Modeling Fuzzy Data with Fuzzy Data Types in Fuzzy Database and XML Models

Li Yan
School of Software, Northeastern University, China

Abstract: *Various fuzzy data models such as fuzzy relational databases, fuzzy object-oriented databases, fuzzy object-relational databases and fuzzy XML have been proposed in the literature in order to represent and process fuzzy information in databases and XML. But little work has been done in modeling fuzzy data types. Actually in the fuzzy data models, each fuzzy value is associated with a fuzzy data type. Explicit representations of fuzzy data types are the foundation of fuzzy data processing. To fill this gap, in this paper, we propose several fuzzy data types, including fuzzy simple data types, fuzzy collection data types and fuzzy defined data types. We further investigate how to declare the fuzzy data types in the fuzzy object-oriented database model and fuzzy XML Schema. The proposed fuzzy data types can meet the requirement of modeling fuzzy data in the fuzzy databases and fuzzy XML.*

Keywords: *Database models, fuzzy data, fuzzy data types, fuzzy databases, fuzzy XML, modeling.*

Received December 7, 2011; accepted May 22, 2012; published online August 5, 2012

1. Introduction

One of the major areas of database research has been the continuous effort to enrich existing database models with a more extensive collection of semantic concepts. Database models have developed from hierarchical and network database models to the Relational Data Base (RDB) model. Currently, computers and databases are widely being applied for non-traditional applications such as CAD/CAM, biology, medicine, multimedia, internet and so on. In order to overcome the limitations of RDBs in these data- and knowledge-intensive applications, some non-traditional data models such as the Object-Oriented Data Base (OODB) model and the Object-Relational Data Base (ORDB) model have been proposed [5]. In addition, with the prompt development of the Internet, the requirement of managing information based on the Web has attracted much attention both from academia and industry. eXtensible Markup Language (XML) is widely regarded as the next step in the evolution of the World Wide Web, and has been the de-facto standard for information exchange over the Web. In the data models above, a common core is data representations, which are the foundation of data processing. Different databases provide diverse data types for expressing various data.

In real-world applications, information is often imperfect, and human knowledge and natural language have a big deal of imprecision and vagueness [3]. Traditional database models assume that the models are a correct reflection of the world and further assume that the stored data is known, accurate and complete. It is rarely the case in real life that all or most of these assumptions are met. So, the fuzzy sets [21] and possibility theory [25] are embedded in many of the existing approaches to deal with imperfect information. In the context of relational model, fuzzy information has been extensively investigated [2, 16, 17, 18]. Also, to model complex objects with imprecision and uncertainty, current efforts have concentrated on the fuzzy OODBs [1, 6, 8, 9, 13] and the fuzzy ORDB [4]. More recently, there are few studies on fuzzily extending XML towards the representation of imprecise and uncertain concepts [7, 10, 15, 19, 20]. Ones can refer to [11, 12] for recent surveys of these fuzzy database models as well as [14] for recent research work of soft computing in XML data management.

Various fuzzy database models have been proposed to model and handle imprecise information in databases, but generally fuzzy data in these fuzzy database models are simply regarded as fuzzy sets or possibility distributions, kind of special values. Relative little work has been carried out in further investigating the forms and types of fuzzy data. In fact, fuzzy data are closely associated with the fuzzy data and fuzzy data types decide the forms of fuzzy data. This likes the situation that crisp data are decided by the corresponding data types. Without a support of fuzzy data types, it is eventually impossible to deal with fuzzy data. It is needed to take into account fuzzy data types while modeling fuzzy data. In this paper, we propose several fuzzy data types, including the fuzzy simple data types for the RDBs, the fuzzy collection data types for the fuzzy OODBs, and the fuzzy defined data types for the fuzzy XML as well as the fuzzy OODBs. We investigate how to declare the fuzzy data types in the fuzzy OODB model and fuzzy XML Schema.

The remainder of this paper is organized as follows: Section 2 discusses the fuzzy sets and fuzzy data representations in the fuzzy databases. In section 3, the fuzzy data with fuzzy data types, including the fuzzy simple data types, fuzzy collection data types and the fuzzy defined data types, are discussed. Fuzzy data type modeling in fuzzy database and XML models is investigated in Section 4. Section 5 concludes this paper.

2. Fuzzy Sets and Fuzzy Data Representations

Different models have been proposed to handle different categories of data quality (or the lack there of). Many current approaches to imprecision and uncertainty are based on the theory of fuzzy sets [21].

Let U be a universe of discourse and F be a fuzzy set in U. A membership function is defined for F as follows:

$$\mu_F:\ U \rightarrow [0,\ 1] \qquad (1)$$

here $\mu_F(u)$ for each $u \in U$ denotes the membership degree of u in the fuzzy set F. Thus, the fuzzy set F is described with the equation 2:

$$F = \{(u_1, \mu_F(u_1)), (u_2, \mu_F(u_2)), ..., (u_n, \mu_F(u_n))\} \qquad (2)$$

when the membership degree $\mu_F(u)$ above is explained to be a measure of the possibility that a variable X has the value u, where X takes values in U, a fuzzy value is described by a possibility distribution π_X [25].

$$\pi_X = \{(u_1, \pi_X(u_1)), (u_2, \pi_X(u_2)), ..., (u_n, \pi_X(u_n)) \qquad (3)$$

In the equation 3, $\pi_X(u_i)$, $u_i \in U$ denotes the possibility that u_i is true. Let π_X be the possibility distribution representation for the fuzzy value of a variable X. It means that the value of X is fuzzy, and X may take one from some possible values (u_1, u_2, ..., u_n) and each one (say u_i) taken possibly is associated with its possibility degree (say $\pi_X(u_i)$).

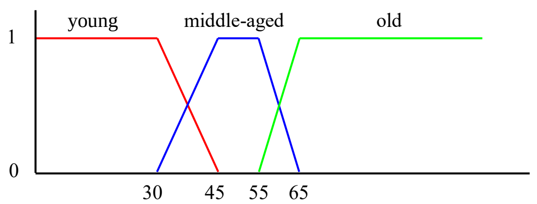

Figure 1. Representations of fuzzy age data.

Fuzzy sets and possibility theory have been used for extending database models. In the fuzzy database models, linguistic labels [22, 23, 24] are generally used to represent fuzzy data, for example, young, middle-aged and old for people's age. But the semantics of the linguistic labels are defined by using fuzzy sets or possibility distributions over a domain. Let us look at the example in Figure 1, in which three linguistic labels young, middle-aged and old are fuzzy values and their membership functions are defined over domain {0, 1, 2, 3, ..., 120}. Also, these three membership functions can simply be expressed using (0, 0, 30, 45), (30, 45, 55, 65) and (55, 65, 120, 120), respectively.

Each datum (or value) is associated with an attribute in the databases and with an element in XML. In the databases, for example, a value is an attribute values, and each attribute has a data type, which is used to represent a domain of instance values. The domain is a set of possible values associated with the attribute and the data type. Let A be an attribute with a given data type (say integer type), $dom(A)$ denote the well-defined domain of attribute A, and a_i be the value of attribute A. Then we have $a_i \in dom(A)$ and a_i must be a number of integer.

In the context of fuzzy databases, attribute values may be fuzzy values. We call the attributes taking fuzzy values fuzzy attributes. A fuzzy attribute value should come from its attribute domain also, but this attribute domain contains a set of fuzzy subsets built over the classical attribute domain (called crisp attribute domain). This set of fuzzy subsets is called fuzzy attribute domain. So for a fuzzy attribute, its attribute domain should be the union of its crisp attribute domain and its fuzzy attribute domain [13]. Formally let A be a fuzzy attribute and then we have:

$$dom(A) = cdom(A) \cup fdom(A) \qquad (4)$$

In the equation 4, $cdom(A)$ and $fdom(A)$ denote the crisp attribute domain and fuzzy attribute domain, respectively. For $cdom(A)$, it is a set of $\{c_1, c_2, \ldots, c_k\}$, where $c_l (1 \leq l \leq k)$ is a crisp value with a given data type. Similarly, $fdom(A) = \{f_1, f_2, \ldots, f_m\}$ is a set of fuzzy subsets, where f_j $(1 \leq j \leq m)$ is a fuzzy value and defined over $cdom(A)$. At this point, a value of attribute A may be a crisp value or a fuzzy value. So, it can be declared that a fuzzy attribute must have a fuzzy data type, which underlying data type is the corresponding data type of the crisp attribute domain. For the fuzzy integer, for example, its underlying data type is integer.

3. Fuzzy Data with Fuzzy Data Types

In the classical databases, some simple data types such as integer and real are redefined and an attribute value can only be an atomic value with the given sample data type. In the OODBs, in addition to atomic value, an attribute value may be a complex value and so complex data types such as collection types are needed. Also, the defined data types are needed in the OODBs and XML Schema.

Fuzziness may exist in the simple data types, the collection data types and the defined data types. Then we have fuzzy simple types, fuzzy collection types and fuzzy defined types.

3.1. Fuzzy Simple Data Types

Simple data types (also, known as atomic data types) are the basic data types predefined by Data Base Management Systems (DBMSs) and XML Schema. The RDB model, OODB model and XML model (Schema) all contain the simple data types. Depending on different systems, there are different kinds of simple data types such as number data type, real data type, integer data type, Boolean data type, string data type, and so on. Basically we classify the simple data types into numerical data types and categorical data types.

A fuzzy simple data type corresponds to a set of fuzzy subsets, which are defined over the corresponding simple data type. For a fuzzy numerical data type, it corresponds to a domain which is a set of fuzzy subsets of the universe integer or real numbers. These fuzzy subsets are generally represented by linguistic labels, which are defined with fuzzy sets or possibility distributions over the corresponding crisp numerical types.

In the example of Figure 1, for example, Age has a fuzzy numerical data type, in which young, middle-aged and old are defined over crisp numerical type.

For a fuzzy categorical data type, it corresponds to a domain which is a set of fuzzy subsets of the universe strings. These fuzzy subsets may be represented by linguistic labels or directly by fuzzy sets or possibility distributions. If linguistic labels are used, they should be also, defined with fuzzy sets or possibility distributions over the corresponding crisp data types.

Let us look at an example. Suppose that we have a set of fuzzy subsets to represent fuzzy grades: {{(1.0, excellent), (1.0, very good), (1.0, good), (0.5, more or less good)}, {(0.5, more or less good), (1.0, common), (0.5, more or less bad)}, {(0.5, more or less bad), (1.0, bad), (1.0, very bad), (1.0, too bad)}}. In this set, a subset, say {(1.0, excellent), (1.0, very good), (1.0, good), (0.5, more or less good)}, is a possibility distribution, which is defined over {excellent, very good, good, more or less good, common, more or less bad, bad, very bad, too bad}.

3.2. Fuzzy Collection Data Types

Collection types are used to represent ordered or unordered collections, which may have fixed or variable sizes. In addition, collection types may allow duplicate or unduplicated collections. The underlying data types of collection data types can be simple types or even another collection types. When the fuzzy simple data types are used as the underlying data types of collection data types, we have fuzzy collection data types. So, the fuzziness of a fuzzy collection data type comes from the fuzziness of its underlying data type. Suppose that we have a set value [very young, young, 32, more or less young] with fuzzy collection data type, which underlying data type is fuzzy numerical. Here two linguistic labels very young and more or less young are fuzzy values. Since very young and more or less young can be regarded as the modified (composite) fuzzy label of young, their membership functions can be computed through the membership function of young. Let the membership function of young be μ_{young}, then we have the followings:

$$\mu_{very\ young}\ (u) = (\mu_{young}\ (u))^2 \tag{5}$$

$$\mu_{more\ or\ less\ young}\ (u) = (\mu_{young}\ (u))^{1/2} \tag{6}$$

3.3. Fuzzy Defined Data Types

Generally two kinds of the defined data types can be identified in databases and programming languages: the first one is called the reduced defined data type and the second one is called the aggregative defined data type. First let us look at the reduced defined data type. A reduced defined data type is formed by putting some constraints on the underlying data type, in which the underlying data type may be a simple type or a collection type. For the data type of integer, for example, we may define a reduced defined data type named MyInterger based on the data type of integer, in which we put a constraint $0 \leq MyInterger \leq 150$ on the data type of integer. Formally, let B be an underlying data type and S be a reduced defined data type of B. It is clear that:

$$dom\ (S) \subseteq dom\ (B) \tag{7}$$

In the equation 7, $dom(S)$ and $dom(B)$ are used to express the domains of S and B, respectively. Actually an underlying data type and its reduced defined data type has a relationship of supertype and subtype. Here the underlying data type is the supertype and the reduced defined data type is the subtype.

Now let us look at the aggregative defined data type. An aggregative defined data type creates a type that defines the properties of real world or conceptual objects. The properties are mainly defined in terms of attributes, and each attribute have a value domain, i.e., data type. We call these data types of the attributes in the aggregative defined data type as component data types. The component data types can be the sample data types, the collection data types and even other defined data types. For example, we may define an aggregative defined data type named Product, which contains four component data types: a categorical data type for its property of ID and three real data types for its properties of length, width and height, respectively.

Formally, let C be a component data type and A bean aggregative defined data type containing C. It isclear that:

$$dom\ (C) \in dom\ (A) \tag{8}$$

In the equation 8, $dom(C)$ and $dom(A)$ are used to express the domains of C and A, respectively. Actually the component data types and the corresponding aggregative defined data type have a relationship of aggregation (i.e., a whole-part relationship). Here the component data types are the constituent parts and the reduced defined data type is the aggregate.

Corresponding to two kinds of the defined data types, we have two kinds of fuzzy defined data types, which are the fuzzy reduced defined data type and the fuzzy aggregative defined data type. The fuzziness of a reduced defined data type comes from that of its underlying data type. Suppose we have a fuzzy data type FuzzyInteger, and we may define a fuzzy reduced defined data type named MyFuzzyInterger based on the data type of FuzzyInteger.

The fuzziness of an aggregative defined data type comes from that of its component data types. We know that an attribute in an aggregative defined data type defines a value domain. When this domain is a set of fuzzy subset, the fuzziness of an attribute value appears. Still let us look at the aggregative defined data type Product above. Suppose that three component data types for its properties of length, width and height are fuzzy real data types. At this point, the aggregative defined data type product above becomes a fuzzy aggregative defined data type.

4. Fuzzy Data Type Declarations

When we model fuzzy data in databases models and XML models, it is necessary to explicitly declare the data types of fuzzy data so that the fuzzy data can be dealt with in right ways. In the following, we present how to declare the fuzzy data types in the fuzzy database models and fuzzy XML model, respectively. Since the fuzzy RDBs only contains the simple data types (crisp or fuzzy), we here concentrate on the fuzzy OODB model.

4.1. Fuzzy Data Type Declaration in Fuzzy Object-Oriented Database Model

The core of the fuzzy OODB model is the nation of fuzzy class [13]. The specification of a fuzzy class includes the definition of ISA relationships, attributes and method implementations. Some additional definitions are needed to specify a fuzzy class. First, weights must be assigned to the attributes of the class. In addition, to these common attributes, a new attribute should be added to the class to indicate the membership degree to which an object belongs to the class. If a class is a fuzzy subclass, its superclasses and the degree to which the class is the subclass of the superclasses should also, be illustrated in the specification of the class. Finally, in the definition of a fuzzy class, fuzzy attributes are explicitly indicated and their data types are declared.

For the fuzzy attributes of fuzzy classes, the data types are fuzzy data types based simple data types or collection data types, which allows the representation of descriptive form of imprecise information. Only fuzzy attributes have fuzzy type and fuzzy attributes are explicitly indicated in a fuzzy database model definition. Therefore, in the definition, we declare only the underlying data type (e.g., integer) of fuzzy attributes and the fuzzy domain. A fuzzy domain is a set of possibility distributions or fuzzy linguistic terms (each fuzzy term is associated with a membership function). For example, a fuzzy attribute *Age* is declared as follows:

Age: Fuzzy Domain {very young, young, old, very old}: type of integer

Then an object attribute defined fuzzy type will have either a crisp value or a fuzzy value given in the type definition. For example, age=21 or age=young.

4.2. Fuzzy Data Type Declaration in Fuzzy XML Model

Compared with XML Document Type Definition (DTD), XML Schema can support rich data types. In [10], fuzzy XML DTD is proposed for fuzzy information modeling. Fuzzy XML Schema is investigated in [20], with a focus on the fuzziness of XML document contents. That is the fuzziness of elements and the fuzziness of attribute values in elements. In the following, we discuss the fuzzy data types in the fuzzy XML Schema.

XML schema supports two kinds of different data types: the simple data types and the complex data types based on the sample data types. The sample data types are further divided into two categories: the first one is the built-in data types provided by the XML Schema (e.g., numerical, integer, real and string), which can directly be used in the declarations of elements and attributes, the other one is the defined sample data types, which are defined on the basis of the underlying data types by users using the simpleType structure. Here the underlying data types may be the built-in data types or the defined simple data types. In the XML Schema, the simpleType supports three kinds of mechanisms to define new simple data types from the underlying data types, which are restriction, list and union. Such mechanisms in the XML Schema provide a natural and effective method to define the fuzzy data types. Since the complex data types are created by the simple data types, here we only discuss the simple data types of fuzzy element values in the XML Schema.

For the fuzzy values of elements, their data types are the fuzzy sample data types, in which the representations of imprecise and uncertain information are allowed. Only fuzzy elements have fuzzy data types, which need to be explicitly indicated in the XML Schema. For this purpose, in addition that the element is declared to be a fuzzy element, like what we do in the above, it is needed to declare the fuzzy domain of the fuzzy element. A fuzzy domain is a set of possibility distributions or fuzzy linguistic terms, in which each term is associated with a membership function, and must be based on an underlying data type (i.e., integer). A fuzzy domain can be defined using the structure of simpleType in XML Schema.

Let us look at a fuzzy element Age. Suppose that its fuzzy domain is {very young, young, old, very old} based on integer data type, and this integer data type has a constraint of being greater than or equal to 0 and less than or equal to 150. First, we define an integer data type named AgeInteger, which meets the constrain of being greater than or equal to 0 and less than or equal to 150.

```
<xsd:simpleType name=”AgeInteger”>
  <xsd:restriction base=”xsd:integer”>
      <xsd:minExclusive value=”0”/>
      < xsd:maxExclusive value =”150”/>
</xsd: simpleType>
```

Based on integer data type AgeInteger defined above, a fuzzy integer data type named FuzzAgeInteger is constructed using the membership functions of the fuzzy terms. A value of the fuzzy integer data type FuzzAgeInteger is a set, in which each element is a pair of data with form say (20, 0.8), being consisted of the values of integer data type AgeInteger and the corresponding membership degrees. This set is actually a fuzzy set with a semantic interpretation of possibility distribution, and formally is represented with a linguistic term (say young). It should especially pointed out that fuzzy integer data type FuzzAgeInteger is not a restricted form of integer data type AgeInteger, but an extended data type based on AgeInteger, which underlying data type is AgeInteger. Since current XML Schema can only put restrictions on the underlying data types when it supports users to define simple data types, and cannot extend the value space of data types. So, fuzzy integer data type FuzzAgeInteger cannot be defined directly by XML Schema. In real-world applications, it is very useful to construct new simple data types with several underlying data types by means of expanding the value spaces of data types. Current version of XML fails to do that. For the complex data types, the focus of XML Schema is on defining its complex structures, not on defining the value spaces of the complex data types. With fuzzy integer data type FuzzAgeInteger, the fuzzy domain of FuzzAgeInteger is declared as follows:

```
<xsd:simpleType name=”FuzzAgeDomain”>
  <xsd:restriction base=”FuzzAgeInteger”>
      <xsd:enumeration value=”very young”/>
      <xsd:enumeration value=”young”/>
      <xsd:enumeration value=”old”/>
      <xsd:enumeration value=”very old”/>
</xsd: simpleType>
```

Since the value of a fuzzy element may take a crisp or fuzzy value, the data type of fuzzy element Age can be declared with the following union type:

```
<xsd:simpleType name=”AgeType”>
  <xsd:union memberType=”AgeInteger
      FuzzAgeDomain”/>
</xsd: simpleType>
```

Finally, fuzzy element Age is declared as follows:

```
<xsd:element name=”age” type=”agetype”
  minOccurs=”0” maxOccurs=”1”/>
```

Its corresponding value may be a crisp value (say age=21) or a fuzzy value (say age=young). It can be seen from the above example that XML Schema provides a flexible mechanism for defining diverse data types. This mechanism is very useful for representing fuzzy data. In the above, we use an example of defining a fuzzy integer to demonstrate the procedure of declaring fuzzy data types. Following a similar way, we can declare other fuzzy data types.

5. Conclusions

In the real world, human knowledge and natural language have a big deal of imprecision and vagueness. Fuzzy technologies have been extensively used for intelligent data processing and intelligent systems. Various fuzzy data models such as fuzzy RDBs, fuzzy OODBs, fuzzy ORDBs and fuzzy XML have been proposed to represent and process fuzzy information in databases and XML. Despite fuzzy data has been employed to model and handle imprecise information in data models since the theory of fuzzy sets was introduced, relative little work has been carried out in further investigating the types of fuzzy data towards the representation and processing of fuzzy data.

In this paper, we focus on modeling fuzzy data with fuzzy data types in the fuzzy databases and fuzzy XML. We identify several fuzzy data types, including fuzzy simple data types, fuzzy collection data types and fuzzy defined data types. For the purpose of representing and processing fuzzy data, we elaborate the explicit declarations of the fuzzy data types in the fuzzy OODB model and fuzzy XML Schema, respectively. In future work, we will develop prototypes of fuzzy OODB system and fuzzy XML database system, and then complete the evaluation experiment of fuzzy data type modeling. Also, we will investigate how to query the fuzzy databases and fuzzy XML with fuzzy data types.

Acknowledgment

This research work was supported by the National Natural Science Foundation of China (60873010, and 61073139 and 61370075).

References

[1] Bordogna G., Pasi G., and Lucarella D., “A Fuzzy Object-Oriented Data Model for Managing Vague and Uncertain Information,” *International Journal of Intelligent Systems*, vol. 14, no. 7, pp. 623-651, 1999.

[2] Buckles P. and Petry E., "A Fuzzy Representation of Data for Relational Database," *Fuzzy Sets and Systems*, vol. 7, no. 3, pp. 213-226, 1982.

[3] Chelliah M., Sankaran S., Prasad S., Gopalan N., and Sivaselvan B., "Routing for Wireless Mesh Networks with Multiple Constraints using Fuzzy Logic," *International Arab Journal of Information Technology*, vol. 9, no. 1, pp. 1-8, 2011.

[4] Cuevasa L., Marínb N., Ponsb O., and Vilab A., "pg4DB: A Fuzzy Object-Relational System," *Fuzzy Sets and Systems*, vol. 159, no. 12, pp. 1500-1514, 2008.

[5] Dietrich W. and Urban D., *Fundamentals of Object Databases: Object-Oriented and Object-Relational Design*, Morgan & Claypool Publishers, USA, 2010.

[6] Dubois D., Prade H., and Rossazza P., "Vagueness, Typicality, and Uncertainty in Class Hierarchies," *International Journal of Intelligent Systems*, vol. 6, no. 2, pp. 167-183, 1991.

[7] Gaurav A. and Alhajj R., "Incorporating Fuzziness in XML and Mapping Fuzzy Relational Data into Fuzzy XML," *in Proceedings of the ACM Symposium on Applied Computing*, France, pp. 456-460, 2006.

[8] George R., Srikanth R., Petry E., and Buckles P., "Uncertainty Management Issues in the Object-Oriented Data Model," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 2, pp. 179-192, 1996.

[9] Gyseghem V. and Caluwe R., "Imprecision and Uncertainty in UFO Database Model," *Journal of the American Society for Information Science*, vol. 49, no. 3, pp. 236-252, 1998.

[10] Ma M. and Yan L., "Fuzzy XML Data Modeling with the UML and Relational Data Models," *Data & Knowledge Engineering*, vol. 63, no. 3, pp. 972-996, 2007.

[11] Ma M. and Yan L., "A Literature Overview of Fuzzy Database Models," *Journal of Information Science and Engineering*, vol. 24, no. 1, pp. 189-202, 2008.

[12] Ma M. and Yan L., "A Literature Overview of Fuzzy Conceptual Data Modeling," *Journal of Information Science and Engineering*, vol. 26, no. 2, pp. 427-441, 2010.

[13] Ma M., Zhang J., and Ma Y., "Extending Object-Oriented Databases for Fuzzy Information Modeling," *Information Systems*, vol. 29, no. 5, pp. 421-435, 2004.

[14] Ma M. and Yan L., *Soft Computing in XML Data Management*, Springer-Verlag, Germany, 2010.

[15] Oliboni B. and Pozzani G., "Representing Fuzzy Information by using XML Schema," *in Proceedings of the 19th International Workshop on Database and Expert Systems Application*, Italy, pp. 683-687, 2008.

[16] Prade H. and Testemale C., "Generalizing Database Relational Algebra for the Treatment of Incomplete or Uncertain Information and Vague Queries," *Information Sciences*, vol. 34, no. 2, pp. 115-143, 1984.

[17] Raju N. and Majumdar K., "Fuzzy Functional Dependencies and Lossless Join Decomposition of Fuzzy Relational Database Systems," *ACM Transactions on Database Systems*, vol. 13, no. 2, pp. 129-166, 1988.

[18] Umano M. and Fukami S., "Fuzzy Relational Algebra for Possibility-Distribution-Fuzzy-Relational Model of Fuzzy Data," *Journal of Intelligent Information Systems*, vol. 3, no. 1, pp. 7-27, 1994.

[19] Üstünkaya E., Yazici A., and George R., "Fuzzy Data Representation and Querying in XML Database," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 15, no. Supplement-1, pp. 43-57, 2007.

[20] Yan L., Ma M., and Liu J., "Fuzzy Data Modeling Based on XML Schema," *in Proceedings of the ACM International Symposium on Applied Computing*, USA, pp. 1563-1567, 2009.

[21] Zadeh A., "Fuzzy Sets," *Information and Control*, vol. 8, no. 3, pp. 338-353, 1965.

[22] Zadeh A., "The Concept of Linguistic Variable and its Application to Approximate Reasoning I," *Information Sciences*, vol. 8, no. 3, pp. 199-251, 1975.

[23] Zadeh A., "The Concept of Linguistic Variable and its Application to Approximate Reasoning II," *Information Sciences*, vol. 8, no. 4, pp. 301-357, 1975.

[24] Zadeh A., "The Concept of Linguistic Variable and Its Application to Approximate Reasoning III," *Information Sciences*, vol. 9, no. 1, pp. 43-80, 1975.

[25] Zadeh A., "Fuzzy Sets as a Basis for a Theory of Possibility," *Fuzzy Sets and Systems*, vol. 1, no. 1, pp. 3-28, 1978.

Li Yan received her PhD degree from Northeastern University, China. She is currently an associate professor at Northeastern University, China. Her research interests include database modeling, XML data management, as well as imprecise and uncertain data processing. She has published some papers in international journals such as Information Systems Frontiers, International Journal of Intelligent Systems and Integrated Computer-Aided Engineering. Also, she has published several edited books with Springer and IGI Global.