ENABLING BETTER DECISIONS THROUGH QUALITY-AWARE REPORTS IN BUSINESS INTELLIGENCE APPLICATIONS

(Research-in-Progress)
Business Intelligence and IQ, Metadata and IQ

Florian Daniel, Fabio Casati, Themis Palpanas, Oleksiy Chayka
University of Trento, Italy
{daniel, casati, themis, chayka}@disi.unitn.it

Cinzia Cappiello
Politecnico di Milano, Italy
cappiell@elet.polimi.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 soliciting and exploiting user feedback.

Key Words: Business Intelligence, Data Quality, Quality Metadata

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 [26]. 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 incomplete or imprecise, 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 arise 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. One of the main issues regards the heterogeneity of the source systems that are also characterized by different levels of quality. In order to deal with low data uniformity, the ETL process becomes more 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 do 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.

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 have) personal knowledge or opinions on the quality of the data.

Therefore, in order to build a quality-aware report solution, we need to allow users to define personalized *quality views* on reports or in general on the data. These profiles would embody any knowledge the user may want to express over the data. Such knowledge may not always be structural but also *situational*. For example, the user may see a detailed report on a specific theme 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 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 re-compute 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.

Furthermore, the user interaction should be also important for capturing user feedback and personalized quality views and for using this information for re-defining the way quality metadata is computed. For example, if several users note that data derived from a specific source cannot 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 these data.

In this paper we present an architecture for managing quality-aware reports in 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 views (also called profiles); v) how to structure user interaction with the reports.

TOWARDS QUALITY-AWARE BI: 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, the Italian Department of Health yearly forecasts the quantity of flu pharmaceuticals (e.g., the traditional aspirin) that is likely to be required during the winter months in order to agree with manufacturers on nation-wide stable and fare prices for its citizens. The forecasts are based on a BI application integrating nation-wide health data.

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 the manufacturing costs. Either way, the expected cost for the state needs to be predicted typically after summer, when the Government prepares the budget for the following year.

The BI solution used for the prediction of the pharmaceutical demands 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.

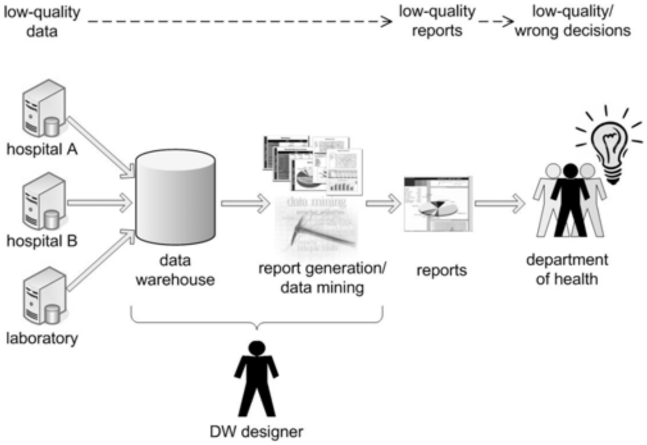
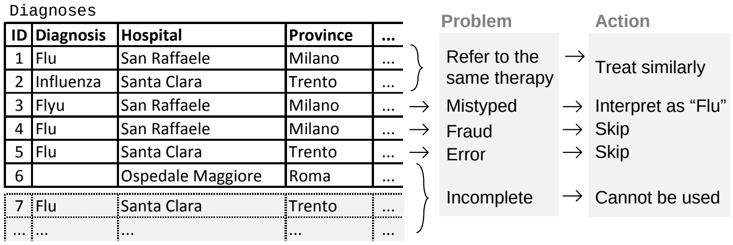


Figure 1: The risk of low-quality data in healthcare BI.

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



Diagnoses

ID	Diagnosis	Hospital	Province	...
1	Flu	San Raffaele	Milano	...
2	Influenza	Santa Clara	Trento	...
3	Flyu	San Raffaele	Milano	...
4	Flu	San Raffaele	Milano	...
5	Flu	Santa Clara	Trento	...
6		Ospedale Maggiore	Roma	...
7	Flu	Santa Clara	Trento	...
...	...	...	...	...

Problem	Action
Refer to the same therapy	Treat similarly
Mistyped	Interpret as "Flu"
Fraud	Skip
Error	Skip
Incomplete	Cannot be used

Figure 2: Typical data quality problems. Dashed and gray shaded rows represent expected but missing data. The *Fraud* and *Error* problems represent simple assumptions by the authors without any further evidence.

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 (note that this cannot be reconstructed from the data), 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; 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.

Generalizing the described healthcare scenario allows us to identify the research challenges that characterize the data quality problem of most business intelligence applications. It is possible to distinguish between *objective* issues, which do not involve the end user, and *subjective* issues, which instead take into account the user’s perspective. Objective issues in business intelligence are:

- *Data quality dimensions definition*: identification of measurable quality properties that appropriately characterize the specific case of data warehousing and data mining.
- *Quality metadata management*: modeling quality metadata associated with warehouse data and reports.
- *Data quality improvement*: enhancement of the data quality in the warehouse, e.g. via data cleaning.
- *Report quality assessment*: derivation of report quality metadata from data warehouse quality metadata.
- *Report quality visualization*: exposition of the quality metadata and related 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.
- Q*uality-aware mining models:* it might for instance be interesting to re-consider known mining models, however considering quality in the training and validation datasets.

Subjective issues include:

- *Quality-related user feedback collection:* while the explicit visualization of quality metadata inside reports enhances the users’ awareness of the objective quality of the reports they are inspecting, collecting and managing quality-related feedback from the user in turn allows the BI application to also leverage subjective quality information.
- *Customized reports construction*: user feedbacks can for instance be used as customization instructions for an individual user’s reports, taking into account personal preferences or knowledge about the quality of the data underlying the reports.
- *User-driven data cleaning*: *w*hile the automated data cleaning process might help mitigate quality problems, in many cases the best evaluator of quality is still the user. Explicitly provided user feedback might for instance help fine-tune the data cleaning process and improve the quality of outputs. 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 or ETL procedures.

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 contributions about data quality dimensions definition and assessment in data warehousing environment but we mainly focus on 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. In fact, 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. In this paper, we do not yet focus on quality-aware mining models, which is however part of our future work.

RELATED WORK

Several studies have focused on the problem of data quality from different perspectives in different domains (e.g. health care [18] and manufacturing [23]) and in different types of information systems (e.g, operational [31] and analytical information systems [9]). The relationship between information quality and business intelligence has also been analyzed in the literature.

Several contributions prove that information quality and business intelligence quality are positively correlated: the higher the information quality the higher the effectiveness of the decision making process. A relevant work on the impact of data errors on the correctness of business decisions was conducted by Ballou and Pazer [2]. Raghunathan [24] investigated this issue with special focus on the accuracy dimension. The analysis shows that the quality of decisions improves with an increasing quality of the data only if the decision maker is aware of the relationship among the problem variables, while it may degrade in the opposite case. The importance of providing information regarding the quality of the data used in decision making has been subsequently tested in [9]. The authors show how information quality awareness positively impacts on the decision making. Furthermore, they demonstrate that also the way quality information is visualized and rendered to the user influences the decision making process.

Several studies have focused on quality-oriented data warehouse design [12][15][17][32][33]. Most of the proposed 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 approaches aim at setting a quality goal, evaluating the current quality status, and finally at analyzing and improving the current situation. However, in the data warehousing context, data quality has typically been associated with the ETL module, since it is in the transformation phase where data can be normalized, cleaned and integrated.

As regards data integration, the identity resolution or duplicate detection (i.e., whether two different pieces of data refer to the same real world object) through record linkage has been largely analyzed [11]. Data standardization has been considered by several data cleaning techniques that aim at solving the problem of structural heterogeneity (for example, representing a date as year-month-day in place of day-month-year, or the location of a room as room number-building-university in place of university-room number-building) [27]. Some of these concepts have been applied to the domain of health-care data. Leitheiser [18], for instance, describes a process model for the data warehouse lifecycle for health care data 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 quality-aware report generation, use, and management.

Recently, there has been lots of interest in databases specifically designed to manage uncertain data [1][5][10][29]. 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 proposed systems however do not deal with the problems of assigning probabilities and of deriving them in complex cases, such as when computing reports. In our 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.

CHARACTERIZING DATA QUALITY IN BI

To assess the quality of data, the research community has identified various dimensions. Data quality can be measured and quantified according to various parameters. Previous work provides different classifications of the data quality dimensions [6][16][19][20][30][34]. By analyzing these classifications, it is possible to define a basic set of data quality dimensions including accuracy, completeness, consistency, timeliness, interpretability, and accessibility, which represent the dimensions considered by the majority of the authors [8]. Those dimensions, and analogous proposals arising from the data quality community, are oriented towards 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 and to avoid making wrong assumptions on the data presented, which might lead to 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). Further, some other dimensions as appropriate amount of data, concise representation, ease of manipulation, relevancy, security, understandability and value-added are also irrelevant for our goals.

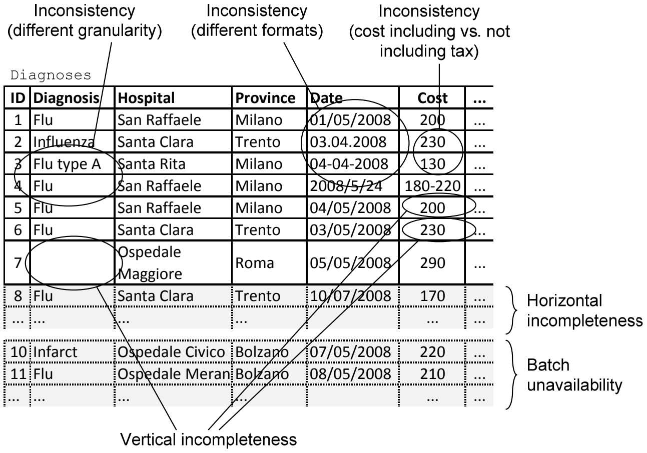
Instead, we are interested in spotting situations where data are loaded monthly but for a given batch load one source did not make the data available, or the data were not loaded due to ETL errors. Similarly, we are interested in data incompleteness problems caused by a source not logging (or the ETL 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.

BI-specific data quality dimensions

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. Specifically, we propose the following quality dimensions as the relevant ones: completeness, consistency, and 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.

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 (as also analyzed in [13] and [21]), 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.


Inconsistency
(different granularity)
Inconsistency
(different formats)
Inconsistency
(cost including vs. not including tax)
Horizontal incompleteness
Batch unavailability
Vertical incompleteness

Diagnoses

ID	Diagnosis	Hospital	Province	Date	Cost	...
1	Flu	San Raffaele	Milano	01/05/2008	200	...
2	Influenza	Santa Clara	Trento	03.04.2008	230	...
3	Flu type A	Santa Rita	Milano	04-04-2008	130	...
4	Flu	San Raffaele	Milano	2008/5/24	180-220	...
5	Flu	San Raffaele	Milano	04/05/2008	200	...
6	Flu	Santa Clara	Trento	03/05/2008	230	...
7		Ospedale Maggiore	Roma	05/05/2008	290	...
8	Flu	Santa Clara	Trento	10/07/2008	170	...
...	...	...	...		...	...
10	Infarct	Ospedale Civico	Bolzano	07/05/2008	220	...
11	Flu	Ospedale Meran	Bolzano	08/05/2008	210	...
...	...	...	...		...	...

Figure 3: Completeness and consistency problems in the data warehouse. Dashed and gray shaded rows represent expected but not complete or available data.

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.

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 then make their own judgment 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. 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 be for example expressed as SQL queries, as discussed later in the paper.

Finally, the distinguished 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).

Consistency denotes the uniformity of the information in a given table. *Syntactic* consistency refers to uniformity in the data format for a specific field (for example, different date formats are used 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 [3]. There are different reasons why the information can be inconsistent (see Figure 3):

- 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.

- 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.
- 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, meters vs. feet, or, as depicted in Figure 3, cost including taxes vs. cost excluding taxes.

From the assessment perspective, consistency is often checked by defining a set of business rules [12][26] and its measure can take various forms: first, there can be a qualitative measure attached to a data set (e.g. a table column) 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, when the value there is suspected to be inconsistent with 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 and precision 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 [14][27], outlier values [22], 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 [5].

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. In fact, in the raw sources, two or more tuples may refer to different representations of the same real world entity. In the integration phase, considering the uncertainty, the system could 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, or links among tuples that denote the possibility that the set corresponds in fact to the same entity (i.e., that they could be merged into one).

Warehouse quality metadata

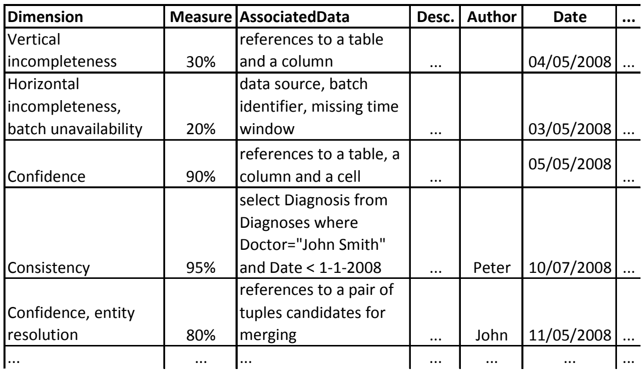
There are three aspects we need to capture when attaching quality metadata to the raw data in the warehouse: i) the quality problem (quality dimensions with related metric and measure), ii) the identification of the cells or tuples to which the metric and measure apply, and iii) descriptive information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

In detail, the first aspect is characterized by:

- the *quality dimension*, which specifies what aspect of the data quality we are focusing on, and the related measure;
- the *measure* for the selected quality dimension, that 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 with 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. Figure 4 illustrates the described ideas (albeit, at the conceptual level). Note that in the current paper we do not discuss how to derive (compute) the intensional or extensional measures and how to deal with conflicting intensional rules. These problems, especially the first, are very hard and can be subject of entire lines of research (e.g., [21][28]).

Last, we also include some descriptive information that can help the analyst to further investigate the causes and consequences of 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.



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 data warehouse.

REPORT QUALITY

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][26], 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. It is necessary to define a report quality assessment procedure in order to compute *report quality* from base data quality, that is, a procedure to populate report quality metadata from base quality metadata.

It is worth clarifying that reports are usually defined as a combination of i) methods on how to obtain the data; ii) formatting and layout information for the rendering of the data (e.g. table-based, graphic); iii) properties of reports (e.g., its title, a textual description, the legend). In the following, we assume that reports are essentially queries over the data warehouse (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, report quality metadata is of the same nature as base data quality metadata. Hence, the problem that we try to solve is i) given a set of base tables, how to derive quality metadata for these tables; and ii) given a query that defines a view over them, how to map quality metadata from the base tables into quality metadata associated with the view, i.e., how base data quality affects 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 reports and aggregated reports.

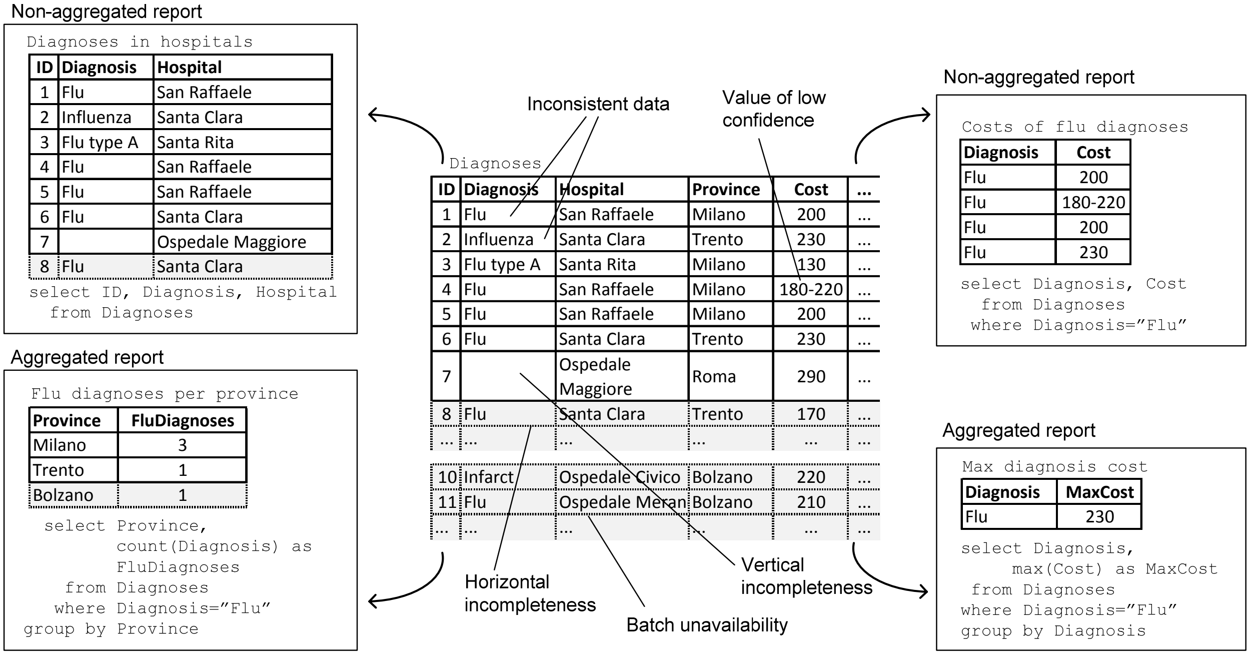
Completeness. Data (in)completeness in non-aggregated reports is carried over from the base data to the final report. Therefore, missing cells (vertical incompleteness) or missing tuples (horizontal incompleteness) in the base data result into missing cells or tuples in the report, i.e., into an *incomplete* report. Figure 5 graphically depicts the described scenario: the base data in the center present all three forms of incompleteness (horizontal and vertical and batch unavailability; we will focus on the inconsistency and confidence problems next); therefore, the non-aggregated report in the upper left corner 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 in the lower left corner 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 (which will be low, if the underlying data is incomplete). This is exemplified in Figure 5 (lower left report) 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 (if we do not consider the highlighted inconsistency problem).

Consistency. 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 at the left in Figure 5 presents the same inconsistencies as its base data, i.e., "Flu" vs. "Influenza" vs. "Flu type A". 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 the lower left report 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 4 flu diagnoses for the Province of Milano, but since 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.

Confidence. 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 5 depicts for example how the cost value "180-220" carries over from the base data to the non-aggregated report (in the lower right corner), maintaining its low level of confidence. The same is true for aggregated reports, 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 5 (lower right report). Assume we need to create a report and compute the maximum cost for flu diagnoses out of the table in Figure 5, 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., "230"), which will also be accurate.



Diagnoses in hospitals

ID	Diagnosis	Hospital
1	Flu	San Raffaele
2	Influenza	Santa Clara
3	Flu type A	Santa Rita
4	Flu	San Raffaele
5	Flu	San Raffaele
6	Flu	Santa Clara
7		Ospedale Maggiore
8	Flu	Santa Clara

```
select ID, Diagnosis, Hospital
  from Diagnoses
```

Flu diagnoses per province

Province	FluDiagnoses
Milano	3
Trento	1
Bolzano	1

```
select Province,
       count(Diagnosis) as
       FluDiagnoses
  from Diagnoses
 where Diagnosis="Flu"
group by Province
```

Diagnoses

ID	Diagnosis	Hospital	Province	Cost	...
1	Flu	San Raffaele	Milano	200	...
2	Influenza	Santa Clara	Trento	230	...
3	Flu type A	Santa Rita	Milano	130	...
4	Flu	San Raffaele	Milano	180-220	...
5	Flu	San Raffaele	Milano	200	...
6	Flu	Santa Clara	Trento	230	...
7		Ospedale Maggiore	Roma	290	...
8	Flu	Santa Clara	Trento	170	...
...	...	...	...	...	...
10	Infarct	Ospedale Civico	Bolzano	220	...
11	Flu	Ospedale Meran	Bolzano	210	...
...	...	...	...	...	...

Costs of flu diagnoses

Diagnosis	Cost
Flu	200
Flu	180-220
Flu	200
Flu	230

```
select Diagnosis, Cost
  from Diagnoses
 where Diagnosis="Flu"
```

Max diagnosis cost

Diagnosis	MaxCost
Flu	230

```
select Diagnosis,
     max(Cost) as MaxCost
 from Diagnoses
where Diagnosis="Flu"
group by Diagnosis
```

Figure 5 Effects of data quality on report quality. The SQL queries show how the reports are computed. Dashed and gray shaded rows represent expected, but not complete or available data.

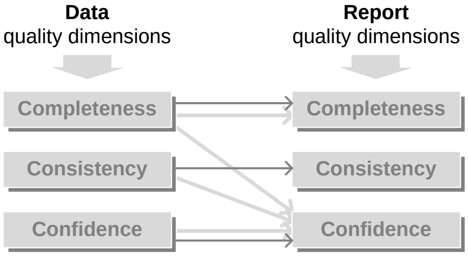


Figure 6: Mapping of base data quality properties into report quality properties. Dark-gray arrows refer to non-aggregated reports, light-gray arrows to aggregated reports.

Figure 6 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.

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):

- *Quality Profile Identification:* How to define users' quality preferences. The users can specify the quality metadata in which they are interested and the acceptable thresholds in their specific context.

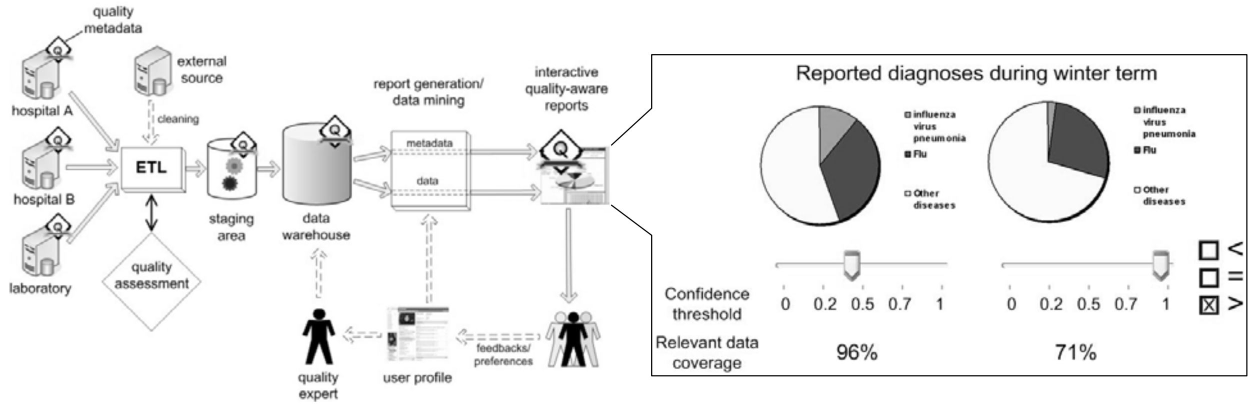
- *Visualization*: How to visualize quality information in a way that it is easy to "consume" and to understand if and which parts of the report are meaningful and can be used to take business decisions. This aspect also has a degree of "subjectiveness" and therefore takes into consideration user preferences and profiles, in terms of both how to show information and how 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 report views 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 profiles as 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.

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 7(a) 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* (e.g. as in [7]) 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.



(a) Quality-aware warehouse architecture with user quality profiles. (b) Interactive quality-aware report

Figure 7: Quality-aware reporting.

Visualization

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.

Interaction

On the interaction side, issues to be addressed and the 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 7(b) 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. By selecting various levels of confidence, the user can e.g. see best/worst cases for the spreading of a disease, along with the respective confidence levels.

The percentage of relevant data coverage is shown under the chart for which the data was taken. This 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. As another example, completeness problems can be addressed either by removing tuples with incomplete data or by inserting the missing values, e.g., by extrapolating/predicting these data from past values.

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) [25]. 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 convey the reasons behind the quality metadata measures, i.e., how values are measured, under which assumptions, defined by whom, etc.

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 i) 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 ii) how to understand when to combine feedbacks and propose them to analysts for incorporation into the ETL and quality measurement processes.

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 the presented work). But the problem of data quality awareness in BI applications is not restricted to reports only; it 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 can be modified in this respect. Quality-aware mining models are a further direction or our future research.

Acknowledgements: This work was supported by funds from the European Commission (contract N° 216917 for the FP7-ICT-2007-1 project MASTER).

REFERENCES

[1] Antova, L., Koch, C., Olteanu, D. "$10^{10\wedge 6}$ Worlds and Beyond: Efficient Representation and Processing of Incomplete Information", *Proceedings of ICDE'07*, pp. 606-615.

[2] Ballou, D.P. and Pazer, H.L. "Framework for the Analysis of Error in Conjunctive, Multi-Criteria, Satisficing Decision Processes", *Decision Sciences* 21(4), pp. 752-770, 1990.

[3] Batini, C. and Scannapieco, M. *Data Quality: Concepts, Methodologies and Techniques*. Springer, 2006.

[4] Bhagwat, D., Chiticariu, L., Tan, W. and Vijayvargiya, G. "An Annotation Management System for Relational Databases". *VLDB Journal*, 14(4), pp. 900-911, 2005.

[5] Benjelloun, O., Das Sarma, A., Halevy, A., Thobald, M. and Widom, J. "Databases with uncertainty and lineage". *VLDB Journal*, 17(2), pp. 243-264, 2008.

[6] Bovee, M., Srivastava, R., and Mak , B. "A Conceptual Framework and Belief Function Approach to Assessing Overall Information Quality". *Proceedings of IQ'01*, pp. 311-328.

[7] Cappiello, C., Francalanci, C., and Pernici, B. "Data Quality Assessment from the Users Perspective". *Proceedings of IQIS'04*, pp. 68-73.

[8] Catarci, T. and Scannapieco, M. "Data Quality under the Computer Science Perspective". *Journal of "Archivi & Computer"*, 2, 2002.

[9] Chengalur-Smith, I. N., Ballou, D. P. and Pazer, H. L. "The Impact of Data Quality Information on Decision Making: An Exploratory Analysis". *IEEE Transactions on Knowledge and Data Engineering*, 11(6), pp. 853-864, 1999.

[10] Dalvi, N. and Suciu, D. "Efficient query evaluation on probabilistic databases". *VLDB Journal*, 16(4), pp. 523-544, 2007.

[11] Elmagarmid, A.K., Ipeirotis, P.G., and Verykios, V.S. "Duplicate Record Detection: A Survey". *IEEE Transactions on Knowledge and Data Engineering*, 19(1), pp. 1-16, 2007.

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

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

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

[15] Helfert, M and Herrmann, C. "Proactive data quality management for data warehouse systems". *Proceedings of the 4th Intl. Workshop on Design and Management of Data Warehouses*, pp. 97-106, 2002.

[16] Jarke, M., Jeusfeld, M.A., Quix, C. and Vassiliadis, P. "Architecture and Quality in Data Warehouses: An Extended Repository Approach". *Information Systems*, 24(3), pp. 229-253, 1999.

[17] Jarke, M. and Vassiliou, Y. "Data warehouse quality: a review of the DWQ project". *Proceedings of IQ'97*, pp. 299–313.

[18] Leitheiser, R. L. "Data Quality in Health Care Data Warehouse Environments". *Proceedings of HICSS'01*.

[19] Levitin, A. and Redman, T. "Quality Dimensions of a Conceptual View". *Information Processing and Management*, 31(1), pp. 81-88, 1995.

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

[21] Naumann, F., Freytag, J. and Leser U. "Completeness of Integrated Information Sources". *Information Systems*, 29(7), pp. 583–615, 2004.

[22] Palpanas, T., Koudas, N. and Mendelzon, A. "Using Datacube Aggregates for Approximate Querying and Deviation Detection". *IEEE Transactions on Knowledge and Data Engineering*, 17(11), pp. 1465-1477, 2005.

[23] Pyzdek, T. *The Six Sigma Handbook, Second Edition*. McGraw-Hill, 2003.

[24] Raghunathan, S. "Impact of information quality and decision-maker quality on decision quality: a theoretical model and simulation analysis". *Decision Support Systems*, 26(4), pp. 275 -286, 1999.

[25] Ramakrishnan, R. and Gehrke, J. *Database Management Systems 3rd edition*. McGraw-Hill, 2002.

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

[27] Sarawagi, S. eds. Special issue on data cleaning. *IEEE Data Engineering Bulletin*, 23(4), 2000.

[28] Scannapieco, M. and Batini, C. "Completeness in the Relational Model: a Comprehensive Framework". *Proceedings of ICIQ'04*, pp. 333-345.

[29] Singh, S., Mayfield, C., Shah, R., Prabhakar, S., Hambrusch, S., Neville, J. and Cheng, R. "Database Support for Probabilistic Attributes and Tuples". *Proceedings of ICDE'08*, pp. 1053-1061.

[30] Strong, D.M., Lee, Y. and Wang, R.Y. "Data Quality in Context". *Communications of the ACM*, 40(5), pp.103-110, 1997.

[31] Su, Y., and Jin, Z. "A methodology for information quality assessment in the designing and manufacturing processes of mechanical products". *Proceedings of the ICIQ '04*, pp. 447-465.

[32] Theodoratos, D. and Bouzeghoub, M. "Data Currency Quality Factors in Data Warehouse Design". *Proceedings of DMDW '99*.

[33] Vassiliadis, P., Bouzeghoub, M. and Quix, C. "Towards Quality-oriented Data Warehouse Usage and Evolution". *Information Systems*, 25(2), pp. 89-115, 2000.

[34] Wang, R. Y. and Strong, D. M. "Beyond accuracy: what data quality means to data consumers". *Journal of Management Information Systems*, 12(4), pp. 5-33, 1996.