Trusted Data in IBM's Master Data Management

Przemyslaw Pawluk, Jarek Gryz
York University
Toronto ON, Canada
Center for Advanced Studies
IBM, Toronto, ON Canada
Email: {pawluk, jarek}@cse.yorku.ca

Stephanie Hazlewood, Paul van Run
IBM Laboratory
Toronto ON, Canada
Email: {stephanie, pvanrun}@ca.ibm.com

***Abstract*—A good business data model has little value if it lacks accurate, up-to-date customer data. This paper describes how data quality measures are processed and maintained in *IBM InfoSphere MDM Server* and *IBM InfoSphere Information Server*. It also introduces a notion of *trust*, which extends the concept of data quality and allows businesses to consider additional factors, that can influence the decision making process. The solutions presented here utilize existing tools provided by IBM in an innovative way and provide new data structures and algorithms for calculating scores for persistent and transient quality and trust factors.**

***Keywords*-Master Data Management; Data Integration; Data Quality; Data Trust**

I. Introduction

Many organizations have come to the realization that they do not have an accurate view of their business-critical information such as customers, vendors, accounts, or products. As new enterprise systems are added, silos are created resulting in overlap and inconsistency of information. This varied collection of systems can be the result of systems introduced through mergers and acquisitions (M&A), purchase of packaged applications for enterprise resource planning (ERP) or customer relationship management (CRM), different variant and versions of the same application used for different lines of business or home grown applications. Data in these systems typically differs both in structure and in content. Some data might be incorrect, some of it might just be old, and some other parts of it might show different aspects of the same entity (for example, a home vs. a work address for a customer).

Master Data Management (MDM) is an approach that decouples master information from the applications that created it and pulls it together to provide a single, unified view across business processes, transactional and analytical systems. Master data is not about all of the data of an organization. It is the data that deals with the core facts about the key entities of a business: customers, accounts, locations and products. Master data is high value data that is commonly shared across an enterprise – within or across the lines of business. MDM applications, such as IBM's InfoSphere Master Data Management Server, contain functionality to maintain master data by addressing key data issues such as governance, quality and consistency. They maintain and leverage relationships between master data entities and manage the complete lifecycle of the data and support multiple implementation approaches.

The quality of master data requires special attention. Different aspects or dimensions of quality need to be considered and maintained in all processes of the enterprise. Trust scores, introduced in this paper, can provide important information to the decision makers. Our approach to the quality of data is slightly different than described so far in the literature [1]–[3]. Our goal is to provide the user with the information about data quality and trust. Trust in this case is the aggregated value of multiple factors, and is intended to cover quality and non-quality aspects of master data. We are not making the attempts to build fixes nor enforce any quality policy. The information provided by us is intend to identify weaknesses of data quality. The data quality enforcement should be then improved based on this information.

This paper focuses on the creation of measures, or trust factors, that serve to determine the trustworthiness of data being managed by MDM applications, specifically those being introduced in IBM's InfoSphere MDM Server. This new notion involves creating *trust scores* for these trust factors that enhance the notion of data quality and the more broad quality-unrelated features such as lineage, security, stewardship etc. All these have one goal – to support businesses in the decision making process, or data stewardship by providing information about different aspects of data.

This work is organized as follows. Section III presents the underpinning principles of Master Data Management (MDM), related concepts as well as the tools we used to prepare the trust scoring prototype. In Section IV we introduce a sample business scenario through which we explain the main ideas in the paper. Section V provides a short overview of data quality and introduces the notion of trust. Section VI presents structures and methods used to acquire, store and process trust factors.

II. Background and Related Work

Data Quality has been explored by several researchers in recent years and its importance has been discussed many times in the literature [1]–[14] usually in context of the single data source. However, some researches have been also done in the context of integrated data [15]–[20]. Most of them present strictly theoretical approach to the topic and provides solutions that are hard to apply or expensive. Moreover, all of them

Copyright (c) IARIA, 2011 ISBN:978-1-61208-115-1

includes only quality factors into the considerations excluding several extremely important non-quality factors such as data lineage or security. Different approaches to data quality and chosen definitions are presented more detailed in Section V.

In this work we would like to extend the notion of data quality and introduce new notion called *data trust* which covers both quality and non-quality factors. We propose a set of tools that can be used to process this information.

III. MDM AND INFORMATION SERVER

Master data management is a relatively fast growing software market. Many customers acknowledge they have data quality and data governance problems and look to large software vendors like IBM for solutions to these problems. Crucial parts of these MDM solutions are data quality and data trust mechanisms [21]–[23]. In this section, we are presenting the MDM environment and the comprehensive approach to the trust and quality that utilizes tools provided by IBM.

A. Definitions

Master Data Management (MDM) provides the technology and processes to manage Master Data in an organization. Master Data is the data an organization stores about key elements or business entities that define its operation. "An MDM solution enables an enterprise to govern, create, maintain, use, and analyze consistent, complete, contextual, and accurate master data information for all stakeholders, such as line of business systems, data warehouses, and trading partners." [21] Master data is high value information that an organization uses repeatedly across many business processes and lines of businesses. For these to operate efficiently, this master data must be accurate and consistent to ensure good decisions. Unfortunately in many organizations, master data is fragmented across many applications, with many inconsistent copies and no plan to improve the situation.

Traditional approaches to master data include the use of existing enterprise applications, data warehouses and even middleware. Some organizations approach the master data issue by leveraging dominant and seemingly domain-centric applications, such as a customer relationship management (CRM) application for the customer domain or an enterprise resource planning (ERP) application for the product domain. However, CRM and ERP, among other enterprise applications, have been designed and implemented to automate specific business processes such as customer on-boarding, procure-to-pay and order-to-cash, not to manage data across these processes. The result is that a specific data domain, such as customer or product, may actually reside within multiple processes, and therefore multiple applications.

In general, master data management (MDM) solutions should offer the following:

- Consolidate data locked within the native systems and applications
- Manage common data and common data processes independently with functionality for use in business processes
- Trigger business processes that originate from data change
- Provide a single understanding of the domain-customer, product, account, location — for the enterprise

Depending on MDM tool those requirements are realized in a different way. Some products decouple data linked to source systems so they can dynamically create a virtual view of the domain, while others include the additional ability to physically store master data and persist and propagate this information. Some products are not designed for a specific usage style, while others provide a single usage of this master data. Even more mature products provide all of the usage types required in today's complex business-collaborative, operational and analytic-as out-of-the-box functionality. These mature products also provide intelligent data management by recognizing changes in the information and triggering additional processes as necessary. Finally, MDM products vary in their domain coverage, ranging from specializing in a single domain such as customer or product to spanning multiple and integrated domains. Those that span multiple domains help to harness not only the value of the domain, but also the value between domains, also known as relationships. Relationships may include customers to their locations, to their accounts or to products they have purchased. This combination of multiple domains, multiple usage styles and the full set of capabilities between creating a virtual view and performance in a transactional environment is known as multiform master data management.

Some of the most common key business drivers for MDM are:

- Revenue Enhancement - More intelligent cross-sell and up-sell via complete understanding of customer (profile, accounts and interactions) to leverage bundling opportunities;
- Consistent Customer Treatment - Blending channels to deliver common customer interactions/experiences across all touch points;
- Operational Savings and Efficiencies - "Once & done" enterprise-wide services for key customer processes such as account changes (name, address);
- Privacy & Regulatory Compliance - Central location for consistent rules of visibility & entitlements;
- M&A Infrastructure - Shortening M&A customer, desktop, and billing integration time frames;

Achieving a high level of data quality is key prerequisite for many of the MDM objectives. Without high quality data the best analytics and business intelligence applications are still going to deliver unreliable input to important business decisions. Another key aspect of the management of the master data is achieving a high level of trustworthiness in the data. It is a key factor for customers to have reliable information about the data. Information about the quality, the origin, the timeliness and many other factors influencing the business decisions based on the provided data.

The introduction of *data governance* in the organization is a

Copyright (c) IARIA, 2011 ISBN:978-1-61208-115-1

vital prerequisite to come to more trusted information. Moving to master data management can be the cornerstone of a data governance program. It is important however, to note that at the same time, moving to MDM cannot be successful without data governance.

Data governance is defined as "the orchestration of people, process and technology to enable an organization to leverage information as an enterprise asset" [24]. It manages, safeguards, improves and protects organizational information. The effectiveness of data governance can influence the quality, availability and integrity of data by enabling cross-organizational collaboration and structured policy-making.

B. MDM Tools

IBM InfoSphere MDM Server is an application that was built on open standards and the Java Enterprise Edition (JEE) platform. It is a real-time transactional application with a service-oriented architecture that has been built to be scalable from both volume and performance perspectives. Shipping with a persistent relational store, it provides a set of predefined entities supporting the storage of master data applicable to each of the product's predefined domains.

This product also includes the MDM Workbench – an integrated set of Eclipse plug-ins to IBM Rational Software Architect/Developer that support the creation of new MDM entities and accompanying services, and a variety of extensions to MDM entities. This tooling reduces the time and breadth of skills required for solution development tailored to the business and allows for flexibility to changing business requirements with its model-driven approach to solution development. We also use the new module of the IBM Information Server Suite, *IBM InfoSphere Information Analyzer* that profiles and analyzes data so that the system can deliver trusted information to users. The Information Analyzer (IA) will be used to scan or sample data in data sources to assess the quality. It enables us to discover the structure of the quality and to give some guidelines how it can be improved. We are also using the complementary tools which are: *IBM InfoSphere QualityStage*, which allows us to define rules to standardize and match free-form data elements which is essential for effective probabilistic matching of potentially duplicate records, and *IBM WebSphere AuditStage*, which enables us to apply professional quality control methods to manage the accuracy, consistency, completeness, and integrity of information stored in databases. We also use statistics provided by *IBM InfoSphere DataStage* to compute chosen quality and trust factors.

This set of tools enables us to create the comprehensive approach to the data quality and data trust management. This approach not only resolves some problems during the data acquisition but also allows us to control the level of data trust and to give up-to-date information about the trustworthiness to the user. This comprehensive approach is novel. Moreover our solution does not require any specialized hardware or operating system and is able to cooperate with many commercial data base solution like DB2, Oracle and others.

1) IBM InfoSphere MDM Server: The InfoSphere Master Data Management Server has a new feature allowing users to define and add quality and trust factors to the data of their enterprise. This new data structure enables the user to store data required to compute scorings for trust and quality of data. Provided wizards allow the user to modify the data model in a simple way.

2) IBM Information Server: IBM Information Server addresses the requirements of cooperative effort of experts and data analysts with an integrated software platform that provides the full spectrum of tools and technologies required to address data quality issues [25]. It supplies users and experts with the tooling that allows the detailed analysis of data through profiling (IBM InfoSphere Information Analyzer and IBM InfoSphere AuditStage), cleansing (QualityStage) and data movement and transformation (DataStage). In this paper we concentrate on data profiling and analysis. Our work is focused mostly on IBM InfoSphere Information Analyzer (IA), IBM InfoSphere AuditStage (AS), and partially on QualityStage (QS).

IA, as an important tool of *data quality assessment* (DQA) process, aids the exposing technical and business issues. The technical issues detection is a simpler part of the process based on technical standards and covers following problems:

- Different or inconsistent standards in structure, format, or values
- Missing data, default values
- Spelling errors
- Data in wrong fields
- Buried information in free-form fields

Business quality issues are more subjective and are associated with business processes such as generating accurate reports. They require the involvement of the experts. IA helps the expert in systematic analysis and reporting of results, thereby allowing him to focus on the real problem of data quality issues. This is done through the following tasks:

Column Analysis – can be performed on all the columns of one or more tables, or selectively on certain columns and allows to run an analysis on the full volume of data, or on a subset using a sampling technique. As a result of this process reference tables can be generated. It enables later use of this information to determine the trustworthiness of data and as an input for data quality improvement process.

Key Analysis – IA offers two type of analysis *Primary Key Analysis* (PKA) and *Foreign Key Analysis* (FKA). PKA identifies primary keys, if not defined, and validate already defined keys. It is an important analysis in terms of duplicates detection and uniqueness verification. The second task (FKA) is defined to determine undefined, and validate defined, relationships between tables. The foreign key analysis job builds a complete set of all the column pairs between the primary key columns and the remaining selected columns in the selected tables. The primary key column of one table is paired with all of the columns of the other tables. Next, the system performs a compatibility test on each column pair to determine whether those columns are compatible with each other. If the

Copyright (c) IARIA, 2011 ISBN:978-1-61208-115-1

column pair is compatible, the columns are flagged and then evaluated further. Columns are considered compatible when format, length, scale, and precision matches. After reviewing the results of the job, user can test for referential integrity and determine if a foreign key candidate should be selected as a foreign key.

Cross-Table Analysis – called also cross-domain analysis, is used to determine whether columns contain overlapping or redundant data. It compares the data values between two columns to locate overlapping data. This type of analysis is a multiple step process which contains following steps:

- Selection of two or more columns
- Run a cross-domain analysis job – a list of all of the possible column pairs in data is generated.
- Compatibility test on each column pair to determine whether those columns are compatible (the same test is performed in FKA).

After the compatibility test, cross-domain analysis displays the results from the compatibility test for the user to review and optionally mark a column redundant.

a) QualityStage: IBM InfoSphere QualityStage (QS) complements IA by investigating free-form text fields such as names, addresses, and descriptions. QS allows user to define rules for standardizing free-form text domains which is essential for effective probabilistic matching of potentially duplicate master data records. QS provides user with a set of functionalities containing functions such as free-form text investigation, standardization, address verification and record linkage and matching as well as survivorship that allows best data across different sources to be merged.

b) AuditStage: IBM WebSphere AuditSatge (AS) enables user to apply professional quality control methods to manage different subjective quality factors of information stored in databases such as accuracy, consistency or completeness. By employing technology that integrates Total Quality Management (TQM) principles with data modeling and relational database concepts, AS diagnoses data quality problems and facilitates data quality improvement effort. It allows performing assessment of the completeness, validity of critical data elements and business rule compliance. User can evaluate the quality of data in terms of specific business rules involving multiple data fields within or across records (or rows) that are logically related. In most cases, the type of business rules needed for business rule analysis will not be documented or even explicitly known before the evaluation begins. Therefore, business rules applicable to data will need to be developed, or at least refined, for this analysis [25]. Sources of that knowledge are:

- knowledgeable people (subject matter experts),
- system documentation, and
- occasionally metadata repositories.

AuditStage is very useful tool for assessment of the factor called consistency allowing cross-table rules validation.

IV. Working Example

Consider a typical scenario in an insurance industry. Insurance companies store information about entities including Customer, which can be a person or an organization and Contracts (variety of insurance policies i.e. home, life or car insurance). The company has to keep some information about employees.

MDM Server supports businesses providing predefined data models, containing many of the essential entities for storing this information. One can add additional attributes to entities in this predefined model using the MDM Workbench to generate so called data extensions.

Once the data domain has been defined, the next step is to impose constraints through rule generation. Those rules may belong to one from the following groups:

- Formatting rules – describing different formatting issues like length, allowed characters etc.
- Integrity constraints – rules describing i.e multiplicity of relations
- Business rules – any other rule i.e. dependencies among different fields and values

Table I shows a few possible rules identified for our example. In practice the number of rules generated and stored can be enormous, from 800 rules when assigning a claim, up to 1800 rules applied when underwriting the insurance policy[1].

V. Trust Notion

Trust is an extension of data quality. Data quality is not the only factor influencing the trustworthiness of data and these two concepts are not necessarily correlated. Low-quality data may be considered to have high trust and vice versa. The value of trust strongly depends on the user requirements and usage context. In this section, we discus data quality and introduce the notion of trust.

A. Data Quality

The concept and importance of DQ has been discussed many times in the literature [1]–[14] usually in context of the single data source. However, some research has been also done in the context of integrated data [15]–[20] emphasizing the importance of data quality assurance in this context. In [1] there are three examples of organizational processes where DQ aspects are extremely important.

- Customer matching – it is a common issue in organizations where more than one system with overlapping databases exists. In such case issues with synchronization appear resulting in inconsistent and duplicate information.
- Corporate house-holding – is a problem of identifying members of household (or related group). This context-dependent issue is widely described in [26].
- Organization fusion – is the issue of integration legacy software in case of organizations or units merge.

Many different definitions of DQ can be found in literature. Some of them concentrate on intrinsic values such as accuracy

[1]Based in internal IBM materials provided by ILOG team

TABLE I
RULES IDENTIFIED IN THE SAMPLE DOMAIN

No	Name	Description
	Format	
F.1	Surname length	The length of surname should be at least 2 signs and at most 30 signs
F.2	Name length	The length of name should be at least 2 signs and at most 30 signs
	Integrity Constraints	
I.1	Policy date and birth date of policy holder	The birth date of the policy holder must be earlier than policy start date
I.2	Claim date	Claim may be done only during the coverage period. Claim date must be later than policy start date and earlier than policy end date
	Business Rules	
B.1	Currency	Data should be verified at least once in five years (60 months). The value of currency factor is equal to $1 - \frac{months(current_date - last_verified_date)}{60}$
B.2	Policy Holder min age	The minimal age of policy holder is 18
B.3	Replaced contract id	`Replaces_contract` contains NULL or id of other contract which has been replaced

[8] and completeness [27] and not consider data in a context. Others try to compute values based on some usage context [11]. Instead of single definition, DQ is often split into dimensions or factors – metrics which are more formally defined and can be used measure and compare quality of data sets. But even then the same feature may be called differently by two researchers. This problem has been noticed by Wang and Strong [13] and Foley and Helfert [7].

Naumann [19] attempts to provide operational definition of DQ as "an aggregated value of multiple IQ-criteria" (Information Quality Criteria). IQ-criteria are there classified into four sets:

- Content-related – intrinsic criteria, concerning the retrieved data,
- Technical – criteria measuring aspects determined by software and hardware of the source, the network and the user,
- Intellectual – subjective aspects of data that shall be projected to the data in the source,
- Instantiation-related – criteria concerned on the presentation of the data.

We will follow the Naumann's approach by defining data quality as a aggregated value of multiple DQ-factors. Later we will extend this definition introducing the notion of trust.

B. Trust Definition

Following Naumann's definition of data quality, we define trust (data trust, DT) as *the aggregated value of multiple DT-factors.* This definition provides flexibility when defining trust for a specific industry and user requirements. The trust factor (DT-factor) may be a DQ-factor, as defined earlier in this section, or non-quality (NQ) factor. Here we will concentrate on NQ factors.

1) Data Lineage: Data lineage captures the ratings of data or data sources based on the origin and/or history of processing the data has been through. For example, some sources may be considered as more accurate than others. Information on how much data has been exposed to factors that may have caused errors or inconsistencies (poorly secured systems, systems with poor error handling and checking) is important when considering how to calculate scores giving a measure of trust in the data.

a) Origination: is a factor that captures the scoring of the source of the data. Setting such rating requires the expertise and is strongly context/usage dependent. It may be used in situations where the information about origin is one of the key factors in decision making process.

b) Traceability: is an extension to the origination factor. It assesses the ability to trace the history of data. It gives us the information how much we know (or may know) about the previous places of storage and transformation done over the data element.

c) Stewardship Status: is the scoring capturing the stewardship assigned to the data. It assess if the data is managed manually or in some automatic, more or less limited, way by the system.

2) Data Security: This group of factors covers the security aspects of the systems storing data elements (now and in the past). The values of those factors can be assigned by the expert as well as using some tool that is able to run a security audit task over the system.

a) Authentication: is a scoring telling us how strong authentication mechanisms of the system are. It encloses, but is not necessarily limited to, permissions, password strength, password update policy etc.

b) Authorization: captures the strength of policies regulating the granting of access to the data and tasks in the system.

c) Roles Policy: concerns the aspects of roles management in the system i.e. using primary (root) roles and secondary roles that are limited.

d) Auditing Policy: captures the scores assessing the strength of auditing policies i.e. tracking dates and users initiating tasks. This kind of information may be crucial in the organizations operating on sensitive or confidential data.

3) Trust of Data Sources: The following factors capturing different aspects of data source trust [1]:

a) Believability: describes how true, real and credible a data source is.

b) Reputation: describes how trustable is the source. It is based on the experts' knowledge and is subjective.

c) Objectivity: defines the impartiality of source in data provisioning.

d) Reliability: is a factor describing whether a source provides data conveying the right information.

These definitions are not operational. Moreover they are qualitative and require transformation into quantitative measures to be applicable in our framework.

DT-factors on the row (entity) level are boolean values capturing the rules' satisfaction. On higher levels (table or query) they are expressed as a percentage of tuples satisfying the rule.

VI. Trust Processing

Trust and quality processing described below is one of the most novel aspects of our work. An important advantage of our approach is the use of existing set of tools, slightly modified or extended to serve in new context. We extend those tools by creating data structures to store and process metadata describing data quality and trust. We have implemented mechanisms for assessing some of the quality and trust factors.

A. Trust Data Structures

MDM provides a mechanism that enables an extension of the existing data with trust/quality factors. These extensions may be defined as *persistent object* and stored in the database or be *transient objects* calculated at run time.

MDM allows adding necessary classes and fields to the existing data model. We have used persistent fields as well as transient fields. Defined objects contain fields representing trust factors like `age` and `volatility`. Values of those fields are taken from database or calculated during the transaction's execution for the persistent and transient objects respectively. The acquisition process and the calculation methods are described in the following subsections.

Persistent objects are stored in the database. There are two possible solutions:

- Extension and extended object stored in the same table – the table is then extended by addition of new attributes.
- Extension is stored in a separate table – there is foreign key relation defined between the table storing extended data and table storing the extension.

In both cases while requesting the entity, there will be added information about the extension to the response.

B. Acquisition and Processing of Trust

In section IV, we have identified a set of rules that define quality requirements. Now, we will explain how these rules may be used to provide the information about quality of data.

1) Transient factors: MDM Server provide user with the ability of creating *behavioral extensions*. A behavior extension allows a client to plug in new business rules or functionality to work in conjunction with existing services or functionality within MDM Server. The following rule implements the rule B.1 from the Table I. It assigns the value of attribute `acrcy` according to this the rule B.1.

```
if years(current_date –
   (last_verified_dt of the Person))
   is more than 5
then
   set crncy to 0
else
   set crncy to
   1–(months(current_date –
   (last_verified_dt of the Person))/60)
```

The extension is executed when triggering event occurs, i.e. we can define the extension triggered by a select event done over `Person`. Before user receives `Person`, the system calls our extension and calculates transient trust factors accordingly, based on values stored in database. Priority enable us to define the order of calls.

MDM allows us to define different triggers for behavioral extensions:

- Action – Specifies component level transaction name. e.g. 'getPerson' or 'updateStudent; each transaction is associated with a particular Module within MDM Server
- Transaction – Applies extension to a specific transaction at the controller level
- Action category – Specifies at the component level what category of transactions are to be impacted by the extension e.g. Add, Update, View, All
- Transaction category – Determines whether the extension will apply to a category of transactions e.g. inquiry, persistence or all at the controller level

The first two call an extension triggered by a specific action on chosen entity (i.e. `updatePerson`) on component or controller level respectively. Action and transaction category, on other hand allows the user to define the extension triggered by specific class of action or transaction, that is, to add or update done over any entity. In addition, extensions may be called before or after the action or transaction initiated by user.

2) Persistent factors: Values of scores for persistent factors are stored in the database. Data structures required to store this information has been defined as a *data extension* in the MDM Server. The acquisition of persistent factors can be off-line or on-line process. We use IA and AS to acquire trust scores. Those tools are used to acquire scorings in off-line mode. MDM Server allows us to modify persistent factors in the on-line mode. In such case we have to define the behavioral extension that is triggered by update or insert event. Figure 1 depicts dependencies among functions in Information Analyzer. Basically it defines order in which chosen functions may be called.

Copyright (c) IARIA, 2011 ISBN:978-1-61208-115-1

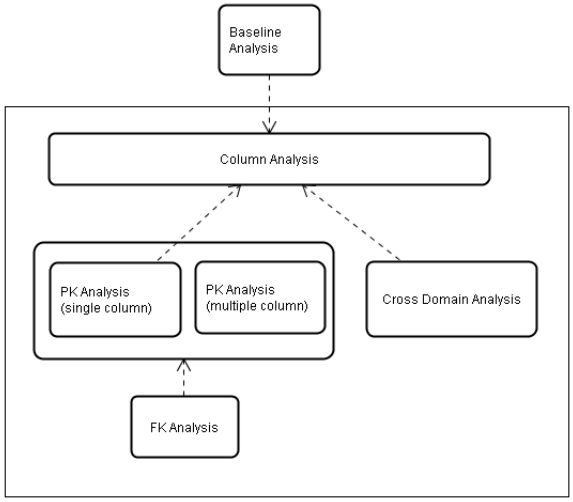

Fig. 1. IBM WebSphere Information Analyzer function dependencies

- Baseline Analysis requires at least column analysis to be performed before it can be run.
- Column Analysis must be run before a single column Primary Key Analysis can be performed.
- A multi-column Primary Key Analysis can be performed independently of any of the other analyses. It invokes Column Analysis automatically under the covers if a Column Analysis has not been performed on the selected columns.
- A Foreign Key Analysis (single or multi-column) can only be performed after a Primary Key Analysis (single or multi-column) is performed.
- Cross Domain Analysis requires Column Analysis to have been run.

Column analysis job evaluates data characteristics by analyzing a frequency distribution of the data values in each column. It is performed to evaluate the characteristics such as: minimum, maximum and average field length, precision, data types, cardinality, nullability and empty values. This task gives the structural view of data and returns inferences (best choices that system could make) in terms of field's length, data class, and uniqueness of values or constants that may indicate unused columns.

While IA is a tool to measure basic aspects of data quality (formatting, data types, precision, etc.), AuditStage (AS) can be used to implement more complex rules i.e. business rules. Results of the execution of such predefined rules are stored in the database and used to determine the overall quality of data.

IA as well as AS, are both run as a scheduled batch jobs. The frequency of such operation depends on user requirements and domain. It is obvious that long time interval between two executions can jeopardize the reliability of results, however, it is expensive operation. One needs to make a tradeoff to minimize costs and maximize the reliability of the results.

Another rich source of information about DQ are results stored by DataStage. This tool, used in general to perform transformations of data and transitions from source to target data source, produces statistics, as a side effect. This information about merged records or unmatched records, gives us very important input for trust computation. This information, processed by MDM's behavioral extensions provides user with the information about consistency among data sources and can also points to underlying problems with data, such as inconsistent data coming from two systems caused by i.e. incorrect matching.

C. Trust processing

The trust alone is just yet another piece of data given to the user. The really important question is *What can be done with this information?* Let's consider now some cases showing usage of the trust in the system.

We have shown that the trust score can be incorporated in our meta-data and linked with each field in the database if desired. This information can be then returned to the user. Even though this information is very detailed, it is not practically useful in all cases. Without algorithms to propagate trust in the query processing, we can only annotate a tuple and return it to the user. However, we can build some statistics over this information that can be used later.

One of the problems that are currently unsolved is propagation of trust scores in the query processing. We are currently working on methods allowing us to estimate the trust of the result of the SQL operator based on the estimated trust of entry set. We are using estimates in this context because it is significantly less expensive than reaching out each time for the data.

There are many interesting problems in this domain. One of them is the impact of the trust of the key attributes on the trust of the result. This issue originates from the observation made by Motro and Rakov [28] that the measure can be considered accurate only if the key of the tuple is accurate. For example, when we consider the *group by* operation, there is significant influence of the group by keys on the trust to the aggregation result. It is intuitive especially in the context of accuracy dimension: simply if the group by key is highly inaccurate, then division into groups cannot be trusted. That leads to low level of trustworthiness of the aggregation result, even if the accuracy of the measure itself is high. Similar problems arise for join operation. However, in this case the accuracy of the keys of the join has to be propagated through the whole tuple, because inaccurate value of one of join components implies that the derived tuple should not be in the result set.

VII. Conclusion and Future Work

Measuring data quality and data trust is one of the key aspects of supporting businesses in decision making process or data stewardship. Master Data Management in other hand supports sharing data within and across lines of business. In such case trustworthiness of the shared data is extremely important. Our investigation has resulted in consistent method of gathering and processing quality and trust factors.

Copyright (c) IARIA, 2011 ISBN:978-1-61208-115-1

In this work we have presented the *IBM InfoSphere MDM Server* and elements of *IBM Information Server* such as DataStage, QualityStage, AuditStage and Information Analyzer, and their ability to handle data quality and data trust. We have also presented the new notion of data trust. The process of gathering and computing data quality and trust factors has been described and explained using example.

At this point we would like to point to some aspects of MDM and DQ that have been mentioned in this work, but have not been covered in detailed. These aspects play important role in quality and trust computation. An extremely important feature in terms of defining business rules or any other rule and reusing them across cooperating systems is common rule repository and common rule engine. Those two elements can allow users to reuse defined rules and minimize probability of inconsistencies across systems. Such approach will also minimize costs because it eliminates a need to redefine rules in each system. Another aspect of trust and data governance not covered by this paper is temporal aspect of trust. In many cases trust strictly depends on time and a value i.e. address can be considered trustworthy only within a given time interval. This paper does not cover those aspects of quality and trust computation but we would like to point that there is ongoing work in IBM to solve this problem.

VIII. ACKNOWLEDGMENTS

We would like to thank Guenter Sauter for valuable discussions and uncovering new aspects of trust.

REFERENCES

[1] C. Batini and M. Scannapieco, *Data Quality: Concepts, Methodologies and Techniques*, ser. Data-Centric Systems and Applications. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[2] P. B. Crosby, *Quality is free : the art of making quality certain / Philip B. Crosby*. McGraw-Hill, New York :, 1979.

[3] L. English, "Information quality improvement: Principles, methods, and management," Seminar, INFORMATION IMPACT International, Inc., 1996, 5th Ed., Brentwood, TN: INFORMATION IMPACT International, Inc.

[4] D. Ballou and H. Pazer, "Modeling data and process quality in multi-input, multi-output information systems," *Management Science*, vol. 31, no. 2, pp. 150–162, 1985.

[5] D. Ballou, R. Wang, H. Pazer, and G. K. Tayi, "Modeling information manufacturing systems to determine information product quality," *Manage. Sci.*, vol. 44, no. 4, pp. 462–484, 1998.

[6] A. Parssian, S. Sarkar, and V. S. Jacob, "Assessing information quality for the composite relational operation join," in *IQ*, 2002, pp. 225–237.

[7] O. Foley and M. Helfert, "The development of an objective metric for the accessibility dimension of data quality," in *Proceedings of International Conference on Innovations in Information Technology*. Dublin: IEEE, 2007, pp. 11–15.

[8] J. Olson, *Data Quality: The Accuracy Dimension*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002.

[9] A. Parssian, S. Sarkar, and V. S. Jacob, "Assessing data quality for information products: Impact of selection, projection, and cartesian product," *Manage. Sci.*, vol. 50, no. 7, pp. 967–982, 2004.

[10] T. C. Redman, *Data quality : the field guide*. Boston: Digital Pr. [u.a.], 2001.

[11] G. K. Tayi and D. P. Ballou, "Examining data quality," *Commun. ACM*, vol. 41, no. 2, pp. 54–57, 1998.

[12] A. R. Tupek, "Definition of data quality," U.S Department of Commerce, Census Bureau Methodology & Standards Council, 2006.

[13] R. Y. Wang and D. M. Strong, "Beyond accuracy: what data quality means to data consumers," *J. Manage. Inf. Syst.*, vol. 12, no. 4, pp. 5–33, 1996.

[14] L. P. Yang, Y. W. Lee, and R. Y. Wang, "Data quality assessment," *Communications of the ACM*, vol. 45, pp. 211–218, 2002.

[15] Y. Cui, J. Widom, and J. L. Wiener, "Tracing the lineage of view data in a warehousing environment," *ACM Trans. Database Syst.*, vol. 25, no. 2, pp. 179–227, 2000.

[16] M. Bouzeghoub and Z. Kedad, *Quality in Data Warehousing*. Kluwer Academic Publisher, 2002.

[17] A. Gupta and J. Widom, "Local verification of global integrity constraints in distributed databases," in *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 26-28, 1993*, P. Buneman and S. Jajodia, Eds. ACM Press, 1993, pp. 49–58.

[18] M. Gertz and I. Schmitt, "Data Integration Techniques based on Data Quality Aspects," in *Proceedings 3. Workshop "Föderierte Datenbanken", Magdeburg, 10./11. Dezember 1998*, I. Schmitt, C. Türker, E. Hildebrandt, and M. Höding, Eds. Aachen: Shaker Verlag, 1998, pp. 1–19. [Online]. Available: citeseer.ist.psu.edu/19916.html

[19] F. Naumann, *Quality-driven query answering for integrated information systems*. New York, NY, USA: Springer-Verlag New York, Inc., 2002.

[20] M. P. Reddy and R. Y. Wang, "Estimating data accuracy in a federated database environment," in *CISMOD*, 1995, pp. 115–134.

[21] A. Dreibelbis, E. Hechler, B. Mathews, M. Oberhofer, and G. Sauter, "Master data management architecture patterns," http://www.ibm.com/developerworks/data/ library/techarticle/dm-0703sauter/index.html, 2007.

[22] W. Fan, "Dependencies revisited for improving data quality," in *PODS '08: Proceedings of the twenty-seventh ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. New York, NY, USA: ACM, 2008, pp. 159–170.

[23] J. Radcliffe and A. White, "Key issues for master data management," Gartner Master Data Management Summit, Chicago, IL, 2008.

[24] IBM, "Ibm master data management: Effective data governance," ftp://ftp.software.ibm.com/software/uk/itsolutions/information-management/information-transformation/ master-data-management/master-data-management-governance.pdf, 2007.

[25] N. Alur, R. Joseph, H. Mehta, J. T. Nielsen, and D. Vasconcelos, *IBM WebSphere Information Analyzer and Data Quality Assessment*, ser. Redbooks. International Business Machines Corporation, 2007.

[26] R. Y. Wang, K. Chettayar, F. Dravis, J. Funk, R. Katz-Haas, C. Lee, Y. Lee, X. Xian, and S. Bhansali, "Exemplifying business oppurtunities for improving data quality from corporate household research." in *Advances in Management Information Systems - Information Quality (AMIS-IQ) Monograph*, April 2005.

[27] Y. W. Lee, L. Pipino, D. M. Strong, and R. Y. Wang, "Process-embedded data integrity," *J. Database Manag.*, vol. 15, no. 1, pp. 87–103, 2004.

[28] A. Motro and I. Rakov, "Not all answers are equally good: estimating the quality of database answers," pp. 1–21, 1997.

Copyright (c) IARIA, 2011 ISBN:978-1-61208-115-1