Merging Probabilistic & Null Values Utilising an Intuitionistic Fuzzy Relational Mediator

Boyan Kolev, Krassimir Atanassov, Panagiotis Chountas, Ilias Petrounias

Abstract— **An Integrated Data Management System is required for representing and managing indicative information from multiple sources describing the state of an enterprise. Current Data Management Systems model enterprises that are crisp. A crisp enterprise is one that is highly quantifiable; relationships are fixed and attributes are atomic valued. The premises for this paper are precise enterprises, data maybe uncertain; multiple sources of information do exist, but uncertainty may be described using different models.**

Index Terms— **Intuitionistic Fuzzy Logic, Null Values, Probabilistic data, Value Uncertainty**

I. INTRODUCTION

The integration of information from multiple information providers has been a lasting problem of research. One may distinguish very roughly between two approaches.

- A stream of researchers is dealing with the semantic integration of independent sources through a common conceptual schema. This is called resolution of intentional inconsistencies.
- A second stream of researchers is dealing with the problem of defining a common answer from sources emitting conflicting answers. This is known as resolution of extensional inconsistencies.

The goal of the first stream is to provide flexible and efficient access to the population of information providers, by means of a global conceptual schema that models the information, contained in the entire population of information providers. This global conceptual schema is qualified with a mapping that defines the elements of the global schema, in terms of elements of the schemes of the information providers under conceptual integration. Queries are translated to queries on the population of information providers; the individual answers are then combined to answer the global query.

The goal of the second stream is to define a common acceptable answer for all sources, with reference to a particular request. However both streams of researchers assume that data are perfect, entail no uncertainty when it comes to querying.

Boyan Kolev is with the CLBME – Bulgarian Academy of Sciences, Bl. 105, Sofia-1113, BULGARIA,(e-mail: bik@clbme.bas.bg)

Krassimir Atanassov is with the CLBME, Bulgarian Academy of Sciences, Bl. 105, Sofia-1113, BULGARIA, (e-mail: krat@argo.bas.bg)

Panagiotis Chountas is with University of Westminster-Health Care Computing Group, HSCS, Northwick Park, London, HA1 3TP, UK, (e-mail: chountp@wmin.ac.uk)

Ilias Petrounias is University of Manchester, Information Systems Group, School of Informatics, PO Box 88, Manchester M60 1QD, UK, (e-mail: ilias.petrounias@manchester.ac.uk)

In this paper, we report an attempt towards an Intuitionistic Fuzzy Mediator, where integrated data in a "relation" represent all possible views with respect to *same* real world proposition. Such a type of a Mediator should allow the querying of multiple sources containing data that entail some type of value uncertainty. Value uncertainty is expressed mainly with the aid of fuzzy, probabilistic and null values as part of a relational data source.

In a Mediator environment the autonomy of the sources under integration must be guaranteed. This simply means that local sources are allowed to choose and express themselves using their own model of value uncertainty (i.e. fuzzy, probabilistic, null value). The Mediator level should however provide a uniform way to integrate the underlying sources. In other words users must be in position to interrogate mediator architecture-system in a single and uniform way.

We report an Intuitionistic Fuzzy Relational Mediator for the merging of probabilistic and null values when it comes to querying of multiple sources containing uncertain data. In our case we focus on probabilistic or null values that are coming from different relational data sources.

The rest of the paper is organized as follows

II. NULL & PROBABILISTIC VALUES IN RELATIONAL DATABASES

In any extended relational environment somebody may distinguish among others two models for treating uncertainty at the attribute-label type which are listed below

- Probabilistic values
- Null values

Probabilistic information is a deviation of ignorant-uncertain information. A probabilistic data value is a set of alternatives. Each alternative has an associated probability that it is the attribute-label value [1], [2], [3]. Such a data value is a set of alternatives, each of which has an associated probability. In [4] the notion of a missing probability is introduced, which is the way to model ignorance in probabilistic data. Ignorance is derived from incomplete data. For example, if the sum of the probabilities of all possible outcomes is less than 1.0, the remaining part is assigned to the so-called missing probability.

A null value represents an *unknown* attribute value, is a value that is known to exist, [5], [6] but the actual value is unknown. The unknown value is assumed to be a valid attribute value, that is, some value in the domain of that attribute. This is a very common kind [7] of ignorant information. An unknown value has various names in the

literature including *unknown null* [8], *missing null* [9], and *existential null* [10].

The meaning of a fact, *F*, with an unknown attribute value over an attribute domain of cardinality *N* is a multiset with *N* members; each member is a set containing an *F* instance with the unknown value replaced a different value from the attribute domain.

The Intuitionistic fuzzy set theory is an extension of the classical fuzzy set theory [11]. Each element of an Intuitionistic fuzzy set has degrees of membership (μ) and non-membership (ν), which don't sum up to 1.0 thus leaving a degree of indefiniteness (π). These are the attractive properties that make Intuitionistic fuzzy set theory a good candidate for expressing belief in the sense of probabilistic databases, and unknown in the sense of databases with null values.

III. INTUITIONISTIC FUZZY RELATIONAL MEDIATOR

In this paper we present the principles for an Intuitionistic Fuzzy Mediator for querying different types of uncertain data from multiple sources. We focus on the interrogation of different sources that may contain probabilistic, fuzzy, and null values under a single platform named as the *"Intuitionistic Fuzzy Mediator"*.

For implementing the Intuitionistic-Fuzzy Mediator architecture the following components need to be defined

- An extended Intuitionistic-Fuzzy Mediator tabular data model with respect to the Mediator level. This will let the users to define different types of queries that may entail a level of uncertainty. The Mediator will also use this model for collecting all the information from the different data sources through the Binders and united it in a single form before it sends them back to users.
- A query language for the utilisation of the binders. Binders convert Intuitionistic fuzzy types queries expressed at the Mediation Level with respect to the semantics of the underlying data sources (i.e. probabilistic, models of uncertainty). Binders collect all available data with respect to a query from multiple sources and convert them to an Intuitionistic fuzzy form before they forward these data to the Mediator - User.
- An enhanced query environment that it will facilitate query processing and optimisation

Finally we present our experiences from trying to implement the "*Intuitionistic Fuzzy Mediator"* with the aid of Intuitionistic Fuzzy PostgreSQL, which is based on the PostgreSQL DBMS.

IV. INTUITIONISTIC FUZZY RELATIONAL MODEL-MEDIATOR LEVEL

In this section we define the Mediator data model and its basic operations. This model will be used for the logical interaction between users and sources with the aid of Binders. It will form the common protocol for the interaction between these elements.

Let *R* be an Intuitionistic fuzzy relation (IFR),

$R = \{<x, \mu_R(x), \nu_R(x)> / x \in X\}$, where $x = <att_1, \ldots, att_n>$ is an ordered tuple belonging to a given universe X, $\{att_1,\ldots,att_n\}$ is the set of attributes of the elements of X, $\mu_R(x)$ is the degree of membership of x in the relation R. In other words, R is an intuitionistic fuzzy subset of X with membership and non-membership functions μ_R and ν_R respectively.

A selection $\sigma_P(R) = \{<x, min(\mu_R(x), \mu(P(x))), max(\nu_R(x), \nu(P(x)))> / x \in X\}$, operation defines a relation, which contains only those tuples from *R* for which a certain predicate *P* is satisfied. We can say that the selection modifies the degrees of membership and non-membership of *R* depending on the corresponding value of the predicate: *P*. The answer-result is a relation that has a degree of membership, which is logically *AND*-ed with the corresponding value of the predicate P.

Project: $\pi_P\{<x, \mu_R(x), \nu_R(x)> / x \in X\}$. The traditional project operator $\pi_f(R)$ selects all attributes f from all tuples in R leaving out other attributes not in f. The semantics of a Intuitionistic fuzzy project operator $\pi_P\{<x, \mu_R(x), \nu_R(x)> / x \in X\}$should be that it selects all attributes f from all *possibilities* in R. Projection does not affect the associated Intuitionistic fuzzy measures. However, if the key k of R is projected out, the individual objects can no longer be identified and the result is meaningless. Therefore, k always has to be part of attribute list f.

Union: $P(A \cup B) = \{<x, max(\mu_A(x), \mu_B(x)), min(\nu_A(x), \nu_B(x))> / x \in X\}$ The Intuitionistic fuzzy union operator merges two IFRs possibly containing possibilities for the same real world objects. To properly calculate the Intuitionistic fuzzy measures in the answer, it is beneficial to enumerate the possible worlds, i.e., consider each possibility of an element existing or not in the operand sets. The intersection and difference can be determined analogously.

A *Cartesian product* of two relations $R \times S$ is identical to the Cartesian product operation defined in the intuitionistic fuzzy sets theory [5], which uses the logical *AND* between the degrees of membership:

Let *S* be another intuitionistic fuzzy relation: $S = \{<y, \mu_S(y), \nu_S(y)> / y \in Y\}$, then:

$$R \times S = \{ <<x, y>, min(\mu_R(x), \mu_S(y)), max(\nu_R(x), \nu_S(y),)> / <x, y> \in X \times Y\}.$$

For the accommodation of probabilistic databases the IFRDB model defines probabilistic variants of Cartesian product and selection operations, section 4. The definition of these operations is based on the notion of a probabilistic conjunction (logical AND). This type of conjunction is applied when the operands carry probabilistic semantics, i.e. they express a probability, not a degree of membership. Therefore the result of the conjunction is an Intuitionistic fuzzy value, in which the degree of truth is the multiplication of the degrees of truth of the operands (analogous to the formula for intersection of probabilities) and the degree of falsity is computed by a formula analogous to the formula for union of probabilities.

In activating these operators as part of the Intuitionistic fuzzy extension of PostgreSQL, it must be explicitly specified the type of the relational sources (i.e. probabilistic, etc) to be merged in order the IFRDBMS to process correctly the result of the query.

V. INTUITIONISTIC FUZZY REPRESENTATION OF PROBABILISTIC DATA-BINDER LEVEL

Consider the sample probabilistic relation *Lives_in* on 'Fig. 4.1'. The relation contains the places where John and Peter live. The column *lives_in* represents the probability distributions of the place where each man lives. Since the sources of information are different, we can't determine which the exact places, where the both men reside, are. We only know them with certain degrees of confidence. The asterisk represents the missing probability or – in the context of the example – the probability that John (Peter) lives "somewhere".

Name	***Lives_in***
John	London [0.4] Manchester [0.2] Soho [0.1] * [0.3]
Peter	London [0.3] Manchester [0.5] * [0.2]

Fig. 4.1 Probabilistic relation *Lives_in*

- The quantity in the brackets expresses a relative onfidence *m(S)* in the given singleton fact *S* and not in any subset of *S*. It is called basic probability assignment.
- The total confidence in *S*, which is called *belief,* is the sum of the probability assignments commited to all subsets of *S*, *bel(S)*.

The plausibility is defined, $Pl(S)=1 - Bel(\neg S)$

The ignorance is defined , $\pi = Pl(S) - Bel(S)$

In the example above, if *S* = "*John lives in London*", the basic probability assignment is *m(S)* = 0.4, and the belief is *bel (S)* = 0.5, because Soho is located in London, therefore it is a subset of the fact *S*.

The plausibility is computed as *Pl (S)* = 1 – *Bel* (¬*S*) = 1 – *Bel* ("*John lives in Manchester*") = 0.8

Following the above mentioned rules, we create the table on 'Fig. 4.2', where each fact is represented along with its belief, plausibility and ignorance, which we use to compute the intuitionistic fuzzy degrees of truth and falsity.

Name	***Lives_in***	***Bel = μ***	***pl***	***1 – pl = ν***	***ig = π***
John	London	0.5	0.8	0.2	0.3
John	Manchester	0.2	0.5	0.5	0.3
John	Soho	0.1	0.8	0.2	0.7
Peter	London	0.3	0.5	0.5	0.2
Peter	Manchester	0.5	0.7	0.3	0.2

Fig.: 4.2 Intuitionistic fuzzy relation *Users*

This probabilistic data can be stored in the intuitionistic fuzzy relation as the degrees of membership and non-membership are populated with the values in the columns μ and ν. To convert the probabilistic data into intuitionistic fuzzy data, we map the belief to the degree of truth and the ignorance to the degree of indefiniteness, i.e. the degree of falsity is :

$$\nu = 1 - Bel - Ig = 1 - Pl$$

Performing selection operation using IFRDBMS

For the accommodation of probabilistic databases the IFRDB model defines probabilistic variants of selection as follows:

$\sigma_P(R) = \{<x, \mu_R(x).\mu(P(x)), \nu_R(x) + \nu(P(x)) - \nu_R(x).\nu(P(x))> / x \in X\}$,

Consider the following probabilistic distribution about Peter's birth place:

{Manchester [0.6], Liverpool [0.3], * [0.1]}

So, if we define an intuitionistic fuzzy predicate *is_the_birth_place_of_Peter (town)*, it will have the following values:

is_the_birth_place_of_Peter (Manchester) = <0.6, 0.3>

is_the_birth_place_of_Peter (Liverpool) = <0.3, 0.6>

Now, assuming that the IFRDBMS has to execute the query "Find all users who live in Peter's birth place", it should be formulated with the following IFSQL command:

```
SELECT name, lives_in, mship, nmship
FROM users
WHERE is_the_birth_place_of_Peter (lives_in);
```

And the results will be the ones, represented in 'Fig. 4.3' (we note that the values of the belief and plausibility are respectively: *bel* = μ and *pl* = 1 - ν

Name	***Lives_in***	***μ***	***ν***
John	Manchester	0.12	0.65
Peter	Manchester	0.3	0.51

Fig. 4.3 Results from selection operation

Performing join using IFRDBMS

The relational operation Cartesian product applies using the formula based on the values of the belief and plausibility (or, more exactly, on their degrees of truth and falsity) of the facts rather than their basic probability assignments.

For the accommodation of probabilistic databases the IFRDB model defines probabilistic Cartesian product as follows:

$R \times S = \{<<x, y>, \mu_R(x).\mu_S(y), \nu_R(x) + \nu_S(y) - \nu_R(x).\nu_S(y)> / <x, y> \in X \times Y\}$

Towns_p:

Town	***Dist.***
London	200 [0.4]
	250 [0.6]
Manchester	150 [1.0]

Towns:

Town	***Dist.***	***μ***	***ν***
London	200	0.4	0.6
London	250	0.6	0.4
Manchester	150	1.0	0.0

Fig. 4.4 Probabilistic relation *Towns_p* and the equivalent intuitionistic fuzzy relation *Towns*

For example, consider the probabilistic relation *Towns_p*, which contains the distances to the towns and the corresponding intuitionistic fuzzy relation *Towns* (see 'Fig.. 4.4').

The answer of the question "How far from here does Peter live?" (presented on 'Fig.4.5') can be achieved by applying join operation over the two relations which is formulated with the following IFSQL command:

SELECT u.lives_in, t.town, t.distance, mship, nmship FROM users u JOIN towns t ON u.lives_in = t.town WHERE u.name = 'Peter'

Town	***Distance***	μ	ν
London	200	0.12	0.8
London	250	0.18	0.7
Manchester	150	0.5	0.3

Fig. 4.5 Results from the join

VI. INTUITIONISTIC REPRESENTATION OF NULL VALUES BINDER LEVEL

In relational databases with null values it is recognised [12] that there are many different types of null values, each of which reflects different intuitions about why a particular piece of information is unknown.

In [12], five, different types of nulls are suggested 'Fig. 5.1'. The labels and semantics of them are defined as follows. Let V be a function, which takes a label and returns a set of possible values that the label may have.

Label (X)	V(x)
Ex-mar	D
Ma-mar	D $\cup$ {$\perp$}
Pl-mar	{$\perp$}
Par-mar (Vs)	Vs
Pm-mar (Vs)	Vs $\cup$ {$\perp$}

Fig. 5.1 Types of Null Values

Intuitively, V (Ex-mar) = D says that the actual value of an existential marker can be any member of the domain D. Likewise, V (Ma-mar) = D $\cup$ {$\perp$} says that the actual value of a maybe marker can be either any member of D, or the symbol $\perp$, denoting a non-existent value. Similarly, V (Par-mar (Vs)) = Vs says that the actual value of a partial null marker of the form pa mar (Vs) lies in the set Vs, a subset of the domain D.

A controversial issue is the use of the $\perp$, which denotes that an attribute is inapplicable. Certainly this is the interpretation of an algebraic manipulation of the unknown information, instead of a conceptual manipulation and interpretation. Conceptually the issue can be resolved with the use of the subtypes. A subtype is introduced only when there is at least one role recorded for that subtype. In the general case the algebraic issue under the use of subtypes is whether the population of the subtypes in relationship to the super-type is:

- Total and Disjoint: Populations are mutually exclusive and collectively exhaustive.
- Non-Total and Disjoint: Populations are mutually exclusive but not exhaustive.
- Total and Overlapping: Common members between subtypes and collectively exhaustive, in relationship to super-type.
- Non-Total and Overlapping: Common members between subtypes and not collectively exhaustive, in relationship to super-type.

The conceptual treatment of null will permit us to reduce the table in 'Fig. 5.1' using only two types of null markers.

Label (X)	V(X)
P-mar (Vs)	{Vs}
Π-mar (D)	{D}

Fig.. 5.2 Types of Null Values: Eliminated Null Markers

In this case the P-mar, marker is used to denote that one of the members of the restricted set Vs $\subseteq$ D, is the actual value, of an attribute value, like in the case of partial values. Partial values may be thought of as a generalisation of null values where, rather than not knowing anything about a particular attribute value, as is the case for null values, we may identify the attribute value as belonging to a set of possible values assuming some type of *background knowledge*. A partial value is therefore a set such that exactly one of the values in the set is the true value.

Background knowledge may be presented as a concept hierarchy of attributes, as integrity constraints, from the integration of conflicting databases, or from knowledge possessed by domain experts. Using such information we propose to re-engineer the database by replacing missing, data by sets of the attribute domain. Concept hierarchies have previously been used for attribute-induced knowledge discovery. However our use of background knowledge in this context is novel.

We assume that attribute values may be given as Intuitionistic Fuzzy concepts, which correspond to proper subsets of the domain. In addition there are rules describing the domain, and these may be formed in a number of ways: they may take the form of integrity constraints, where we have certain restrictions on domain values; they may arise from the integration of conflicting attribute values in multi databases, they may be also rules specified by a domain expert, with the aid of data analysis or *data mining*.

To this extent we can treat a null value as a probabilistic type information and treat it in terms of query processing with the aid of Intuitionistic Fuzzy logic as this defined in "Section four".

A fact may also represent information that is compatible with its domain, based on some specified or unspecified criterion. However, the information source cannot testify these values, but does not have any reason to reject them. In this latter case the Π-mar, marker to denote that a label

value can be "any" value derived from the label domain D. To this extent we can treat a null value in the sense of fuzzy type information. For this reason we are currently working on defining a parallelism between fuzzy data and Intuitionistic fuzzy data.

VII. KNOWLEDGE AND QUERY PROCESSING

So far we shown how querying is performed with the aid of aid of Intuitionistic Fuzzy PostgreSQL (IFPG), which supplements the well known open-source RDBMS PostgreSQL. The IFPG extends the fundamental relational operators, as these defined as part of the IFRDB Model in section three.

As part of the IFPG environment we can treat probabilistic data and null values-markers. We have shown the correspondence between probabilistic and Intuitionistic fuzzy data and how to replace them with the equivalent Intuitionistic fuzzy distribution that corresponds to probabilistic treatment of information. As for null values – markers we treat them as a kind of partial value in terms of data replacement and in terms of query processing we could treat them as as a probabilistic type information with the aid of Intuitionistic Fuzzy logic as this defined in "Section four".

At this point we realise that it will be useful to propagate a user request-query not to all data sources participating in the Mediation architecture, but only to those sources that are relevant. This is quite critical when it comes to query processing and execution, since it will allow faster query execution. For this reason we equip the Mediation Architecture with a repository that contains the description of constraints related to all underlying data sources. Using the Constraints Repository a particular query can determine if a data source contains relevant content and consequently if it needs to be queried or not.

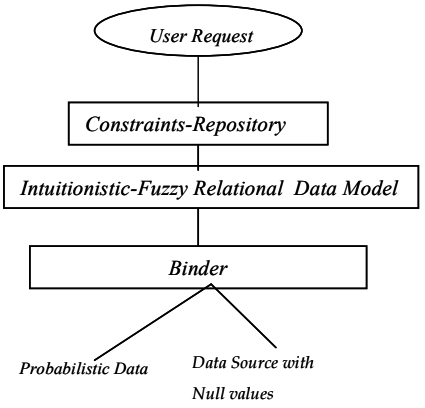

Fig.7.1. Intuitionistic Fuzzy Relational Mediator

In a Mediator environment it will be useful not to query all sources, but only those that contain information relevant [13], [14], [15] to our request. This is quite critical for achieving better query performance. For this reason we equip our Mediation architecture with a repository that contains various constraints (i.e. Intuitionistic Fuzzy Range Constraints, Intuitionistic Fuzzy Functional Dependencies, association rules) that are related to the information sources that participate in the Mediation Architecture. To this extent we can interrogate the constraints repository to find out if a particular source contains relevant information with respect to particular request. Let us consider a sample mediator system with two information sources S_1, S_2 that contain demographic data about citizens across EU. They have the following obviously simplified schemas

Source S_1: S_1 (Name, Lives, Income): French Citizens

Source S_2: S_2 (Name, Lives, Income): Italian Citizens

In mediators, information sources i.e.(S_1, S_2) often have various constraints such as:

- Range constraints: such as "The average income per person is estimated to be in the range of €50K". Considering a finite universe of discourse, say X whose cardinality is N. Let us suppose that X={X_1, X_2, …. , X_n} and the Intuitionistic fuzzy number ~a given by

$$\sim a = \{(x_i, \mu_i, \nu_i): x_i \in X, I = 1,2\ldots N\}$$

We can express the above constraint as follows $\sim Income_{50K}$ {(49, .8, .1), (50, .9, .02) (51, .7, .15)}

- "All persons stored at a source have a unique identifier". A classical data integrity constraint
- Functional Dependencies: for instance, a source relation S_1(Name, lives, income) has a functional dependency Name→(Lives, ~Income).

These constraints are very useful to compute answers to queries. There are several reasons we want to consider constraints separately from the query language.

Describing constraints separately from the query language can allow us to do reasoning about the usefulness of a data source with respect to a valid user request.

Assume sources (S_1, S_2) and the following constraints:

- (**C_1**): All Citizens at source S_1, received an average income of $\sim Income_{60K}$
- (**C_2**): All Citizens at source S_2, received an average income of $\sim Income_{50K}$
- (**C_3**): Each citizen in EU (in S_1, S_2) has a unique, identifier. `Name→(Lives,~Income)`

These constraints carry a rich set of semantics, which can be utilized in query processing. For instance, consider the following queries that ask information about EU Citizens.

- Query **Q_1** asks for EU citizens receiving an amount of income under €51K_. We do not need to access S1 due to constraint **C_1**.
- Query **Q_2** asks for asks for EU citizens receiving an amount of income above €59K_. We do not need to access S2 due to constraint **C_2**.
- Query **Q_3** asks for citizens living in both places S_1, S_2 (Italy, France). We can take the natural join or intersection of these two sources on the name attributes to compute answers. Notice that we cannot compute the answers in this way if constraint **C_3** does not hold.

Some of constraints can be represented as *source constraints*. Each source constraint is defined on one data source only. A constraint for a data source is a set of conditions, such that for any projected database instance of source, these conditions must be satisfied by the tuples in the database. For example the first two constraints (**C_1 C_2**) in the EU example can be represented as source constraints. In particular, for C_1, the condition "$\sim Income_{60K}$"should be satisfied by any instance of source S_1, not just for a particular instance.

The same implies for C_2 with respect to source S_2.

What about C_3? Can we represent the constraint C_3 as two source constraints ($C_{3'}$, $C_{3''}$):

- $C_{3'}$ for S1: Name→(Lives, ~Income)
- $C_{3''}$ for S_2: Name→(Lives, ~Income)

However, they do not correctly describe C_3. These two source functional dependencies do not disallow the case where the two sources have two same persons that use different identifiers (i.e. Names). Clearly this case is not allowed by C_3. As a consequence, some queries may not be answered properly. Thus, if we replace C_3 with these two source constraints, we cannot take the natural join of S_1 and S_2 to answer the query Q_3. The limitations of source constraints show the need to use global constraints.

A global constraint is a condition that should be satisfied by any instance, for all data sources participating in the mediator system. Global constraints can be introduced during the design phase of a mediator system. They are used to capture the semantics of the application domain, no matter how many sources are in the system.

At this point we thought that it would be useful to have a tool to check and simulate the interaction between the constraints repository and the relational data sources that participate in the Mediator. We construct a Generalised network that simulates the interaction between the local sources and the constraints repository in real time. This will allow the activation of the constraints related to Mediation architecture as well as the update of data and their constraints.

VIII. SIMULATION OF THE INTUITIONISITIC FUZZY MEDIATOR

Generalized Nets (GNs, see [16]) are extensions of Petri nets and Petri net modifications and extensions. GN-transitions can have two temporal components (moment of transition firing and its duration), an index matrices (see [17]) - its (i,j)-th element is a predicate that determines whether a token from the i-th input place can be transferred to the j-th output place and other components. GNs can have three global time-components: moment of GN-activation, elementary time-step and duration of the GN-functioning. The GN-tokens enter the GN with initial characteristics and at the time of their transfer in the net they obtain next (current and final) characteristics. When a given GN has only a part of its components, it is called a reduced GN. A large number of operations, relations and operators (global, local, dynamical, and others) are defined over the GNs. Below we use a reduced GN with a part of the above described components: without the temporal ones. In [18] a bibliography on the publications and a short review of the results in GN theory are given.

The generalized net depicted on "*Fig.8.1"* represents the functionality of an intuitionistic fuzzy mediator dealing with multiple loacal relational sources. The mediator may use data, retrieved from the DBs, to apply a particular Data Mining Technique (DMT) in order to extract useful implicit information, background knowledge. The data sources may utilise various kinds of data uncertainty models, which makes it necessary to use a mediator that provides a uniform way to integrate the sources.

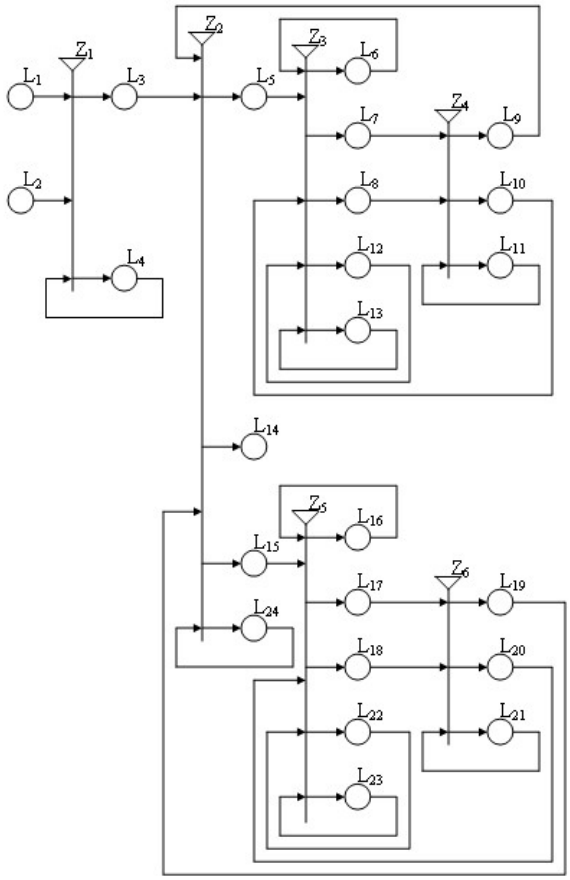

Fig.8.1. Generalized Net for the Intuitionistic Fuzzy Mediator

Description of the places

L_1: A new current request enters the GN
L_2: A new criterion enters the GN
L_3: The current request is presented with a canonical form
L_4: Contains a token which represents the criteria queue
L_5: A list of necessary DBs for the current request
L_6: Contains set of available DB tokens
L_7: Current DB request processing
L_8: Collection of all necessary DBs for the current request
L_9: Necessary information retrieved from DBs
L_{10}: DBs to return in L_6
L_{11}: Processing of necessary DBs for the current request
L_{12}: Contains all DBs that aren't used for a long time
L_{13}: A special token that will generate DBs
L_{14}: Answer to the request
L_{15}: A list of necessary DMTs for the current request
L_{16}: Contains set of available DMT tokens
L_{17}: Current DMT request processing
L_{18}: Collection of all necessary DMTs for the current request
L_{19}: Necessary information obtained by the means of DMT
L_{20}: DMTs to return in L_{16}
L_{21}: Execution of necessary DMTs for the current request
L_{22}: Contains all DMTs that aren't used for a long time
L_{23}: A special token that will generate DMTs
L_{24}: Current status of the mediator

As a token (we will refer to it as "request token") enters the GN through place L_1, its characteristic carry a new request to the mediator. Meanwhile some criterion tokens

may enter the GN through place L_2 and the criteria they carry in their characteristics are inserted into the criteria queue stored in the characteristic of the token in place L_4. After L_1 the request token moves to L_3 where it obtains a characteristic, which represents the canonical form of the request, considering some or all of the criteria from the characteristic of the token in L_4. The predicate matrix for transition Z_1 is the following:

$r_1 =$

	L_3	L_4
L_1	True	False
L_2	False	True
L_4	False	True

Transition Z_2 describes the interaction of the mediator with the DB sources. After L_3, the request token moves unconditionally to L_5, where the characteristic function considers the requested kinds of uncertainty and the status of the mediator (stored in the characteristic of the token in L_{24}) to produce a list of necessary DBs for processing the request. When the data is retrieved from the necessary DBs (this process is described in transitions Z_3 and Z_4 below), the token appears in place L_9 carrying the retrieved data in its characteristic. Afterwards the token moves either to L_{15}, if it is necessary to apply a DMT over the retrieved data, or to L_{14} otherwise. The predicate matrix of transition Z_2 is the following:

$r_2 =$

	L_5	L_{14}	L_{15}	L_{24}
L_3	True	False	False	False
L_9	False	W_1	$\neg W_1$	True
L_{19}	False	True	False	False
L_{24}	False	False	False	True

where:W_1 = *"DMTs need to be applied over the query results"*

In place L_6 reside all available database tokens. These are tokens, the characteristics of which contain all the data in a database. The databases may have different kinds of value uncertainty. If a database hasn't been used for a long time, the corresponding token moves from L_6 to L_{12}. When the creation of a new database is needed, the token in L_{13} splits and the newly created token moves to L_6.

After L_5 the request token moves to place L_7 where it stays until the data retrieval finishes. The database tokens needed for the current request move from L_6 to L_8. The predicate matrix for transition Z_3 is the following:

$r_3 =$

	L_6	L_7	L_8	L_{12}	L_{13}
L_6	$\neg W_2 \wedge \neg W_3$	False	W_2	W_3	False
L_5	False	True	False	False	False
L_{10}	True	False	False	False	False
L_{12}	False	False	False	True	False
L_{13}	False	False	False	False	True

where:
W_2 = *"the database is necessary for the current request"*
W_3 = *"the database hasn't been used for a long time"*

Transition Z_4 represents the process of working with the databases in order to process the current request. The request token stays in L_7 until the processing of the request finishes. Meanwhile each of the database tokens moves from L_8 to L_{11} where it loops while there is still work to do with it. Afterwards the database tokens shall return to place L_6 through place L_{10}. As soon as there are no more database tokens in L_{11}, the processing is considered finished and the request token moves to place L_9 where it obtains the retrieved data as new characteristic. The predicate matrix for transition Z_4 is the following:

$r_4 =$

	L_9	L_{10}	L_{11}
L_7	W_4	False	False
L_8	False	False	True
L_{11}	False	W_5	$\neg W_5$

where:
W_4 = *"no tokens in L_{11}"*
W_5 = *"no more work to do with the database"*

When the request token appears in place L_9, its characteristic contain the result of retrieving data from the DB sources. If the request necessitates the application of some DMT, the token shall proceed to place L_{15}, otherwise it exits the GN through L_{14}.

Transitions Z_5 and Z_6 are analogous to transitions Z_3 and Z_4 and describe the process of executing one or more DMTs according to the current request. Place L_{16} holds all available DMTs in the so called DTM tokens. If a DMT hasn't been used for a long time, the corresponding token moves from L_{16} to L_{22}. When a new DMT has to be introduced, the token in L_{23} splits and the newly created token moves to L_{16}.

After L_{15} the request token moves to place L_{17} where it stays until the DMT algorithm finishes its execution. The DMT tokens necessary for the current request are collected in place L_{18}. The predicate matrix for transition Z_5 is the following:

$r_5 =$

	L_{16}	L_{17}	L_{18}	L_{22}	L_{23}
L_{16}	$\neg W_6 \wedge \neg W_7$	False	W_6	W_7	False
L_{15}	False	True	False	False	False
L_{20}	True	False	False	False	False
L_{22}	False	False	False	True	False
L_{23}	False	False	False	False	True

where:
W_6 = *"the DMT has to be used in the current request"*
W_7 = *"the DMT hasn't been used for a long time"*

Transition Z_6 represents the process of executing the DMTs needed for the current request. The request token stays in L_{17} until the processing of the DMTs finishes. Meanwhile each of the DMT tokens is being processed in place L_{21}. Afterwards the database tokens shall return to place L_{16} through place L_{20}. As soon as there are no more database tokens in L_{21}, the processing is considered finished and the request token moves to place L_{19} where it obtains the result of the execution of the requested DMTs over the

retrieved data as new characteristic. The predicate matrix for transition Z_6 is the following:

$r_6 =$		L_{19}	L_{20}	L_{21}
	L_{17}	W_8	False	False
	L_{18}	False	False	True
	L_{21}	False	W_9	$\neg W_9$

where:

W_8 = *"no tokens in* L_{21}*"*

W_9 = *"no more work to do with the DMT"*

After L_{19}, the request token exits the GN through place L_{14}.

IX. Conclusions

The "*Intuitionistic Fuzzy Mediator"* can be implemented with the aid of Intuitionistic Fuzzy PostgreSQL (IFPG), which supplements the popular open-source RDBMS PostgreSQL with the functionality of an Intuitionistic fuzzy relational database management system (IFRDBMS). IFPG defines an extension of the SQL (IFSQL) in order to manage with Intuitionistic fuzzy queries and relations. An IFPG user has the ability to define its own predicates, i.e. functions that return Intuitionistic fuzzy Boolean (IFB) value. This is a composite value with two components - degree of truth and degree of falsity. The predicates may be used directly in the WHERE clause even in combination with ordinary Boolean expressions. The IFRDBMS then applies the selection operation using the Intuitionistic fuzzy condition specified in the WHERE clause. Also, IFPG applies the Cartesian product operation over the relations participating in the FROM clause.

IFPG provides the ability to choose the type of value uncertainty in the database by supporting probabilistic and fuzzy database types. According to the database type chosen by the user, IFPG applies the corresponding variants of the selection and the Cartesian product operations, as well as the logical OR and logical AND operations. One can find more for IFPG or download it at http://pgfoundry.org/projects/ifpg.

References

[1]D., Barbara et al . A probabilistic relational data model. In *Proceedings of the International Conference on Extending Database Technology,* pp 60–74, 1990

[2] D., Barbara et al. The management of probabilistic data. *IEEE Trans. Knowl. Data Eng.* Vol.*4*, No.5, pp 487–502,1992

[3] R., Cavallo, M. Pittarelli . The theory of probabilistic databases. In *Proceedings of the International Conference on Very Large Data Bases*, IEEE Computer Society Press, pp 71–81, 1987

[4] Choenni, S., et al (2004). Extending the Relational Model with Uncertainty and Ignorance.. TR-CTIT-04-29, ISSN 1381-3625, pp. 1-16, University of Twente, The Netherlands, 2004

[5] A.L.P. Chen, F.S.C. Tseng, Evaluating aggregate operations over imprecise data, IEEE Transactions on Knowledge and Data Engineering 8) 273-284.1996

[6] L DeMichiel, Resolving database incompatibility: an approach to performing relational operations over mismatched domains,EEE Transactions on Knowledge and Data Engineering 4 (1989) 485±493.

[7 R.R. Yager, K. Engemann, D. Filev, On the concept of immediate probabilities, International Journal of Intelligent Systems10 (1995) 373±397.

[8] E., Codd. Extending the Data Base Relational Model to Capture More Meaning. *ACM* Trans. Database Systems, Vol.4, No.4, 397-434, 1979

[9] B.S., Goldstein. Constraints on Null Values in Relational Databases. *Proc. 7th Int. Conf. on VLDB*, IEEE Press, pp. 101-110, 1981

[10] J., Biskup. A Foundation of Codd's Relation Maybe-Operations. *XP2 Workshop on Relational Database Theory,* 1981

[11] K., Atanassov. *Intuitionistic Fuzzy Sets*, Springer-Verlag, Heidelberg

12] C., Zaniolo Database relations with null values. *J. Comput. Syst. Sci.* Vol. *28*, pp 142–166,1984

[13] Calvanese et al. Answering queries using views over description logics knowledge bases. ACM PODS, pp.386–391, 2000.

[14] J., Ullman. Principles of Database and Knowledge-base Systems. Computer Science Press, New York, 1988.

[15] L., Chen. Describing and Utilizing Constraints to Answer Queries in Data-Integration Systems. IJCAI 2003 Workshop on Information Integration on the Web, 2003

[16] Atanassov, K. Generalized Nets. World Scientific, Singapore, New Jersey, London, 1991.

[17] Atanassov, K. Generalized index matrices. - Compt. Rend. de l'Academie Bulgare des Sciences, Vol.40, 1987, No 11, 15-18.

[18] Radeva, V., M. Krawczak, E. Choy. Review and bibliography on generalized nets theory and applications. Advanced Studies in Contemporary Mathematics, Vol. 4, No. 2, 173-199.