Fuzzy XML with Implementation

Goran Panić*
Đure Jakšića 87 23323 Idoš
Serbia
goran.pn@gmail.com

Miloš Racković
Faculty of Science
Trg Dositeja Obradovića 4
21000 Novi Sad
Serbia
rackovic@dmi.uns.ac.rs

Srdjan Škrbić
Faculty of Science
Trg Dositeja Obradovića 4
21000 Novi Sad
Serbia
srdjan.skrbic@dmi.uns.ac.rs

ABSTRACT

Standard XML format does not allow for imprecise or incomplete values. This is one of the requests imposed on this format by many real-word usages. Using fuzzy logic in order to introduce indefiniteness in XML has been researched in several different papers in the last decade. While these papers were mostly focused on setting up theories and the syntax, this paper has practical usage as its main goal. Application called 'Fuzzy XML editor' was created and described in this research. This editor is intended to work with fuzzy XML and to support XSD and DTD schemas.

Categories and Subject Descriptors

H.4.m [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*Complexity measures, Performance measures*

General Terms

Theory

Keywords

Applications, information sciences, fuzzy databases, information retrieval

1. INTRODUCTION

XML (Extensible Markup Language) has revolutionized fields of data storage and transfer in the last decade. There are many positive sides to XML which made it de facto a standard for many different uses. Values stored in XML elements and attributes are absolutely in accordance with their definition. Also, the whole structure of XML file is predefined and non-changeable. However, in reality, most information comes in the form of imprecise or incomplete values (e.g. fast car, tall building). Storing imprecise information is not one of the things standard XML supports. Fuzzy sets and fuzzy logic are theories used in many different fields where the use and processing of imprecise and incomplete information were needed. Adding fuzzy logic on top of standard XML would result in XML containing insufficiently defined and imprecise information.

Several papers dated to previous decade can be found, discussing the use of fuzzy logic in defining XML indefiniteness. However, the published papers focus mostly on setting up theories and defining syntax. There are only a small number of papers that deal with the implementation process and practical usage. Some simplistic implementations were presented, limited to a restricted set of instructions which often do not meet the needed functionality for any sort of practical use. Without practical usage, it is hard to get a clear view of actual possibilities of different theories. Also, the performance and complexity of implementing seemingly good syntax solutions do not always have to be satisfactory. In order to overcome these problems, this paper focuses on practical solutions in real-world environment. This is achieved by implementing to a level of usable applications. For numerical computation, MATLAB (Matrix laboratory) software was used (for the sake of simplicity and good performance). MSSQL (Microsoft SQL Server) is used as underlying database with XQuery queries. Fuzzy priorities and thresholds will be used and implemented in XQuery queries for the first time. However, prioritized fuzzy XQuery is out of scope and will be left for a subsequent paper.

Second chapter provides an overview of existing papers, in the field of fuzzy logic usage with XML. Chapter 3 explains XML indefiniteness, presenting extended XSD (XML Schema Definition) and DTD (Document Type Definition) schemas for indefinite XML. The features of application specifically developed for this paper are presented in details in Chapter 4.

2. RELATED WORK

Related research presented in this paper was conducted in the last five years at the University of Novi Sad. The main topic was implementation of system capable of using priority fuzzy logic with relational databases. In addition, complete fuzzy relational database application solution was designed and implemented [1, 2, 3].

The use of fuzzy logic in indefinite XML is an area where multiple papers have been presented in the last decade. Most influential research topics are: defining indefiniteness in XML values and defining indefiniteness in XML structure.

For defining indefiniteness in XML structure usually XPath and XQuery languages were extended by the authors. Presented solutions find elements in XML documents whose positions are not exactly stated. Examples of such solutions are given in: [4, 5, 6, 7, 8, 9].

*Corresponding author, tel. +381 64 289 8959

BCI'12, September 16–20, 2012, Novi Sad, Serbia.

Local Proceedings also appeared in ISBN 978-86-7031-200-5, Faculty of Sciences, University of Novi Sad.

Most significant form of XML indefiniteness is the indefiniteness in XML values. For structural definitions of fuzzy XML documents, XML schemas are usually used. However, in some works schema graphs were used for the same purpose. Such examples are presented in: [10, 11, 12, 13, 14, 15, 16].

XML documents are often stored into databases that support XML. Papers related to this subject discuss extending database models by introducing fuzzy types or extending the SQL and XQuery syntax by using fuzzy elements. See more on this topic in following papers: [17, 18, 19, 20, 21, 22, 23, 24, 25].

In this paper implementation of indefiniteness in XML values is researched. This assume definition of XML model that allows storing of fuzzy values. The main focus is on the implementation of practical fuzzy XML applications.

3. FUZZY XML SYNTAX

Best solutions presented in related works were combined and new model for XML fuzzy data storage was developed. For instance, [6] describes the most general solution to the problem of storage of fuzzy XML documents. To defined the syntax for indefinite XML, XSD and DTD schemas are used. The focus of this paper was on the allowing fuzzy constructs in XML element values. The indefiniteness in the XML structure was out of scope for this paper and is left for the future work.

Membership function is most important when it comes to fuzzy set usage. It is not an easy task to define membership function, while its storage and processing leverage the level of complexity of the implementation. There are two ways of building such solution. The first is that fuzzy set should include certain number of simple and well-known function types (e.g. increasing, decreasing, triangular, trapezoidal). In choosing groups of fuzzy sets to implement, one needs to pay special attention to the choice of sets. The purpose is to cover as many real systems as possible. Still, regardless of how many fuzzy sets are introduced, they are still restricted. So, cases whose functions are not described are still unreachable to such solutions.

The second option is to define syntax which can model the fuzzy set. In this paper, this option was selected due to its versatility. XML is metalanguage, intended to offer expandability, so there was no need to specifically change XML language syntax to support fuzzy logic. Still, in order to design XML structure flexible enough to enable defining wide variety of fuzzy sets, there are certain conditions. These are presented below.

For example, there is an XML element, <temperature>, which is undefined and is of a fuzzy type. The element can contain an arbitrary number of <fuzzy> elements which represent fuzzy sets. All fuzzy sets are uniquely named using the name attribute, such as <fuzzy name="cold">. Fuzzy sets are further defined by <function> - membership functions which can vary in number. Membership functions can be constrained by supplying minValue and maxValue attributes. As presented here: <function minValue="0" maxValue="10">; this cannot overlap within a single fuzzy set. The default values of these attributes are maxValue: plus infinity, minValue: minus infinity. The value of the <function> element defines the membership function for certain area defined by the minValue and maxValue attributes. In this value $\boldsymbol{x}$ is a variable that collects values from given intervals. The level of membership is taken from the interval [0,1]. Text used to define the value has a standardized format so that parser is able to transform it into the corresponding mathematical function. For example, functions are defined as this: $\boldsymbol{A}*\boldsymbol{x} + \boldsymbol{B}$. Membership will be set to zero if there are intervals with undefined membership function.

The easiest way to define the previously described fuzzy type is to use some of the existing XML schemas. XSD schema which defines the described fuzzy type is presented below. The element containing function describes part of the membership function limited by minValue and maxValue constraints. The functions element classifies function parts into a complex membership function. The fuzzy element is a fuzzy set which is defined by a set of membership functions. Fuzzy XSD schema is presented in listing 1

Listing 1: XSD schema for fuzzy element

```
<xs:schema id="Fuzzy" elementFormDefault="
    qualified" xmlns:xs="http://www.w3.org/2001/
    XMLSchema">
  <xs:complexType name ="fuzzy">
    <xs:sequence>
      <xs:element name ="fuzzy" minOccurs ="1"
          maxOccurs ="unbounded" type="functions"
          />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name ="functions">
    <xs:sequence>
      <xs:element ref="function" minOccurs ="1"
          maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string"/>
  </xs:complexType>

  <xs:element name="function">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="minValue" type="
              xs:string" />
          <xs:attribute name="maxValue" type="
              xs:string" />
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

So, by referencing Fuzzy.xsd schema in a XML document and using Fuzzy type for element type, indefinite fuzzy XML can be written. An example of this is given in listing 2.

Listing 2: Using Fuzzy XSD schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="Test" elementFormDefault="qualified
    " xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation ="Fuzzy.xsd"/>
  <xs:element name="measurement">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name" type="xs:string"/>
        <xs:element name="location" type="xs:int"/
            >
        <xs:element name="time" type="xs:dateTime"
            />
        <xs:element name="temperature" type="fuzzy
            "/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Same can be achieved by using other types of XML schema documents. Using DTD schema for the same task is presented in listing 3.

Listing 3: DTD schema for fuzzy element

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT measurement (temperature)>
<!ELEMENT temperature (fuzzy+)>
<!ELEMENT fuzzy (function+)>
<!ATTLIST fuzzy name CDATA #REQUIRED>
<!ELEMENT function (#PCDATA)>
<!ATTLIST function
  minValue CDATA #REQUIRED
  maxValue CDATA #REQUIRED>
```

Full example of fuzzy XML documents defined in accordance to presented syntax, storing air temperature measurements are given in listing 4.

Listing 4: Fuzzy XML example

```
<measurement>
  <name>New station</name>
  <location>41</location>
  <time>2012-02-03 10:30</time>
  <temperature>
    <fuzzy name="cold">
      <function maxValue="0">1</function>
      <function minValue="0" maxValue="10">-1/(Max
          -Min)*x + Max/(Max-Min)</function>
    </fuzzy>
    <fuzzy name="hot">
      <function minValue="10" maxValue="20">1/(Max
          -Min)*x - Min/(Max-Min)</function>
      <function minValue="20">1</function>
    </fuzzy>
  </temperature>
</measurement>
```

The value of fuzzy elements can include real text, the so called linguistic definitions. They start with a # sign. These variables are predefined fuzzy sets. Example of such definition can be found in listing 5.

Listing 5: Linguistic definition example

```
<fuzzy name="cold">#Maybe</fuzzy>
```

4. FUZZY XML EDITOR

Application 'Fuzzy XML editor' is implemented to demonstrate practical aspect of working with fuzzy XML syntax. This application enables creation and working with fuzzy XML, XSD and DTD documents, and has been developed specifically for the needs of this paper. Main application window is presented on figure 1.

.NET framework and WPF technology were used in implementation of this application. Fuzzy XML tool components and their mutual relations are presented in figure 2.

The application consists of several logical modules organized according to the MVVM (Model View View-Model) pattern and its principles.

The **Data** layer is the lowest level, dealing with the data storage. The storage options are: text files (file system component) and database (database component). Microsoft SQL server 2008 R2 support was implemented in order to work with the database. Its potential for working with XML is extended by fuzzy elements.

The **Model** layer contains several components, where each has a task to communicate with certain object types. The *XML model* supports working with fuzzy XML objects, while

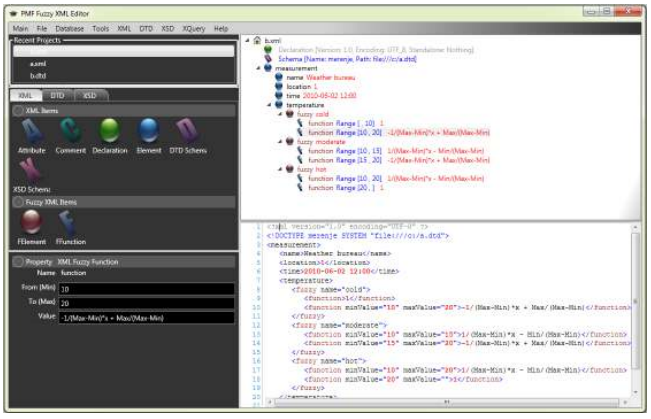

Figure 1: Fuzzy XML editor.

the DTD and XSD models are in charge of working with fuzzy XML schemas. The *XQuery model* implements working with fuzzy XQuery, and includes the support for priorities and thresholds. However, fuzzy XQuery is out of scope of this paper. Each of the components can implement the part for working with the fuzzy syntax apart from working with the standard language syntax.

The **View-Model** layer makes it possible to communicate with the layer below and consists of several modules. The *Command* group classifies modules for working with commands that implement interfaces for model approach. The *Syntax Validation* component enables validation of document syntax. The *Fuzzy* group enables working with membership functions and fuzzy linguistic variables. The *Fuzzy* group relies on the **Matlab** module which is an interface to external application - MATLAB. Calculations of the membership level are done using MATLAB libraries.

The **View** layer is the last layer. It implements the user interface, so that the user can perform functionalities defined in the *Command* group. The *Main Window* component is the main application window which includes the user controls from the *Controls* group and enables calling of the other independent dialogue windows from the *Dialogs* group.

Executable version and other data of the application can be found at `http://www.is.pmf.uns.ac.rs/fuzzydb`.

5. CONCLUSION

This paper successfully unites the process of defining syntax and implementing it. An application solution that enables usage of the fuzzy logic constructs with XML data has been provided. Users are given a possibility to define the arbitrary membership functions, and their computation is achieved in real time with the usage of the MATLAB software. The practical side of this implementation and its satisfactory performance has been demonstrated.

The presented application solution will be a basis for further development of XML indefiniteness. In future work, definition and implementation of the fuzzy XQuery interpreter that allows the usage of priorities is planned. Also, the emphasis of future work is on improving the implemented functionalities and the syntax, further improvement of the performance and finally, introduction of indefiniteness in the XML structure.

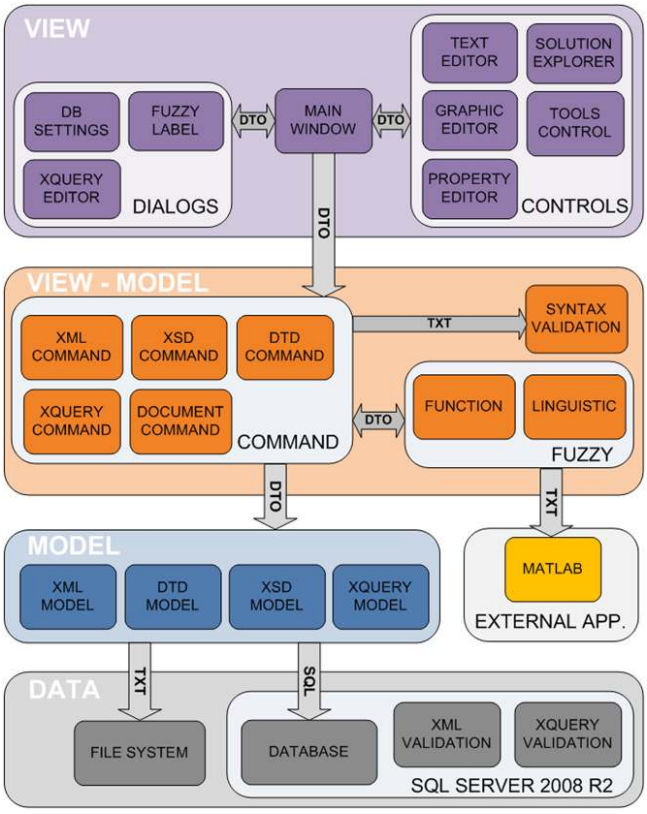

Figure 2: Application component block diagram.

6. ACKNOWLEDGMENT

Research was partially supported by the Ministry of Science and Technological Development of Republic of Serbia, through project no. OI-174023 'Intelligent techniques and their integration into wide-spectrum decision support'.

7. REFERENCES

[1] S. Škrbić, M. Racković, and A. Takači. The PFSQL query execution process. Novi Sad J. Math, 41(2):161–179, 2011.

[2] A. Takači, and S. Škrbić. Data Model of FRDB with different data types and PFSQL, in: J. Galindo (Ed.). Handbook of research on fuzzy information processing in databases, Information Science Reference, Hershey, PA, pages 407–434. 2008.

[3] S. Škrbić, M. Racković, and A. Takači. Towards the methodology for development of fuzzy relational database applications. Comput. Sci. Inf. Syst. 8(1):27–40, January 2011.

[4] A. Campi, E. Damiani, S. Guinea, S. Marrara, G. Pasi, and P. Spoletini. A fuzzy extension for the XPath query language, J. Intell. Inf. Syst. 33:285–305, December 2009.

[5] E. Damiani, S. Marrara, and G. Pasi. FuzzyXPath: Using fuzzy logic an IR features to approximately query XML documents, in: Proc. 12th Int. Fuzzy Syst. Assoc., Cancun, Mexico, pages 199–208. 2007.

[6] L. Yan, Z.M. Ma, and J. Liu. Fuzzy data modeling based on XML schema, in: Proc. ACM Symp. Appl. Comput. New York, USA, pages 1563–1567. 2009.

[7] Z.M. Ma, J. Liu, and L. Yan. Fuzzy data modeling and algebraic operations in XML. Int. J. Intell, Syst. 25(9):925–947, September 2010.

[8] E.J.F. Thomson, and G.Radhamani. Fuzzy logic based XQuery operations for native XML database systems. Int. J. Database Theor. App. 2(3):13–20, September 2009.

[9] A. Lo, K. Kianmehr, M. Kaya, and R. Alhajj. Wrapping VRXQuery with self-Adaptive fuzzy capabilities, in: Web Intelligence, IEEE/WIC/ACM Inter. Conf. Fremont, CA, pages 750–756. November 2007.

[10] B. Oliboni, and G. Pozzani. Representing Fuzzy information by using XML schema, in: Database and Expert Systems Application. DEXA Proc. 19th Inter. Conf. Database Expert Syst. App., Turin, Italy, pages 683–687. September 2008.

[11] W. Hailong, Z.M. Ma, Y. Li, and C. Jingwei. A unified formalism for fuzzy data types representation, in: FSKD '08. Fifth Inter. Conf., Shandong, China, pages 167–171. October 2008.

[12] H. Enrique, P. Eduardo, J.M. Jose, A. Sergio, and A. Karina. A fuzzy linguistic model to evaluate the quality of Web sites that store XML documents. Int. J. Approx. Reason. 46(1):226–253, January 2007.

[13] T. Chris, K. Wafa, and V. Toan. Universal fuzzy system representation with XML. Comp. Stand. Inter. 28(2):218–230, December 2005.

[14] C. Combi, B. Oliboni, and R. Rossato. Evaluating fuzzy association rules on XML documents, in: Computational Intelligence, Theory and Applications, Advances in Soft Computing 33, Springer, Heidelberg, pages 435–448. 2005.

[15] E. Damiani, B. Oliboni, and L. Tanca. Fuzzy techiques for XML data smushing, in: Proc. Inter. Conf. 7th Fuzzy Days Computational Intell., Theor. App., Dortmund, Germany, pages 637–652. 2001.

[16] E. Damiani, L. Tanca, and F.A. Fontana. Fuzzy XML queries via context-based choice of aggregations. Kybernetika. 36(6):635–655, 2000.

[17] J. Seto, S. Clement, D. Duong, K. Kianmehr, and R. Alhajj. Fuzzy query model for XML documents. Lect. Notes Comput. Sc. 5788, pages 333–340. 2009.

[18] J. Ying, and S. Shidlagatta. A framework of fuzzy triggers for XML database systems, in: Proc. Inform. Reuse Integrat. IEEE International Conference, Las Vegas, USA, pages 296–299. August 2009.

[19] A. Chan, N. Situ, K. Wong, K. Kianmehr, and R. Alhajj. Fuzzy querying of nested XML, in: Proc. Inform. Reuse Integrat. IEEE Inter. Conf. Las Vegas, USA, pages 238–243. July 2008.

[20] R.D. Rodrigues, A.J.O. Cruz, and R.T. Cavalcante. A proposal for a fuzzy database architecture incorporating XML. Fuzzy Set. Syst. 160(2):269–279, January 2009.

[21] G. Abhishek, and A. Reda. Incorporating fuzziness in XML and mapping fuzzy relational data into fuzzy XML, in: Proc. ACM Symp. Appl. Comput. New York, USA, pages 456–460. 2006.

[22] C.D. Barranco, J.R. Campaña, J.M. Medina, Towards a XML fuzzy structured query language, in: Proc. EUSFLAT Conf., Barcelona, Spain, pages 1188–1193. 2005.

[23] Z.M. Ma. Fuzzy database modeling with XML. Springer, New York, February 2005.

[24] Z.M. Ma, L. Yan, and F. Zhang. Modeling fuzzy information in UML class diagrams and object-oriented database models. Fuzzy Set. Syst. 186(1):26–46, January 2012.

[25] M. Koyuncu. Intelligent fuzzy queries for multimedia databases. Int. J. Intell. Syst. 26(10):930–951, Jul 2011.