International Journal of New Computer Architectures and their Applications (IJNCAA) 6(2): 65-72
The Society of Digital Information and Wireless Communications, 2016 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

Inconsistency Resolution In The Virtual Database Environment Using Fuzzy Logic

Amr Abdelrahman, Ali El-Bastawissy, Mohamed Kholief
Computing & Information Technology Department, AAST
Cairo, Egypt
E-mail: amrmnabil@yahoo.com, alibasta@hotmail.com, kholief@gmail.com

ABSTRACT

Data integration from different data sources may result in data inconsistencies due to different representation of the same objects at the data source. Many researchers have tried to solve this problem manually or using source features. None of them took the user's preferences to source features into account. This paper proposes using fuzzy logic with multiple constraints, in accordance with user preference, to resolve inconsistencies. This approach uses token-based cleaner, a content based inconsistency detection algorithm, to detect inconsistencies. Then, uses fuzzy logic to resolve inconsistencies. An experiment was conducted using our fuzzy algorithm on a trained dataset that reflects our designated point of view. The result indicates that multiple constraints decision making is a suitable technique for resolving inconsistencies.

KEYWORDS

Data warehouses; Data analysis; Data integration; Fuzzy logic; Data processing.

1 INTRODUCTION

When trying to connect information sources that were developed independently, even if they are intended to be used in the same domain and hold the same kind of data, sources differ in many ways, this problem is called the Heterogeneity Problem. Data integration is the process of taking several databases or other information sources and making the data in these sources work together as if they were a single database [1].

Data inconsistencies are bound to appear, as a result of the heterogeneity problem. Not only does the detection of the records that represent the same object is needed, but also, records that make reference to the same entity are combined by fusing them into one representation and resolving any conflicts from different data sources [15].

There could be two levels of inconsistencies [1]:

- Schema Level Inconsistency due to the different data models, or naming, such as one system uses gender and the other uses sex, and structure conflicts. Structure conflicts refer to, within different sources, the same object exists in different representations, within the same domain.

- Instance level inconsistencies happen because data in the sources may be expressed in several natural languages, different ways of measurement systems, in other cases, there are discrepancies in the facts among the sources in data values that express the objects [2]. Resolving conflicts on the Instance level is a major activity in systems that integrate data. No system that integrates data can return a consistent answer to a user's queries if these conflicting data are not fixed. Conflicts on the instance level have received significantly much less attention, than conflicts on the schema level, and only recently has the importance of these types of conflicts increased, because of the major role they play in the procedure of integration of data. Instance-level conflicts are credited to low quality of data, problems in the data collection procedure, the data entry procedure, or because sources are not kept up to date [1].

Data sources have their different characteristics and quality criteria (Timeliness, Cost, Accessibility, Accuracy...etc.). Data sources vary in recency; some data result from more accurate sources than others. Integrating data from different data sources will yield data inconsistency. To be able to resolve inconsistencies in data, it is sensible to check the provider's data qualifications and, the user's preference to quality criteria. Some users desire a lower timeliness or an increased accessibility, according to their needs [2].

In this paper, it is suggested to use data source quality measures together with user preference to resolve data inconsistencies caused by data integration. Fig. 1 shows the framework for resolving

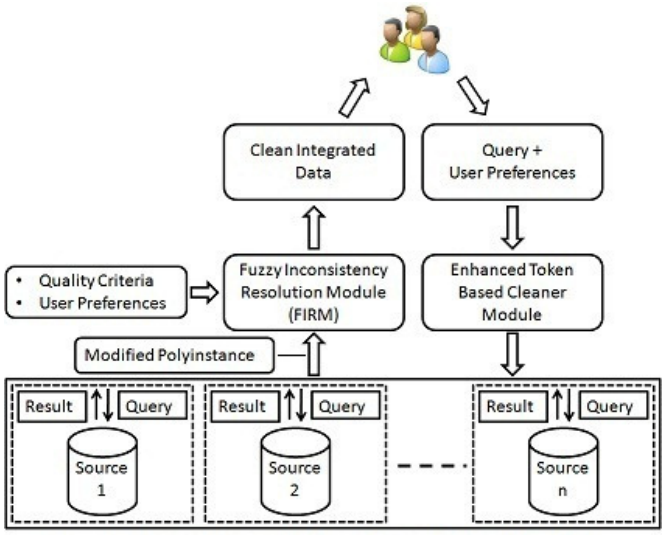

Fig. 1 Fuzzy Inconsistency Resolution Module

inconsistencies. The user poses the query to the enhanced token-based cleaner module, it queries the data sources, and returns a dataset called the Polyinstance. It is then, processed by the Fuzzy Inconsistency Resolution Module (FIRM) which, in turn, returns a clean dataset to the user. In our methodology, data source quality criteria and user preferences are used as the bases to a fuzzy multi-constraints decision making procedure to select the "best" data source's data as the solution the inconsistency. The results of the experiment indicates that the performance of the algorithm is ideal.

Inconsistency resolution procedure involves choosing one source for a data item from a universe of sources a set of constraints or a quality vector. Evaluation of how well does each source satisfy each quality attribute and combine the weighted quality attributes into the source selection procedure to get the best answer is needed.

The organization of the paper is as follows. Section 2 gives a review of the inconsistency detection and resolution trails in literature. Section 3 describes the fusion algorithm. Section 4 justifies the outcomes of the experiment and gives an evaluation of the algorithm. Section 5 concludes the contribution of the paper.

2 APPROACHES TO INCONSISTENCY RESOLUTION

Data integration and inconsistency handling techniques are addressed in many researches. Object fusion in mediator systems [8] is a data integration system with inconsistency resolution module where precision degrees of the sources can be considered when defining the mappings. It means that one source is preferred over the other in case of conflicts, and, no assignment for the different qualifications of the participating data sources (Accuracy, Clearance, and Etc.). This technique has a limited amount of mappings between the participating data sources, and the global schema. It avoids dealing with inconsistencies where it takes a pre-determined decision about how to deal with conflicts when encountered. This decision is based on taking the data from the preferred source. HUMMER [9, 11] is a data Integration system that integrates heterogeneous sources into a single, and consistent view. It consists of a three-step process: Schema mapping, Duplicate detection, and data fusion. However, it considers all the participating data sources to have the same qualifications. Multiplex [7, 10, 12] provides an approximation of the true answer by sandwiching it between a lower bound set of accurate but incomplete answers, and, an upper bound set of complete but, not accurate answers. It had many limitations. It cannot identify the same object as one if it is described slightly differently and, assumes that all sources have the same qualifications. Fusionplex [7, 16] assumes the Multi database model offered in [10] with some extensions. It tries to avoid the defined limitations of that model, where it applies pre-defined conflict resolution policies defined with the global schema based on the quality of data sources and other quality parameters provided by the user to answer user's queries. For each global schema attribute (construct), it defines a policy and, uses these policies when facing conflicts. However, it is only valid for a federated database, all columns of the data source inherit the source's features, and fuses closely semantically related attributes separately. It does not deal with the imprecision of established pre-defined conflict resolution policies.

The literature suggests identifying objects based on its content, then, resolving any inconsistencies using quality measures of the data source. The inconsistency resolution algorithm needs to consider the imprecision of the data source quality measures, and the Imprecision in decision making (choosing which data source to use) which the fuzzy set theory can be used to model and, define the criteria and the importance of the criteria.

The Enhanced Token-Based Cleaner [1] proposed a technique for inconsistency detection and resolution based on what the data sources contain. It is a suitable algorithm to detect inconsistent data. It stores in a repository called fusion metadata the key fields which are chosen by the user and other data which aids the technique to create a token used to identify inconsistent duplicate data. It contains three modules: data integration module, inconsistency detection module, and inconsistency resolution module. Data integration module and the inconsistency detection module are used to detect the records representing the same real-world object. Then, the inconsistency resolution module fuses attributes based on a set of quality measures. These quality measures are set by the system designer not calculated and, it does not include the user preference for some system quality measures over the others.

The proposed extension to this method bears the user preference in mind. Thus, giving a more desirable answer to the posed query. A user may pose a query with the cost in mind over accuracy or Timeliness over Accessibility.

There is a problem with the accuracy and the imprecise measurements of the source and attribute quality, and the imprecise entry of the user's preferences making the use of fuzzy logic a very beneficial since fuzzy logic is tolerant of imprecise data.

Fig. 2 shows that the Enhanced Token-Based cleaner contains three modules [1]:

1. Data integration module: to define the data integration environment.

2. Inconsistency detection module: to detect duplicates

3. Inconsistency resolution module: to resolve the detected in the inconsistency detection module.

After the data integration module is finished, the result is a constructed dataset, Polyinstance that has the required fields, and detectors. The Enhanced Token-Based duplicate detection algorithm, where the authors described an algorithm that detects duplicates, where the defined algorithm detects duplicates based on a set of user chosen fields which have the highest confidence to define most uniquely identify the duplicates. The Inconsistency Resolution

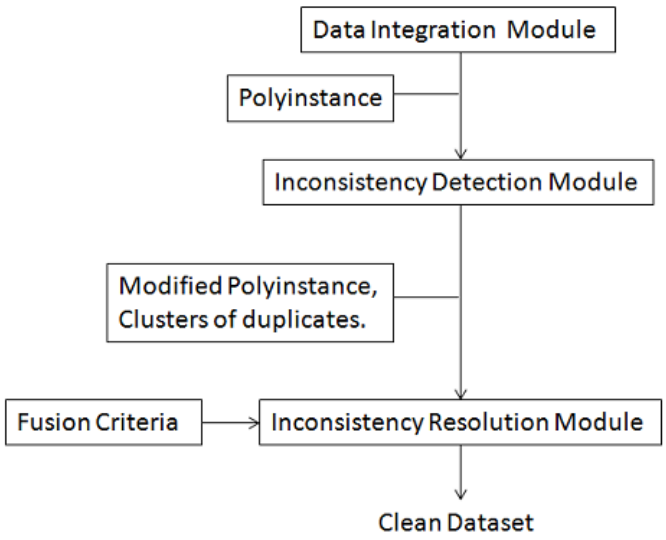

Fig. 2 Token Based Cleaner

Module follows these steps to resolve inconsistencies:

1. The input of this module is the detected duplicates with the defined surrogate key.

2. Cluster the Polyinstance to a set of polytuples, each polytuple is a set of records detected to represent the same object with the defined surrogate key for these records.

3. Split each polytuple to a set of mono-attributes, each mono-attribute is the cluster key, plus one attribute from the polytuple, if this attribute is semantically dependent on another attribute then add this attribute to this mono-attribute to become a cluster key and dependent attributes.

4. Using the mono-attributes fuse each attribute into one value using Fuzzy Logic and attribute preferences.

5. Join the mono-attributes using the defined key.

Repeat steps 3, till 5 with each polytuple.

3 FUZZY INCONSISTENCY RESOLUTION MODULE (FIRM)

A user poses a query and his preferences to the source quality criteria to the Enhanced token-based cleaner. It receives the query, integrates the data, detects duplicates, and returns a Polyinstance containing the detected duplicates and their source. Passes the Polyinstance to the Fuzzy Inconsistency

Resolution Module to get a clean, and integrated dataset.

A. Multiple Constraint Decision Making

In nearly every real life problem uncertainty is present. Generally, measurements causes uncertainty. The boundaries of ways of measuring with instruments and unavoidable problems in way of measuring causes it. In many problems, from the confusion and lack of clarity rooted in our languages uncertainty emerges. The theory of fuzzy sets was presented by Prof. Lotfi A. Zadeha in seminal paper in 1965. With the imprecise boundaries of these sets. Fuzzy rule bases are "a series of fuzzy operators to fuzzify, aggregate and then map the input membership values onto an output domain, and then aggregates and defuzzify the membership values of the resultant variables" [6, 13, 14].

In the procedure of choosing a data source with a consistent answer, multiple objectives constraints the decision process and the importance of each one of these objectives is different. Now, consider that in the choice process maximizing the accuracy and minimizing cost is required. A user may give a higher weight for accuracy emphasizing its importance over the cost. Two main problems with multi-objective decision making are to obtain information that is meaningful about the satisfaction of the objectives by the many options (data sources) and also to weight the relative influence of each one of the options to the decision maker.

An algorithm to choose a consistent data source from a set of data sources has been described by T.J.Ross [3].

B. System Inputs

The inconsistency detection module constructs a Polyinstance containing detected duplicates with the defined surrogate key, (Table 1), together with the source features which are stored in the fusion metadata (Table 7).

1) Example

Split the Polyinstance (Table 1) into clusters of duplicates (Table 2). Then split the clustered Polyinstance further into mono-attributes based on the semantic dependency.

- Mono attribute 1 (Table 3): This mono attribute constructed only from MName, which is not semantically dependent on any of other attributes.
- Mono attribute 2 (Table 4): this mono attribute is constructed from MAddress and ZipCode where both attributes are semantically dependent on each other.
- Mono attribute 3 (Table 5): this mono attribute is constructed from DoB.

2) Measuring Source Quality

Different areas have been discussed data quality, like computer science, statistics, and management. Choosing quality measures to distinguish between the quality of data sources is the beginning of any activity related with data quality. In this research four quality measures are chosen, as a sample for the quality measures that can be used:

a) Accuracy[4]

Measures the similarity in a value v to a value v', that is the real value of an object that v aims to represent.

b) Timeliness[4]

Concerns how current the data is. If an owner of a car is up to date, meaning that, the car registration corresponds to the current owner of the car, then the timeliness is high.

c) Cost[5]

The cost is a measure of how much money will be paid in case of deciding on using a specific data source.

d) Accessibility[5]

The accessibility dimension reflects the ease of attainability of the data.

(Table 7) shows a sample source features metadata.

C. FIRM Algorithm

Define n data sources, $A = \{a_1, a_2, ..., a_n\}$, and r objectives, $O = \{O_1, O_2, ..., O_r\}$. Let O_i be objective number i. Then, the membership degree of an alternative, a, in O_i, $\mu O_i(a)$, is the degree to which an alternative a meets the expectations the criteria for the objective. Consider a set {P} constrain to being linear and ordinal, be a set of preferences. The decision function is given by associating each objective with a weight expressing its influence to the maker of the decision. This function is expressed by intersecting r-tuples, known as the decision measure, M(Oi,bi), involving preferences and objectives.

$$D = M(O_1, b_1) \cap M(O_2, b_2) \cap ..... \cap M(O_r, b_r) \qquad (1)$$

For any alternative a classical inclusion can replace the decision measure, a, on the form

$$M(O_i(a), b_i) = b_i \rightarrow O_i(a) = \overline{b_i} \vee O_i(a) \qquad (2)$$

Table 1: Polyinstance

TupleNo	MID	MName	Maddress	ZipCode	MSalary	DoB	Source
1	256	Mohamed	Cairo	12346	2000	10/12/1980	S1
2	256	Mohamed	Fayoum	12348	2000	NULL	S2
3	256	Ahmed	Giza	12346	3000	10/10/1985	S3
4	511	Karim	Giza	12346	3000	NULL	S1
5	511	Karim	Alex	23456	4000	27/8/1960	S2
6	511	Karim	Alex	23456	5000	NULL	S3

Table 2: Clustered Polyinstance

Cluster 1 Polytuple 1							
TupleNo	MID	MName	Maddress	ZipCode	MSalary	DoB	Source
1	256	Mohamed	Cairo	12346	2000	10/12/1980	S1
2	256	Mohamed	Fayoum	12348	2000	NULL	S2
3	256	Ahmed	Giza	12346	3000	10/10/1985	S3
Cluster 2 Polytuple 2							
TupleNo	MID	MName	Maddress	ZipCode	MSalary	DoB	Source
4	511	Karim	Giza	12346	3000	NULL	S1
5	511	Karim	Alex	23456	4000	27/8/1960	S2
6	511	Karim	Alex	23456	5000	NULL	S3

Table 3: Mono Tuple 1 MName

Cluster#	MName	Source
1	Mohamed	S1
1	Mohamed	S2
1	Ahmed	S3

Table 4: Mono Tuple 2 MAddress, ZipCode

Cluster#	MAddress	ZipCode	Source
2	Cairo	12346	S1
2	Fayoum	12348	S2
2	Giza	12346	S3

Table 5: Mono Tuple 3 DoB

Cluster#	DoB	Source
3	10/12/1980	S1
3	NULL	S2
3	10/10/1985	S3

The joint intersection of r decision measures will be a sensible decision model,

$$D = \bigcap_{i=1}^{r} (\bar{b}_i \cup O_i) \qquad (3)$$

The optimum solution, a—, is the alternative that maximizes D. define;

$$Ci = (\bar{b}_i \cap O_i)$$

Table 6: Fusion Criteria Granularity

Metadata Name	Granularity
Timeliness	attribute
Accuracy	attribute
Cost	source
Availability	source

Table 7: Source Features Metadata

Source	Accessibility	Cost
S1	0.2	0.7
S2	0.4	1.0
S3	1.0	0.4
MName	Timeliness	Accuracy
S1	0.4	1.0
S2	1.0	0.5
S3	0.1	0.5
MAddress—ZipCode	Timeliness	Accuracy
S1	0.4	1.0
S2	1.0	0.5
S3	0.1	0.5
DoB	Timeliness	Accuracy
S1	0.4	1.0
S2	1.0	0.5
S3	0.1	0.5

, hence $\mu_{C_i}(a) = max[\mu_{\bar{b}_i}(a), \mu_{O_i}(a)]$ (4)

Then, the optimum solution, expressed in a membership form, is given by

$$\mu_D(a^*) = max_{a\in A}[min\{\mu_{C_1}(a), \mu_{C_1}(a), ..., \mu_{C_i}(a)\}] (5)$$

If a numerical tie occurs among two or more alternatives. A special procedure shall be followed. If x and y, are two alternatives, and their decision values are equal, that is;

D(x) = D(y) = $max_{a\in A}[D(a)]$, where a = x = y.

Since D(a) = $min_i[Ci(a)]$, there exists some alternative k such that, $C_k(x) = D(x)$ and some alternative g such that $C_g(y) = D(y)$. Let

$$\hat{D}(x) = min_{i\neq k}[C_i(x)] \; and \; \hat{D}(y) = min_{i\neq g}[C_i(y)]. (6)$$

Then, match $\hat{D}(x)$ and $\hat{D}(y)$. If, $\hat{D}(x) > \hat{D}(y)$, select x as our optimum solution. If a tie still persists, that is, $\hat{D}(x) = \hat{D}(y)$, then, there exist some other alternatives j and h such that $\hat{D}(x) = C_j(x) = \hat{D}(y) = C_h(y)$. Then, formulate;

$$\hat{\hat{D}}(x) = min_{i\neq k,j}[C_i(x)] \; and \; \hat{\hat{D}}(y) = min_{i\neq g,h}[C_i(y)] (7)$$

Now, match $\hat{\hat{D}}(x) and \hat{\hat{D}}(y)$. Until an optimum alternative emerges or all of the alternatives have been tested the tie-breaking procedure continues in this way. If there is still a tie, another tie-breaking procedure, such as a refining the preference scales, should be used.

Example

Measuring our sources features (Table 7) resulted. Now, formulate the equations:

For attribute DoB A = {a1, a2, a3}
O = {O1, O2, O3, O4} = {Time, Cost, Accessibility, Accuracy}
P = {p1, p2, p3, p4}

$$O_1 = \{\frac{0.4}{S_1} + \frac{1.0}{S_2} + \frac{0.1}{S_3}\}$$
$$O_2 = \{\frac{0.7}{S_1} + \frac{0.8}{S_2} + \frac{0.4}{S_3}\}$$
$$O_3 = \{\frac{0.2}{S_1} + \frac{0.4}{S_2} + \frac{1.0}{S_3}\}$$
$$O_4 = \{\frac{1.0}{S_1} + \frac{0.5}{S_2} + \frac{0.5}{S_3}\}$$

These ratings are expressed in Zadeh's notations

For attribute MName A = {a1, a2, a3}
O = {O1, O2, O3, O4} = {Time, Cost, Accessibility, Accuracy}
P = {p1, p2, p3, p4}

$$O_1 = \{\frac{0.4}{S_1} + \frac{1.0}{S_2} + \frac{0.1}{S_3}\}$$
$$O_2 = \{\frac{0.7}{S_1} + \frac{0.8}{S_2} + \frac{0.4}{S_3}\}$$
$$O_3 = \{\frac{0.2}{S_1} + \frac{0.4}{S_2} + \frac{1.0}{S_3}\}$$
$$O_4 = \{\frac{1.0}{S_1} + \frac{0.5}{S_2} + \frac{0.5}{S_3}\}$$

These ratings are expressed in Zadeh's notations

For attribute MAddress/ZipCode A = {a1, a2, a3}
O = {O1, O2, O3, O4} = {Time, Cost, Accessibility, Accuracy}
P = {p1, p2, p3, p4}

$$O_1 = \{\frac{0.5}{S_1} + \frac{0.6}{S_2} + \frac{0.7}{S_3}\}$$
$$O_2 = \{\frac{0.7}{S_1} + \frac{0.8}{S_2} + \frac{0.4}{S_3}\}$$
$$O_3 = \{\frac{0.2}{S_1} + \frac{0.4}{S_2} + \frac{1.0}{S_3}\}$$
$$O_4 = \{\frac{0.9}{S_1} + \frac{0.8}{S_2} + \frac{0.7}{S_3}\}$$

These ratings are expressed in Zadeh's notations

- **Test Case 1**

The first set of preferences gives high rank for the least cost $p_1 = 0.8$, $p_2 = 0.9$, $p_3 = 0.7$, $p_4 = 0.5$

Results

MName	MAddress	ZipCode	DoB
Mohamed	Giza	12346	NULL

- **Test Case 2**

The second set of preferences gives high rank for accuracy $p_1 = 0.4$, $p_2 = 0.5$, $p_3 = 0.7$, $p_4 = 0.9$

Results

MName	MAddress	ZipCode	DoB
Ahmed	Giza	12346	10/10/1985

- **Test Case 3**

The third set of preferences gives high rank for accessibility $p_1 = 0.5$, $p_2 = 0.7$, $p_3 = 0.8$, $p_4 = 0.7$

Results

MName	MAddress	ZipCode	DoB
Ahmed	Giza	12346	10/10/1985

JUSTIFICATION AND EVALUATION

The justification for this model goes in the following way. When the quality measure number i becomes vita in the decision process, its preference p_i increases, in turn decreases its complement $\bar{p}_i$, which causes the QM associated with a weight expressing its vitality, $C_i(a)$, to decrease, hence, increasing the possibility that the QM of a particular source $C_i(a) = O_i(a)$, where now $O_i(a)$ will be the value of the decision function, D, representing an alternative a. When this process is repeated for all other data sources, a. It became known that the greatest value $O_i(a)$ for other alternatives will result in choosing of the optimum solution, a*. This is how the process works.

In the original Inconsistency resolution module in the Enhanced token-based cleaner, the author specified where each field is preferably acquired. Addition of new sources is a hard task where the source preference of the whole system must be evaluated after adding each source, there is no way to include a user's preference for the measured quality vector, to get a more desirable and resilient answer to his query.

A data source is chosen as a source of reference, created three copies of the same database, and induced errors in the copies, thus, errors in the data can be calculated. Quality measures to evaluate the quality of each source are used. To test the effectiveness of the algorithm, the effectiveness is defined as:

$$Effectiveness = \frac{Number\ Of\ Correct\ Fields}{Total\ Number\ Of\ Fields}$$

The algorithm it tested using the user's preference maximum value, a perfect answer is obtained that typically matches the original data. Then, the method is tested using different user's preferences and calculate the effectiveness of the algorithm. It is evident that the higher the user's preferences are, the more effective is the algorithm. Some quality measures directly affect the effectiveness of the

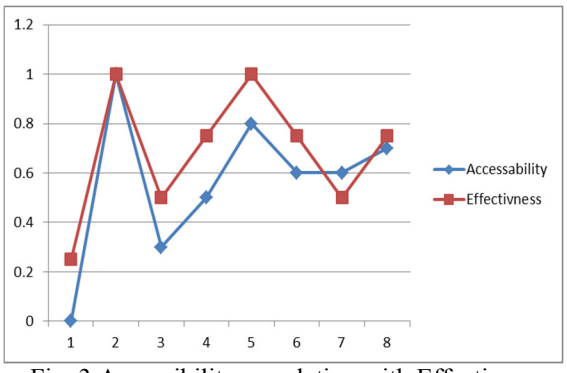

Fig. 3 Accessibility correlation with Effectiveness

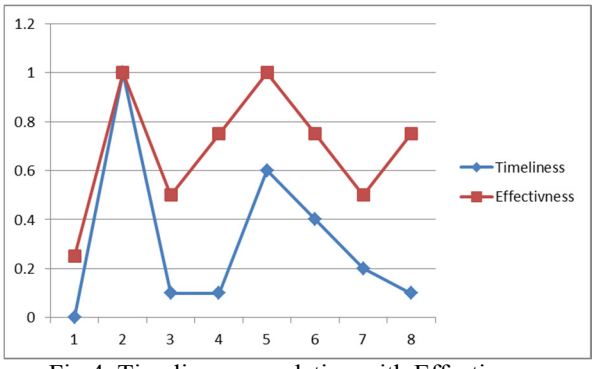

Fig.4: Timeliness correlation with Effectiveness

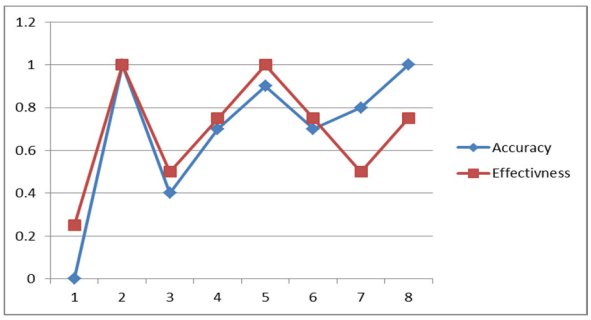

Fig.5: Accuracy correlation with Effectiveness

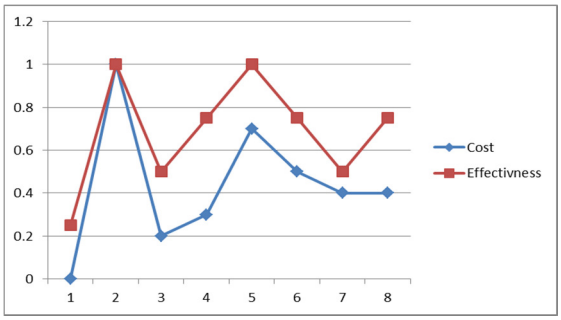

Fig.6: Cost correlation with Effectiveness

algorithm like accuracy. Fig.5 shows that accuracy directly affects the effectiveness of FIRM. While others have little effect on the effectiveness of the algorithm such as timeliness. Fig. 4 shows little effect of timeliness on the effectiveness. If the user preferences are equal and are low in value, the algorithm generates error values.

This algorithm depends on a quality vector (Timeliness, Cost, Accuracy, and Accessibility) when applied to inconsistent data its range of effectiveness is between 50% - 100% depending on the user choice of quality measure preferences.

CONCLUSION

This paper addresses a new perspective about solving the data fusion problems. Making use of meta-data called quality measures of the data sources. An extension to the relational data model is proposed. A process of data integration is presented, it consists of three modules: (1) data integration module, (2) inconsistency detection module and (3) inconsistency resolution module.

In the Inconsistency resolution module a flexible conflict resolution fuzzy logic algorithm that uses source quality measures guided by user's preferences to get an adequate clean dataset was proposed.

Directions for future work could possibly include automatically generating conflict resolution procedure according to the particular properties being used, discovery of relevant sources automatically for integration into the virtual database, dealing with sources with heterogeneous properties (different properties for the different source parts).

REFERENCES

[1] A. El Qutaany, O. Hegazy, A. El-Bastawissy "Data Cleansing in the Virtual Integration Environment", Informatics and Systems (INFOS), 2010 The 7th International Conference, 104-139 (2010).

[2] X. Wang, L. Huang, X. Xu, Y. ZHANG AND J. CHEN "A Solution for Data Inconsistency in Data Integration", Journal Of Information Science and Engineering Vol.27, 681-695 (2011).

[3] T. J. Ross "Fuzzy Logic with Engineering Applications, Third Edition", Wiley, 289-294 (2010).

[4] C. Batini, M. Scannapieca, "Data Quality Dimensions" in Data Quality Concepts, Methodologies and Techniques, Springer, 36-66 (2006).

[5] Y. W. Lee, L. L. Pipino, J. D. Funk, R. Y. Wang, "Cost Benefit Analysis" in A journey to data quality, The MIT Press, 28-42 (2006).

[6] A. Celikyilmaz, and I. B. Turksen, "Introduction" in Modeling Uncertainty with Fuzzy Logic, Springer, 47-56 (2009).

[7] P. Anokhin, and A. Motro , "Fusionplex: Resolution of Data Inconsistencies in the Integration of Heterogeneous Information Sources", Sience Direct Information Fusion Vol. 7, Issue 2, 176-196 (2007).

[8] Y. Papakonstantinou, S. Abiteboul, and H. Garcia-Molina. "Object fusion in mediator systems". Proc. of the 22nd Int. Conf. on Very Large Data Bases (VLDB‟96) , 413–424 (1996).

[9] F. Naumann, A. Bilke, J. Beliholder, and M. Weis." Data fusion in three steps: Resolving schema, tuple, and value inconsistencies". IEEE Data Eng. Bull., 21–31 (2006).

[10] A. Motro. "Multiplex: A Formal Model for Multidatabases and Its Implementation". In Proceedings of NGITS-99, 4th International Workshop on Next Generation Information Technologies and Systems. Lecture Notes in Computer Science, Volume1649, Springer, 138–158 (1999).

[11] A. Bilke, F. Naumann, J. Bleiholder, C. Bohm, and K. Draba, "Automatic Data Fusion with HumMer", Proceedings of the 31st VLDB Conference, Trondheim, Norway, 1251-1254 (2005).

[12] J. Berlin and A. Motro. "Autoplex: Automated Discovery of Content for Virtual Databases", CooplS '01 Proceedings of the 9th International Conference on Cooperative Information Systems, Vol. 2172, 108-122 (2001).

[13] S. Opricovic, "Fuzzy VIKOR with an application to water resources planning", Expert Systems with Applications, Vol. 38, Issue 10, 12983–12990 (2011).

[14] H. Ren and H. Zhou, "Triangular Fuzzy Multi-attribute Decision Making Based on Risk Attitude of Decision Maker", International Journal of Hybrid Information Technology Vol.8, No.7, 161-168 (2015).

[15] A. M. saeed, "Data Integration in Multi-sources Information Systems", International Journal of Computational Engineering Research (IJCER), Vol. 5, Issue 1, 63-69 (2015).

[16] I. Carol and S. Britto Ramesh Kumar, "Conflict Identification and Resolution in Heterogeneous Datasets: A Comprehensive Survey", International Journal of Computer Applications, Vol. 113, No. 12, 22-27 (2015).