the source does not provide the data or if the ETL process fails for some reason (e.g., it is unable to connect to the source). Unlike other incompleteness scenarios, batch unavailability is relatively easy to detect and to communicate to the report viewers. In the table in Figure 3, for instance, the entire batch from the Province of Bolzano is missing.

Completeness can be modeled as *extensional* or *intensional* metadata, and at different levels of granularity. Extensionally, completeness for cells is expressed as a binary value (i.e., true/false). For columns and tuples, it is expressed in percentages (100% representing full completeness), for each column in case of vertical completeness and for the entire table for horizontal ones.

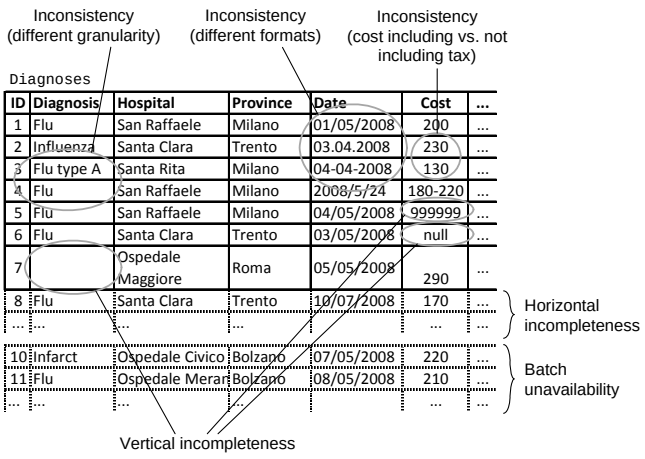


**Figure 3: Completeness and consistency problems in the DW. Dashed lines represent incomplete or unavailable data.**

Since data warehouses are typically loaded in batches, completeness measures can also refer to batches of load. Indeed, in practice it does happen that different batches may have different degrees of completeness, and, as mentioned, entire batches may be unavailable. Even more frequent is the case where completeness is related to specific data sources. This information is very important not only because we can know, when computing a report, if the report is complete or not (e.g., a report not querying St. John's data may be complete even if St. John's data is incomplete), but also because users can make own judgments on the quality assumption.

The latter two cases and the reasoning above show the need for an intensional measure of incompleteness, where a rule is stored rather than a mere measure (cf. business rules [12], [30]). In general rules are functions over the dataset that identify a set of tuples, and for these tuples define a completeness measure in terms of percentage. A textual description is also attached to the rule. Informally, examples of these rules are "all entries by Dr. Smith are 70% complete on average". Formally, functions can for example be expressed as SQL queries.

Finally, the case of batch (un)availability is measured in terms of which batch loads are (un)available. Each batch is also associated to a time window to which the extraction refers (e.g., batch 22 correspond to May 2008), which is something then useful to provide information at report viewing time. The measure can be associated to a data source (e.g., all data from that source for batch X are unavailable) or to a data source *and* a table (e.g., all surgical procedures data from that source for batch X are unavailable).

In the current paper we do not discuss two important issues: first, how to derive (compute) the intensional or extensional measures, and second, how to deal with conflicting intensional rules. These problems, especially the first, are very hard and can be subject of entire lines of research [25], [33]**.**

**Consistency** denotes the uniformity of the information in a given column. *Syntactic* consistency refers to uniformity in the data format (e.g. different date formats in the table in Figure 3). This is typically something that is detected and corrected at data cleaning time (e.g. via normalization), and is not discussed further.

*Semantic* consistency refers to the satisfaction of semantic rules defined over a set of data items. There are different reasons why the information can be inconsistent (see Figure 3):

1. Different understanding of the semantics of the field: For example a date field may refer to the patient surgery date, or to the date the diagnosis was made.
2. Different abstraction/precision level: the semantics of the field may be commonly understood, but the degree of precision or detail when entering the data can be different. For example, a doctor may generically enter "Flu" while another can enter "Flu type A" which is more precise.
3. Different units: in this case the understanding and granularity are the same but the interpretation differs on the unit of measure, such as Celsius vs Fahrenheit or, as depicted in Figure 3, cost including taxes vs. cost excluding taxes.

From an assessment perspective, consistency is often checked by defining a set of business rules [12], [30], and its measure may take various forms: first, there can be a qualitative measure attached to a data set to denote if the values there are overall consistent or not. However inconsistencies typically occur between data sources, or between different persons entering data. For this reasons, (in)consistency is also expressed in terms of:

- A measure of "inconsistency" applied to a data cell whose value is suspected to be inconsistent with the other values (a fine-grained quantitative measure is meaningless here, while a qualitative distinction with a few, possibly as few as two distinct values for (in)consistency suffice for our purposes).
- An intensional description that labels as inconsistent or possibly inconsistent the data from a given source or entered from a data entry agent.

In general the approach to intensional description is the same as for completeness: a description of an inconsistency is represented by a textual description, by a function over the data and the warehouse metadata (provenance and batch information, such as data source or time or data load) that identifies tuples affected by the quality issue, and by the inconsistency problem. As an example again pounding on Dr Smith, we can state that diagnosis data entered by Dr. Smith and related to flu is inconsistent.

**Confidence** describes the perceived accuracy of the data, or the degree of trust (or, from the opposite perspective, the degree of uncertainty) that the data present in a table or set of tables is accurate. In general, reasons for marking data as uncertain (low confidence) include lack of trust in a data source, potential errors or uncertainty detected during data cleaning [31], [16], outlier values [27], and others. As for the above measures, we can define confidence as a probability (certainty) measure associated to cells, tuples, tables, or data sources, as for example done in Trio [4].

However confidence has more sides that need to be addressed and that cannot be covered by the above representation. A key problem is that a large number of uncertainty issues in data warehouses are caused by *entity resolution* issues. This means that a confidence representation must include the possibility of expressing that two or more tuples may refer to the same real world entity. In addition, as proposed in other quality-aware systems, we may need to provide alternative versions of the truth (possible worlds) as opposed to simple uncertainty measures. To this end, confidence representation can take these two additional forms:

- Alternative values (or numeric ranges) for a cell.
- Links among tuples that denote that the set might correspond to a same entity (i.e., that they *could* be merged into one).

| Dimension | Measure | AssociatedData | Desc. | Author | Date | ... |
|---|---|---|---|---|---|---|
| Vertical incompleteness | 30% | references to a table and a column | ... | | 04/05/2008 | ... |
| Horizontal incompleteness, batch unavailability | 20% | data source, batch identifier, missing time window | ... | | 03/05/2008 | ... |
| Confidence | 90% | references to a table, a column and a cell | ... | | 05/05/2008 | ... |
| Consistency | 95% | select Diagnosis from Diagnoses where Doctor="John Smith" and Date < 1-1-2008 | ... | Peter | 10/07/2008 | ... |
| Confidence, entity resolution | 80% | references to a pair of tuples candidates for merging | ... | John | 11/05/2008 | ... |
| ... | ... | ... | ... | ... | ... | ... |

**Figure 4: Example quality metadata to be stored in the DW.**

## 3.3 Warehouse quality metadata

We now briefly describe the metadata that we have to store in the warehouse in order to capture the various aspects of data quality that we outlined above.

There are three aspects we need to capture when attaching quality metadata to the raw data in the warehouse: i) the quality problem (the metric and the measure), ii) the identification of the cells or tuples to which the metric and measure apply, and iii) descriptive information such as why a certain statement on data quality is made, when it was entered, by whom, and so on.

The first aspect is characterized by

1. The *quality dimension*, which specifies what aspect of the data quality we are focusing on. This is one of the quality dimensions we introduced in the previous section.
2. The *measure* for the selected quality dimension. This is a value (could be a percentage, a range, or a binary value) that reflects to which degree there is a data quality problem. It can also be expressed as a function, as discussed next.

The second aspect refers to associating these metadata with the raw data at different granularities, that is, at the levels of individual cells or tuples, as well as entire columns or tables. There are two ways for making this association, namely, extensional and intensional. In extensional, we have to make explicit associations of specific metadata to individual pieces of data in the warehouse. On the other hand, intensional allows us to map specific metadata to a set of data in the warehouse. This mapping may be expressed using a function, and for example an SQL query, therefore providing a large degree of flexibility and control [39]. Figure 4 illustrates the described ideas (albeit, at the conceptual level).

Last, we also attach some descriptive information that can help the analyst to further investigate the causes and consequences of certain quality problems. Examples of such information may be the reason for some data quality problem, explanatory notes, the author of the rule, the date the rule was added, and so on.

# 4. REPORT QUALITY

In the previous sections, we have described data quality with respect to the raw data in the warehouse. Now we show how quality issues in the base tables of the warehouse affect the quality of the reports presented to users, and how we can interact with the user to inform or get feedback about report quality. Although it is commonly recognized that in BI there is a strong correlation between the quality of data and the quality of business decisions [12], [30], we believe that explicitly assigning quality values to reports as a way to communicate the risk of low-quality decisions has not yet been investigated adequately. Doing so requires us to address the following problems:

1. First, we need to define what a *report* is, and which the different types of reports we need to consider are, in order to better understand the report quality problems.
2. Once we know the different quality problems that we need to model at the base data and at the report level, we need to understand how to *compute report quality* from base data quality, that is, how to populate report quality metadata from base quality metadata.

In the following we discuss these issues. In the next section we instead discuss user interaction with the reports and the personalization of quality metadata.

## 4.1 Reports and report types

To get and analyze data from a DW, users employ reporting tools that are able to render various reports. Usually reports are defined as a combination of i) methods on how to obtain the data; ii) formatting and layout information for the rendering of the data; iii) properties of reports (e.g., its title, a textual description, the legend). The following are examples of reports:

- *Standard report*: table-based representation of data queried from the DW;
- *Chart report*: graphical representation of data queried from the DW;
- *Pivot report*: allows the comparison of two different data sets against each other; it helps to discover data correlations;
- *Comparison report*: shows differences in a data set considered at two different instants of time.

In the following, we assume that reports are essentially queries over the DW, rendered in some form (tabular or graphical). We therefore assume that a report represents a set of views over the base data. Such views can be computed on the fly or at specified time points, usually (but not necessarily) right after the completion of a new batch load of the warehouse. We consider two types of views: non-aggregated views and aggregated views. *Non-aggregated views* are essentially tables or charts that show raw data from the warehouse, *aggregated views* are tables or charts that show aggregated data (for the purpose of this paper, these are essentially queries with a *group by* statement in them).The above division will help us analyze quality mappings later in this section. In the following, we will use the terms *view* and *report* interchangeably.

Since reports are tables, the quality metadata that describe reports are of the same nature as the one describing base tables. Hence, in general, the problem that we try to solve is to find out, given a set of base tables, the quality metadata on these tables, and a query that defines a view over them, how to map quality metadata on the base tables into quality metadata associated with the view. We do not solve this problem here, but we discuss issues and identify which quality problems at the base data level map into quality problems at the report level, taking in particular into account the issue of aggregations that are common in reports.

## 4.2 From Data Quality to Report Quality

In order to understand which quality properties are relevant to characterize reports, we investigate how base data quality affect s the quality of final reports. That is, we look at completeness, consistency, and confidence of base data and derive similar quality properties for reports. For a better understanding, we discuss separately the cases of non-aggregated and aggregated reports.

*Data*(*in*)*completeness* in non-aggregated reports is carried over from the base data to the final report. Therefore, missing cells (vertical incompleteness) or missing tuples (horizontal incompleteness) in the base data result into missing cells or tuples in the report, i.e., into an *incomplete* report. Figure 5 graphically depicts the described scenario: the base data at the left present all three forms of incompleteness (horizontal and vertical and batch unavailability; we will focus on the inconsistency problem next); therefore, the non-aggregated report at the right misses the diagnosis for the "Ospedale Maggiore" and the second tuple for the hospital "Santa Clara", just like the base data; also, no data about the Province of Bolzano can be shown, as the whole respective batch is not available in the data. Notice that not all incomplete tables map into incomplete reports, as the report might select a portion of the base table that is complete. In general this applies to all quality dimensions, and relates to the problem of identifying which base table metadata maps into report quality metadata.

For aggregated reports, data incompleteness may lead to missing cells in the report only if in the base data all the values of a group are missing (e.g., used to calculate a sum). In Figure 5, for instance, the aggregated report lacks the diagnoses for the Province of Roma, due to the missing diagnosis in the base data (actually, we don't know whether it should be in the report or not, as we do not know which exact diagnosis value is missing for that Province). The batch unavailability of the data from the Province of Bolzano, on the other hand, should be in the aggregated report, but the respective data is missing in the DW. If instead an aggregated value is computed over a column/attribute with only partially missing data, a value can be computed and, hence, the report is not incomplete. In this case, we can say that the incompleteness of the base data affects the *confidence* of the final report. This is exemplified in Figure 5 by the tuple regarding the Province of Trento (we should actually see 2 flu diagnoses), which is computed over incomplete data; the tuple regarding the Province of Milano is correct.

*Data consistency* problems in the generation of non-aggregated reports will carry over from the base data to the reports. That is, misunderstandings of the semantics of fields and different abstraction levels or units will unavoidably show up in the report as inconsistent data, just like they are in the base data. For instance, the non-aggregated report in Figure 5 presents the same inconsistencies as its base data, i.e., "Flu" vs. "Influenza". For aggregated reports, data consistency problems typically lead to low *report confidence*: if aggregated values (e.g., the number of flu diagnoses per Province in Figure 5) are computed over a column with inconsistency problems, the final result will be characterized by a low confidence, as we cannot be sure whether all relevant values have been considered in the computation or not. Indeed, there would be 3 flu diagnoses for the Province of Milano, but the hospital "Santa Rita" has entered "Influenza" instead of "Flu", the respective tuple could not be counted. The confidence of that data for the Province of Milano is hence low.

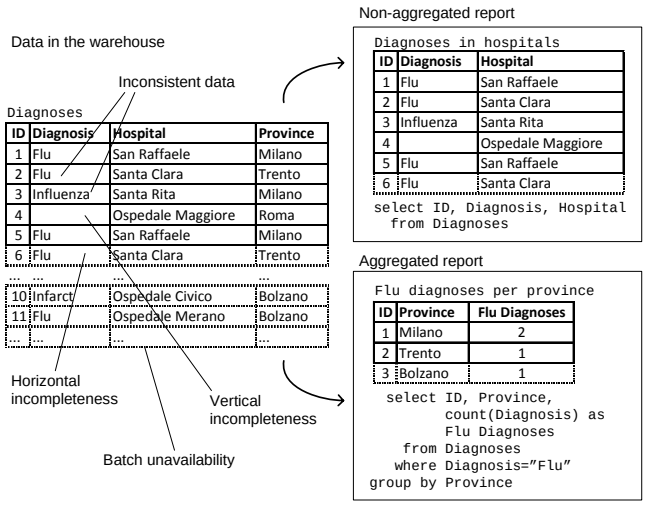


**Figure 5: Effects of data completeness, batch unavailability, and data consistency on report quality. The SQL queries show how the reports are computed. Dashed lines represent expected, but not complete or available data.**

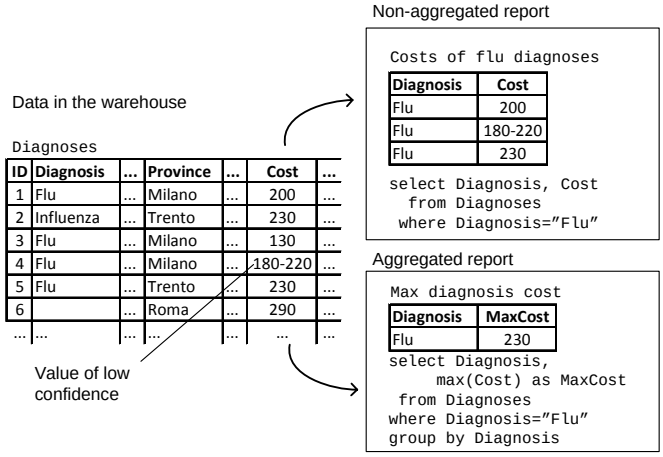


**Figure 6: Effects of data confidence on report quality. Note that the value "180-220" is intended as an indication for low confidence in absence of a precise specification of quality metadata.**

*Data confidence* properties carry over in the computation of non-aggregated reports and directly affect the *report confidence*. In non-aggregated reports, data values with low confidence just carry over, resulting in a report that includes values with low confidence. Figure 6 depicts for example how the cost value "180-220" carries over from the base data to the non-aggregated report, maintaining its low level of confidence. The same is true for aggregated reports as well, where aggregated values with low confidence may lead to an aggregated value of low confidence.

However, in some cases aggregated reports may eliminate the lack of confidence originating from the base data. Consider the following example, depicted in Figure 6. Assume we need to create a report and compute the maximum cost for flu diagnoses out of the table in Figure 6, which presents one value ("180-220") with low confidence. Even though there is an evident confidence problem, the report will contain the correct result (i.e., "290"), which will also be accurate.

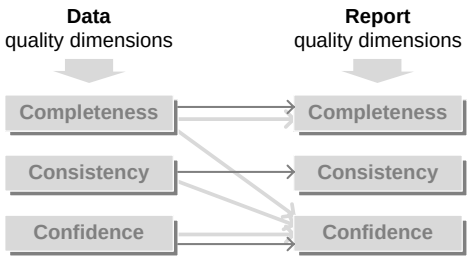


**Figure 7: Mapping of raw data quality properties into report quality properties. Dark-gray arrows refer to non-aggregated reports, light-gray arrows to aggregated reports.**

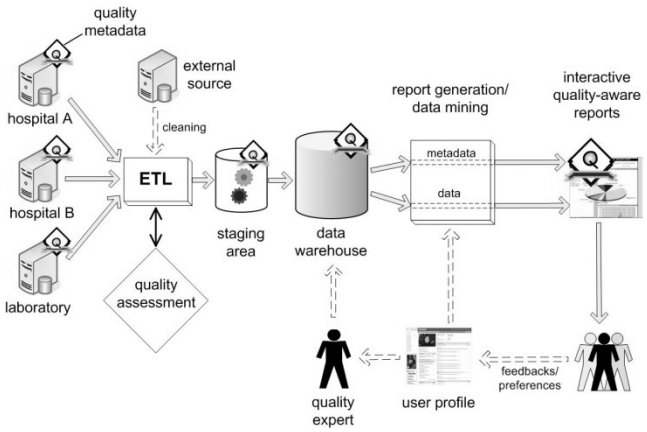


**Figure 8: Quality-aware reporting and user quality profiles for the fine-tuning of quality metadata (the "Q" rhombuses).**

Figure 7 graphically summarizes the above discussion on how report quality is determined by raw data quality. The dark-gray arrows in the figure represent the mapping for data quality properties into report quality properties for non-aggregated reports; light-gray arrows represent the mapping for aggregated reports.

# 5. USER INTERACTION WITH QUALITY-AWARE REPORTS

Once we have a quality metadata framework for reports as discussed above and a way to compute report quality metadata, we can use this information to visualize quality-aware reports and support user interaction with them, as well as leverage quality information for the analytics algorithm developed on top of the warehouse or on top of the reports. Specifically, we envision the following opportunities (and consequent research challenges):

- *Visualization*: How to visualize quality information in a way that is easy to "consume" and understand if and which parts of the report are meaningful and can be used to take business decisions. This aspect also has a degree of "subjectiveness" and therefore includes taking into considerations user preferences and quality profiles, both in terms of how to show information and, from a more semantic viewpoint, to include users' personal beliefs on data quality.
- *Interaction*: How to have the user interact with the report in 3 ways: i) to "simulate" alternative view reports based on varying assumptions on the data quality and based on what to consider for the report computation (e.g., confidence thresholds); ii) to define personal quality profiles in the meaning described above; iii) to provide feedback on the quality assumptions to be used by the system to correct data cleaning procedures or quality metadata computation procedures, so that the user's knowledge can be used not only for the subjective report views but also as "objective" information for the benefit of all report consumers.
- *Analytics*: This aspect is related to data and reports as consumed by applications (and typically by BI applications). The techniques here are application specific, so in this paper we state some general issues related to quality-aware business intelligence.

In the following, we detail research issues and preliminary ideas related to these topics. We begin by discussing the quality profiles which is the user-specific metadata common to all the above issues. Then we briefly discuss research issues in the areas of visualization, interactions, and analytics over quality-aware reports.

## 5.1 Quality profiles

Quality profiles are user-specific report configuration data that define (i) the *appearance* of quality metadata (how the report is graphically tagged) and (ii) how to *filter* and adjust imperfect data that contributes to the computation of a report.

We refer to the latter information as *quality tuning* metadata. We distinguish between *general* tuning data and *report-specific* tuning data. The first include user beliefs that are applicable across all reports (e.g., "I always trust data from St. John's"). The latter includes tuning of a specific report model (e.g., on the monthly report on the average cost of surgical procedures by unit) or even a report instance (the above report computed for June 2008).

Figure 8 extends the scenario architecture introduced in Figure 1 with user/quality profile metadata, and highlights how user feedback and preferences may drive the report generation and data mining processes. Collected tuning metadata may also be assessed by a quality expert, possibly propagating feedback into the actual quality metadata in the DW.

The tuning metadata can include this information:

- At the simplest level, tuning can simply mean having a personalized version of the quality metadata. This implies that the end user can view and edit, e.g., the completeness or confidence values, or even the intensional descriptions of the quality metadata. (Note that we are not concerned here with the UI and in general the user support for editing such metadata easily, but just in the end results, that is, the quality profile). This "rewriting" can be applied to some or to all original metadata entries, and can also include new entries not originally captured (e.g., the warehouse may believe that data from St. John is complete but the end user may know that this is not the case). It also applies to metadata specific to a report, as well as to general metadata, so that for example the user can also state that while the warehouse metadata states that St. John's data is in general uncertain, from their perspective it is certain.

- Users can define threshold levels and metadata aggregation functions (as e.g. in [8]) for data to be included in the report computation. For example, it is possible to express that, although the quality values are not changed, only data with a quality level above a threshold are included.
- The above approach can be generalized by having end users define a metadata policy that determines which metadata and which data are to be considered in the generation of the (personalized) report. In general, users can define two kinds of queries over the quality metadata to express this behavior: the first is *metadata filtering functions*, that is, a set of queries or procedures that outputs which metadata entries should be considered. For example, one can decide to only consider quality metadata entries by herself only or by a quality metadata manager she trusts. This is analogous to an "intensional tuning", where the tuning is specified by using functions. The second is *data filtering functions*, that is, intensional definition of functions that define thresholds for data to be included in the report (e.g., never include data that is less than 30% complete).
- The final category includes *replacement functions*, that is, definition of algorithms to replace data with quality problems with estimates. For example, users may decide to replace incomplete data with estimates from the previous month.

## 5.2 Visualization and interaction

Visualizing personalized quality information and letting users interact with such information to modify the report based on their perceived quality is a key goal of this line of research. In general visualization, especially when end users are involved, is a complex issue as a poor visualization paradigm may defeat the purpose of the entire work.

Any visualization approach needs to be accompanied by an HCI study to assess its understandability. The development and assessment of the visualization paradigm is in progress; here we limit ourselves to some key aspects and discussion points.

Just like (graphical) report design is a human-intensive activity and cannot be easily demanded to automated generation if we want a usable result, the same holds for quality metadata. While we can develop default representations for the various quality issues identified earlier, it should be possible to tailor the final results to the users and to the nature of the problem. Hence, we expect the quality-aware report design infrastructure to allow for report design of quality aspects (based on a set of primitive quality visualization concepts) as well, rather than imposing a default visualization. In our research, we focus on which these quality visualization primitives are and on how they can be combined in a quality-aware report.

On the interaction side, the main issues to be addressed and functionality to be provided are the following:

*Interactive quality exploration*: Users should be able to "play" with the quality information and preview what the report results would be if quality metadata were different. For example, Figure 9 represents different projections of the same chart that will be changed depending on the selected confidence level (in the figure denoted by the slider position). Depending on such a parameter, relevant data will be considered to create a chart that will be updated on the fly. Trying to select various levels of confidence, the user can see for instance the best/worst case of the spreading of a particular disease, with certain extent of confidence in such a fact.

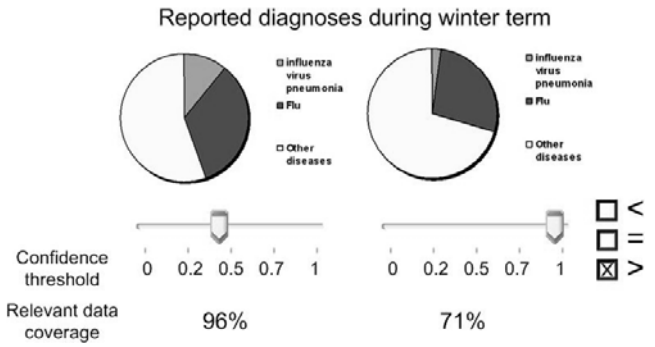


**Figure 9: Interactive quality-aware report.**

The percentage of relevant data coverage is shown under the chart for which the data was taken. The value shows the user how much data is taken into consideration when constructing the chart and how much data with lower confidence is left out of the analysis.

Completeness problems may also be addressed by removing tuples with incomplete data or by inserting missing values, e.g., by extrapolating/predicting data from past values.

In general, we envision the following ways for interacting with report quality metadata: users should be able to "turn off" some quality problems (e.g., assume the information is correct and complete and disregard the quality metadata entry), to change the quality measures, to edit the logic of intensional rules (e.g., maintain the rule that data from St. John is low confidence, but exclude the case of Dr. Hyde who is known to be reliable), or to compensate for the quality problems with custom logic (e.g., predicting missing values). All this requires visualization primitives and, for the case of intensional rule editing, requires approaches similar to Query By Example (QBE) [29]. Again, the interaction for exploration purposes may be based on defaults or it may be designed ad hoc. For example, for editing an intensional rule, we can either provide a generic QBE paradigm, or we can provide a simplified interface, designed ad hoc for a given report or rule, where for example users can explicitly define / modify intentional rules based on doctors or hospital of provenance.

Finally, to support all of the above, we also need to let the user know the reasons behind the quality metadata measures, that is, why the values are the way they are, based on what assumptions, and who defined these assumptions.

*Quality feedback*: The result of the exploration can be stored either as part of the personalized quality profile, or it can be proposed as a generally applicable rule that modifies the report metadata and possibly also the base quality data. This topic presents challenges ranging from how to map report metadata changes back into base metadata changes, to how to assess the validity of the proposed changes. Analogously to approaches on case based reasoning, here we need to assess the quality of the feedback and decide 1) how to embody it into the metadata so that it can be shown to different end users to make them aware that other users have made different assumptions, and 2) how to understand when to combine various feedbacks and propose them to analysts for incorporation into the ETL and quality measurement processes.

## 5.3 Analytics

The components of the solution we presented above are focused on the interaction of the quality-aware data warehouse with the human users of the system. Another aspect of the proposed approach that is equally important is the introduction and use of data analysis and data mining techniques that are quality-aware.

Most of the current data mining techniques assume that the data they operate on are complete and accurate. Therefore, they produce exact, deterministic results based on these data values. This is not true for privacy preserving data mining, where the objective is to perturb the original values while producing a correct final result (see [40] for a brief overview of the work in this area). In our case we face a different problem, namely, how to analyze and mine data that are inherently not exact and complete. The system we are proposing will be coupled with quality-aware data mining algorithms, which will treat all the quality metadata as first class citizens. These metadata will be used by the data mining models, along with the data values, in order to produce the final results. This way, the mining results will incorporate knowledge of the inaccuracy/incompleteness of the original data and will expose to the users a range of possible answers with confidence values characterizing their accuracy. Recent studies have looked at similar problems in the areas of OLAP [8] and deviation detection [1].

# 6. RELATED WORK

Several studies have focused on the problem of data quality. They have examined different aspects of this problem, ranging from the definition of the term "quality" in this context, to the management and processing of all the quality-related information, and in such diverse domains as health care [19] and manufacturing [28].

## 6.1 Data quality

The problems of characterizing and dealing with data quality have been the focus of several studies. Data quality problems emerge in literally every application domain, and techniques for addressing such problems have been proposed in the literature.

Data quality can be measured and quantified according to various parameters. Previous works provide different classifications of the data quality dimensions [35], [20], [6], [17], [26]. These classifications provide a basic set of data quality dimensions, among which accuracy, completeness, consistency, interpretability, timeliness, and understandability [32]. However, defining objective measures for all the above dimensions is a challenging task, and an active area of research.

Data quality has also been identified as an important problem in other domains as well, such as manufacturing, and business processes. In this context, several approaches have been proposed for managing data quality problems, with the most prominent one being the Six Sigma approach [28].

## 6.2 Data provenance

When data are stored in some database, we are interested in keeping track of information related to the provenance of these data [36]. An important issue in data providence is its characterization. That is, to find the answers of questions like "*why is a piece of data in the output?*" and "*where is the piece of data copied from?*" Buneman et al. [7] target these issues and propose a framework for describing and understanding provenance. The more recent work of Green et al. [15] describes provenance in the context of incomplete and probabilistic databases. In [24] the authors specifically focus on provenance quality data in scientific workflows. Provenance metadata may also play a major role in assigning and managing quality measures.

Sometimes the propagation of annotations is dependent on the syntax of the query. One may want to control the propagation of annotations in a schema. The custom propagation schemes allow the user to specify where to obtain annotations from. Bhagwat et al. [5] present propagation schemes that are essentially based on where data is copied from.

## 6.3 Identity resolution

Another problem relevant to quality is that of identity resolution or duplicate detection (i.e., whether two different pieces of data refer to the same real world object).

Duplicate detection through record linkage has been extensively studied [11]. Many of these approaches are based on different flavors of clustering algorithms. A clustering technique is also the basis of the approach proposed by Andritsos et al. [1]. Using rule-based approaches [21], [14], it is easier to create a large number of training pairs that are either clearly non duplicates or clearly duplicates. Despite that, the rule-based approaches require user intervention in rule management scenarios. Recent approaches have also focused on the problem of how to efficiently support the duplicate identification operation in the context of relational database systems [16].

Several data cleaning techniques have also been proposed for the problem of structural heterogeneity (for example, representing a date as *year/month/day* in place *day/month/year*, or the location of a room as *room number-building-university* in place of *university-room number-building*) [31].

## 6.4 Uncertainty in databases

Problems related to data uncertainty have been studied in the past in the area of databases and data warehouses. Several studies have proposed a framework for quality-oriented data warehouse design [18], [38], [37], [12]. These frameworks take into account the entire lifecycle of the data warehouse, and are able to track the quality of data at each stage of the process. The above approaches aim at setting a quality goal, evaluating the current quality status, and finally at analyzing and improving the current situation. The same principles have also been applied to the domain of health care data [19], where a process model for the data warehouse lifecycle of health care data is described, that is able to capture errors in the design, integration, and use of the warehouse. Nevertheless, none of the above studies focuses on the specific problems relevant to report generation, use, and management, when quality measures are taken into account.

Recently, there has been lots of interest in databases specifically designed to manage uncertain data [3], [4], [34], [10]. In this case, data are coupled with a probability value indicating the degree of confidence to the accuracy of the data. These probabilities are then taken into account by the database management system when processing the data to produce answers to user queries. The difference to our approach is that the above systems do not deal with the problems of assigning these probabilities and of deriving them in complex cases, such as when computing reports. In this case, we need to reason about quality measures that are assigned to objects of different granularities (e.g., cells, tuples, or tables), and we also need to use semantics as to how to combine the different quality measures.

## 7. CONCLUSION AND OUTLOOK

In this paper we have investigated an end-user-centric business intelligence view on the problem of low data quality, proposing what we call *quality-aware business intelligence*. We have discussed how low quality data in input affects the quality of the output of a business intelligence application, i.e., the reports. Accordingly, we have proposed the use of *quality-aware reports*, allowing the end-users to interactively "play" with report quality metadata, finally enabling them i) to be aware of the quality of the report they are looking at, ii) to fine-tune a report based on personal knowledge about the quality of the underlying data, and iii) to provide and share with other users quality-related feedback.

In this study, we have highlighted novel challenges and open issues in handling low quality data in business intelligence applications, which we believe will play a major role in business intelligence over the next years. We are currently pursuing the research directions outlined in the previous sections at both the warehouse and the report levels (which represent the focus of our work), but we know that we have only touched on the full problem and that there are plenty of related issues that still demand an answer.

**Acknowledgements**

We would like to thank Cinzia Cappiello for her valuable comments and suggestions. This work was supported by funds from the European Commission (contract N° 216917 for the FP7-ICT-2007-1 project MASTER).

## 8. REFERENCES

[1] C. C. Aggarwal, P. S. Yu: Outlier Detection with Uncertain Data. SDM 2008: 483-493.

[2] P. Andritsos, A. Fuxman, and R. Miller, "Clean Answers over Dirty Databases: A Probabilistic Approach," in ICDE, 2006.

[3] L. Antova, C. Koch, and D. Olteanu, "10^10^6 Worlds and Beyond: Efficient Representation and Processing of Incomplete Information," in ICDE, 2007, pp. 606-615.

[4] O. Benjelloun, A. D. Sarma, A. Y. Halevy, M. Theobald, and J. Widom, "Databases with uncertainty and lineage," VLDBJ, vol. 17, no. 2, p. 243–264, 2008.

[5] D. Bhagwat, L. Chiticariu, C. W. Tan, and G. Vijayvargiya, "An annotation management system for relational databases," VLDBJ, vol. 14, no. 4, pp. 373-396, 2005.

[6] M. Bovee, R. P. Srivastava, and B. Mak, "A Conceptual Framework and Belief Function Approach to Assessing Overall Information Quality," in IQ, 2001, pp. 311-328.

[7] P. Buneman and S. Khanna, "On Propagation of Deletions and Annotations through Views," in PODS, 2002.

[8] D. Burdick, P. M. Deshpande, T. S. Jayram, R. Ramakrishnan, S. Vaithyanathan: OLAP over uncertain and imprecise data. VLDB J. 16(1): 123-144 (2007).

[9] C. Cappiello, C. Francalanci, and B. Pernici, "Data Quality Assessment from the Users Perspective," in IQIS, 2004, pp. 68-73.

[10] N. Dalvi and D. Suciu, "Efficient query evaluation on probabilistic databases," VLDBJ, vol. 16, pp. 523-544, 2007.

[11] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate record detection: A survey," IEEE Transactions on Knowledge and Data Engineering,, vol. 19, no. 1, pp. 1-16, 2007.

[12] L. P. English, Improving Data Warehouse and Business Information Quality. John Wiley & Sons, 1999.

[13] A. Even and G. Shankaranarayanan, "Understanding Impartial Versus Utility-Driven Quality Assessment in Large Data-sets," ICIQ'07.

[14] H. Galhardas, D. Florescu, D. Shasha, E. Simon, and C.-A. Saita, "Declarative Data Cleaning: Language, Model, and Algorithms," in VLDB, 2001, pp. 371-380.

[15] T. J. Green, G. Karvounarakis, and V. Tannen, "Provenance Semirings," in PODS, 2007.

[16] S. Guha, N. Koudas, A. Marathe, and D. Srivastava, "Merging the Results of Approximate Match Operations," in VLDB, 2004, pp. 636-647.

[17] M. Jarke, M. A. Jeusfeld, C. Quix, and P. Vassiliadis, "Architecture and Quality in Data Warehouses: An Extended Repository Approach," Information Systems, vol. 24, no. 3, pp. 229-253, 1999.

[18] M. Jarke and Y. Vassiliou, "Data warehouse quality: a review of the DWQ project," in IQ, 1997, pp. 299-313.

[19] R. L. Leitheiser, "Data Quality in Health Care Data Warehouse Environments," in HICSS, 2001.

[20] A. Levitin and T. Redman, "Quality Dimensions of a Conceptual View," Information Processing and Management, vol. 31, no. 1, pp. 81-88, 1995.

[21] E. Lim, J. Srivastava, S. Prabhakar, and J. Richardson, "Entity Identification in Database Integration," in IEEE International Conference on Data Engineering, 1993, p. 294–301.

[22] P. Missier, S. Embury, M. Greenwood, A. Preece, and B. Jin, "Quality views: capturing and exploiting the user perspective on data quality," in VLDB, 2006.

[23] P. Missier, S. Embury, M. Greenwood, A. Preece, and B. Jin, "Managing Information Quality in e-science: the Qurator workbench," in SIGMOD, 2007.

[24] P. Missier, S. Embury, R. Stapenhurst. "Exploiting provenance to make sense of automated data acceptance decisions in scientific workflows", IPAW'08, Salt Lake City, USA.

[25] F. Naumann, J. C. Freytag, and U. Leser, "Completeness of integrated information sources," Information Systems, vol. 29, pp. 583-615, 2004.

[26] F. Naumann, Quality-Driven Query Answering for Integrated Information Systems. Springer, 2002.

[27] T. Palpanas, N. Koudas, and A. Mendelzon, "Using Datacube Aggregates for Approximate Querying and Deviation Detection," IEEE transactions on knowledge and data engineering, vol. 17, no. 11, pp. 1465-1477, 2005.

[28] T. Pyzdek, The Six Sigma Handbook, Second ed. McGraw-Hill, 2003.

[29] R. Ramakrishnan and J. Gehrke, Database Management Systems, 3rd ed. McGraw-Hill, 2002.

[30] T. C. Redman, Data Quality for the Information Age. Artech House, 1996.

[31] S. Sarawagi, "Special Issue on Data Cleaning," Bulletin of the IEEE Technical Committee on Data Engineering, vol. 23, no. 4, 2000.

[32] M. Scannapieco and T. Catarci, "Data Quality under the Computer Science Perspective," Archivi & Computer, vol. 2, 2002.

[33] M. Scannapieco and C. Batini, "Completeness in the Relational Model: a Comprehensive Framework," in IQ, 2004, pp. 333-345.

[34] S. Singh, et al., "Database Support for Probabilistic Attributes and Tuples," in ICDE, 2008, pp. 1053-1061.

[35] D. M. Strong, Y. W. Lee, and R. Y. Wang, "Data Quality in Context," Commun. ACM, vol. 40, no. 5, pp. 103-110, 1997.

[36] W. C. Tan, "Provenance in Databases: Past, Current, and Future," IEEE Data Eng. Bull., vol. 30, no. 4, pp. 3-12, 2007.

[37] D. Theodoratos and M. Bouzeghoub, "Data Currency Quality Factors in Data Warehouse Design," in DMDW, 1999.

[38] P. Vassiliadis, M. Bouzeghoub, and C. Quix, "Towards Quality-oriented Data Warehouse Usage and Evolution," Information Systems, vol. 25, no. 2, pp. 89-115, 2000.

[39] D. Srivastava, Y. Velegrakis: Intensional associations between data and metadata. SIGMOD Conference 2007: 401-412.

[40] V. S. Verykios, et al., "State-of-the-Art in Privacy Preserving Data Mining," ACM SIGMOD Record, vol. 3, no. 1, pp. 50-57, 2004.

[41] R. Y. Wang and D. M. Strong, "Beyond accuracy: what data quality means to data consumers," Journal of Management Information Systems, vol. 12, no. 4, pp. 5-33, 1996.