Knowledge-Based Systems 38 (2013) 62–73

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Prioritized fuzzy logic based information processing in relational databases

Srdjan Škrbić, Miloš Racković, Aleksandar Takači *

University of Novi Sad, Faculty of Science, Department of Mathematics and Informatics, Trg Dositeja Obradovića 4, Novi Sad, Serbia

ARTICLE INFO

Article history:
Available online 8 February 2012

Keywords:
Fuzzy
Relational
Database
Application
Development

ABSTRACT

Many years of research related to fuzzy logic and fuzzy set theory extensions to relational databases have not lead to stable implementations, standardized languages or fuzzy relational database application development tools and methods. The main goal of this paper is the modelling and the implementation of a set of tools that allow usage of fuzzy logic enriched with priorities in relational database applications. In order to achieve that goal, at first, the relational data model is extended with the elements of fuzzy set theory. After that, a fuzzy extension of the SQL query language, called the PFSQL, is defined. An interpreter for that language is integrated inside an implementation of the fuzzy JDBC driver. An implementation of the CASE tool for modelling of fuzzy relational database schemas rounds up a set of tools for the implementation of Java fuzzy database applications. In this sense, this paper presents a step towards a methodology for the fuzzy relational database application development.

1. Introduction

Most of the real world information contains imprecise and incomplete values. The relational data model does not facilitate support for this kind of information. Attribute values are, by definition, absolute. Fuzzy sets and fuzzy logic have been successfully used in many similar applications, where management of imprecise and incomplete data was necessary. That is why the idea to incorporate fuzzy set theory and fuzzy logic to relational databases seams feasible and useful.

The main goal of the five-year long research conducted at the University of Novi Sad was the implementation of a system capable of using priority fuzzy logic with databases. Moreover, we have defined and implemented a complete fuzzy relational database application development solution. The system can be divided into four main parts:

- fuzzy query language PFSQL (Prioritized Fuzzy Structured Query Language),
- new fuzzy relational data model based on fuzzy extensions of the relational model,
- CASE (Computer Aided Software Engineering) tool for fuzzy relational database modelling and
- interpreter for the PFSQL query language weaved inside a fuzzy extension of the JDBC (Java DataBase Connectivity) driver that allows PFSQL querying from Java programs.

In an effort to create such a system, we encountered two main problems. The first one is a question of a method for storing fuzzy values using a relational database management system. The second, much more complex, is related to the design of the SQL language extensions that allow usage of prioritized fuzzy logic. Implementation of an appropriate interpreter that uses a fuzzy database is an integral part of the second problem. In this paper, we offer solutions to these problems and compare these solutions to the previous ones.

Giving solutions to these fundamental problems, we round them up by implementing the CASE tool that simplifies fuzzy database modelling and the fuzzy JDBC driver that allows easy querying. In this way, we obtain a system for the fuzzy relational database application development, to the best of our knowledge, the first of its kind.

The implementation of the priority fuzzy logic language interpreter is heavily dependent on the mathematical background. It is first necessary to formally explain how the actual calculations are done. The needed mathematical model is defined as a generalization of the known result – the Prioritized Fuzzy Constraint Satisfaction Problem. In this way, we obtain the concept of the Generalised Prioritized Fuzzy Constraint Satisfaction Problem (GPFCSP) that we describe in this paper as the needed mathematical background.

In the next section we give an overview of the existing results and solutions related to the use of fuzzy logic with databases. The third section contains the definition and other details related to the GPFCSP system. The next section gives the details related to the PFSQL language design, the fuzzy relational data model related to it and describes the main ideas in the PFSQL interpreter and fuzzy JDBC driver implementation. The fifth section describes how we applied the described system and gives some examples

* Corresponding author.
E-mail address: sasatakaci@gmail.com (A. Takači).

0950-7051/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2012.01.017

of its usage. At the end, we compare our approach to the most successful competitors – the FSQL (Fuzzy Structured Query Language) language and the FIRST-2 fuzzy relational data model, and draw some conclusions.

2. Related work

The first model that introduces similarity relations in the relational model is the Buckles–Petry model [1]. This paper gives a structure for representing inexact information in the form of a relational database. The structure differs from ordinary relational databases in two important aspects: components of tuples need not be single values and a similarity relation is required for each domain set of the database. Zvieli and Chen [2] offered a first approach to incorporate fuzzy logic in the ER (Entity–Relationship) model. Their model allows fuzzy attributes in entities and relationships.

Fuzzy functional dependencies and fuzzy normal forms, as well as algorithms for dependency preserving and lossless join decompositions of fuzzy relations in specific fuzzy extensions of the relational model are investigated in [3,4].

Umano and Fukami proposed the FREEDOM-O, a fuzzy database system which is an extension of the relational data model [5]. This system supports a fuzzy data model, and querying. It is the first implementation of a fuzzy database system. After that result, other researchers have proposed similar fuzzy extensions to the relational model such as in [6–9].

Another serious attempt to implement a fuzzy database system is given in [10,11]. The authors propose fuzzy extensions of the classical SQL and implement a system that allows using fuzzy conditions in the place of Boolean ones.

The GEFRED (Generalised Model of Fuzzy Relational Databases) model [12] is a probabilistic model that refers to generalised fuzzy domains and admits the possibility distribution in domains. This is a fuzzy relational database model that has the representational capabilities for a wide range of fuzzy information. In addition, it describes a flexible way to handle this information. The GEFRED model experienced subsequent expansions, such as [13–15].

Chen and Kerre [16] introduced the fuzzy extension of several major EER (Extended Entity–Relationship) concepts. Fuzzy logic was applied to some of the basic EER concepts connected to the notion of subclass and super class. Chaudhry, Moyne and Rundensteiner [17] proposed a method for designing fuzzy relational databases following the extension of the ER model of Zvieli and Chen. They also proposed a design methodology for the FRDBs (Fuzzy Relational Databases) that contains extensions for representing the imprecision of data in the ER model, and a set of steps for the derivation of a FRDB from this extended ER model.

Galindo, Urrutia and Piattini [18] describe a way to use the fuzzy EER model to model the database and represent the modeled fuzzy knowledge using a relational database in detail. This work gives insight into some new semantic aspects and extends the EER model with the fuzzy capabilities. The model is called the FuzzyEER model. Also, a way to translate the FuzzyEER model to the FIRST-2, a database schema that allows representation of fuzzy attributes in relational databases is given. The FIRST-2 schema introduces a concept of the Fuzzy Meta-knowledge Base (FMB). In addition, in this work, authors introduce and describe the specification and the implementation of the FSQL – an SQL language with fuzzy capabilities in great detail.

The current state of the art in this area includes mature fuzzy EER model extensions that describe a wide range of modelling concepts for full flavoured fuzzy database modelling. These conceptual models are supported by the robust models for the fuzzy data representation in relational databases, such as the FIRST-2. The possibilities to translate the conceptual models to the relational-based ones are also studied in detail. In addition, the FSQL is the first implementation of the fuzzy database query language that incorporates the majority of fuzzy logic concepts. Fuzzy logic has also been often applied to enrich various search and data mining algorithms [19,20].

In [21] the authors studied the possibilities to extend the relational model with the fuzzy logic capabilities. The subject was elaborated in [22,23], where a detailed model of fuzzy relational databases was given. One of the main features of the model is that it allows any fuzzy subset of the domain to be the attribute value which was not the case in previous FRDB models.

Moreover, using the concept of GPFCSP from [24,25], the authors found a way to introduce priority queries into fuzzy relational databases, which resulted in the PFSQL query language [26]. The PFSQL allows conditions in WHERE clause of a query to have a different priority i.e. importance degree. It is one of the first languages with such capabilities. The GPFCSP gives the theoretical background for the implementation of priority queries [26]. The authors have also succeeded in formalising the PFSQL queries by obtaining their interpretation in an existing fuzzy logic. They found that the $\text{Ł}\Pi\frac{1}{2}$ logic provides enough elements [27,28].

In difference to the earlier research attempts, the basic goal of our research is to develop an integrated set of tools that allows the fuzzy relational database application implementation. The main purpose of this paper is to round up all results obtained in this five-year long research and to describe the final solution as a whole.

3. Theoretical background

3.1. PFCSP

A prioritized fuzzy constraint satisfaction problem (PFCSP) is a type of the fuzzy constraint satisfaction problem (FCSP) in which the notion of priority is introduced. The PFCSPs are first introduced by Dubois et al. [29]. An axiomatic framework was given in [24] and applied in the agent-based automated negotiation [30]. The most important factors in that implementation are the Schur-concave t-norms. They are introduced in such a way that the value with the biggest priority has the largest impact on the result given by a Schur-concave t-norm. For more information on the Schur-concave t-norms and t-norms in general see [31].

Takači [32] gives an alternative axiomatic framework for the PFCSP, similar to the one given in [24], but with a stricter notion of priority. We recall that definition here.

Definition 3.1. When given (X, D, C^f, ρ), where

1. $X = \{x_i | i = 1,2,\ldots,n\}$ is a set of variables,
2. $D = \{d_i | i = 1,2,\ldots,n\}$ is a finite set of domains. Each domain d_i is a finite set containing the possible values for the corresponding variable x_i in X,
3. C^f is a set of fuzzy constraints, that is,
$$C^f = \{R_i^f | \mu_{R_i^f} : d_{i1} \times \cdots \times d_{ik_i} \to [0,1], \quad i = 1,\ldots,m, 1 \leqslant k_i \leqslant n\}, \tag{1}$$
4. $\rho: C^f \to [0,\infty)$

and a simultaneous valuation $v_X(x_1,\ldots,x_n)$, $x_i \in d_i$ of all variables in X, shortly denoted v_X and an operator $\oplus: [0,1]^n \to [0,1]$, $g: [0,\infty) \times [0,1] \to [0,1]$, constraints $R_i^f \in C^f$ for $i = 1,\ldots,n$ and a satisfaction degree $\alpha_\rho(v_X)$ which is calculated in the following way:

$$\alpha_\rho(v_X) = \oplus_{i=1}^n \left(g\left(\rho\left(R_i^f\right), \mu_{R_i^f}(v_X)\right)\right), \tag{2}$$

this system is a PFCSP if the following axioms are satisfied: