

# A Conceptual Level Design Methodology for Probabilistic Relational Databases

Ping-Tsai Chung and Fahmeed Afzal  
Department of Computer Science  
Long Island University, Brooklyn Campus, New York

## **ABSTRACT** –

*When multiple heterogeneous databases show different values for the same data item, its actual value is not known with certainty. To develop corporate data warehouses, which consolidate data from multiple heterogeneous data sources, has become an important issue for designing modern business information systems. Probabilistic relational databases have extended from the relational database model by incorporating probability measures to capture the uncertainty associated with data items.*

*In this paper, we develop a two-layer conceptual-level design model to incorporate data uncertainty at the conceptual level database design phase. The lower layer is an Improved Entity-Relationship Data (IERD) modeling layer with data uncertainty and the upper layer is a normalization layer for probabilistic relations. When probabilistic data are already available for an application, it is useful in arriving at the desired level of normalization in the upper layer when reorganizing existing relationships in database in the lower layer. This probabilistic database design methodology will be helpful for us to develop a semiautomated software tool for the logical design for probabilistic relational database applications. The resulting database relations will be represented by XML formats, which are particularly suitable to a distributed networking environment.*

**Key words:** *Probabilistic relational databases, entity-relationship modeling, database normalization design, database software system development*

## **I. INTRODUCTION**

Database systems are an essential component of modern business information systems. In some real business applications, the actual value of a data item may be unknown or not yet realized. For example, the data of market surveys are often expressed in a consolidated manner in which the details of individual customer trends are summarized. Such uncertain

information is of considerable importance in designing new services or products in many different industries. For example, analysis of consolidated customer calling patterns for a telecommunications company can help to create more attractive pricing and promotional plans, perhaps attracting new customers from a competitor. Also, for a high-volume manufacturing company, collecting and analysis of

consolidated customer purchase trends, seasonality, and so on can help the company plan its production and lower its inventory levels, saving money for other purposes.

Another source of data uncertainty is *data heterogeneity* [4]. When two or more heterogeneous databases provide different values for the same data item, its actual value is not known with certainty. This has become an important concern in developing corporate data warehouses which consolidate data from multiple heterogeneous data sources. In heterogeneous databases, there are some *information integration problems* [3]: different databases differ in *Model* (relational versus object-oriented); in *Schema* (normalized versus unnormalized); in *Terminology* (for example, consultants or subcontractors); in *Measure* (for example, meters versus feet).

Moreover, there are some *domain mismatch problems* do exist in heterogeneous databases such as name conflicts for semantically related data items; data types conflicts for semantically related data items.

In this paper, we develop a two-layer conceptual-level design model to incorporate data uncertainty at the conceptual level database design phase: the first layer is an *Improved Entity-Relationship Data (IERD) modeling layer* with data uncertainty and the second layer is a *normalization layer* for probabilistic relations. When probabilistic data are already available for a database application, it is useful in arriving at the desired level of normalization in the upper layer when reorganizing existing relationships in database in the lower layer.

This probabilistic database design methodology will be helpful for us to develop a *semiautomated software tool* for the logical design for probabilistic relational database applications. The resulting database relations will be represented by XML formats, which are particularly suitable to Internet environment. In Section II, we present two-layer conceptual level probabilistic database design model. We first discuss the improved entity-relationship data modeling layer with data uncertainty. We then present the Normalization layer for probabilistic relations. In Section III, we discuss the development of semiautomated tool called *LogicDT.PDB* for the logical design for probabilistic relational databases. Finally, in Section IV, we make conclusions of this paper and discuss future possible directions of this research work.

## **II. TWO-LAYER CONCEPTUAL LEVEL PROBABILISTIC DATABASE DESIGN MODEL**

### **II.A. IMPROVED ENTITY-RELATIONSHIP DATA (IERD) MODELING LAYER WITH DATA UNCERTAINTY**

This Section presents a two-layer conceptual level probabilistic database design model. For the following, we introduce some notations to support the work to consolidate uncertainty data from multiple heterogeneous data sources.

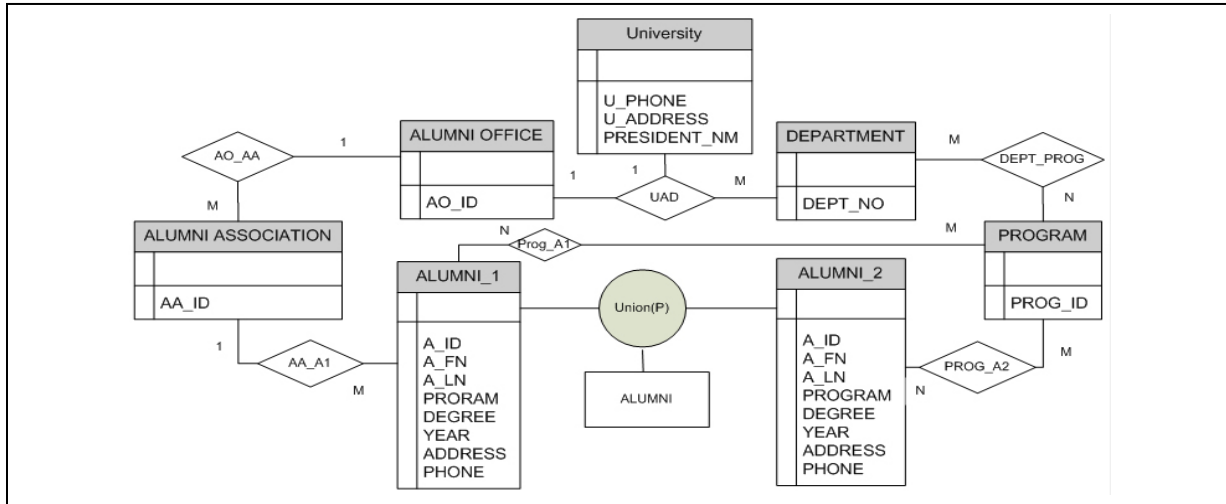


Figure 1. Improved ERD (IERD) for the probabilistic ALUMNI database.

Suppose there are  $m$  data sources (i.e.,  $m$  relations) collected for the same data (i.e., one relation), where we assume that *these  $m$  relations have the same primary key, and the primary key holds the same data type*. Furthermore, we assume that  $m$  is an integer, and it is finite and small enough so that the Database Management Information System (e.g. *the data warehouse or the mediator*) could support and perform the information. Note that there are in general two information integration approaches [3]: the first one is *Data Warehousing* approach which makes copies of the data sources at central site and transform it to a common schema. The second approach is called *Mediation* which creates a *view* of all data sources, as if they were integrated. The mediator answers a view query by translating it to terminology of the data sources and query them.

In Figures 1, we illustrate the UNION (P) operator in a probabilistic ALUMNI database application example. In this ALUMNI database for a University, there are two data sources to collect and analyze Alumni information. The relations of entities of this ALUMNI database are showed in the Figure 1. The Improved ERD (IERD) for the probabilistic ALUMNI database after applying a consolidation operator for ALUMNI relations is illustrated in Figure 2. In Figure 3, it shows two conflicting ALUMNI relations consolidated as probabilistic relation.

The meaning of these new consolidation notations are as follows.

(1) UNION(P): it (automatically) performs the union operation for the attributes of the  $m$  input database relations.

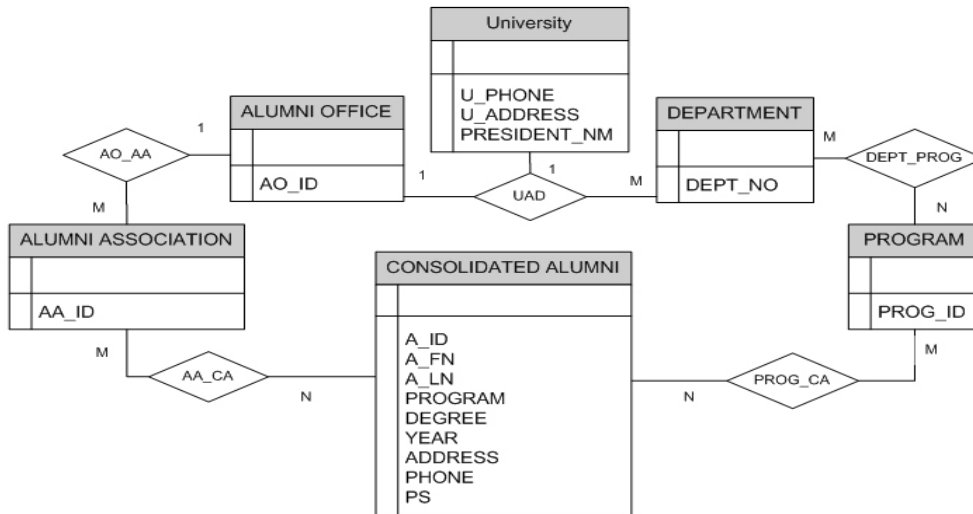


Figure 2. The Improved ERD (IERD) for the probabilistic ALUMNI database after applying UNION(P) operator for ALUMNI relations.

Alumni 1 Relation							
A_ID	A_FN	A_LN	PROGRAM	DEGREE	YEAR	ADDRESS	PHONE
001	Nancy	Davolio	CS	MS	1999	507 - 20th Ave. E.	(206) 555-9857
002	Andrew	Fuller	CS	BS	1995	908 W. Capital Way	(206) 555-9482
003	Janet	Leverling	MAT	MS	1990	722 Moss Bay Blvd.	(206) 555-3412
004	Margaret	Peacock	PHY	BS	1992	4110 Old Redmond Rd.	(206) 555-8122
005	Steven	Buchanan	ACC	BS	2000	14 Garrett Hill	(718) 555-4848
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Alumni 2 Relation							
A_ID	A_FN	A_LN	PROGRAM	DEGREE	YEAR	ADDRESS	PHONE
001	Nancy	Luise	CS	MS	1999	507 - 20th Ave. E.	(206) 555-9857
002	Andrew	Fuller	CS	BS	1995	908 W. Capital Way	(206) 555-9482
003	Janet	Leverling	CS	MS	2000	722 Moss Bay Blvd.	(206) 555-3412
004	Margaret	Peacock	PHY	BS	1992	4110 Old Redmond Rd.	(206) 555-8122
005	Steven	Buchanan	ACC	BS	2000	405 E 16 STREET	(718) 555-4848
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

A_ID	A_FN	A_LN	PROGRAM	DEGREE	YEAR	ADDRESS	PHONE	pS
001	Nancy	Davolio	CS	MS	1999	507 - 20th Ave. E.	(206) 555-9857	0.5
001	Nancy	Luise	CS	MS	1999	507 - 20th Ave. E.	(206) 555-9857	0.5
002	Andrew	Fuller	CS	BS	1995	908 W. Capital Way	(206) 555-9482	1.0
003	Janet	Leverling	MAT	MS	1990	722 Moss Bay Blvd.	(206) 555-3412	0.5
003	Janet	Leverling	CS	MS	2000	722 Moss Bay Blvd.	(206) 555-3412	0.5
004	Margaret	Peacock	PHY	BS	1992	4110 Old Redmond Rd.	(206) 555-8122	1.0
005	Steven	Buchanan	ACC	BS	2000	14 Garrett Hill	(718) 555-4848	0.5
005	Steven	Buchanan	ACC	BS	2000	405 E 16 STREET	(718) 555-4848	0.5
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Figure 3. Conflicting ALUMNI relations consolidated as probabilistic relation.

(2) INTERSECT(P): it (automatically) perform the intersect operation for the attributes of the  $m$  input database relations.

(3) CONSOLIDATE(P): the database designer or database administrator (DBA) (semiautomatically) performs a consolidation operation for the *selected* attributes of the  $m$  input database relations.

The purpose of the operator CONSOLIDATE(P) is to create a *schema view* describing only data that the database application may access in the local databases.

For example, suppose ALUMNI\_1 (from Alumni Office) and ALUMNI\_2 (from Department) have the following attributes in their Schemas in Figure 4. A Consolidated ALUMNI view can be created semiautomatically in Figure 4 by database designer or database DBA.

ALUMNI\_1 (A\_ID A\_FN, A\_LN, Program, Degree, Year, Phone, Address, Email\_addr, employer, title)

ALUMNI\_2 (A\_ID A\_FN, A\_LN, Program, Highest\_degree, Year, Phone, Email)

CONSOLIDATED ALUMNI (A\_ID A\_FN, A\_LN, Program, Highest\_degree, Year, Phone, Email)

Figure 4. The consolidated ALUMNI relation after applying a CONSOLIDATE(P) operator for ALUMNI relations.

## II. B. NORMALIZATION LAYER FOR PROBABILISTIC RELATIONS

We now present the normalization layer for the probabilistic relations. In case of probabilistic relations, the concepts of functional dependencies have been generalized to the *stochastic dependencies*, and *probabilistic normal forms* have been defined in a way that preserves certain stochastic dependencies properties [4]. A *Bayesian network* captures the stochastic dependencies among attributes (random variables) in the form of a directed graph with attributes represented as nodes. That is, we define a Bayesian network  $G_R = (\bar{R}, E_R)$  representing the dependency structure of  $R$ , where  $R$  is a probabilistic relation scheme. A *direct* influence of one attribute  $X$  on another attribute  $Y$ , is represented by an edge  $X \rightarrow Y$ . All *indirect* influences are captured by paths containing two or more edges.

The main objectives of normalization of database design are to reduce and control data redundancies in a database application so that the database designer could reduce and control the chance of the *update anomalies: insertion anomalies, deletion anomalies, and modification anomalies* [1].

Definitions of Probabilistic Normal Forms (PNFs) could be found in [4]. In general, the definitions of PNFs are more general cases of conventional normal forms. We summarize some PNFs properties in [4] as follows,

**Proposition 1.** 1PNF is exactly the same as 1NF if a relation scheme does not have any multivalued attribute.

**Proposition 2.** A relation  $R$  is in 2NF if it is in 2PNF.

**Proposition 3.** A relation  $R$  is in 3NF if it is in 3PNF.

**Proposition 4.** A relation  $R$  is in (Boyce-Codd Normal Form) BCNF if it is in BCPNF.

Note that BCNF is stricter than 3PNF in the general case, but BCNF and 3PNF are exactly the same for relations with stochastic dependencies which contain no nontrivial functional dependencies.

**Proposition 5.**

Let  $R$  be a relation scheme contains no nontrivial functional dependencies, and let  $K$  be its primary key,  $R$  is in BCPNF if and only if  $R$  is in 3PNF.

A Bayesian network for the probabilistic ALUMNI relation is illustrated in Figure 5.

### III. SEMIAUTOMATED TOOL FOR THE LOGICAL DESIGN FOR PROBABILISTIC RELATIONAL DATABASES

This section, we discuss the development of a semiautomated software system tool called *LogicDT.PDB* for developing the logical design for probabilistic relational databases.

*The LogicDT.PDB* implemented based on the two layer architecture illustrated in Figure 6. There are three major components developed for *LogicDT.PDB*:

(1) *Inputs of LogicDT.PDB - IERD Input and GUI Input*

The default input of the Inputs of *LogicDT.PDB* is through the IERD INPUT, users (*Database designer or DBA*) specifies the entities, attributes of the entities, identifies the relationships between entities; furthermore, users specifies the consolidation operators if a relation has data from different data sources. Also, *LogicDT.PDB* can provide a user-friendly input interface *Manu* implemented by Visual Basic, C# or Java technologies.

(2) *Shared Data Structure between Layers in LogicDT.PDB*

After users completed an input design work in the 1<sup>st</sup> layer (*IERD layer*), the second layer (Normalization layer) component of the *LogicDT.PDB* system will retrieve the information of attributes for each database relation from a shared data structure in the *IERD layer*.

When probabilistic data are already available for an application, it is useful in arriving at the desired level of normalization in the second layer. The new normalized database information will be helpful for reorganizing existing IERD relationships in database in the first layer (i.e., *IERD layer*).

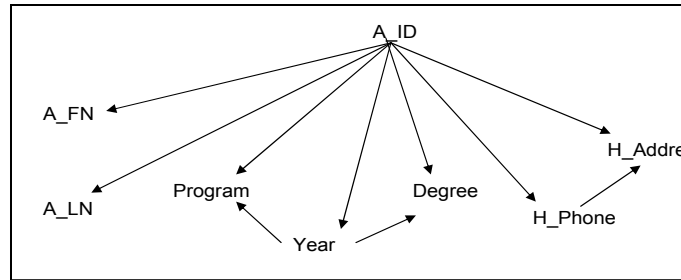


Figure 5. Dependency structure for probabilistic ALUMNI relation.

A_ID	A_FN	pS
001	Nancy	1.0
002	Andrew	1.0
003	Janet	1.0
004	Margaret	1.0
005	Steven	1.0
⋮	⋮	⋮

A_ID	YEAR	PROGRAM	DEGREE	pS
001	1999	CS	MS	1.0
002	1995	CS	BS	1.0
003	1990	MAT	MS	0.5
003	2000	CS	MS	0.5
004	1992	PHY	BS	1.0
005	2000	ACC	BS	1.0
⋮	⋮	⋮	⋮	⋮

A_ID	A_LN	pS
001	Davolio	0.5
001	Luise	0.5
002	Fuller	1.0
003	Leverling	1.0
004	Peacock	1.0
005	Buchanan	1.0
⋮	⋮	⋮

A_ID	PHONE	ADDRESS	pS
001	(206)555-9857	507 - 20th Ave. E.	1.0
002	(206)555-9482	908 W. Capital Way	1.0
003	(206)555-3412	722 Moss Bay Blvd.	1.0
004	(206)555-8122	4110 Old Redmond Rd.	1.0
005	(718)555-4848	14 Garrett Hill	0.5
005	(718)555-4848	405 E 16 Street	0.5
⋮	⋮	⋮	⋮

Figure 6. Probabilistic ALUMNI relation decomposed into four relations

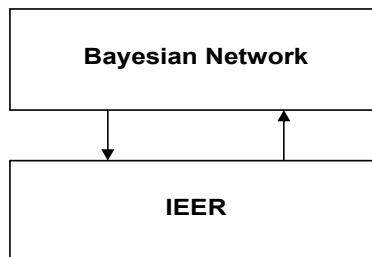


Figure 7. Two layer architecture for semiautomatic logical design software system.

### (3) Output of *LogicDT.PDB* – XML Outputs

There are two types of XML outputs could be generated for the final relations in the *IEDR* layer. The first type of XML output is *extracting XML documents from the IERD* [1]; usually they are the XML schema documents with a specified entity as the root in the IERD. The second type of XML is an independent XML output document for each relation. For example, XML output for relations *ALUMNI-A\_FN*, *ALUMNI-A\_LN* are illustrated in Figures 8(a) and 8(b).

Note that in *LogicDT.PDB*, if a user is not satisfied with the final relations in the IEDR layer, he or she could modify the specified information of IERD and through the normalization layer until the results of database relations are completely satisfied by the user (i.e., database application designer).

## IV. CONCLUSIONS AND FUTURE WORK

In this work, we proposed two-layer architecture for semiautomatic logical design software system *LogicDT.PDB* development.

There are several directions to extend this research work. First, one could find out more appropriate notation for IERD to represent consolidation operation for probabilistic databases. Also, it is very important that one could develop an

efficient and meaningful mechanism for generating a consolidated view from heterogeneous data sources for a relation. Second, we could develop different algorithms to assign probability values to data. One example we have seen in the Figure 3. Third, we should figure out how many data sources that *LogicDT.PDB* can support. Finally, we could develop different mechanisms to perform the conversion of decomposed relations in the IERD layer to the different types of XML documents. Some of these research works could be found in [8].

## V. REFERENCES

- [1] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, Fifth Edition, Addison-Wesley, 2006.
- [2] S. W. Dietrich, S. D. Urban, *An Advanced Course in Database Systems: Beyond Relational Databases*, Prentice Hall, 2005.
- [3] J. Ullman and J. Widom, *Database Systems: The Complete Book*, Prentice Hall, 2002.
- [4] D. Dey and S. Sarkar, "Generalized Normal Forms for Probabilistic Relational Data," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 14, No. 3, May/June, 2002, pp. 485 - 497.
- [5] A. L. P. Chen, J. S. Chiu and F. S. C. Tseng, "Evaluating Aggregate Operations over Imprecise Data", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 8, No. 2, April 1996, pp. 273- 284.
- [6] R. Cavallo and M. Pittarelli, "The Theory of Probabilistic Databases," *Proceedings of the 13th International Conference on Very Large Data Bases*, 1987.

[7] D. Barbara, H. Garcia-Molina, and D. Porter, "The Management of Probabilistic Data," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 4, No. 5, October, 1992.

[8] Fahmeed Afzal, *Logical Design Methodology for Probabilistic Relational Databases*, Master Thesis, Department of Computer Science, Long Island University, Brooklyn, New York, 2006.

```
- <Alumni_FN>
- <RECORD>
  <A_ID>001</A_ID>
  <A_FN>Nancy</A_FN>
  <pS>1.0</pS>
</RECORD>
- <RECORD>
  <A_ID>002</A_ID>
  <A_FN>Andrew</A_FN>
  <pS>1.0</pS>
</RECORD>
- <RECORD>
  <A_ID>003</A_ID>
  <A_FN>Janet</A_FN>
  <pS>1.0</pS>
</RECORD>
- <RECORD>
  <A_ID>004</A_ID>
  <A_FN>Margaret</A_FN>
  <pS>1.0</pS>
</RECORD>
- <RECORD>
  <A_ID>005</A_ID>
  <A_FN>Steven</A_FN>
  <pS>1.0</pS>
</RECORD>
</Alumni_FN>
```

Figure 8 (a). XML output for Relation ALUMNI-A\_FN

```
- <Alumni_LN>
- <RECORD>
  <A_ID>001</A_ID>
  <A_LN>Davolio</A_LN>
  <pS>1.0</pS>
</RECORD>
- <RECORD>
  <A_ID>001</A_ID>
  <A_LN>Luise</A_LN>
  <pS>1.0</pS>
</RECORD>
- <RECORD>
  <A_ID>002</A_ID>
  <A_LN>Fuller</A_LN>
  <pS>1.0</pS>
</RECORD>
- <RECORD>
  <A_ID>003</A_ID>
  <A_LN>Leverling</A_LN>
  <pS>1.0</pS>
</RECORD>
- <RECORD>
  <A_ID>004</A_ID>
  <A_LN>Peacock</A_LN>
  <pS>1.0</pS>
</RECORD>
- <RECORD>
  <A_ID>005</A_ID>
  <A_LN>Buchanan</A_LN>
  <pS>1.0</pS>
</RECORD>
</Alumni_LN>
```

Figure 8 (b). XML output for Relation ALUMNI-A\_LN