A Rule Based Taxonomy of Dirty Data

Lin Li, Taoxin Peng, Jessie Kennedy
School of Computing
Edinburgh Napier University
Edinburgh, UK
{l.li, t.peng, j.kennedy}@napier.ac.uk

***Abstract*—There is a growing awareness that high quality of data is a key to today's business success and that dirty data existing within data sources is one of the causes of poor data quality. To ensure high quality data, enterprises need to have a process, methodologies and resources to monitor, analyze and maintain the quality of data. Nevertheless, research shows that many enterprises do not pay adequate attention to the existence of dirty data and have not applied useful methodologies to ensure high quality data for their applications. One of the reasons is a lack of appreciation of the types and extent of dirty data. In practice, detecting and cleaning all the dirty data that exists in all data sources is quite expensive and unrealistic. The cost of cleaning dirty data needs to be considered for most of enterprises. This problem has not attracted enough attention from researchers. In this paper, a rule-based taxonomy of dirty data is developed. The proposed taxonomy not only provides a mechanism to deal with this problem but also includes more dirty data types than any of existing such taxonomies.**

Keywords- Dirty data; data quality; data cleaning;

I. INTRODUCTION

Today, data has become more and more important, with many human activities relying on it. As data have kept increasing at an explosive rate, a great number of database applications have been developed in order to derive useful information from these large quantities of data, such as decision support systems and customer relationship management systems (CRM). It has now been recognized that an inordinate proportion of data in most data sources is dirty [1]. Due to the 'garbage in, garbage out' principle, dirty data will distort information obtained from it [2]. Obviously, a data warehouse with a high proportion of dirty data is not reliable for the purpose of data mining or deriving business intelligence and the quality of decisions made on the basis of such business intelligence is also not convincing. Therefore, before using these database applications, dirty data needs to be cleaned. Nevertheless, research shows that many enterprises do not pay adequate attention to the existence of dirty data and have not applied useful methodologies to ensure high quality data for their applications. One of the reasons is a lack of appreciation of the types and extent of dirty data [3]. Therefore, in order to improve the data quality, it is necessary to understand the wide variety of dirty data that may exist in the data sources as well as how to deal with them.

From the literature, some work has been undertaken exclusively to identify problems (dirty data types) that affect data quality and has resulted in taxonomies of dirty data. For example, Kim *et al* [1] and Oliveira *et al* [4] have proposed two different taxonomies of dirty data and have presented 33 and 35 dirty data types respectively. Some work, although not undertaken exclusively for the purpose of generating a taxonomy of dirty data, has highlighted the problems arising due to poor data quality and groups of dirty data types have been proposed . For example, according to the constraints of Müller and Freytag's pre-defined data model[5], data resulting from data collection that does not conform to the constraints of the data model are considered to be data anomalies. They roughly classify data anomalies into three different sets, namely syntactical anomalies, semantic anomalies and coverage anomalies and together 8 dirty data types are identified. Rahm and Do [6] distinguish the observed data quality problems into two sets, namely single-source problems and multi-source problems. Within each set, data quality problems have been classified into schema-level problems and instance-level problems respectively. These problems reflect the different dirty data types that could be captured according to different levels and 19 problems have been introduced in their work. Compared with Müller and Freytag's and Rahm and Do's work, the two taxonomies of dirty data are based on a much more complete set of dirty data types.

Data cleaning is a labour-intensive, time-consuming and expensive process. In practice, cleaning all dirty data types introduced by the two taxonomies mentioned above is unrealistic and simply not cost-effective when taking into account the needs of a business enterprise. For example, a company might only be able to afford to clean a specific group of types of dirty data to satisfy some specific needs. The problem then becomes how the business can make a selection according to their different business needs. In this paper, this problem is referred to as the Dirty Data Selection (DDS) problem. Although there are several taxonomies of dirty data existing in the literature, none of them are designed for this purpose. For example, in Oliveira *et al*'s taxonomy of data quality problems, 35 dirty data types have been introduced, which is considered as the most comprehensive taxonomy so far in the literature. In this case, by only showing these 35 dirty data types, it is difficult to tell which possible dirty data types should be selected to deal with for different data sets. Thus a

DOI: 10.5176_2010-2283_1.2.52

motivation of this work is to develop a taxonomy that can help in solving data quality problems, such as the DDS problem. This paper presents a rule-based taxonomy of dirty data. The taxonomy presented not only covers a larger range of dirty data types than any of the existing taxonomies but can also help in dealing with the DDS problem when specific business needs are considered.

The remainder of this paper is organized as follows: In section II, a rule-based taxonomy of dirty data is presented in detail (organized according to four different categories). In section III, a method of dealing with dirty data is presented. An example of using the method is given in section IV. In section V, existing related works are compared and discussed. Finally, the paper is concluded and future work is discussed in section VI.

II. A Rule-Based Taxonomy of Dirty Data

This section presents a new taxonomy, based on data quality rules. According to David Loshin [7], by relating business impacts to data quality rules, an organization can employ the data quality rules for measuring the business expectations and the improvement of data quality can be viewed as a function of conformance to business expectations. By integrating control processes based on data quality rules, business users are able to determine how best the data can be used to meet their own business needs. Thus, data quality rules play an important role in the improvement of data quality for a business.

In this paper, dirty data is defined as the data flaws that break any of data quality rules. Therefore, when data quality rules are obtained, data can be assessed as to whether or not the data is dirty according to the description of these rules. The proposed taxonomy is built based on the data quality rules. Before presenting the proposed taxonomy, data quality rules are first introduced.

A. Data Quality Rules

Adelman *et al* proposed a set of data quality rules which, according to the authors, have been categorized into four groups namely: business entity rules; business attribute rules; data dependency rules; and data validity rules. Business entity rules specify rules about business objects or business entities. Business attribute rules are rules about data elements or business attributes. Data dependency rules specify different types of dependencies between business entities or business attributes. Data validity rules govern the quality of data values [8]. We use these four groups of data quality rules to classify dirty data types into four different categories. According to Adelman *et al* , the four groups of data quality rules are further divided into a list of sub-rules from which a tree structure classification of data quality rules is obtained. By analyzing data quality rules on the leaf nodes, we have identified the dirty data types in each category. Table I lists the entire data quality rules based on the four different categories.

TABLE I. Data quality rules

Rule Category	Data Quality Rule
1.Business entity rules	R1.1 Entity uniqueness rules
	R1.2 Entity cardinality rules
	R1.3 Entity optionality rules
2.Business attribute rules	R2.1 Data inheritance rules
	R2.2 Data domains rules
3.Data dependency rules	R3.1 Entity-relationship rules
	R3.2 Attribute dependency rules
4.Data validity rules	R4.1 Data completeness rules
	R4.2 Data correctness rules
	R4.3 Data accuracy rules
	R4.4 Data precision rules
	R4.5 Data uniqueness rules
	R4.6 Data consistency rules

B. Dirty Data Types

Four groups of dirty data types are obtained according to the four different rule categories from table I. Each group of dirty data types is detailed below.

1) Business entity rules related dirty data types: Business entity rules about business entities which are subject to three data quality rules namely entity uniqueness rules, entity cardinality rules and entity optionality rules [8]. Within this group, the following dirty data types are identified:

- Cardinality relationship problem: As an example of this problem, the number of employees by counting the number of tuples from the Employee table, is not the same as the number of employees by summing the number of employees in each department in the Department table.
- Recursive relationship problem: As an example of this problem, suppose in a department of a university, one person may supervise many other persons and each supervised person may have many supervisors at the same time. Such information is recorded in the table *people (ID*, name, supervise)*. Suppose the information 'Jack is supervising Rose and Rose is supervising Jack' is found in the table. Clearly, this is not going to happen in the real world.
- Optionality relationship problem: the entity optionality rule identifies the minimum number of times two business entities can be related. For example, an online store requires that when a customer has purchased a product on line, the customer's delivery information must be in the delivery table. Otherwise, a missing tuple from the delivery table will cause a problem such as an undelivered item.

- Reference defined but not found: When a relationship is instantiated through a foreign key, the referenced instance of the entity must exist in the related table.

2) Business attribute rules related dirty data types: Business attribute rules specify rules about business attributes or data elements, which are subject to two data quality rules namely data inheritance rules and data domains rules. As data inheritance rules are object oriented related rules, we do not consider this rule in our work. Therefore, in this group, the following dirty data types are identified:

- Set violation: For an enumerated data type, its value should be within the allowable value set. For example, suppose the allowable data value set for "city" attribute is {London, Edinburgh, Manchester, Birmingham}, then the value of "New York" is not allowable.
- Data value out of value range: As an example of this problem, suppose the age of human being in a database is defined as "18<=age<30". It is not allowed that an age value of '10' or '35' is entered in the table.
- Data value constraint violation: When some constraints are used to regulate data values, the data value should conform to those constraints. A constraint may be used to regulate a single piece of data or multiple data values. For example, a medical experiment requires the age of the people who participate should below 30 (inclusive). Then the constraint for "age" attribute is "age<=30". If data has been found that its age value is "35", then such data is not expected in the table.
- Use of wrong data type: When the value of an attribute such as "Name" is set to be a string data type, it is not expected that a numeric value be found for the "Name" attribute.
- Syntax violation: Syntax violation happens when data value does not conform to the defined pattern or format for its attribute. For example, when the format of "Date" attribute is defined as the pattern of "DD/MM/YYYY", then the value of "2010-03-05" is not expected. The correct value should be "05/03/2010".

3) Data dependency rules realted dirt data types: Data dependency rules apply to data relationships between two or more business entities or business attributes. The dirty data types identified in this group are:

- Data relationship constraint violation: As an example of this problem, an employee who has been assigned a project is not allowed to enroll in a training program, i.e., this employee's data is not supposed to be found in the training table.
- Contradiction data: The existence of an attribute value is determined or constrained by the value of another attribute. For example, suppose it is defined that when the status of a loan is "funded", then the value of loan amount must be greater than zero.
- Wrong derived field data: This problem occurs when a data value is derived from two or more other attribute values. For example, a miscalculation of an employee's income by miscomputing the tax will result in a wrong derived field data.
- Wrong data among related attributes: This problem occurs when the value of one attribute is constrained by the value of one or more attributes in the same business entity or in a different but related business entity. For example, the value of annual expenses in a department is constrained by the sum of all distinct expenses in that department.

4) Data validity rules related dirty data types: Data validity rules govern the quality of data values, there are six data validity rules (Rule 4.1~ Rule 4.6, see table I). The dirty data types identified by the six validity rules are:

- Missing tuple: Entity completeness requires that all instances exist for all business entities, i.e., all records are present in the table.
- Missing value: It is required that all attributes for a business entity contains all allowable values. It should be clear that Null value is different from missing value. When a constraint of "null-value allowed" is enforced on the data set, null value indicates "value unknown or nonexistent". A missing value simply indicates whether a value should exist for the attribute or not.
- Meaningless data value. The data value for an attribute must be correct and reflect the attribute's intended meaning. When the data value is beyond the context of the attribute, the data value is a meaningless data value. For example, the value for the attribute "address" is defined as a set of allowable characters which reflect a person's address in the real world. If "£$%S134" is entered, it does not make any sense to a valid address data.
- Extraneous data entry: An example of extraneous data entry is the entry of address and name in the name field.
- Lack of data elements: An example of this problem is when a part of post code misses from attribute "PostCode" , i.e., "5DT" missing from "EH10 5DT" .
- Erroneous entry: An example of erroneous entry is when a student's age is entered as "26" rather than the student's real age "27".
- Entry into wrong field: This problem occurs for example when the value a person's name is entered into its address field.
- Identity rule violation: As an example of this problem, suppose in table *employee (Emp_No., Name, Emp_NIN, DoB)*, Emp_No. is defined as the primary key. According to the values of Emp_No. from employee table, the uniqueness of Emp_No. is

guaranteed. But it does not mean that each employee is properly identified in the data. For example, a person may have two records with two distinct Emp_No. but identical values for national insurance number (NIN). Suppose it is required that each person can only has one unique Emp_No in the table. Obviously, they are duplicate records referring to the same person.

- Wrong reference: This is the case when a reference is defined but its value is wrong which breaks the attribute's dependency rules.
- Outdated value: It is required that the data value must be accurate in terms of its state in the real world. If not, its value is said to be an outdated value because it does not represent its real state in the real world.
- Imprecision data: It is required that all data values for a business attribute must be as precise as required by the attribute's business requirements. As an example of imprecision data, suppose an analysis of the financial position of an auditor requires the value of the data has precision to the pence, if the value is based on the unit of pounds, then the data is an imprecision data.
- Ambiguous data: The use of abbreviation of data for instance, sometimes may cause an ambiguous meaning which is not as precise as required by the attribute's intended meaning. For example, when an abbreviation word "MS" is used to represent a company's name, it is difficult to tell whether it stands for "Morgan Stanley" (a global financial services firm) or "Microsoft" (a global software company) when both of the companies have been recorded in the same data source.
- Misspelling: A misspelling problem, for example, when "John Smith" is entered as "Jonh Smyth".
- Duplicate record in single/multi data source: Rule 4.5 specifies that each business entity instance must be unique. Duplicate records may happen for example, when a person's name and address are represented in different ways, the same entity may be represented more than once in the same or different data sources.
- Inconsistent record in single/multi data source: Rule 4.6 specifies the data value should be consistent. Inconsistent data can be found in both single and multi data sources. For example, in different data sources, the data vale of the same person's address may be recorded differently. Suppose this person has only one valid address, these records are inconsistent records.
- Different representations for the same data: in addition to inconsistent record, data conflicts may arise when multiple data sources are integrated. Usually, different data sources are typically developed and maintained independently to serve specific needs. When these data sources are integrated, due to the different representations for the same data, problems are observed. Specifically, these differences may be due to the different use of abbreviations, special characters, word sequence, measurement unit, encoding format, aggregation levels and alia names.

According to the descriptions of the data validity rules, some schema-level problems can also be identified. For example, one of the data completeness rules requires that all business attributes for each business entity exist. In this case for example, if an employee's address is represented in a different number of fields in different data sources and they are each correct in their own data source, when they come to be integrated, problems will occur. In data uniqueness rules, two of them are related with the definition of attributes (homonyms and synonyms) which are also related to schema-levels problems. In this paper, we do not consider schema-level problems in our taxonomy.

With the above dirty data types analyzed based on the data quality rules, table II lists these dirty data types, each of which has been assigned with a type number (DT.1 ~ DT.38).

TABLE II. DIRTY DATA TYPES

No.	Dirty Data Type
DT.1	Cardinality relationship problem
DT.2	Recursive relationship problem
DT.3	Optionality relationship problem
DT.4	Reference defined but not found
DT.5	Set violation
DT.6	Data value out of value range
DT.7	Data value constraint violation
DT.8	Use of wrong data type
DT.9	Syntax violation
DT.10	Data relationship constraint violation
DT.11	Contradiction data
DT.12	Wrong derived field data
DT.13	Wrong data among the related attribute
DT.14	Missing tuple
DT.15	Missing value
DT.16	Meaningless data value
DT.17	Extraneous data entry
DT.18	Lack of data elements
DT.19	Erroneous entry
DT.20	Entry into wrong field
DT.21	Identity rule violation
DT.22	Wrong reference
DT.23	Outdated value
DT.24	Outdated reference
DT.25	Imprecision data
DT.26	Ambiguous data

No.	Dirty Data Type
DT.27	Misspelling
DT.28	Duplicate record in single data source
DT.29	Duplicate record in multi data source
DT.30	Inconsistent record in single data source
DT.31	Inconsistent record in multi data source
DT.32	Different representations due to abbreviation
DT.33	Different representations due to special characters
DT.34	Different representations due to word sequence
DT.35	Different representations due to measurement unit
DT.36	Different representations due to encoding format
DT.37	Different representations due to aggregation level
DT.38	Different representations due to use of alia name

C. The Taxonomy

In Table I, data quality rules have been organized in a tree structure. The proposed taxonomy follows the same structure and classifies the dirty data according to the four different categories of data quality rules. As the dirty data types obtained in section II.B are based on analyzing the rules on the leaf nodes, the four categories of dirty data have been further classified into distinct dirty data types according to the corresponding rules on the leaf nodes. Table III shows the proposed taxonomy.

TABLE III. RULE-BASED TAXONOMY OF DIRTY DATA

Dirty Data Category	Data Quality Rules	Dirty Data Type
Business entity rules related dirty data	R1.2 Entity cardinality rules	DT.1, DT.2
	R1.3 Entity optionality rules	DT.3, DT.4
Business attribute rules related dirty data	R2.2 Data domain rules	DT.5~DT.9
Data dependency rules related dirty data	R3.1 Entity relationship dependency rules	DT.10
	R3.2 Attribute dependency rules	DT.11~DT.13
Data validity rules related dirty data	R4.1 Data completeness rules	DT.14, DT.15
	R4.2 Data correctness rules	DT.16~DT.20
	R4.3 Data accuracy rules	DT.21~DT.24
	R4.4 Data precision rules	DT.25~DT.27
	R4.5 Data uniqueness rules	DT.28, DT.29
	R4.6 Data consistency rules	DT.30~DT.38

In this taxonomy, 38 different dirty data types have been identified under different data quality rules, which forms an even larger collection of dirty data compared with any of the existing taxonomies or classifications[1][3][4][5]. This will be further discussed in section V.

Considering the DDS problem described in Section I, when specific needs of a business enterprise is taken into account, it is unrealisitic and not cost-effective to clean all of the dirty data types. As business rules can be used as guidelines for the validation of information quality, with the help of the proposed rule-based taxonomy, it is reasonable for a business enterprise to pick up a few most important groups of business rules rather than all of rules to deal with, according to its own business priorities. Therefore, the DDS problem is solved. A method for this purpose is detailed in next section.

III. A METHOD

According to David Loshin, 'integrating control processes based on data quality rules communicates knowledge about the value of the data in use, and empowers the business users with the ability to determine how best the data can be used to meet their own business needs'. It also recommended that 'organizing data quality rules within defined data quality dimensions can enable the governance of data quality management and data stewards can use data quality tools for determining minimum thresholds for meeting business expectations, monitoring whether measured levels of quality meet or exceed those business expectations' [7]. The proposed taxonomy of dirty data is a data quality rule based taxonomy which forms relationships between dirty data types and data quality rules. When these data quality rules are organized under the defined data quality dimensions, a relationship between data quality dimensions and dirty data types can also be formed, which will be used to develop a method to deal with data quality problems. This method begins with a mapping between business rules and data quality dimensions.

A. Mapping between Data Quality Rules and Data Quality Dimensions

Amongst the research work regarding data quality dimensions [9][10][11], the following four data quality dimensions: accuracy, completeness, consistency and currentness have been considered to be the dimensions of data quality involving data values [12]. From table III, it can be seen that data uniqueness rules are associated with the data validity category. R4.5 evaluates a special data quality problem which is caused by duplicate records. Because of the popularity, complexity and difficulty of this problem, it has attracted a large number of researchers [13]. Therefore, apart from the four data quality dimensions, an extra data quality dimension "Uniqueness" is introduced for dealing with duplicate records exclusively in the proposed method.

Brief introductions of these five dimensions are given below:

- Accuracy dimension: The accuracy of the datum refers to the degree of closeness of its value *v* to some value *v'* in the attribute domain considered correct for the entity *e* and attribute *a*. If the datum's value *v* is the same as a correct value *v'*, the datum is said to be accurate or correct.

- Completeness dimension: Completeness is the degree to which a data collection has values for all attributes of all entities that are supposed to have values.
- Currentness dimension: A datum is said to be current or up to date at time *t* if it is correct at time *t*. A datum is out of date at time *t* if it is incorrect at *t* but was correct at some moment preceding t.
- Consistency dimension: Data is said to be consistent with respect to a set of data model constraints if it satisfies all the constraints in the set.
- Uniqueness dimension: Uniqueness of the entities within a data set implies that no entity exists more than once within the data set.

With the five data quality dimensions, a new classification of the dirty data types is introduced beginning with a mapping of data quality rules with data quality dimensions. Table IV shows the result of the mapping:

TABLE IV. DATA QUALITY DIMENSIONS AND DATA QUALITY RULES

Data Quality Dimension	Data Quality Rules
Accuracy dimension	R2.2, R3.2, R4.2, R4.4
Completeness dimension	R1.3, R4.1
Currentness dimension	R4.3
Consistency dimension	R1.2, R3.1, R4.6
Uniqueness dimension	R4.5

B. A Classification

The result of Table IV provides immediate help for the proposed classification of dirty data within the new taxonomy. Combining the result from table III and IV, the classification of dirty data based on data quality dimensions is achieved in table V.

TABLE V. DATA QUALITY DIMENSIONS AND DIRTY DATA TYPES

Data Quality Dimension	Dirty Data Type
Accuracy dimension	DT.5~DT.9, DT.11~DT.13, DT.16~DT.20, DT.25~DT.27
Completeness dimension	DT.3, DT.4, DT.14, DT.15
Currentness dimension	DT.21~DT.24
Consistency dimension	DT.1, DT.2, DT.10, DT.30~DT.38
Uniqueness dimension	DT.28, DT.29

Therefore, the task of data cleaning can be considered as cleaning dirty data by different data quality dimensions. The DDS problem described in Section I can therefore be solved by forming a relationship between the defined data quality dimensions and dirty data types with the help of the rule-based taxonomy of dirty data.

C. The Method

By utilizing the classification in III.B, a method of dealing with dirty data is described below:

a) Create an order of the five dimensions according to the business priority policy.

b) Identify data quality problems.

c) Map the data types identified in b) into the dimensions against the classification table.

d) Decide dimensions to be selected based on the budget.

e) Select appropriate algorithms, which can be used to detect dirty data types associated with dimensions identified in c).

f) Execute the algorithms.

IV. AN EXAMPLE

With the proposed method in section III.C, data cleaning tasks are considered as improving the data quality represented by the defined data quality dimensions according to the needs of a business. An example in this section shows how the proposed method can help enterprises in dealing with DDS problem.

Let's consider an information system used by a telecommunication company. Such an information system brings efficiencies in the day-to-day telecom businesses operations such as : line maintenance, line installation, billing, cash collection etc and has helped to increase level of customer satisfaction by providing them better service. Examples of customer services provided by the system are that customers are able to down load their monthly statements and make enquiries about telephone numbers of other customers on the internet. Customers need to get correct the correct information when using these services. Therefore, the company needs to make sure that data maintained in the system is accurate enough and up to date to provide correct information for their customers.

However, dirty data might exist in the system, such as misspelt data (DT.27), duplicate records (DT.28, DT.29), data entered into a wrong field (DT.20), out of range data value (DT.6), missing data within a record (DT.15), late updated data (DT.23, DT.24), Different representation of customer name and address (DT.33, DT.34) etc. These dirty data may cause mistakes to happen which angers customers. For example, no customer wishes to pay the bill for someone else every month which might occur due to the problem of late update of data regarding change of address.

Therefore, in order to bring a better customer service to maintain customer loyalty, this telecommunication company choose the five data quality dimensions mentioned in section III.A to regulate the quality of the data, i.e., accuracy dimension, completeness dimension, consistency dimension, currentness dimension and uniqueness dimension.

With the help of the rule-based taxonomy of dirty data, a classification of the identified data quality problems based on

the five data quality dimension is obtained and is shown in table VI.

TABLE VI. DATA QUALITY DIMENSIONS AND DIRTY DATA TYPES

Data Quality Dimension	Dirty Data Type
Accuracy dimension	DT.6, DT.20, DT.27
Completeness dimension	DT.15
Currentness dimension	DT.23, DT.24
Consistency dimension	DT.33, DT.34
Uniqueness dimension	DT.28, DT.29

As mentioned at the beginning of this paper, the cost of data cleaning has to be taken into account when applying data cleaning applications to large, comprehensive businesses. In this example, with the help of the rule based taxonomy of dirty data, the identified data quality problems have been organized under all the defined data quality dimensions. Suppose cleaning all of the dirty data for this telecommunication company is unrealistic. Therefore, the problem that the company has to face is how to select a group of types of dirty data to deal with, which is actually a DDS problem. According to the company's priority policy, the time constraints are often very stringent for web available data. For example, customers need to have correct information for their online statements. Therefore, the currentness dimension and accuracy dimension are much more urgent than others. The order of the five data quality dimensions for this organisation is therefore: Currentness, Accuracy, Consistency, Uniqueness and Completeness in descending by priority. The proposed method provides a systematic approach to cope with the problem. It is easy to select which of these dirty data types cause accuracy and currentness related problems. In this case, dirty data that causes problems in the currentness and accuracy dimensions need to be dealt with first. Therefore, those data cleaning algorithms or methods designed for dirty data types DT.6, DT.20, DT.23, DT.24, DT.27 should be firstly applied to the system. With the other existing taxonomies, which only show a list of dirty data types, it is difficult to tell which group of dirty data should be considered first and it will be too expensive for the system to run all algorithms for all the possible dirty data candidates. However with the help of rule based taxonomy of dirty data and the priority of quality dimensions, this problem is solved by only selecting dirty data types in the currentness and accuracy dimensions.

The solution to this problem can be described as:

a) The order of dimensions: Currentness, Accuracy, Consistency, Uniqueness and Completeness.

b) Data quality problems: DT.6, DT.15, DT.20, DT.23, DT.24, DT.27, DT.28, DT.29, DT.33, DT.34.

c) Mapping: see table VI.

d) Dimensions to be dealt with: see the table below.

e) Select appropriate algorithms from table VII.

TABLE VII. AN EXAMPLE

Data Quality Dimension	Dirty Data Type	Solutions/Algorithms
Currentness	DT.23,DT.24	Concurrency control algorithm, general purpose algorithm, AIRSTD approach …
Accuracy	DT.6, DT.20, DT.27	Active learning algorithm, Statistical-model based outlier detection algorithm, Trigram analysis technique …

f) Execute the selected algorithms.

V. RELATED WORK

Müller and Freytag [5] identify a set of errors (anomalies) that will affect data quality. The set includes lexical error, domain format error, irregularities, constraint violation, missing value, missing tuple, duplicates and invalid tuple. Müller and Freytag's classification of anomalies does not present as many dirty data types as the other three works. This is because Müller and Freytag's work does not consider problems from multi data sources. Their work limited the data quality problems to single data source.

Rahm and Do [6] classify data quality problems into two groups: single-source and multi-source problems. In each group, problems are described at different levels: schema and instance. For instance, at the instance level with single-source problems, data errors mainly come from data entry, such as misspellings, duplicates and contradictory values etc, while at the same level with multi-source problems, data errors present mainly because of the integration (overlapping, contradicting and inconsistent data). However, at single-source, they do not divide the problems into those that occur in a single relation and those that occur in multi relations as Oliveira *et al*'s done [4].

Kim *et al*'s work [1] presents a comprehensive taxonomy of dirty data, which is hierarchically structured. According to the different ways in which dirty data manifest, all dirty data that can be captured from different data sources is classified into the following three categories:

- Missing data;
- Not missing data but wrong data;
- Not missing, not wrong but unusable data;

The three categories of dirty data form the main body of the taxonomy work. For the rest of the taxonomy work, the authors apply a hierarchical decomposition method to the three categories of dirty data and produced a taxonomy with 33 types of dirty data types.

Oliveira *et al* produce a very complete taxonomy (Oliveira *et al*, 2005). They adopted a bottom-up approach, from the lowest level where data quality problems may exist (the ones that occur in a single attribute value of a single tuple) to the highest level (those that involve multi-source problems). At the single source level, problems are further divided into two sub-groups: those that occur in a single relation and those that result from existing relationships among relations. At the multi source level, the data quality problems are decomposed into 9

problems. The work also proposed some dirty data types that Kim *et al* have not mentioned, e.g. DT.7, DT.13, DT.16, DT.18. Although Oliveira *et al*'s work is by far the most comprehensive one compared with the other three works, the work still lacks of some dirty data types mentioned by the other works. For example, some dirty data types mentioned by Kim *et al* (DT.1, DT.19, DT.25, DT.34) are not included in Oliveira *et al*'s work.

In this paper, the rule-based taxonomy of dirty data proposed four categories of dirty data types against business rules in a tree structure. Within the category of business attribute rules, 5 dirty data types are identified. There are 4 dirty data types identified with each of the categories of business entity rules and data dependency rules. The majority of dirty data types are related to the category of data validity rules, which has 25 dirty data types. This is because the data value related problems are much more common than others. In total, there are 38 distinct dirty data types that are identified. The proposed taxonomy has considered dirty data types not only appeared within both a single data source and multiple data sources, but also from the angles of both a single relation and multiple relations. Comparing our work with the four existing works [1][3][4][5], it is clear that our taxonomy is most complete. For example, D26, D38, D12, D24, D13, D10 are dirty data types that are not mentioned in the works by Müller and Freytag [5] and Rahm and Do [6]. Comparing with the two formal taxonomy works by Kim *et al* [1] and Oliveira *et al* [4], apart from the problems due to the transaction management facilities [1] and at the schema level, the proposed taxonomy not only covers all dirty data types from these two taxonomies but also includes a new dirty data type, D.18, lack of data element. However, due to the research scope, schema-level related problems are not considered in the proposed taxonomy. For example, naming conflicts and structure conflicts are two schema level heterogeneties methioned in Rahm and Do's work [6]. Similarly, two schema-level problems are also identified in Oliveira *et al*'s work [4], i.e., Syntax inconsistency both in multiple relations in a single data source and among multiple data sources. This consideration agrees with the suggestion by Kim *et al* [1]. A systematic classification of schema related problems has been proposed by Kim and Seo [14], which covers all the schema-related problems mentioned by the two existing works [5][6].

Although it is believed that our taxonomy is very comprehensive, it is not ensured that it covers all possible dirty data types that may exist. However it is believed that most usual or unusual dirty data types are covered by this taxonomy. In addition, the example in section IV shows that it can easily help to solve the DDS problem.

VI. Conclusion and Future Work

In this paper, a rule-based taxonomy of dirty data is developed. Compared with existing works, this taxonomy provides a larger collection of dirty data types than any of existing taxonomies. With the help of the taxonomy, a new classification of dirty data based on data quality dimensions is proposed. Some existing works have also proposed a large collection of dirty data types, such as a collection of 35 dirty data types by Oliveira *et al*. However, by only looking at the dirty data types, it is difficult to tell which group of dirty data should be considered first and it would be very expensive for the system to run all algorithms for all the possible dirty data candidates. This is the Dirty Data Selection (DDS) problem. To deal with this problem, a method is developed. An example of using this method in section IV shows that it can be used by business enterprises to solve such a problem, by prioritizing the expensive process of data cleaning, therefore maximally benefit their organizations.

Future work will involve the development of a data cleaning tool to deal with dirty data types based on the proposed method. The challenge remains of how to organize the sequence to deal with the dirty data types that are identified as well as selecting suitable methods/algorithms according to different problem domains.

References

[1] W. Kim, B.Choi, E.Hong, S.Kim and D.Lee, "A Taxonomy of Dirty Data," Data Mining and Knowledge Discovery,7,81-99.

[2] L.Mong, "IntelliClean: A knowledge-based intelligent data cleaner," Proceedings of the ACM SIGKDD,Boston,USA,2000.

[3] W.Kim, "On three major holes in Data Warehousing Today," . Journal of Object Technology, Vol.1, No.4, 2002.

[4] P.Oliveira, F.T.Rodrigues, P.Henriques, and H.Galhardas, "A Taxonomy of Data Quality Problems," Second International Workshop on Data and Information Quality (in conjunction with CAISE'05), Porto, Portugal,2005.

[5] H. Müller and J.C.Freytag, ". Problems, Methods, and Challenges in Comprehensive Data Cleansing," Tech. Rep. HUB-1B-164,2003.

[6] E.Rahm and H.Do, "Data Cleaning: Problems and Current Approaches," IEEE Bulletin of the Technical Committee on Data Engineering. vol.23, 41, No.2, 2000.

[7] Monitoring Data Quality Performance Using Data Quality Metrics, http://www.it.ojp.gov/documents/Informatica_Whitepaper_Monitoring_DQ_Using_Metrics.pdf

[8] S.Adelman, L.Moss, and M.Abai, Data Strategy, Addison-Wesley Professional, 2005.

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

[10] M.Jarke, M.A.Jeusfeld, C.Quix, and P.Vassiliadis, "Architecture and Quality in Data Warehouses: an Extended Repository Approach," Information Systems, Vol.24, No.3, 1999.

[11] M.Bovee, R.P.Srivastava, and B.Mak, "A conceptual Framework and Belief-Function Approach to Assessing Overall Information Quality," . In Proceedings of the 6th International Conference on Information Quality, MIT Boston-MA,2001.

[12] C.Fox, A.Levitin, and T.Redman, "The notion of data and its quality of dimensions, " Information Processing & Management., vol. 30, no. 1. pp. 9-19, 1994.

[13] A.K.Elmagarmid, P.G.Ipeirotis, and V.S.VeryKios, "Duplicate Record Detection: A Survey," IEEE Trans. Knowl.Data Eng.,Vol.19, no1, pp. 1-16, 2007.

[14] W.Kim, and J.Y.Seo, "On classifying schematic and data heterogenety in multidatabase systems," IEEE Computer, 24(12), 1991.

Mr Lin Li received his BSc degree in Computer Science from Shenyang Aerospace University, China in 2004 and MSc degree in Software Engineering from the University of Manchester, UK in 2006. He is now pursuing his PhD degree at the school of Computing from Edinburgh Napier University,UK. His research interests include data cleaning and data quality improvement for database applications.

Professor Jessie Kennedy has been with the School of Computing at Edinburgh Napier University, UK for 19 years where she has held the post of professor since May 2000. Her research interests are in the fields of database systems, user interfaces to databases and data visualisation with special focus on bioinformatics and in particular biodiversity informatics. She has been working closely with taxonomists in the Royal Botanic Garden Edinburgh and has developed database systems for classifying biodiversity, an information visualisation tool for visualising multiple overlapping classification hierarchies as found in taxonomy, a model and ontology for capturing and relating character concept definitions in plant taxonomy and ontology-driven tool for the automated generation of data entry forms for taxonomic databases. Her work on taxonomic data models, databases and visualisation has continued with her involvement on the SEEK project. Related to this work she has been developing a Taxonomic Concept Transfer Schema in collaboration with the Global Biodiversity Information Facility and the International Taxonomic Databases Working Group. Her visualisation research has also included investigating techniques for visual exploration of micro array data, and on the EU OPaL project, investigating visualisation techniques for browsing partner attributes and assessment information in on-line communities.

Dr Taoxin Peng is a lecturer and researcher at the school of Computing from Edinburgh Napier University, UK. His main research interests include data quality and data cleaning, model-based reasoning, knowledge representation and temporal Reasoning