[8] J. Paredaens, P. De Bra, M. Gyssens and D. Van Gucht, The Structure of the Relational Database Model, Springer-Verlag, Berlin Heidelberg, 1989.

[9] Rundensteiner E. A., Hawkes L. W. and Bandler W., On Nearness Measures in Fuzzy Relational Data Models, Int. J. of Approximate Reasoning 3, 1989, 267-298.

[10] Tansel A. U., Clifford J., Gadia S., Jajodia S., Segev A. and Snodgrass R., Temporal Databases: Theory, Design and Implementation, The Benjamin/Cummings Publishing Company Inc., 1993.

[11] Zadeh L. A., The concept of a linguistic variable and its application to approximate reasoning, Information Sciences, Vol. 8, 1975, 199-249 (Part I), 301-357 (Part II).

[12] Zadeh L. A., Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets and Systems, Vol. 1, No. 1, 1978, 3-28.

[13]Zemankova M. and Kandel A., Implementing Imprecision in Information Systems, Information Sciences 37, 1985, 107-141.

**University of South Florida**
USF

October 30, 1998

Dr. Werasak Kurutach
Department of Computer Engineering
Mahanakorn University of Technology
51 Cheum-Sampan Rd., Nong Chok
Bangkok 10530
THAILAND

Dear Dr. Kurutach:

This letter is in response to your query about the acceptance ratio and rationale for acceptance for the NAFIPS'98 conference. The review process included two reviews from scholars in the field. These reviews were then shared with the author to allow him/her to improve the final paper. The reviews were also used to decide on acceptance of papers. Approximately 20% of the submitted papers were rejected. Only papers that showed a clear contribution or an idea with potential use or a theory that explained real phenomena were accepted.

We appreciate your contribution to NAFIPS'99 and hope you will participate in NAFIPS'99 in New York.

Sincerely,

Lawrence O. Hall

Lawrence O. Hall
NAFIPS'98 Program Chair
internet:hall@csee.usf.edu
Phone:813-974-4195
FAX:813-974-5456

# Processing of Binary Temporal Constructors for Fuzzily-Bounded Time Intervals in Temporal Databases

Werasak Kurutach
Department of Computer Engineering
Mahanakorn University of Technology
51 Cheum-Sampan Rd., Nong Chok, Bangkok, THAILAND
E-mail. werasak@mut.ac.th

## Abstract

*This paper presents an extended notion of two primitive set-theoretic operations for the use in a fuzzy interval-based temporal model and also studies its properties. Such a model has been proposed to handle the fuzziness aspect of time that can arise in the temporal database environment[4]. In general, set-theoretic operators are regarded as temporal constructors in query processing of temporal databases. Therefore, our major aim of this study is to provide the concept and techniques to process those temporal constructors with the existence of fuzziness in temporal data. The result of this understanding can provide a basic ground for further study of a query evaluation in temporal databases that need to handle the fuzziness of time.*

## 1 Introduction

Representation and processing of time are important in many areas of research such as temporal databases [10], artificial intelligence [1], [6] and control systems in a dynamic environment [8]. However, different fields have distinct inherent features involving time [7]. In this paper, we will concentrate on a treatment of temporal knowledge from the temporal database perspective. Generally, temporal databases are databases whose recorded data are attached by a time dimension (i.e. either *valid time* or *transaction time* or both) and the processing of temporal data is handled implicitly by the database system. In the literature of this research field, it has been recently recognized that imperfect knowledge may be present in time observed from the modeled world [2], [3], [9]. Among several kinds of imperfection, fuzziness is believed to be one of the most primitive aspects [5]. Unfortunately, only few temporal database researchers have focused on this aspect. Recently, we have introduced a fuzzy interval-based temporal model for temporal databases [4]. In that work, the structure and its semantic employed in the temporal model and six unary temporal constructors have been proposed. However, such constructors are primitive and only used for extracting the components of a time interval, e.g. the starting point of a time interval. Generally, in a temporal query processing, *binary temporal* (or *bitemporal*) *constructors* are necessarily required for producing a time interval from two time intervals. Such bitemporal constructors can be built on the concepts of set-theoretic operations such as the *union* operator and the *intersection* operator. Therefore, this work will present an extension of the two classical set-theoretic operators to operating on fuzzily-bounded time intervals. Such extended operators will acts as bitemporal constructors for our fuzzy interval-based temporal model.

The paper is organized as follows. In Section 2, the preliminary notion of fuzzy time representation and its semantics interpretation will be presented as well as some unary temporal constructors. Section 3 shows how set-theoretic operations, particularly union and intersection, can be processed on fuzzily-bounded time intervals and used as bitemporal constructors. Finally, section 4 provides some discussion on further work and summerizes this paper.

## 2 The Preliminary Concept of A Fuzzy Time Model

### 2.1 Temporal Specifications

In a fuzzy interval-based temporal model [4], knowledge of fuzzy time can be represented by a *temporal specification* which is formally defined as follows.

**Definition 2.1** *A temporal specification is a 2-tuple* $\langle [s,e], C(d) \rangle$*, where* $s$ *and* $e$ *are the starting and ending time points, respectively, that are represented by* unimodal[1] *fuzzy natural numbers. That is to say,* $[s,e]$ *models the imprecisely known boundaries of a time interval, and* $C(d)$*, is the duration constraint*

[1] *A unimodal fuzzy set is a fuzzy set that is convex and normal.*

*restricting the possible time periods of the fuzzy time interval.*

A duration constarint $C(d)$ defined by $c_1 \leq d \leq c_2$, where $c_1$ and $c_2$ are integers greater than zero, means that a possible time period is a value between $c_1$ and $c_2$ inclusive. To achieve more useful information from the construct of a temporal specification, its semantic is needed to be specified in terms of a fuzzy set of intervals.

**Definition 2.2** *Let $\mathcal{T}$ be a universal set of temporal specifications and $\aleph$ be a set of natural numbers corresponding to the chronons[2] on a time axis. The semantic interpretation, denoted by $\Im$, is a function assigning the meaning to each temporal specification, i.e. $\Im : \mathcal{T} \longrightarrow [0,1]^{\aleph \times \aleph}$, where $[0,1]^{\aleph\times\aleph}$ is a set of fuzzy subsets defined on $\aleph \times \aleph$. In the other words, if $T_S = \langle [s,e], C(d)\rangle$ is in $\mathcal{T}$, the semantic interpretation $\Im(T_S)$ can be defined by*

$$\begin{aligned}\Im(T_S) = \{&\mu_{\Im(T_S)}([n_s,n_e])/[n_s,n_e] \mid \\ &[n_s,n_e] \in \aleph \times \aleph, (n_e \ominus n_s) \models C(d), \text{and} \\ &\mu_{\Im(T_S)}([n_s,n_e]) = \min(\mu_s(n_s), \mu_e(n_e))\} \quad (1)\end{aligned}$$

*where $(n_e \ominus n_s)$ means $n_e - n_s + 1$, and $\models$ denotes "satisfies".*

From Definition 2.1 and 2.2, it is obvious that $s$ and $e$ are *interactive* [11] via the interactivity constraint $C(d)$ which is needed to produce meaningful results and this is helpful in reducing imprecision.

### 2.2 Unary Temporal Constructors

Based on the interpretation defined by Definition 2.2, we can define three functions, which are used as unary temporal constructors, as follows.

$$\begin{aligned}
start(T_S) &= \{\mu_{start(T_S)}(n_s)/n_s \mid \mu_{start(T_S)}(n_s) = \\ &\quad \sup_{n_e} \mu_{\Im(T_S)}([n_s,n_e])\} \\
end(T_S) &= \{\mu_{end(T_S)}(n_e)/n_e \mid \mu_{end(T_S)}(n_e) = \\ &\quad \sup_{n_s} \mu_{\Im(T_S)}([n_s,n_e])\} \\
dur(T_S) &= \{\mu_{dur(T_S)}(d)/d \mid \mu_{dur(T_S)}(d) = \\ &\quad \sup_{d=n_e\ominus n_s} \min(\mu_{start(T_S)}(n_s), \mu_{end(T_S)}(n_e))\}
\end{aligned}$$

By rewriting a temporal specification in terms of its unary operators, one can obtain the same semantic as shown in the Lemma 2.1 below. Therefore, expressing a temporal specification in the latter form has no *information loss* with respect to the former and will be more useful in its processing and in providing nonconflicting information among the three components.

**Lemma 2.1** *Given $T_S = \langle [s,e], C(d)\rangle$ be a temporal specification and let $T_S' = \langle [start(T_S), end(T_S)], \tilde{d} = dur(T_S)\rangle$ be another temporal specification obtained by applying unary operators to $T_S$. Then, both of them will have the same semantics, that is,*

$$\Im(T_S) = \Im(T_S') \quad (2)$$

## 3 Binary Temporal Constructors

In terms of temporal databases, the union and intersection operators are called *binary temporal constructors* which take two (fuzzy) time intervals[3] specified by temporal specifications and produce a new (fuzzy) one. But, first, let us define the definitions of set-theoretic operations for the case of precise time intervals. In such a case, given that the two intervals $[a,b]$ and $[c,d]$ be *overlapping* (or *meeting* ($\bowtie$) for the case of the union operation) [1], the operations are defined as

$$\begin{aligned}[a,b] \cup [c,d] &= [\min(a,c), \max(b,d)] \quad \text{and} \\ [a,b] \cap [c,d] &= [\max(a,c), \min(b,d)]\end{aligned}$$

It should be noted that the union operation on two meeting or overlapping precise time intervals will result in a precise time interval and so is the intersection operation on two overlapping intervals. Similarly, the notion of *definability* can be designed as the condition to allow the results of the two binary temporal constructors operating on two fuzzy time intervals to be a fuzzy time interval.

**Definition 3.1** *Let $t_I$ and $t_I'$ be two fuzzy time intervals defined by the temporal specifications $T_I$ and $T_I'$, respectively, and let $\mathcal{I}$ be the universal set of precise and valid time intervals. Then, $T_I \theta T_I'$, where $\theta \in \{\cup, \cap\}$, is said to be* defined *if and only if $\forall i \forall i' \in \mathcal{I}$,*
$(\mu_{\Im(T_I)}(i) > 0) \wedge (\mu_{\Im(T_I')}(i') > 0) \rightarrow i\theta i' \in \mathcal{I}$

Assume that the union and intersection operations are defined according to Definition 3.1, their semantics can be defined as follows.

[2] A chronon, as assumed in the discrete time model[10], is a nondecomposable segment of the continuous time line and is isomorphic to a natural number.

[3] A fuzzy time point can be regarded as a special case of a fuzzy time interval with the duration constraint $d = 1$.

**Definition 3.2** *Let $T_I$ and $T_I'$ be the temporal specifications of two (fuzzy) time intervals, and let $I$ be the set of precise and valid time intervals with $i, i', i'' \in I$. Then, the operation $T_I \theta T_I'$, where $\theta \in \{\cup, \cap\}$, can be defined by*

$$\mu_{\Im(T_I \theta T_I')}(i) = \sup_{i = i' \theta i''} \min(\mu_{\Im(T_I)}(i'), \mu_{\Im(T_I')}(i'')) \quad (3)$$

Based on Definition 3.2, we can derive the interpreted temporal components of $T_I \theta T_I'$, where $\theta \in \{\cup, \cap\}$, as follows

$$\mu_{start(T_I \theta T_I')}(n_s) = \sup_{n_e} \mu_{\Im(T_I \theta T_I')}([n_s, n_e]) \quad (4)$$

$$\mu_{end(T_I \theta T_I')}(n_e) = \sup_{n_s} \mu_{\Im(T_I \theta T_I')}([n_s, n_e]) \quad (5)$$

$$\mu_{dur(T_I \theta T_I')}(d) = \sup_{d = n_e \ominus n_s} \mu_{\Im(T_I \theta T_I')}([n_s, n_e]) \quad (6)$$

Since the structure of a temporal specification is an important notion in representing fuzzy temporal knowledge in the model, it is essential to express the result received from a binary temporal constructor in terms of a temporal specification, that is, the temporal specification of $T_I \theta T_I'$ can be expressed by $([start(T_I \theta T_I'), end(T_I \theta T_I')], \tilde{d} = dur(T_I \theta T_I'))$. However, to do so, the notion of lossless of information with respect to the semantic interpretation $\Im(T_I \theta T_I')$ may have to be sacrificed. Fundamentally, information in a temporal specification $([start(T_I \theta T_I'), end(T_I \theta T_I')], \tilde{d} = dur(T_I \theta T_I'))$ is said to be *lossless* if and only if

$$\Im(T_I \theta T_I') = \Im(([start(T_I \theta T_I'), end(T_I \theta T_I')], \tilde{d} = dur(T_I \theta T_I')))$$

and *lossy* if and only if

$$\Im(T_I \theta T_I') \subset \Im(([start(T_I \theta T_I'), end(T_I \theta T_I')], \tilde{d} = dur(T_I \theta T_I')))$$

Therefore, in general, we will have

$$\Im(T_I \theta T_I') \subseteq \Im(([start(T_I \theta T_I'), end(T_I \theta T_I')], \tilde{d} = dur(T_I \theta T_I'))) \quad (7)$$

In order to receive lossless information from the binary temporal constructors, one needs to enforce the definability of the operations as follows.

**Lemma 3.1** *Given two temporal specifications $T_I$ and $T_I'$, if $T_I \theta T_I'$ is defined according to Definition 3.1, then information expressed in the temporal specification $([start(T_I \theta T_I'), end(T_I \theta T_I')], \tilde{d} = dur(T_I \theta T_I'))$ is lossless.*

Now we will consider a formalism to derive the interpreted temporal components of the results of the union and intersection. Its formulation is straightforward and provides the same semantic as Equations 4 and 5 for the *start* and *end* functions, respectively. The duration constraints can be derived from the duration constraints of the operands in a very simple way. Then, the interpreted duration constraint in Equation 6 can be evaluated from those three as illustrated in Equation 2 of Definition 3.2.

**Procedure 3.1** *Let $T = ([s, e], C(d))$ be the temporal specification resulting from the defined operation $T_j \theta T_k$, where $\theta \in \{\cup, \cap\}$. Assume that $C_j(d)$ and $C_k(d)$ be the duration constraints in $T_j$ and $T_k$ and be expressed by $d_{j_l} \leq d \leq d_{j_u}$ and $d_{k_l} \leq d \leq d_{k_u}$, respectively. Then, for the union operation, i.e. $\theta$ is $\cup$, the components $start(T)$ and $end(T)$ can be characterized by*

$$\mu_{start(T)}(n) = \max\{\sup_{n \leq n'} \min(\mu_{start(T_j)}(n), \mu_{start(T_k)}(n')), \sup_{n \leq n'} \min(\mu_{start(T_k)}(n), \mu_{start(T_j)}(n'))\} \quad (8)$$

$$\mu_{end(T)}(n) = \max\{\sup_{n \geq n'} \min(\mu_{end(T_j)}(n), \mu_{end(T_k)}(n')), \sup_{n \geq n'} \min(\mu_{end(T_k)}(n), \mu_{end(T_j)}(n'))\} \quad (9)$$

*respectively, and the duration constraint $C(d)$ can be expressed by*

$$\max(d_{j_l}, d_{k_l}) \leq d \leq d_{j_u} + d_{k_u} \quad (10)$$

*For the intersection operation, i.e. $\theta$ is $\cap$, the components $start(T)$ and $end(T)$ can be characterized by*

$$\mu_{start(T)}(n) = \max\{\sup_{n \geq n'} \min(\mu_{start(T_j)}(n), \mu_{start(T_k)}(n')), \sup_{n \geq n'} \min(\mu_{start(T_k)}(n), \mu_{start(T_j)}(n'))\} \quad (11)$$

$$\mu_{end(T)}(n) = \max\{\sup_{n \leq n'} \min(\mu_{end(T_j)}(n), \mu_{end(T_k)}(n')), \sup_{n \leq n'} \min(\mu_{end(T_k)}(n), \mu_{end(T_j)}(n'))\} \quad (12)$$

*respectively, and the duration constraint $C(d)$ can be expressed by*

$$1 \leq d \leq \min(d_{j_u}, d_{k_u}) \quad (13)$$

In fact, as illustrated in Procedure 3.1, the derivation of the membership functions in Equations 4 and 5 is very simple, especially in terms of parameterized membership functions such as trapezoidal functions, when $T_I \theta T'_I$ is defined.

**Example 3.1** *Let us have two temporal specifications,*

$$
\begin{aligned}
T_1 &= ([f(x;5,10,15,20), f(x;40,45,50,55)], \\
&\quad 36 \leq d \leq 51) \text{ and} \\
T_2 &= ([f(x;10,15,20,25), f(x;60,65,70,75)], \\
&\quad 51 \leq d \leq 66)
\end{aligned}
$$

*Since $T_1 \cup T_2$ and $T_1 \cap T_2$ are defined, then we can obtain*

$$
\begin{aligned}
T_1 \cup T_2 &= ([f(x;5,10,15,20), \\
&\quad f(x;60,65,70,75)], 51 \leq d \leq 117)
\end{aligned}
$$

*and*

$$
\begin{aligned}
T_1 \cap T_2 &= ([f(x;10,15,20,25), \\
&\quad f(x;40,45,50,55)], 1 \leq d \leq 51)
\end{aligned}
$$

Basically, there is a strong restriction on the temporal relationship between two fuzzy time intervals that allows the intersection and union operations to be *defined*. It can be described as follows

**Proposition 3.1** *Let $T_I$ and $T'_I$ be the temporal specifications of two fuzzy time intervals. Also let*

$$
\begin{aligned}
n_a &= \sup Supp(start(T_I)), \\
n_b &= \inf Supp(end(T_I)), \\
n'_a &= \sup Supp(start(T'_I)) \text{ and} \\
n'_b &= \inf Supp(end(T'_I)),
\end{aligned}
$$

*where $Supp(\,)$ is the support of a fuzzy set. Then, given that the $[n_a, n_b]$ and $[n'_a, n'_b]$ are valid intervals,*

1. *the operation $T_I \cap T'_I$ is defined if and only if $[n_a, n_b]$ and $[n'_a, n'_b]$ are overlapping*[4]
2. *the operation $T_I \cup T'_I$ is defined if and only if*
   (a) *$[n_a, n_b]$ and $[n'_a, n'_b]$ are overlapping, or*
   (b) *$[n_a, n_b]$ and $[n'_a, n'_b]$ are meeting*

**Proposition 3.2** *Following Proposition 3.1, given that the $[n_a, n_b]$ is an invalid interval (i.e., $n_b < n_a$) and $[n'_a, n'_b]$ is a valid intervals,*

1. *the operation $T_I \cap T'_I$ is defined if and only if $[n_b, n_a] \sqsubseteq [n'_a, n'_b]$*[5] *(the symbol $\sqsubseteq$ means* during *or* equality *relationship)*
2. *the operation $T_I \cup T'_I$ is defined if and only if*
   (a) *$[n_b, n_a] \sqsubseteq [n'_a, n'_b]$, or*
   (b) *$[n_a, n_b]$ and $[n'_a, n'_b]$ are meeting*

**Proposition 3.3** *Following Proposition 3.1, given that both $[n_a, n_b]$ and $[n'_a, n'_b]$ are not valid intervals, the operations $T_I \cap T'_I$ and $T_I \cup T'_I$ are never defined*

Proposition 3.3 also implies that if any temporal specification $T_S$ has the interpreted starting and ending compoenents possibly overlapping to some extent, the union and intersection operations on itself are not defined. That is, $T_S \cap T_S$ and $T_S \cup T_S$ are not defined in such a case. Propositions 3.1 – 3.3 show a very simple way of verifying whether the intersection and union operations are defined, especially when parameterized functions such as trapezoidal functions are employed. We can also conclude from the propositions that if

$$max(Poss(T_I \prec T'_I), Poss(T'_I \prec T_I)) = 0$$

then $T_I \theta T'_I$ is always defined, where $\theta \in \{\cap, \cup\}$. The above condition seems to be intuitive, and this issue should be generalized when the *undefined* intersection and union operations are to be processed. The treatment of such undefined operations will be left for further investigation.

To justify the semantics of the proposed set-theoretic operations, it is necessary to show that they satisfy some intuitive mathematical properties.

**Proposition 3.4** *Let $\mathcal{T}$ be a universal set of temporal specifications and $U$ be the temporal specification specifying the interval equivalent to the time axis. Both union and intersection of (fuzzy) time intervals satisfy the following laws.*

1. *Identity*

$$
\begin{aligned}
\forall T \in \mathcal{T}, \quad & T \cup U = U \\
& T \cap U = T
\end{aligned}
$$

*Note that $T \cup \phi$ and $T \cap \phi$ are not defined in our approach. An obvious reason is that $\phi$ is not a normalized interval*

[4] In terms of temporal relations, this means $Poss(T_I \prec T'_I) = 0$ and $Poss(T'_I \prec T_I) = 0$ (see [4] for more details)

[5] In terms of temporal relations this means $Poss(end(T'_I) \preceq start(T_I)) = 0$ and $Poss(end(T_I) \preceq start(T'_I)) = 0$ (see [4] for more details)

*2. Associative*

$$\forall T_1, T_2, T_3 \in \mathcal{T},$$
$$(T_1 \cup T_2) \cup T_3 = T_1 \cup (T_2 \cup T_3)$$
$$(T_1 \cap T_2) \cap T_3 = T_1 \cap (T_2 \cap T_3)$$

*3. Commutative*

$$\forall T_1, T_2 \in \mathcal{T}, \quad T_1 \cup T_2 = T_2 \cup T_1$$
$$T_1 \cap T_2 = T_2 \cap T_1$$

*4. Distributive*

$$\forall T_1, T_2, T_3 \in \mathcal{T}$$
$$T_1 \cap (T_2 \cup T_3) = (T_1 \cap T_2) \cup (T_1 \cap T_3)$$
$$T_1 \cup (T_2 \cap T_3) = (T_1 \cup T_2) \cap (T_1 \cup T_3)$$

It should be noted that the property of *idempotent* which is satisfied in the case of precise valid time interval, i.e.

$$\forall t \in \mathcal{T}, \quad t \cup t = t$$
$$t \cap t = t$$

is not necessarily satisfied in the case of fuzzy time intervals. However, a more generalized property can be derived. That is, $\forall T \in \mathcal{T}, \forall t \in I$

$$\mu_{(T \cup T)}(t) \geq \mu_{T}(t), \text{ and}$$
$$\mu_{(T \cap T)}(t) \leq \mu_{T}(t)$$

This implies that $\mathfrak{S}(T) \subseteq \mathfrak{S}(T \cup T)$ and $\mathfrak{S}(T) \subseteq \mathfrak{S}(T \cap T)$. But, when the constraint on its possible durations is not enforced, i.e., $(0 \leq d \leq \omega)$, the equality will replace the subset predicate.

## 4 Conclusion

This work has discussed an extended concept of two primitive set-theoretic operations, union and intersection, to be used as binary temporal constructors in our fuzzy interval-based temporal model. Such a model can be employed as a basis for temporal databases with fuzzy knowledge of time. It has been shown that algorithms of the operations are quite simple when unimodal and parameterized membership functions have been used with the enforcement of the notion of *definability*. In addition, it has been proved that the two temporal constructors satisfy intuitive mathematical properties, i.e. identity, associative, commutative and distributive. Though the concept of definability of the operations seems to be sufficient to allow simple computation in a temporal data model with tuple timestamping, it needs to further investigate the case when the operations are not defined. This may be useful for an attribute timestamping data model[10].

**Acknowledgement.**

The author would like to thank Prof. H. Takagi for his helpful comments on this work and The Thailand Research Fund for providing a grant to the project.

## References

[1] Allen J. F., Maintaining Knowledge about Temporal Intervals, *Communications of ACM*, Vol. 26, No. 11, November 1983, pp. 832-843.

[2] Dutta S., Generalized Events In Temporal Databases, *Proc. of IEEE Int. Conf. on Data Engineering*, Los Angeles, CA, USA, February 1989, pp. 118-125.

[3] Dyreson C. E. and Snodgrass R. T., Valid-time indeterminacy, *Proc. of IEEE Int. Conf. on Data Engineering*, Vienna, Austria, April 1993, 335-343.

[4] Kurutach W., Modeling Fuzzy Interval-based Temporal Information: A Temporal Database Perspective, *Proc. of the Int. Joint Conf. of the 4th IEEE Int. Conf. on Fuzzy Systems and the 2nd Int. Fuzzy Engineering Symp.*, Yokohama, Japan, March 20-24, 1995, pp. 741-748.

[5] Kurutach W., Managing Different Aspects of Imperfect Data in Databases: A Unified Approach, *Proc. of the 1995 IEEE Int. Conf. on Systems, Man and Cybernetics*, Vancouver, BC, Canada, October 22-25, 1995, pp. 2812-2817.

[6] Kahn K. and Gorry G. A., Mechanizing Temporal Knowledge, *Artificial Intelligence* 9, 1977, pp. 87-108.

[7] Marchetti S., Misano P. and Pernici B., Temporal Data Management Systems: A Comparative View, *IEEE Trans. on Knowledge and Data Engineering*, Vol. 3, No. 4, December 1991, pp. 504-523.

[8] Qian D., Representation and use of imprecise temporal knowledge in dynamic systems, *Fuzzy Sets and Systems* 50, 1992, pp. 59-77.

[9] Schiel U., Representation and Retrieval of Incomplete and Temporal Information, *Technical Report DSC 02/1987*, Departamento De Sistemas E Computacao, Universidade Federal Da Paraiba, Brazil, May 1987.

[10] Tansel A. U., Clifford J., Gadia S., Jajodia S., Segev A. and Snodgrass R., *Temporal Databases: Theory, Design and Implementation*, The Benjamin/Cummings Publishing Company Inc., 1993.

[11] Zadeh L. A., The concept of a linguistic variable and its application to approximate reasoning, *Information Sciences*, Vol. 8, 1975, 199-249.

NJIT
New Jersey Institute of Technology

*A Public
Research University*

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING

December 3, 1998

Asst. Prof. Dr. Werasak Kurutach
Head, Department of Computer Engineering
Assistant President for IT
Mahanakorn University of Technology
51 Cheum-Sampan Rd., Nong Chok
Bangkok 10530, Thailand

Dear Prof. Kurutach:

I am pleased to share with you some facts about the papers submitted to 1998 IEEE International Conference on Systems, Man and Cybernetics, San Diego, CA, October 11-14, 1998.

1) Every paper was reviewed by three experts in the field and
2) The acceptance rate is 80%.

We would like to thank you very much for your great contributions to the conference. Please let me know if you need any further help.

Sincerely,

MengChu Zhou, Ph.D.
Program Chair
1998 IEEE International Conference on Systems, Man and Cybernetics
(973) 596-6282
(973) 596-5680 Fax
zhou@njit.edu

UNIVERSITY HEIGHTS
NEWARK, NJ 07102-1982
973. 596. 3512, 3510
973 596. 5680 FAX

# Handling Fuzziness in Temporal Databases

Werasak Kurutach
Department of Computer Engineering
Mahanakorn University of Technology
51 Cheum-Sampan Rd., Nong Chok, Bangkok 10530, Thailand
e-mail: werasak@mut.ac.th

## ABSTRACT

This paper proposes a new data model, called *FuzzTime*, which is capable of handling both aspects of fuzziness and time of data. These two features can always be encountered simultaneously in many applications. This work is aimed to be a conceptual framework for advanced applications of database systems. Our approach has extended the concept of the relational data model to have such a capability. The notion of linguistic variables, fuzzy set theory and possibility theory have been employed in handling the fuzziness aspect, and the discrete time model has been assumed. Some important time-related operators to be used in a temporal query evaluation with an existence of fuzziness are also discussed.

## 1 INTRODUCTION

Primarily, database modeling is an important concept underlying all information systems. To develop an intelligent information system, we need to increase the capability of database systems. Traditional database management systems can't handle imperfect data available together with their implicit temporal information, and this temporal information itself can also be imperfect. In fact, there are many types of imperfection that can appear in data to be represented [7]. However, in this work, we will emphasize *the fuzziness aspect*, a primitive kind of imperfection. Even though the two features (time and fuzziness) are important in this respect, work on incorporating both of them in a database system has been received little attention. Most of previous proposals concern either of them [10, 9]. Some exception is research accomplished by Dutta [2] who proposed a temporal data model capable of handling fuzzy events. In addition, Dyreson [3] has proposed a temporal data model which can treat the probabilistic kind of uncertainty in temporal knowledge.

In this work, we will propose a temporal data model, called *FuzzTime*, which can represent fuzzy states/events together with their corresponding fuzzy time intervals/points and also can evaluate simple temporal queries of those information. The concept of the relational data model [8] will be extended to incorporate such a capability. The fuzzily-bounded time interval model previously proposed by the author [6] is to be employed in order to allow *FuzzTime* to handle fuzzy time. The notion of linguistic variables [11] can be viewed as corresponding to the concept of attributes in the relational data model, and, therefore, attribute values in this sense can be fuzzy. In addition, the possibility theory [12] will be used to provide an interpretation of a fuzzy value. Some major time related operators for query evaluations with fuzziness existence in both temporal and nontemporal data will be presented.

The paper is organized as follows. Section 2 will explain the preliminary concepts, the definition of a temporal specification and the notion of temporal constructors, to be employed as a basis of modeling fuzzy time in our *FuzzTime*. Then, the structure of the model and its semantic interpretation will be described in Section 3. In Section 4, we will show how various operators concerning the fuzzy time concept can be evaluated. Finally, we will summarize our work in Section 6.

## 2 THE PRELIMINARY CONCEPTS

### 2.1 Temporal Specification

Imprecise knowledge of time can be modeled by a construct called *a temporal specification* which is composed of two elements - an approximation of the duration of time and a time interval with two fuzzy bounds. Such a structure can represent both a fuzzy time point and a fuzzy time interval. Formally, it can be defined as follows.

**Definition 2.1** [1] *A temporal specification is a 2-tuple* $\langle [s,e], C(d) \rangle$, *where* $s$ *and* $e$ *are the starting*

[1] This definition is defined based on a *discrete time model* where the time line is divided into the shortest intervals of a fixed duration. The shortest duration of time is called a *chronon*. It is nondecomposable unit of time and isomorphic to the natural numbers [4].

*and the ending time points, respectively, that are represented by fuzzy natural numbers. In other words, the component* $\langle s, e \rangle$ *of the 2-tuple models the approximately known boundaries of a time interval. The other component,* $C(d)$, *is the duration constraint modeling the possible time periods of the interval.*

The notation $C(d)$ in the above definition is defined as a range of integers in order to restrict all possible lengths of a fuzzy time interval. That is, a duration constraint $C(d)$, represented as $c_1 \leq d \leq c_2$, where $c_1$ and $c_2$ are integers greater than zero, means that the possible durations of the imprecisely known interval are the values between $c_1$ and $c_2$ inclusive. Moreover, in order to reserve the meaningful semantic of a temporal specification, the semantic interpretation is to be proposed and is formally defined as follows.

**Definition 2.2** *Let* $I$ *be a universal set of temporal specifications and* $\mathbb{N}$ *be a set of natural numbers corresponding to the chronons on a time axis. The semantic interpretation, denoted by* $\Im$, *is a function assigning the meaning to each temporal specification, i.e.,* $\Im : I \longrightarrow [0,1]^{\mathbb{N}\times\mathbb{N}}$, *where* $[0,1]^{\mathbb{N}\times\mathbb{N}}$ *is a set of fuzzy subsets defined on* $\mathbb{N}\times\mathbb{N}$. *In the other words, if* $T_S = \langle\langle s, e\rangle, C(d)\rangle$ *is in* $I$, *the semantic interpretation* $\Im(T_S)$ *can be defined by*

$$\begin{aligned}\Im(T_S) = \{&\mu_{\Im(T_S)}([n_s, n_e])/[n_s, n_e] \mid \\ &[n_s, n_e] \in \mathbb{N}\times\mathbb{N}, (n_e - n_s) \models C(d) \text{ and} \\ &\mu_{\Im(T_S)}([n_s, n_e]) = \min(\mu_s(n_s), \mu_e(n_e))\}\end{aligned}$$

*where* $[n_s, n_e]$ *means* $n_s \leq n_e + 1$, *and* $\models$ *denotes "satisfies".*

## 2.2 Unary Temporal Constructors

Based on the semantic interpretation, it is necessary to define three functions. Each of them takes a temporal specification as the argument and produces the constructed fuzzy set which describes the vague meaning of the starting time, the ending time or the duration. These are called *unary temporal constructor*.

Let $T_S = \langle\langle s, e\rangle, C(d)\rangle$ be a temporal specification with the interpretation $\Im(T_S)$. Then, the functions *start*, *end* and *dur* of $T_S$ can be respectively defined by

$$\begin{aligned}start(T_S) &= \{\mu_{start(T_S)}(n_s)/n_s \mid \mu_{start(T_S)}(n_s) = \\ &\qquad \sup_{n_e \in \mathbb{N}} \mu_{\Im(T_S)}([n_s, n_e])\} \\ end(T_S) &= \{\mu_{end(T_S)}(n_e)/n_e \mid \mu_{end(T_S)}(n_e) = \\ &\qquad \sup_{n_s \in \mathbb{N}} \mu_{\Im(T_S)}([n_s, n_e])\} \\ dur(T_S) &= \{\mu_{dur(T_S)}(d)/d \mid \mu_{dur(T_S)}(d) = \\ &\qquad \sup_{\substack{d = n_e - n_s \\ n_s, n_e \in \mathbb{N}}} \min(\mu_{start(T_S)}(n_s), \\ &\qquad \mu_{end(T_S)}(n_e))\}\end{aligned}$$

By rewriting a temporal specification in terms of its unary operators, one can obtain the same semantic as shown below.

**Proposition 1** *Given* $T_S = \langle\langle s, e\rangle, C(d)\rangle$ *be a temporal specification and let* $T_S' = \langle\langle start(T_S), end(T_S)\rangle, d = dur(T_S)\rangle$ *be another temporal specification obtained by applying unary operators to* $T_S$. *Then, both of them will have the same semantics, that is,* $\Im(T_S) = \Im(T_S')$.

## 2.3 Binary Temporal Constructors

In terms of temporal databases, the union and intersection operators are called *binary temporal constructors* which take two (fuzzy) time intervals[2] specified by temporal specifications and produce a new (fuzzy) one. But, first, let us define the definitions of set theoretic operations for the case of precise time intervals. In such a case, given that the two intervals $[a, b]$ and $[c, d]$ be *overlapping* (or *meeting* [5] for the case of the union operation) [1], the operations are defined in terms of membership functions as

$$\mu_{[a,b]\cup[c,d]}(t) = \begin{cases} 1.0 & \text{if } t \in [\min(a,c), \max(b,d)] \\ 0.0 & \text{otherwise}\end{cases}$$

and

$$\mu_{[a,b]\cap[c,d]}(t) = \begin{cases} 1.0 & \text{if } t \in [\max(a,c), \min(b,d)] \\ 0.0 & \text{otherwise}\end{cases}$$

respectively. It should be noted that the union operation on two meeting or overlapping precise time intervals will result in a precise time interval and so is the intersection operation on two overlapping intervals. This basic concept can be generalized and leads to the notion of *definability* of the operations for fuzzy time intervals.

**Definition 2.3** *Let* $t_{s_1}$ *and* $t_{s_2}$ *be two fuzzy time intervals defined by the temporal specifications* $T_{S_1}$ *and* $T_{S_2}$, *respectively, and let* $I$ *be the universal set of precise and valid time intervals. Then,* $T_{S_1}\theta T_{S_2}$, *where* $\theta \in \{\cup, \cap\}$, *is said to be* defined *if and only if* $\forall x \forall y \in I\ (\mu_{\Im(T_{S_1})}(x) > 0) \wedge (\mu_{\Im(T_{S_2})}(y) > 0) \Rightarrow (x\theta y) \in I$.

Assume that the union and intersection operations are defined according to Definition 2.3, their semantics can be defined as follows.

[2] A fuzzy time point can be regarded as a special case of a fuzzy time interval with the duration constraint $d = 1$.

**Definition 2.4** *Let $Ts_m$ and $Ts_n$ be the temporal specifications of two (fuzzy) time intervals, and let $I$ be the set of precise and valid time intervals with $i, j, k \in I$. Then, the operation $Ts_m \theta Ts_n$, where $\theta \in \{\cup, \cap\}$, can be defined by*

$$\mu_{\Im(Ts_m \theta Ts_n)}(i) = \sup_{i = j \theta k} \min(\mu_{\Im(Ts_m)}(j), \mu_{\Im(Ts_n)}(k))$$

Based on Definition 2.1, we can derive the interpreted temporal components of $Ts_m \theta Ts_n$, where $\theta \in \{\cup, \cap\}$, as follows

$$\mu_{start(Ts_m \theta Ts_n)}(n_s) = \sup_{n_e} \mu_{\Im(Ts_m \theta Ts_n)}([n_s, n_e])$$

$$\mu_{end(Ts_m \theta Ts_n)}(n_e) = \sup_{n_s} \mu_{\Im(Ts_m \theta Ts_n)}([n_s, n_e])$$

$$\mu_{dur(Ts_m \theta Ts_n))}(d) = \sup_{d = n_e - n_s} \mu_{\Im(Ts_m \theta Ts_n)}([n_s, n_e])$$

## 3 FuzzTime: A FUZZY TEMPORAL DATA MODEL

### 3.1 The Structure of FuzzTime

In this section, a fuzzy temporal data model, called *FuzzTime*, is to be introduced. The structure of *FuzzTime* is expanded from the relational data model to allow the representation of both temporal data and fuzzy data. In our data model, a relation scheme consists of three primitive components - a finite set of attributes, a finite set of domains and a function that associates with each attribute a domain. That is, a *temporal relation scheme* $R$ is defined as a three-tuple

$$(A \cup A_T, D \cup D_T, dom) \tag{1}$$

where $A \cup A_T$ is a finite set of attribute names, $D \cup D_T$ is a finite set of data domains. A function *dom* associates each attribute name with a data domain, i.e., $dom(A) = D_A$ where $A \in A \cup A_T$ is an attribute name and $D_A \in D \cup D_T$ is the domain of attribute $A$. As one can see, *FuzzTime* has enforced a finite set of special attributes $A_T$ in the scheme, and these are the implicit attributes[3] (which is in contrast to the explicit attributes in $A$). It contains all attributes that are relevant to the temporality features - the number of reincarnation (to be explained later), the starting time, the ending time and the lower and the upper bounds of the duration constraint. That is,

$$A_T = \{A_r, A_s, A_e, A_l, A_u\} \tag{2}$$

where $A_r$ is the reincarnation attribute, $A_s$ is the starting time attribute, $A_e$ is the ending time attribute, $A_l$ is the lower bound attribute and $A_u$ is the upper bound attribute. The last four attributes represent the elements in the construct of a temporal specification defined in Definition 2.1. The corresponding domain set of $A_T$ is $D_T$. Since every attribute in $A_T$ takes a natural number as its value, $D_T$ can be a singleton set whose element is the set of natural numbers.

Alternatively, the temporal relation scheme above can be defined as a collection of attributes. Thus, it can be considered in more details by emphasizing the structure of an attribute. The struture of an attribute is defined as an extended notion of a linguistic variable in order to allow fuzziness to be treated. That is, an attribute *Attr* is defined as[4]

$$Attr = (A, U, T, G, M) \tag{3}$$

where $A$ is the attribute name, $U$ is the basic data domain, $T$ is the domain of linguistic terms, $G$ is the syntactic rule[5] and $M$ is the semantic rule. $M$ can be defined as a function

$$M : U \cup T \rightarrow [0, 1]^U \tag{4}$$

where $M(u) = \{1/u\}$ for every $u \in U$. More specifically, the semantic rule applied to a linguistic term in $T$ will result in a fuzzy set. Any element of $U \cup T$ can be assigned as a value of the attribute $A$. In the other words, it can be said that

$$dom(A) = U \cup T$$

*A temporal relation instance* $r$ (*or a temporal relation, for short*) of a scheme $R$ consisting of $n$ attributes $(A_1, A_2, \ldots, A_n)$ can be defined as follows

$$r \subseteq \times_{i=1}^{n} dom(A_i) \times_{j \in \{r, s, e, l, u\}} dom(A_j)$$

The constructs of both an attribute and a relation scheme make *FuzzTime*, an extended relational data model, able to handle both aspects of fuzziness and time in a homogeneous framework.

### 3.2 The Concept of A Key

An identification of an object is also important in any data model, and *FuzzTime* is no exception. *A time-invariant key (TIK)* primarily can be considered as a key of a snapshot relation of a temporal relation[6]. In the other words, it can be employed as an

[3] An implicit attribute is an attribute that will be handled by the system and needs not be declared explicitly by users.

[4] Since an attribute name is used to identify the concept of an attribute, the terms *attribute name* and *attribute* are to be used interchangeably.

[5] In this work, for the sake of simplicity, the syntactic rule is not of our interest.

[6] A snapshot relation is a relation at a point of time. Hence, it can be obtained by slicing a temporal relation at a time point.

identification of an object (a tuple) in the snapshot relation. Therefore, it is necessary to assume that all time-invariant key attributes of a relation scheme in *FuzzTime* have precise values, that is, their term sets are the empty set ($T = \phi$). In order to support the concept of an object's reincarnation, an additional implicit attribute, denoted by $A_r$, is needed in the relation scheme in order to identify the number of reincarnation (such as the first time, the second time and so on). Thus, every temporal relation will have the set of attributes $TIK \cup \{A_r\}$ as a key to uniquely identify a tuple. However, there is an important constraint arising in this respect. It will enforce that an object cannot be reincarnated while it is still valid. Formally, let $t_1$ and $t_2$ be any two tuples in a temporal relation, and $T_{S_1}$ and $T_{S_2}$ denote two temporal specifications $\langle [t_1[A_s], t_1[A_e]], t_1[A_l] \le d \le t_1[A_u] \rangle$ and $\langle [t_2[A_s], t_2[A_e]], t_2[A_l] \le d \le t_2[A_u] \rangle$, respectively. The constraint can be stated as follows.

*If* $t_1[TIK] = t_2[TIK]$ *and* $t_1[A_r] \ne t_2[A_r]$, *then* $T_{S_1} \cap T_{S_2}$ *is not defined according to Definition*2.3

### 3.3 An Example

A simple example given in this section illustrates a fuzzy temporal database. It contains data about the temperatures of patients. These data represent events in the real world that occur since January 1, 1998. Hence, it can be assumed that chronon 0 is corresponding to the date *January 1,1998*, and the granularity of concern is *day*. Now, let *Patient* be a relation scheme of the database with $A = \{Name, Temp\}$. The domains of the two attributes are specified as

$$\begin{aligned} dom(Name) &= \{x \mid x \text{ is a patient's name}\} \\ dom(Temp) &= \{35.0, 35.5, 36.0, ..., 45.0\} \cup \\ &\quad \{\text{too high, high, medium, low,} \\ &\quad \text{too low}\} \end{aligned}$$

Since the attribute name *Name* is supposedly the *TIK* attribute, its term set is the empty set. Based on the relation scheme, a relation instance could be shown as the following table.

| $A_r$ | Name | Temp | $A_s$ | $A_e$ | $A_l$ | $A_u$ |
|---|---|---|---|---|---|---|
| 1 | John | 35.0 | 0 | $\sim 6$ | 5 | 8 |
| 1 | Jim | high | 45 | $\sim 50$ | 5 | 7 |
| 1 | Max | medium | 47 | 51 | 5 | 5 |
| 1 | Nick | high | $\sim 50$ | $\sim 50$ | 1 | 1 |
| 2 | John | 39.0 | $\sim 25$ | $\sim 30$ | 6 | 6 |

Note that the vertical double line is used here to separate the explicit attributes (i.e., *Name* and *Temp*) from the implicit ones. Also, the notation $\sim$ denotes the term *around*. For instance, $\sim 6$ refers to the term *around Jan 7, 1998*. Based on the data in the table, it can be illustrated from the first tuple that John's temperature is 35°$C$ during the time between *Jan 1, 1998* and *around Jan 7, 1998*, and the number of days that he had this temperature is between 5 and 8 days inclusive. However, his temperature measured several days later is also recorded (*reincarnation*) and shown in the last tuple. Jim's temperature is *high* for either 5, 6 or 7 days during the time between *Feb 15, 1998* and *around Feb 20, 1998*. It should be noted that the fourth tuple represents an event which occurred at a time point, that is, Nick has a *high* temperature on a day *around Feb 20, 1998*. Then, one may characterize the semantics of some fuzzy terms as follow.

$$\begin{aligned} M(high) &= T(x : 37, 39, 40, 42) \\ M(medium) &= T(x : 35, 36, 37, 38) \\ M(\text{around Jan 7, 98}) &= T(x : 4, 6, 6, 8) \\ M(\text{around Feb 15, 98}) &= T(x : 43, 45, 45, 47) \\ M(\text{around Feb 20, 98}) &= T(x : 48, 50, 50, 52) \end{aligned}$$

where $T(x : a, b, c, d)$ is a trapezoidal function defined as [5]

$$T(x : a, b, c, d) = \begin{cases} 1 & \text{when } b \le x \le c \\ 0 & \text{when } x < a \text{ and } x > d \\ \frac{a-x}{a-b} & \text{when } a \le x \le b \\ \frac{d-x}{d-c} & \text{when } c \le x \le d \end{cases}$$

### 3.4 An Interpretation of The Model

Based on the possibility theory [12], we can interpret a fuzzy value of an attribute as a possibility distribution. Generally, for each tuple $t$ in a relation instance $r$, there will be a corresponding semantic, denoted by $M(t)$, in the form of possible tuples each of which has an associted degree of possibility. The attribute values in each possible tuple are precise. More formally,

$$M(t) = \{\mu(t_1)/t_1, \mu(t_2)/t_2, ..., \mu(t_m)/t_m\}$$

where $\mu(t_i) = \inf_j Poss(t[A_j] = t_i[A_j])$, $t[A_j]$ and $t_i[A_j]$ denote the respective values of the attribute $A_j$ of the tuple $t$ and $t_i$, and $1 \le j \le n$. The possibility degree $Poss(t[A_j] = t_i[A_j])$ is derived from the grade of membership that $t_i[A_j]$ is a member of $M(t[A_j])$. For example, from the second tuple, it is possible with the degree 0.5 that Jim's temperature is 38°$C$ during *15-20 Feb 1998*, where the degree 0.5 is derive from $\min(\mu_{M(high)}(38), \mu_{M(\sim 50)}(50))$.

## 4 TIME RELATED OPERATORS FOR QUERY EVALUATION

In this section, we will consider two major operators that are related to temporal data and extensively used in traditional temporal data models.

**Time Extraction Operators** ($\varphi$ and $\varphi'$)

In the *FuzzTime* model, there are two operators to extract temporal knowledge of an event. They are useful for enquiring *when an event has possibly occured* ($\varphi'$) and *what knowledge of the event's occurrence time is represented* ($\varphi$). Primarily, the two operators take a fuzzy temporal relation and a selection condition as the input. A selection condition is used to select tuples from the relation for being extracted their time stamping. The output of the operators will be represented as a temporal relation with a special attribute[7](denoted by $A_\mu$) whose value indicates the degree of membership of each tuple in the relation, In the other words, it can be shown as a set of an ordered pair consisting of temporal specifications and corresponding degrees of membership. That is, given $r$ be a relation instance and *cond* be a condition, the operators can be defined as follows.

$$\begin{aligned}
&\varphi(cond : r) = \\
&\quad \{(<[t[A_s], t[A_e]], t[A_l] \le d \le t[A_u]>, \mu) | \\
&\quad t \in r, \mu = \min(\mu_{cond}(t), t[A_\mu])\} \\
&\varphi'(cond : r) = \\
&\quad \{(<[start(T_S), end(T_S)], d_l \le d \le d_u>, \mu) | t \in r, \\
&\quad T_S =<[t[A_s], t[A_e]], t[A_l] \le d \le t[A_u]>, \\
&\quad dur(T_S) \equiv d_l \le d \le d_u, \\
&\quad \mu = \min(\mu_{cond}(t), t[A_\mu])\}
\end{aligned}$$

where $\mu_{cond}(t)$ is the degree that the tuple $t$ satisfies the condition. For example, a user may need to retrieve a piece of represented temporal knowledge of an event that Jim's temperature is $38°C$. This can be written as

$$\varphi(Name = "Jim" \wedge Temp = 38 : Patient)$$

which will result in $\{(<[45, \sim 50], 5 \le d \le 7>, 0.5)\}$. The degree 0.5 can be evaluated by

$$\begin{aligned}
&\mu_{(Name="Jim" \wedge Temp=38)}(t) = \\
&\quad \min(\mu_{M(t[Name])}(Jim), \mu_{M(t[Temp])}(38))
\end{aligned}$$

It is obvious that only the second tuple in the table satisfies the condition with the degree greater than 0.

Four other unary operators need to be defined in order to allow one to extract individual temporal element of a temporal relation. They are extended from the unary operators defined previously for a temporal specification in Section 2.

$$\begin{aligned}
S(r) &= \{<start(T_S), \mu> | t \in r, \\
&\quad T_S =<[t[A_s], t[A_e]], t[A_l] \le d \le t[A_u]>, \\
&\quad \mu = t[A_\mu]\} \\
E(r) &= \{<end(T_S), \mu> | t \in r, \\
&\quad T_S =<[t[A_s], t[A_e]], t[A_l] \le d \le t[A_u]>, \\
&\quad \mu = t[A_\mu]\} \\
L(r) &= \{<d_l, \mu> | t \in r, \\
&\quad T_S =<[t[A_s], t[A_e]], t[A_l] \le d \le t[A_u]>, \\
&\quad dur(T_S) \equiv d_l \le d \le d_u, \mu = t[A_\mu]\} \\
U(r) &= \{<d_u, \mu> | t \in r, \\
&\quad T_S =<[t[A_s], t[A_e]], t[A_l] \le d \le t[A_u]>, \\
&\quad dur(T_S) \equiv d_l \le d \le d_u, \mu = t[A_\mu]\}
\end{aligned}$$

For instance, one may want to find the starting time when Jim's temperature is $38°C$. This can be expressed as

$$S(\varphi(Name = "Jim" \wedge Temp = 38 \;:\; Patient)) = \{<45, 0.5>\}$$

However, the operators $\varphi$ and $\varphi'$ need to be extended to allow a logical expression of order pairs of a condition and its corresponding relation as the input. This extension is useful in two situations. Firstly, one may need to extract a common occurrence time of various kinds of events. The other is when the user want to request for an occurrence time that either of various events has taken place. It can be said that a kind of events is represented in a relation of a database. Therefore, in general, the time extraction operators can be defined as follows:

$$\begin{aligned}
\varphi(\vee_{i=1}^{n}(cond_i : r_i)) &= \cup_{i=1}^{n}\varphi(cond_i : r_i), \\
\varphi(\wedge_{i=1}^{n}(cond_i : r_i)) &= \cap_{i=1}^{n}\varphi(cond_i : r_i), \\
\varphi'(\vee_{i=1}^{n}(cond_i : r_i)) &= \cup_{i=1}^{n}\varphi'(cond_i : r_i), \\
\varphi'(\wedge_{i=1}^{n}(cond_i : r_i)) &= \cap_{i=1}^{n}\varphi'(cond_i : r_i)
\end{aligned}$$

Any two temporal specifications in the process of this operation that are union/intersection *definable* are needed to be coalesced. As a consequence, the value of the attribute $A_\mu$ in a tuple $t$ resulted from such a coalescence can be evaluated as follows.

$$\begin{aligned}
t[A_\mu] &= \max_i(t_i[A_\mu]), \\
&\quad \text{where } \exists t_i \in r_i, t[T_S] = \cup_i t_i[T_S] \\
t[A_\mu] &= \min_i(t_i[A_\mu]), \\
&\quad \text{where } \exists t_i \in r_i, t[T_S] = \cap_i t_i[T_S]
\end{aligned}$$

where $t_i[T_S]$ refers to the temporal specification of a tuple $t_i$.

[7]This special attribute in a *based relation* has a default value of 1.0 for each tuple, and, therefore, this attribute is not shown in the structure of a based relation

### Temporal Relationship Based Operator ($\delta$)

This operator is designed to retrieve data from a relation by using a temporal relationship as a condition. That is, a tuple will be selected for further processing if and only if it relates to the temporal specification specified in the condition with the degrees greater than the threshold values. Due to the space limitations, we will show only two relationships, and the others can be specified similarly. Generally, these two operators are generally referred to as *time slice* operators. They can be defined as follows.

Given $r$ be a relation instance of a relation scheme consisting of $n$ explicit attributes (i.e., $\mathcal{A} = \{A_1, ..., A_n\}$), $T_S = < [n_s, n_e], l \le d \le u >$ be a temporal specification, $\sqsubseteq$ be the *during or equality relationship* (see [6] for more details of temporal relationships), the time slice can be defined as follows:

$$\delta_{\sqsubseteq T_S}(r) = \{< t, \mu > | \\ t \in r,\ \mu = Poss(\Im(t) \sqsubseteq \Im(T_s))\}$$

and

$$\begin{aligned} \delta_{\sqsupseteq T_S}(r) = \{< t', \mu > | \\ & \exists t \in r, t'[A_i] = t[A_i], 1 \le i \le n, \\ & t'[A_s] = n_s, t'[A_e] = n_e, \\ & t'[A_l] = l, t'[A_u] = u, \\ & \mu = Poss(\Im(\varphi(t)) \sqsubseteq \Im(T_s))\} \end{aligned}$$

where

$$Poss(\Im(\varphi(t)) \sqsubseteq \Im(T_s)) = \\ \sup_{i,j \in T, i \sqsubseteq j} \min(\mu_{\Im(\varphi(t))}(i), \mu_{\Im(T_S)}(j))$$

**Proposition 2** *The fuzzy temporal relation resulted from the time slice operators satisfy the key constraint defined previously (in Section 3.2).*

## 5 CONCLUSIONS

This work has proposed techniques to expand the relational data model to handle both fuzzy and temporal data. The structure of the data model has been designed to support a fuzzily-bounded time interval model as well as the concept of reincarnation of an object. The notion of an identification, known as *a key constraint*, has been enforced in the model. Important time-related operators have been discussed. They allow the capability of accessing the implicit temporal attributes in a database. There are some major issues needed to be further investigated for this fuzzy temporal data model such as temporal aggregate functions, an extended relational algebra, the case of undefined union and intersection and a design of a suitable query language.

**Acknowledgement:** This work is made possible partially by The Thailand Research Fund's grant number PDF/50/2540. The author would like to thank Prof. H. Takagi for sending him some materials for this research.

## 6 REFERENCES

[1] Allen J. F., "Maintaining knowledge about temporal intervals", *CACM*, Vol. 26, 1983, 832-843.

[2] Dutta S., "Generalized Events In Temporal Databases", *Proc. of IEEE Int. Conf. on Data Engineering*, Los Angeles, CA, USA, Feb. 1989, 118-125.

[3] Dyreson C., "Valid-time indeterminacy", *PhD dissertation*, Department of Computer Science, The University of Arizona, USA, 1994.

[4] Jensen C. S., Clifford J., Gadia S. K., Segev A. and Anodgrass R., "A glossary of temporal database concepts", *SIGMOD Record*, 21(3), September 1992, 35-43.

[5] Klir G. J. and Yuan B., *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall Inc., 1995.

[6] Kurutach W., "Modeling Fuzzy Interval-based Temporal Information: A Temporal Database Perspective", *Proc. of the 4th IEEE Int. Conf. on Fuzzy systems*, Yokohama, Japan, March 1995, 741-748.

[7] Kurutach W., "Managing Different Aspects of Imperfect Data in Databases: A Unified Approach", *Proc. of the 1995 IEEE International Conference on System, Man and Cybernetics*, Vancouver, BC, Canada, October 22-25, 1995, 2812-2817.

[8] Paredaens J., Bra P. D., Gyssens M. and Gucht D. V., *The Structure of the Relational Database Model*, Springer-Verlag, Berlin Heidelberg, 1989.

[9] Petry F. E., *Fuzzy Databases: Principles and Applications*, Kluwer Academic Publishers, 1996.

[10] Tansel A. U., Clifford J., Gadia S., Jajodia S., Segev A. and Snodgrass S., *Temporal Databases: Theory, Design and Implementation*, The Benjamin/Cummings Publishing Company, 1993.

[11] Zadeh L. A., "The concept of a linguistic variable and its application to approximate reasoning", *Information Sciences*, Vol. 8, 1975, 199-249 (Part I), 301-357 (Part II).

[12] Zadeh L. A., "Fuzzy sets as a basis for a theory of possibility", *Fuzzy Sets and Systems*, Vol. 1, No. 1, 1978, 3-28.

**5TH INTERNATIONAL CONFERENCE ON SOFT COMPUTING**

**OCTOBER 16-20, 1996**
**IIZUKA, FUKUOKA, JAPAN**

**Phone:+81-948-24-2771 FAX:+81-948-24-3002**
**E-mail: iizuka98@flsi.cird.or.jp**

IIZUKA'98

August 14, 1999

Asst. Prof. Werasak Kurutach, PhD(UNSW)
Head, Department of Computer Engineering
Assistant President for IT
Mahanakorn University of Technology
51 Cheum-Sampan Rd., Nong Chok
Bangkok 10530, Thailand

Dear Dr. Kurutach,

In reply to your inquiry on the refereeing process of the IIZUKA'98, I am writing this letter.

Three referees in corresponding field reviewed each paper submitted to the international conference, so called IIZUKA conference, and unless all of them rejected the paper, it was to be accepted. The rate of acceptance was around 80 %. This is based on the policy of IIZUKA conference that it should make a novel ideas emerge out of the traditional discipline. In other words, the low acceptance rate possibly impedes this emergence.

Sincerely yours,

Takeshi Yamakawa
Organizing Committee Chair of IIZUKA'98

Methodologies for the Conception, Design and Application of Soft Computing
Proceedings of IIZUKA '98, pp. 541-544

# Generalized Bitemporal Constructors for A Fuzzily-Bounded Time Interval Model

**Werasak Kurutach**
Department of Computer Engineering, Mahanakorn University of Technology
51 Cheum-Sampan Rd., Nong Chok, Bangkok 10530, THAILAND
e-mail: werasak@mut.ac.th

**Key Word:** fuzzy time, temporal databases, bitemporal constructors, temporal specifications, time modeling

## Abstract

Bitemporal constructors are important operators to be processed in queries against a temporal database. In our recent work [7], their processing has been proposed in the context of a fuzzy interval-based temporal model [6]. However, such a method can be carried out only under a restriction known as the *definability* of the operations. This constraint limits their expressive power as well as the scope of their applicability. Therefore, in this paper, we will present a generalization of the bitemporal constructors by relaxing such a restriction. This can be achieved by introducing the concept of *a null interval* for the case of the intersection operation and the notion of *a conjunctive specification* for the union operation.

## 1 Introduction

Primarily, a temporal database model describes a database as a collection of data with their time-stamps, and those time-stamps are implicitly handled by the system [8]. During the last two decades, there have been many proposals of temporal data models and their query languages [5], but issues in the possibility of an occurrence of *fuzziness* and/or *probabilistic uncertainty* in temporal data have been received little attention [2, 3, 9]. Recently, a fuzzily-bounded time interval model has been proposed as a basis for extending a temporal data model to handle fuzzy knowledge of time [6]. Moreover, we introduced the processing of bitemporal constructors with a restriction on their operands for the model [7]. In this paper, we will propose a generalized notion of the processing by relaxing such a restriction. This relaxation allows the more expressive power and more realistic in its applications. Generally, the notion of bitemporal constructors is important in a temporal query evaluation and a temporal relation construction. The concept of a temporal constructor is that it takes two temporal elements (i.e. time intervals) as the operands and provides a new temporal element (i.e. a time interval) as the result. Bitemporal constructors are corresponding to set-theoretic operators such as union and intersection and, therefore, we will use such terminology from the set theory. This work assumes the discrete time model which has been used extensively by most temporal database researchers. In this model, a nondecomposable portion (called *a chronon*) of the continuous time line is isomorphic to a natural number and, hence, time is linearly ordered [10]. From this point of view, a fuzzy time point can be defined as a fuzzy subset of the set of natural number and it is assumed that this fuzzy set is convex and normal.

The paper is organized as follows. Section 2 will present a background concept that is to be used as a basis for its following sections. In section 3, we will describe the concepts of generalized bitemporal constructors and their properties. Finally, section 4 will summarize our work and provides some discussion.

## 2 A Fuzzily-Bounded Time Interval and Its Semantic: A Preliminary Concept

In our fuzzy interval-based temporal model[6], a construct called *a temporal specification* has been introduced as a primitive concept to handle fuzzy knowledge of a time interval[1]. It is composed of two major components - the fuzzily-bounded interval and its uncertain length. Formally, it can be defined as follows.

**Definition 2.1** *A temporal specification is a 2-tuple* $([s,e], C(d))$, *where* $s$ *and* $e$ *are the starting and ending time points, respectively, that are represented by fuzzy natural numbers. In other words, the component* $[s,e]$ *of the 2-tuple models the approximately known boundaries of a time interval. The other component,*

[1] Based on the discrete time model assumption, a time point is a special case of a time interval.

*$C(d)$, is the duration constraint modeling the possible time periods of the interval.*

From Definition 2.1, the duration constraint $C(d)$ represents the uncertain length of a fuzzy time interval. This representation is assumably in the form of the lower bound and the upper bound of a range of integers. For example, a duration constraint $c_1 \leq d \leq c_2$ (denoted above as $C(d)$), where $c_1$ and $c_2$ are integers greater than zero, means that the possible durations (or lengths) are the values between $c_1$ and $c_2$ inclusive. However, the components of a temporal specification need to be interact in a meaningful way in order to produce a correct semantic of the temporal specification. This semantic is defined as a (fuzzy) set of possible intervals which arises from such an interaction. We formally call the interaction as *the semantic interpretation* and define it as follows.

**Definition 2.2** *Let $\mathcal{T}$ be a universal set of temporal specifications and $\aleph$ be a set of natural numbers corresponding to the chronons on a time axis. The semantic interpretation, denoted by $\Im$, is a function assigning the meaning to each temporal specification, i.e. $\Im : \mathcal{T} \longrightarrow [0,1]^{\aleph \times \aleph}$, where $[0,1]^{\aleph \times \aleph}$ is a set of fuzzy subsets defined on $\aleph \times \aleph$. In the other words, if $T_S = \langle [s,e], C(d)\rangle$ is in $\mathcal{T}$, the semantic interpretation $\Im(T_S)$ can be defined by*

$$\begin{aligned}\Im(T_S) = \{ & \mu_{\Im(T_S)}([n_s, n_e])/[n_s, n_e] \mid \\ & [n_s, n_e] \in \aleph \times \aleph, (n_e \ominus n_s) \models C(d), \text{ and} \\ & \mu_{\Im(T_S)}([n_s, n_e]) = \min(\mu_s(n_s), \mu_e(n_e))\}\end{aligned}$$

*where $(n_e \ominus n_s)$ means $n_e - n_s + 1$*[2]*, and $\models$ denotes "satisfies".*

Based on the semantic interpretation, it is necessary to define three functions. Each of them takes a temporal specification as the argument and produces the constricted fuzzy set which describes the vague meaning of the starting time, the ending time or the duration.

Let $T_S = \langle [s,e], C(d)\rangle$ be a temporal specification with the interpretation $\Im(T_S)$. Then, the functions *start*, *end* and *dur* of $T_S$ can be respectively defined by

$$\begin{aligned} start(T_S) &= \{\mu_{start(T_S)}(n_s)/n_s \mid \\ &\quad \mu_{start(T_S)}(n_s) = \sup_{n_e \in \aleph} \mu_{\Im(T_S)}([n_s, n_e])\} \\ end(T_S) &= \{\mu_{end(T_S)}(n_e)/n_e \mid \\ &\quad \mu_{end(T_S)}(n_e) = \sup_{n_s \in \aleph} \mu_{\Im(T_S)}([n_s, n_e])\}\end{aligned}$$

[2] That is, $(n_e \ominus n_s)$ is the number of the chronons between $n_s$ and $n_e$ inclusive.

$$\begin{aligned} dur(T_S) &= \{\mu_{dur(T_S)}(d)/d \mid \mu_{dur(T_S)}(d) = \\ &\quad \sup_{\substack{d = n_e \ominus n_s, \\ n_s, n_e \in \aleph}} \min(\mu_{start(T_S)}(n_s), \\ &\quad \mu_{end(T_S)}(n_e))\}\end{aligned}$$

By rewriting a temporal specification in terms of its unary operators, one can obtain the same semantic as shown below.

**Lemma 2.1** *Given $T_S = \langle [s,e], C(d)\rangle$ be a temporal specification, and let $T_S' = \langle [start(T_S), end(T_S)], \tilde{d} = dur(T_S)\rangle$ be another temporal specification obtained by applying unary operators to $T_S$. Then, both of them will have the same semantics, that is, $\Im(T_S) = \Im(T_S')$.*

# 3 Generalized Bitemporal Constructors

## 3.1 The Union Operation and The Concept of Conjunctive Specifications

In terms of temporal databases, the union operator is called a binary temporal (or bitemporal) constructor which operates on two time intervals and produces another time interval. However, the intervals that are unionable in that sense must be either *meeting* or *overlapping* [1]. This requirement is not suitable for *an attribute time-stamping temporal data model* [4]. There is a similar affect when using the union operator on the fuzzy time intervls defined by two temporal specifications. In such a case, the processing is quite simple if we enforce that all possible intervals are either *meeting* or *overlapping* [7]. The union operation under that constraint is said to be *defined*. As we mentioned previously, this restriction should be relaxed, and this will lead to the concept of *a conjunctive specification*[3] which can be defined as follows.

**Definition 3.1** *Let $T_j$ and $T_k$ be two temporal specifications. Then, $T_j \cup T_k$ is said to be* a conjunctive specification *if and only if it is not* defined, *that is, provided $\mathcal{I}$ is the universal set of precise and valid intervals in the traditional sense,*

$$\begin{aligned}&\exists i, \exists i' \in \mathcal{I}, \\ &\quad (\mu_{\Im(T_j)}(i) > 0 \wedge \mu_{\Im(T_k)}(i') > 0) \rightarrow i \cup i' \notin \mathcal{I}\end{aligned}$$

In general, a conjunctive specification can be $T_{S_1} \cup \ldots \cup T_{S_q}$ or $\bigcup_{i=1}^{q} T_{S_i}$, where $j \neq k, 1 \leq j,k \leq q$, and $T_{S_j} \cup T_{S_k}$ is not defined. Using precise intervals as a simple example, suppose $i_1$ and $i_2$ be

[3] This is similar to the notion of *a temporal element* proposed in a traditional temporal data model [4].

two nonconsecutive and nonoverlapping intervals, and $i = i_1 \cup i_2$ be the occurrence time of an event e. Then, i is said to be a *conjunctive interval*, and the event e occurs during both $i_1$ and $i_2$. That is to say, a conjunctive specification is to model the concept of an reincarnation of an object or an event. For example, a fact *A patient named John has been sick during around-mid-Jan-97 and around-end-Jan-97 and during around-5-Feb-97 and around-10-Feb-97* can be written as

Patient(John,sick,
$\langle$[around-mid-Jan-97,around-end-Jan-97] , $C(d)\rangle$
$\cup$ $\langle$[around-5-Feb-97,around-10-Feb-97] , $C'(d)\rangle$ )

It is important to specify the result when a unary temporal constructor is applied to a conjunctive specification. Since there is more than one time interval that an event takes place or is valid (e.g. reincarnation of an object), it will be natural to obtain nondecreasing values as the output of the constructors. Assume $\cup_{i=1}^{q} T_i$ be a conjunctive specification where $Poss(T_j \prec T_k)$ for every $j, k$ such that $1 \leq j < k \leq q$. Then, we can define that

$$\begin{aligned} start(\cup_{i=1}^{q} T_i) &= (start(T_1), \ldots, start(T_q)) \\ end(\cup_{i=1}^{q} T_i) &= (end(T_1), \ldots, end(T_q)) \\ dur(\cup_{i=1}^{q} T_i) &= (dur(T_1), \ldots, dur(T_q)) \end{aligned}$$

From the example of John's sickness above, by using unary operators, it can be received the times when John has started to be sick and has been recovered as

$start(\ldots) =$
($start$($\langle$[around-mid-Jan-97,around-end-Jan-97] , $C(d)\rangle$), $start$($\langle$[around-5-Feb-97,around-10-Feb-97] , $C'(d)\rangle$)).

and

$end(\ldots) =$
($end$($\langle$[around-mid-Jan-97,around-end-Jan-97] , $C(d)\rangle$), $end$($\langle$[around-5-Feb-97,around-10-Feb-97] , $C'(d)\rangle$))

respectively.

### 3.2 The Intersection Operation and The Null Interval

When the intersection operation is not *defined* as mentioned previously, it is necessary to introduce a special interval called *the null interval* in order to handle such a situation. The null interval, denoted by [], is interpreted as the empty set defined on the underlying time line. Thus, its duration or the cardinality of its interpretation is regarded as zero. Since a precise time interval $i$ is the set of time points between the lower and the upper boundaries inclusive. One has

$$\forall i \in \mathcal{I} \cup \{[]\}, i \cup [] = i \text{ and } i \cap [] = []$$

Generally, the semantic of the temporal specification, which is the result of an undefined intersection operation, consists of two kinds of intervals - the null interval and the valid intervals. For the latter, we can characterize the starting point and the ending point of an interval as follows.

Let $T = \langle [s, e], C(d) \rangle$ be the temporal specification resulting from the intersection between two temporal specifications $T_j$ and $T_k$. Assume $C_j(d)$ and $C_k(d)$ be the duration constraints in $T_j$ and $T_k$ and are expressed by $d_{j_l} \leq d \leq d_{j_u}$ and $d_{k_l} \leq d \leq d_{k_u}$, respectively. Then, the components $s$ and $e$ can be characterized by

$$\begin{aligned} \mu_s(n_s) = \max\{ & \sup_{n'_s \leq n_s} \min(\mu_{start(T_j)}(n_s), \mu_{start(T_k)}(n'_s) \mid \\ & \exists n'_e \in \mathbb{N}, \mu_{\Im(T_k)}([n'_s, n'_e]) > 0 \wedge n'_e \geq n_s), \\ & \sup_{n'_s \leq n_s} \min(\mu_{start(T_k)}(n_s), \mu_{start(T_j)}(n'_s) \mid \\ & \exists n'_e \in \mathbb{N}, \mu_{\Im(T_j)}([n'_s, n'_e]) > 0 \wedge n'_e \geq n_s)\} \end{aligned}$$

and

$$\begin{aligned} \mu_e(n_e) = \max\{ & \sup_{n_e \leq n'_e} \min(\mu_{end(T_j)}(n_e), \mu_{end(T_k)}(n'_e) \mid \\ & \exists n'_s \in \mathbb{N}, \mu_{\Im(T_k)}([n'_s, n'_e]) > 0 \wedge n'_s \leq n_e), \\ & \sup_{n_e \leq n'_e} \min(\mu_{end(T_k)}(n_e), \mu_{end(T_j)}(n'_e) \mid \\ & \exists n'_s \in \mathbb{N}, \mu_{\Im(T_j)}([n'_s, n'_e]) > 0 \wedge n'_s \leq n_e)\} \end{aligned}$$

respectively, and the duration constraint $C(d)$ can be expressed by

$$0 \leq d \leq \min(d_{j_u}, d_{k_u})$$

For the null interval, its degree of membership can be evaluated as

$$\begin{aligned} \mu_{\Im(T_j \cap T_k)}([]) &= \max(Poss(end(T_j) \prec start(T_k)), \\ &\qquad Poss(end(T_k) \prec start(T_j)) \\ &= \max\{ \sup_{n_e < n'_s} \min(\mu_{end(T_j)}(n_e), \\ &\qquad \mu_{start(T_k)}(n'_s)), \\ &\qquad \sup_{n'_e < n_s} \min(\mu_{end(T_k)}(n'_e), \\ &\qquad \mu_{start(T_j)}(n_s))\} \end{aligned}$$

# EUFIT '99

7th European Congress on Intelligent Techniques and Soft Computing
Aachen, Germany, September 13-16, 1999

Promenade 9, 52076 Aachen, Germany
Phone: +49/2408/6969
Telefax: +49/2408/94582
E-Mail: eufit@mitgmbh.de
http://www.mitgmbh.de/elite/eufit.html

Dr. Werasak Kurutach
Mahanakorn University of Technology
Department of Computer Engineering
10/51 Cheum-Sampan Rd., Nong Chok

Bangkok 10530
Thailand

Aachen, 21.07.1999

**Telefax: +66 2 988 3687**

**EUFIT '99 - Notification of Acceptance for Presentation**

Dear Dr Kurutach,

Herewith we would like to confirm that your paper entitled:

**On Incomplete Data in Fuzzy Temporal Database**

has been accepted for presentation at the conference. Your paper is scheduled in session CC7 'Fuzzy Data Bases and Information Retrieval' which will take place on September 16, 1999 at 13:30h

We would like to inform you that the acceptance rate of the papers presented at the conference is about 65%. There are two reviewers for each paper

Should you have any further questions on the conference please do not hesitate to contact us. We are looking forward to meeting you in Aachen in September

Best regards,

i.A. D. Sieli On

Karl Lieven
Chairman Organization Committee

It should be emphasized that, in terms of the relational data model, a temporal specification whose semantic interpretation contains the null interval should not be allowed as a value of the temporal attribute in a *base relation*, but only in a *virtual relation*.

Based on the two generalized bitemporal constructors, we obtain their properties as follows.

**Property** Assume $T = \bigwedge_{i=1}^{p} T_i$ and $T' = \bigwedge_{i=1}^{q} T'_i$ be two conjunctive specifications.

- $T \sqcap T' = \bigwedge_{j=1,k=1}^{p,q} (T_j \sqcap T'_k)$ and
- $T \sqcup T' = \bigwedge_{j=1,k=1}^{p,q} (T_j \sqcup T'_k)$

**Property** Let $\mathcal{T}$ be a universal set of temporal specifications. Then $\forall T_i, T_j, T_k \in \mathcal{T}$

- $(T_i \sqcup T_j) \sqcap T_k$ is a conjunctive specification if and only if $T_i \sqcup T_j$ is not *defined*, and
- $T_i \sqcup T_j \sqcap T_k$ is *defined* if and only if $T_i \sqcap T_k$ and $T_j \sqcap T_k$ are *defined*.

## 4 Conclusions

This paper presents two generalized bitemporal constructors which can be employed in a temporal query evaluation based on our previous work. The proposed constructors will give a realistic result with a high expressiveness, that is, if the relational data model is used as a basis for modeling a database, the tuples in the output relation resulting from the bitemporal constructors are *temporally independent*. However, it may need further investigation to make the processing algorithms more simple when parameterized membership functions are used. In addition, there are other useful temporal constructors, DIFFERENCE and COMPLEMENT, which are under our investigation.

## Acknowledgements

We would like to thank The Thailand Research Fund for providing a grant for this work and Prof. H. Takagi for his helpful comments.

## References

[1] Allen J. F., Maintaining knowledge about temporal intervals, *Communications of the ACM*, Vol. 26, No. 11, November 1983, 832-843.

[2] Dutta S., Generalized Events In Temporal Databases, *Proceedings of IEEE Int. Conf. on Data Engineering*, Los Angeles, California, USA, Febuary 1989, 118-125.

[3] Dyreson C. E., *Valid time indeterminacy*, PhD dissertation, Department of Computer Science, The University of Arizona, 1994.

[4] Gadia S. K., The role of temporal elements in temporal databases, *Bulletin of the Technical Committee on Data Engineering*, December 1988.

[5] Kline N., An Update of the Temporal Database Bibliography, *SIGMOD Record*, Vol. 22, No. 4, December 1993, pp. 66-80.

[6] Kurutach W., Modelling Fuzzy Interval-based Temporal Information: A Temporal Database Perspective, *Proceedings of the 4th IEEE Int. Conf. on Fuzzy Systems*, Yokohama, Japan, March 20-24, 1995, pp. 741-748.

[7] Kurutach W., Processing of Binary Temporal Constructors for Fuzzily-Bounded Time Intervals in Temporal Databases, submitted to NAFIPS98, Florida, USA, 1998.

[8] McKenzie L. E. and Snodgrass R. T., Evaluation of Relational Algebras Incorporating the Time Dimension in Databases, *ACM Trans. on Database Systems*, 1991, pp. 501-543.

[9] Nair S. S., *A relational model for incomplete information in temporal databases*, PhD dissertation, Department of Computer Science, Iowa State University, 1993.

[10] Tansel A. U., Clifford J., Gadia S., Jajodia S., Segev A. and Snodgrass R., *Temporal Database: Theory, Design and Implementation*, the Benjamin/Cummings Publishing Company Inc., 1993.

# On Incomplete Data in A Fuzzy Temporal Database Model

Werasak Kurutach
Department of Computer Engineering
Mahanakorn University of Technology
51 Cheum-Sampan Rd., Nong Chok, Bangkok 10530, THAILAND
e-mail: werasak@mut.ac.th

ABSTRACT: This paper presents the concept of incomplete data arising in a fuzzy temporal database model and their approximation. Interactivity among the temporal attributes is important in the process of approximating the unknown values or reducing the space of incompleteness. In addition, the relationship between the current time and incomplete data has been investigated. Semantics constraint in an application domain can be used in enhancing such the relationship so that the incompleteness is less. This study is motivated by our contention that the understanding of these issues is important to evaluating queries in such a database model. However, to emphasise the point, only the temporal attributes are mainly discussed. Our approach is based on the theory of fuzzy set, and a fuzzy temporal data model has been presented as a basic framework for our discussion.

KEYWORDS: Incomplete data, Fuzzy Temporal Data Model, Fuzzy Time, Temporal Data, Database Systems

## 1. INTRODUCTION

Incomplete data have been an important issue in the database literature for around two decades [Codd (1979), Lipski (1979)]. Generally, there is no consensus on the meaning of the term "incomplete data". It may mean either partially known values [Lipski (1979)] or missing values [Codd (1979)]. However, in this work, incomplete data will be used in the sense of missing values of attributes. In this paper, we will discuss issues on the incompleteness aspect of temporal data in an extended relational data model. These temporal data can be fuzzy values. This data model has been proposed by the author to allow a treatment of both time and fuzziness [Kurutach (1998)]. Until recently, temporal data models and fuzzy data models have been researched separately for several years [Petry (1996), Tansel (1993)]. During the last few years, researchers have paid their attention to methods of handling both temporal and fuzzy data [Kurutach (1995), De Tre (1997)]. However, most of them have concentrated on the issues of data models and query operators. It is our contention that incompleteness of fuzzy temporal data is also an important issue to enhance the expressive power of the model, and, additionally, being able to estimate its value will facilitate the capability of providing the meaningful and useful information from databases, especially when the notion of the current time (or "now" [Clifford (1997)]) is concerned.

The paper is organised as follows. Section 2 will provide the concepts of a discrete time model and fuzzy time. These notions will be used as a basis for the discussion in the sequel sections. Section 3 introduces the underlying data model that emphasises fuzziness in temporal data. In section 4, we discuss how incomplete data can arise in temporal knowledge and how to estimate their meanings. Section 5 shows the relationship between the value denoting the current time and incompleteness of data. It also indicates that a semantic constraint of an application can be used to reduce the space of incompleteness, but not be able to remove it. Finally, in the last section, we will conclude this work and discuss further investigation.

## 2. DISCRETE TIME MODEL AND FUZZY TIME: BACKGROUND CONCEPTS

In general, time is a continuous variable and can be represented by real numbers. However, most temporal database works have assumed the discrete time model because of its simplicity and relative ease of implementation. In such a model, the continuous time line is divided into nondecomposable and equivalent-length segments (called *chronons*). Each chronon is viewed as isomorphic to a natural number. Time in this sense is said to be *linearly ordered*. It is also assumed that a single granularity of time is used in defining the nondecomposable unit of time, and hence, a chronon can be denoted by an integer. To assume a discretization of the time axis implies a simplification of its computational handling and avoids the problem derived from differentiating between open and closed intervals.

A time point in the discrete time model can be specified by an integer which is corresponding to a chronon on the time axis. Hence, primarily, the notion of *a fuzzy number* [Kaufmann(1988)] can be employed in modeling a fuzzy time point as well as a fuzzy length of time. Basically, the length (or duration) of time can be regarded as the numbers of chronons from the starting time point to the ending time point inclusive. For computational efficiency and simplicity reasons, *a trapezoidal function* will be used in defining the grade of membership of a fuzzy number. It can be defined as follow.

$$T(x : a, b, c, d) = \begin{cases} 0 & \text{if } x < a \text{ or } x > d, \\ \dfrac{(x-a)}{(b-a)} & a \le x < b, \\ 1 & \text{if } b \le x \le c, \\ \dfrac{(d-x)}{(d-c)} & \text{if } c < x \le d, \end{cases}$$

Assume $M = T(x.\ a_m, b_m, c_m, d_m)$ and $N = T(x\ a_n, b_n, c_n, d_n)$ be two fuzzy number. Then, two basic arithmetic operators can be defined as follows [4].

$$M + N = T(x: a_m + a_n, b_m + b_n, c_m + c_n, d_m + d_n)$$
$$M - N = T(x: a_n - d_m, b_n - c_m, c_n - b_m, d_n - a_m)$$

Also, note that $T(x: c, c, c, c)$ is a precise number $c$, and we will use both notations interchangeably.

## 3. A FUZZY TEMPORAL DATABASE MODEL

Basically, a fuzzy temporal database is defined as a collection of temporal relations (or tables) that can have fuzzy data, and each temporal relation contains tuples of the same types. To emphasise the point of discussion in this paper, a tuple will be viewed as composed of two main parts: *nontemporal* and *temporal* parts. Formally, given a temporal relation $r$, a tuple $h$ in $r$ can be defined by

$$h = <v, t>$$

where $v \in dom(A_1) \times \ldots \times dom(A_n)$ ($dom(A_i)$, $1 \le i \le n$, is the domain of attribute $A_i$) and $t$ is a *temporal specification* [Kurutach (1995)]. The set of $n$ attributes $\{A_1, \ldots, A_n\}$ in the scheme of $r$ are nontemporal ones. In general, the meaning of the tuple $h$ is that the fact represented by the value $v$ is valid in the real world during the time specified by the temporal specification $t$.
Primarily, a temporal specification defines a time interval which may be fuzzy itself. Conventionally, a time interval is represented by its starting time and its ending time, i.e. $[t_{start}, t_{end}]$, and its duration can be evaluated as $t_{end}-t_{start}+1$. However, when the feature of fuzziness is concerned, it has been argued that all three pieces of knowledge of time, the starting time, the ending time and the duration or the length of time, are required for a time interval representation [Kurutach (1995)]. As a consequence, a temporal specification $t$ can be defined as

$$t \in dom(A_{start}) \times dom(A_{end}) \times dom(A_{lower}) \times dom(A_{upper}) \times dom(A_{version})$$

where $A_{start}$ is the starting time attribute, $A_{end}$ is the ending time attribute, $A_{lower}$ is the lower bound attribute, $A_{upper}$ is the upper bound attribute and $A_{version}$ is the reincarnation attribute[1]. The lower and upper bound attributes are used to restrict the duration of the time, and their values are the number of *chronons* in the time interval. The two attributes $A_{start}$ and $A_{end}$ can have the same domain which is the set of fuzzy numbers. Each of those fuzzy numbers is corresponding to a fuzzy set of smallest units defined on the time axis.
In the conventional data model, the concept of a key has been used to uniquely identify a tuple in a relation or, correspondingly, an object in the real world [Elmasri (1994)]. In what follows, the notion of a key in a fuzzy temporal data model will be defined so that this traditional property of the key is to be maintained.

Let $R = \{A_1, A_2, \ldots, A_n\} \cup \{A_{start}, A_{end}, A_{lower}, A_{upper}, A_{version}\}$ denote the schema of the relation $r$ above. A set of attributes $K \subseteq \{A_1, A_2, \ldots, A_n\}$ is said to be *a time invariant key* if and only if

[1] The concept of this attribute will be made clear later.

$$\forall h, g \in r_x \text{ and } h \neq g, \qquad h[K] \neq g[K]$$

, where $r_x$ is the time slid relation of $r$ at a time point $x$. The value of K uniquely identify an object in the real world, but not a tuple in a relation $r$. Therefore, it cannot be a key of the relation. In fact, there could be more than one tuple in $r$ that describe one object but at different states. Consequently, a key of the relation must consist of a time invariant key and a temporal attribute. When the temporal part is considered, it is obvious that neither of $A_{start}$, $A_{end}$, $A_{lower}$ and $A_{upper}$ can be a key attribute. This is because their values can be fuzzy. Fortunately, a value of the attribute $A_{version}$ of an object indicates the number of times that the object changed its state. In the other words, it will be updated or increased by 1 every time an attribute value of the object has been changed. The value of $A_{version}$ must be automatically generated by the system and cannot be modified by any user. Therefore, it is obvious that

$$\forall h, g \in r \text{ and } h \neq g, \qquad h[K][A_{version}] \neq g[K][A_{version}]$$

That means the set of attributes $K \cup \{A_{version}\}$ can be defined as a key of the fuzzy temporal relation.

## 4. INCOMPLETENESS OF TEMPORAL DATA

Primarily, fuzziness can be represented in the structure of a temporal specification as discussed in the previous section. However, in the real world, some part of temporal knowledge (corresponding to some component of the temporal specification) may not be known. This unknown information results in what is called *incompleteness* of temporal knowledge. There are fifteen possible cases of incomplete data in a temporal specification. These cases arise in only four temporal attributes (i.e. $A_{start}$, $A_{end}$, $A_{lower}$ and $A_{upper}$) in combination. Incompleteness cannot appear in the attribute $A_{version}$, because its value is generated by the system. However, in those cases, any unknown component can be estimated by using the knowledge of the known components. This estimation is important because it reduces the scope or the space of possible values of the unknown components. In what follow, each case will be shown with the symbol '*' to denote the unknown data of an attribute.

Case 1 $h = \langle v, \langle *, e, d_l, d_u, version\# \rangle\rangle$

In this case, the starting time point where the value $v$ becomes valid is not known. However, the estimation of the unknown starting time point can be achieved by using the other known temporal attributes as follows

$$\mu_{h[A_{start}]}(x) = \sup_{x=y-z+1} \min(\mu_e(y), \mu_{d_1 \tilde{\cup} d_2}(z))$$

where $\mu_{d_1 \bar{\cup} d_2}(z)$ is defined by

$$\mu_{d_1 \bar{\cup} d_2}(z) = \sup\{\min(\mu_{d_1 \cup d_2}(x), \mu_{d_1 \cup d_2}(y)) \mid x \leq z \leq y\}$$

That is, in general,

$$h[A_{start}] = h[A_{end}] - (h[A_{lower}] \tilde{\cup} h[A_{upper}]) + 1$$

Case 2 $h = \langle v, \langle s, *, d_l, d_u, version\# \rangle\rangle$

In this case, the ending time point where the value $v$ becomes invalid. However, the estimation of the unknown ending time point can be achieved by using the other known temporal attributes as follows

$$\mu_{h[A_{end}]}(x) = \sup_{x=y+z-1} \min(\mu_s(y), \mu_{d_1 \tilde{\cup} d_2}(z))$$

That is, in general,

$$h[A_{end}] = h[A_{start}] + (h[A_{lower}] \tilde{\cup} h[A_{upper}]) - 1$$

Case 3 $h = \langle v, \langle s, e, *, d_u, version\# \rangle\rangle$

In this circumstance, the value of the lower bound attribute is not known. Generally, the difference between $s$ and $e$ is the duration which must lie between * and $d_u$. That means

$$* \leq (e - s + 1) \leq d_u$$

Therefore, there are two extreme values, 1 and $(e - s + 1)$, that can be chosen as the estimated value of $h[A_{lower}]$. If the latter is selected, the scope of incompleteness of the lower bound attribute is smaller. Hence, the value of the unknown lower bound attribute can be

$$h[A_{lower}] = h[A_{end}] - h[A_{start}] + 1$$

Case 4 $h = < v, <s, e, d_l, *, version\# >>$
This is similar to Case 3, except that the two extreme values of $h[A_{upper}]$ are $(e - s + 1)$ and $u$, where $u$ is the upper end of the time axis. The former should be chosen for the same reason as in Case 3. Thus, the value of the unkown upper bound attribute can be

$$h[A_{upper}] = h[A_{end}] - h[A_{start}] + 1$$

Case 5 $h = < v, <s, e, *_l, *_u, version\# >>$
When the values of the two bound attributes are unknown, they can be estimated with the same values as follows.

$$h[lower] = h[A_{upper}] = h[A_{end}] - h[A_{start}] + 1$$

Case 6 $h = < v, <*, *, d_l, d_u, version\# >>$
When we have only knowledge of the two bound attributes, the values of the starting time point and the ending time point can be evaluated by

$$h[A_{start}] = T[x : 0,0,0,0] \cup (T[x : u,u,u,u] - h[A_{lower}] + 1)$$

$$h[A_{end}] = (T[x : 0,0,0,0] + h[A_{lower}] - 1) \cup h[A_{lower}]$$

Case 7 $h = < v, <*, e, *_l, d_u, version\# >>$
Basically, the value $(e - d_u)$ mean the starting time point of the time interval with the ending time point $e$ and the duration $d_u$. However, any time point cannot precede the starting time point of the time axis (i.e. 0). Therefore, the unknown value of attribute $A_{start}$ can be evaluated as

$$\mu_{h[A_{start}]}(x) = \begin{cases} 0 & \text{if } x < 0, \\ \mu_{(h[A_{end}] - h[A_{upper}] + 1)}(x) & \text{otherwise} \end{cases}$$

, and the unknown value of attribute $A_{lower}$ is

$$h[A_{lower}] = 1$$

Because of the space limitations, we will leave Case 8 $h = < v, <*, e, d_l, *, version\# >>$, Case 9 $h = < v, <s, *, *, d_u, version\# >>$ and Case 10 $h = < v, <s, *, d_l, *, version\# >>$ here. They can be evaluated using a similar approach to Case 7.

Case 11 $h = < v, <s, *, *, *, version\# >>$

$$h[A_{end}] = h[A_{start}] \bar{\cup} T[x : u,u,u,u]$$

$$h[A_{lower}] = 1 \text{ and } h[A_{upper}] = T[x : u,u,u,u] - h[A_{start}] + 1$$

Case 12 $h = < v, <*, e, *, *, version\# >>$

$$h[A_{start}] = h[A_{end}]\bar{\cup}T[x:0,0,0,0]$$

$$h[A_{lower}] = 1 \text{ and } h[A_{upper}] = h[A_{end}] - T[x:u,u,u,u] + 1$$

Case 13 $h = < v, <*, *, d_l, *, version\# >>$

$$h[A_{start}] = T[x:0,0,0,0]\bar{\cup}(T[x:u,u,u,u] - h[A_{lower}]) + 1$$

$$h[A_{end}] = T[x:u,u,u,u]\bar{\cup}(T[x:0,0,0,0] + h[A_{lower}] - 1)$$

$$h[A_{lower}] = u$$

Case 14 $h = < v, <*, *, *, d_u, version\# >>$

$$h[A_{start}] = h[A_{end}] = T(x:0,0,u,u)$$

$$h[A_{lower}] = 1$$

Case 15 $h = < v, <*, *, *, *, version\# >>$

This is the case of total lack of temporal knowledge. Therefore, the space of incompleteness cannot be reduced. That is,

$$h[A_{start}] = h[A_{end}] = T(x:0,0,u,u)$$

$$h[A_{lower}] = 1 \text{ and } h[A_{upper}] = u$$

## 5. SEMANTICS OF THE CURRENT TIME

The current time is a moving time. The value *now* has been employed to denote such a time in temporal databases, and it can be considered as a time variable [Clifford (1997)]. The validity of time-varying information sometimes depends on the current-time value. For example, assume that the current time is June 7, 1999 and the granularity of time is *day*. Then, a tuple

$$h = \text{<<John, Manager, 50K>, <1 June 9, } now\text{, 7, 7, 1>>}$$

means the time that John is a manager and has an income of 50K per annum starts from June 1, 1999 up until now. The value 1 of the last attribute shows that this is the first state of the object. That is, John started working as a manager with the salary 50K.

From the above, *now* is used to indicate that a fact is valid until the current time and *the future time of its validity is currently unknown*. In the other words, incomplete data is inherent in the notion of using *now* as a time variable denoting the current time. Another problem is that the lower and upper bounds of the time duration will encounter the difficulty of being updated when the ending time is moving. The problems can be treated into two situations.

Firstly, provided that the starting time $s$ is certainly before the current time *now*, the fact can be simply represented as

$$h[A_{start}] = h[A_{end}] - (h[A_{lower}]\bar{\cup}h[A_{upper}]) + 1$$

$$h = <v, <s, e, *, *>>$$

and the ending time $e$ is specified by

$$e = T(n, now + \Delta d, now + \Delta d, u, u)$$

where $s\text{-}now \leq \Delta d \leq u\text{-}now$. $\Delta d$ is called an *offset span* whose semantics depending on applications [Jensen (1994)]. For example, in a database of the employment history, it may be a policy of the company that if an employee is either promoted or demoted, he/she must be noticed before its effective by 30 days. In this situation, the value of $\Delta d$ is 30. Secondly, provided that the starting time $s$ is not certainly before the present time, the ending time $e$ is specified by

$$e = T(n, s + \lfloor\Delta d\rfloor, s + \lceil\Delta d\rceil, u, u)$$

For example, assume that today is August 15, 1999, if the company employs a new employee who will start working on September 1, 1999, then the temporal knowledge of the employment status of the new employee will be

$$t = <1 \text{ Sept } 1999, e, *, *>$$

where $e$ is a fuzzy time point characterized by

$$e = T(n, 1 \text{ Sept } 1999 + 14, 1 \text{ Sept } 1999 + 14, u, u)$$

## 6. CONCLUSION

In this work, we have proposed a fuzzy temporal data model based on a discrete time model. This data model has been employed as the framework to discussing issues on incomplete data in fuzzy temporal knowledge. We have shown that the space of incompleteness of data can be reduced by using interactivity among temporal attributes. Moreover, when the current time is concerned and the variable *now* is employed, incompleteness is inherent in its representation. Also, a semantic constraint of the application can be used to enhance such completeness.

Acknowledgement. This work has been supported by a grant from The Thailand Research Fund (PDF/50 ?540).

## REFERENCES

Clifford J., Dyreson C., Isakowitz T., Jensen C. S. and Snodgrass R., "On the Semantics of "Now" in Databases", ACM Transactions on Database Systems, Vol. 22, No. 2, June 1997, pp. 171-214.
Codd E. F. Extending the Database Relational Model to Capture More Meaning, ACM Trans. on Databases, Vol. 4, No. 4, Dec. 1979, pp. 397-434.
De Tre G., De Caluwe R., Van der Cruyssen B., Van Gyseghem N., Toward Temporal Fuzzy and Uncertain Object-Oriented Database Management Systems, NAFIPS97, pp. 63-67.
Jensen C. S. and Snodgrass R., Temporal Specialization and Generalization, IEEE Transactions on Knowledge and Data Engineering, Vol. 6, No. 6, December 1994, pp. 954-974.
Kaufmann A. and Gupta M. M., Introduction to Fuzzy Arithmetic, Van Nostrand Reinhold Inc., 1988.
Kurutach W., Modeling Fuzzy Interval-based Temporal Information: A Temporal Database Perspective, Proc. of the 4th IEEE Int. Conf. on Fuzzy Systems, Yokohama, Japan, March 1995, pp. 741-748.
Kurutach W., Handling Fuzziness in Temporal Databases, Proc. of IEEE Int. Conf. on SMC, 1998, pp.
Lipski W., On Semantic Issues Connected with Incomplete Information Databases, ACM Transactions on Databases, Vol. 4, No. 3, Sept. 1979, pp. 262-296.
Elmasri R. and Navathe S. B., Fundamentals of Database Systems, The Benjamin/Cummings Publishing Company, Inc., 1994.
Petry F. E., Fuzzy Databases: Principles and Applications, Kluwer Academic Publishers, 1996.
Tansel A. U., Clifford J., Gadia S., Jajodia S., Segev A. and Snodgrass S., Temporal Databases: Theory, Design and Implementation, The Benjamin/Cummings Publishing Company, 1993.