2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Reengineering Fuzzy Nested Relational Databases into Fuzzy XML Model

Weijun Li
College of Information Science and Engineering
Northeastern University
Shenyang, China
liweijun51@163.com

Xu Chen
College of Information Science and Engineering
Northeastern University
Shenyang, China
fallback@vip.qq.com

Z. M. Ma
College of Information Science and Engineering
Northeastern University
Shenyang, China
zongminma@ise.neu.edu.cn

***Abstract*—Data interchange on the Web is a common task today and XML has been the de-facto standard of information representation and exchange over the Web. Also information imperfection is inherent in the real-world applications. Fuzzy information has been extensively investigated in the context of database models. Also fuzzy XML modeling recently receives more attention. In order to present fuzzy data from the fuzzy databases with XML, this paper concentrates on fuzzy information modeling in the fuzzy XML model and the fuzzy nested relational database model. The formal approach to mapping a fuzzy nested relational database (FNRDB) schema into a fuzzy DTD model is developed in the paper.**

Keywords—fuzzy XML; fuzzy nested relational mode; fuzzy DTD; reengineering

I. INTRODUCTION

Nowadays data interchange between different data models is a common task. It is especially true in Web-based applications. XML has been the de-facto standard for data representation and exchange on the Web mainly as it is a self-descriptive format, supports a flexible representation of data, and it is an open and free pattern [26]. To manage XML data, it is necessary to integrate XML and databases [2]. The mapping form databases into XML can benefit database interoperability over the Web in a standard form. For this purpose, various databases, including relational (e.g., [5, 14, 15]), object-oriented (e.g., [22, 23, 26]) and object-relational (e.g., [5]) databases, have been used for mapping to XML document.

At the same time, some data are inherently imprecise and uncertain since their values are subjective in the real world applications. Fuzzy information has been extensively investigated in the context of relational model [4, 27, 28, 30]. Also in order to model uncertain data and complex-valued attributes as well as complex relationships among objects, current efforts have concentrated on the conceptual data models [21, 34], the fuzzy nested relational data model (also known as an NF^2 data model) [35] and the fuzzy object-oriented databases [3, 11, 12, 20]. More recently, the fuzzy object-relational databases are proposed [8] which combine both characters of fuzzy relational databases and fuzzy object-oriented databases. Ones can refer to [18, 19] for recent surveys of these fuzzy data models.

Despite fuzzy values have been employed to model and handle imprecise information in databases since Zadeh introduced the theory of fuzzy sets [36], relative little work has been carried out in extending XML towards the representation of imprecise and uncertain concepts. Abiteboul et al. [1] provide a model for XML documents and DTDs (Document Type Definitions) and a representation system for XML with incomplete information. The representations of probabilistic data in XML are proposed in other previous research papers, such as [13, 24, 29, 31]. Without presenting XML representation model, the data fuzziness in XML document is discussed directly according to the fuzzy relational databases in [10] and the simple mappings from the fuzzy relational databases to fuzzy XML document are provided also. In [25] a XML Schema definition for representing fuzzy information is proposed. They adopt the data type classification for the XML data context. A fuzzy XML data model which is based XML DTD is proposed in [17], in which the mapping of the fuzzy XML DTD from the fuzzy UML data model and to the fuzzy relational database schema are discussed, respectively. In [33], a fuzzy XML data model based on XML Schema is developed. The algebraic operations in fuzzy XML are discussed in [16].

Being the extension of relational data model, the NF^2 database model is able to handle complex-valued attributes and may be better suited to some complex applications such as office automation systems, information retrieval systems and expert database systems [35]. In [7], the fuzzy NF^2 database model is proposed for managing uncertainties in images. More recently, the formal mapping of fuzzy XML into fuzzy nested relational databases is investigated in [32]. This paper concentrates on fuzzy information modeling in the fuzzy XML model and the fuzzy nested relational database model. In particular, the formal approach to mapping a fuzzy nested relational database (FNRDB) schema to a fuzzy XML DTD model is developed.

The remainder of this paper is organized as follows. Section 2 presents fuzzy sets and possibility distributions. The fuzzy

The work was supported by National Natural Science Foundation of China (61073139 and 61370075) and by Program for New Century Excellent Talents in University (NCET-05-0288).

XML data model and fuzzy nested relational databases are introduced in Section 3. In Section 4, the approaches to mapping the fuzzy nested relational schema to the fuzzy XML model are developed. Section 5 concludes this paper.

II. FUZZY SETS AND POSSIBILITY DISTRIBUTIONS

The concept of fuzzy sets was originally introduced by Zadeh [36]. Let U be a universe of discourse and F be a fuzzy set in U. A membership function

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

is defined for F, where $\mu_F(u)$, for each $u \in U$, denotes the membership degree of u in the fuzzy set F. Thus, the fuzzy set F is described as follows:

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

The fuzzy set F is consisted of some elements just like the conventional set. But, not being the same as the conventional set, each element in F may or may not belong to F, having a membership degree to F which needs to be explicitly indicated. So in F, an element (say u_i) is associated with its membership degree (say $\mu_F(u_i)$), and they occur together in form of (u_i, $\mu_F(u_i)$). When the membership degrees that all elements in F belong to F are exactly 1, the fuzzy set F reduces to a conventional one.

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 [37].

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

Here, $\pi_X(u_i)$, $u_i \in U$ 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, ..., and u_n and each one (say u_i) taken possibly is associated with its possibility degree (say $\pi_X(u_i)$).

III. REPRESENTATION OF FUZZY DATA IN XML AND NESTED RELATIONAL DATABASES

A. Fuzzy XML Model

Fuzzy values can be represented with fuzzy sets. Two kinds of fuzziness can be identified in XML documents. The first is the fuzziness in elements (we use membership degrees associated with such elements) and the second is the fuzziness in attribute values of elements (we use fuzzy set to represent such values). Note that, for the latter, there exist two interpretations on it (i.e., disjunctive semantics and conjunctive semantics) and they may occur in child elements with or without further child elements in the ancestor-descendant chain. The basic data structure of fuzzy XML data model is a data tree [16].

Definition: Let V be a finite set (of vertices), $E \in V \times V$ be a set (of edges) and $\ell: E \rightarrow \Gamma$ be a mapping from edges to a set Γ of strings called labels. The triple $G = (V, E, \ell)$ is an edge labeled directed graph.

Based on the data tree, we introduce the formal definition of fuzzy XML data tree.

Definition: Fuzzy XML data tree F is a 6-tuple, $F = (V, \psi, \ell, \tau, \kappa, \delta)$ where

-- $V = \{V_1, \ldots, V_n\}$ is a finite set of vertices.

-- $\psi \subset \{(V_i, V_j) \mid V_i, V_j \in V\}$, (V, ψ) is a directed tree.

-- $\ell: V \rightarrow (L \cup \{\text{null}\})$, here L is a set of labels. For each object $v \in V$ and each label $\nabla \in L$, $\ell(v, \nabla)$ specifies the set of objects that may be children of v with label ∇.

-- $\tau \rightarrow T$, T is a set of types.

-- κ is a mapping which constrains the number of children with a given label. Also κ associates with each object $v \in V$ and each label $\nabla \in L$, an integer-valued interval function. $\kappa(v, \nabla) = [\min, \max]$, where $\min \geq 0$, $\max \geq \min$. We use κ to represent the lower and upper bounds.

-- δ is a mapping from the set of objects $v \in V$ to local possibility functions. It defines the possibility of a set of children of an object existing given that the parent object exists.

Definition: Suppose $F = (V, \psi, \ell, \tau, \kappa, \delta)$ and $f' = (V', \psi', \ell', \tau', \kappa', \delta')$ are two fuzzy data trees. f' is a sub-tree of F, written $f' \propto F$, when

-- $V' \subseteq V$, $\psi' = \psi \cap V' \times V'$.

-- if $i \in V'$ and $(j, i) \in \psi$, then $j \in V'$.

-- ℓ' and τ' indicate the restriction of ℓ and τ to the nodes in V', respectively.

-- $\kappa' \in \kappa$.

Definition: Let fuzzy data trees $f_1 = (V_1, \psi_1, \ell_1, \tau_1, \kappa_1, \delta_1)$ and $f_1 = (V_2, \psi_2, \ell_2, \tau_2, \kappa_2, \delta_2)$ be the sub-trees of $F = (V, \psi, \ell, \tau, \kappa, \delta)$. f_1 and f_2 are isomorphic (recorded $f_1 \doteq f_2$), when

-- $V_1 \cup V_2 \subseteq V$, $\psi_1 \cup \psi_2 \subseteq \psi$ and $\tau_1 \cup \tau_2 \subseteq \tau$.

-- There is a one-to-one mapping, $\xi_\ell: \ell_1 \rightarrow \ell_2$, which makes $\forall \xi_\ell(\ell_1) = \ell_2$.

Theorem: Fuzzy data tree F and its sub-tree f' are isomorphic.

The above theorem follows the analysis of last two definitions. It is quite straightforward.

The fuzziness in XML document was discussed in [17], in which several fuzzy constructs are introduced for fuzzy XML data modeling. First, a possibility attribute, denoted Poss, which takes a value of [0, 1], is applied. This possibility attribute is applied together with a fuzzy construct called Val to specify the truth degree of a given element belonging to (or being included) in the XML document. Based on pair <Val Poss> and </Val>, possibility distribution for an element can be expressed. Also, possibility distribution can be used to express fuzzy element values. For this purpose, second, a fuzzy construct called Dist is applied to specify a possibility distribution. Typically, a Dist element has multiple Val elements as children, each with an associated possibility. Since we have two types of possibility distribution, the Dist construct should indicate the type of a possibility distribution being disjunctive or conjunctive.

Concerning how to accommodate these fuzzy constructs and correspondingly modify the DTD of XML document, ones can refer to [17].

B. *Fuzzy Nested Relational Model*

A fuzzy NF^2 relational schema is a set of attributes (A_1, A_2, ..., A_n, pM) and their domains are D_1, D_2, ..., D_n, D_0, respectively, where D_i ($1 \le i \le n$) can be one of the following:

(1) The set of atomic values. For each element $a_i \in D_i$, it is a typical simple crisp attribute value.

(2) The set of null values, denoted *ndom*, where null values may be *unk*, *inap*, *nin*, and *onul*.

(3) The set of fuzzy subset. The corresponding attribute value is an extended possibility-based fuzzy data.

(4) The power set of the set in (1). The corresponding attribute value, say a_i, is multivalued one with the form of $\{a_{i1}, a_{i2}, ..., a_{ik}\}$.

(5) The set of relation values. The corresponding attribute value, say ai, is a tuple of the form $<a_{i1}, a_{i2}, ..., a_{im}>$ which is an element of $D_{i1} \times D_{i2} \times ... \times D_{im}$ ($m > 1$ and $1 \le i \le n$), where each D_{ij} ($1 \le j \le m$) may be a domain in (1), (2), (3), and (4) and even the set of relation values.

The domain D_0 is a set of atomic values and each value is a crisp one from the range [0, 1], representing the possibility degree that the corresponding tuple is true in the NF2 relation. We assume that the possibilities of all tuples are precisely one in the chapter. Then for an attribute $Ai \in R$ ($1 \le i \le n$), its attribute domain is formally represented as follows:

$$\tau_i = dom \mid ndom \mid fdom \mid sdom \mid <B_1 : \tau_{i1}, B_2 : \tau_{i2}, \ldots, B_m : \tau_{im}> \quad (4)$$

Here B_1, B_2, …, B_m are attributes.

A relational instance *r* over the fuzzy NF^2 schema ($A_1 : \tau_1$, $A_2 : \tau_2$, ..., $A_n : \tau_n$) is a subset of Cartesian product $\tau_1 \times \tau_2 \times ... \times \tau_n$. A tuple in *r* with the form of $<a_1, a_2, ..., a_n>$ consists of n components. Each component a_i ($1 \le i \le n$) may be an atomic value, null value, set value, fuzzy value, or another tuple.

An example of the fuzzy NF^2 relation is shown in Table 1. It can be seen that *Tank_Id* and *Start_data* are crisp atomic-valued attributes, *Tank_body* is a relation-valued attribute, and *Responsibility* is a set-valued attribute. In the attribute *Tank_body*, two component attributes *Volume* and *Capacity* are fuzzy ones.

TABLE I
PRESSURED AIR TANK RELATION

Tank_Id	Tank_body				Start_Date	Responsibility
	Body_Id	Material	Volume	Capacity		
TA1	BO01	Alloy	about 2.5e+03	about 1.0e+06	1. 01/12/99	John
TA2	BO02	Steel	2. about 2.5e+04	about 1.0e+07	28/03/00	{Tom, Mary}

IV. MAPPING FUZZY NESTED RELATIONAL SCHEMA TO FUZZY XML DTD

To map the fuzzy nested relational databases into the fuzzy XML, first we can map the nested relational database schema into the XML DTD without considering fuzzy information, and then we rebuild the created DTD after considering fuzzy information in the fuzzy nested relational databases. For the former, we have the following rules for the mapping.

Rule 1: for a nested relational schema, we create a root node element in XML model. Its DTD is described as

```
<!ELEMENT root (element*)>
```

Rule 2: for a nested relation, we correspondingly create a non-leaf element node in the XML DTD.

Rule 3: a key in the nested relation can be directly described with the attribute declaration of DTD as follows.

```
<!ATTLIST Ename Aname ID …>
```

Rule 4: for a foreign key in the nested relation, if it is a reference use for single ID, it is described with the attribute declaration of DTD as follows.

```
<!ATTLIST Ename Aname IDREF …>
```

If a foreign key of the nested relation is a reference use for several ID, it is described with the attribute declaration of DTD as follows.

```
<!ATTLIST Ename Aname IDREFS …>
```

Rule 5: a common attribute, which is neither a key nor a foreign key, is described with the attribute declaration of DTD as follows.

<!ELEMENT Ename (original-definition)>

Rule 6: the constraint that attribute values are not empty in the nested relation can be described with the attribute declaration of DTD as follows.

<!ALLIST Ename Aname original-definition #REQUIRED>

Rule 7: for the nested relation which key does not contain foreign key or contains several foreign keys, we create a corresponding sub-element directly under the root element. Its DTD is represented as

<!ELEMENT root (element*)>

Rule 8: for the nested relation r_1 which key contains only one foreign key, let its parent nested relation be r_2. At this point, r_1 is directly mapped into a sub-element of the element mapped from r_2 as follows.

<!ELEMENT element$_2$ (element$_1$*)>

Here element$_2$ and element$_1$ are the elements mapped from r_2 and r_1, respectively.

Rule 9: if two nested relations r_1 and r_2 only have one-to-one relationships and also the foreign key from r_1 to r_2 is not empty, r_1 is directly mapped into a sub-element of the element mapped from r_2 as follows.

<!ELEMENT element$_2$ (element$_1$*)>

Here element$_2$ and element$_1$ are the elements mapped from r_2 and r_1, respectively.

Rule 10: if there are several many-to-one relationships from r_0 to r_1, …, r_k, r_0 is respectively mapped into a sub-element of the elements mapped from r_1, …, r_k as follows.

<!ELEMENT element$_1$ (element$_0$*)>

……

<!ELEMENT element$_k$ (element$_0$*)>

Here element$_0$, element$_1$, …, element$_k$ are the elements mapped from r_0, r_1, …, r_k,, respectively.

Rule 11: if there are one-to-many relationships from nested relation r_1 to nested relation r_2, r_1 and r_2 are all mapped into sub-elements of the root element as follows.

<!ELEMENT root (element$_1$*, element$_2$*)>

Here element$_2$ and element$_1$ are the elements mapped from r_2 and r_1, respectively. Then attribute declarations ID and IDREF are applied in the declarations of element$_1$ and element$_2$, respectively.

Rule 12: if there are many-to-many relationships in nested relation r_1, r_1 is mapped into a sub-element of the root element as follows.

<!ELEMENT root (element$_1$)>

Here element$_1$ is the element mapped from r_1. Then attribute declaration ID is applied in the declaration of element$_1$.

Rule 13: the limitation of default attribute values in the nested relation is represented with attribute declaration in DTD as follows.

<!ALLIST element Aname original-definition “default”>

Rule 14: for each attribute A_i ($1 \le i \le$ k-1) in the nested attribute (A_1, …, A_{k-1}), A_i can be defined empty and then it is mapped into an element, which content is represented as A_i* or A_i.

<!ELEMENT Ename (Aname1*, Aname2+,…)>

For each attribute A_j ($k \le j \le n$), A_j can be defined empty and then it is mapped into an element, which content is represented as A_j? or A_j.

<!ELEMENT Ename (…, Aname1?, Aname2)>

Now we consider possible effects of fuzziness in the nested relations to the DTD created above. At this point, we need to rebuild the created DTD. We have the following rules.

Rule 15: for the nested relation *r* with fuzzy attribute values, if it is mapped into a leaf node element, then it is further rebuilt as follows.

- We create a Dist sub-element node in the element mapped from *r*, and get
  <!ELEMENT element (Dist)>
- We further create a Val sub-element node within the Dist sub-element, which number of occurrence is defined as +, and finally get
  <!ELEMENT Dist (Val+)>
- Within the Val sub-element, we deal with the attributes in *r*. Here we apply the processing procedure which is similar to the processing procedure used for the mapping without fuzziness, but the default value of Val is declared as 1.0. We finally get
  <!ELEMENT Val (original-definition)>
  <!ATTLIST Val Poss CDATA “1.0”>

Rule 16: for the nested relation *r* with fuzzy attribute values, if it is mapped into a non-leaf node element, then it is further rebuilt as follows.

- We create a Dist sub-element node in the element mapped from *r*, and get
  <!ELEMENT element (Dist)>
- We further create a Val sub-element node within the Dist sub-element, which number of occurrence is defined as +, and finally get
  <!ELEMENT Dist (Val+)>
- Within the Val sub-element, we deal with the attributes in *r*. Here we apply the processing procedure which is similar to the processing procedure used for the mapping without fuzziness, but the default value of Val is declared as 1.0. We finally get
  <!ELEMENT Val (element$_1$*, element$_k$+, element$_1$?, element$_1$,…)>
  <!ATTLIST Val Poss CDATA “1.0”>

Applying the rules defined above, we can finally obtain the fuzzy DTD mapped from the fuzzy nested relational databases. The fuzzy nested relational databases shown in Table 1 can be mapped into the fuzzy XML DTD model shown in Fig. 1 and the fuzzy XML document shown in Fig. 2.

```
<!ELEMENT Pressured air tank relation (Dist)>
    <!ATTLIST Tank_Id FID IDREF #REQUIRED>
<!ELEMENT Dist (Val+)>
    <!ATTLIST Dist type (disjunctive)>
<!ELEMENT Val (Tank_body?, Start_Date?, Responsibility?)>
    <!ATTLIST Val Poss CDATA "1.0">
<!ELEMENT Start_Date (#PCDATA)>
        <!ELEMENT Responsibility (Dist)>
            <!ELEMENT Dist (Val+)>
        <!ATTLIST Dist type (conjunctive)>
         <!ELEMENT Val (#PCDATA)>
        <!ATTLIST Val Poss CDATA "1.0">
<!ELEMENT Tank_body (Dist)>
    <!ATTLIST Body_Id FID IDREF #REQUIRED>
<!ELEMENT Dist (Val+)>
    <!ATTLIST Dist type (disjunctive)>
<!ELEMENT Val (Material?, Volume?, Capacity?,)>
    <!ATTLIST Val Poss CDATA "1.0">
<!ELEMENT Material (#PCDATA)>
           <!ELEMENT Volume (Dist)>
            <!ELEMENT Dist (Val+)>
        <!ATTLIST Dist type (disjunctive)>
         <!ELEMENT Val (#PCDATA)>
        <!ATTLIST Val Poss CDATA "1.0">
          <!ELEMENT Capacity (Dist)>
            <!ELEMENT Dist (Val+)>
        <!ATTLIST Dist type (disjunctive)>
         <!ELEMENT Val (#PCDATA)>
        <!ATTLIST Val Poss CDATA "1.0">
```

Fig. 1. Fuzzy XML DTD model mapped from fuzzy nested relational databases in Table 1

```
<Pressured air tank relation Tank_Id = "TA1">
   <Tank_body Body_Id = "BO01">
        <Material>Alloy</Material>
     <Volume>
        <Dist type = "disjunctive">
         about  2.5e+03
       </Dist>
     </Volume>
      <Capacity>
        <Dist type = "disjunctive">
         about  1.0e+06
       </Dist>
     </Capacity>
   </Tank_body>
   <Start_Date>01/12/99</Start_Date>
   <Responsibility>John</Responsibility>
</Pressured air tank relation>
< Pressured air tank relation Tank_Id = "TA2">
   <Tank_body Body_Id = "BO02">
        <Material>Steel</Material>
     <Volume>
        <Dist type = "disjunctive">
         about  2.5e+04
       </Dist>
     </Volume>
      <Capacity>
        <Dist type = "disjunctive">
         about  1.0e+07
       </Dist>
     </Capacity>
   </Tank_body>
   <Start_Date>28/03/00</Start_Date>
    <Responsibility>
      <Dist type = "conjunctive">
        <Val Poss =1.0>Tom</Val>
        <Val Poss =1.0>Mary</Val>
      </Dist>
    </Responsibility>
</Pressured air tank relation>
```

Fig. 2. Fuzzy XML document mapped from fuzzy nested relational databases in Table 1

V. CONCLUSION

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. XML is widely regarded as the next step in the evolution of the World Wide Web, and has been the de-facto standard. This creates a new set of data management requirements involving XML, such as the need to publish and query XML documents. On the other hand, fuzzy sets have been extensively applied to deal with information imprecision and uncertainty in the practical applications, and fuzzy database modeling is receiving increasing attention for intelligent data processing.

This paper focuses on fuzzy information modeling in the fuzzy nested relational database model and the fuzzy XML model. In order to represent fuzzy complex objects of the fuzzy nested relational databases with XML, we investigate the fuzzy DTD tree construction based on the hierarchical XML DTD and the fuzzy nested relational database (FNRDB) schema. Finally we develop the formal approach to mapping a fuzzy nested relational database (FNRDB) schema to a fuzzy DTD model.

REFERENCES

[1] Abiteboul, S., Segoufin, L. and Vianu, V., 2001, Representing and Querying XML with Incomplete Information, Proc. 12th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, 150-161.

[2] Bertino, E. and Catania, B., 2001, Integrating XML and Databases, IEEE Internet Computing, July-August, 84-88.

[3] Bordogna, G., Pasi, G. and Lucarella, D., 1999, A Fuzzy Object-Oriented Data Model for Managing Vague and Uncertain Information, International Journal of Intelligent Systems, 14: 623-651.

[4] Buckles, B. P. and Petry, F. E., 1982, A Fuzzy Representation of Data for Relational Database. Fuzzy Sets and Systems, 7 (3): 213-226.

[5] Carey, M. J., Kiernan, J., Shanugasundaram, J., Shekita, E. and Subramanian, S., 2000, XPERANTO: Middleware for publishing object-relational data as XML documents, Proceedings of the 26th International Conference on Very Large Databases, 646-648.

[6] Chamorro-Martínez, J., Medina, J. M., Barranco, C. D., Galán-Perales, E. and Soto-Hidalgo, J. M., 2007, Retrieving Images in Fuzzy Object-

Relational Databases Using Dominant Color Descriptors, Fuzzy Sets and Systems, 158 (3): 312-324.

[7] Chianese, A., Picariello, A., Sansone, L. and Sapino, M. L., 2004, Managing Uncertainties in Image Databases: A Fuzzy Approach, Multimedia Tools and Applications, 23: 237-252.

[8] Cuevasa, L., Marínb, N., Ponsb, O. and Vilab, M. A., 2008, pg4DB: A Fuzzy Object-Relational System, Fuzzy Sets and Systems, 159 (12): 1500-1514.

[9] Fernandez, M. F., Kadiyska, Y., Suciu, D., Morishima, A. and Tan, W. C., 2002, SilkRoute: A framework for publishing relational data in XML, ACM Transactions on Database Systems, 27 (4): 438-493.

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

[11] George, R., Srikanth, R., Petry, F. E. and Buckles, B. P., 1996, Uncertainty Management Issues in the Object-Oriented Data Model, IEEE Transactions on Fuzzy Systems, 4 (2): 179-192.

[12] Gyseghem, N. V. and Caluwe, R. D., 1998, Imprecision and Uncertainty in UFO Database Model, Journal of the American Society for Information Science, 49 (3): 236-252.

[13] Hung, E., Getoor, L. and Subrahmanian V. S., 2003, PXML: A Probabilistic Semistructured Data Model and Algebra. In: Proc. 19th the International Conference on Data Engineering, 467-478.

[14] Kappel, G., Kapsammer, E. and Retschitzegger, W., 2004, Integrating XML and relational database systems, WWW Journal, 7 (2): 343-384.

[15] Lo, A., Alhajj, R. and Barker, K., 2006, VIREX: Visual relational to XML conversion tool, Visual Languages and Computing, 17 (1): 25-45.

[16] Ma, Z. M., Liu, J. and Yan, L., 2010, Fuzzy Data Modeling and Algebraic Operations in XML, International Journal of Intelligent Systems, 25 (9): 925 - 947.

[17] Ma, Z. M. and Yan, L., 2007, Fuzzy XML Data Modeling with the UML and Relational Data Models, Data & Knowledge Engineering, 63: 972-996.

[18] Ma, Z. M. and Yan, L., 2008, A Literature Overview of Fuzzy Database Models, Journal of Information Science and Engineering, 24 (1): 189-202.

[19] Ma, Z. M. and Yan, L., 2010, A Literature Overview of Fuzzy Conceptual Data Modeling, Journal of Information Science and Engineering, 26 (2): 427-441.

[20] Ma, Z. M., Zhang, W. J. and Ma, W. Y., 2004, Extending Object-Oriented Databases for Fuzzy Information modeling, Information Systems, 29 (5): 421-435.

[21] Ma, Z. M., Zhang, W. J., Ma, W. Y. and Chen, G. Q., 2001, Conceptual Design of Fuzzy Object-Oriented Databases Using Extended Entity-Relationship Model, International Journal of Intelligent Systems, 16: 697-711.

[22] Naser, T., Alhajj, R. and Ridley, M. J., 2008, Flexible approach for representing object oriented databases in XML format, Proceedings of the 10th International Conference on Information Integration and Web-based Applications Services, 430-433.

[23] Naser, T., Kianmehr, K., Alhajj, R. and Ridley, M. J., 2007, Transforming object-oriented databases into XML, Proceedings of the 2007 IEEE International Conference on Information Reuse and Integration, 600-605.

[24] Nierrman, A. and Jagadish, H. V. 2002, ProTDB: Probabilistic Data in XML. Proceedings of the 28th International Conference on Very Large Data Bases. pp. 646-657.

[25] Oliboni, B. and Pozzani, G., 2008, Representing Fuzzy Information by Using XML Schema, Proceedings of the 2008 19th International Conference on Database and Expert Systems Application, 683-687.

[26] Peres, F. F. F. and Mello, R. dos S., 2009, A rule-based conversion of an object-oriented database schema to a schema in XML schema, Proceedings of the 4th IEEE International Conference on Digital Information Management, 198-204.

[27] Prade, H. and Testemale, C., 1984, Generalizing Database Relational Algebra for the Treatment of Incomplete or Uncertain Information and Vague Queries. Information Sciences, 34: 115-143.

[28] Raju, K. V. S. V. N. and Majumdar, K., 1988, Fuzzy Functional Dependencies and Lossless Join Decomposition of Fuzzy Relational Database Systems, ACM Transactions on Database Systems, 13 (2): 129-166.

[29] Senellart, P. and Abiteboul, S., 2007, On the Complexity of Managing Probabilistic XML Data, Proc. 26th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, 283-292.

[30] Umano, M. and Fukami, S., 1994, Fuzzy Relational Algebra for Possibility-Distribution-Fuzzy-Relational Model of Fuzzy Data, Journal of Intelligent Information Systems, 3: 7-27.

[31] Van Keulen, M., De Keijzer, A. and Alink, W., 2005, A Probabilistic XML Approach to Data Integration, Proceedings of the 2005 International Conference on Data Engineering, 459-470.

[32] Yan, L., Liu, J. and Ma, Z. M., 2010, Formal translation from fuzzy XML to fuzzy nested relational database schema, Soft Computing in XML Data Management (ISBN: 978-3-642-14010-5), Springer-Verlag, Studies in Fuzziness and Soft Computing, 255: 35-54.

[33] Yan, Li, Ma, Z. M. and Liu, Jian, 2009, Fuzzy Data Modeling Based on XML Schema, Proceedings of the 2009 ACM International Symposium on Applied Computing, March 8-12, 2009, Hawaii, USA, 1563-1567.

[34] Yazici, A., Buckles, B. P. and Petry, F. E., 1999, Handling Complex and Uncertain Information in the ExIFO and NF2 Data Models, IEEE Transactions on Fuzzy Systems, 7 (6): 659-676.

[35] Yazici, A., Soysal, A., Buckles, B. P. and Petry, F. E., 1999, Uncertainty in a Nested Relational Database Model, Data & Knowledge Engineering, 30 (3): 275-301.

[36] Zadeh, L. A., 1965, Fuzzy Sets, Information and Control, 8 (3): 338-353.

[37] Zadeh, L. A., 1978, Fuzzy Sets as a Basis for a Theory of Possibility, Fuzzy Sets and Systems, 1 (1): 3-28.