Advances in Fuzzy Temporal Relational Databases: A Review

NADEEM MAHMOOD[1] AND AQIL BURNEY[2]
Department of Computer Science
University of Karachi
University Road, Karach-75270, PAKISTAN
nmahmood@uok.edu.pk

Abstract: - This paper investigates the need for the development of fuzzy-temporal relational database model (FTRDM) and investigates some important and necessary issues and concepts for the development of such model. This paper highlights the importance of building temporal ontology for database application in the relational environment. The main emphasis is on the conceptual framework and paper presents important contributions made by the researchers in developing temporal relational models and temporal query languages. Most of the real world applications involve imprecise and uncertain data and fuzzy set theory has been extensively used by the researchers for the management of fuzzy (non-crisp) data in relational database environment. We propose the guideline for the representing imprecise and vague information concepts in a conceptual data model. The significant contributions in this area are also explored. Fuzzy data has its own dynamics and time has its own complexity, which leads to fuzzy-temporal or temporal-fuzzy models. This paper presents the methodology and concepts pertaining to the development of fuzzy and fuzzy temporal query language and surveyed the important contributions in this direction.

Key-Words: - Temporal ontology, temporal relational model, imprecise data, fuzzy set theory, fuzzy-temporal model, fuzzy-temporal query language

1 Introduction

Time is the one of the most difficult aspect to handle in real world applications such as database systems. Relational database model proposed by [1] offer very little built-in query language support for temporal data management [2]. The relational model itself incorporates neither the concept of time nor any theory of temporal semantics. The relational data model only support functionality to access a single state (most recent one) of the real world, called as snapshot, and to transition from one database state to another (updates) thereby giving up the old state. There exist, however, many application domains which need to have access not only to the most recent state, but also to past and even future states, and the notion of data consistency must be extended to cover all of these database states. The management of temporal data in the relational database environment was usually handled with the help of ad-hoc methods or through application programs.

Temporal databases (TDB), provides a common framework for all applications that requires some temporal aspects [3]. Temporal databases formally allow the definition of time oriented data in a unified manner, such as definition of temporal data semantics with query support. Many applications in the real world requires management of time varying data such as financial applications, insurance applications, reservation systems, medical information management systems, decision support systems, CRM applications, HR applications.

Efforts to incorporate the temporal domain into database management system have been ongoing for more than two decades and extensive research has been conducted and dozens of temporal models have been proposed [4][5][6][7][8][9][10][11][12]. An important survey of temporal database models was conducted by [2].

There are many interesting proposals regarding the development of temporal query language and some of them are also implemented [6][11][13]. Most of the proposed temporal query languages are the extension of SQL. However there are query languages developed for object oriented database. An important survey of query languages is conducted by [14][15].

There are many research issues still to be answered regarding the management of imprecise data in a temporal database environment [16]. Sometimes we are very much certain about the data and on the other hand there are events where we don't know the exact information about data. The data has both crisp and (fuzzy) no crisp nature and to manage fuzzy data in the temporal relational environment is a difficult task. For instance let's assume the patient database where we want to store the blood pressure and we define the fuzzy sets for that purpose whether it is low, very low, high normal or very high. There are some important research publications regarding fuzzy temporal database [16][17][18][19].

The research in the area of fuzzy and temporal databases is ongoing for atleast two decades and many proposals are published [20]. The research is divided in three categories, extension of the relational model, object oriented model and object relational model. Most of the proposed models are the extensions of the conventional relational model [1], because it is the most widely used model and well implemented.

Another important research issue is the development of conceptual database model involving fuzzy and temporal data. Most of the proposed conceptual models are the extensions of the classical entity relationship model [21]. These extensions are mainly divided into two categories temporal extensions and fuzzy extensions. The important contribution for temporal extension to the ER model is presented by [22]. A solid review of fuzzy extensions of ER model is presented in [23]. There are very few extensions which deal with both the fuzzy and temporal data together. This paper provides a brief survey of some of the important extensions of ER model.

An important consideration in a fuzzy temporal database is the concept of time granularity [24]. Different facts have different time associations, and the unit of time varies from facts to facts. The conversion of granularities from one another is a complex task and must be addressed in temporal database design. [25] presents a formal representation of fuzzy-temporal data and proposed fuzzy temporal operators for query language. Nevertheless, crucial operations such as insert, update and delete are not addressed.

Extensive research has been conducted on multiple aspects of fuzzy temporal database management, ranging from issues related to fuzzy conceptual temporal data modeling to physical fuzzy temporal data model with special focus on fuzzy temporal query language development is underway.

The paper is organized as follows. The second section investigates the temporal ontology related to temporal database development. Section 3, 4 & 5 introduces the temporal logic and its application in the design of temporal database models and gives a brief summary of temporal query languages. Section 6 & 7 discusses the fuzzy ontology and fuzzy logic in the context of fuzzy database and fuzzy temporal databases. This section includes the important discussion related to fuzzy time and data representation and the concepts related to building fuzzy temporal database system. Section 8, 9 & 10 discuss the need of the fuzzy conceptual data model, a brief summary of fuzzy temporal conceptual models and summarizes the significant fuzzy query languages developed with salient features. Finally we concluded and discuss issues for future research.

2 Time Ontology

2.1 Time Domain

A time domain is a ordered pair $(T ; \leq)$ where T is a non-empty set of time instants and "$\leq$" is total order on T. A time domain is dense [26] if it is an infinite set and for all $t, t' \in T$ with $t < t'$, there exists $t'' \in T$ such that $t < t'' < t'$. A time domain is discrete if every element except the last element (if any) has an immediate successor, and every element except the first (if any) has an immediate predecessor. Dense time is isomorphic to rational or real numbers whereas continuous time is isomorphic to the real numbers i.e. no gaps between consecutive instants and each number correspond to a time point [26].

2.2 Time Granularity

Partitioning of the time-line into a finite set of smaller segments called granules. Each non-empty subset G(i) is called a granule of the granularity [24]. For e.g. birthdates are typically measured at granularity of days, business appointments to granularity of hours and train schedules to granularity of minutes. Mixing of granularities create problems in handling temporal data. e.g., birthdates are typically measured at granularity of days business appointments to granularity of hours train schedules to granularity of minutes.

A granularity [24] is a mapping G from the integers (the index set) to subsets of the time domain such that:

(1) If $i < j$ and G(i) and G(j) are non-empty, then each element of G(i) is less than all elements of G(j).

(2) If $i < k < j$ and G(i) and G(j) are non-empty, then G(k) is non-empty.

Mixed granularities are of basic importance to modeling real-world temporal data. Mixing granularities create problems. What are the semantics of operations with operands at differing granularities? Can times be converted from one

Comparison Predicate	Equivalent Predicates on Endpoints
I_1 before I_2	$end(I_1) < begin(I_2)$
I_1 after I_2	$end(I_2) < begin(I_1)$
I_1 during I_2	$(begin(I_1) > begin(I_2) \wedge end(I_1) \leq end(I_2)) \vee$ $(begin(I_1) \geq begin(I_2) \wedge end(I_1) < end(I_2))$
I_1 contains I_2	$(begin(I_2) > begin(I_1) \wedge end(I_2) \leq end(I_1)) \vee$ $(begin(I_2) \geq begin(I_1) \wedge end(I_2) < end(I_1))$
I_1 overlaps I_2	$begin(I_1) < begin(I_2) \wedge end(I_1) > begin(I_2) \wedge end(I_1) < end(I_2)$
I_1 overlapped_by I_2	$begin(I_2) < begin(I_1) \wedge end(I_2) > begin(I_1) \wedge end(I_2) < end(I_1)$
I_1 meets I_2	$end(I_1) = begin(I_2)$
I_1 met_by I_2	$end(I_2) = begin(I_1)$
I_1 starts I_2	$begin(I_1) = begin(I_2) \wedge end(I_1) < end(I_2)$
I_1 started_by I_2	$begin(I_1) = begin(I_2) \wedge end(I_2) < end(I_1)$
I_1 finishes I_2	$begin(I_1) > begin(I_2) \wedge end(I_1) = end(I_2)$
I_1 finished_by I_2	$begin(I_2) > begin(I_1) \wedge end(I_1) = end(I_2)$
I_1 equals I_2	$begin(I_1) = begin(I_2) \wedge end(I_1) = end(I_2)$

Figure 1: Allen's temporal comparisons predicates for time intervals

granularity to another? How expensive is maintaining and querying times at different granularities? Mappings between different granularities in a lattice have to be provided plus an anchor point.

2.3 Time points and time intervals

Three common approaches of time representation are a single time point, an interval and a set of time intervals. In few models time is expressed using single time points called as events [4][5]. Most of the temporal models use time intervals to represent time. A detailed study of point and interval based temporal database models was published by Bohlen [27]. Allen [28] proposed a set of thirteen temporal comparisons predicates for time intervals (fig. 1) which are considered to be the benchmark.

3 Temporal Logic

The term temporal logic has been broadly used to cover all approaches to the representation of temporal information within a logical framework, and also more narrowly to refer specifically to the modal-logic type of approach introduced around 1960 by Arthur Prior under the name of tense logic [29] and subsequently developed further by logicians and computer scientists.

Temporal logic is an extension of conventional (propositional) logic, which incorporates special operators that cater for time [30]. With temporal logic one can specify and verify how components, protocols, objects, modules, procedures and functions behave as time progresses. The specification is done with temporal logic statements that make assertions about properties and relationships in the past, present, and the future.

There are two major types of temporal logics, linear time and branching time. Linear temporal logic (LTL) is a language of assertions about computations. Branching time logic is also called as computation tree logic (CTL) [30]. Temporal logic is a form of modal logic where the modal operators deal with the concept of time. Common temporal operators are as follows:

◊ is eventually
□ is always
O is next time
U is until

3.1 Temporal Specifications

Lot of research published in the area of temporal specification for temporal reasoning systems [31]. Temporal logic provides an excellent mechanism for the development of temporal specifications for temporal reasoning process. Let's assume a patient diagnostic process, where temporal logic can be used to axiomatize the following flow of event.

A: rise in the BP level, B: stroke, C: fits

A occurred before B, and C occurred after event B, implies that A occurred before C, without knowing the time association of those events Our goal is to provide a generic temporal specification for the temporal database system which can be used for reasoning w.r.t. time.

It is observed that in most of the applications domains the total order of time has no real significance and it deals with finite existing temporal events or facts related to the respective domains. For example in the patient database application the primary focus is to built temporal specifications where time is associated with facts and the time granularity defines the exact nature of time associated with the facts.

3.2 Temporal Logic for Databases

Useful application of temporal logic has found in the area of database management systems and the related disciplines. Temporal logic, by virtue of its internal semantics provides the bases for developing temporal query language. Temporal query language has its own semantics with respect to abstract temporal databases and time-point-indexed sequences of database states [14]. First order temporal logic (FOTL) provides an effective mechanism for the development of query language for temporal database. An important contribution is the development of temporal language called "Templog" [32], which is an extension of Datalog and provides the expressiveness and completeness for a temporal query language.

It is merely impossible to manage (store) the complete states of a temporal relation explicitly using temporal logic. Realistically speaking, each temporal model has its own semantics; it varies from application to application. The validity or transaction period of a fact can only be determined by a time encoding strategy. In most of the temporal models, time encoding strategy is based on intervals or periods, or with the help of temporal elements (may be infinite set of time points). These ranges or intervals can be represented as explicit sets. Temporal logic, due to its well-structured foundations, provides a high level of abstraction for querying temporal databases [14].

3.3 Semantics of FOTL

"Let DB be a point-stamped temporal database with a data domain D, a point-based time domain T_p, and a (snapshot) schema ρ. The satisfaction relation DB, θ, t $\models$ Q, where Q is an FOTL formula, θ a valuation, and t $\in$ T_p, is defined inductively with respect to the structure of the formula Q:" [33].

$$DB, \theta, t \models r_j(x_{i1}, \ldots, x_{ik})$$
$$\text{if } (\theta(x_{i1}), \ldots, \theta(x_{ik})) \in r_j^{DB(t)} \qquad (1)$$

$$DB, \theta, t \models x_i = x_j \qquad \text{if } \theta(x_i) = \theta(x_j) \qquad (2)$$

$$DB, \theta, t \models Q_1 \wedge Q_2 \qquad \text{if } DB, \theta, t \models Q_1$$
$$\text{and } DB, \theta, t \models Q_2 \qquad (3)$$

$$DB, \theta, t \models \neg Q_1 \qquad \text{if not } DB, \theta, t \models Q_1 \qquad (4)$$

$$DB, \theta, t \models \exists x_i . Q_1$$
$$\text{if there is a} \in D \text{ such that } DB,$$
$$\theta[x_i \rightarrow a], t \models Q_1 \qquad (5)$$

$$DB, \theta, t \models Q_1 \text{ since } Q_2 \text{ if } \exists t_2 . t_2 < t$$
$$\text{and } DB, \theta, t_2 \models Q_2$$
$$\text{and } (\forall t_1 . t_2 < t_1 < t \textit{ implies } DB, \theta, t_1 \models Q_1) \qquad (6)$$

$$DB, \theta, t \models Q_1 \text{ until } Q_2 \text{ if } \exists t_2 . t_2 > t$$
$$\textit{and } DB, \theta, t_2 \models Q_2$$
$$\text{and } (\forall t_1 . t_2 > t_1 > t \textit{ implies } DB, \theta, t_1 \models Q_1) \qquad (7)$$

Where $r_j^{DB(t)}$ is the instance of the relation r_j in the snapshot DB(t) of the database DB at the instance t. The answer to an FOTL query Q over DB is the set of tuples:

$$Q(DB) := \{(t, \theta(x_1), \ldots, \theta(x_k)) : DB, \theta, t \models Q\} \quad ,$$

where $x_1, \ldots, x_k$ are the free variables of Q.

4 Temporal Relational Algebra

There have been extensions proposed to the relational algebra to incorporate time dimension for the last three decades but only few extensions are well accepted by the database community, such as [34][35][36]. An excellent evaluation criterion for the temporal relational algebra (TRA) was proposed by [37] is presented below in table 1.

The temporal relational operations are to be defined in contrast with the relational operations such as, T-UNION, T-DIFFERENCE, T-JOIN, T-PROJECTION, and T-SELECTION. Other temporal relational operations such as, T-INTERSECTION, T-CARTESIANPRODUCT, T-DIVISION etc. can also be defined in terms of the basic operations. The definitions of new temporal operations are to be consistent with the definitions of the relational algebra.

	Conflicting Criteria
1	All attrIbutes in a tuple are defined for same interval(s)
5	Each set of legal tuples is a legal relation
15	Restricts relations to first normal form
16	Supports a 3-D view of historical state and operations
17	Supports basic algebraic equivalences
23	Tuples are time-stamped
24	Unique representation for each temporal relation
	Compatible Criteria
2	Consistent extension of the snapshot algebra
3	Data periodicity is supported
4	Each collection of legal attribute values is a legal tuple
6	Formal semantics are well defined
7	Has the expressive power of a temporal calculus
8	Includes aggregates
9	Incremental semantics defined
10	Intersection, theta JOIN, natural JOIN, and quotient are defined
11	Is, in fact, an algebra
12	Model doesn't require null attribute values
13	Multidimensional time-stamps are supported
14	Reduces to the snapshot algebra
18	Supports relations of all four classes
19	Supports rollback operations
20	Supports multiple stored schemas
21	Supports static attributes
22	Treats valid time and transaction time orthogonally
25	Unisorted (not multisorted)
26	Update semantics are specified

Table 1: Evaluation Criterion for the TRA

5 Temporal Query Languages

Efforts have been made to develop a complete temporal query language which can work with the relational database environment like SQL. There are significant developments in the development of temporal query language which provides full temporal support i.e. temporal interface, temporal operators and query optimization techniques. Table 2 summarizes some of the most important query languages developed in the past with their strengths and limitations.

Most of the commercial database management systems provide special data types for capturing date, time and timestamps. Newer versions of structured query language provide a limited support for querying and managing temporal data. None of the products listed above is said to be a complete temporal database.

6 Fuzzy Logic:

Fuzzy logic in contrast with the binary logic provides the possibility of intermediate values instead of rigid and crisp boundaries. For instance the patient condition can be normal, stable, critical and severe and it also gives the degree of flexibility that how close to these values. In binary logic there are no middle values either the patient is stable or unstable. Fuzzy logic deals with uncertain and imprecise information in a much flexible and organized manner [42][43].

Following definitions of fuzzy set and the operations performed on fuzzy sets are stated by [44].

6.1 : Let X be some set of objects, with elements noted as x. Thus, X = {x}.

6.2 : A fuzzy set A in X is characterized by a membership function
mA(x) which maps each point in X onto the real interval [0.0, 1.0]. As
mA(x) approaches 1.0, the "grade of membership" of x in A increases.

6.3 : A is EMPTY iff for all x, mA(x) = 0.0.

6.4 : A = B iff for all x: mA(x) = mB(x) [or, mA = mB].

6.5 : mA' = 1 - mA.

6.6 : A is CONTAINED in B iff mA <= mB.

6.7 : C = A UNION B, where: mC(x) = MAX(mA(x), mB(x)).

6.8 : C = A INTERSECTION B where: mC(x) = MIN(mA(x), mB(x)).

Query Language / Citation	Important Features	Limitations
IXSQL [38]	IXSQL is an extension of SQL2 for (time) interval data. Normalization of timestamps so that they are aligned (identical or disjoint) Introduced two important functions UNFOLD: decompose a interval-timestamped tuple into a set of point-timestamped tuples. FOLD: collapse a set of point timestamped tuples into value-equivalent tuples.	In efficient evaluation of queries due to the unfold operation. Neither purely point-based nor interval-based view Fold/unfold only preserve information of a point-based view Normalization step using unfold/fold looses interval information
ATSQL [39]	ATSQL provides temporal extension to SQL by introducing temporal statement modifiers. Language mechanism is independent of the syntactic complexity of the queries. ATSQL assumes interval-timestamped tuples with strong interval semantics. Value-equivalent tuples with adjacent or overlapping intervals are permitted.	Represents a much more fundamental change to the language than other approaches, e.g., abstract data types. Implementation of modifiers has still to be solved TSDB’
TSQL2 [11]	TSQL2 is an extension of SQL2 to support both time points and time intervals. TSQL 2 uses syntactic defaults that make the formulation of common temporal queries more convenient. A default valid clause is implicitly placed after the select clause:	Informally defined semantics Syntactic defaults are not scalable The language is complex
HTQUEL [7]	HTQUEL is based on the same query language as TQuel. Time is considered as discrete equidistant time intervals. A relational algebra is defined and completeness is proved.	Aggregate operations are not defined for HTQUEL yet.
SQL/TP [40]	A point-based temporal extension of SQL Defined on temporal relations that use point timestamps Simple, unambiguous, and well-defined semantics	Additional attribute needed to express several queries and preserve information,
TOSQL [41]	TOSQL is an extension of SQL. It allows access to both current data and their previous versions, but does not include update semantics	
TQuel [6]	TQuel is an extension of Quel with language constructs to retrieve facts that have been time stamped with a validity interval.	No algebraic language. Due to tuple substitution TQuel lacks in query optimization.
ATSQL2 [13]	ATSQL2 is a temporally complete language. It is based on extending relational schema to store time-varying data. It can be integrated with the relational database environment by providing temporal data definition and modification language with temporal constraint specification.	

Table 2: Brief Survey of Temporal Query Languages

7 Fuzzy Ontology

7.1 Fuzzy Number [45]

The fuzzy quantity A with membership function $\mu_A(x)$ is a fuzzy number [46] if:

1.
$\forall \alpha \in [0,1], A_\alpha = \{x \in R \mid \mu_A(x) \geq \alpha\}$ $(\alpha - cuts\ of\ A)$ $is\ a\ covex\ set.$

2. $\mu A(x)$ is an upper-semi-continuous function.

3. The support set of A, defined as Supp(A) = $\{x \in R \mid \mu_A(x) > 0\}$, is a bounded set of R, where R is the set of real numbers.

7.2 Fuzzy Time Representation

Most of the research published regarding temporal database model focused on the definition of temporal attributes such as valid time(VT) and transaction time (TT). Both VT and TT has start and end values represented by VST, VET and TST, TET. Although there are few proposals where other time attributes are also used [2]. Applying the same concept of validity to fuzzy database may change the definition of valid time. Let's suppose the valid time of a patient recovery process after a certain treatment, where we are not exactly sure that when it will be valid.

There are different proposals for handling fuzzy valid time (FVT) in fuzzy databases. One solution to the problem is to assign fuzzy number to each date value recorded in FVT attribute [45]. The interpretation of the fuzzy value in the attributes can be determined with the help of fuzzy probability distribution function. Table 3 represents the FVT values in a fuzzy relation.

Name	Pat. ID	Age	Cond.	FVST	FVET
Aamir	1101	35	stable	~11-07-2009	~13-07-2009
Aamir	1101	35	normal	~14-07-2009	~undefined
…	…	…	…	…	…

Table 3: Sample Relation with Fuzzy VST and VET

7.3 Fuzzy Data Representation

Fuzzy data has its own dynamics and there are many approaches for the representation of fuzzy data [47]. The traditional relational data model cannot represent fuzzy data in the form relations due to 1NF assumptions and fuzzy data is many valued. There is no concept of fuzzy data types in the relational model and as well the fuzzy relational operators. However there are proposed extensions of the relational model to incorporate fuzzy data [23][47]. Patient database requires both temporal and as well as fuzzy concepts. In this section we present the representation criteria for our proposed system from conceptual schema to implementation. The fuzzy data definition may have different representation based on the fuzzy membership function and the criteria for its implementation. Table 4 represents an example fuzzy relation.

P. No.	Activation _start	Activation _end	Update _time	Patient BP	µ
1002	01-12-2009 12:30 pm	03-12-2009 1:00 pm	01-12-2009 1:00 pm	high	0.1
1004	31-11-2009 10:00 am	03-12-2009 2:00 pm	31-11-2009 10:00 am	normal	0.5
1201	12-12-2009 09:00 am	Current Time	12-12-2009 08:50 am	low	0.4
1109	12-12-2009 10:50 am	Current Time	12-12-2009 09:50 am	very high	0.8
1201	---			Very low	0.2
---	---	---	---	---	

Table: 4 Fuzzy Data Regarding Patient Blood Pressure(BP)

8 Fuzzy Conceptual Database Models

Traditionally the conceptual data model is used for the development of database by mapping the logical schema into physical schema. The entity relationship (ER) model is used in relational database environment for this purpose. There are many extensions to the classic entity relationship model incorporating time dimension found in the literature [22]. However there are extensions of ER model for handling imprecise data and are presented in [23].

The relational model has no inbuilt mechanism for managing imprecise nature of the data and the basic definition of a relation changes when we allow many values instead of atomic value. It is observed that in most of the application domains the fuzzy dimension is also time oriented [16][17][19]. Figure 2 represents a fuzzy temporal conceptual model proposed by [19]. Very few representations of conceptual modeling fuzzy temporal information are present in the literature. Most of the fuzzy extensions are dependent on the type of fuzzy database and there is no common or generic framework exists for this purpose. Object oriented models such as UML and IFO are also used for the semantic relationships exist between the objects and later mapped into a object oriented database [48]. Table 5 summarizes the important conceptual models proposed with some important properties.

9 Fuzzy Relational Database Model

The fuzzy temporal relational database model [20] is the extension of the relational model, managing and representing imprecision and vagueness of data in the form of a relation. Primarily there are three ways for incorporating imprecision in a fuzzy relational model. First the imprecision to be the member of entity set, secondly the degree of relationship membership, and lastly the imprecision in a data value.

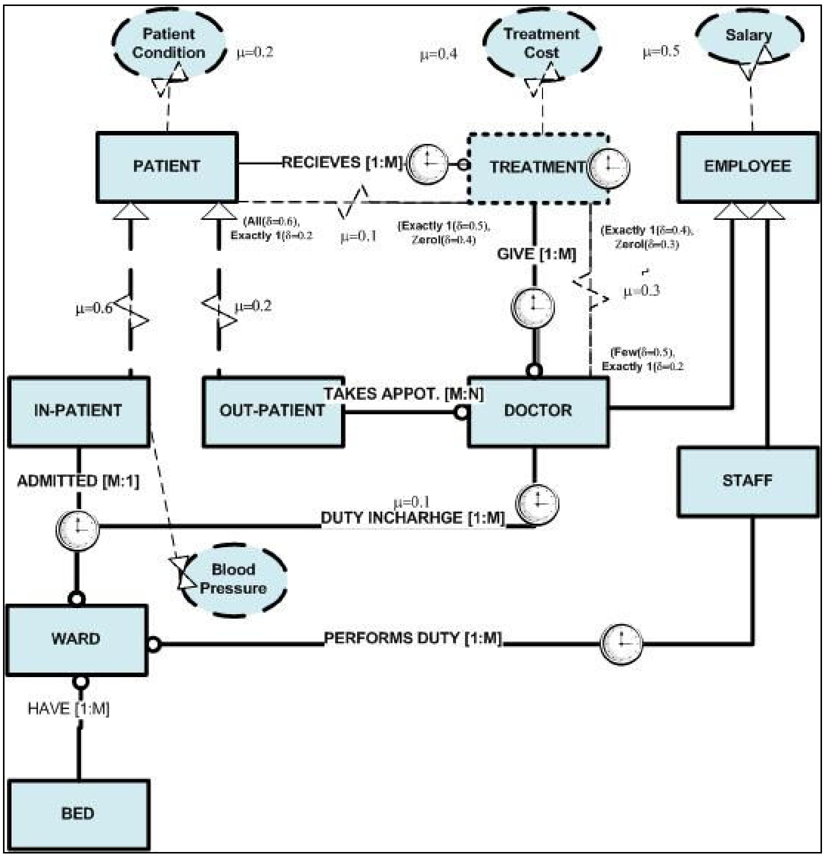

Figure 2: FTRM for Patient Database

Model Citation	Fuzziness levels	Fuzzy Constraints	Rep. of Fuzziness	Graphical Rep.
OO modeling using EERM [49]	Attribute, entity, and relationship Generalization-specialization and aggregation. formal approach to mapping a fuzzy extended entity-relationship model to a fuzzy object-oriented database schema	NO	Possibility theory	YES
Fuzzy IFO Model [48]	Attributes, objects and classes	NO	Similarity relations; Possibility theory	YES
Fuzzy UML Data	Attributes, classes and relationship	NO	Possibility theory	YES
Fuzzy temporal ER/FTER[17]	Attribute, entity, and relationship Transaction time and valid time	NO	Possibility theory	YES
FTRM [19]	Attribute, entity and relationship	NO	Possibility theory	YES
Fuzzy ER/EER[50]	Fuzzy extensions to basic ER/EER concepts such as super-class/sub-class, gen-spec, and shared subclass category are discussed. The fuzzy attribute inheritance, multiple inheritance and selective inheritance.	YES Inheritance, participation & cardinality constraint.	Possibility theory	NO
Zvieli & Chen Model [51]	classes, relationships and attributes domains may be fuzzy and may have degrees of membership. Related with the fuzzy occurrences of objects/entities and relationships. attributes are authorized to take imprecise, & vague values.			

Table 5: Fuzzy Conceptual Database Models

9.1 Fuzzy Relation:[20]

Let U be the Cartesian product of n universes of discourse U_1, U_2, U_3,….U_n. Then and n-ary fuzzy relation r in U is a relation which is characterized by a n-variate membership function ranging over U, i.e. $\mu_r : U \rightarrow [0,1]$. A tuple of the fuzzy relation r can be expressed as $t_j = (u_{j1}, u_{j2}, \ldots, u_{jn}, \mu_r(u_{j1}, u_{j2}, \ldots, u_{jn}))$ with $u_{j1} \in U_1, \ldots, u_{jn} \in U_n$. Table 6 represents a fuzzy relation based on the above definition.

Patient No.	Condition	μ
1101	Stable	1.0
2111	Stable	0.5
1123	Normal	0.4

Table 6 Fuzzy Relation

9.2 Fuzzy Data Types

There are different representations as far as fuzzy data types are concerned and there are many implementations to these approaches [47] proposed different fuzzy data types such as:

A single scalar (Behavior = good, represented by the possibility distribution, 1/good)

A set of possible scalar assignments (Behavior = {good, bad}, represented by {1/good, 1/bad})

A set of possible numeric assignments (Age = {20, 21}, represented by { 1/20, 1/21})

A possibility distribution in a scalar domain (Behavior = {0.6/bad, 0.7/normal})

A real number belonging to [0, 1], referring to degree of matching (Quality = 0.9)

An Undefined value with possibility distribution, Undefined = {0/u: u e U}

A NULL value given by NULL = {1/Unknown, l/Undefined}

9.3 Fuzzy Functional Dependencies

Functional dependencies (FDs), plays a vital role in understanding the relationships exist between the attributes in a relation. With the help of FDs one can determine the normal form of a relation and also the definitions of key constraints [52]. The classic relational model did not give the concept of handling of fuzzy data in a relation because by doing so the basic structure of the relation may change [53]. According to the classical definition of FD [54], for a classical relation r(R), in which R denotes the set of attributes, we say r satisfies the functional dependency FD: X→Y where XY⊆R if

$$(\forall t \in r)(\forall s \in r)(t[X] = s[X] \Rightarrow t[Y] = s[Y])$$

For a relation instance r(R), where R denotes the schema, its attribute set is denoted by U, and X, Y ⊆ U, we say r satisfies the fuzzy functional dependency FFD: $X \rightarrow^F Y$, if

$$(\forall t \in r)(\forall s \in r)(SED_\alpha(t[X], s[X]) \leq SED_\alpha(t[Y], s[Y])) \quad (7)$$

9.4 Fuzzy Similarity and Proximity Relation [19]

Assume A and B are two fuzzy sets, the similarity of fuzzy sets A and B being developed as

$$S_{AB} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (8)$$

where $\cap$ and $\cup$ denote fuzzy intersection and union of fuzzy sets A and B, respectively. $|\cdot|$ denotes the size of a fuzzy set. The computation of fuzzy similarity is easy in case of triangular and trapezoidal functions. For Gaussian membership function, trapezoidal function is used to approximate the long tails of bell shaped curve (see fig. 3). A detailed study of similarity functions for data driven fuzzy models is [55].

Fuzzy models are complex in nature and similarity measures are used to simplify fuzzy models. Similarity measure determines the degree to which the different fuzzy sets are equal. There are two broad categories of similarity measures, geometric and set-theoretic. The geometric similarity measure is based on graph and the set-theoretic based on the size of the fuzzy sets. Consistency-index is the maximum membership degree of the intersection of two fuzzy sets [56].

$$S_C(A,B) = \sup_{x \in X} \mu_{A \cap B} = \max_{x \in X} [\mu_A(x) \wedge \mu_B(x)] \quad (9)$$

Where $\wedge$ is the maximum operator

The theoretical analysis of similarity has been dominated by geometric models. These models represent fuzzy sets as points in a metric space and the similarity between the sets is regarded as an inverse of their distance in the metric space. Denoting the distance between A and B as D(A,B), the similarity of A and B can be written as:

$$s = S(A,B) = \frac{1}{1 + D(A,B)}, \quad s \in (0,1]. \quad (10)$$

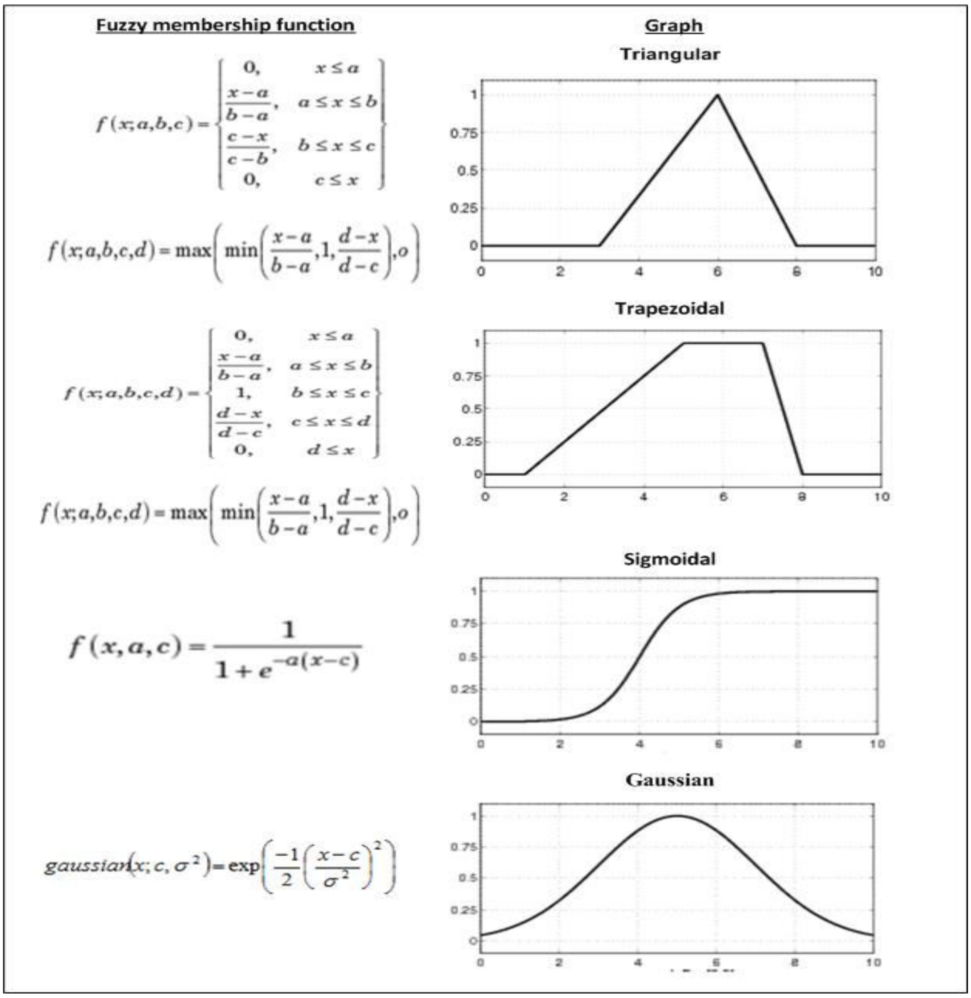

Figure 3: Fuzzy Membership Functions with Respective Graphs

A similarity relation [19][42] is a mapping, $S_j : D_j \, X \, D_j \rightarrow [0,1]$ such that for $x, y, z \in D_j$,

(i) $S_j(x, x) = 1$ (reflexivity)
(ii) $S_j(x, y) = S_j(y, X)$ (symmetry)
(iii) $S_j(x, Z) \geq \max\{\min[s_j(x, y), s_j(y, z)]\}$ (max-min transitivity).

	Critical	Severe	Stable	Normal
Critical	1	0.7	0.4	0
Severe		1	0.8	0.3
Stable	--		1	0.6
Normal	--	--	--	1

Table 7 Similarity Degree Relation

A proximity relation [42] is a mapping,

$S_j : D_j \, X \, D_j \rightarrow [0,1]$, such that for x, y □ D_j,

(i) $S_j(x, x) = 1$ (reflexivity)
(ii) $S_j(x, y) = S_j(y, X)$ (symmetry)

Table 7 represents the similarity degree S of fuzzy terms of patient condition.

10Fuzzy Query Languages

There are research proposals regarding the development of fuzzy query language (FQL) which can integrate with the traditional SQL to work in relational database environment. Unfortunately there is no significant progress regarding the implementation of FQL for relational databases. The development of FQL requires the firm theoretical basis for integrating fuzzy logic and fuzzy set theory proposed by Zadeh [42] with the classical relational model proposed by Codd [1]. There are many important considerations and concepts regarding the development of FQL [57]. However there are some important proposals regarding the development of FQL are summarized in table 8.

Fuzzy-Temporal Query Language	Salient Features
Fuzzy extension of SQL [58]	Introduced an application, called HAVANE,with two main constraints: the number of answers has to be controlled and some parts of queries are not Boolean conditions. Allow fuzzy queries against a usual database, i.e. composed of precise facts.
FQL [59]	FQL for the relational databases is to be constructed as an enhancement of the relational domain Allow four new types of the complex fuzzy statements. Each fuzzy statement is formally represented as a certain fuzzy set. In the similar way, each fuzzy WFF (F-WFF) is represented by a certain fuzzy set.
FSQL [60]	FSQL is an extension of the SQL language. Allows flexible queries for both traditional or fuzzy attributes Following linguistic labels may be defined on any attribute. Fuzzy Comparators Fulfilment thresholds (T) Function CDEG(attribute): Fuzzy Constants
FTS-SQL [61]	FTS-SQL satisfies the closure property Queries can be on both crisp and fuzzy attributes FTS-SQL allows source options
FTER query language [17]	Extension of TQuel The concepts of fuzzy temporal expressions Fuzzy temporal selection fuzzy temporal join fuzzy temporal projection supports both valid and transaction times
FOQL [62]	FOQL is an extension of OQL formulate queries in a more natural way FOQL is based on ODMG standard FOQL provides an explicit mechanism for controlling the number of images output/displayed using threshold and index collection constructs controlling the number of images output/displayed using threshold and index collection constructs.

Table 8 Fuzzy-Temporal Query Languages

11Conclusion

This paper has pointed out the important notions of time which are significant in building temporal ontology for the design of temporal database applications. There are many research proposals for temporal database model which mainly includes the extension of the relational model and in object oriented models. We mainly focused on the relational model and surveyed some of the important temporal database models proposed along with temporal query languages.

Database management system is responsible for managing huge amount of data related to a specific domain. In most of the applications only precise and crisp data is recorded in the database; however there are some critical applications where data is often vague, imprecise and uncertain. This uncertain, imprecise, and non-crisp nature of the data can be managed with the help of fuzzy logic and there are extensions of the relational model to manage fuzzy data in the literature.

The fuzziness may be introduced at different levels such as attribute, object, entity, class, relation, relationship etc. Most of the proposed models introduced fuzziness at attribute level. This paper presents several approaches to manage fuzziness and advocates the importance of introducing fuzziness at different levels such as relation, attribute, relationship and constraint definition. There are many real world applications which involve temporal data and also involve fuzziness at the same time, and similarly fuzzy data may have temporal dimension. This paper also presents the important consideration in the development of FTRDM.

Fuzzy ontology is another important concept because the definition of the concepts, characterization, relationships and the terminology is essential to describe a given mini world. Fuzzy ontology frameworks are proposed for different domain applications, however there is no significant research published in the area of fuzzy temporal database system and there is a need of a generic fuzzy and temporal ontological framework for fuzzy relational database.

The conceptual database model is a back bone of any database application. Traditional entity relationship model is not capable to represent fuzzy and temporal concepts in the model. This paper investigates the importance of extended ER models to capture fuzzy and temporal data. There are very few extensions of ER model to capture both fuzzy and temporal dimension. This paper also summarizes the important contributions in the development of fuzzy and temporal conceptual models with some important characteristics.

This paper is mainly focus on the extension of the relational model to incorporate fuzzy and temporal data. This paper further investigates the important issues regarding the fuzzy and temporal aspects and reviewed the important contributions in this domain. However there are some research rears which need further attention including fuzzy temporal relational algebra, fuzzy constraint definitions and query optimization methods.

References:

[1] E. F. Codd (1970) A relational model of data for large shared data banks. Communcations of the ACM Vol. 13, 1970, pp. 377–387.

[2] C. S. Jensen and R. A. Snodgrass, Temporal data management. IEEE Trans. Knowledge Data Eng. Vol. 11, No. 1, 1999, pp. 36–44.

[3] A. U. Tansel , J. Clifford , S. Gadia , S. Jajodia , A. Segev , R. Snodgrass, Temporal databases: Theory, Design, and Implementation, Benjamin-Cummings Publishing Co., USA, 1993.

[4] A. U. Tansel, Adding Time Dimension to Relational Model and Extending Relational Algebra. Information Systems, Vol. 11, No. 4, 1986, pp. 343-355.

[5] J. Clifford, and A. Croker, The Historical Relational Data Model (HRDM) and Algebra Based on Lifespans. In Proc. of the International Conf. on Data Eng. IEEE Computer Society Press, 1987, pp. 528-537.

[6] R. T. Snodgrass, The Temporal Query Language TQuel. ACM Transactions on Database Systems, Vol. 12, No. 2, 1987 pp 247-298.

[7] S. K. Gadia, A Homogeneous Relational Model and Query Languages for Temporal Databases. ACM Transactions on Database Systems, Vol. 13, No. 4, 1988, pp. 418-448.

[8] S. K. Gadia, and C. S. Yeung, A Generalized Model for a Relational Temporal Database. In Proc. of the ACM SIGMOD Int. Conf. on Management of Data, Chicago, 1988, pp. 251-259.

[9] N. A. Lorentzos, The Interval-extended Relational Model and its Application to Validtime Databases. In A. Tansel, J. Clifford, S. Gadia, S. Jajodia, A. Segev, and R. Snodgrass, editors, Temporal Databases: Theory, Design, and Implementation, Benjamin/Cummings Publishing Company, 1993, pp. 67-91.

[10] S. Navathe, and R. Ahmed, Temporal Extensions to the Relational Model and SQL. In A. Tansel, J. Clifford, S. Gadia, Jajodia, S., Segev, A. and Snodgrass, R. editors, Temporal Databases: Theory, Design, and Implementation, Benjamin/Cummings Publishing Company, 1993, pp. 92-109.

[11] R. T. Snodgrass, editor, The TSQL2 Temporal Query Language. Kluwer Academic Publishers, USA, 1995.

[12] Aqil Burney, Nadeem Mahmood, Kamran Ahsan, TempR-PDM: A Conceptual Temporal Relational Model for Managing, Patient Data, Proceedings Int. WSEAS conference on Recent Advances in Artificial Intelligence, Knowledge Engineering and Data Bases, University of Cambridge, UK, (2010a) pp. 237-243.

[13] S. Andreas "A Generalisation Approach to Temporal Data Models and their Implementations" PhD Thesis Departement Informatik, ETH Zurich, November 1997 Extract: Origins of ATSQL2.

[14] J. Chomicki, Temporal Query Languages: A Survey. In Gabbay, D. and Ohlbach, H., editors, Temporal Logic, First Int. Conf., Springer-Verlag, LNAI 827, 1994, pp 506-534.

[15] G. Ozsoyoglu and R. T. Snodgrass, Temporal and Real-Time Databases: A Survey, IEEE Transactions on Knowledge and Data Eng., Vol. 7 No. 4, 1995, pp. 513-532.

[16] D. Dubois, and H. Prade, Processing fuzzy temporal knowledge. IEEE Trans. Syst. Man Cybern. Vol. 19, No. 4, 1989, pp. 729-744.

[17] L. Deng, Z. Liang, Y. Zhang, A Fuzzy Temporal Model and Query Language for FTER Databases, Eighth IEEE Int. Conf. on Intelligent Systems Design and Applications 2008.

[18] J. Galindo, (Ed.), Handbook of Research on Fuzzy Information Processing in Databases. Hershey, PA, USA: Information Science Reference, 2008.

[19] A. Burney, M. Nadeem, J. Tahseen, H. Saleem, (2010b). Conceptual Fuzzy Temporal Relational Model (FTRM) for Patient Data. WSEAS Transactions on Information Science and Applications Issue 5, Volume 7, 2010, pp 725-734.

[20] N. Chaudhry, J. Moyne, and E. A. Rundensteiner, An extended database design methodology for uncertain data management, Information Sciences 121, 1999, pp. 83-112.

[21] P. P. Chen, The entity-relationship model: toward a unified view of data, ACM Transactions on Database Systems, Vol. 1, No. 1, 1976, pp. 9-36.

[22] H. Gregersen and C. S. Jensen, Temporal Entity-Relationship Models-A Survey, IEEE Transactions on Knowledge and Data Eng. Vol. 11, No. 3, 1999, pp. 464-497.

[23] Z. M. Ma & Li Yan, A Literature Overview of Fuzzy Conceptual Data Modeling, Journal of

Information Science and Engineering Vol. 26, 2010, pp. 427-441.
[24] C. Bettini, S. Jajodia, and S. Wang, Time Granularities in Databases, Data Mining, and Temporal Reasoning. Berlin: Springer-Verlag, 2000.
[25] W. Kurutach, Handling Fuzziness in Temporal Databases. In: Proc. of IEEE Int. Conf. on Systems, Man and Cybernetics, Vol. 3, 1998, pp. 2762–2767.
[26] C. S. Jensen, and C. E. Dyreson, (Edt.) The Consensus Glossary of Temporal Database Concepts - February 1998 Version. In Temporal Databases: Research and Practice, volume 1399 of Lecture Notes in Computer Sci., 1998, pp. 367-405, Springer-Verlag.
[27] M. H. Bohlen, R. Busatto, and C.S. Jensen, "Point versus Interval Based Temporal Data Models,"Proc. IEEE Int. Conf. Data Eng., 1998, pp 192-200.
[28] J. Allen, Maintaining knowledge about temporal intervals, Communications of the ACM Vol. 26, No. 11, 1983, pp.832-843.
[29] A. N. Prior. Time and Modality. Clarendon Press, 1957.
[30] M. D. Gabby, A. M. Reynolds, M. Finger, Temporal Logic Mathematical Foundations and Computational Aspects, Volume 2, Oxford Science Publications, University of Oxford, 2000.
[31] L. Vila, A survey on temporal reasoning in artificial intelligence, AI Communications Vol. 7, No. 1, 1994, pp. 4-28.
[32] M. Baudinet, J. Chomiki, P. Wolper, Temporal deductive databases. In: Tansel A, Clifford J, Gadia S, Jajodia S, Segev A, Snodgrass R (Edt.) Temporal data bases: theory, design and implementation. The Benjamin Cummings Pub. Co. California, 1993.
[33] J. Chomicki and D. Toman. Temporal Logic in Database Query Languages. Encyclopedia of Database Systems, 2009: pp. 2987-2991 NewYork: Macmillan.
[34] N. A. Lorentzos and R. G. Johnson, TRA: A Model for a Temporal Relational Algebra. Proc. Of Conf. on Temporal Aspects in Information Systems, Sophia-Antipolis, France, North Holland Publishing Company, 1987.
[35] A. Tuzhilin and J. Clifford. A Temporal Relational Algebra as a Basis for Temporal Relational Completeness. In D. McLeod, R. Sacks-Davis, and H.-J. Schek, editors, International Conference on Very Large Data Bases, VLDB'90, 1990, pp 13–23.
[36] D. Dey, T. Barron, and V. Storey. A complete temporal relational algebra. VLDB Journal, Vol. 5, No. 3, 1996, pp 167-180.
[37] J. E. McKenzie and R. T. Snodgrass, Supporting Valid Time in an Historical Relational Algebra: Proofs and Extensions. Technical Report TR_91_15, Department of Computer Science, University of Arizona, Tucson, AZ, 1991.
[38] N. A. Lorentzos and Y. G. Mitsopoulos, SQL Extension for Interval Data, IEEE Transactions on Knowledge and Data Engineering, Vol. 9, No. 3, 1997, pp. 480-499.
[39] M. H. B¨ohlen and C. S. Jensen, Seamless Integration of Time into SQL. Technical Report R-96-2094, Institute for Electronic Systems, Aalborg University, 1996.
[40] Toman, D., Point-based temporal extensions of SQL. In International Conference on Deductive and Object-Oriented Databases, 1997.
[41] Ariav, G., 'A temporally oriented data model', ACM Transactions on Database Systems Vol. 11, No. 4, pp. 499-527, 1986.
[42] L. A. Zadeh, Fuzzy Sets. Information and Control, Vol. 8, 1965, pp. 338-353.
[43] L. A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, Information Sciences Vol. 8, 1975, pp. 43-80.
[44] J. F. Brule, 1992. Fuzzy systems – a tutorial. From baechtel@iccgcc.decnet.ab.com Newsgroup: comp.ai. 11 pp
[45] J. Campana, M. C. Garrido, N. Marin and O. Pons, "A Fuzzy Set-Based Approach to Temporal Databases", Scalable Uncertainty Management, 2007, pp. 31-44.
[46] D. Dubois and H. Prade, Fuzzy Numbers. An Overview. In: Bezdek (Ed.) The Analysis of Fuzzy Information. CRS Press, Boca Raton, 1985.
[47] J. M. Medina, M. A. Vila, J. C. Cubero and O. Pons, Towards the implementation of a generalized fuzzy relational database model. Fuzzy Sets and Systems, Vol. 75, 1995, pp. 273-289.
[48] S. Abiteboul, and R. Hull, IFO: A formal semantic database model, in: ACM Transactions on Database Systems, Vol.12, No.4, 1987, pp. 525-565.
[49] Z. M. Ma, W. J. Zhang, W. Y. Ma, G. Q. Chen, Conceptual Design of Fuzzy Object-Oriented Databases Using Extended Entity-Relationship Model, International Journal of Intelligent Systems, Vol. 16, 2001, pp. 697-711.

[50] G. Q. Chen, and E. E. Kerre, Extending ER/EER concepts towards fuzzy conceptual data modeling, Proceedings of the 1998 IEEE Int. Conf. on Fuzzy Systems, Vol. 2, 1998, pp. 1320-1325.

[51] A. Zvieli, and P. P. Chen, Entity-relationship modeling and fuzzy databases, Proc. of the 1986 IEEE Int. Conf. on Data Eng., 1986, pp. 320-327.

[52] S. Ben Yahia, H. Ounalli and A. Jaoua, An extension of classical functional dependency: dynamic fuzzy functional dependency, Information Sciences 119 (1999), pp. 219-234.

[53] J. C. Cubero and M. A. Vila, "A new definition of fuzzy functional dependency in fuzzy relational databases," Int. J. Intell. Syst., Vol. 9, No. 5, 1994, pp. 441–448.

[54] Z. M. Ma, Li Yan and Gui Li (2005) Using Fuzzy Analogical Reasoning to Refine the query answers for relational databases with imprecise information. Lipo Wang Yaochu Jin (Eds.) Fuzzy Systems and Knowledge Discovery 2nd Int. Conf., FSKD 2005, Changsha, China, August 2005 Proceedings, Part 1 Springer-Verlag Berlin Heidelberg. pp 267-275.

[55] M. Y. Chen and D. A. Linkens, "Rule-base self-generation and simplification for data-driven fuzzy models," Fuzzy Sets Syst., Vol. 142, No. 2, 2004, pp. 243-265.

[56] M.Setnes,R.Babuska,U.Kaymak,andH.R.vanNa utaLemke,"Similarity measures in fuzzy rule base simplification," IEEE Trans., Syst.,Man, Cybern., Vol. 28, 1998, pp. 376–386.

[57] J. Galindo, "New Characteristics in FSQL, a Fuzzy SQL for Fuzzy Databases". WSEAS Transactions on Information Science and Applications 2, Vol. 2, 2005, pp. 161-169 (World Scientific and Engineering Academy and Society, www.wseas.org, ISSN: 1790-0832).

[58] P. Bosc, M. Galibourg, and G. Hamon, "Fuzzy Querying with SQL: Extensions and Implementation Aspects," Fuzzy Sets and Systems, Vol. 28, 1988, pp. 333-349.

[59] Y. Takahashi, A fuzzy query language for relational databases. In P. Bosc & J. Kacprzyk (Eds.), Fuzziness in database management systems, Germany: Physica-Verlag. 1995, pp.365-384.

[60] R. Carrasco, M.A. Vila, J. Galindo, J. Fsql: a flexible query language for data mining. Enterprise Information Systems IV Dordrech, Holanda: Kluwer Academic Publisher, 2003, pp. 68-74.

[61] A. K. Sharma, A. Goswami, D. K. Gupta, FTS-SQL:A Query Language for Fuzzy Multidatabases Proceedings of the World Congress on Engineering, Vol. 1 WCE 2009, London, U.K

[62] S. Nepal, M. Ramakrishna, J. Thom, A fuzzy object query language (FOQL) for image databases. In: Proceedings of the Int. Conf. on Database Systems for Advanced Applications, 1999.