JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 27, 681-695 (2011)

A Solution for Data Inconsistency in Data Integration*

XIN WANG, LIN-PENG HUANG, XIAO-HUI XU, YI ZHANG AND JUN-QING CHEN
Department of Computer Science and Engineering
Shanghai Jiao Tong University
Shanghai, 200240 P.R. China

Data integration is a problem of combining data residing at different sources and providing the user with a unified view of these data. An important issue in data integration is the possibility of conflicts among the different data sources. Data sources may conflict with each other at data value level which is defined as data inconsistency. So in this paper, a solution for data inconsistency in data integration is proposed. An approximate object-oriented data model extended with data source quality criteria is defined. On the basis of our data model, we provide a data inconsistency solution strategy. To accomplish our strategy, fuzzy multi-attribute decision making approach based on data source quality criteria is applied to select the "best" data source's data as the data inconsistency solution. A set of experiments is designed and performed to evaluate the effectiveness of our strategy and algorithm. The experimental results indicate that our solution performs ideally.

Keywords: data integration, data inconsistency, decision making, data source quality criteria, data fusion

1. INTRODUCTION

Data integration has been a long-standing challenge in the database and AI communities [1]. Given a collection of heterogeneous and autonomous data sources, a data integration system allows its users to perceive the entire collection as a single source, query it transparently, and receive a single, unambiguous answer [2].

Dealing with inconsistencies is one of the important challenges in data integration. Data sources may conflict with each other at three different levels [3]: schema level: sources are in different data models or have different schemas within the same data model; data representation level: data in the sources is represented in different natural languages or different measurement systems; and data value level: there are factual discrepancies among the sources in data values that describe the same objects.

A data value level inconsistency exists when two objects (or tuples in the relational model) obtained from different data sources are identified as versions of each other and some of the values of their corresponding attributes differ [3]. It is referred to as data inconsistency in [2] and may be detected in query results when processing users' queries.

For example, in a data integration system, two employee records stored in two different data sources refer to the same person – Smith. And the age of Smith recorded in these two data sources are 30 and 35 respectively. This is a data inconsistency. If we propose a query in data integration system about the age of Smith, data inconsistency will be detected from the query results.

Received March 10, 2009; revised May 26 & July 16, 2009; accepted September 26, 2009.
Communicated by Suh-Yin Lee.
* The work was supported by the National Natural Science Foundation of China under Grant No. 60970010 and 973 Program of China No. 2009CB320705.

In a diverse distributed environment, data sources have their individual advantages and disadvantages. For example, some data sources' data are more recent, whereas others are more dated; some data come from authoritative sources, whereas other may have dubious pedigree [2]. Then in order to resolve data inconsistency, it is reasonable to look at the qualifications of its individual data providers. Obviously, "good" data sources should be preferred when resolving data inconsistency.

So in this paper, we make use of data source quality criteria to resolve data inconsistency in data integration. Firstly, we define data source quality criteria and provide a data integration data model which extends the data model of [4] with data source quality criteria vector. Then based on our data model, we provide data inconsistency solution strategy. In our strategy, fuzzy multi-attribute decision making approach based on data source quality criteria is applied to select the "best" data source's data as the data inconsistency solution. The experimental results indicate that our algorithm has ideal performance.

In summary, this paper makes the following main contributions: (1) a new data model for data integration is provided to meet the needs of defining, discovering and resolving data inconsistency. (2) Fuzzy multi-attribute decision making approach is the first time to be used in data inconsistency solution for data integration. Accordingly, our solution is the only method being able to process data source quality criteria with qualitative values. (3) We design and perform an experimental evaluation of our algorithm and strategy.

The rest of the paper is organized as follows: section 2 discusses related work; section 3 describes the model used in our paper; section 4 demonstrates data inconsistency solution; section 5 provides the experiments; section 6 gives the conclusion.

2. RELATED WORK

There are some approaches for resolving data inconsistency in data integration based on the content of the conflicting data [3]. [5, 6] detect the existence of data inconsistencies and provide users with some additional information on their nature. [7, 8] resolve the data inconsistency through use of probabilistic data. Though they have some advantages in some aspects, none of them take into account the influence of data source quality.

[4, 9] have done some research on data integration based on data source quality. However, they do not provide data inconsistency solution.

There are some research works on data inconsistency solution based on data source properties such as [2, 10]. These research works can only process data source properties with quantitative values. Nowadays, as described in [11], this can not satisfy the needs of business application and more and more researchers consider it unsuitable. This is because in order to describe characteristics of data sources better, the qualitative values should be used for describing data source properties. In data integration, the ability of processing qualitative values of data source properties for data inconsistency solution is needed.

However, in our solution, we apply fuzzy multi-attribute decision making approach to resolve data inconsistency by processing data source quality criteria with quantitative values and qualitative values.

In addition, [2] only provides fusion strategies instead of some specific algorithms and does not give the illustration of its data inconsistency solution effects. Taking [2] and [11] one step further, we analyze the effects of our solution by experiments.

3. THE INTEGRATION DATA MODEL

Data model for data integration is important for describing and reasoning about the contents of data sources. And it is also critical for describing, discovering and resolving data inconsistency. In this section, we propose an approximate object-oriented data model extending with data source quality criteria. According to our data model, data is organized with "class". Each class represents a collection of objects. Attributes describe the detailed features of classes.

In order to integrate data from distributed data sources, our solution applies data schemas as unified formats or "views", which all the heterogeneous data sources should follow in data integration. There are two kinds of data schemas: global schema and local schema. Local schemas are provided by data sources describing local data by using local classes. Global schemas are obtained by integrating local classes in local schemas and describe all the data in data integration system by using global classes.

3.1 Data Source Quality Criteria

In our data inconsistency solution, we make use of data source QoS properties – we call data source quality criteria to represent data quality in data sources. For the value of data source quality criteria, some of them may be provided by data source itself, some may be obtained informally from other Internet users, and there are also web sites that are dedicated to calculating the data source QoS properties.

There are six data source quality criteria used in this paper including [3, 12, 13]:

- **Cost:** cost is the amount of money that network retrieval cost and/or amount of money to be paid for the information in this query.
- **Time:** time is a common measure of the performance's time.
- **Availability:** availability means the probability that at a random moment the data source is available.
- **Clearance:** clearance means the security clearance needed to access the data.
- **Reliability:** Reliability represents the ability of a data source to function correctly and consistently despite system or network failures.

For the granularity of data source quality criteria, in this paper, all data source QoS properties are assumed to be inherited by all individual objects and all their attribute values within the same data sources.

According to the discussion above, data source quality criteria vector for attribute A on local class C in data source s is defined as following: $q(s.C.A) = (q_{time}(s.C), q_{cost}(s.C), q_{av}(s.C), q_{cln}(s.C), q_{rel}(s.C))$.

3.2 Data Model for Data Integration

In this paper, we modify data integration data model in [4] by adding data source quality criteria. Our data model is also designed without using relational model which is applied in [4] for the convenience of semantic data integration.

Data model used in our solution includes (1) classes and a class hierarchy: there is a

partial order $\prec$ such that $C \prec D$ whenever class C is a subclass of class D. (2) A set of attributes associated with each class. (3) A data source quality criteria vector associated with each attribute for a certain class.

Classes contain objects. The attribute should be single-valued. And the attribute value should be atomic value such as string or integers. An object can belong to more than one class. It is possible to declare a pair of classes to be disjoint which means that no object can belong to both classes.

There are some definitions based on our data model are provided as follows:

- For attribute A on class C, $o.A = y$ means the value of attribute A for object o is y. Here y is called the A-filler of o.
- For attribute A on class C and A's data source quality criteria vector Q, $o.A.Q = qv$ means the value of data source quality criteria vector of attribute A for object o is qv. Here o is an object of C and data source s provides C and o. And qv is expressed by $q(s.C.A) = (q_{time}(s.C), q_{cost}(s.C), q_{av}(s.C), q_{cln}(s.C), q_{rel}(s.C))$.

Data Schema

There are two kinds of data schemas used in our solution as unified formats or "views" to describe data in data integration system. They are based on the data model we provide.

Local schema: Local schemas are provided by data sources in data integration system to describe each data source's local data with local classes and generate "local class tree" by using our data model.

Global schema: Global schemas are obtained by integrating the local classes in local schemas and generating "global class tree" which describes all the data in data integration system with global classes. The user poses queries in terms of the global schema. It is a collection of virtual classes of local schemas and no data actually are stored in global classes.

Data Inconsistency Definition

Definition 1 Data Inconsistency: in query result R, if (1) for object set OS:$\{o_i\}$ $(1 \le i \le n)$, each object $o_i \in OS$ is obtained from different data sources and refers to the same object RO in the real world and (2) attribute set $\{A_j\}$ $(1 \le j \le m$, A_j is an attribute on local class C and o_i is an object of $C)$ refer to the same attribute of RO and the values of them appear in R and (3) the corresponding attribute in the global classes to all $A_j \in \{A_j\}$ is GA (4) the corresponding value $o_i.A_j$ of each $A_j \in \{A_j\}$ are not equal which means GA is not single-valued. Then we say there is a data inconsistency existing in OS of R. And we call attribute set $\{A_j\}$ – inconsistent local attribute set, GA – inconsistent global attribute and OS – polyobject.

Specifically, in polyobject OS of R, for every $o_i \in OS$ and $A_j \in \{A_j\}$ where $\{A_j\}$ is an inconsistent local attribute set, if $o_i.A_j$ appears in data integration query result, $o_i.A_j.Q$ is collected and recorded for data inconsistency solution. For the global attributes that do not have data inconsistencies, the data source quality criteria vector can be ignored or set to the same value.

4. DATA INCONSISTENCY SOLUTION

We assume data inconsistencies have been detected according to the methods mentioned in [2] by using keys to identify objects that are versions of each other. Key here means an attribute or a set of attributes to decide objects referring to the same object in the real world (*e.g.* we can use social security number of a person as the key to identify whether two objects refer to the same person). And how to select the key is not discussed in detail in this paper. We assume that each global class in global schema is fitted with a key and the key of query result R can be constructed and be used to cluster R. In the resulting clusters, if there exist polyobjects (data inconsistency), they will be resolved.

Specifically, in this section, if data inconsistency caused by inconsistent local attribute set $\{A_j\}$ ($1 \le j \le m$, A_j is an attribute on local class C, o_i is an object of C, $o_i \in OS$, $1 \le i \le n$) exists in polyobject OS: $\{o_i\}$ ($1 \le i \le n$) of query result R, it will be resolved by the way of selecting the most appropriate $o_i.A_j$ provided by the "best" data source. The selection of the "best" data source will use fuzzy multi-attribute decision making approach [14, 15] based on $\{o_i.A_j.Q\}$ ($1 \le j \le m$, A_j is an attribute on local class C, o_i is an object of C, $A_j \in \{A_j\}$, $o_i \in OS$, $1 \le i \le n$) which can select the most appropriate data source providing $o_i.A_j$ with the highest degree of membership belonging to the positive ideal solution.

Our data inconsistency solution strategy for data inconsistency in query result R consists of the following steps.

(1) Obtain Fusion Matrix

Firstly, according to inconsistent local attribute set $\{A_j\}$ in query result R, we obtain $o_i.A_j.Q$ of each $o_i.A_j$ ($A_j \in \{A_j\}$, o_i is provided by s_i). In $o_i.A_j.Q$, some of the data source quality criteria are quantitative criteria which values are quantitative values such as numbers. For example, cost is a quantitative criterion in our paper and in Table 3, the value of cost of data source S_1 is 49. And others are qualitative criteria which values are qualitative values such as some qualitative words including "good", "bad" or "very bad" *et al*. For example, clearance in our paper is a qualitative criterion and in Table 3, the value of clearance of S_1 is "very bad" ("very bad" corresponds to (1, 1, 2) according to Table 1).

So in our strategy, we introduce triangular fuzzy number [14] to represent the values of qualitative criteria.

Definition 2 Set $F(R)$ be the total fuzzy sets of R and set $f \in F(R)$. The membership function is $\mu_f(x) = \begin{cases} \dfrac{x-a}{b-a}, & x \in [a, b] \\ \dfrac{x-c}{b-c}, & x \in [b, c] \end{cases}$, $a \le b \le c$. b is the mean of f, a is the upper bound of f and c is the lower bound of f written as $a = m_L$; $c = m_R$. Then we say f is the triangular fuzzy number written as $f = (m_L, m, m_R) = (a, b, c)$.

For qualitative criterion, it can use triangular fuzzy number scaling method which is improved from bipolar scaling method to transform the value of it into triangular fuzzy number. The transformation form between two kinds of data used in this paper is shown in Table 1.

Table 1. Transform qualitative value by using triangular fuzzy number scaling.

Criteria	Triangular Fuzzy Number						
	(0, 0, 1)	(1, 1, 2)	(2, 3, 4)	(4, 5, 6)	(6, 7, 8)	(7, 8, 9)	(9, 10, 10)
Negative Criteria	highest	very high	high	not high	low	very low	lowest
Positive Criteria	worst	very bad	bad	not bad	good	very good	perfect

To unify the expression of data source quality criteria, the value of quantitative criterion can also be written as the triangular fuzzy number form. For example "i" is a quantitative number and it can be expressed as (i, i, i).

Once $o_i.A_j.Q$ is derived and the transformation mentioned above is performed, fusion matrix $F_{n\times 6}$ can be obtained where $F = (f_{ij})_{n\times 5} = (q_{ij})_{n\times 5}$ and $f_{ij} = (a_{ij}, b_{ij}, c_{ij})$. The ith row of F represents the quality criteria vector $o_i.A_j.Q$ of the candidate data source s_i.

(2) Scale

Then for $f_{ij} = (a_{ij}, b_{ij}, c_{ij})$, the negative criteria values are scaled according to $r_{ij} = (\frac{\min\limits_{1\le i\le n} a_{ij}}{c_{ij}}, \frac{\min\limits_{1\le i\le n} b_{ij}}{b_{ij}}, \frac{\min\limits_{1\le i\le n} c_{ij}}{a_{ij}} \wedge 1)$, $(1 \le i \le n;\ 1 \le j \le 5)$. $r_{ij} = (\frac{a_{ij}}{\max\limits_{1\le i\le n} c_{ij}}, \frac{b_{ij}}{\max\limits_{1\le i\le n} b_{ij}}, \frac{c_{ij}}{\max\limits_{1\le i\le n} a_{ij}} \wedge 1)$ $(1 \le i \le n;\ 1 \le j \le 5)$ is used for positive criteria. So we obtain the matrix $R = (r_{ij})_{n\times 5}$.

(3) Construct Fusion Decision Matrix

Every data source quality criterion has its weight in our solution. We assume the weight of each data source quality criterion has quantitative value. We set weight vector $w = (w_1, w_2, w_3, w_4, w_5)$ where w_j $(w_j \ge 0)$ represents the weight of jth data source quality criterion and $\sum_{j=1}^{5} w_j = 1$. The key issue in weight vector estimation is the determination of the confidence level of such estimates [16]. However, the weight vector can also be specified by users to express their preferences for data source quality criteria in data inconsistency solution or it can be provided by experts.

Set $D = (d_{ij})_{n\times 5}$ and weight vector $w = (w_1, w_2, w_3, w_4, w_5)$. According to $r_{ij} = (g_{ij}, e_{ij}, f_{ij})$, $d_{ij} = (d'_{ij}, e'_{ij}, f'_{ij}) = (w_j g_{ij}, w_j e_{ij}, w_j f_{ij})$, $(1 \le i \le n;\ 1 \le j \le 5)$. So the fusion decision matrix is $D = (d_{ij})_{n\times 5}$.

(4) Compute Distance to the Positive Ideal Solution and Negative Ideal Solution for Alternatives

In this paper, we compare two triangular fuzzy numbers using v_{ij} defined as following $v_{ij} = (d'_{ij} + 2e'_{ij} + f'_{ij})/4$ for the triple d'_{ij}, e'_{ij}, f'_{ij} of the triangular fuzzy number d_{ij}. So we can obtain the matrix $V = (v_{ij})_{n\times 5}$. Row vector $V_i = \{v_{i1}, v_{i2}, v_{i3}, v_{i4}, v_{i5}\}$ represents the relative quality of s_i.

Quality vector of the positive ideal solution denoted by M^+ is defined as:

$M^+ = (M_1^+, M_2^+, M_3^+, M_4^+, M_5^+)$ and $M_j^+ = \max\limits_{1\le i\le n} v_{ij}$, $(1 \le i \le n;\ 1 \le j \le 5)$.

Quality vector of the negative ideal solution denoted by M^- is defined as:

$M^- = (M_1^-, M_2^-, M_3^-, M_4^-, M_5^-)$ and $M_j^- = \min\limits_{1\le i\le n} v_{ij}$, $(1 \le i \le n;\ 1 \le j \le 5)$.

The Euclidean distance l_i^+ between candidate data source s_i and the positive ideal solution is defined as:

$$l_i^+ = \sqrt{\sum_{j=1}^{6}(v_{ij} - M_j^+)^2}, \ (1 \le i \le n; 1 \le j \le 5).$$

The Euclidean distance l_i^- between candidate data source s_i and the negative ideal solution is defined as:

$$l_i^- = \sqrt{\sum_{j=1}^{6}(v_{ij} - M_j^-)^2}, \ (1 \le i \le n; 1 \le j \le 5).$$

(5) Fuzzy Optimize for Data Source Selection

Firstly, calculate degree of membership of each candidate data source s_i belonging to the positive ideal solution.

Define a membership function $\mu(V_i)$ which provides the degree of membership of V_i belonging to M^+:

$$\mu(V_i) = \frac{(l_i^-)^2}{(l_i^+)^2 + (l_i^-)^2}, \ (1 \le i \le n).$$

The definition of V_i indicates that $\mu(V_i)$ also represents the degree of membership of candidate data source s_i belonging to the positive ideal solution M^+. Set vector $\mu = (\mu(V_1), \mu(V_2), \ldots, \mu(V_n))$. Sorting the vector μ and the final decision is the candidate data source with the maximum of $\mu(V_i)$. Then the data of that data source provided in data inconsistency is selected as the solution result and joined to the query result R. After all the data inconsistencies in query result R are resolved, R is inconsistency-free and is provided to user.

Case Study

For simplicity, a polyobject of global class: person (ID, Name, Age, Salary) may be visualized in Table 2.

Table 2. Example of polyobject.

ID	Name	Age	Salary
5218	Smith	38 (*qv*)	75,000 (*qv*)
5218	Smith	35 (*qv*)	null (*qv*)
5218	Smith	35 (*qv*)	77,000 (*qv*)
5218	Smith	38 (*qv*)	null (*qv*)
5218	Smith	40 (*qv*)	null (*qv*)
5218	Smith	45 (*qv*)	77,000 (*qv*)
5218	Smith	60 (*qv*)	null (*qv*)
5218	Smith	57 (*qv*)	50,000 (*qv*)

Table 3. The original inconsistent data and decision data.

Value of global attribute "Age"	38	35	35	38	40	45	60	57
Candidate Data Sources / Data Source Quality Criteria vector	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
Time	68	56	28	37	95	43	81	75
Cost	49	85	76	29	68	56	12	32
Availability	91	72	96	55	43	84	24	65
Clearance	(1, 1, 2)	(9, 10, 10)	(7, 8, 9)	(4, 5, 6)	(2, 3, 4)	(7, 8, 9)	(2, 3, 4)	(4, 5, 6)
Reliability	(9, 10, 10)	(4, 5, 6)	(7, 8, 9)	(4, 5, 6)	(6, 7, 8)	(7, 8, 9)	(2, 3, 4)	(9, 10, 10)

In Table 2, the global attributes "Age" and "Salary" are inconsistent global attributes. "*qv*" represents the data source quality criteria vector of local class which provides the corresponding attribute value. For the consistent global attribute, "*qv*" can be ignored such as "ID" and "Name".

As described in Table 3, we take the inconsistent global attribute "Age" in Table 2 as an example to illustrate our solution strategy. The data source quality criteria vector value of each A_j in inconsistent local attribute set $\{A_j\}$ corresponding to "Age" is shown in Table 3.

In this example, the unit of each quality criterion for all candidates is the same and weight vector is w = (0.4, 0.2, 0.1, 0.2, 0.1).

(1) According to Table 3, fusion matrix $F_{8\times5}$ is obtained.

$$F_{8\times5}=\begin{bmatrix}(68,68,68) & (49,49,49) & (91,91,91) & (1,1,2) & (9,10,10)\\ (56,56,56) & (85,85,85) & (72,72,72) & (9,10,10) & (4,5,6)\\ (28,28,28) & (76,76,76) & (96,96,96) & (7,8,9) & (7,8,9)\\ (37,37,37) & (29,29,29) & (55,55,55) & (4,5,6) & (4,5,6)\\ (95,95,95) & (68,68,68) & (43,43,43) & (2,3,4) & (6,7,8)\\ (43,43,43) & (56,56,56) & (84,84,84) & (7,8,9) & (7,8,9)\\ (81,81,81) & (12,12,12) & (24,24,24) & (2,3,4) & (2,3,4)\\ (75,75,75) & (32,32,32) & (65,65,65) & (4,5,6) & (9,10,10)\end{bmatrix}.$$

(2) After scaling phase, the matrix $R_{8\times5}$ is obtained as follows,

$$R_{8\times5}=\begin{bmatrix}(0.716,0.716,0.716) & (0.576,0.576,0.576) & (0.948,0.948,0.948) & (0.1,0.1,0.222) & (0.9,1,1)\\ (0.589,0.589,0.589) & (1,1,1) & (0.75,0.75,0.75) & (0.9,1,1) & (0.4,0.5,0.667)\\ (0.295,0.295,0.295) & (0.894,0.894,0.894) & (1,1,1) & (0.7,0.8,1) & (0.7,0.8,1)\\ (0.389,0.389,0.389) & (0.341,0.341,0.341) & (0.573,0.573,0.573) & (0.4,0.5,0.667) & (0.4,0.5,0.667)\\ (1,1,1) & (0.8,0.8,0.8) & (0.448,0.448,0.448) & (0.2,0.3,0.444) & (0.6,0.7,0.889)\\ (0.453,0.453,0.453) & (0.659,0.659,0.659) & (0.875,0.875,0.875) & (0.7,0.8,1) & (0.7,0.8,1)\\ (0.853,0.853,0.853) & (0.141,0.141,0.141) & (0.25,0.25,0.25) & (0.2,0.3,0.444) & (0.2,0.3,0.444)\\ (0.789,0.789,0.789) & (0.376,0.376,0.376) & (0.677,0.677,0.677) & (0.4,0.5,0.667) & (0.9,1,1)\end{bmatrix}.$$

(3) After constructing fusion decision matrix based on weight vector, $D_{8\times5}$ and $V_{8\times5}$ are obtained.

$$D_{8\times5} = \begin{bmatrix} (0.286,0.286,0.286) & (0.115,0.115,0.115) & (0.095,0.095,0.095) & (0.02,0.02,0.044) & (0.09,0.1,0.1) \\ (0.236,0.236,0.236) & (0.2,0.2,0.2) & (0.075,0.075,0.075) & (0.18,0.2,0.2) & (0.04,0.05,0.067) \\ (0.118,0.118,0.118) & (0.179,0.179,0.179) & (0.1,0.1,0.1) & (0.14,0.16,0.2) & (0.07,0.08,0.1) \\ (0.156,0.156,0.156) & (0.068,0.068,0.068) & (0.057,0.057,0.057) & (0.08,0.1,0.133) & (0.04,0.05,0.067) \\ (0.4,0.4,0.4) & (0.16,0.16,0.16) & (0.045,0.045,0.045) & (0.04,0.06,0.088) & (0.06,0.07,0.089) \\ (0.182,0.182,0.182) & (0.132,0.132,0.132) & (0.088,0.088,0.088) & (0.14,0.16,0.2) & (0.07,0.08,0.1) \\ (0.342,0.342,0.342) & (0.028,0.028,0.028) & (0.025,0.025,0.025) & (0.04,0.06,0.089) & (0.02,0.03,0.044) \\ (0.316,0.316,0.316) & (0.075,0.075,0.075) & (0.068,0.068,0.068) & (0.08,0.1,0.133) & (0.09,0.1,0.1) \end{bmatrix}.$$

(4) According to the definitions of positive ideal solution and negative ideal solution, vector M^+ and M^- are obtained as follows,

M^+ = (0.4, 0.2, 0.1, 0.195, 0.098), M^- = (0.118, 0.028, 0.025, 0.026, 0.031).

Then we could obtain Euclidean distance l_i^+ and l_i^-, which are shown as follows,

$l_1^+ = 0.221$, $l_2^+ = 0.172$, $l_3^+ = 0.285$, $l_4^+ = 0.270$, $l_5^+ = 0.152$, $l_6^+ = 0.231$, $l_7^+ = 0.246$, $l_8^+ = 0.179$,
$l_1^- = 0.213$, $l_2^- = 0.274$, $l_3^- = 0.225$, $l_4^- = 0.102$, $l_5^- = 0.317$, $l_6^- = 0.179$, $l_7^- = 0.128$, $l_8^- = 0.241$.

(5) According to the definition of $\mu(v_i)$, the vector μ is obtained: μ = (0.482, 0.717, 0.384, 0.125, 0.813, 0.375, 0.213, 0.644).
According to μ, candidate data source s_5 is the final decision. Then "40" is selected as data inconsistency solution result for global attribute "Age".

Then we can use the similar way to resolve the data inconsistency of inconsistent global attribute "Salary" in Table 2 and we do not discuss it in detail here.

5. EXPERIMENTS

5.1 Experiment Setup

Data Sources

We design five testing distributed data sources in our experiments which provide their data source quality criteria with quantitative values and qualitative values. We integrate testing data sources with global schemas we have given based on the Expo Data Center system [17]. The purpose of introducing testing distributed data sources is to generate inconsistent data for testing queries. And we assume data sources with good quality provide good data.

Data Source Quality Criteria

There are two kinds of data source quality criteria – negative criteria and positive criteria. For negative criterion including cost and time, the higher the value is, the lower the quality is. On the contrary, for positive criterion including availability, clearance and reliability, the higher the value is, the higher the quality is. On the other hand, the quantitative criteria include time, cost and availability. The qualitative criteria include reliability and clearance.

The weight vector of data source quality criteria is provided according to the importance of data source quality criteria we designed. The values of quality criteria are generated randomly or evaluated according to the performance of data sources. And we use the value of b to represent a triangular fuzzy number $f = (m_L, m, m_R) = (a, b, c)$, if it is used in Figs. 1-6.

Queries

We use a query generator to generate queries based on global schema. Each generated query refers to data inconsistencies in data sources. For a query, polyobjects are detected according to the presetting query key.

Correctness Vector

To test the effectiveness of algorithms, we introduce a new vector − correctness vector in our experiments. The correctness vector is set to every set of inconsistent data in data sources to represent the real accurate degree of the data.

The correctness vector is the combination of elements in {0, 1, 2, 3, 4, 5} representing the correctness of 0%, 20%, 40%, 60%, 80%, 100% respectively. For example, the correctness vector of an inconsistent data set is like the form of (s_1:1, s_2:5, s_3:2, s_4:3, s_5:4) which means for instance, in this inconsistent data set the correctness of data provided by s_1 is 20%. After all data inconsistencies in query result R are resolved, we calculate the average correctness of data inconsistency solution for R which can represent the effectiveness of data inconsistency solution algorithms.

The correctness vectors are generated according to data quality we designed for testing data sources.

Comparison Strategies

In data integration, as there is no researches on data inconsistency solution being able to resolve data source properties with qualitative values, we adopt random and round robin strategies as comparison strategies in our experiments. Round Robin strategy resolves data inconsistencies by selecting data in inconsistent data set by turn according to data sources. Random strategy for data inconsistency solution chooses the data source's data randomly.

5.2 Experiments

The experimental results illustrating the average values of data inconsistency solution parameters are shown in Figs. 1-6 in which green line represents our strategy, the red line and the black line represent random strategy and round robin strategy respectively.

The average correctness of each strategy for data inconsistency solution is shown in

Fig. 1. The curve of our strategy naturally increases and it can reach almost 80% (4.0) which means our strategy can increase its average correct rate of data inconsistency solution as our algorithm runs and can obtain high average correctness. As it resolves more data inconsistencies, the curve flattens out as these resolving results provide less benefit to average correctness. The reason of this is because some times "good" data sources maybe provides "bad" data. This can be resolved by introducing mechanisms to help data sources to improve their "bad" data. Despite of these challenges, however, our strategy still can obtain high average correctness. On the other hand, the average correctness of random strategy is almost a constant – 60% (3) in data inconsistency solution since it treats each candidate data sources' data as equally important. And round robin strategy is a little better than random strategy in average correctness and is still much worse than our strategy. From Fig. 1, we can see that the average correctness of our strategy is much better than random and round robin strategies.

Figs. 2-4 show the average values of the selected data sources' data source quality criteria in data inconsistency solution. In Figs. 2-4, the average value of each positive data source criterion is the key component to the curves: the higher the average value of positive data source criterion is, the "better" data sources are. It is clearly shown that the curves of our strategy are much higher than the other two strategies' curves. This is be-

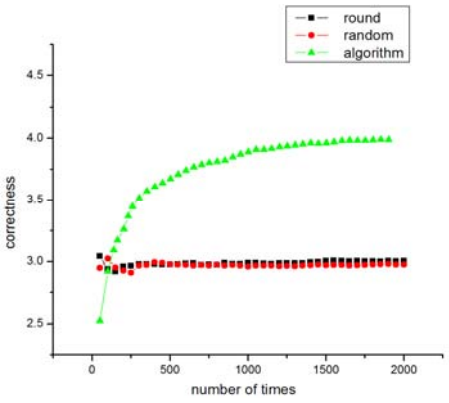

Fig. 1. Comparison of correctness.

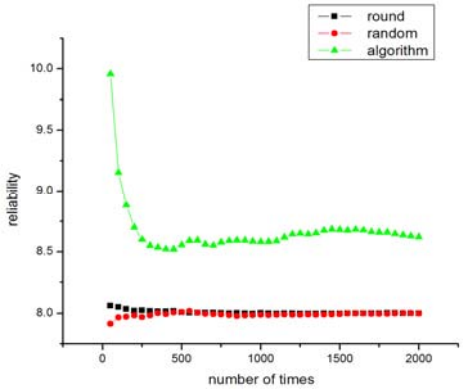

Fig. 2. Comparison of reliability.

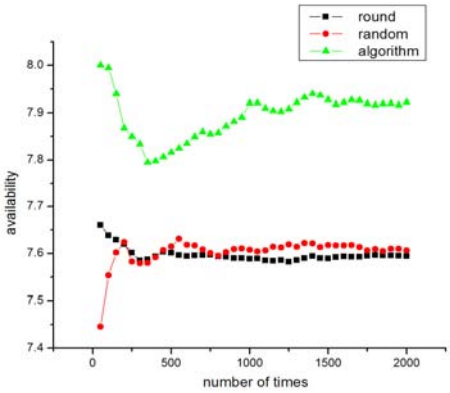

Fig. 3. Comparison of availability.

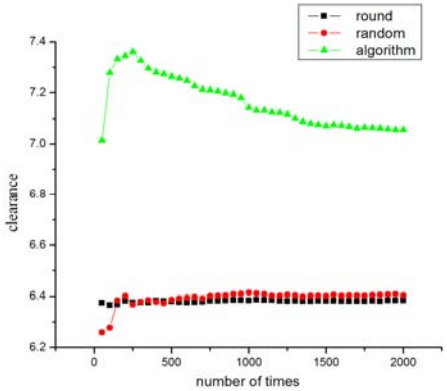

Fig. 4. Comparison of clearance.

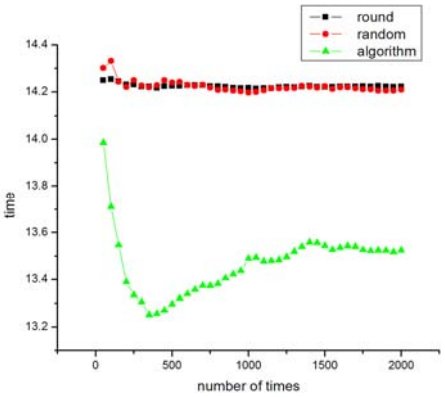

Fig. 5. Comparison of time.

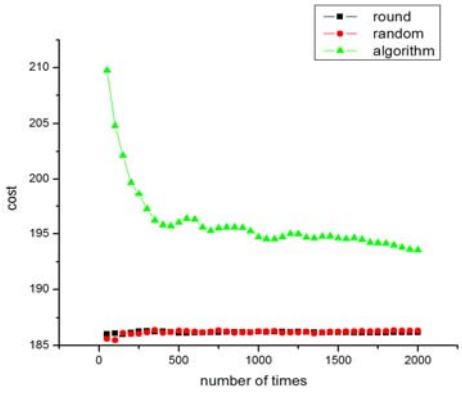

Fig. 6. Comparison of cost.

cause in our strategy we consider "good" data sources providing "good" data, and in data inconsistency solution, we always select "good" data sources. In Figs. 5 and 6, for negative data source quality criteria, the lower the average values of them are, the "better" the data sources are. In Fig. 5, the curve of our strategy is much lower than the other two strategies' curves. In Fig. 6, curve of our strategy is higher than the other two strategies' curves. This because, in our experiment the cost of data sources with average good quality have higher cost and this is common in reality. However, by using our strategy, the curve slowly decreases. The reason about it is that our strategy is making a balance in data sources' cost and average quality.

From the above discussion, we can see that our strategy in data inconsistency solution can obtain high average correctness of data inconsistency solution and the selected data sources have better average data quality.

6. CONCLUSION

In this paper, a solution for data inconsistency in data integration is proposed. The effectiveness of our solution is analyzed by experiments and the experiments confirm that our algorithm and strategy has ideal performance.

In the future, there are still some improvements to do, for example, we will look to improve the accuracy of data inconsistency solution.

REFERENCES

1. X. Chai, M. Sayyadian, A. Doan, A. Rosenthal, and L. Seligman, "Analyzing and revising data integration schemas to improve their matchability," in *Proceedings of the 34th International Conference on Very Large Data Bases*, 2008, pp. 773-784.
2. A. Motro and P. Anokhin, "Fusionplex: Resolution of data inconsistencies in the integration of heterogeneous data sources," *Information Fusion*, Vol. 7, 2006, pp. 176-196.
3. P. Anokhin, "Data inconsistency detection and resolution in the integration of het-

erogeneous information sources," Ph.D. Thesis, School of Information Technology and Engineering, George Mason University, 2001.
4. A. Y. Levy, A. Rajaraman, and J. J. Ordille, "Querying heterogeneous information sources using source descriptions," in *Proceedings of the 22nd International Conference on Very Large Data Bases*, 1996, pp. 251-262.
5. S. Agarwal, A. M. Keller, G. Wiederhold, and K. Saraswat, "Flexible relation: An approach for integrating data from multiple possibly inconsistent databases," in *Proceedings of the 11th International Conference on Data Engineering*, 1995, pp. 495-504.
6. M. A. H. Andez, S. J. Stolfo, and U. Fayyad, "Real-world data is dirty: Data cleansing and the merge/purge problem," *Data Mining and Knowledge Discovery*, Vol. 2, 1998, pp. 9-37.
7. E. Lim, J. Srivastava, and S. Shekhar, "Resolving attribute incompatibility in database integration: an evidential reasoning approach," in *Proceedings of the 10th International Conference on Data Engineering*, 1994, pp. 154-163.
8. F. S. C. Tseng, A. L. P. Chen, and W. P. Yang, "A probabilistic approach to query processing in heterogeneous database systems," in *Proceedings of the 2nd International Workshop on Research Issues on Data Engineering*: *Transaction and Query Processing*, 1992, pp. 176-183.
9. Y. Papakonstantinou, S. Abiteboul, and H. Garcia-Molina, "Object fusion in mediator systems," in *Proceedings of the 22nd International Conference on Very Large Data Bases*, 1996, pp. 413-424.
10. A. Motro, P. Anokhin, and A. C. Acar, "Utility-based resolution of data inconsistencies," in *Proceedings of the International Workshop on Information Quality in Information Systems*, 2004, pp. 35-43.
11. Z. Li, F. C. Yang, and S. Su, "Fuzzy multi-attribute decision making-based algorithm for semantic web service composition," *Journal of Software*, Vol. 20, 2009, pp. 583-596.
12. H. Tong and S. Zhang, "A fuzzy multi-attribute decision making algorithm for web services selection based on QoS," in *Proceedings of IEEE Asia-Pacific Conference on Services Computing*, 2006, pp. 51-57.
13. R. Y. Wang and D. M. Strong, "Beyond accuracy: What data quality means to data consumers," *Journal of Management Information Systems*, Vol. 12, 1996, pp. 5-30.
14. J. M. Benítez, J. C. Martín, and C. Román, "Using fuzzy number for measuring quality of service in the hotel industry," *Tourism Management*, Vol. 28, 2007, pp. 544-555.
15. L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning," *Information Science*, Vol. 8, 1975, pp. 199-249.
16. F. E. Uzoka, "A fuzzy-enhanced multicriteria decision analysis model for evaluating university academics research output," *Information Knowledge Systems Management*, Vol. 7, 2008, pp. 273-299.
17. X. Wang, L. P. Huang, and Y. Zhang, "A grid middleware – DISQ for data integration," in *Proceedings of International Conference on Computer Science and Software Engineering*, 2008, pp. 62-65.

Xin Wang (王欣) received the B.S. and M.S. degrees in Computer Science from Shandong University. She is currently working on towards Ph.D. degree in Computer Science at Computer Science and Engineering Department of Shanghai Jiao Tong University, Shanghai, P.R. China. Her research interests include data integration and data uncertainty.

Lin-Peng Huang (黄林鹏) as Professor of Computer Science and Engineering Department of Shanghai Jiao Tong University. His general interests include distributed computing, service computing and program language.

Xiao-Hui Xu (徐小辉) received the B.S. and M.S. degrees in Computer Science from Chongqing University. He is currently working on towards Ph.D. degree in Computer Science at Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, P.R. China. His research interests include service computing and software reliability.

Yi Zhang (章义) received the B.S. and M.S. degrees in Computer Science from Shandong University. He is currently working on towards Ph.D. degree in Computer Science at Computer Science and Engineering Department of Shanghai Jiao Tong University, Shanghai, P.R. China. His research interests include data integration and distributed computing.

Jun-Qing Chen (陈俊清) received his B.S. and M.S. degrees in Computer Science from Fuzhou University. He is currently working on towards Ph.D. degree in Computer Science at Computer Science and Engineering Department of Shanghai Jiao Tong University, Shanghai, China. His current research interests include type and effect system, formal analysis and verification service composition and dynamic service update in OSGi.