ITRM: A Temporal Data Model Supporting Indeterminacy

Jia-Dong Ren, Hui-Li Peng and Xiao-Jian Zhang †,

†Yanshan University, College of Information Science and Engineering, Qinhuangdao, China

Summary
The depiction of indeterminate temporal and atemporal data is important to obtain a suitable model of database management systems. This paper presents a formal relation model, called ITRM (Indeterminate Temporal Relation Model), for the representation, manipulation and query of indeterminate information. Moreover, indeterminate time interval is classified and temporal relation algebra operators are extended. All the operations of the algebra are defined and are proved to be closed. The given examples show that the ITRM model can efficiently represent indeterminate data.

Key words:
Indeterminacy; ITRM; Time interval; Temporal relational algebra operator

1. Introduction

The need of supporting valid-time in database area is of major importance since information that changes over time must also be recorded. Temporal database is a database in which temporal data are associated to atemporal data. These temporal data decide the valid time, they specify the time when the information holds in real world modeled by the database [1]. Temporal data as well as atemporal data may be determinate or indeterminate. For atemporal data, this corresponds to cases where "know accurately what" or "don't know accurately what". While for temporal data, the meaning is "know accurately when" or "don't know accurately when".

Many complex applications such as the history data of stock market in finance and economics system, patient data record in medical system, decision support system and geographic information system are all need handling temporal data and this information frequently can't be totally determined [2].

However, current database management systems can't afford to represent, store and obtain the indeterminate temporal and atemporal information. Most of the existing approaches dealing with uncertain temporal database are concentrated on the theory of fuzzy set, probability theory and null value [3].

In this paper, the main object is to show a formal relation model, named ITRM, which supports determinate and indeterminate information. More precisely, we introduce ITRM model that allows represent, manipulate and query indeterminate temporal data, definite indeterminate temporal primitive and relation algebra operations.

The remainder of the paper is organized as follows. Section 2 is related work. In Section 3 the statements of indeterminate data and the definition of ITRM model are presented. The extended algebra operators are presented and the property of the operations is proved in Section 4 with appropriate query examples. Finally, conclusion is presented in Section 5.

2. Related work

The research of indeterminate information can be divided two parts: indeterminate temporal information and indeterminate atemporal information. The latter one indicates the non-temporal attributes in temporal models, that is, the attributes which don't change with time.

For atemporal indeterminacy, [4] presented a probability relational data model based on NF2, and defined new operators.

Researches about indeterminate temporal information exist since the 1990's [5]. However, Snodgrass and Dyreson's method is the extension of probabilistic database model (PDM), which restricted to probability theory for indeterminate temporal attributes.

M.Kamil describes an approach to representing uncertainly specified historical time periods and implementing the prototype [6]. In [7] an indeterminate interval is represented by an ordered pair of endpoints and a probability distribution is associated to each set of the possible endpoints, but this probability distribution is unknown in many situations. Based on it, [8] assumption that the probability distribution associated to a set of possible endpoints is unknown, so the restriction on the beginning and the ending points of an indefinite interval can be relaxed. However, it only restricted certain kinds of indefinite temporal interval, not using it for modeling. Jarulog C presented three methods for the representation of indeterminate intervals [9], extended existing operators and presented new operators deal with indeterminate valid-time.

Manuscript received September 5, 2006.
Manuscript revised September 25, 2006.

3. ITRM: a model supporting indeterminacy

In this section, we describe the related knowledge of temporal primitive of ITRM and give its definition.

3.1 The definition of indeterminate time interval

We identify time interval as the basic specification; it is duration between two specific time points which stand for the lower and upper bounds of the interval. Instant is a specific anchored time moment. For example, interval [15, 20] is represented by instants 15 and 20. In this paper we focus on time intervals, which consist of indeterminate intervals and determinate intervals.

Definition 1 Determinate interval (*di*): Determinate interval is the finite set of consecutive definite time points, that is to say $di = [t_1, t_2] = \{t_k \mid t_k \in T, t_1 \le t_k \le t_2\}$.

Definition 2 Indeterminate interval (*ii*): The beginning and/or ending point are indeterminate can be called indeterminate interval, denoted as *ii*=[s, e]. It can be divided into absolute indeterminate interval and relative indeterminate interval.

(1) For absolute specification, there are several sorts of situations:

The first situation is that one endpoint is indeterminate, such that [s*, e] or [s, *e]. The endpoint with star notation (*) denotes indeterminate on that endpoint. For example, ([19*, 20]) denotes valid beginning time of an event is 19 or after 19 and ending time is 20.

The second one is two endpoints are both indeterminate, such that [s*, *e]. For instance, [95*, *100] indicate that the beginning point is after 95 and the ending point is before 100.

Notice that we do not consider indeterminate interval [*s, e] and [s, e*], because they can be transformed into the union of [s, e], [0, *s] or [e*, NOW]. In time domain *T*, equation [*s, e] = [0, *s] $\cup$ [s+1, e] and [s, e*]=[s, e−1] $\cup$ [s*, NOW] are always true.

The last one is one of the endpoints is unknown, such that $[?^-, c]$ or $[c, ?^+]$。 Denotation "$?^-$"and "$?^+$" denote the beginning point and the ending point is unknown separately. For instance, an event occurred before time 34 can be represented as $[?^-, 34]$.

(2) Statements like [*t*] can be evaluated as relative indeterminate interval. Such as "about 2002", "around 3 P.M" are typical examples

3.2 Operations between indeterminate time intervals

In this section the definitions of arithmetic operations between indeterminate time intervals, including overlap, union and difference, are given.

3.2.1 Indeterminate overlap operator ($\cap_{\mathbf{ii}}$)

The expression $s_1 \cap_{\mathbf{ii}} s_2$ return the set of valid time that are contained or possibly contained in the valid time of s_1 and s_2. The indefiniteness of this operator can be further extended to allow the entire interval to vanish, denoted as [<>]. For example the possible values for [<10, *15>] are [10, 10], [10, 11], [10, 12], [10, 13], [10, 14], [10, 15] or NULL.

Let s_1= [ss_1, se_2], s_2= [ss_2, se_2] are two indefinite intervals, [ss_1*, se_1] $\cap_{\mathbf{ii}}$ [ss_2, *se_2] =

$$\begin{cases} \text{NULL} & se_1 < ss_2 \text{ or } se_2 < ss_1 \\ \{[<\text{Larger}(ss_1, ss_2)^*, {}^*\text{Smaller}(se_1, se_2)>]\} & \text{else} \end{cases}$$

For example: [1990*, 1998] $\cap_{\mathbf{ii}}$ [1992,*1997] = {[1992*, *1997]}

Function Larger () returns the larger value of the two parameters, Smaller () returns the smaller one of the two. For example, [6, 10] $\cap_{\mathbf{ii}}$ [9, 12] = {[9, 10]}. In particular, Larger () =Smaller () returns the set of one point. For instance, [15*, 20] $\cap_{\mathbf{ii}}$ [20,*25] = {20}.

3.2.2 Indeterminate union operator ($\cup_{\mathbf{ii}}$)

The expression $s_1 \cup_{\mathbf{ii}} s_2$ returns the union or possible union between s_1 and s_2, as follows.

[ss_1*, se_1] UNION_T [ss_2*, se_2] =

$$\begin{cases} \{s_1, s_2\} & se_1 < ss_2 \text{ or } se_2 < ss_1 \\ \{[\text{Smaller}(ss_1, ss_2)^*\text{Smaller}(se_1, se_2), \text{Larger}(se_1, se_2)]\} & \text{else} \end{cases}$$

For example: [1990*, 1998] $\cup_{\mathbf{ii}}$ [1992, *1997] = {[1990*1992, 1998]} ;

3.2.3 Indeterminate difference operator ($-_{\mathbf{ii}}$)

The expression $s_1 -_{\mathbf{ii}} s_2$ returns the value which belongs to s_1 but not s_2.

[ss_1*, se_1] $-_{\mathbf{ii}}$ [ss_2*, se_2] =

$$\begin{cases} \{s_1\} & se_1 < ss_2 \text{ or } se2 < ss_1 \\ \{[ss_1^*, ss_2 - 1], [ss_2^* se_2, se_1]\} & ss_1 < ss_2 < se_2 < se_1 \\ \text{NULL} & ss_2 < ss_1 < se_1 < se_2 \\ \{[ss_1^*, ss_2 - 1]\} & \text{else} \end{cases}$$

For example: [1990*, 1997] $-_{\mathbf{ii}}$ [1992, *1999] = {[1990*, 1991]}.

3.3 Null value

One kind of indeterminate data is that the valid time is known to be one of a specific set or range of values, as is shown in Section 3.1. Another kind of indeterminacy is the case for which the valid value of temporal (atemporal) data is not known (abbreviated as *unk*), does not exist (abbreviated as *dne*), or there is no information on whether temporal (atemporal) data exists or not (abbreviated as *ni*). All of these can be considered as a sort of null value. For instance, if we know an event happened actually but we don't know when it happened, then it is *unk*.

3.4 ITRM model

Here, we present ITRM model to represent indeterminate data in a database environment. The attribute of temporal relational schema consists of time-varying attributes and non-time-varying ones. As far as time-varying attributes concerned, using binary (t, v) represents attribute value v is valid in $t = [t_1, t_2]$.

3.4.1 Definition of ITRM

Definition 3 Let a relation R have schema *SchR* which is a collection of rules of the form *SchR*: (A_1, A_2, …, A_n). Each attribute A_j may be atom or temporal atom (including determinate and indeterminate) or set of atom or set of temporal atom which are defined below. $D = \{D_1, D_2, \ldots, D_n\}$ is a finite set of domains. Let r, an instance of R, be composed of a set of ordered k-tuples of the form $<a_1, a_2, \ldots a_k>$, which is a subset of $\{D_1, D_2, \ldots, D_n\}$. The domains D_i ($1 \le i \le k$), can be one of the following:

(1) D_i is the domain of an atomic-valued attribute. Each value a_j is an element of D_i; that is, it is a typical simple crisp attribute value.
(2) D_i is the domain of determinate temporal atomic-valued attribute. Each value is an ordered pair (t, v), where $t \subseteq T, v \subseteq U$.
(3) D_i is the domain of indeterminate temporal atomic-valued attribute. Each value is an ordered pair (t, v), where $t \subseteq T$, $v \subseteq U$, denoted as [s*, e], [s, *e], [s*, *e], [$?^-$, c] or [c, $?^+$].
(4) D_i is the domain of a null-valued attribute. $D_i = \{unk, dne, ni\}$.
(5) D_i is the domain of set-valued attribute whose values are crisp sets represented as $\{a_{i1}, a_{i2}, \ldots, a_{im}\}$. Any value of this attribute is a subset of the power set of D_i.
(6) D_i is the domain of temporal set-valued attribute (including determinate and indeterminate attribute) whose values are sets represented as $\{(t_1, v_1), \ldots, (t_n, v_n)\}$. Any value of the attribute is a subset of the power set of D_i.

Examples of the various attribute types discussed above are represented in ITRM, as shown in Table 1. SSN (Social Security Number), TelNo and BIR (Birthday) are all atomic attributes; ADDRESS is temporal set-valued attribute; HI (History Information) is temporal atomic attribute, which is used to record the history information of employees. For example, one employee whose SSN=125 once worked in C1 company between [1995*, *1998], but the detailing time is not clear

Table 1: EMP relations with indeterminate valid time

SSN	TelNo	ADDRESS	BIR	HI
125	892561	{([1990*, 1998], A1) ([1998, NOW], A2)}	1970	([1995*, *1998], C1)
200	*ni*	{([1992, *1997], A1) ([1997, 2003], A2)}	*unk*	([*2000*], C2)
410	*dne*	([?-, 2002], A2)	1978	*ni*

3.4.2 Constraint satisfaction

Some general constraints should be satisfied, when working with indeterminate temporal data. Take predicate "before" as an example, the property of transitivity of functions as follows.

$$t1 \text{ before } t2 \wedge t2 \text{ before } t3 \rightarrow t1 \text{ before } t3$$

According to [9], besides above constraint, there are another two constraints should be satisfied by ITRM.

The first one agrees with the situation that on endpoint of the indeterminate duration is completely known. Every constraint is characterized by an ordered list which defined on a set of continuous possible points. A fixed-endpoint constraint indicates that if a possible point turns out to be a factual point of the event, then all points before it in the ordered list must also be actual points. Take interval [2000, *2005] as an example, the ordered list is {2001, 2002, 2003, 2004}, if 2002 becomes a determinate time, then 2001 is a determinate one, too.

The second one agrees with the situation when none of the endpoints of the indeterminate duration is completely known. Every such constraint is described by a range $[t_1,\ldots,t_2]$, A subinterval constraint $[t_1,\ldots,t_2]$ states that the actual duration of the event can be null or any subinterval inside the range specified by the boundary points t_1 to t_2. For example, the possible set of [1998*, *2005] is T = {1998,…, 2005}, the actual value is any non-empty subset of T.

4. Primitive Algebraic Temporal Operators Based ITRM

Let r and q be two temporal relations with the same relational scheme R, which both contain indeterminate time. Let Attr (A_a) be all the atomic-valued attributes, Attr (A_{di}) all the determinate temporal atomic-valued attributes, Attr (A_{ii}) all the indeterminate temporal atomic-valued attributes, Attr (A_e) the empty value attributes, Attr (S) the set of all the atomic attributes, and Attr (TS) be all the set of temporal atomic attributes. Let t_r be a tuple in relation r, t_q be a tuple in relation q and t a tuple in the resulting relation. In the following, we use $\cap_{\mathbf{ii}}$, $\cup_{\mathbf{ii}}$ and $-_{\mathbf{ii}}$ denote indeterminacy, $\cap_{\mathbf{di}}$, $\cup_{\mathbf{di}}$ and $-_{\mathbf{di}}$ represent determinacy. The following definitions are extension of [8], which is not including indeterminacy.

4.1 The indeterminate temporal union operator

The union of two temporal relations r and q, $r \cup_t q$, is a new temporal relation with identical headings to relations r and q, consisting of all tuples appearing in either or both of the two relations, and the temporal elements for all the temporal attributes are computed by taking the unions of the temporal elements of the corresponding temporal attributes of the two relations r and q.

$r \cup_t q = \{t|\ ((\exists t_r \in r)\ (\exists t_q \in q)$
$((t\ [\text{Attr}\ (A_a)] = t_r\ [\text{Attr}\ (A_a)] = t_q[\text{Attr}\ (A_a)])$
$\wedge\ (t\ [\text{Attr}\ (A_{di})] = t_r[\text{Attr}\ (A_{di})] \cup_{di} t_q[\text{Attr}\ (A_{di})])$
$\wedge\ (t\ [\text{Attr}\ (A_{ii})] = t_r\ [\text{Attr}\ (A_{ii})] \cup_{\mathbf{ii}} t_q\ [\text{Attr}\ (A_{ii})]))$
$\wedge\ ((t\ [\text{Attr}\ (S)] = t_r\ [\text{Attr}\ (S)] = t_q[\text{Attr}\ (S)])$
$\wedge\ ((t\ [\text{Attr}\ (TS)] = t_r\ [\text{Attr}\ (TS)] \cup_{di} t_q[\text{Attr}\ (TS)])$
$\wedge\ (t\ [\text{Attr}\ (TS)] = t_r\ [\text{Attr}\ (TS)] \cup_{\mathbf{ii}} t_q[\text{Attr}\ (TS)]))$
$\vee\ ((\exists t_r \in r)\ ((t\ [\text{Attr}\ (A_a)] = t_r[\text{Attr}\ (A_a)])$
$\wedge\ (t\ [\text{Attr}\ (A_{di})] = t_r[\text{Attr}\ (A_{di})])$
$\wedge\ (t\ [\text{Attr}\ (A_{ii})] = t_r[\text{Attr}\ (A_{ii})])\ \wedge\ (t\ [\text{Attr}\ (S)] = t_r[\text{Attr}\ (S)])$
$\wedge\ (t\ [\text{Attr}\ (TS)] = t_r\ [\text{Attr}\ (TS)])))$
$\vee\ ((\exists t_q \in q)\ ((t\ [\text{Attr}\ (A_a)] = t_q\ [\text{Attr}\ (A_a)])$
$\wedge\ (t\ [\text{Attr}\ (A_{di})] = t_q\ [\text{Attr}\ (A_{di})])$
$\wedge\ (t\ [\text{Attr}\ (A_{ii})] = t_q[\text{Attr}\ (A_{ii})])\ \wedge\ (t\ [\text{Attr}\ (S)] = t_r[\text{Attr}\ (S)])$
$\wedge\ (t\ [\text{Attr}\ (TS)] = t_r\ [\text{Attr}\ (TS)]))))$

4.2 The indeterminate temporal intersection operator

The intersection of two temporal relations r and q, $r \cap_t q$, is a new temporal relation with identical heading to relations r and q, consisting of all tuples appearing in both of the relations r and q, the temporal elements for all the temporal attributes are computed by taking the intersections of the temporal elements of the corresponding temporal attributes of the two relations r and q. In the resulting relation, tuples having empty temporal elements must be discarded. Then, the temporal intersection of the two relations r and q is defined as follows.

$r \cap_t q = \{t|\ (\exists t_r \in r)\ (\exists t_q \in q)$
$((t\ [\text{Attr}\ (A_a)] = t_r[\text{Attr}\ (A_a)] = t_q[\text{Attr}\ (A_a)])$
$\wedge\ (t\ [\text{Attr}\ (S)] = t_r[\text{Attr}(S)] = t_q\ [\text{Attr}\ (S)])$
$\wedge\ (t[\text{Attr}(A_{di})] = (t_r[\text{Attr}(A_{di})$

4.3 The indeterminate temporal difference operator

The difference of two temporal relations r and q, $r -_t q$, a new temporal relation with identical headings to relations r and q, consisting of all tuples appearing in relation r but not in relation q. The temporal elements for all the temporal attributes are computed by taking the differences of the temporal elements of the corresponding temporal attributes of the two relations r and q. In the resulting relation, tuples having empty temporal elements must be discarded. Then, the temporal difference of the two relations r and q can be formally defined as follows.

$r -_t q = \{t|\ ((\exists t_r \in r)\ (\forall t_q \in q)((t\ [\text{Attr}\ (A_a)] = t_r[\text{Attr}\ (A_a)])$
$\wedge\ (t\ [\text{Attr}(S)] = t_r[\text{Attr}\ (S)])\ \wedge\ ((t\ [\text{Attr}(A_a)] \neq t_q[\text{Attr}\ (A_a)])$
$\wedge\ (t\ [\text{Attr}\ (A_{di})] = t_r[\text{Attr}\ (A_{di})])$
$\wedge\ (t\ [\text{Attr}\ (A_{ii})] = t_r[\text{Attr}\ (A_{ii})])$
$\wedge\ (t\ [\text{Attr}\ (TS)] = t_r[\text{Attr}\ (TS)])))$
$\vee\ ((\exists t_r \in r)\ (\exists t_q \in q)\ ((t[\text{Attr}(A_a)] = t_r[\text{Attr}(A_a)] = t_q[\text{Attr}$ $(A_a)])\ \wedge\ (t\ [\text{Attr}(S)] = t_r[\text{Attr}(S)] = t_q[\text{Attr}\ (S)])$
$\wedge\ ((t[\text{Attr}\ (A_{di})] = (t_r[\text{Attr}\ (A_{di})] -_{di} t_q[\text{Attr}\ (A_{di})]) \neq \phi\ )$
$\vee\ (t[\text{Attr}\ (A_{di})] = (t_r[\text{Attr}\ (A_{ii})] -_{\mathbf{ii}} t_q[\text{Attr}\ (A_{ii})]) \neq \phi\ )$
$\wedge\ ((t[\text{Attr}\ (TS)] = (t_r[\text{Attr}\ (TS)] -_{di} t_q[\text{Attr}\ (TS)]) \neq \phi\ )$
$\vee\ (t[\text{Attr}\ (TS)] = (t_r[\text{Attr}\ (TS)] -_{\mathbf{ii}} t_q[\text{Attr}\ (TS)]) \neq \phi\ )))))$

4.4 The indeterminate temporal timeslice operator

Temporal projection can't be used to answer a relation along a given temporal element. So a new operation named Timeslice, $TS\ (t\ [A_t])$, which takes the intersection of the given temporal element and each temporal element of the relation, needs to be defined. In the resulting relation, tuples having empty temporal elements are not considered.

$TS\ (t\ [A_t]) = \{t|\ (\exists t_r \in r)\ ((t\ [\text{Attr}\ (A_a)] = t_r\ [\text{Attr}\ (A_a)])$
$\wedge\ ((t\ [\text{Attr}\ (A_{di})] = tr[\text{Attr}(A_{di})] \cap_{di} di) \neq \phi\ )$
$\vee\ ((t\ [\text{Attr}\ (A_{ii})] = tr\ [\text{Attr}\ (A_{ii})] \cap_{\mathbf{ii}} ii \neq \phi\ )))\}$

4.5 Closure property of the temporal relational algebra operations

In this section, temporal operations are proved to be closed in U, where U is the underlying domain of the temporal relations. For all the following propositions, let r be a temporal relation with relation scheme Sch R : (A_1, A_2,…, A_n), where Attr (A_a) = $\{A_{a1}, A_{a2}, \ldots, A_{am}\}$ ($m \geq 0$) is the set of all atomic attributes. Attr (A_{di}) = $\{A_{di1}, A_{di2}, \ldots, A_{din}\}$ ($n \geq 0$) is the set of all determinate temporal attributes. Attr (A_{ii}) = $\{A_{ii1}, A_{ii2}, \ldots, A_{iin}\}$ is the set of all indeterminate temporal attributes. Attr(S) = $\{S_1, S_2, \ldots, S_q\}$ ($q \geq 0$) is the atomic-set attributes. Attr (TS) = $\{TS_1, TS_2, \ldots, TS_{tq}\}$ is the temporal atomic-set attributes. Let D_{ai} be the underlying domain of the atomic attribute A_{ai} ($0 \leq i \leq m$), D_{tp} be the underlying domain of the atomic attribute A_{tp} ($0 \leq p \leq n$), D_{sk} be the underlying domain of the atomic attribute A_{sk} ($0 \leq k \leq q$), then the underlying domain of relation r is $D_{a1} \times D_{a2} \times \ldots \times D_{am} \times D_{t1} \times D_{t2} \times \ldots \times D_{tn} \times D_{s1} \times D_{s2} \times \ldots \times D_{sn}$.

Proposition 1 Temporal union operation $\cup_t$ is closed in U.

Proof. Let q be a temporal relation with the same relation scheme and the same underlying domain as relation *r*.

Then, from the definition of $\cup_t$ we can see that, underlying domain of relation $s=r\cup_t q$ is $D_{a1}\times D_{a2}\times\ldots\times D_{am}\times D_{t1}\times D_{t2}\times\ldots\times D_{tn}\times D_{s1}\times D_{s2}\times\ldots\times D_{sn}$, which is the same with *r*. So, the output of $\cup_t$ is a temporal relation with the same scheme and the same underlying domain as the input relation *r* and *q*. Thus, the $\cup_t$ is closed in *U*. The situation of $\cap_t$ and $-_t$ are the same with $\cup_t$.

Proposition 2 The Timeslice operation *TS* (*t* [A_t]) is closed in *U*.

Proof The output of the Timeslice operation of temporal relation *r* is a temporal relation whose underlying domain is the underlying domain of relation *r*. So the Timeslice operation is closed in *U*.

4.6 Combine indeterminate valid-time in query languages

The ITRM model for indeterminate valid-time can be used to enhance user interfaces. Examples of these queries include acquiring indeterminate and determinate time intervals which meet given temporal constrains, acquiring query results that are established to meet condition or possibly meet this condition, finding the time when a given condition has no possible of being satisfied.

Take Table1 as an example, some queries can be determinate, others are not. The query answering of the employee' SSN which lived in A2 before 2003 are 125 and 200. But the result of querying the employee's SSN which lived in A2 before 1997 is indeterminate.

The following query retrieves the SSN of employees with possible or certain worked in C1 when they lived in A1.

SELECT SSN
FROM EMP
WHERE (ADDRESS='A1' OVERLAP$_T$ HI =*'C1')
The answer is SSN＝125。

5. Conclusio

This paper proposed a comprehensive relation model that contained temporal (atemporal) data with determinacy and indeterminacy. New forms were proposed to represent the atemporal and temporal indeterminacy present in temporal databases. Also, temporal relation algebra and query language were elaborated, the advantage of this approach was that it associated a traditional temporal algebra with indeterminacy. All these new characteristics made this model better treat indeterminate temporal (atemporal) data including storage and query. Therefore, the application scopes of temporal databases were extended furthermore.

References

[1] Jensen C.S and Dyreson C.E.(eds.) A Consensus Glossary of Temporal Database Concepts- February 1998 Version. 1998: 367-405
[2] Tang Y, Tang N, Ye X P. Temporal Information Processing Research Survey [J]. Guangzhou: Acta Scientiarum Naturalium Universitatis Sunyatsen 2003, 42(4): 4-8
[3] Abdullah Uz Tansel. On Handling Time-varying Data in the Relational Data Model. Information and Software Technology. 46(2004):119-126
[4] S C Zhang, C Q Zhang. A Probabilistic Data Model and its Semantics. Journal of Research and Practice in Information Technology, 2003, 35 (4): 237-257
[5] C E. Dyreson, R.T. Snodgrass. Supporting Valid-time Indeterminacy. ACM Transaction on Database System. 1998, 23(1): 1-57
[6] K.Matousek, J.Uhlir. On Representing Uncertain Historical Time. Proceedings of the 15th International Workshop on Database and Expert Systems Applications (DEXA'04), 2004 : 1529-4188
[7] Jaruloj Chongstitvatana. Imprecise Intervals in Temporal Databases. IEEE 1998:233-237
[8] G Garani. A Generalized Temporal Algebra. Data & Knowledge Engineering. 2006, 57 (3): 283-310
[9] M.A. Bassiouni, M. Llewellyn. Extending Temporal Query Languages to Handle Imprecise Time Interval. Computer Language.25 (1999):39-54

Ren Jiadong received the B.S. and M.S. degrees in computer science from Northeast Heavy Machine College in 1989 and 1994, respectively. He received the Dr degree from Harbin Institute of Technology University in 1999. After working as a teacher (from 1989), an assistant professor (from 1999) in the Yanshan University, and he has been a professor at Yanshan University since 2005. His research interest includes spatio-temporal Data Model, XML Data Model, and Data Mining. He is a member of IEEE SMC Society and director of Electrical Education Consortium of Hebei Province in China.

Peng Huili received the B.S. degrees in Computer Information Science from Henan Normal University in 2004. She is a postgraduate of Yanshan University since 2004. Her main research interest includes Temporal Data Model and Data Mining.

Zhang Xiaojian received the B.S. degrees in Computer Information Science from Henan Normal University in 2004. He is a postgraduate of Yanshan University since 2004. His main research interest includes Web Usage Mining and Temporal Data Model