Conflict Resolution in Flexible Environments

Panagiotis Chountas & Ilias Petrounias

Department of Computation, UMIST
PO Box 88, Manchester M60 1QD, UK
E-mail: {chountap, ilias}@sna.co.umist.ac.uk

Abstract. This paper is presenting a way of integrating conflicting temporal information from multiple information providers. It is shown that uncertain information can arise when the provider integration process through a global query mechanism, requires information not directly presented in the community of information providers. This occurs for two reasons mainly: I) Query selection predicates are directing to level concepts that are at a lower level than those do that exist in the instance level of the database. II) An element in the query selection predicate is a member of more than one high level concepts. A way of integrating conflicting temporal information from multiple information providers considering a property-based resolution is presented.

1. Introduction

The increasing need for applications to utilise a community of information providers in their effort to answer an information request, has been a lasting problem of research [1],[2]. In requesting a community of information providers two types of incompatibility may occur namely structural and semantical. The former may be caused by differences in the attributes domain, format, units, and granularity (intentional inconsistencies). The latter can be caused by attribute value conflict, since similar defined attributes, utilise different levels of a conceptual hierarchy (tree, lattice) in defining themselves (extensional inconsistencies).

The focus of this paper is on integrating the independent providers by means of a global conceptual schema that models the information, contained in the entire population of information providers and resolves extensional inconsistencies. 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.

Previous work in the area of heterogeneous databases focused on reconciling among different database designs and eventually among different semantics [2], [3]. However when some members from the population of information providers overlap, that is they provide support for the same information, it is not surprising to receive temporal or factual *conflicting* information. Information providers may supply contradicting fact instances, which are either time-stamped or not. In the case of time-stamped data, different time dimensions may be considered, such as valid and transaction time. Furthermore if different types of certain temporal information are considered as defined in [4], [5] conflicts may arise because information providers provide conflicting descriptions in terms of the valid time dimension about:

The exact duration of a time-stamped fact instance (definite temporal information).The constrained duration of a time-stamped fact instance (indefinite temporal information).The possible known-unknown pair (K, D), where K is the frequency of repetition and D the duration of a periodical or infinite time-stamped fact instance. Existential Inconsistencies prevent the virtual database system from answering uniquely global queries. This paper is proposing a framework that allows the integration of extensional inconsistencies for temporal and conflicting information and gives permission to users to express their preferences, considering the different properties of the information providers that make them particularly attractive in certain circumstances to certain groups of users.

The rest of the paper is organised as follows. Section 2 presents the problems in finding the most authoritative and flexible answer and issues that may come up, through resolution in a flexible manner. Section 3 presents a time model for supporting definite, indefinite, infinite temporal information. Section 4 defines an extended relational environment where temporal and conflicting information is captured. Section 5 presents an extended relational algebra for extraction of flexible answers. Section 6 concludes and points to open research issues.

2. Problems in Estimating Flexible Answers

A property-based resolution simply discusses the information *properties* involved in a conflicting situation, decides on the importance of them and then asks for the best-extracted answer, considering the availability of the information providers. Let us consider the following description about travellers 'Ann' and 'Liz'.

Problem Formulation: Consider the following conflicting queries and whether these could be answered with authority or not:

Q_1: When does Liz visit Brazil? Q_2: When does Ann visit Brazil?

$Q_{3:}$ Which are all the people who visited the Southern Hemisphere?

$Q_{4:}$ Which are all the people who visited the Northern Hemisphere?

The above queries are not easy to be answered because of the following inconsistencies:

If we consider a lattice-structured domain, then Brazil has two parents (Northern Hemisphere and Southern Hemisphere, as shown in Figure 1). Therefore Q_3 and Q_4 are not easy to be answered. We cannot estimate with precision the exact date of arrival and stay in Brazil for both travellers (Liz, Ann). Therefore Q_1 and Q_2 are not easy to be answered either. The above queries are not easy to be answered because of the following inconsistencies: If we consider a lattice-structured domain, then Brazil has two parents (Northern Hemisphere and Southern Hemisphere, as shown in "Figure 1". Therefore Q_3 and Q_4 are not easy to be answered. We cannot estimate with precision the exact date of arrival and stay in Brazil for both travellers (Liz, Ann). Therefore Q_1 and Q_2 are not easy to be answered either. The above inconsistencies are presented in Table I through relation R. An analytical observation of Table I gives rise to the following requirements/issues in terms of data representation:

R	Person	Concept	VT(R)
X_1	Ann	Brazil	[01/03/00,29/05/00]
X_2	Ann	Southern Hemisphere	[01/06/00,29/08/00]
X_3	Ann	Northern Hemisphere	{[01/06/00,29/08/00], [01/09/00,29/11/00]}
X_4	Ann	Brazil	[?,?] duration (90 days somewhere in year 2000)
X_5	Liz	Brazil	[01/03/00,29/05/00]
X_6	Liz	Northern Hemisphere	{[01/06/00,29/08/00], [01/09/00,29/11/00]}
X_7	Liz	Brazil	[?,?] duration (90 days somewhere in year 2000)

Table 1. Relation R representing the schedules of Ann and Liz

A time model and representation that presents temporal information in terms of the following physical measurements duration *D* (e.g. tuples $\{X_1, X_5, X_2\}$) and frequency K of reappearance, if information is periodical, where *D*, *K* may not be known, (e.g. tuples $\{X_4, X_7\}$). In our effort to classify tuples in Table I based on a virtual decision attribute which simply declares the fact that a person has visited Brazil, it can be seen that tuples $\{X_2, X_3, X_6\}$ cannot be classified with an exclusive Boolean Yes or No.

Therefore, there is a need for algebraic operations that will support approximate answers, based on approximation spaces. A similar problem would arise if a user requested through an algebraic operation all the countries that Ann or Liz have visited in the Northern and Southern Hemispheres.

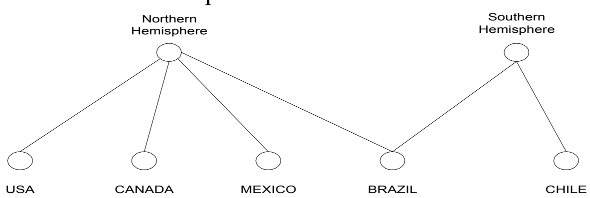

Fig. 1. Lattice Structured Domain

3. The Time Model

In this section the basic elements for a temporal representation are defined in accordance to [4]. The central concepts are a timeline and a time point where the former is comprised of the latter. The term duration is defined as an absolute distance between two time points. However, the term duration may also imply the existence of two bounds an upper bound and a lower bound (indefinite temporal Information). A time interval is defined as a temporal constraint over a linear hierarchy of time units denoted H_r. H_r is a finite collection of distinct time units, with linear order among those units. For instance, H_1=day⊆month⊆year, are all linear hierarchies of time units defined over the Gregorian calendar. A time interval is presented in the form of $[C+K\times X, C^{'}+K\times X]$ where $C^{'}=C+D$, $D\in N^{*}$, thus an interval is described as a set of two linear equations defined in a linear time hierarchy (e.g. H_2 = day⊆month⊆year).

The lower time point t_{lower} is described by the equation $t_{lower} = C+K*X$. The upper point t_{upper} is described by the equation $t_{upper} = C^{'}+K*X$. C is the time point related to an instantaneous event that triggered a fact, K is the repetition factor, $K\in N^{*}$ or the lexical 'every' (infinite-periodical information).X is a random variable, $X\in N$, including zero, corresponding to the first occurrence of a fact instance restricted by a constraint. The Sum-Product $K*X+D$ is defined according to a linear hierarchy. D may be in the range between a lower and upper bound $G_1 \leq D \leq G_2$ where ($G_1 \leq G_2 \wedge D \leq Y$). $Y_i \ldots Y_j$ are general or restricted constraints on the time points t_{lower}, t_{upper}. Constraints are built from arbitrary linear equalities or inequalities (e.g. $t_{lower} = C+7X$ and $0 \leq X \leq 5$). Limiting the random variable X results in specifying the lower and upper bound of a time window. The above interval representation permits the expression of the following types of information:

I) *Definite Temporal Information*: The duration (D) of an event is constant (D = t). All times associated with facts are known precisely in the desired level of granularity. Let t_{lower}, t_{upper} be the lower and upper linear points, of a time interval, that determines when a fact instance is defined, in the real world.

$t_{lower} = C+K*X$ (1), $t_{upper} = C^{'}+K*X$ (2), $C^{'} = C+D$ (3),$(K*X) = 0$ (4)

II) *Indefinite Temporal Information*: is defined when the time associated with a fact has not been fully specified [5]. Therefore the duration of a fact is indeterminate or *bounded*. This may occur for two reasons: either the duration of a fact is bounded or the duration is known and the start and end point of the time interval are not exactly known. This is defined as following:

$C_L \leq C \leq C_R$ (1), $D_L \leq D \leq D_R$ (2)

Let t_{lower}, t_{upper} be the lower and upper linear points, of a time interval, that determines when a fact instance is defined, in the real world.

Adding (2), (1): $C_L + D_L \leq C \leq C_R + D_R \Rightarrow C_L + D_L \leq C^{'} \leq C_R + D_R$ which is the new expression for equation (3) whereas $(C_L + D_L) \leq C^{'} \leq C_R + D_R) \in H_r$. Therefore the time interval (3), that a sample fact instance is defined over, is indeterminate.

III)*Infinite Temporal Information*: is defined when an infinite number of times are associated with a fact [6]. Infinite temporal information includes the following types of information.

a) *Periodic*: A fact instance is repeated over a time hierarchy with the following characteristics: a constant frequency of repetition K, it has an absolute and constant duration D, and X a random variable that denotes the number of reappearances for an event. Therefore the duration of every fact instance constituting a fact type and consequently the duration of a fact type is well known.

b)*Unknown Recurring Information*: Generally is described in the following intervalic form, t = [⊥, ⊥]. The intuition is that the duration D of an event is assumed to be known or constrained and the frequency of reoccurrence ($K=?$) is not known. However by definition it is known that if an event is recurring, then its next reappearance cannot occur before the previous one is ended. Considering Ann's staying to Brazil according to provider A_4, she is staying for a period of ($D=90$ *days*) in Brazil. Using the best case scenario, it can be assumed that *Ann* is visiting Brazil every ($K=90$) days, since nothing else is known about Ann's trip to Brazil. It is also known that $K \geq 1$. Therefore the following conclusion can be made $D\leq K$.

It can be also assumed that the time point C related to an instantaneous event that triggered a fact is not known with precision in the time hierarchy H_r. C it may be constrained by an application, like as $C_L\leq C \leq C_R$ or be left unspecified; constraints are part of an answer to a query. In this sense t_{lower}, t_{upper} are expressed as following:

$t_{lower} = K*X + C$ (1) $\quad D \leq K \wedge K\neq 0\ K\in N^*$ (5)

$t_{upper} = K*X + C'$ (2) $\quad C_L\leq C \leq C_R$ (6)

$C' = C + D$ (3) $\quad D_L\leq D\leq D_R$ (7)

$a_1\leq X\leq a_v$, $a_1=0$ denotes first occurrence of an event (4)

Adding (6), (7) and using (3) it can be deduced that $C_L + D_L\leq C'\leq C_R + D_R$ (8). (5) is defined through (7) when $D_L\leq K\leq D_R$ or $K_L\leq K\leq K_R$ where $D_L= K_L$, $D_R = K_R$ (9). The product $K\times X$ through (9) and (4) is defined as follows

$$K_L* (a_1...a_v)\leq K*X \leq K_R* (a_1...a_v)\ (10)$$

Considering [(8), (6)] and [(8), (10)], t_{lower}, t_{upper} can be redefined, respectively as follows

$$t_{lower}: K_L* (a_1...a_v) + C_L \leq K*X + C \leq K_R* (a_1...a_v) + C_R$$

$$t_{upper}: K_L* (a_1...a_v) + C_L + D_L\leq K*X + C'\leq K_L* (a_1...a_v) + C_R + D_R$$

The interpretation is that the time space determined by a recurring event is bounded and consisting of time intervals defined by the lines $\{K_L* (a_1...a_v)+C_L... K_R* (a_1...a_v)+ C_R\}$, for t_{lower}, the earliest and latest times for a recurring event to start and $\{K_L* (a_1...a_v)+ C_L + D_L... K_L* (a_1...a_v)+ C_R + D_R\}$ for t_{upper}, the earliest and latest times for a recurring event to be ended, respectively.However each time line alone, a_λ $(\lambda\leq v)$, is expressing a separate-monadic periodic event. Proving this argument will enable us to use the above intervalic representation, for expressing periodical facts. In this case the parameters (K, C, D, X) are well known. Let us consider for inductive purposes a pair of the above lines named as t_L and t_R, where $t_L = K_L* (a_1...a_v) + C_L$, $t_R = K_L* (a_1...a_v)+ C_L + D_L$.

Proof

For $v=1$, $t_L = K_L* (a_1...a_1) + C_L= C_L$, $t_R = K_L *(a_1...a_1)+ C_L + D_L= C_L + D_L$, $a_1=0$ denotes the first occurrence of an event. t =[t_L, t_R] denotes a time interval that points to the first occurrence of a periodic event. Let us assume that t_L, t_R are expressing a periodic event (a) and stand for λ, occurrences, $\lambda\in N$, $t_{L(\lambda)} = K_L* (a_1...a_\lambda) + C_L$, $t_{R(\lambda)} = K_L*(a_1...a_\lambda)+ C_L + D_L$ (1)

If the above argument is correct then, t_L, t_R must stand for $\lambda+1$ occurrences, $\lambda\in N$

$t_{L(\lambda+1)} = K_L* (a_1...a_{\lambda+1}) + C_L$,(2,) $t_{R(\lambda+1)} = K_L *(a_1...a_{\lambda+1})+ C_L + D_L$ (3), $a_{(\lambda+1)} = a_\lambda +1$ (4)

(2) is rewritten through (1) and (4) as following:

$$t_{L(\lambda+1)} = K_L * (a_1 \ldots a_{\lambda+1}) + C_L = K_L * (a_1 \ldots a_\lambda + 1) + C_L = t_{L(\lambda)} + K_L$$

(3) is rewritten through (1) and (4) as following:

$$t_{R(\lambda+1)} = K_L * (a_1 \ldots a_{\lambda+1}) + C_L + D_L = K_L * (a_1 \ldots a_\lambda + 1) + C_L + D_L = t_{R(\lambda)} + K_L$$

The new expressions for (2) and (3) are proving that the $\lambda+1$ occurrences of a periodic event *(a)* arising from the λ ones by adding the characteristic frequency of repetition, which in fact is the definition of a periodic event. Next, an extended relational model and algebraic operators are defined for extraction of data, after estimating the validity (*Bel*) and possibility (*Pls*), degrees to which a tuple belongs to the a relation R.

4. Representation of Conflicting Information

Definition: Let T be a set of time intervals $T=\{[t_L, t_R]$ where $t_L=C+K\times X$, $t_R=C'+K\times X \wedge a_1 \leq X \leq a_v\}$ and D a set of non temporal values. A generalised tuple of temporal arity x and data arity l is an element of $T^x \times D^l$ together with constraints on the temporal elements. In that sense a tuple can be viewed as defining a potentially infinite set of tuples. Each extended relation consists of generalised tuples as defined above. Each extended relation has a virtual tuple membership attribute formed by a selection predicate either value or temporal that models the necessary (*Bel*) and possible degrees (*Pls*) to which a tuple belongs to the relation. The domain of tuple membership attribute is the Boolean set Ω = {true, false}. The possible subsets to that are {true}, {false} and Ω. The support set for tuple membership can be denoted by a pair of numbers (*Bel*, *Pls*) where:

$$Bel = \text{m}\ \{|\text{true}|\}$$

$$Pls = \text{m}\{|\text{true}|\} + \text{m}\{\Omega\} \text{ with property } 0 \leq Bel \leq Pls \leq 1$$

A tuple with (*Bel*, *Pls*) = [1,1] corresponds to a tuple that qualifies with full certainty. A tuple with (*Bel*, *Pls*) = [0,0] corresponds to a tuple that is believed not to qualify with full certainty. A tuple with (*Bel*, *Pls*) = (0,1) corresponds to complete ignorance about the tuple's membership. At this point two issues arise:

- the generalisation of the closed world assumption (CWA)
- the estimation of the (*Bel*, *Pls*) measures

The CWA is assuming that facts not found in the database are considered to be false. Since tuples memberships in our model vary between $0 \leq Bel \leq Pls \leq 1$ CWA needs to be extended. In generalising the CWA it is assumed that if a fact is not represented in the extended relation, then it must have Bel=0, and $Pls \leq 1$. Using Table I it can be appreciated that using the generalised CWA, the concepts {Southern Hemisphere, Northern Hemisphere} will not be replaced by their children as defined in Figure 1 since there is no support for them. There is only one vote and this can only be capitalised by the high level concepts {Southern Hemisphere, Northern Hemisphere}. However, the query 'Which are all the people who visited Brazil" is still expecting an answer. In answering this query, all tuples in Table II have to be classified according to the selection predicate *location = "Brazil"* which forms a virtual attribute (it exists as long as the execution of the query), it is not stored as part of Table II. In this sense tuples {X_1, X_4, X_5, X_7} are believed to satisfy the virtual attribute *location = "Brazil"* with certainty (*Bel*, *Pls*) = [1,1]. However, it cannot be said with full certainty (*Bel*, *Pls*) = [1,1] whether {X_2, X_3, X_6} satisfies the virtual attribute or not. In order to estimate the (*Bel*, *Pls*) measures, the ability to identify higher and lower level concepts for elements defined from a structured domain (either lattice, or tree) as specified by a particular application is needed. Let *l* be an element defined by a structured domain *L*. *U*(*e*) is the set of higher level concepts, i.e. $U(e) = \{n | n \in L \wedge n$ is an ancestor of $l\}$, and *L*(*e*) is the set of lower concepts $L(e) = \{n | n \in L \wedge n$ is a descendent of $l\}$. If *l* is a base concept then $L(e) = \varnothing$ and if *l* is a top level concept, then $U(e) = \varnothing$. If *L* is an unstructured domain then $L(e) = U(e) = \varnothing$. Considering tuple {$X_2$, X_3, X_6} and the selection predicate *location = "Brazil"* then *L*(*e*), *U*(*e*)are defined as follows:

U(Brazil) = {Southern Hemisphere, Northern Hemisphere} *L*(Brazil) = ∅

Rule 1: If ($|U(e)| > 1 \wedge L(e) = \varnothing$), e.g. |*U*(Brazil)|=2, *L*(Brazil) = ∅, then it is simply declared that a child or base concept has many parents (lattice structure). Therefore a child or base concept acting as a selection predicate can claim any tuple (parent) containing elements found in *U*(*e*), as its ancestor, but not with full certainty ($Bel > 0$, $Pls \le 1$). This is presented by the following interval (*Bel*, *Pls*) = (0,1].

Now consider the case where the selection predicate is defined as follows *location = "Southern Hemisphere"*. Tuple {X_2} fully satisfies the selection predicate and thus (*Bel*, *Pls*) = [1,1]. Tuples {X_3, X_6} do not qualify as an answer and thus (*Bel*, *Pls*) = [0,0]. However it cannot be said with full certainty ((*Bel*, *Pls*) = [1,1]) whether {X_1, X_4, X_5, X_7} satisfy the selection predicate or not, since Brazil belongs to both concepts {Southern Hemisphere, Northern Hemisphere}. Using the functions *L*(*e*), *U*(*e*) this can be deduced as follows

U(Southern Hemisphere) = {∅}

L(Southern Hemisphere) = {Brazil, Chile}

B= *U*(*L*(Southern Hemisphere)∧(R.*concept*)) = *U*(Brazil) = {Southern Hemisphere, Northern Hemisphere}. Formally the function can be defined as follows:

$B((l_1),(l_2)) = U(L(l_1) \wedge (l_2))$ where l_1 is a high level concept, l_2 is a base concept are elements defined in a lattice structured domain. If both arguments are high level concepts or low level concepts then $B((l_1),(l_2)) = \varnothing$. Function $B((l_1),(l_2))$ is defined only in a lattice structured domain.

Rule 2: If $B((l_1),(l_2))$ is defined and $|B((l_1),(l_2))| > 1$, then it is simply declared that multiple parents, high level concepts, are receiving a base concept as their own child. Therefore a parent or high level concept acting as a selection predicate can claim any tuple (child) containing elements found in $(L(l_1) \wedge (l_2))$, as its descendant, but not with full certainty ($Bel > 0$, $Pls \le 1$), presented by the following interval (*Bel*, *Pls*) = (0,1].

Similarly a temporal selection predicate can use the above functions (*U*(*e*), *L*(*e*), $B((l_1),(l_2))$) for imprecise temporal information, representing the time dimension as intervals, by labelling each node in the lattice with a time interval. The use of a lattice-structured domain by an application permits also the representation of temporal information at different levels of granularity.

Next an extended relational algebra is defined which operates on our model. The operations differ from the traditional ones in several ways: The selection/join condition of the operations may consist of base concepts or high level concepts.

Membership threshold ($Bel \geq \Phi$, *Pls*) may be specified with a selection/join condition to constrain the number of result tuples. The results of extended relational operators either retain or generate the new tuple membership in the case where more than one selection criteria are specified as base concepts or high level concepts.

5. Extended Relational Algebra

We are considering, for illustration purposes, the four operations σ (selection), π (projection), $\bowtie$ (join), $\cup$ (set union).

Selection: Selection is defined as follows: σ_P (R): $=\{t \mid t \in R \wedge P(t) = true\}$ where P denotes a selection condition. There are two types of a selection condition. A data selection condition (P_d) considers the snapshot relation R in Table 1. The temporal selection condition (P_t) is specified as a function of three arguments P_t: = < *K,D,C*> which is mapped to the time hierarchy H_r. It has to be mentioned that temporal constraints are included in the result tuples. The combined predicate over relation VT(R) in Table II is defined as follows: P:= $P_d \mid P_t \mid P_d \wedge P_t$. The selection support function $F_s(t_{A1..An}, P)$ returns a (*Bel*, *Pls*) pair indicating the support level of tuple *t* for the selection condition P, where $A_1..A_n$ is the set of attributes, excluding the virtual membership attribute. The selection support function F_s utilises the ($U(e)$, $L(e)$, $B((l_1),(l_2))$) functions in conjunction with Rule-1 and Rule-2, as defined in section 4, for estimating the actual support values. Recall that a compound predicate is formed by a conjunction of two or more atomic predicates. In this paper it is assumed that the atomic predicates are mutually independent.

The support for the compound predicate P:= $P_d \mid P_t \mid P_d \wedge P_t$ is computed based on the multiplicative rule.

$F_s(P) = (F_s(t_{A1..An}, P_t) \wedge F_s(t_{A1..An}, P_d)) =(Bel_1 \times Bel_{2.....} \times Bel_n, Pls_1 \times Pls_2 \times Pls_n)$ (1).

A discussion on combining supports of dependent predicates can be found in [7], [8].

Move (μ): Sometimes it is needed to move a temporal interval, that is given an existing interval, a new interval with the same duration but in another time. In that sense periodical or recurring patterns can be detected. This can be achieved with the use of Scale and Shift operators. **Scaling** is defined as multiplying the repetition factor $K \times X$ by a factor of α, and $\alpha \in N$. α is defined as the difference of (K_z - K_x) and $x < z$. α is either an odd (2n+1) or even (2n) number. If the scaling factor α, is not constant for all the time intervals that a fact is defined over, then the specific fact is at least recurring. If the scaling factor is constant and the duration is constant, then the fact is called periodical. Scaling may result in moving in higher linear time hierarchies in the case of recurring patterns e.g. (H_2= day$\subseteq$month$\subseteq$year). **Shifting** is defined as adding an offset θ to the upper and lower bounds of a sequence of time intervals. Shifting guarantees that there is no altering in the linear time hierarchy. Shifting is measured as the absolute value of the difference between the upper and lower bounds of successive time intervals. If the offset θ is constant then a fact is called periodical, otherwise is recurring. Assuming two time intervals of the form: It_1= [t_{lower}, t_{upper}], It_2 = [t_{lower1}, t_{upper1}], where $t_{lower} = K \times X + C$, $t_{upper} = K \times X + C'$, t_{lower1}

$= K_1 \times X + C_1$, $t_{upper1} = K_1 \times X + C_1^{'}$. In such cases an interval can be derived from another by applying a combination of transformations (shifting the lower and upper bound of It_1 by a factor of α and an offset of θ). The lower bound has to be scaled by and shifted simultaneously as following: $\alpha * x + \theta = (K_1\text{-}K) \times X + (C_1\text{-} C)$, where $\alpha = (K_1\text{-}K)$, $\theta_1 = (C_1\text{-} C)$. The upper bound of It_1 has to be scaled and shifted by a factor of $\alpha = (K_1\text{-}K)$, and an offset of $\theta_2 = (C1'\text{-} C^{'})$. The combined transformation for both bounds is of the following form: $\alpha * x + \theta = \{\{\alpha \times X + \theta_1\}, \{(\alpha) \times X + \theta_2\}\}$.

Projection: π_X (R):={t(X) | t ∈R}, where R is a relation on scheme S, t is a tuple with scheme X and X is a subset of S (X ⊆ S). Projection retains all valid time values like standard projection. Projection is defined on top of a selection. The intuition is that, as an operator it does not modify F_s, that is the tuple membership.

Join: Let *R, S* be two extended relations, P be the join condition and Q the membership threshold condition. The extended join operator is defined as a Cartesian Product, followed by an extended selection: $R \bowtie_P^Q S \equiv \sigma_P^Q(R \times S)$ where the tuple membership function is deviated by F_s (1) as in the case of the extended select operation. The time interval that the tuple membership is defined over is the intersection of the time intervals that the sources ($A_1 \ldots A_n$) are defined. $\Delta t(_{Fs}) = \Delta t_1 A_1 \cap \Delta t_2 A_2 \cap \ldots \cap \Delta t_n A_n$. (2). Assuming two intervals with lower bounds $t_L = C + KX$ and $t_{L'} = C' + K_1X$ and upper bounds $t_R = C_1 + KX$ and $t_{R'} = C_1' + K_1X'$. The time interval for the result is defined as $t_{L'} = C'' + K'X$ the common lower bound where $C' = \max(C, C')$, and $K' = \min(K, K_1)$, and $t_{U'} = C_3 + K'X$ the common upper bound where $C_3 = \max(C_1\ C_1')$, and $K' = \min(K, K_1)$.

Union: Union compatibility means that two extended relations *R, S* are union compatible if and only if they have the same arity or degree and their corresponding attributes are based on the same domain. For the set operators, including union, uncertainty can be introduced when relations with different levels of refinement for the same information are combined. Without extra knowledge it is reasonable to choose the information with the finest granularity as the one to be classified with full certainty (*Bel*, *Pls*) = [1,1]. Information not in the finest granularity is classified with no full certainty (*Bel*, *Pls*) = (0,1]. Both types of information are part of the result tuples, accompanied by different beliefs. Union is formally defined as follows

$R \cup S \equiv$ {t| (∃r) (∃s) (r ∈ *R* ∧ s ∈ *S* ∧ t. *K* = r. *K* = s. *K*) ∧ (t (*Bel*, *Pls*) = F_s (r. (*Bel*, *Pls*), s (*Bel*, *Pls*)) } *K* is the arity of the relation, F_s denotes the selection support function. Tuples with different valid times are not merged, independently of the fact that they are expressing the same snapshot tuple.

6. Open Issues

Our model can operate as a summary interface for the temporal probabilistic model, proposed by [10], since it uses general semantics, true, uncertain, false. The difference is that in our model indefinite temporal information is encoded as part of the tuples, through our time model. Furthermore infinite temporal information can also be encoded. The temporal relational model with algebraic operations permits the representation and querying of structured (tree or lattice) domains. Uncertainty resolution both at the instance and query level can be represented and resolved with the use of algebraic operations which determined the tendency of things to occur (*Bel*) and might happen ability of things to occur (*Pls*). It is considered that in a virtual integrated environment the community of member databases is not stable. Frequent changes in this community take place with new information providers added or existing ones modified and deleted. The target is to design algorithms for projective and selective transformation defined as:

Projective transformation: Discover the columns of the new contribution with respect to the validity and the semantics of the initial degrees of belief and possibility.

Selective transformation: Discover new tuples from the new contribution via the projective transformation with the aid of a membership threshold, without destroying the validity and the semantics of our initial view.

In both stages of discovery the likelihood of a successful discovery must be estimated and eventually all new tuples must be tagged with a level of confidence.

References

1. **Motro A, Anokhin P, Berlin J**, Intelligent Methods in Virtual Databases, Proceedings of the Fourth International Conference on Flexible Query Answering Systems, FQAS 2000, Physica-Verlag Heidelberg, October, 2000, Warsaw Poland
2. **Hull R,** Managing Semantic Heterogeneity in Databases: A Theoretical Perspective. PODS 1997: 51-61, Proceedings of the Sixteenth *ACM SIGACT-SIGMOD-SIGART* Symposium on Principles of Database Systems, May 12-14, 1997, Tucson, Arizona
3. **Lim E, Sristava, J, Prabhakar S, Richardson J**, Entity Identification Problem in Database Integration, Proceedings of the 9th IEEE, Data Engineering Conference, *IEEE Computer Society Press* April 1993, Vienna, Austria
4. **Chountas, P, Petrounias, I.,** Representation of Definite, Indefinite and Infinite Temporal Information, Proceedings of the 4th International Database Engineering & Applications Symposium (IDEAS'2000), *IEEE Computer Society Press*, September 2000, Japan.
5. **Koubarakis M**, Database Models for Infinite and Indefinite Temporal Information, Information Systems, Vol. 19, No. 2, pages 141-173, 1994
6. **Chountas, P, Petrounias, I**., Modelling and Representation of Uncertain Temporal Information, Requirements Engineering Journal*, 5(3), pp 144-156, Springer Verlag, 2000.*
7. **Lakshmanan L.V.S, Leone R, Subrahmanian V. S**, ProbView: *A Flexible Probabilistic Database System*, ACM Transactions on Database Systems, Volume 22, No. 3 (Sep. 1997)
8. **Baldwin J. F**, Evidential Support Logic Programming**"** ***Fuzzy sets and Systems*****,** 24, pp 1-26, 1987
9. **C. Dyreson, R. Snodgrass**, *Support Valid-Time Indeterminacy*, ACM Transactions on Database Systems, Vol. 23, No. 1, pp. 1-57, 1998