SEMINAR REPORT

DATA FUSION
JENS BLEIHOLDER and FELIX NAUMANN

CSE-703 Data Quality
Prof. Jan Chomicki

Arun Sharma
UB Person: 50206920

Introduction

Data Fusion is integration of data in of the same real world objects in single consistent and clean representation. Since consistency has been an active area of research in information integration, it is important to focus on data fusion part in data integration system. The article [1] is an overview of techniques in the field of data fusion used in data integration system these days. The main goal of data integration system is to provide user complete yet concise information without separately accessing the main data source.

Various topics related to data fusion techniques along with their drawbacks are discussed sequentially such as concept of data integration and each of its components are discussed in brief except Data Fusion (which is core part of this paper). Important concepts such as Data Transformation, Deduplication and the significance of completeness and conciseness (analogous to precision and recall). Some drawbacks and limitation are also discussed subsequently with the flow of the paper for example Conflicts and how to resolve different type of conflicts. Further we discuss relation operations such as Join and Union approaches with help of examples and some other statistical techniques. Finally, we will end it with concluding it with limitation

- Data Fusion
- Data Transformation
- Duplication Detection
- Completeness and Conciseness
- Conflicts
- Relational Operations and Techniques
- Other Techniques
- Conclusion

There are many works from many authors we will encounter in this report while explaining the concepts and the drawbacks faced while applying them. For example, Felix Naumann (one of the author) collaborated with many author including Jens Bleiholder in past producing groundbreaking research in the field of Data Fusion. In [2], both Naumann and Bleiholder published the data integration system and this shortcoming when it comes to conflicts and data fusion deals with this situation. They also discussed conflicts in detail with their resolving strategies. In [3] Wies and Naumann takes data deduplication objects in XML Documents. They used domain-independent algorithm along with string comparison using optimized edit distance. Hernandez and Stolflo [4] discuss the problem of dealing with duplicates while merging multiple databases of common entities. They developed a data cleansing task which provides a programming rule module for finding duplicates on a real-world dataset. Motro [5] argued both solution based on ranking and fusion as flawed and provides us an alternate approach based on knowledge of the performance of the data including recentness, availability, accuracy and cost which combines in utility function and gives an overall value to the user. In [6], the author discusses two modules in data fusion, Data Quality Broker and Quality Notification Services.

Basic Concepts and Definitions

Data Fusion is combination of results, and presenting them to the user is performed by the integration system.

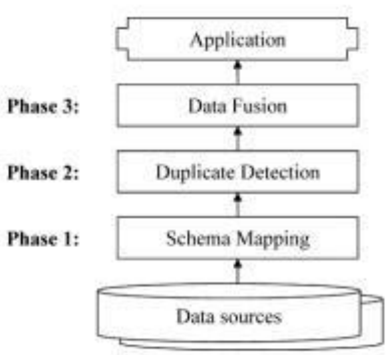


Fig. 1. A data integration process.

The above diagram is the stepwise process of data integration including data preprocessing (such as Schema Mapping and Duplication detection) and Data Fusion. While doing some preprocessing we may encounter some conflicts which were first discussed by Dayal et. al [7]. Application part is beyond the scope of the report however interested readers can refer to the paper. The schema Mapping comes under data transformation and Duplicate detection comes separately.

Data Transformation

Data integration usually deals with heterogeneous schemata and present to user for query result, it must be transformed into global schema. Two approaches are common in this process, Schema Integration and Schema Mapping. Schema Integration generate a new schema that is complete and correct with respect to the source schemata which is both minimal and understandable. Batini et al. [8] gives an overview of difficulties faces in this process. On the other hand, schema mapping assumes a given target schema which is driven by the need to include a set of sources in each integrated information system. Besides, there is one other technique called scheme matching which semi-automatically finds the correspondence between two schemata [9]. Since the goal of both approaches is same, schema mapping is easier to perform and much of the remaining work can be handled in data fusion end. After this step, all objects of a certain type are represented homogeneously.

Duplicate Detection

The goal of data deduplication is to identify multiple representation of the same real world object. The first step to resolve deduplication is the cosine similarity measure which is used in many information retrieval system, but there are 2 problems to solve while doing this process, effectiveness and efficiency. Effectiveness is the quality of the similarity measure and the choice of threshold. They are both domain specific and domain independent (such as edit distance) and tuning thresholds leads to tradeoff between precision and recall.

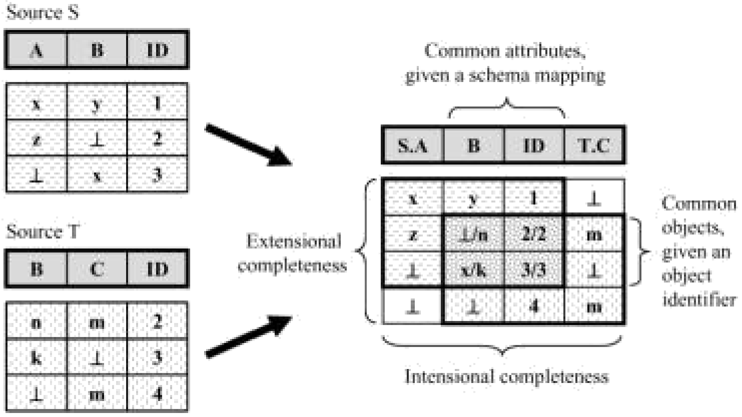
Completeness and Conciseness

There are two broad goals in data integration achieve high completeness and conciseness. High completeness means that no data is left behind means low precision and high recall. This results in adding more data objects such as rows etc. leading to more duplication results in high recall.

Conciseness is removal of duplicate data without any loss of information to provide consistency such as deduplication of rows. Both operation performed in schema level as well as in Data Level.

Extensional Completeness means the percentage of real word objects and it is used for measuring duplicates in the dataset. Intentional completeness is number of unique attributes in the dataset so that we can perform mapping among them.

Similarly, Extensional Conciseness is the number of unique objects in the dataset divided number of all unique objects in the dataset (not to be confused with Extensional Conciseness which is number of unique objects in the dataset divided by total number of objects in the dataset). Intentional Conciseness deals with unique attributes to overall unique attributes in the dataset. The following is the pictorial example of these definitions:



Source S

A	B	ID
x	y	1
z	⊥	2
⊥	x	3

Source T

B	C	ID
n	m	2
k	⊥	3
⊥	m	4

S.A	B	ID	T.C
x	y	1	⊥
z	⊥/n	2/2	m
⊥	x/k	3/3	⊥
⊥	⊥	4	m

Fig 2: Example used to describe above definitions

Some conflicts we have noticed in the above figure such as deciding x/k or null/n, which strategies are used to resolve these conflicts are discussed in the next section.

Conflicts

There are basically three types of conflicts: schematic, identity and data conflicts. Schematic conflicts deal with different data sources or same attributes name. Identity conflicts on the other hand handles same real world objects from different data sources. However, these are resolved in the first 2 phases of data integration, data conflicts still an issue (as we have seen in the previous figure) [1]. Hence, data conflicts are the multiple representation of the same real world data. Hence there are two types of data conflicts which are common, Uncertainties and Contradiction. Uncertainties arises when a non-null value and one or more null values describes the same property of an object. Contradictions on the other hand is when different non-null values are used to describe the same property of an object. For resolving these conflicts, we apply different strategies which are classified as follows:

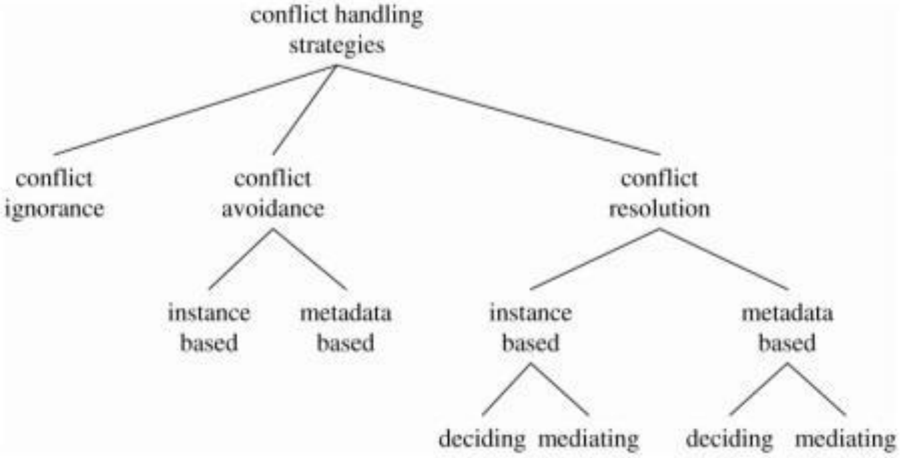
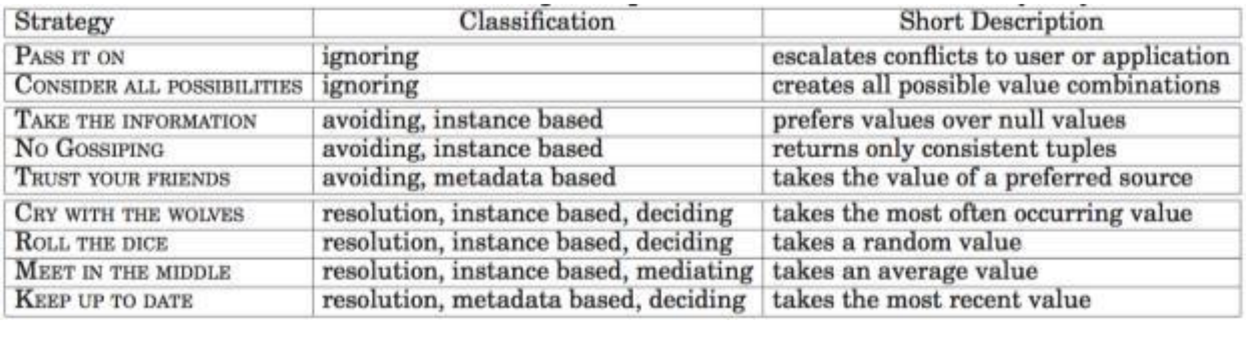


Fig 3: Conflict Classification

We will briefly describe these strategies one by one as starting from conflict ignorance strategy which is not to make any decision regarding the data conflict. To resolve this, we use Pass it On technique. On the contrary, Conflict avoidance acknowledge the conflicts but do not resolve or detect the existing conflicts. Instead they handle the data conflicts with their unique decision such as Trust your Friend technique where we prefer on object over another. Conflict resolving regards all the data and the metadata before deciding to resolve the conflict. Notice that they are further divided into deciding and mediating in which the former decides the value among those objects which are already taking part in the conflict whereas the later decides the value which does not taking part in the conflict. Here we can apply strategies such as KeepUpToDate where we decide based on timestamps. Likewise, there are several techniques which are used to resolve these issues which are address later in this report. The table below represents other techniques which are beyond the scope of the paper.



Strategy	Classification	Short Description
Pass it on	ignoring	escalates conflicts to user or application
Consider all possibilities	ignoring	creates all possible value combinations
Take the information	avoiding, instance based	prefers values over null values
No Gossiping	avoiding, instance based	returns only consistent tuples
Trust your friends	avoiding, metadata based	takes the value of a preferred source
Cry with the wolves	resolution, instance based, deciding	takes the most often occurring value
Roll the dice	resolution, instance based, deciding	takes a random value
Meet in the middle	resolution, instance based, mediating	takes an average value
Keep up to date	resolution, metadata based, deciding	takes the most recent value

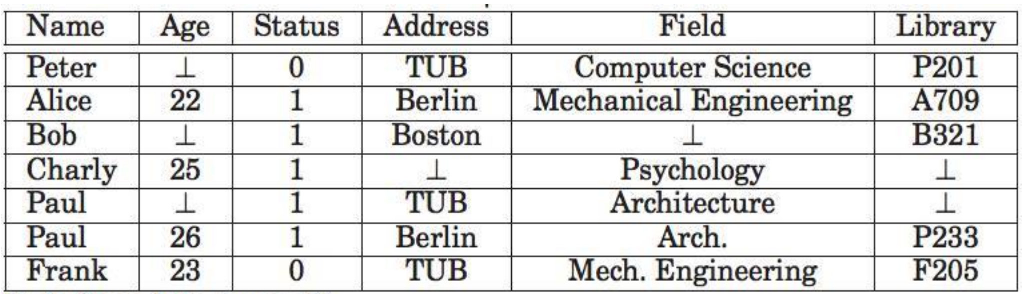
Fig 4: Strategy used to resolve conflicts

Relational Operations and Techniques

Here the actual discussion of Data Fusion approaches such as union and join are discussed in brief. First data is integrated from different sources into one single integrated table. Different approaches such as Joins where tuples from different tables are joined into one single table and other is Union approach where first we build a common schema and then append the different tuple set from the source table. From these we consider some properties such as value preservation, object preservation and uniqueness preservation.

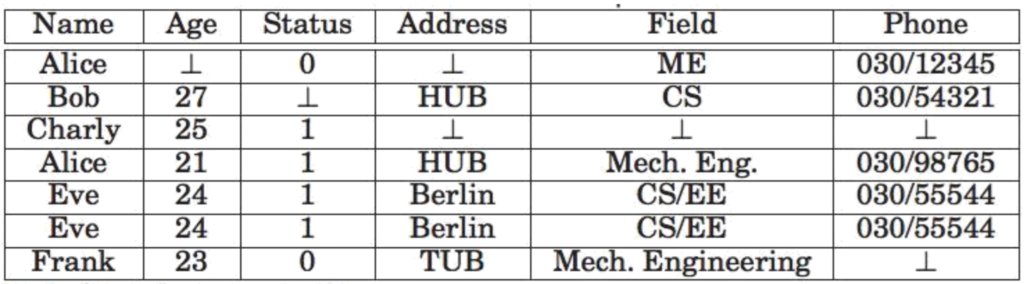
Join Approach

Join approaches in general increase intentional completeness, as attributes from different sources are separately included in the result. Except full-join, no join can access extensional completeness. We will concentrate on example having two tables given below:



Name	Age	Status	Address	Field	Library
Peter	⊥	0	TUB	Computer Science	P201
Alice	22	1	Berlin	Mechanical Engineering	A709
Bob	⊥	1	Boston	⊥	B321
Charly	25	1	⊥	Psychology	⊥
Paul	⊥	1	TUB	Architecture	⊥
Paul	26	1	Berlin	Arch.	P233
Frank	23	0	TUB	Mech. Engineering	F205

Data from data source U_1.



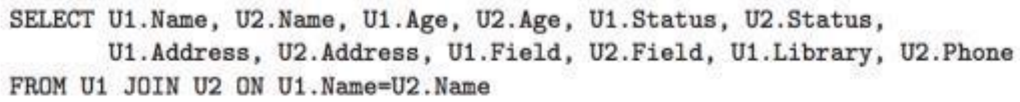
Name	Age	Status	Address	Field	Phone
Alice	⊥	0	⊥	ME	030/12345
Bob	27	⊥	HUB	CS	030/54321
Charly	25	1	⊥	⊥	⊥
Alice	21	1	HUB	Mech. Eng.	030/98765
Eve	24	1	Berlin	CS/EE	030/55544
Eve	24	1	Berlin	CS/EE	030/55544
Frank	23	0	TUB	Mech. Engineering	⊥

Data from data source U_2.

Fig 5: Dataset for student information

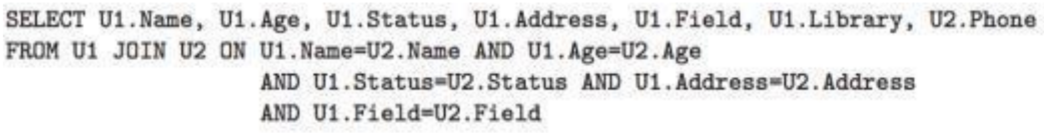
Standard Join

Equijoin: Combining two relation if the join condition is equality among two columns which are common. These join uses real world identifier such as Name, ID etc. for uniqueness. For example, in the given SQL statement Name is the only unique identifier to join the table:



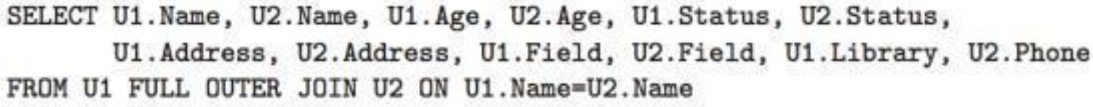
```
SELECT U1.Name, U2.Name, U1.Age, U2.Age, U1.Status, U2.Status,
       U1.Address, U2.Address, U1.Field, U2.Field, U1.Library, U2.Phone
FROM U1 JOIN U2 ON U1.Name=U2.Name
```

Natural Join: When two columns are join based on attributes value and attribute which are not present in either of two tables are not considered for joining.



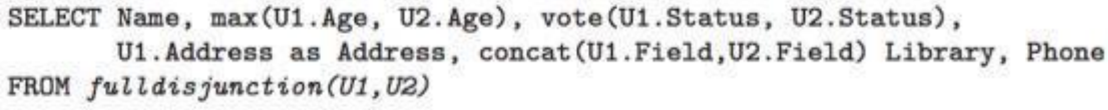
```
SELECT U1.Name, U1.Age, U1.Status, U1.Address, U1.Field, U1.Library, U2.Phone
FROM U1 JOIN U2 ON U1.Name=U2.Name AND U1.Age=U2.Age
                   AND U1.Status=U2.Status AND U1.Address=U2.Address
                   AND U1.Field=U2.Field
```

Full-Outer Join: Considering attribute values of one of the two relation. Attributes does not contribute to the relation are padded with null values.



```
SELECT U1.Name, U2.Name, U1.Age, U2.Age, U1.Status, U2.Status,
       U1.Address, U2.Address, U1.Field, U2.Field, U1.Library, U2.Phone
FROM U1 FULL OUTER JOIN U2 ON U1.Name=U2.Name
```

Full Disjunction Join: Since outer joins are not associative, involvement of 2 or more tables is necessary for joining the result. It behaves same as outer joins.

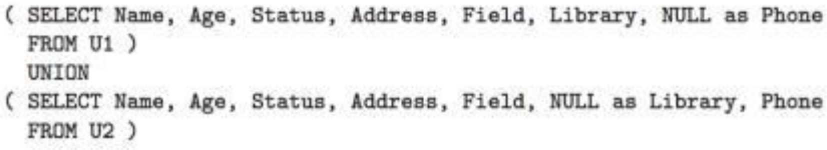


```
SELECT Name, max(U1.Age, U2.Age), vote(U1.Status, U2.Status),
       U1.Address as Address, concat(U1.Field,U2.Field) Library, Phone
FROM fulldisjunction(U1,U2)
```

Union Approach

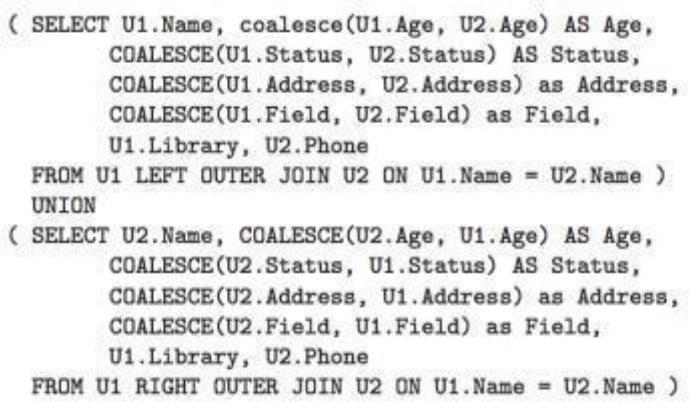
It is a natural way to accomplish extensional completeness since it combines two union compatible results and remove exact duplicates. Further they are classified into many but we will discuss only Outer and Merge/Priority Merge.

Outer Union: It combines two non-union compatible results and padded them with null values if attribute values are not given. This performs regular joins as follows:



```
( SELECT Name, Age, Status, Address, Field, Library, NULL as Phone
  FROM U1 )
  UNION
( SELECT Name, Age, Status, Address, Field, NULL as Library, Phone
  FROM U2 )
```

Merge/Priority Merge: It removes COALESC function of SQL to remove any uncertainties. Further missing values are replaced from the values of other tuples. However, contradictions are not resolved hence priority merge uses priority remove these contradiction and conflicts. The example related to this is as follows:



```
( SELECT U1.Name, coalesce(U1.Age, U2.Age) AS Age,
        COALESCE(U1.Status, U2.Status) AS Status,
        COALESCE(U1.Address, U2.Address) as Address,
        COALESCE(U1.Field, U2.Field) as Field,
        U1.Library, U2.Phone
  FROM U1 LEFT OUTER JOIN U2 ON U1.Name = U2.Name )
  UNION
( SELECT U2.Name, COALESCE(U2.Age, U1.Age) AS Age,
        COALESCE(U2.Status, U1.Status) AS Status,
        COALESCE(U2.Address, U1.Address) as Address,
        COALESCE(U2.Field, U1.Field) as Field,
        U1.Library, U2.Phone
  FROM U1 RIGHT OUTER JOIN U2 ON U1.Name = U2.Name )
```

Increasing Conciseness in Union Results
It can be removed by doing group by operation followed by aggregation operation on the value of attributes such as max, sum or coalesce.

Other Techniques

They are neither union based nor join based. This can be done by considering all possibilities or considering consistent possibilities:
All possibilities: Simply adding additional column to each table helps us to decide the fate of the tuple (just as regression and classification in machine learning). Either it can be done with continuous or discrete probabilities. Also, probability distribution of the data also takes place.
Consistent Possibilities: Consistent database is created using only insertion and deletion from the consistent database. Hence, overall goal is to find a consistent answer.

Conclusion

From the above discussion, we conclude that union approaches are well suited for data fusion than Join approach. Join approaches usually retain information than union as it concentrates on completeness rather than conciseness and achieving completeness is comparatively is easier task than conciseness. One of the limitation in Data Fusion is it is more domain dependent task i.e. we must look at the dataset before deciding which type of approach should be appropriate. Resolving consistencies could not be accomplished without applying any function. Finally, we can say that performing data fusion operation is possible in real world and much of the work must be done regarding this field.

REFERENCES

[1] BLEIHOLDER, J. AND NAUMANN, Data fusion. ACM Comput. Surv. 41(1): 1:1-1:41 (2008)

[2] BLEIHOLDER, J. AND NAUMANN, F. 2006. Conflict handling strategies in an integrated information system. In *Proceedings of the IJCAI Workshop on Information on the Web (IIWeb)*.

[3] WEIS, M. AND NAUMANN, F. 2004. Detecting duplicate objects in XML documents. In *Proceedings of the International Workshop on Information Quality Informative Systems (IQIS)*.

[4] HERNA´NDEZ, M. A. AND STOLFO, S. J. 1998. Real-World data is dirty: Data cleansing and the merge/purge problem. *Data Mining Knowl. Discov. 2*, 1, 9–37.

[5] MOTRO, A., ANOKHIN, P., AND ACAR, A. C. 2004. Utility-Based resolution of data inconsistencies. In *Proceed- ings of the International Workshop on Information Qualities in Information Systems (IQIS)*. ACM Press, 35–43.

[6] SCANNAPIECO, M., VIRGILLITO, A., MARCHETTI, C., MECELLA, M., AND BALDONI, R. 2004. The DaQuinCIS archi- tecture: A platform for exchanging and improving data quality in cooperative information systems. *Inf. Syst. 29*, 7, 551–582.

[7] DAYAL, U. AND HWANG, H.-Y. 1984. View definition and generalization for database system integration in a multidatabase system. *IEEE Trans. Softw. Eng. 10*, 6 (Nov,), 628–645.

[8] BATINI, C., LENZERIN, M., AND NAVATHE, S. B. 1986. A comparative analysis of methodologies for database schema integration. *ACM Comput. Surv. 18*, 4, 323–364.

[9] RAHM, E. AND BERNSTEIN, P. A. 2001. On matching schemas, automatically. Tech. Rep. MSR-TR-2001-17, Microsoft Research, Redmond, Washington. February.