Heterogeneous data-integration and data quality: Overview of conflicts

This paper describes the state of the art of data quality in the data-integration process of heterogeneous data sources. We define the concepts of data quality, of "non-quality" data (their reasons and costs) and the dimensions of data quality. We focus then, on the problems of data quality in the data-integration process of heterogeneous data sources and we present different strategies to handle with conflicts. We give a small insight on our work to study the domain constraints global consistency. Some related work will be presented at the end.

This paper appears in: Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), 2012 6th International Conference on, Issue Date: 21-24 March 2012, Written by: Boufares, F.; Ben Salem, A.

SECTION I
INTRODUCTION

Current work on extracting knowledge from huge amounts of data in data warehouses focus on finding interesting information, not trivial, previously unknown and potentially useful. It therefore seems essential to evaluate the quality of data stored in databases and data warehouses. Indeed, the data used in the decision-making are becoming more and more complex. They have heterogeneous formats and they come from distributed sources. In a context where the challenges of the organizations and the administrations are becoming more numerous, have a data warehouse from the integration of heterogeneous sources [7] [14] becomes totally vital. Many questions should be asked during the integration process about the quality of data warehouse. For example, the global consistency of integrity constraints defined on the sources must be studied. The integration process has to take account not only the types of data but also their semantics.

Indeed, data quality and, more broadly, the information quality has taken a leading role within organizations and in the last ten years in academia. Data contribute to the success of the activity of an organization. Their quality is therefore a critical issue for the organization and it is more difficult to manage in the data-integration system due to data provenance from multiple sources. These sources are distributed, autonomous and heterogeneous and they have different schemas and levels of quality. The objective of data-integration system is to provide correct, complete, timely and consistent data, by defining a group of indicators easy to understand, to communicate, inexpensive and simple to calculate. Generally, the information quality is expressed or characterized by a set of attributes or factors that describe the data provided to users or processes which produce these data.

This paper is organized as follows: section 2 introduces the concept of data quality by defining the “non-quality” data and the dimensions of data quality. Section 3 highlights the problems of data quality in the data-integration process from heterogeneous sources. Section 4 presents the different handling strategies of conflicts. Section 5 gives a small insight for our contribution to study the constraints global consistency. Related work will be presented in section 6. Finally, are presented in conclusion our research perspectives.

SECTION II
DATA QUALITY

A. Data Quality definition

It is a generic term describing both the characteristics of data (comprehensive, reliable, relevant and timely, consistent) and the process which ensures them. The goal is to obtain data without: duplication, spelling errors, omission, and unnecessary change and conform to the defined structure. Data have a good quality if they satisfy the requirements of their users. In other words, the data quality depends on their use as well as their values. To satisfy the intended use, data must be accurate, timely, relevant, complete, understandable and trustworthy [23].

B. Data Quality in the databases

The term “quality” is a part of wide appreciative practical values. Quality is defined in terms of positive criteria and joining the concept of “excellence”.

We speak about the quality of a database (DB) if the DB can effectively represent for which it was originally conceived, so if it satisfies the needs of users.

Four characteristics resulting from the above [3]:

- The notion of quality is multidimensional: it may be related, for example, to the accuracy, consistency, credibility and opportunity of the information.
- Some dimensions are measurable (e.g., the logical consistency of a DB) and others are not (e.g., credibility of information disseminated by DB).
- Some dimensions are concurrent: recent and current information are not necessarily consistent because the process of test and correction did not finish. Inversely, coherent and consistent information will not necessarily be current and fresh.
- The notion of quality varies according to use: administrative operations of a DB require maximum precision in order to deal fairly with each record, but statistical manipulations of a DB tolerate a certain margin of error.

These four characteristics invalid the word “total quality” which is used in many studies on computer science quality. This quality, by definition, cannot be “complete”: it is inevitably a compromise between several criteria.

Managing data quality in the domain of DBs, consist on ensuring:

- Syntactic data accuracy (checking constraints that prevent, in case of violations, suspected data to be stored in the DB).
- Semantic data accuracy (that is, the conformity of the model and the data to truly reflect the real world modeled).

This traditional approach, relied on limited techniques such as integrity constraints and conflicts management, has been extended to allow the quality management especially in DBs, to lead the data-integration between different management systems of heterogeneous DBs.

C. Why managing the non-quality data?

According to ISO 8402, the “non-quality” is considered as the global status compared to the quality specified. It refers at the same time, to the direct product (or service) and to the associated actions for this product, which can be a step in customer satisfaction [18].

“Non-quality” is the global gap between the target and the obtained quality. This gap can be easily evaluated in economic terms. “Non-quality” is reflected in general by a defect, unconformity, or an anomaly [8]. So we can speak about nonquality data if there are errors on these data, duplicates, inconsistencies, or missing, incomplete, unclear and outdated values.

Landry [16] declares that among the reasons which lead to the “non-quality” are actually objective and known:

- Technical problems related to outdated information systems such as historic programming errors.
- Human and business problems such as data entry errors caused by employees, lack of standard repository or more generally a common language.

New approaches multi-channel or data exchange Business to Business (B2B) between partners that induce external data which present themselves potential sources of error.

The impact and thus the cost of “non-quality” data is not the same according the type of population (Customer Relationship Management, large account or Small and Medium Enterprises) and the use that it made (bank data, medical data, sensitive military data or CRM data). It is not easy to estimate the “cost of non-quality”. In addition, it is relatively easy to evaluate how much the implementation of an improved procedure cost. The expected benefits are more difficult to quantify due to unmeasurable aspects which follow the improvement of computer science system quality, such as credibility or reliability of the information.

As an example, several studies in the U.S.A in various sectors (such as banking, travel agencies) reported an error rate of 5% to 30% in the DBs (this rate is evaluated, for example, on the ratio between the number of records containing at least one logic error and the total number of records in a DB). In financial terms, the costs of “non-quality” are devalued on a loss about 5% to 10% of the income for the exanimate companies (such as costs of control, correction and maintenance of dubious quality data, costs of treatment for complaints from dissatisfied customers or to repair the damage) [3].

A study in the U.S.A, has estimated the cost, of non-quality of data handled by companies each year, more than 600 billion dollars [23].

D. Dimensions of data quality

To ensure data quality, Toulemonde [23] has defined a set of dimensions:

- *Duplication:* data are repeated. The entity is managed by several information systems under different identifiers and therefore its view is not unified.
- *Standard:* the values are correct compared to repair interval or domain. For a lack of coding standards, the organization “Hospital Cochin” may appear as “CHU Cochin”, “C.H.U Cochin” or “CHU C”.
- *Integrity:* all necessary data are available for the business need. It is impossible to make a campaign emailing customers with a DB that does not contain the email address.
- *Accuracy:* The data represent the reality where they are verifiable from an external source. The postal code does not match the locality, the phone has changed or the ID was not updated when moving from the organization.
- *Interpretability:* a data must be represented in an unambiguous format.
- *Opportunity:* Data are updated at time of use. The monthly sales report must include all the updated results of the month for all sales regions.

The data must have the necessary quality to support the type of use. In other words, quality is as important for data needed to evaluate a risk, as those used in a mass marketing operation.

It should be noted also that each organization must create its own operational definitions based on the objectives and priorities in order to define indicators for each dimension, and check by regularly measuring their progress over time.

Each dimension can be measured either subjectively by collecting user perception, or objectively through automatic monitoring of specific indicators.

SECTION III
Quality Problems in the Heterogeneous Data-Integration Process

To illustrate the problems of data quality, we chose to give concrete examples taken from two tables (Children and Patients) of different DBs. Children table represents, in the integration process, the list of sick children in a Children's hospital, while the Patients table represents the list of an hospital patients.

Structure of table Children (Tab. 1):

Component	Domain/Constraint
CNumero	VARCHAR2(10)
CNom	VARCHAR2(20)
CPreNom	VARCHAR2(20)
CDateNais	DATE
CSexe	CHAR(1) : M or F
CAge	NUMBER(3) : [0..10]
CAdresse	VARCHAR(100)
CGrade	NUMBER(2) : [0..20]
CCatSociale	NUMBER(1) : [0..5]

TABLE I. Children Table

The structures of the two tables (Tab. 1 and Tab. 2) are not compatible. The incompatibilities have two types: syntactic and semantic.

Structure of table Patients (Tab. 2):

Component	Domain/Constraint
PNumero	VARCHAR2(10)
PNom	VARCHAR2(20)
PPreNom	VARCHAR2(20)
PDateNais	DATE
PSexe	NUMBER(1) : 0 or 1
PAge	NUMBER(3) : [0..150]
PAdrNum	VARCHAR(10)
PAdrRue	VARCHAR(50)
PNumero	VARCHAR2(10)
AdrCP	VARCHAR(5)
PAdrVille	VARCHAR(20)
PGrade	NUMBER(2) : [0..10]
PCatSociale	NUMBER(1) : [0..10]
PProfession	VARCHAR(20)

TABLE II. Patients Table

The table below (Tab. 3) summarizes these incompatibilities:

Children	Patients	Compatibility
CNumero	PNumero	Yes
CNom	PNom	Yes
CPreNom	PPreNom	Yes
CDateNais	PDateNais	Yes
CSexe	PSexe	No
CAge	PAge	No
CAdresse	PAdrNum	No
	PAdrRue	
	AdrCP	
	PAdrVille	
CGrade	PGrade	No
CCatSociale	PCatSociale	No
	PProfession	No

TABLE III. INCOMPATIBILITY BETWEEN THE TWO TABLES (CHILDREN AND PATIENTS)

By integrating or merging data from the two tables Children and Patients, a new table is created called Sick whose data are heterogeneous and we should answer to such questions:

- What is the structure of the result table, what are its constraints?
- What are the transformations that we have to do?
- How to correct erroneous data?
- How to eliminate the duplicates data? What are duplicates data?

```
CREATE TABLE Sick (
SNumero VARCHAR2(10), SNom VARCHAR2(20),
SPreNom VARCHAR2(20), SDateNais DATE,
SSexe CHAR(1), SAdrNum VARCHAR(10),
SAdrRue VARCHAR(50), SAdrCP VARCHAR(5),
SAdrVille VARCHAR(20), SCatSociale NUMBER(1),
SProfession VARCHAR(20),
CONSTRAINT PK_Sick PRIMARY KEY (SNumero),
CONSTRAINT NN_SNom CHECK (SNom IS NOT NULL),
CONSTRAINT CK_SSexe CHECK (SSexe = 'M' OR
SSexe = 'F'), CONSTRAINT CK_SCatSoc CHECK
(SCatSociale BETWEEN 1 AND 10));
```

How to describe this new structure of the integration of these two tables?

In the literature, many problems were identified when analyzing a set of different data sources as a whole. Even if individually each source has no quality problems, their simultaneous manipulation in a data analysis or integration operation (for instance) may result in the appearing of data quality problems among the data sources. Among these problems, we have the eight below [5], [6], [19]:

A. Syntax inconsistency

Depending on the data source, there are different representation syntaxes among attributes whose type is the same. For example, in the table **Children** from DB1, the attribute *CDateNais* has the syntax (dd/mm/yyyy), while in the table **Patients** from DB2; the attribute *PDateNais* has the syntax (yyyy/mm/dd).

Another problem with the date format is: if a francophone person who is responsible for human resources records children information in DB1 and uses the date format (dd/mm/yyyy) for the dates of birth of **Children** *CDateNais* and in the other hand, an Anglo-Saxon person records the patients information in DB2 and uses the date format (mm/dd/yyyy) for dates of birth of **Patients** *PDateNais*. This type of conflict can also arise problems in the integration process. Then displayed as *11/12/2010* on the screen of a human resources staff in Paris, date of birth of a child (the component *CDateNais)* is correct, and then it is displayed *12/11/2010* for the component *PDateNais* on the screen of his Anglo-Saxon colleague.

B. Different measure units

Depending on the data source, different measure units are used in attributes that are related. For example, the component weight of patients in DB1 is measured in kg while in DB2 is measured in pounds.

Toulemonde [23] in his paper gave as an example for this problem, NASA in 1999 lost the Mars Climate Orbiter satellite due to erroneous data. Indeed, the satellite was destroyed during its orbit around Mars at an altitude of 50 km from the surface (altitude normally provided was 150 km) by air turbulence and friction. The survey showed that some parameters were calculated, by Anglo-Saxon measurement units and they are transmitted to the navigation team, waiting data in units of the metric system. This “small” error has cost 125 million dollars to American taxpayers.

C. Inconsistency representation

Different sets of values, from the same type or not, are used in related attributes from distinct data sources to represent the same situations. For example, to represent the attribute *CSexe* the values F and M are used in DB1, while in DB2 the values 0 and 1 are used *for PSexe* attribute.

However this problem of inconsistency seems to be less complicated if we have two datasets with the same number of values as each value has its image in the other set (we can have 0 corresponds to M and 1 corresponds to F or the inverse), but it becomes more complicated if we start with two sets with a different number of values for each, as the case of social class component of a child *CCatSociale* which is between 0 and 5 in DB1 and *PCatSociale* component of a patient which is between 0 and 10 in DB2.

In this case, we have a classification problem with different values for each dataset since there is no bijection between values. Several cases are considered: some values return more than one value and other return zero values in a dataset. We must find a solution that allows the best classification for these values.

D. Redundancy about an entity

The same entity is represented bwy an equal or equivalent representation in more than one tuple from different data sources. For example, the tuple **Children**< 10, ‘Tabib Adam’, 01/12/2007, ‘123, rue de la Paix’ > in DB1 is equivalent to the tuple **Patients**< 27, Tabib A. ‘,01/12/2007, ’123, rue de la Paix'> in DB2.

E. Inconsistency about an entity

There are inconsistencies or contradictions among one or more attribute values of a same entity, represented in more than one tuple in different data sources. For example, the tuple **Children**<10, 'Tabib Adam', *01/12/2007*, '123, rue de la Paix'> in DB1 is inconsistent with the tuple **Patients** <27, 'Tabib Adam', *01/12/2007*, '321, Place de la Paix' > in DB2.

F. Cardinality constraint violation

Equivalent tables belonging to different sources individually respect a constraint inherent to the domain, but violate it when considered as a whole. For example, in an hospital service the number of administrative staff should be less than 10. We can have 7 persons in the service S1 and 6 people are working in the service S2. The union of these two services should not give 13 people.

G. Source dependence

Dong and all [12] defined the problem of sources dependence (original and copy problem). In many applications, data management, require the data-integration from multiple sources. Each source provides a set of values and other sources can often provide conflicting values. To submit quality data to users, it is essential that data-integration systems can resolve conflicts and discover the true values. In general, we expect real values to be supplied by other sources than the false values, and then we can take the value given by most sources as truth. Unfortunately, a false value can be spread by copying and makes the discovery of truth extremely difficult.

Data sources can also easily copy, reformat and modify data from other sources, propagating erroneous data. These issues make the identification of high quality information and sources non-trivial.

H. Semantic inconsistency

Let us consider the previous two tables (Children and Patients). Given the attributes synonymous below (for a person the age and the grade) (Tab. 4):

Children

CNum	CGrade	CAge
C1	17	5
C2	7	7
C3	10	10

Patients

CNum	CGrade	CAge
P1	7	77
P2	3	3

TABLE IV. SAMPLE DATA FOR THE TWO TABLE CHILDREN AND PATIENTS

The grades, in English exam, for children C1, C2 and C3 are noted on 20, while the grades of patients P1 and P2 are noted on 10.

The integration of these data provides the result table Sick:

SNum	SGrade	SAge
C1	17	5
C2	7	7
C3	10	10
P1	7	77
P2	3	3

TABLE V. SICK TABLE: INTEGRATION OF DATA WITHOUT ANY TRANSFORMATION

Syntactic integration according to the type of data is not sufficient when used types are compatible (in our example, the grade has the same type Number in the both tables). But, even with the same syntactic type we have two different intervals for each grade.

The result of the integration data will be as shown in the table Sick (Tab. 5). So, to have controls noted on the same scale, a semantic processing is required. Grades should be recorded in the same way (either on 10 or on 20), as presented in the tables SickT1, SickT2 (Tab. 6).

SickT1

SNum	SGrade /10	SAge
C1	8.5	5
C2	3.5	7
C3	5	10
P1	7	77
P2	3	3

SickT2

SNum	SGrade /20	SAge
C1	17	5
C2	7	7
C3	10	10
P1	14	77
P2	6	3

TABLE VI. DATA INTEGRATION WITH TWO DIFFERENT TRANSFORMATIONS

However, for ages, the integration is syntactically and semantically correct, and we have no need for transformation.

The questions here are: How to store these semantics information? And what they represent?

SECTION IV

CONFLICT HANDLING STRATEGIES

Previously, we presented a partial list of problems on data quality in different data-integration systems and in particular those whose sources are heterogeneous.

Several approaches [22], [20], [17], [13] and recently [10] address some conflict handling strategies. The approach proposed by Bleiholder and Nauman [10] is based on handling the two conflicts (contradiction and uncertainty) in the data integration phase.

A *contradiction* is a conflict between two or more different non-null values used to describe the same property of an object. In the data-integration process, two or more data sources can provide two or more different values *for* the same attribute.

An *uncertainty* is a conflict between a non-null value and one or more null values used to describe the same property *of* an object. This is caused by missing information, e.g., null values in the table, or by an attribute completely missing in one table. In the latter case, we assume that the missing attribute values are added with null values.

There are several simple strategies for dealing with these conflicts. They can be classified as seen in Fig 1 and fall into three main classes. The first division of strategies into the three classes is based on the way they handle (or do not handle) conflicting data: ignorance, avoidance, and resolution.

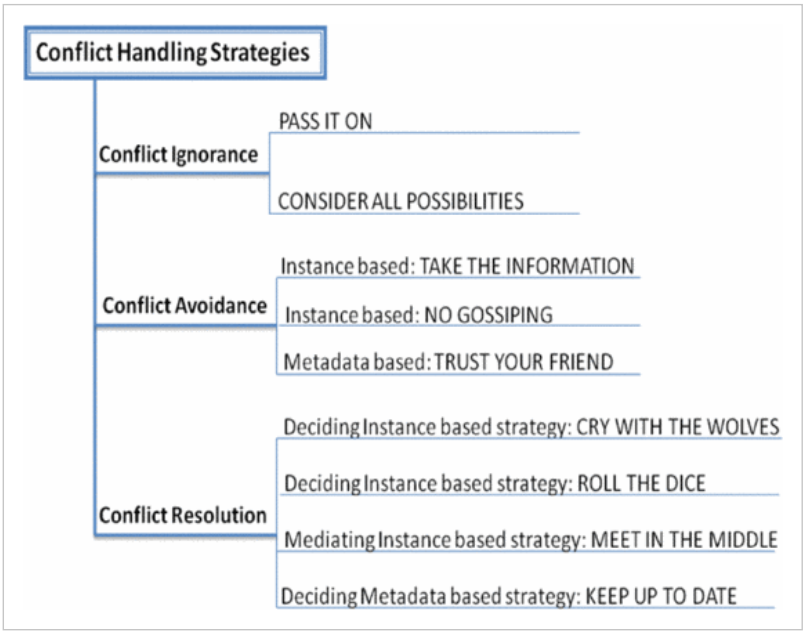

Figure 1. A classification of the conflict handling strategies [10]

A. Conflict ignorance

Conflict ignorance describes strategies that do not respect any conflict to make a decision. These strategies are easy to implement in the integration process [10]. Two representatives are:

- *Pass it on*. This strategy simply takes all conflicting values, passes them on to the user and lets him deciding how to handle possible conflicts among the values.
- *Consider all possibilities*. This strategy tries to be as complete as possible by enumerating all eventualities and giving the user the choice among all "possible worlds", all possible combinations of attribute values, occasionally creating combinations that are not already present in the sources.

B. Conflict avoidance

The conflict avoiding strategies can be divide into two classes. One that takes metadata into account when taking a decision and one that does not. These strategies handle inconsistent data. They do not regard the conflicting values before deciding on how to handle inconsistencies [10].

The basic idea in instance based strategy is that existing information is taken and information not present is left aside. The Take the information strategy leaves aside null values and is the natural way of dealing with uncertainties.

No gossiping is also in instance based strategy. It is used by the consistent query answering approaches. If you are unsure how to handle inconsistencies, why no. 1eave them out and report only on the certain facts? Here, only consistent answers, fulfilling certain constraints on the query, are included in the result of a query leaving aside all inconsistent ones.

Trust your friends is an example of a metadata based conflict avoidance. The intuition behind this strategy is to trust a third party to either provide the correct value or the correct strategy. Whom to trust is decided once and carried out for all data values, no matter if there is a conflict or not.

More details can be found in [10].

C. Conflict resolution

In contrast to the conflict ignoring and the conflict avoidance strategies, conflict resolution ones do regard all the data and metadata before deciding on how to resolve a conflict. This approach is computationally more expensive than other strategies but provides means to resolve the conflict as flexibly as possible [10].

Conflict resolution strategies are subdivided into deciding and mediating strategies. The main characteristic of a deciding strategy is that it chooses its value from all the already present values.

- *Cry with the wolves* is an instance based strategy. The intuition of this strategy is that correct values prevail over incorrect ones, given enough evidence. It reflects the principle of following the decision of the majority, of choosing the most common value among the conflicting ones.

For example, for the problem of inconsistent representation and if it is used in different DBs, the syntax representation of the Sexe component, 'M' and 'F' and in some DBs it is used the syntax 0 and 1, we will opt for majority representation which is 'M' and 'F'.

- *Roll the dice* is also an instance based strategy. It considers all conflicting values and picks one at random. Although this may not seem to be a very intelligent decision, it is still a valid strategy to resolve conflicts. Lacking any input to decide upon a value, a random value is a good choice.

For the same example *of* the *CSexe* component, whether there are DBs that use the syntax 'M' and 'F' and others which use 0 and 1, we can randomly choose a format and apply it to everyone.

- *Keep up to date* is a metadata based strategy. This strategy uses the most recent value and requires some additional time-stamp information about the recency. This information can be present in the tables as a separate attribute or can be provided by other means, such as data lineage facilities. In a data stream environment there is a naturally given *order* of the tuples coming in so that the recency lies in the data itself.

Mediating strategies on the other hand may choose a value that does not necessarily exist among the conflicting values. An example of a mediating strategy:

- *Meet in the middle*. This strategy follows the principle of compromise and does not prefer one value over the other but instead tries to invent a value that is as close as possible to all present values. Another principle used can be to minimize the error, or to take the average.

For example, the component address of a child, *CAdresse* in DB1 is "13, Epinay sur Seine" and the address of a patient *PAdresse* in DB2 is "13, Seine s Epinay" or even "13, Epinay/seine". In the integration, we can for example choose "13, Epinay sur Seine," which seems to be the clearest and most comprehensive.

We can find in the literature some algorithms which compare Strings [9], [11], [24].

SECTION V
Our Contribution

Our goal is to examine the domain constraints global consistency in our data-integration system for heterogeneous sources. For this, we developed an algorithm [4] that defines the necessary steps to study the domain constraints global consistency to ensure proper data-integration.

Our process initially examines the domain constraints global consistency for a single component. For example, if a user has specified that patient age should be in one hand *PAge>0* and the other hand *PAge<0*.

The current DB Management Systems (such as ORACLE 109, MySQL or PostGres) accept this type of declaration statement. The problem is detected only during the data creation. We wanted to allow the user to « compile » from the beginning this kind of conflict by checking the consistency of these constraints [BB07], [5].

The new process has to study the global coherence for two components to integrate. As the example already presented above: *CGrade* with the constraint K1 = [0. 20] and *PGrade* with the constraint K2 = [0.. 10].

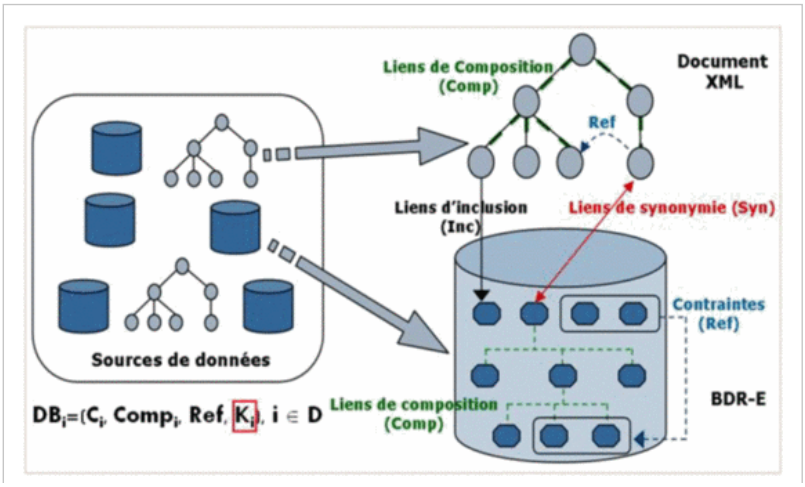

Figure 2. Integration environment for heterogeneous data sources

For the integration we will have no syntactic conflict, but we may have semantic conflicts. So in this case, we must decide which constraint to consider and which to ignore.

For our final data warehouse DW (the result after data-integration), we have only *one* generic constraint K among these various constraints. To resolve this problem, we decide to use the method Cry with the wolves, a method already defined in the previous paragraph as a strategy to resolve conflicts [10].

A final step is to define the necessary transformations by considering the generic constraint K. Otherwise, the data will be erroneous and therefore of non-quality.

SECTION VI
Related Work

The data-integration from multiple sources is a well known and mature problem but it lacks the approaches dealing with data quality.

The approach of Peralta and all [21] uses a set of dimensions quality (freshness and accuracy). They defined a Framework and algorithms based on the quality factors for data-integration systems. Also, in the approach of Motro and all [17], they defined a linear function, “Utility”, which includes a set of features for many dimensions quality. This function ensures the management of data inconsistencies in the integration process of multiple sources. While in our contribution, we are interested to study the constraints and their consistencies in the data-integration systems from heterogeneous sources (Fig 2).

In their work, Akoka and all [1] have defined a meta-model to evaluate and improve quality. This meta-model defines the objectives of quality, its dimensions, factors and metrics. However, in our contribution, we built a meta-model that implements the elements necessary to study the consistency of constraints and especially domain constraints. Our meta-model thus allows defining different types of domains with a different number of values for each component of our DW.

The approach Bleiholder and Nauman [10] handle the conflicts (contradiction and uncertainty) in the data fusion. They identify various strategies to classify the conflicts and use these strategies to decide between the different constraints. However, in our contribution, we use the presence of semantic information to decide between the different domain constraints. We usually have for one case, several choices of the appropriate constraint for better data-integration.

The work of Arenas and all [2] and Fuxman and all [13] consist on developing algorithms for rewriting queries based on user-defined constraints to provide consistent answers queries. For this, they define the ConQuer system which takes care of rewriting these applications, with the respect of the constraints already defined, in SQL. It handles inconsistencies by filtering them.

SECTION VII
Conclusion

Nowadays, complex applications such as extraction of knowledge, data mining, e-learning and web applications use heterogeneous and distributed data. In this context, the need for integration and evaluation of data quality is increasingly felt. The Web has accelerated the rate at which useful information is produced and disseminated, but has also eased the ability to spread false information. Data quality is a very important subject; it concerns everyone in all application areas (scientific, governmental, commercial or industrial).

This paper describes an ongoing research project dedicated to the evaluation and improvement of data quality in enterprise information systems. A list *of* problems has been identified; some approaches for handling some *of* these conflicts have been presented. We focus on the constraints at any level of the integration. The constraints defined on the data sources must be respected. For that, a study on the domain constraints global consistency, in one hand, and a study on the consistency of other constraints such as cardinality ones or functional dependencies, in the other hand, are under development to be *added* in our heterogeneous data integration process [7], [14].

A formal approach for evaluating the data quality in a heterogeneous data warehouse (the result of integration) will be studied, by considering not only syntactic but also semantic aspects. The consequences on the knowledge extraction will be explained.

Recent works highlight the importance of documents and their textual data for decision-making. These documents (e.g. reports, customer complaints) allow better understanding and explain some activities of the organization. Generally, these documents are available in XML format and are described by different structures. So, integrating the XML documents create a big challenge which is the data quality. This is a part of our future research.

FOOTNOTES

No Data Available

REFERENCES

1. J. Akoka, L. Berti-Équille, O. Boucelma, M. Bouzeghoub, I. Comyn-Wattiau, M. Cosquer, V. Goasdoué, Z. Kedad, S. Nugier V. Peralta, M. Quafafou, S. Sisaïd-Cherfi, "Évaluation de la qualité des systèmes multisources. Une approche par les patterns", Proccedings of the 2nd Workshop on Data and Knowledge Quality (QDC 2008) in conjunction with the French National Conf. On Extraction and Management of Knowledge (Extraction et Gestion des Connaissances-EGC), Nice, France, January 29, 2008, pp. 1-9.

Show Context

2. M. Arenas, L. Bertossi, J. Chomicki, "Consistent Query Answers in Inconsistent Databases", Proccedings ACM Symposium on Principles of Database Systems (PODS), 1999, pp. 1-12.

Show Context

3. I. Boydens, « Evaluer et améliorer les qualités des bases de données », Techno 7, Publication technique de la Smals-MvM, Bruxelles, 1998, pp. 1-11.

Show Context

4. A. Ben Salem, F. Boufarès, Mémoire de stage de recherche en Master 2 PLS, Laboratoire d'Informatique de Paris Nord (LIPN), Institut Galilée-Université Paris 13, Septembre 2010, Villetaneuse, France.

Show Context

5. D. Berrabah, F. Boufarès, "Constraints Satisfaction Problems in Data Modeling", Proccedings of the 5th International Conference on Soft Computing as Transdisciplinary Science and Technolody (IEEE Systems, Man, and Cybernetics Society), (CSTST'08), October 27-31, 2008, Cergy-Pontoise Paris, France, pp. 292-297.

Show Context

6. D. Berrabah, F. Boufarès, "Constraints Checking in UML Class Diagrams: SQL vs OCL", Proccedings of the 18th International Conference of Database and Expert Systems Applications (DEXA'07), September 3-7, 2007, Regensburg, Germany. LNCS no4653, pp. 593-602. Springer 2007.

Show Context

7. M. Badri, F. Boufarès, S. Hamdoun, V. Heiwy, K. Lellahi, Construction and Maintenance of Heterogeneous Data Warehouses, Information Sciences reference: Data Warehousing Design and Advanced Engineering Applications Methods for Complex Construction, Hershey, New York, 2009, pp. 189-204.

Show Context

8. L. Berti-Équille. "Qualité des données", Techniques de l'Ingénieur H3700, collection Technologies logicielles - Architecture des systèmes, 2006, pp. 1-19.

Show Context

9. M. Bilenko, R.J. Mooney, "Adaptive Duplication Detection Using Learnable String Similarity Measures", Proccedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD' 03), Washington, DC, USA, 2003, pp. 39-48.

Show Context

10. J. Bleiholder and F. Naumann, "Conflict Handling Strategies in an Integrated Information System", Humboldt-Universitat zu Berlin Unter den Linden 6 Berlin, May 22-26, 2006, Edinburgh, UK, pp. 1-6.

Show Context

11. W.W. Cohen, J. Richman, Learning to Match and Cluster Large High-Dimentional Data Sets For Data Integration, Proccedings The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Alberta, Canada, 2002.

Show Context

12. X.L. Dong, L. Berti-Equille, and D. Srivastava, "Integrating conflicting data: the role of source dependence", Proccedings of the International Conference on Very Large Databases (VLDB'09), Lyon, France, August 2009.

Show Context

13. A. Fuxman, E. Fazli, R. J. Miller, "Con Quer: Efficient Management of Inconsistent Databases", Proccedings of the ACM SIGMOD International Conference on Management of Data 2005, Baltimore, Maryland, USA, June 2005, pp. 155-166.

Show Context

14. S. Hamdoun, F. Boufares, "Un formalisme pour l'intégration de données hétérogènes", 6ème journées Francophones sur les entrepôts de données et l'Analyse en ligne (EDA'10), Djerba (Tunisie), 11-13 juin 2010, pp. 107-119.

Show Context

15. D. Kostadinov, V. Peralta, A. Soukane, X. Xue, "Intégration de données hétérogènes basée sur la qualité", Grenoble, 2005, pp. 1-16.

16. X. Landry, "La qualité de données en tant que pierre angulaire des projets d'intégration", Decideo.fr, 17 Février 2010.

Show Context

17. A. Motro, P. Anokhin, A.C. Acar, "Utility-based Resolution of Data. Inconsistencies", Proccedings of International Workshop on Information Quality in Information Systems (IQIS'04), Maison de la Chimie, Paris, France, 2004, pages 35-43.

Show Context

18. E.F. Mami, A. Benhabib, S. Ghomri, "Les coûts du non qualité", Telemcen-Algérie, 2004, pp. 1-20.
Show Context

19. P. Oliveira, F. Rodrigues, P. Henriques, H. Galhardas, "A Taxonomy of Data Quality Problems", 2nd International Workshop on Data and Information Quality (DIQ' 05), Porto, Portugal, 2005, pp. 219-233.
Show Context

20. Y. Papa konstantinou, S. Abiteboul, and H. Garcia-Molina, "Object fusion in mediator systems", In Proccedings of International Conference on Very Large Databases (VLDB'96), Bombay, India, 1996, pp. 413-424.
Show Context

21. V. Peralta, R. Ruggia., M. Bouzeghoub, "Data Quality Evaluation in a Data-integration System", Proccedings of the 2rd Alberto Mendelzon International Workshop on Foundations of Data Management (AMW'2007), Punta del Este, October 26th 2007, pp. 1-25.
Show Context

22. V.S. Subrahmanian, S. Adali, A. Brink, R. Emery, J. Lu, A. Rajput, T. Rogers, R. Ross, and C. Ward, "Hermes: A heterogeneous reasoning and mediator system", Technical report, University of Maryland, 1995.
Show Context

23. C. Toulemonde, JEMM research-Informatica, "Exploiter le capital de votre organisation", Un livre blanc de JEMM research - Des données de qualité, 2008, pp. 1-26.
Show Context

24. W.E. Winkler, "The state of record linkage and current research problems", Statistics of Income Division, Internal Revenue Service Publication R99/04, 1999.
Show Context

AUTHORS

F. Boufares
No Bio Available

A. Ben Salem
No Bio Available

CITED BY

None

KEYWORDS

IEEE Keywords

Accuracy, Data warehouses, Organizations, Pediatrics, Semantics, Syntactics

INSPEC: Controlled Indexing

data integration, data integrity, data warehouses

INSPEC: Non-Controlled Indexing

conflict handling, data quality, data warehouse, deterogeneous data-integration, domain constraint global consistency, heterogeneous data sources

Authors Keywords

Data Quality, Data Warehousing, Data-integration process of heterogeneous data sources, Domain Constraints

CORRECTIONS

None