A Traceable Data Fusion Based on Data Provenance

Zhao Qiang[1], Zhang Yongxin[1,2,*], Wang Dequan[3] and Ding Yanhui[4]

[1]School of Mathematical Sciences, Shandong Normal University, Jinan 250014, China; [2]Shandong Provincial Key Laboratory of Software Engineering, Jinan, China; [3]School of Computer Science, Fudan University, Shanghai 200433, China; [4]School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China

Abstract: Data fusion is a hot topic in data integration which at least includes the two stages: entity resolution and data conflict resolution. However, the existing fusion process is transparent and the fusion stages are isolated. So in this paper, we proposed a traceable data fusion mechanism based on data provenance which can trace the data sources of fusion results and the evolutionary process. The mechanism mainly targets forwards entity resolution and data conflict resolution stage. We represented the provenance of data origin using PI-CS which is more accurate because PI-CS can record the intermediate process of data evolution. In order to record the evolution process of data fusion, we proposed two transformation provenances: entity resolution provenance and data conflict resolution provenance which record respectively the evolution process of entity resolution and data conflict resolution. Finally, we give an example to validate the availability of the traceable mechanism for data fusion.

Keywords: Data conflict resolution, data fusion, data provenance, entity resolution.

1. INTRODUCTION

In recent years, as the rapid development of internet, the scale of web data is expanding, the shared scope broader and the data forms have considerable variety. The web pages on the internet are scaling up at dramatic rates and the data creates a huge, complex and heterogeneous data environment. The 23rd statistical report of internet development and state in China reported by CNNIC shows that by the end of 2008, the total number of web pages in China is more than 16 billion and its scales rise 90% compared with the previous year.

The traditional data integration methods are more concerned with the target integrated data and ignore the intermediate process of data integration including data fusion [1]. However, the major causes impacting the quality of integrated data are the varied data forms, rich levels, multiple data sources and varying quality of intermediate dataset of intermediate data. In web data integration, data fusion is the quality assurance of integrated data and by the process of entity resolution and data conflict resolution, the integrated data can be effectively eliminate redundancy and distinguished true value from fraud. It not only provides the panoramic view about the target objects for end users, but also provides high-quality data support for further data analysis and mining.

The traditional data integration technologies focus on the form of integrated data and the emphasis of management is the data format. However, if we need to share large-scale heterogeneous web data, the traditional database technologies are showing their wrinkles on data integration. Especially for data fusion, it falls short in several aspects, including the interpretability of fused data and the debugging ability of the data fusion process. On the one hand, data fusion as a whole is treated as a black box for users in the process of data fusion. The users cannot understand the source of the data and trace the evolutionary process of data, and it makes data fusion lack interpretability and traceability. On the other hand, for some practical systems which require very high-quality integrated data, it needs proper manual intervention in the process of data fusion to improve the data quality. But, since it is difficult to debug and trace the process for current data fusion, the manual intervention is hard to add in. Therefore, during the process of data fusion, it is very important to analyze the process of data origin and evolution, and then evaluate the data quality and accuracy, and revise the results. And it is a challenge in the research of data fusion to provide the traceable mechanism for data fusion.

Data provenance refers to the chronology of the ownership, custody or location of a historical object [2] and it can provide effective support for traceable mechanism of data fusion. The current researches of data provenance on data integration mostly concentrate in schema mapping. Most recently, some scholars began to explorer the implementation of data provenance on data conflict resolution. So far, the research on the traceable mechanism of web data fusion as a whole is not found.

For the characteristics that the data fusion process is transparent and the fusion stage is isolated, we proposed a traceable mechanism of data fusion based on data provenance in order to make the fusion results more explainable and make the fusion process more debuggable. The mechanism is mainly for two stage including entity resolution and data conflict resolution, and it enables users to trace the data origin of fusion results and evolution process of entity reso-

1874-110X/14 2014 Bentham Open

lution and conflict resolution. In Summary, we make the following three contributions:

(1) For the representation of data provenance information, we employ PI-CS [3] to express where the data came from. Because PI-CS can record the intermediate process of data evolution, it produces precise provenance for outer join and union in contrast to traditional Lineage-CS [4].

(2) In order to record the evolution process of data fusion, we also proposed two new transformation provenances: ER Provenance and DCR Provenance which record the evolution process of entity resolution and data conflict resolution.

(3) Finally, the examples and analysis show that the traceable mechanism we proposed can effectively trace the origin of fusion results and the evolution process of entity resolution and conflict resolution. It makes the fusion results more explainable.

This paper is organized as follows. We briefly review some related research efforts in Section 2. The background of data provenance and data fusion is described in Section 3, and the proposed traceable mechanism is introduced in Section 4 respectively on the view of provenance representation and provenance query. Examples and analysis are reported in Section 5, and in the last section we draw conclusions and point out some future directions.

2. RELATED WORKS

With the rapid development of Web, data integration faces the challenges of the large-scale heterogeneous data share. However, the traditional data integration focuses on the form of integrated data and the management emphasis is the data format, and it makes traditional database technologies be under-powered for data integration, data model, query, index, etc. For this situation, data space which is proposed by such companies as Google and Yahoo has drawn comprehensive attention. Whether traditional data integration and data space also implemented the data share in the heterogeneous data sources, in which data provenance contains the same data evolution process in one data source and among several data sources.

Data provenance is the information about data origin and the process of data evolution over time, and it includes the static information of data source and the dynamic information of data evolution that many researchers do not distinguish between the two parts [5-8]. Data provenance has very extensive applications which can used to trace the evolution among different data sources and within the same data source. In the meanwhile, data provenance is an important content of data management. Especially in web data integration, because of the heterogeneity and freedom of data sources and the multi-phase of data integration, we do need to trace the data origin and the evolution process to ensure the integrated data quality and interpretability.

Current researches about data provenance on data integration mainly focus on schema-level provenance that is in the stage of schema mapping. In [9], the scholars developed a suite of tools named TRAMP which can support the debug and trace of schema mappings and transformation queries. On the aspect of the representation of data provenance, the scholars extended current data provenance and proposed transformation provenance and mapping provenance to record the process of schema transformation and schema mapping. And on the aspect of provenance query, they extended SQL and provided the implementation of all provenance operations. In [10], the authors presented a debugger for understanding and exploring schema mappings and described the relationship between source and target data with the schema mapping called routes.

Most recently, some scholars began to study to apply data provenance to resolve data conflict. Beneventano et al proposed a data integration system called MOMIS [11]. They applied data conflict resolution operators by applying Lineage-CS and PI-CS and compare the implementation and effects of the two provenances. In particular, the authors extended PI-CS to handle conflict resolution functions and evaluated the effect of it.

In conclusion, there have been some researches about the application of data provenance in data integration, but there are no researches on how to apply data provenance in several data integration process including data fusion. In order to assure the quality of integrated data, the research of data provenance in data fusion process is an urgent subject.

3. PROBLEM DEFINITIONS

Several notions of provenance have been introduced in the literature to describe the relationship between the input and output data of a transformation. The most common semantics is to define the provenance of an output data item of a transformation to contain all the input data items that contributed to its existence. Therefore, we call such provenance data provenance. Data provenance describes the relationship of the input and output data, however, it does not provide any information about which parts of a transformation were used to derive an output tuple that this kind of information is named as transformation provenance. Transformation provenance can describe the contribution of each operator in a transformation and is very important for tracing the process of data evolution.

In the scenario of data fusion, data transformation operators correspond to entity resolution or data conflict resolution. To understand the process of data fusion, it can help us to understand the process of fused data evolution that we find out which data transformation operator corresponds to which stage of data fusion. We call the transformation provenance of entity resolution Entity Resolution Provenance (ER Provenance) and the one of data conflict resolution is named as Data Conflict Resolution (DCR Provenance).

In the traceable mechanism of data fusion, we only considered the two fusion stages: entity resolution and data conflict resolution in this paper (Fig. **1**). For entity resolution, we employ ER Provenance to present the corresponding references of resolved entities. And for data conflict resolution, we imply the conflict resolution functions described in [12] and employ DCR Provenance to show by which functions the resolved results are gotten.

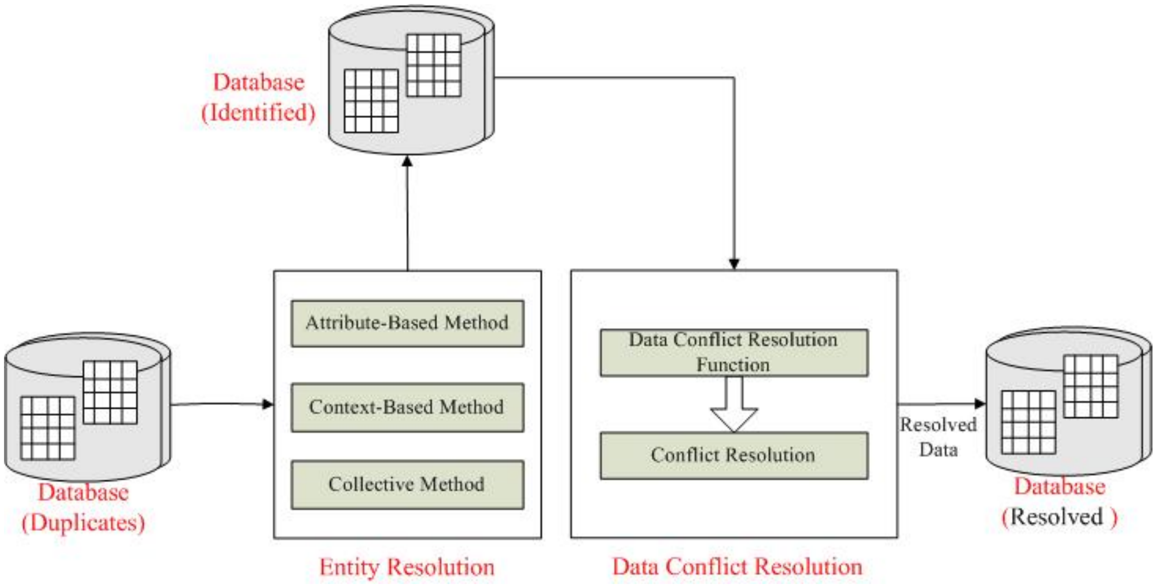


Fig. (1). Data fusion process including entity resolution and data conflict resolution.

4. A TRACEABLE MECHANISM OF DATA FUSION

4.1. Presentation of Fusion Provenance

4.1.1. Data Provenance

Data provenance describes the relationship between a result of a transformation and the inputs which contributed to it. Generally, it is interpreted as the corresponding relationship between the input tuples t_i of a query q and the output tuple t_o which contributed to it in a relational algebra. The majority of existing provenances are represented using Lineage-CS [4] which models the provenance of a result tuple t of a query q as a list $W(q,t) = < Q_1^*, \cdots, Q_n^* >$, where Q_i^* is the input subset that contributes to it. However, Lineage-CS has its obvious disadvantage because it cannot describe which input tuples were combined to produce a result tuple.

For showing this information precisely, we employ PI-CS [3] to represent the provenance of data origin in this paper. In view of the defects of Lineage-CS, PI-CS represents the provenance of data origin as witness list. A witness list w is an element from $\left(Q_1^\varepsilon \times \cdots \times Q_n^\varepsilon\right)$ where $Q_i^\varepsilon = Q_i \cup \perp$. The value $\perp$ indicates that no tuple from an input relation belongs to the witness list w. Thus, a witness list includes each input tuple which contributes to the output and it can serve as an intermediate result for tracing the data origin. We give the formal definition of PI-CS as below [9].

Definition 1 (PI-CS PROVENANCE). For an algebra operator op with inputs $Q_1, \ldots, Q_n$, and a tuple $t \in op\left(Q_1, \ldots, Q_n\right)$ a set $P(op,\mathrm{t}) \subseteq \left(Q_1^\varepsilon \times \cdots \times Q_n^\varepsilon\right)$ where $Q_i^\varepsilon = Q_i \cup \perp$ is the PI-CS provenance of t if it fulfills the following conditions:

$$op\left(P(op,t)\right) = \{t\} \tag{1}$$

$$\forall w \in P(op,t) : op(w) \neq \varnothing \tag{2}$$

$$\neg \exists P' \subseteq W : P' \supset P(op,t) \wedge P' \models (6.1),(6.2),(6.4) \tag{3}$$

$$\forall w, w' \in P(op,t) : w \prec w' \Rightarrow w \notin P(op,t) \tag{4}$$

4.1.2. Entity Resolution Provenance

Entity resolution provenance shows that the fusion results are produced from references that are to be resolved. So we model the provenance as what tuples and rational operators contribute to the output tuples. If data provenance is data-centric, then entity resolution provenance is data-centric and operator-centric. For a query q, the relational operation provenance is modeled as an annotated algebra tree [9]. For a fusion result tuple t and the witness list ω of its data provenance, the corresponding annotated algebra tree express which operators contribute to t. A 1 indicates this operator on ω influences t, a 0 indicates it does not. So entity resolution provenance can be represented as a set of annotated algebra trees where each witness list corresponds to an annotated algebra tree and the annotation of an annotated algebra tree can be determined by the data provenance.

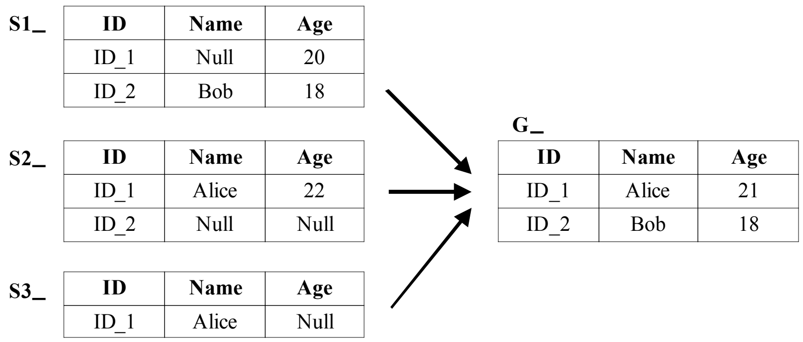
Definition 2 (Entity Resolution Provenance, ER Provenance). For the fusion result of a query q, its ER provenance can be expressed as a set of annotated algebra trees.

$$ERPorv(q,t) = \left\{\left(AlgTree_q, \theta_w\right) \mid w \in P(t)\right\} \tag{5}$$

$$\theta_w = \begin{cases} 0 & if\ op(w) = \varnothing \\ 1 & else \end{cases} \tag{6}$$

4.1.3. Entity Resolution Provenance

Data conflict resolution provenance shows from which attribute value of which records and by which conflict reso-



S1_

ID	Name	Age
ID_1	Null	20
ID_2	Bob	18

S2_

ID	Name	Age
ID_1	Alice	22
ID_2	Null	Null

S3_

ID	Name	Age
ID_1	Alice	Null

G_

ID	Name	Age
ID_1	Alice	21
ID_2	Bob	18

Fig. (2). Example of data fusion.

lution method each value of fusion result tuples comes from. Just as ER provenance, data conflict resolution provenance is data-centric and operator-centric. It's important to note that we only discuss in this paper some conflict resolution strategies and functions summarized in [12]. Naumann et al carried on a comprehensive summary on existing conflict resolution methods which include nearly all existing methods based on relation extension.

Like ER provenance, we introduce annotated attribute tree to model fusion result values coming from which attribute values of which tuples.

Definition 3 (Anotated Attribute Tree). For a fusion result tuple t, its annotated attribute tree is $\left(AttrTree_t, \vartheta\right)$ where $AttrTree_t = \left(V, E\right)$ and the nodes of the tree corresponds to each attributes of t. $\vartheta : V \in AttrTree_t \rightarrow \left\{0,1\right\}$ is a indicating function which describes if the corresponding attribute contributes to fusion result attribute.

Based on data provenance and annotated attribute tree, we give the definition of data conflict resolution provenance.

Definition 4 (Data Conflict Resolution Provenance, DCR Provenance). The data conflict resolution provenance of a fusion result tuple t can be expressed as a set $DCRPorv\left(t\right)$ of annotated attribute trees defined as follow.

$$DCRPorv\left(t\right) = \left\{\left(AttrTree_t, \theta_{A_i}\right) \mid A_i \in Attr_R\right\} \qquad (7)$$

$$\vartheta_{A_i} = \begin{cases} 0 & if\, t_i \in R_i.A_i \\ 1 & else \end{cases} \qquad (8)$$

A_i is an attribute of the source schema R and $Attr_R$ is the attribute set of R. (Since model heterogeneous usually is solved in the process of data fusion, the source schema agrees with the target schema). And $t_i \in R_i.A_i$ expresses that an attribute t_i of t comes from the attribute A_i of the ith tuple.

4.2. Query of Fusion Provenance

In order to define and query data fusion provenances in DBMS, we need to implement the provenances in DBMS. In the aspect of DBMS, we used PostgreSQL which is an open source database and extend its query to support the definition and operation of data provenances. In the aspect of implementation of data provenance, we employed the method similar to [9].

For data provenance, we adopted the implementation of PI-CS. And for ER Provenance and DCR Provenance proposed in this paper, we implemented by extending Perm relational provenance management system [13]. As Perm nicely supports PI-CS and uses a native SQL implementation of provenance, Perm will help to implement the proposed fusion provenances. In addition, to seamlessly integrate into PostgreSQL and use SQL to express query operations, we also implemented the fusion provenances by query rewrite.

5. EXPERIMENTS

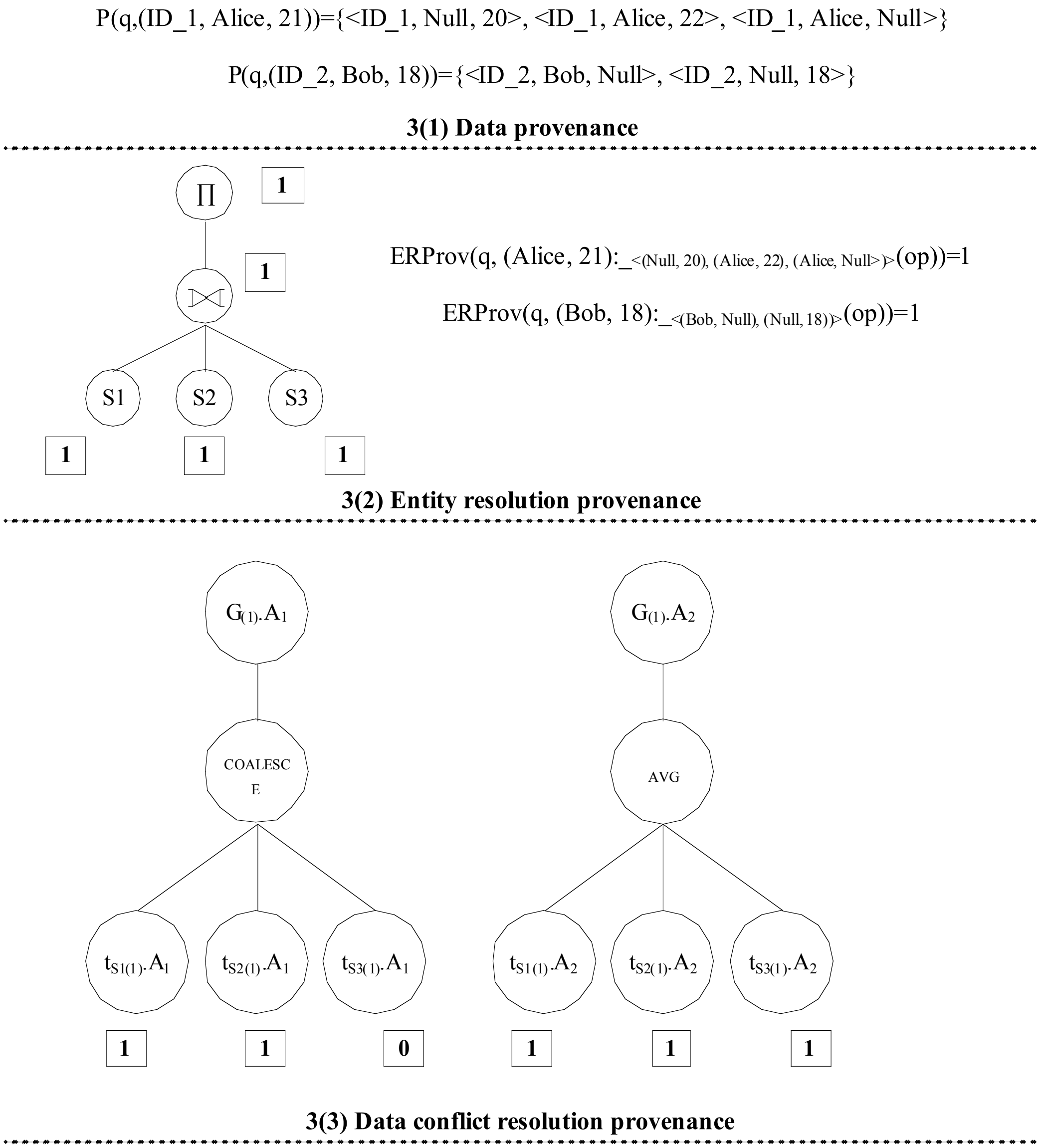
To validate the availability of the traceable mechanism for data fusion, we give an example in this section to analyze fusion provenances proposed in this paper.

As shown in Fig. (**2**), the left side is data from different data sources whose schemata are resolved and the left is fused data by entity resolution and data conflict resolution. Among them, the attribute Name is resolved by the conflict resolution function COALESCE and the attribute Age is resolved by AVG.

The fused results can be gotten from source data by the follow query.

```
SELECT COALESCE( S1.ID, S2.ID, S3.ID ) as ID,
COALESCE(S1.Name, S2.Name, S3.Name) as Name,
AVG(ISNULL(S1.Age,     0),     ISNULL(S2.Age,0),
ISNULL(S3.Age,0)) as Age,
FROM S1 FULL OUTER JOIN S2 ON S1.ID=S2.ID
        FULL OUTER JOIN S3 ON S1.ID=S3.ID
```

Fig. (**3**) shows the fusion provenance corresponding to this query and its fusion results. The data provenance records the tuples which contribute to fusion results and is implemented by PI-CS (see Fig. **3A**). Entity resolution provenance records the query operators and the tuples which contributed to fusion tuple (see Fig. **3B**). And data conflict resolution provenance records the tuples contributed to fusion results and corresponding resolution methods. In the left side of Fig. (**3C**), the conflict resolution function COALESCE is used for the attribute name of the first tuple. Since the function only gets the first non-null value, the value from data source S3 cannot be gained and does not contribute to fusion results.


P(q,(ID_1, Alice, 21))={<ID_1, Null, 20>, <ID_1, Alice, 22>, <ID_1, Alice, Null>}
P(q,(ID_2, Bob, 18))={<ID_2, Bob, Null>, <ID_2, Null, 18>}
∏
1
⋈
1
S1
S2
S3
1
1
1
ERProv(q, (Alice, 21):_<(Null, 20), (Alice, 22), (Alice, Null>)>(op))=1
ERProv(q, (Bob, 18):_<(Bob, Null), (Null, 18))>(op))=1
G(1).A1
COALESCE
tS1(1).A1
tS2(1).A1
tS3(1).A1
1
1
0
G(1).A2
AVG
tS1(1).A2
tS2(1).A2
tS3(1).A2
1
1
1

3(1) Data provenance

3(2) Entity resolution provenance

3(3) Data conflict resolution provenance

Fig. (3). Example of fusion provenance.

CONCLUSION

On the basis of characteristics that the existing fusion process is transparent and the fusion stages are isolated. We proposed a traceable data fusion mechanism based on data provenance which can trace the data sources of fusion results and the evolutionary process. The mechanism mainly targets forwards entity resolution and data conflict resolution stage. We represented the provenance of data origin using PI-CS which is more accurate because PI-CS can record the intermediate process of data evolution. In order to record the evolution process of data fusion, we proposed two transformation provenances: entity resolution provenance and data conflict resolution provenance which record respectively the evolution process of entity resolution and data conflict resolution.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (No. 61303007, 11001155), Shandong Provincial Key Laboratory of Software Engineering (No. 2013SE02), Shandong Provincial Key Laboratory of Software Engineering (No. 2013SE02), Shandong Province higher Educational Science and Technology Program (No. J11LA08), the Outstanding Young Scientists Foundation Grant of Shandong Province (No. BS2013DX044) and Postdoctoral Science Special Foundation of Shandong Province (No. 201303057).

REFERENCES

[1] X. L. Dong and F. Naumann, "Data fusion - resolving data conflicts for integration", *Proceedings of the VLDB Endowment (PVLDB)*, vol. 2, no. 2, 1654-1655, 2009.

[2] Y.L. Simmhan, B. Plale, and D. Gannon, "*A Survey of Data Provenance Techniques*", Technical Report IUB-CS-TR618, Computer Science Department, Indiana University, Bloomington, IN 47405, 2005.

[3] A. Meliou, W. Gatterbauer, K. Moore, and D. Suciu, "The complexity of causality and responsibility for query answers and non-answers", *Proceedings of the VLDB Endowment,* vol. 4, no. 1, pp. 34-45, 2010.

[4] Y. Cui, J. Widom, and J. L. Wiener, "Tracing the lineage of view data in a warehousing environment", *ACM Transactions on Database Systems*, vol. 25, no. 2, pp. 179-227, 2000.

[5] M. Gao, C.Q. Jin, X.L. Wang, X.X. Tian, and A.Y. Zhou, "A survey on management of data provenance", *Chinese Journal Of Computers*, vol. 33, no. 3, pp. 373-389, 2010.

[6] P. Buneman, S. Khanna, and W.C. Tan, "Why and where: a characterization of data provenance", In: *Proceedings of the 8th International Conference on Data Theory*, London, UK, pp. 316-330, 2001.

[7] J. Cheney, L. Chiticariu, and W. Tan, "Provenance in databases: why, how, and where", *Foundations and Trends in Databases*, vol. 1, no. 4, pp. 379-474, 2009.

[8] Y.L. Simmhan, B. Plale, and D. Gannon, "*A Survey of Data Provenance Techniques*", Technical Report IUB-CS-TR618, Indiana University, Bloomington, 2005.

[9] B. Glavic, G. Alonso, R.J. Miller, and L.M. Haas, "TRAMP: understanding the behavior of schema mappings through provenance", *Proceedings of the VLDB Endowment*, vol. 3, no. 1, pp. 1314-1325, 2010.

[10] L. Chiticariu, and W.-C. Tan, "Debugging schema mappings with routes", In: *Proceedings of the 32nd International Conference on Very Large Data Bases*, Seoul, Korea, pp. 79-90, 2006.

[11] D. Beneventano, A.R. Dannoui, and A. Sala, "Data Lineage in the MOMIS Data Fusion System", In: *1st International Workshop on Managing Data Throughout its Lifecycle* (*DaLi 2011*), in conjunction with ICDE 2011, Hannover, pp. 53-58, 2011.

[12] J. Bleiholder, and F. Naumann, "Conflict handling strategies in an integrated information system", In: *Proceedings of the International Workshop on Information Integration on the Web (IIWeb).* Edinburgh, UK, 2006.

[13] B. Glavic, and G. Alonso, "Perm: processing provenance and data on the same data model through query rewriting", In: *IEEE 25th International Conference on Data Engineering*, Shanghai, China, pp. 174-185, 2009.

Received: November 21, 2014 Revised: December 02, 2014 Accepted: December 05, 2014