Object-Relational Implementation of Evidential Databases

Fatma Ezzahra Bousnina
LARODEC/ISG-University of Tunis
fatmaezzahra.bousnina@gmail.com

Sayda Elmi
LARODEC/ISG-University of Tunis
saida.elmi@ensma.fr

Mohamed Anis Bach Tobji
LARODEC/ESEN-University of Manouba
anis.bach@isg.rnu.tn

Mouna Chebbah
LARODEC/FSEGJ-University of Jandouba
chabbeh.mouna@gmail.com

Allel HadjAli
LIAS/ENSMA-University of Poitiers
allel.hadjali@ensma.fr

Boutheina Ben Yaghlane
LARODEC/IHEC-University of Carthage
boutheina.yaghlane@ihec.rnu.tn

***Abstract*—Due to the exploding number of information stored and shared over Internet, and the introduction of new technologies to capture and transit data, managing imperfect data is an important issue in many applications. An important tool for reasoning with imperfect data is the evidence theory, which is a generalization of the Bayesian inference. We call databases whose data imperfection are processed thanks to the evidence theory, the *evidential databases*. In this paper, we design the evidential database meta-model using an Oriented-Object modelling language (UML) and we implement it using an Object-Relational database. Although the implementation is not native, it showed an acceptable scalability.**

Keywords—Evidential Databases; Dempster-Shafer Theory; Object-Relational database; SQL3.

I. Introduction

Computer Science created lots of opportunities but also lots of challenges when it comes to the treatment of information efficiently in various domains like weather forecasting [9], economy [10] and medicine [16], [17].

For example, the e-health is a field that appears in the intersection of medical care and informatics. One of the challenges of this emergent domain is the treatment of massive medical data and also improvement of diagnosis quality to make at the end a better decision. E-health is one of the areas that includes imperfect information. To deal with uncertain values of database attributes, several models were proposed. The most studied and known are: probabilistic databases [1], [5], [6], possibilistic databases [3], [4] and evidential databases [2], [13], [14]. The advantage of the evidential databases is twofold: (i) it allows modelling both uncertainty and imprecision due to the lack of information; and (ii) it represents a generalization of the probabilistic model.

Table I stores information about patients' *Diseases* and *Symptoms*. It has three attributes *ID*, *Disease* and *Symptom*. Symptoms are vague in paediatrics domain that is why values in table I are modelled via the theory of belief functions. The latter represents clearly several types of imperfection such as imprecision, uncertainty and inconsistency. In this article, we present an Object-Relational implementation of an evidential database.

We introduce a meta-model of the evidential database using UML. In this meta-model, we take into account the complexity of evidential database attributes and operations applied on these data, basically the belief and plausibility functions. Then, we implement the obtained conceptual model, i.e., the class diagram, using the Object-Relational paradigm rather than the relational one. This choice is argued by: the first is that attributes of evidential databases are complex and multivalued, the second is to respect the first normal form constraint. Finally, we evaluate our implementation over several evidential databases, where we varied many parameters, basically the size of data, number and size of focal elements and the imperfection rate in the database.

The rest of the paper is organized as follows: Section 2 is devoted to present the background material. We present basic concepts of theory of evidence and also basic notions related to evidential database. In section 3, we present the Object-Relational design and implementation of evidential databases. Finally, in section 4, conclusions and future works are drawn.

TABLE I. A Medical Evidential Table

ID	Disease	Symptom	CL
1	$Diabetes$	$Fatigue$ 0.4 $\{Fatigue, Nausea\}$ 0.6	[0.5 ; 1]
2	$Diabetes$ 0.1 $Stroke$ 0.9	$Vertigo$	[0.4 ; 0.8]
3	$Anemia$ 0.3 $\{Diabetes, Stroke\}$ 0.7	$Fainting$	[1 ; 1]

II. Background Material

A. Evidence Theory

Evidence theory, known as the Dempster-Shafer theory or the belief functions theory, was introduced by Dempster [7], [8] and mathematically formalized by Shafer [15]. The theory of evidence represents plainly the imprecision, the inconsistency and the uncertainty of information. Some background material of this theory are presented in this section:

A *frame of discernment* $\Theta = \{\theta_1, \theta_2, ..., \theta_n\}$, also called *universe of discourse* is a finite, non empty and exhaustive set of n elementary and mutually exclusive hypotheses for a given problem.

Example 1: The frame of discernment Θ_{DE} contains the set of hypotheses relative to the attribute $Disease$ and Θ_{SY} is the frame of discernment of the attribute $Symptom$:

Θ_{DE} = {Anemia, Diabetes, Stroke}

Θ_{SY} = {Fatigue, Nausea, Vertigo, Fainting}

The *power set* $2^\theta = \{\varnothing, \theta_1, \theta_2, ..., \theta_n, \{\theta_1, \theta_2\}, .., \{\theta_1, \theta_2, ..., \theta_n\}\}$ includes all subsets of Θ.

A *mass function*, m^Θ, is a mapping from 2^Θ to the interval $[0, 1]$. The mass m^Θ of an hypothesis A is noted $m^{\Theta(A)}$ and it represents our belief on the truth of A. The amount $m^{\Theta(A)}$ is called *basic belief assignment* (*bba*). It is defined such that:

$$\sum_{A \subseteq \Theta} m^\Theta(A) = 1 \tag{1}$$

If $m^\Theta(A) > 0$, A is called *focal element*. The set of all focal elements is denoted F and the couple $\{F, m^\Theta\}$ is called *body of evidence*.

Example 2: The value of the attribute $Symptom$ in the first tuple is given as follows:

$m^{\Theta_{FS}}(Fatigue)$ = 0.4

$m^{\Theta_{SY}}(\{Fatigue, Nausea\})$ = 0.6

The *belief function* denoted bel represents the degree of faith committed exactly to hypothesis A, such that:

$$bel(A) = \sum_{B, A \subseteq \Theta : B \subseteq A} m^\Theta(B) \tag{2}$$

The *plausibility function* denoted *pl* is the sum of masses relative to subsets B and not contradictory with subsets A, such that:

$$pl(A) = \sum_{B, A \subseteq \Theta : A \cap B \neq \varnothing} m^\Theta(B) \tag{3}$$

Example 3: Suppose the mass function given in the previous example 2, we compute the belief and the plausibility functions.

$bel(Fatigue)$ = 0.4 ; $pl(Fatigue)$ = 1

$bel(Nausea)$ = 0 ; $pl(Nausea)$ = 0.6

B. Extended Belief and Plausibility functions

In probability theory, probabilities of comparisons of two independent probability distributions can easily be computed. In standard Dempster-Shafer theory the definitions of the belief and plausibility functions can not handle comparisons like (=, ≠, <, >,≤, ≥). That is why the definition of belief *bel* and plausibility *pl* functions were extended in [2], [12], [13] to handle the comparison between two independent basic belief assignments (*bbas*).

Defintion 1: Equality: Let x and y be two random independent variables and their mass functions, m^{Θ_x}, m^{Θ_y}: $2^\Theta \longrightarrow [0, 1]$, respectively. $A, B \subseteq \Theta$ [2], [12], [13].

$$bel(x = y) = \sum_{|A|=1} m^{\Theta_x}(A) * m^{\Theta_y}(A) \tag{4}$$

$$pl(x = y) = \sum_{A \cap B \neq \varnothing} m^{\Theta_x}(A) * m^{\Theta_y}(B) \tag{5}$$

Example 4: Suppose we have information about $Disease$ of patient 2 and $Disease$ of patient 3 as shown in table I.

We want to compute the *bel* and the *pl* values for the proposition that they have the *same* disease.

$Disease_2$ = {Diabetes, 0.1}+ {Stroke, 0.9}

$Disease_3$ = {Anemia, 0.3}+ { {Diabetes,Stroke }, 0.7}

Then, according to definition 1:

bel(x=y)= 0

pl(x=y)= 0.1*0.7 + 0.9*0.7 = 0.7

Defintion 2: Inequality: Let x and y be two random independent variables and their mass functions, m^{Θ_x}, m^{Θ_y}: $2^\Theta \longrightarrow [0, 1]$, respectively. $A, B \subseteq \Theta$ [2], [12], [13].

$$bel(x \neq y) = \sum_{A \cap B = \varnothing} m^{\Theta_x}(A) * m^{\Theta_y}(B) \tag{6}$$

$$pl(x \neq y) = \sum_{A \subseteq \Theta} m^{\Theta_x}(A) * \sum_{B \subseteq \Theta \ and \ [A = B \longrightarrow |A| > 1]} m^{\Theta_y}(B) \tag{7}$$

Example 5: We carry on with the same example of table I. We want to compute the *bel* and the *pl* of patient 2 and patient 3 for the proposition that they have *different* diseases.

According to definition 2:

bel(x≠y)= 0.1*0.3 + 0.9*0.3 = 0.3

pl(x≠y)= 0.1*0.3 + 0.9*0.3 + 0.1*0.7 +0.9*0.7= 1

Defintion 3: Let x and y be two random independent variables and their mass functions, m^{Θ_x}, m^{Θ_y}: $2^\Theta \longrightarrow [0, 1]$, respectively. $A, B \subseteq \Theta$ [2], [12], [13].

$$bel(x < y) = \sum_{A \subseteq \Theta} m^{\Theta_x}(A) * \sum_{B \subseteq \Theta \wedge A <^{\forall} B} m^{\Theta_y}(B) \tag{8}$$

$$pl(x < y) = \sum_{A \subseteq \Theta} m^{\Theta_x}(A) * \sum_{B \subseteq \Theta \wedge A <^{\exists} B} m^{\Theta_y}(B) \tag{9}$$

Defintion 4: Let x and y be two random independent variables and their mass functions, m^{Θ_x}, m^{Θ_y}: $2^{\Theta} \longrightarrow [0,1]$, respectively. $A, B \subseteq \Theta$ [2], [12], [13].

$$bel(x \leq y) = \sum_{A \subseteq \Theta} m^{\Theta_x}(A) * \sum_{B \subseteq \Theta \wedge A \leq^{\forall} B} m^{\Theta_y}(B) \tag{10}$$

$$pl(x \leq y) = \sum_{A \subseteq \Theta} m^{\Theta_x}(A) * \sum_{B \subseteq \Theta \wedge A \leq^{\exists} B} m^{\Theta_y}(B) \tag{11}$$

Example 6: We want to compare values of attribute $Disease$ for patient 2 and patient 3 of table I. We compute in this example the bel and the pl for the proposition (x<y) and (x≤y) where x is the $disase$ of patient 2 and y is the $disase$ of patient 3. We have the following information: $Stroke$ is more serious than $Diabetes$ which is it self more serious than $Anemia$: $Anemia < Diabetes < Stroke$. Values of attribute $Disase$ for patients 2 and 3 are given as follows:

$Disease_2$ = {Diabetes, 0.1}+ {Stroke, 0.9}

$Disease_3$ = {Anemia, 0.3}+ { {Diabetes,Stroke }, 0.7}

According to definitions 3 and 4.

bel(x<y)= 0

pl(x<y)= 0

bel(x≤y)= 0.1*0.7= 0.07

pl(x≤y)= 0.1*0.7 + 0.9*0.7= 0.7

Defintion 5: Let x and y be two random independent variables and their mass functions, m^{Θ_x}, m^{Θ_y}: $2^{\Theta} \longrightarrow [0,1]$, respectively. $A, B \subseteq \Theta$ [2], [12], [13].

$$bel(x > y) = \sum_{A \subseteq \Theta} m^{\Theta_x}(A) * \sum_{B \subseteq \Theta \wedge A >^{\forall} B} m^{\Theta_y}(B) \tag{12}$$

$$pl(x > y) = \sum_{A \subseteq \Theta} m^{\Theta_x}(A) * \sum_{B \subseteq \Theta \wedge A >^{\exists} B} m^{\Theta_y}(B) \tag{13}$$

Defintion 6: Let x and y be two random independent variables and their mass functions, m^{Θ_x}, m^{Θ_y}: $2^{\Theta} \longrightarrow [0,1]$, respectively. $A, B \subseteq \Theta$ [2], [12], [13].

$$bel(x \geq y) = \sum_{A \subseteq \Theta} m^{\Theta_x}(A) * \sum_{B \subseteq \Theta \wedge A \geq^{\forall} B} m^{\Theta_y}(B) \tag{14}$$

$$pl(x \geq y) = \sum_{A \subseteq \Theta} m^{\Theta_x}(A) * \sum_{B \subseteq \Theta \wedge A \geq^{\exists} B} m^{\Theta_y}(B) \tag{15}$$

Example 7: We want to calculate the bel and the pl of patient 2 and patient 3 of table I for the proposition (x>y) and (x≥y) for the attribute $Disease$ where $Stroke > Diabetes > Anemia$. According to definitions 3 and 4.

bel(x $>$ y)= 0.1*0.3+0.9*0.3 = 0.3

pl(x $>$ y)= 0.1*0.3+0.9*0.3+0.9*0.7=0.93

bel(x $\geq$ y)= 0.1*0.3+09*0.3=0.3

pl(x $\geq$ y)= 0.1*0.3+09*0.3+0.9*0.7=0.93

C. Evidential databases

An *Evidential database* (*EDB*), is also called *Dempster-Shafer database*. It was introduced by Lee [2], [13]. An EDB stores both perfect and imperfect data which are modelled with evidence theory.

An *EDB* has N tuples and D attributes. An *evidential value*, noted V_{ta} is the value of an attribute a for a tuple t that represents a *bba*.

$$V_{ta} : 2^{\Theta_a} \rightarrow [0,1] \tag{16}$$

$$\text{with } m_{ta}(\varnothing) = 0 \text{ and } \sum_{A \subseteq \Theta_a} m_t(A) = 1 \tag{17}$$

The set of focal elements relative to the bba V_{ta} is noted F_{ta} such that:

$$F_{ta} = \{x \subseteq \Theta / m_{ta}(x) > 0\}$$

The *Confidence Level, CL* represents the degree of belief of the expert about each tuple t in the evidential database EDB. A confidence level is a pair of belief and plausibility ; $\{CL = [b;p]; CL \in [0;1]; b \leq p\}$.

An evidential database EDB stores multiple types of data:

- When the focal element is singleton and its mass function is 1 then the bba is called *certain*. In this case the information is called *perfect*.
- When focal elements are singletons, the bba is called *bayesian* and the information is *probabilistic*.
- When focal elements are nested, the information is *possibilistic* and the bba is called *possibilistic*.
- When none of the previous cases is present the bba is named *evidential* and the information is called *evidential information*.

Example 8: Table I presents an example of the various types of information presented previously.

- The bba of attribute $Disease$ in the first tuple is certain.
- The bba of attribute $Disease$ of the second tuple is bayesian.
- The bba of attribute $Symptom$ of the first tuple is nested.

- The bba of attribute $Symptom$ of the third tuple is evidential.

The evidential database models [12], [13], [2] were proposed to handle perfect and imperfect databases, they can represent uncertain, imprecise and inconsistent data.

In the classical relational databases model, data are stored to be later queried using the relational operators: select, project, join operators. Similarly, evidential data are stored in evidential databases to be further interrogated. That is why relational operators were redefined to match the rules of an evidential database model. We cite for example, the extended select, project operators [13] and also the extended Cartesian product, the intersect and the join operators redefined in [2], [12].

1) Extended Evidential Selection: The extended selection consists on extracting from an evidential database EDB, tuples whose values satisfy the condition C of a given query Q. It returns a new relation according to a threshold of belief and plausibility.

A condition C can be *atomic* or *compound*. An atomic condition handles simple comparisons using the comparison operators as ($=$, $\neq$,$<$, $>$, $\leq$, $\geq$). A compound condition is constructed from atomic conditions using logical connectives as (conjunction, disjunction, negation).

The resulting confidence level CL represents the degree of belief for every tuple satisfying the condition C. It is computed as follows:

$$CL = [b * bel; p * pl] = [b_c; p_c] \tag{18}$$

With bel and pl are respectively the belief and the plausibility of a tuple t; b and p are the belief and the plausibility of the attribute's value satisfying C of the resulting tuple t. The calculated confidence level is the interval $[b_c; p_c]$.

Example 9: We process a selection query with an atomic condition over the medical table I.

Q_1: SELECT * FROM EDB WHERE ($Disease$ = $\{Diabetes\}$)

The result of query Q_1 is shown in table II. We calculate for each resulting tuple its new confidence level CL using equation 18.

TABLE II. RESULT OF Q_1

ID	Disease	Symptom	CL
1	$Diabetes$	$Fatigue$ 0.4 $\{Fatigue, Nausea\}$ 0.6	[0.5;1]
2	$Diabetes$ 0.1 $Stroke$ 0.9	$Vertigo$	[0.04;0.08]
3	$Anemia$ 0.3 $\{Diabetes, Stroke\}$ 0.7	$Fainting$	[0;0.7]

$Q_1.CL(t_1)$ = [1*0.5 ; 1*1] = [0.5 ; 1]

$Q_1.CL(t_2)$ = [0.1*0.4 ; 0.1*0.8] = [0.04 ; 0.08]

$Q_1.CL(t_3)$ = [0*1 ; 0.7*1] = [0 ; 0.7]

Now, we process a selection query with a compound condition over the medical table I.

Q_2: SELECT * FROM EDB WHERE ($Symptoms$ = $\{Fatigue\}$) $\wedge$ ($b_c(\{Fatigue\}) \geq$ 0.2)

The result of query Q_2 is presented in table III. We calculate for each resulting tuple its new confidence level CL using equation 18 according to the compound condition.

TABLE III. RESULT OF Q_2

ID	Disease	Symptom	CL
1	$Diabetes$	$Fatigue$ 0.4 $\{Fatigue, Nausea\}$ 0.6	[0.2 ; 1]

$Q_2.CL(t_1)$ = [0.4*0.5 ; 1*1] = [0.2 ; 1]

$Q_2.CL(t_2)$ = [0*0.4 ; 0*0.8] = [0 ; 0]

$Q_2.CL(t_2)$ = [0*1 ; 0*1] = [0 ; 0]

Only tuple t_1 satsifies the condition $C = b_c\{Fatigue\}) \geq$ 0.2

2) Extended Evidential Projection: The extended projection consists on extracting from an evidential database EDB, one attribute or more from attributes of EDB. It returns a relation that includes evidential attributes according to condition C of a given query Q. The relation satisfies a threshold of belief and plausibility. The condition C can be *atomic* or *compound*.

Example 10: We process a simple projection query over the medical table I.

Q_3: SELECT $Disease$ FROM EDB

The result of query Q_3 is shown in table IV.

TABLE IV. RESULT OF Q_3

Disease
$Diabetes$
$Diabetes$ 0.1 $Stroke$ 0.9
$Anemia$ 0.3 $\{Diabetes, Stroke\}$ 0.7

Now, we process another projection query with a compound condition.

Q_4: SELECT ID, $Disease$ FROM EDB WHERE ($Symptom$ like _a%) $\wedge$ (CL[bel ; pl] $>$ [0.6 ; 1])

The result of query Q_4 is shown in table V.

We calculate for each resulting tuple its new confidence level CL using equation 18. We compute CLs for each $Symptom$ of the medical table I if its second letter is a.

TABLE V. RESULT OF Q_4

ID	Disease
3	$Anemia$ 0.3 $\{Diabetes, Stroke\}$ 0.7

Resulting tuples should verify the condition C = (CL[bel ; pl] >[0.6 ; 1]).

For example, the symptom in first tuple $Fatigue$, obeys to the first condition where the second letter is a but does not satisfy to the second condition over the confidence level.

III. DESIGN AND IMPLEMENTATION OF AN EVIDENTIAL DATABASE

A. The meta-model of evidential databases

In a database, tables store data in a matrix format; attributes in columns and objects in lines.

In evidential databases, we qualify a table by evidential. Its attributes are based on the structure of BBAs. An attribute has a name and contains one *basic belief assignment* (BBA) for each object. A BBA is composed of one or more *focal elements*. A focal element has a mass and contains *hypotheses*. Each hypothesis has a content.

The belief and plausibility functions compute belief and plausibility of a set of hypotheses in a BBA. They are defined as methods at the BBA structure.

Each evidential table stores N objects. An object is identified by its ID and its *confidence level* (CL). The latter is associated with an interval of belief and plausibility. It quantifies the confidence level degree of the tuple's source.

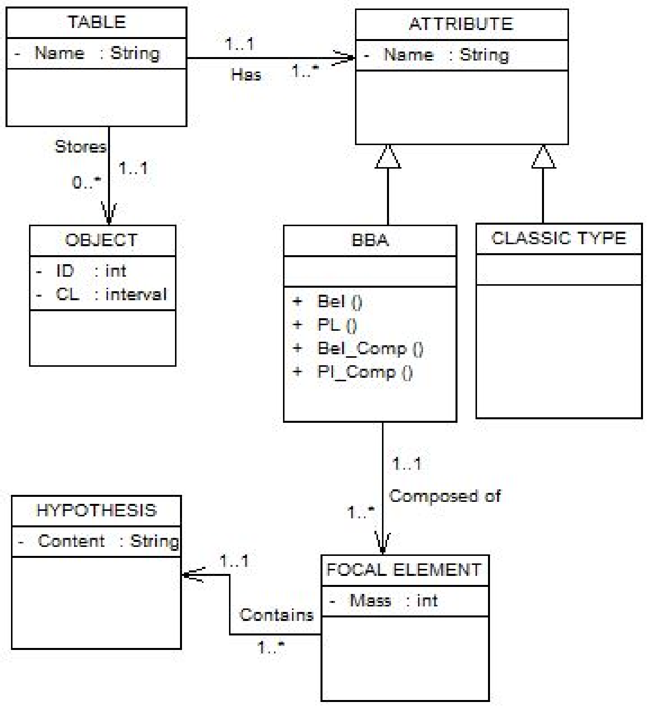
Figure 1 is the class diagram of the evidential database meta-model.

B. Object-Relational Implementation of Evidential Databases

At the best of our knowledge, there is no efficient implementation of evidential databases that offer a scalable and flexible solution to manage data as BBAs. In this paper we rely on a commercial Object-Relational Database Management System (ORDBMAS), Oracle 10g. It's main asset is the Oriented-Object feature that facilitate the implementation of the complex structure of an evidential database, as designed in section III-A. We define the BBA type which is basically a collection of focal elements, whose type contains two attributes, a collection of hypotheses, and its mass value. As explained above, this modelization is impossible in the relational framework, because of the first normal form constraint.

In addition to the previous advantage, a commercial ORDBMS offers an interesting I/O cost optimization. Oracle for example, uses a System Global Area (SGA) to keep in cache reused data (the database buffer cache) and reused queries (the shared pool). The database buffer cache contains previous extracted data. In case of querying some of (or these) data, the database server avoid disk access and returns the result directly from physical memory. Also, queries executions plans whose computation is costly, are stored in the shared pool in the SGA. In case of executing a query existing in

Fig. 1. Meta-Model of Evidential Databases



the pool, the system avoids syntactic analysis and execution plan computation which save important CPU time. Another feature of DBMSs; the indexes. These structures accelerate information extraction especially if we use *Nested tables* vs. *Varying Arrays* to store collections. Indeed, a *varying arrays* attribute (in Oracle, the type is named varray) is physically stored in the same segment of the table. On the other hand, a *nested table* attribute is stored in a separate segment. Thus, data of that segment are indexable. Assume the attribute symptom in table I. Symptom is a collection of focal elements. If we use the nested table type, we will have the possibility to create an index on that attribute. It implies that selection on objects with a symptom criterion will be more efficient.

To use the Object-Relational model in Oracle, we use SQL3. We want to create a BBA type, that is mainly a collection of focal elements. Focal element structure contains a mass value, and also a collection of hypotheses (not only one hypothesis, because focal elements could be a set of hypotheses). So first of all, we create the type *hypos*, that is a collection of a basic data type:

```
CREATE TYPE FOCALELEMENT AS OBJECT
(content HYPOS, mass NUMBER, MEMBER
FUNCTION Includes(search FOCALELEMENT)
RETURN NUMBER, MEMBER FUNCTION Intersect
(search FOCALELEMENT) RETURN NUMBER);
```

The BBA type is defined as an object with one attribute; *content* and two methods; *BEL* and *PL*. *Content* is defined as a collection of focal elements. *Bel* and *Pl* are methods that compute respectively belief and plausibility of comparison with another BBA.

```
CREATE TYPE BBA AS OBJECT (content
FOCALELEMENT, MEMBER FUNCTION Bel_Comp
(search BBA, op CHAR(2)) RETURN NUMBER,
MEMBER FUNCTION Pl_Comp (search BBA, op
CHAR(2) RETURN NUMBER, MEMBER FUNCTION
Bel(FOCALELEMENT F) RETURN NUMBER,
MEMBER FUNCTION Pl(FOCALELEMENT F) RETURN
NUMBER);
```

To integrate tuple uncertainty level, we define the type CL that is an interval of numbers:

```
CREATE TYPE CL AS OBJECT (Bel NUMBER, Pl
NUMBER);
```

To create an evidential table, like in example I, we first define the type on which it is based, i.e., the type *diagnosis* as below:

```
CREATE TYPE diagnosis AS OBJECT (id
NUMBER, disease BBA, diagnosis BBA, clv
CL);
```

And then we create the object table, called *diagnoses*, based on the type *diagnosis*:

```
CREATE TABLE diagnoses OF diagnosis
(PRIMARY KEY(id),
NESTED TABLE disease STORE AS tab_diseases
(NESTED TABLE content STORE AS hypos),
NESTED TABLE symptom STORE AS tab_symptoms
(NESTED TABLE content STORE AS hypos);
```

For short, note that we did not present the bodies of types BBA and $FOCAL\ ELEMENT$ because of the paper's length constraint.

The methods Bel and Pl are very important. We may select objects whose attribute's values (BBAs) are compared as follows:

- symptom="Fatigue". The search criterion is a single value. "Fatigue" can be seen as a BBA with a single focal element, that has a single hypthesis with mass equal to one. It is a certain BBA.
- symptom={"Fatigue","Nausea"}. This latter is a BBA with one focal element that is {"Fatigue","Nausea"} with mass equal to one.
- e.symptom=d.symptom, consists in comparing two BBAs. In our example, we may need to compare symptoms of patient with id=1 and patient with id=2. Thus, we will compare two imperfect symptoms stored as BBAs.

In all these cases, the Bel (respectively Pl) function evaluates the comparison of a BBA attribute, with another value, be it perfect or imperfect. For example, the query below select patients whose diseases are equal to "Anemia" with a belief grater that 0.2. The searched $symptom$ is processed by our Bel function as a BBA with a single focal element whose mass is equal to one.

```
SELECT d.* FROM disease d
WHERE d.Bel(BBA(FOCALELEMENT('Anemia',1)),
'=') > 0.2;
```

Thanks to this implementation, we can select objects that satisfy a criterion over a belief or plausibility threshold. In addition, comparison doesn't concern only equality, but also inequality, through the operators $\{<,>,\leq,\geq\}$ usable in second parameter of the BBA methods Bel and Pl. Thus, we can select, for example, patients that have diagnoses more serious than the diagnosis *Diabetes*, or compare between gravity of two patients' diagnoses.

IV. Experiments and Discussion

In this section, we evaluate the proposed object relational implementation from a performance point of view. We used a windows 8 desktop with a 2.67 GHZ CPU and 8GB RAM. We used SQL3 and PL/SQL for implementation on Oracle 10g server. The size of the System Global Area (SGA) in Oracle server is set to 1GB.

A. Data sets

We used synthetic data sets following these parameters (1) n the size of the database, (2) $\%IR$ the imperfection rate of data, i.e., number of imperfect objects over N, (3) nfe the maximum number of focal elements per BBA, (4) sfe the maximum size of each focal element (5) a number of attributes and (6) d size of attribute domain.

To generate a synthetic evidential database, the used algorithm uses a main procedure that generates a synthetic *BBA*. This procedure operates as follows: it computes randomly a fixed number of focal elements in the interval $[1, nfe]$. Then for each focal element, it generates randomly a number in the interval $[1, sfe]$; thats the size of current focal element. Each hypothesis in the focal element is randomly generated in the interval $[1, d]$, d being the cardinality of our attribute domain. Masses of focal elements are generated in the interval $[0, 1]$. We used the random function of JAVA which is based on the uniform law.

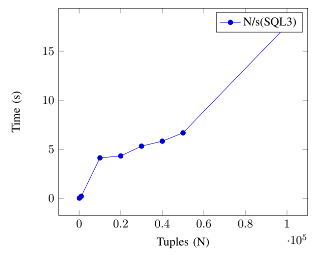
We managed several constraints like uniqueness of hypotheses into one focal element, uniqueness of focal elements in a BBA, and normalization of a BBA (sum of focal elements masses must be equal to one).

Then, we generate for each tuple one BBA per attribute. We repeat this operation for $\%PI$ percent of N. Remaining tuples are perfect and contain in every attribute one BBA with one singleton focal element whose mass is equal to one.

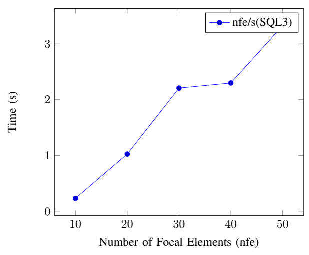
B. Evaluation and discussion

The experiments done on the Object-Relational implementation showed interesting results from a performance point of a view although it has some limits when some parameters of the database reach some thresholds.

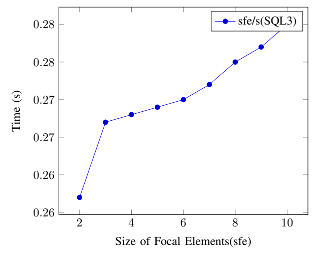
Default values used in experiments are fixed to $N = 1000$, $nfe = 4$, $sfe = 2$, $d = 10$ and $\%IR = 70$. In the first experiments, we varied the parameter N from 1000 tuples to 500000. The system crashed when we set $N =$ to 500000. For $N = 100000$ the system answers our test query in approximately 17 seconds. Experiments produced very acceptable time of execution until the limit of 60000 tuples (about 6 seconds at the worst). Figure 2(a) shows the experiment results.



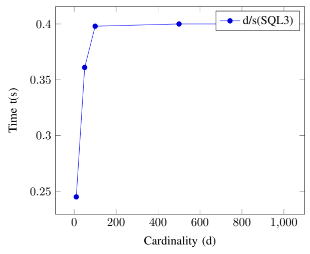
(a) Number of tuples variation



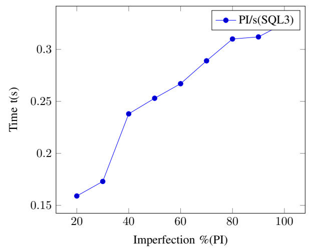
(b) Number of focal elements variation



(c) Size of focal elements variation



(d) Cardinality variation



(e) Percentage of imperfection variation

Then, we varied the number of focal elements. Until $nfe = 50$, the system answers with acceptable execution time, without exceeding four seconds. We judge the value of 50 as appropriate, because an expert (a physician in our example) does not give a such great number of believable hypotheses (in our case diagnoses). Results of this experiment are shown in figure 2(b).

For focal elements' sizes, we varied the parameter sfe from 2 to 10. Figure 2(c) shows the results. Execution time did not exceed 0.3 second. This acceptable performance is due to *Bel_Comp* (respectively *Pl_Comp*) PL/SQL implementation. First of all, PL/SQL is integrated in Oracle (such that every transactional programming language), which reduces I/O costs. Second, scanning focal elements for belief/plausibility computations' amounts to scanning Oracle nested tables. Searching these structures is optimized by Oracle which offers for them sequential and direct access.

Each attribute is characterized by its domain cardinality. It refers to the frame of discernment size. We varied this parameter from 10 to 1000, the results are presented in figure 2(d). The performance is again very acceptable although we reached a high size of cardinality (1000). Note that complexity of a generated *BBA* also depends on number and size of focal elements that are controlled by parameters nfe and sfe. If we do not control these parameters, we can reach 21000 focal elements, with sizes that could reach 1000 This situation is not realistic, because expressing a so huge number of focal elements in one *BBA* is impossible.

To mimic real imperfect databases, our solution should process objects that are either perfect or imperfect. To show the impact of imperfect objects on our solution performance, we varied the rate of imperfect objects in the database from 20% to 100%. It's logical to note performance decrease in case of high values of the imperfection rate. Processing evidential BBAs vs. certain BBAs involves more belief/plausibility computations. But in general, the performance was acceptable (it didn't exceed 0.5 second) even when the rate reached 100%.

An important feature of commercial databases consists in using memory caches to speed-up queries answering. In general, two memory caches are used in most of commercial solutions. The database buffer cache, for keeping previous extracted data, and the shared pool for keeping syntactic and execution information of previous queries. To evaluate the contribution of caching data and queries in our solution, we compared first execution of the test query, with next executions. Naturally, the difference is clear and expected. Table VI shows the result of this experimentation for different values of the parameter n (size of the database).

TABLE VI. CONTRIBUTION OF CACHES TO QUERIES RE-EXECUTION

Database size	First execution time (s)	Next executions' times (s)
1000	0.2	0.03
5000	0.8	0.04
10 000	4.1	0.06
50 000	5.8	0.12
70 000	6.4	0.16
100 000	17	1.2

V. CONCLUSION

In this paper, we implemented the evidential database model in Object-Relational context. We benefit from flexibility of the Oriented-Object model which offers possibility to use complex structures and collections as attributes, in opposite to the classic relational model. We added a new type called *BBA* to allow storage and extraction of imperfect attributes' values as evidence distribution, named *Basic Belief Assignment*. The new type offers four methods to compute Belief and Plausibility of hypotheses in one BBA, and also to compare current BBA with others and quantify its belief and plausibility.

The Object-Relational context also offers a standardized and powerful query language; SQL. In the introduced solution, SQL3 is used to create and then query the evidential database. Users can easily use the language, yet known for any database user, and programmers can easily develop applications based on evidential data without manipulating complicated packages. In addition, a commercial Object-Relational database optimizes I/O and CPU costs, essentially by using cache memories. We showed that information extraction using these techniques provides a clear performance advantage. Indexes can also be used to accelerate access to data.

Nevertheless, our solution is non-native. Implementation of a native solution should give more possibilities to information extraction, like joining data as presented in [13] and processing the problematic union and projection operators as presented in [2]. A native solution should be more efficient, especially if we use specific indexes, adapted to the evidential nature of data [11]. A complete framework for implementing an evidential database server is a promising perspective, especially that the same framework can also manage probabilistic and possibilistic data because of the generality of the Dempster-Shafer theory.

REFERENCES

[1] Charu C. Aggarwal. *Managing and Mining Uncertain Data*, volume 35 of *Advances in Database Systems*. Springer, 2009.

[2] David A Bell, J. W. Guan, and Suk Kyoon Lee. Generalized union and project operations for pooling uncertain and imprecise information. *Data & Knowledge Engineering*, 18:89–117, 1996.

[3] Patrick Bosc and Olivier Pivert. About projection-selection-join queries addressed to possibilistic relational databases. *IEEE Transactions on Fuzzy Systems*, 13(1):124–139, 2005.

[4] Patrick Bosc and Olivier Pivert. Modeling and querying uncertain relational databases: A survey of approaches based on the possible worlds semantics. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 18(5):565–603, 2010.

[5] Nilesh N. Dalvi and Dan Suciu. Efficient query evaluation on probabilistic databases. *VLDB J.*, 16(4):523–544, 2007.

[6] Nilesh N. Dalvi and Dan Suciu. Management of probabilistic data: foundations and challenges. In *PODS*, pages 1–12, 2007.

[7] Arthur P Dempster. Upper and lower probabilities induced by a multiple valued mapping. *The Annals of Mathematical Statistics*, 38(2):325–339, 1967.

[8] Arthur P Dempster. A generalization of bayesian inference. *Journal of the Royal Statistical Society, Series B*, 30:205–247, 1968.

[9] Ratul Dey and Sanjay Chakraborty. Convex-hull & dbscan clustering to predict future weather. In *Computing and Communication (IEMCON), 2015 International Conference and Workshop on*, pages 1–8. IEEE, 2015.

[10] Stefan Dietze, Davide Taibi, Hong Qing Yu, and Nikolas Dovrolis. A linked dataset of medical educational resources. *British Journal of Educational Technology*, 2015.

[11] Anouar Jammali, Mohamed Anis Bach Tobji, Arnaud Martin, and Boutheina Ben Yaghlane. Indexing evidential data. In *Complex Systems (WCCS), 2014 Second World Conference on*, pages 196–201, 2014.

[12] Suk Kyoon Lee. An extended relational database model for uncertain and imprecise information. In *Proceedings of the 18th Conference on Very Large Data Bases*, pages 211–220, Canada, 1992.

[13] Suk Kyoon Lee. Imprecise and uncertain information in databases : an evidential approach. In *Proceedings of the 8th International Conference on Data Engineering*, pages 614–621, 1992.

[14] E.P. Lim. An evidential reasonong approach to attribute value conflict resolution in database integration. *IEEE Transactions On Knowledge And Data Engineering*, 8(5):707–723, 1996.

[15] Glenn Shafer. A mathematical theory of evidence. *Princeton University Press*, 1976.

[16] Elizabeth Sillence and Pam Briggs. Trust and engagement in online health a timeline approach. *Handb PsycholCommun Technol*, 33:469–487, 2015.

[17] Robin Williams. Why is it difficult to achieve e-health systems at scale? *Information, Communication & Society*, 19(4):540–550, 2016.