An Extension of the Relational Model to Intuitionistic Fuzzy Data Quality Attribute Model

Diana Boyadzhieva, Boyan Kolev

Abstract—**The model we suggest makes the data quality an intrinsic feature of an intuitionistic fuzzy relational database. The quality of the data is no more determined by the level of user complaints or ad hoc sql queries prior to the data load but it is stored explicitly in relational tables and could be monitored and measured regularly. The quality is stored on an attribute level basis in supplementary tables to the base user ones. The quality is measured along preferred quality dimensions and is represented by intuitionistic fuzzy degrees. To consider the preferences of the user with respect to the different quality dimensions and table attributes we create additional tables that contain the weight values. The user base tables are not intuitionistic fuzzy but we have to use an intuitionistic fuzzy RDBMS to represent and manipulate data quality measures.**
[1]

Index Terms—**data quality, quality model, intuitionistic fuzzy, relational database**

I. INTRODUCTION

Throughout the history of the data quality discipline, the customer-oriented applications of data have been the focus of many data quality initiatives. The discipline has generally focused on the domains customer information files, campaign management, compliance and transparency, enterprise information management (from business intelligence to master data management), integration in its various styles. However it is hard to define the exact essence of the data quality and that's why a lot of definitions exist [1] – [3] that stress different aspects of the task. If we have to provide a short, informal and intuitive definition of the concept, we could say that *data quality gives information about the extent to which the data is missing or incorrect.* But we could also define the data quality with a focus on the process character of the task: *A high-quality data is one that is fit for its intended uses (in operations, decision-making, planning, production systems, science etc.) and data quality is the process that encompasses all the tasks involved in the assurance of these high-quality data.* Juran defines quality simply as "fitness for use" [4]. The ISO 9000 revision IS9000:2005 defines quality as: "Degree to which a set of inherent characteristics fulfills requirements" [5].

II. THE MODEL JUSTIFICATION

Data quality could be controlled across several different aspects of the existence and operation of an information system. The data quality could concern:

- The design of the database – i.e. the quality of the logical or physical database schema or
- Could refer the data values that are inserted, stored and updated during the entire data flow of the information.

Here we concentrate on the second subject - the attribute and tuple level data quality. Important issues here are how a high data quality is achieved in a system and how often complementary tasks are performed in order to maintain the desired level of data quality. A lot of researchers and practitioners have developed methodologies and tools to enhance the data quality in an IS, mainly by identification and cleaning of the errors in data prior to the data load into the IS or during an integration process. In this approach, the assertion is that only high quality data enter the database. The problem here is that the extent of this "high" quality is not exactly measured, as well as the fact that the quality of data usually degrades with the time of the data existence in the system. In this paper we present a model where data quality is incorporated in the overall design of a database. The relational model is extended with supplementary tables where the exact quality level on an attribute level is explicitly saved. Such a model readily provides quality information at disposal. The quality measures should be continuously updated during the life-cycle of the data in the information system in order to reflect the actual quality of the attribute values which is not always a constant. Attribute-based approach is presented also in [6] but we leverage intuitionistic fuzzy logic. We do not put requirements on the database to be an intuitionistic fuzzy one but we need to use an intuitionistic fuzzy RDBMS to represent and manipulate the data quality measures. We use the Intuitionistic Fuzzy PostgreSQL (IFPG) [7], [8], giving the possibility to store and manage intuitionistic fuzzy relations.

III. THE INTUITIONISTIC FUZZY DATA QUALITY ATTRIBUTE MODEL

Before the explanation of the model, we shortly describe the notion of quality dimensions. For many people data quality means just accuracy. However the quality of data is better represented if it is measured also along other - descriptive for the specific data - qualitative characteristics. Each of these descriptive qualitative characteristics is called a quality dimension. The choice of quality dimensions that will be measured depends on the user requirements and is the theoretical, empirical and intuitive aproaches are described in [9]

In the intuitionistic fuzzy data quality attribute model, we store the quality on an attribute level basis – i.e. we store measures of the quality of the values in the user tables /tables I a)/. We keep these quality measures in supplementary table that we call quality table /tables I b)/. We propose to store and monitor data quality not for all attributes in a user table but only for some of them – those that bring critical values for the user. The user requirements,

Diana Boyadzhieva, Sofia University "St. Kliment Ohridski", Faculty of Economics and Business Administration, 125 Tzarigradsko chaussee Blvd., Bl. 3, Sofia-1113, BULGARIA
Boyan Kolev, Centre for Biomedical Engineering - Bulgarian Academy of Sciences, Acad.G.Bonchev Str., Bl.105, Sofia-1113, BULGARIA

the potential type of tasks and requests to the data determine which these attributes of a special interest are. For each such attribute of a special interest we add in the quality table one record for each quality dimension that we want to measure. The table contains two attributes which represent μ and ν intuitionistic fuzzy degrees that measure the quality along the respective quality dimension.

Let we agree upon the following terminology. The attributes in the user tables (containing the source data) we will call ordinary attributes. The extent to which it is sure that a given characteristic of the data is present along a quality dimension we will call presence of quality or positive quality. The extent to which it is sure that a given characteristic of the data does not exist along a quality dimension we will call absence of quality or negative quality. The indefiniteness about the presence of quality we will call indefinable quality.

In the defined terminology, μ measures the degree of positive quality, ν measures the degree of negative quality and the indefinable quality is 1 - μ - ν. If the user table contains a few attributes and if the tracked quality dimensions are not a lot, we could not create a separate quality table but keep the ordinary attributes and the quality attributes in a single table. However to keep the things clear we offer to follow an alternative approach – to create the attributes that will keep the quality measures in a separate table (we call it quality table) that refers the respective user table with the ordinary attributes /tables I a), b)/ The intuitionistic fuzzy degree μ is represented by the attribute MSHIP and the intuitionistic fuzzy degree ν is represented by the attribute NMSHIP.

The relative importance that the user assigns to each quality dimension of an ordinary attribute is modeled as a weight. This weight gives the share of the respective quality dimension in the calculation of the quality of a given value in the respective ordinary attribute. Actually these weights give the relative importance that the user assigns to each dimension. We assume the weights are normalized, i.e. for each ordinary attribute, the dimension weights sum up to 1. The weights are stored in a dimension-weights table /tables1 c)/.

Furthermore, we expand the model with another metadata table which contains the weight of the quality of each ordinary attribute value in the calculation of the total quality of a tuple in a table /tables I d)/. These weights give the relative importance of an ordinary attribute for the calculation of the quality of a tuple. The table represents the attribute weights for the attributes of all tables in the database. We assume the weights are normalized, i.e. for each table, the attribute weights sum up to 1.

TABLES I, a), b), c), d)

TableX

Attr1_key	Attr2	Attr3	Attr4

a)

TableX_Quality

Attr1_Key	Attribute_Name	Dimension_Name	MSHIP	NMSHIP

b)

Dimension_Weights

Table_Name	Attribute_Name	Dimension_Name	Weight

c)

Attribute_Weights

Table_Name	Attribute_Name	Weight

d)

To calculate the quality measures, three methods could be utilized. In the first one the data editor introduces the measures based on user-defined criteria. In the second one, the system calculates the quality measures based on a set of user-defined logic or calculations (for instance a set of real-world categorical words like very weak, weak, strong, very strong, etc. could be automatically mapped to a number value). In the third one – the quality values could be result from the integration and data cleansing tool. In this case supplementary to the cleansed data, on the basis of the manipulations on the data the data cleansing tool should provide on its output aslo enough information for calculation of the intuitionistinc fuzzy degrees for the data quality along the respective quality dimensions. Principles that can help the users develop usable data quality metrics are described in [10].

TABLES II, a), b), c), d)

Client

ID	FName	LName	Address	Phone	Salary
100001	Peter	Ivanov	18 Rakovski Str.	844567	1000

a)

Client_Quality

ID	Attribute_Name	Dimension	MSHIP	NMSHIP
100001	Address	Currency	0.8	0.1
100001	Phone	Currency	0.7	0.1
100001	Salary	Currency	0.6	0.1
100001	Salary	Believability	0.8	0.1

b)

Dimension_Weights

Table	Attribute_Name	Dimension	Weight
Client	Address	Currency	1
Client	Phone	Currency	1
Client	Salary	Currency	0.4
Client	Salary	Believability	0.6

c)

Attribute_Weights

Table	Attribute_Name	Weight
Client	Address	0.4
Client	Phone	0.4
Client	Salary	0.2

d)

Let us consider an example where a company has to conduct a marketing campaign. We deside to keep track not only of the client data but also of the quality of data on an attribute-level basis. We extend the relational model with supplementary tables, which contain the quality measures for each attribute on one or more quality dimensions. In our example, this supplementary table for the table Client /tables II a)/ is the table Client_Quality /tables II b)/ presented only with records for a given Client ID. We can consider this table an intuitionistic fuzzy relation, where the degrees of membership and non-membership represent the extent to which the corresponding attribute value fulfils the quality requirements at a certain quality dimension. In the table Client_Quality we add one record for each quality dimension that has to be tracked for those client attributes that are of a special interest. Each record contains respectively the μ and ν measures of the quality along the respective dimension. For instance the Salary attribute has to be measured along two quality dimensions – currency and believability, thus for this attribute in the table Client_Quality we add two records / tables II b)/ In the record for client with ID 100001, the salary' currency MSHIP contains a measure showing the extent to which the Salary is current, NMSHIP contains a measure showing the extent to which the Salary is not current. The last row in our example measures the probability that the salary of the client with ID 100001 is the real one or the probability that the client lied about his salary. In other words, the intuitionistic fuzzy degrees of membership and non-membership answer the question (vague terms are highlighted) 'How *high* is the *believability* that the salary for client with ID 100001 is the one pointed in the database?'

We will use IFPG database engine to represent and manipulate data quality measures. An important feature of this intuitionistic fuzzy RDBMS is the processing of queries with intuitionistic fuzzy predicates, e.g. predicates which correspond to natural language vague terms like 'high', 'cheap', 'close', etc. These predicates are evaluated with intuitionistic fuzzy values, which reflect on the degrees of membership and non-membership of the rows in the query result, which is in fact an intuitionistic fuzzy relation.

IV. Calculating the quality for an attribute value at a certain dimension

We can create an intuitionistic fuzzy predicate which presents the quality of a certain attribute value at a certain dimension. Given this functionality the user is capable to filter the data on quality-measure basis.

```
CREATE PREDICATE high_qualty_client_attr_dim
       (integer, varchar, varchar)
AS '
SELECT MSHIP, NMSHIP
FROM Client_Quality
WHERE ID = $1
    AND Attribute_Name = $2
    AND Dimension = $3
' LANGUAGE sql;
```

The user can now make queries of the kind 'List all clients with *high believability* for the real value of their salaries' and even define threshold to filter those records with demanded minimal value of the quality measure:

```
SELECT ID, Address, Phone, Salary,
   'Believability' as Quality_Dim,
   MSHIP, NMSHIP
FROM Client
WHERE high_qualty_client_attr_dim
   (ID, 'Salary', 'Believability')
HAVING MSHIP > 0.6;
```

ID	Address	Phone	Salary	Quality_Dim	MSHIP	NMSHIP
100001	18 Rakovski Str.	844567	1000	Believability	0.8	0.1

V. Calculating the overall quality for an attribute value

Since an attribute value may have more than one quality dimension, the overall quality of the attribute value has to be calculated considering the quality measures of all its dimensions. This may help the user make analyses on the basis of the total quality of a certain attribute value. For the purpose we introduce a metadata table *Dimension_Weights* /tables II c)/, containing weights of the quality dimensions, which participate in the calculation of the overall quality of each attribute value:

The calculation of the overall quality of attribute values in table *Client* is performed with the following SQL query:

```
SELECT Client_Quality.ID, Client_Quality.Attribute_Name,
    SUM(Client_Quality."mship" * Dimension_Weights.Weight),
    SUM(Client_Quality."nmship" * Dimension_Weights.Weight)
FROM Client_Quality JOIN Dimension_Weights
    ON Client_Quality.Attribute_Name = Dimension_Weights.Attribute_Name
        AND Client_Quality.Dimension = Dimension_Weights.Dimension
WHERE Dimension_Weights.Table_Name = 'Client'
GROUP BY Client_Quality.ID, Client_Quality.Attribute_Name;
```

Follows the result of the query applied on the table Client with the example data for just one client.

ID	Attribute_Name	MSHIP	NMSHIP
100001	Address	0.8	0.1
100001	Phone	0.7	0.1
100001	Salary	0.72	0.1

This intuitionistic fuzzy relation represents the overall quality of attribute values in table *Client*. For instance the third row of the table answers a question of the kind 'How *high* is the overall *possibility* that the salary of the client with ID 100001 is the one pointed in the database?'

Analogously we can create an intuitionistic fuzzy predicate which presents the overall quality of a certain attribute value. Thus the user is capable to filter the data based on the total attribute value quality.

```
CREATE PREDICATE high_quality_for_client_attribute_value (integer, varchar)
AS
'SELECT SUM(Client_Quality."mship" * Dimension_Weights.Weight),
    SUM(Client_Quality."nmship" * Dimension_Weights.Weight)
FROM Client_Quality JOIN Dimension_Weights
    ON Client_Quality.Attribute_Name = Dimension_Weights.Attribute_Name
        AND Client_Quality.Dimension = Dimension_Weights.Dimension
WHERE Dimension_Weights.Table_Name = ''Client''
    AND Client_Quality.Attribute_Name = $2
    AND Client_Quality.ID = $1 '
LANGUAGE sql;
```

The user can now make queries of the kind 'List all clients with *high* overall *possibility* for the real value of their salaries' and even define threshold to filter those records with demanded minimal value of the quality measure:

```
SELECT ID, Address, Phone, Salary, MSHIP, NMSHIP
FROM Client
WHERE high_quality_for_client_attribute_value
      (ID, 'Salary')
HAVING MSHIP > 0.6;
```

ID	Address	Phone	Salary	MSHIP	NMSHIP
100001	18 Rakovski Str.	844567	1000	0.72	0.1

VI. Calculating the overall quality of a tuple

For some kind of analyses, the quality of data in a tuple as a whole may be of importance. For calculating the overall quality of a tuple we consider the overall qualities of each of the attribute values in the tuple. For the purpose we introduce another metadata table *Attribute_Weights* /tables II d)/, containing weights of the quality of attributes, which participate in the calculation of the overall quality of each tuple:

The calculation of the overall quality of tuples in the relation *Client* is performed with the following SQL query:

```
SELECT Client_Quality.ID,
    SUM(Client_Quality."mship" * DW.Weight * AW.Weight),
    SUM(Client_Quality."nmship" * DW.Weight * AW.Weight)
FROM Client_Quality
    JOIN Dimension_Weights DW
        ON Client_Quality.Attribute_Name = DW.Attribute_Name
        AND Client_Quality.Dimension = DW.Dimension
    JOIN Attribute_Weights AW
        ON Client_Quality.Attribute_Name = AW.Attribute_Name
WHERE DW.Table_Name = 'Client'
    AND AW.Table_Name = 'Client'
GROUP BY Client_Quality.ID;
```

The result intuitionistic fuzzy relation represents the overall quality of tuples in table *Client*, each row of which answers the question 'How *high* is the overall *quality* of data about client with ID 100001 pointed in the database?'

ID	MSHIP	NMSHIP
100001	0.744	0.1

Analogously an intuitionistic fuzzy predicate *high_quality_tuple* may be created which can help the user make queries of the kind 'List all the clients, the information about which is more than 60% *reliable*':

```
CREATE PREDICATE high_quality_tuple (integer)
AS
'SELECT SUM(Client_Quality."mship" * DW.Weight * AW.Weight),
        SUM(Client_Quality."nmship" * DW.Weight * AW.Weight)
FROM Client_Quality
    JOIN Dimension_Weights DW
        ON Client_Quality.Attribute_Name = DW.Attribute_Name
        AND Client_Quality.Dimension = DW.Dimension
    JOIN Attribute_Weights AW
        ON Client_Quality.Attribute_Name = AW.Attribute_Name
WHERE DW.Table_Name = ''Client''
    AND AW.Table_Name = ''Client''
    AND Client_Quality.ID = $1
GROUP BY Client_Quality.ID '
LANGUAGE sql;
```

The following select uses the *high_quality_tuple* predicate and returns only those records that have positive quality grater than the specified threshold.

```
SELECT ID, Address, Phone, Salary, MSHIP, NMSHIP
FROM Client
WHERE high_quality_tuple(ID)
HAVING MSHIP > 0.6;
```

ID	Address	Phone	Salary	MSHIP	NMSHIP
100001	18 Rakovski Str.	844567	1000	0.744	0.1

VII. Calculating the overall quality of the attributes

On the basis of the currently available values in a user table and their current quality, we could calculate the overall quality of the attributes in a user table. For a given attribute we consider the overall quality of an attribute value in a tuple and we avarage this quality along all the records. The following query performs these calculations for the table Client:

```
SELECT QS.Attribute_Name,
       avg(QS.sum_Quality_MSHIP) as Attr_Quality_MSHIP,
       avg(QS.sum_Quality_NMSHIP) as Attr_Quality_NMSHIP
FROM (SELECT ID, DW.Attribute_Name,
          sum (Client_Quality."mship" * DW.Weight) AS sum_Quality_MSHIP,
          sum (Client_Quality."nmship" * DW.Weight) AS sum_Quality_NMSHIP
      FROM Client_Quality
          JOIN Dimension_Weights DW
              ON Client_Quality.Attribute_Name = DW.Attribute_Name
              AND Client_Quality.Dimension = DW.Dimension
      WHERE DW.Table_Name = 'Client'
      GROUP BY ID, DW.Attribute_Name) AS QS
GROUP BY QS.Attribute_Name;
```

The result is an intuitionistic fuzzy relation that contains as many rows as is the number of the attributes in *Client* whose quality we track. Each row represents the overall quality of the respective attribute on the basis of the current quality of the all the values in this attribute.

Attribute_Name	Attr_Quality_MSHIP	Attr_Quality_NMSHIP
Address	0.8	0.1
Phone	0.7	0.1
Salary	0.72	0.1

VIII. Conclusion

The utility of this model could be in several directions. First, the queries, could manipulate only the values (records) having a quality greater than a certain threshold. Second – a query could act over all the records but the result could provide also a measure for the quality of the respective result along given dimensions or as a total. Third - a quality measuring method could be devised for calculation of the current quality of a given table or of the whole database. Fourth – the introduction of quality tracking in the database will outreach the framework of the information system and will make the employees put greater emphasis on the quality of their work. As the users are in fact the ultimate judges of how high quality of the data they need, then they will best take care to consider and improve quality of the data on an on-going basis.

References

[1] R.Y. Wang, D. Strong, L.M. Guarascio, "Beyond Accuracy: What Data Quality Means to Data Consumers", *Technical Report TDQM-94-10, Total Data Quality Management Research Program*, MIT Sloan School of Management, Cambridge, Mass., (1994)

[2] K. Orr, "Data quality and systems theory, "*Communications of the ACM"*, 41(2), 66-71, (1998)

[3] G.K. Tayi, D. P. Ballou, "Examining Data Quality", *Communications of the ACM*, 41(2), 54-57 (1998)

[4] Joseph Juran, *"The Quality Control Handbook"*, McGraw-Hill, New York, 3rd edition, 1974

[5] ISO 9000:2005, *"Quality Management systems"*, *Fundamentals and vocabulary*, 2005.

[6] R.Y. Wang, M.P. Reddy, H.B. Kon, "Towards Quality Data: An attribute-based Approach", *Decision Support Systems*, vol. 13, 1995.

[7] Kolev, B., "Intuitionistic Fuzzy PostgreSQL", *Advanced Studies in Contemporary Mathematics*, 11 (2005), No. 2, pp 163-177

[8] Kolev B., P. Chountas, K. Atanassov, I. Petrounias, "Representing Uncertainty and Ignorance in Probabilistic Data Using the Intuitionistic Fuzzy Relational Data Model", Issues in the Representation and Processing of Uncertain and Imprecise Information. Fuzzy Sets, Generalized Nets and Related Topics, *Akademicka Oficyna Wydawnicza EXIT*, Warszawa 2005, pp. 198-208

[9] C. Batini, M. Scannapieca, *"Data Quality - Concepts, Methodologies and Techniques"*, Springer, 26-42 (2006)

[10] Leo L. Pipino, Yang W. Lee, and Richard Y. Wang, "Data Quality Assessment", *"Communications of the ACM"*, April 2002/Vol. 45, No. 4ve, 211 - 218