

基于时态的次协调数据库的研究

黄 慧¹, 毛宇光^{2,3}

(1 三江学院 计算机科学与工程学院, 江苏 南京 210012;

2 南京航空航天大学 信息科学与技术学院, 江苏 南京 210016;

3 南京大学 计算机软件新技术重点实验室, 江苏 南京 210093)

摘 要: 基于时态的次协调数据库的数据模型是用来处理以时间为推移的数据库中不完全信息和矛盾信息. 本文基于这样的数据模型, 首先, 提出三史制和超期数据转移算法及时间点抽取算法, 这些算法可用于处理大量的过期数据, 同时使用双向链表存储此类数据; 其次, 提出基于时态的次协调数据库查询语言 (TPQL) 实现此种数据库的相关查询与操作功能.

关键词: 次协调数据库; 四值逻辑; 时态; 三史制; 双向链表; TPQL 语言

中图分类号: TP31

文献标识码: A

文章编号: 1000-7180(2012)12-0124-05

Research on Temporal Paraconsistent Database

HUANG Hui¹, MAO Yu-guang^{2,3}

(1 Sanjiang University, College of Science and Engineering, Nanjing 210012, China; 2 College of Information

Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China;

3 State Key Lab for Novel Software Technology, Nanjing University, Nanjing 210093, China)

Abstract: Temporal Paraconsistent database is introduced to manipulate incomplete and inconsistent information in database that vary over time. In this article, firstly, three historical segments, age transferring algorithm and time point selection algorithm are provided to solve the problem of overdue information, meanwhile, doubly linked list is used to store data; secondly, TPQL, as an extension of UcQL, is given for the sake of query and manipulation on this kind of data.

Key words: paraconsistent database; four-valued logic; temporality; three historical segments; doubly linked list; TPQL

1 引言

在现实世界里, 随处都可遇到不确定信息^[1-2], 即不完全信息和不一致信息 (又称矛盾信息). 但人们认为数据库中不应该包含不确定信息, 使得这种信息一直未受到重视^[1-2]. 目前使用的数据库还不能专门处理这种信息, 然而生活中矛盾信息, 不完全信息却是很常见的^[1-2]. 为了描述含有不完全信息的数据库, 文献[1]提出了次协调数据库, 并给出了相应的模型查找不确定信息. 在此模型基础上, R. Bagai 提出基于时态的次协调数据库模型^[3], 此后,

对于基于时态的次协调数据库模型一直未有明显的进展. 不同的时间点记录的这种不完全信息在实际中发挥着不同的作用. 因此, 此模型在实际中有着重大的意义. 但此模型只提供了相应的代数运算, 还有很多问题有待解决. 本文将解决以上提出的问题.

2 基于时态的次协调数据库模型

2.1 协调关系模型

和传统的数据库不同, 次协调数据库引入了四值逻辑^[1-2], 用符号 (c, t, f, u) 表示, 分别为矛盾、真、假、未知. 次协调关系的模式 Σ 总是成对出现, R

$= \langle R^+, R^- \rangle$, 其中 R^+, R^- 为 $T(\Sigma)$ 的任意子集. 关系模式 Σ 是属性名的有限集, 对于任一个属性名 $A \in \Sigma$, $\text{dom}(A)$ 是 A 的非空域, 如果一个元组在 Σ 上则必需满足对每个 $A \in \Sigma$, $t(A) \in \text{dom}(A)$. $T(\Sigma)$ 表示在 Σ 上所有元组的集合. 这里, R^+ 是被认为具有 R 属性的元组的集合, R^- 是被认为不具有 R 属性元组的集合. 由于矛盾信息的存在, R^+ 和 R^- 的交集未必为空. 同样, 由于不完全信息的存在, R^+ 和 R^- 的并集不一定是 $T(\Sigma)^{[1-2]}$.

TEST

+		-	
P1	S1	P1	S3
P1	S2	P2	S1
P2	S1		
P2	S3		

例1 TEST在模式 $\{P, S\}$ 上, P 表示病人, S 表示症状. 这里 P 的域为 $\{P1, P2\}$, S 的域为 $\{S1, S2, S3\}$, TEST 表明病人 $P1$ 的症状 $S1$ 和 $S2$ 为阳性, 症状 $S3$ 为阴性, 病人 $P2$ 的症状 $S1$ 和 $S3$ 为阳性, 症状 $S1$ 还同时测得为阴性, 此时病人 $P2$ 的 $S1$ 症状矛盾, 病人 $P2$ 的症状 $S2$ 未知.

2.1.1 次协调数据库中的集合运算符交、并、差 ($\cap, \cup, -$)

定义1 令 R 和 S 是模式 Σ 上的两个次协调关系, 则有

$$(R \cup_p S)^+ = R^+ \cup S^+, (R \cup_p S)^- = R^- \cap S^-;$$

$R^+ \cup S^+$ 和传统数据库一样, 可解释为具有 R 或 S 属性 (either R or S), $(R \cup_p S)^-$ 解释为不具有 R 或 S 属性 (not (either R or S)), 即表示为 $R^- \cap S^-$. 同理, 对于 $\cap_p, -_p$, 则有

定义2

$$(R \cap_p S)^+ = R^+ \cap S^+, (R \cap_p S)^- = R^- \cup S^-;$$

$$(R -_p S)^+ = R^+ \cap S^-, (R -_p S)^- = R^- \cup S^+.$$

2.1.2 次协调数据库中的关系运算

定义3 有关系模式 Σ 和 Δ 且 $\Sigma \subseteq \Delta$, 对于任意元组 $t \in T(\Sigma)$, 使得 t' 为:

$$\{t' \in T(\Delta) \mid t'(A) = t(A), \text{ for all } A \in \Sigma\}.$$

定义4 次协调数据库中的投影运算 (Π)

令 R 为模式 Σ 上的次协调关系, 且 $\Delta \subseteq \Sigma$, 则 R 在 Δ 上的投影, 用 $\Pi_\Delta(R)$ 表示:

$$\Pi_\Delta(R)^+ = \Pi_\Delta(R^+),$$

$$\Pi_\Delta(R)^- = \{t \in T(\Delta) \mid t^\Sigma \subseteq R^-\}.$$

定义5 令 R 是模式 Σ 上的次协调关系, Δ 是

任意模式. 那么, R 在 Δ 上的投影, 记为 $\Pi_\Delta(R)$:

$$\Pi_\Delta(R)^+ = \Pi_\Delta((R^+)^\Sigma \cup \Delta),$$

$$\Pi_\Delta(R)^- = \{t \in T(\Delta) \mid t^\Sigma \cup \Delta \subseteq (R^-)^\Sigma \cup \Delta\}.$$

定义6 次协调数据库中的连接运算 ($\bowtie$)

令 R 和 S 是模式 Σ 和 Δ 上的次协调关系, 则 R 和 S 的连接, 用 $R \bowtie S$ 表示:

$$(R \bowtie S)^+ = R^+ \bowtie S^+,$$

$$(R \bowtie S)^- = (R^-)^\Sigma \cup \Delta \cup (S^-)^\Sigma \cup \Delta.$$

定义7 次协调数据库中的选择运算 (σ)

令 R 是模式 Σ 上的次协调关系, $\sigma_C(R)$ 是 R 上满足条件 C 的元组:

$$\sigma_C(R)^+ = R^+ \cap \{t \mid C \text{ is } t \text{ or } c \text{ for } t\},$$

$$\sigma_C(R)^- = R^- \cup \{t \mid C \text{ is } f \text{ or } c \text{ for } t\}.$$

2.1.3 次协调数据库中的条件运算 in

定义8

$$t \text{ in } R = \begin{cases} c & \text{if } t \in R^+ \cap R^- \\ t & \text{if } t \in R^+ - R^- \\ f & \text{if } t \in R^- - R^+ \\ u & \text{otherwise} \end{cases}$$

2.2 时态模型

在自然界中, 时间轴上是连续存在的. 如果按照现实世界的本来面目来记录时间, 则数据量过于庞大, 而且人们往往并不需要获取现实世界所有时间的状态. 解决这一问题的方法是建立合适的时间模型, 使得它不仅能应用到数据模型中, 而且能满足不同应用场合的要求^[4-8].

2.2.1 基于时态的次协调关系数据模型

在次协调关系数据库中, 数据是随着时间的变化而变化的, 要反映这样一个变化情况, 就要引入基于时态的次协调数据库. 其中 $T = \{1, 2, 3, \dots\}$, 表示 T 是一个时间集合, 从 1 时刻开始, 一直延续下去.

下面给出一个基于时态的次协调关系的实例:

例2 有两个病人 $P1, P2$, 他们被间断性地检测三个症状 $S1, S2, S3$. 设 $PAT(P, S)$ 是基于时态的次协调数据库的关系模式. PAT 在 1, 2, 3 时刻的表示如图 1. PAT 表示在 1 时刻 $P1$ 的 $S1, S3$ 症状呈阳性, $S2$ 症状呈阴性; 2 时刻 $P2$ 的 $S1$ 症状未知, $S3$ 症状矛盾. 其他可以此类推.

2.2.2 基于时态的次协调关系运算符

令 $\langle T(1), T(2), T(3), \dots \rangle, \langle S(1), S(2), S(3), \dots \rangle$ 是具有相同模式的基于时态的次协调关系, 则

$$T \cap_i S = \langle T(1) \cap_p S(1), T(2) \cap_p S(2), T(3)$$

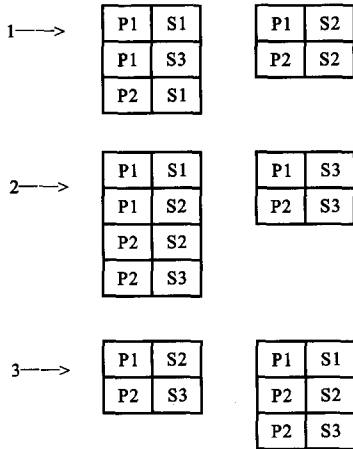


图1 基于时态的次协调关系 PAT

$\cap_P S(3), \dots >$

$\cap_P$ 表示基于时态的次协调关系交运算; 由于基于时态的次协调关系运算是以次协调数据模型为基础的, 所以每个时刻的运算符采用次协调数据模型中介绍的运算符。

其他运算如并、差、投影、连接、选择可以类似定义, 并用符号 ($\cup$ 、 $-$ 、 Π 、 $\bowtie$ 、 σ) 表示。

3 基于时态的次协调数据库的数据转移

对于随时间推移产生的大量数据进行查询或处理时, 这种操作常常需要的是近期的而非久远前产生的数据, 三史制的方法将数据分为古代、近代和现代, 并仅对现代数据进行查询和处理工作。而对于过期的数据采用超期数据转移算法将不重要的数据移出, 移出的大量数据采用抽取时间点的方法进行保存。

3.1 三史制

为了提高时态数据库存储和查询性能, 根据 Hbase 模型, 把次协调关系数据库的历史按时间划分为 3 个时代和两个过渡区间, 并称这一存储方式为三史制。3 个时代共有 6 个时刻变量 OB(OldAge-Began)、OE(OldAgeEnd)、MB(MidAgeBegin)、ME(MidAgeEnd)、CB(CurAgeBegin) 和 CE(CurAgeEnd), 时态约束条件为 $0 = OB < MB \leq OE < CB \leq ME < CE$ 。区间 $[OB, OE]$ 、 $[MB, ME]$ 、 $[CB, CE]$ 分别称为古代、近代和现代。两个过渡区间 $[MB, OE]$ 、 $[CB, ME]$ 分别称为古代—近代过渡期和近代—现代过渡期^[5-6]。过渡期中的历史数据在两个相关历史时代中各存一份, 由时代转移模块自动而平滑地转移过渡期中超期的历史数据。近代和古代的数据分别保存在硬盘和磁带中, 现代数据从硬盘调入内存进行查询等操作处理, 如图 2 所示。

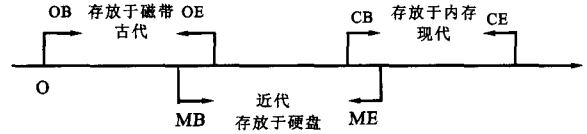


图2 三个时期和两个过渡区间

3.2 转移算法与时间点抽取算法

现代史中的数据是在内存上处理的, 由于是间断性的获取数据, 所以时间上并非连续。间断性的获取数据使得内存上隔一个时间段就会有新数据加入基于时态的次协调数据库关系中, 同时, 排在靠前时刻的数据在相对长时间后成为近代史中的数据, 应将其删除并插入到近代史中。这里 C 的风格书写的算法描述了这一过程。

算法1 现代—近代转移算法

输入: 现代史、近代史

输出: 现代史

CurAge Transfer(CurAge, MidAge)

```
{
    CB_Old = GetOldCB(); // 获取旧的“现代”始点
    CB_New = GetNewCB(); // 获取新的“现代”始点
    ME_Old = GetOldME(); // 获取旧的“近代”终点
    ME_New = GetNewME(); // 获取新的“近代”终点
    if(OldCB < NewCB)
    {
        for each record in ([OldCB, NewCB] ∩ CurAge)
        {if Record_not_in(MidAge)
            {
                // 追加该记录到近代史
                Append_MidAge(record);
                Delete(cq); // 从现代史删除
            }
        }
        // 将近代史的数据压缩存储, 见算法 2
        CompressMidAgeData(MidAgeData);
    }
    else printf("error!");
}
```

时间点抽取算法在时代转移算法时进行。当过渡期中数据从现代向近代转移时调用 CompressMidAgeData 函数将庞大的数据压缩后保存。先选取需要压缩的相对时间点 Point, 并以 Point 为中心选取前后 n 个时间点的数据做压缩。时间量子为某一时代选取的时间点的总个数, 如 Point 为 100, n 为 50, 则选取 $[50, 150]$ 时间点的信息进行压缩。然后将时间量子除以压缩倍数, 如果整除, 则将相应时间点上数据保存。

算法 2 时间点抽取算法

输入:近代史,相对时间点,前后位移,压缩倍数

输出:压缩的近代史

MidAge Compress MidAge Data(MidAge,Point,n,m)

```

{
    //选取近代史相对时间点
    RelPoint = MidAge. Point;
    ArraySize = RelPoint/m; //压缩后时间点个数
    //使用数组存放压缩后信息
    RecordType Arr[] = RecordType[ArraySize];
    //获取需要压缩的近代数据
    CompressData = GetCompressData (RelPoint - n,
    RelPoint+n)
    i=0;
    For each record in CompressData
    {
        If(Record_TimePoint/CompressNum==0)
        {
            //将符合压缩条件的信息加入数组
            Arr[i] = record;
            i++;
        }
        Else
        {
            //将不符合压缩条件的信息删除
            Delete(record)
        }
    }
}

```

3.3 基于时态的次协调数据库的存储

对于现代史中的数据进行查询及处理的工作,考虑到如何能获取某一特定时间点的信息以及以该特定时间点信息为标准的前后时间点的信息,这里使用双向链表可以方便的完成这些操作. 当有新的信息产生时,插入时间点的算法可以将新信息插入到时态次协调数据库中;同时,有信息不需要时,删除时间点算法可以将无用的信息从时态次协调数据库中删除.

考虑到存储结构的合理性,允许从当前时间点访问前 n 个或后 n 个时间点,可以使用双向链表处理现代史的数据. 双向链表还可以处理现代史中的数据插入、删除情况,每个结点有五个域,分别存储时间点、(+)、(-)表的首地址及指向前后时间点的指针(prev,next 域). 如图 3 所示(插入新的时间结点与删除结点算法与双向链表结构算法类似,此处略).

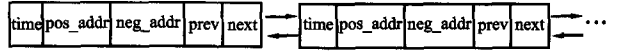


图 3 双向链表存储结构

4 基于时态的次协调数据库的查询语言 (TPQL)

对时态次协调数据库进行查询,还需要提供适合该模型的查询语言. 由于数据被划分为 3 个时代及 2 个过渡期,这里提供 SET 语句用于划分时代及过渡期. 查询语句及更新语句中均使用了 PEIROP 子句及 FIRST, NEXT, PREVIOUS 对任意时间点进行查找及维护.

4.1 划分时期及过渡期语句

定义语句中新定义了 SET 子句,用于划分时期并划定过渡期长度. 语法如下:

```

<schema_def> ::= <contra_table_def> | <noncontra_table_def> .....
<set_clause> ::= SET <old_age> | <mid_age> | <cur_age> | <transfer_period>;
<old_age> ::= OldAgeBegin <value_expression>
<mid_age> ::= MidAgeBegin <value_expression>
<cur_age> ::= CurAgeBegin <value_expression>
<transfer_period> ::= Old_Mid_Len | Mid_Cur_Len <value_expression>

```

4.2 查询语句

基于时态的次协调数据库查询语言是 UcQL 语言^[6]的一个扩展,本节根据双向链表的存储结构又新增了关键词 PEIROP, PEIROP 可连接 FIRST, NEXT, PREV, BETWEEN 和 EQUAL,在语法上满足了时态查询的基本要求. 语法如下:

```

<select_clause> ::= [COMPLEMENT]
SELECT[ALL|DISTINCT] <selection>
FROM[COMPLEMENT] <history_relation_ref> {, <history_relation_ref>} *
WHERE <search condition>
PEIROP <peirod_clause> PTINTERSECT | PTUNION
<select_clause>;
<selection> ::= * | <scalar_exp_commallist>
< history_relation_ref > ::= < history_relation_name >
<search condition> ::= [<slo>] <pred> [<dlo>] [<slo>] <pred> ...]
<slo> ::= PNOT | UNKNOWN
<dlo> ::= PAND | POR
<pred> ::= <simple_pred> | <complex_pred>
<simple_pred> ::= <vterm> <comp_op>

```

```

<vterm>
<vterm>::=不含聚集函数的算术表达式
<comp_op>::=<|>|<|=|<|LIKE
<complex_pred>::=<between_predicate>|<in_
predicate>|<exists_predicate>
<exists_predicate>::=EXISTS <select_clause>
.....
<peirod_clause>::=TIME BETWEEN <value_ex-
pression> PAND <value_expression>
|FIRST<history_relation_name>
|NEXT<value_expression>
|PREV<value_expression>
|EQUAL<history_relation_name>

```

<value expression>表示时刻值, **FIRST** 表示第一时刻, 即返回时间点值为 1 的信息; **NEXT** 表示当前时刻的下一时刻值, 返回当前时间点值加上 1 的时刻的信息; **PREV** 表示当前时刻的前一时刻值, 返回当前时间点值减去 1 的时刻的信息; **EQUAL** 返回<history_relation_name>描述的时间点的值, **PTINTERSECT** 表示时态次协调数据库中的交运算, **PTUNION** 表示时态次协调数据库中的并运算。

4.3 插入语句

语法如下:

```

<insert_statement>::=<insert_node>|<insert_tu-
ple>
<insert_node>::=INSERT INTO <history_rel at ion
_name> VALUES <node_list>;
<node_list>::=<time><pos_address><neg_ad-
dress>
<insert_tuple>::=<insert_statement>
<insert_statement>::=<insert_contratable>|<in-
sert_noncontratable>

```

更新语句与删除语句与插入语句类同, 此处略。

5 应用

例 3 下列语句用相对时刻动态地划分时期。

SET MidAgeBegin=AGO(3);

SET Old_Mid_Len=1;

第一个语句用来设置近代史的数据开始于最近 3 个时间点, 第二个语句用来设置古代史和近代史的过渡期为 1 个时间点。

例 4 基于时态的次协调关系 PAT, 时间点为 0 开始设为古代史, 时间点为 10 开始设为近代史, 从时间点为 20 开始设为现代史, 古代—近代史和近代—现代史的过渡期均为 3 个时间点。如果使用 insert 语句插入时间点且当前时间点为 30, 情况可用

图 4 表示, 其中古代—近代史过渡期为时间点[9—11], 近代—现代史的过渡期为时间点[19—21]。

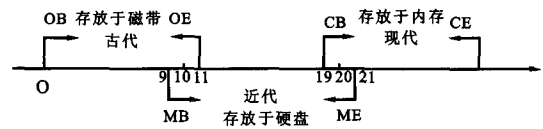


图 4 关系 PAT 的三个时期和两个过渡期

SET OldAgeBegin=0

SET MidAgeBegin=10

SET CurAgeBegin=20

SET Old_Mid_Len=3

SET Mid_Cur_Len=3

当有新的时间点插入, 如果要已将超期的数据从现代史中删除并插入近代史, 则应重新设置现代史时间并调用算法 1 和算法 2, 如:

SET CurAgeBegin=30.

6 结束语

本文在基于时态的次协调数据库模型的基础上, 扩展了此数据库的应用。结合三史制的方法和超期数据转移算法、时间点抽取算法转移过期数据, 并采用双向链表的数据结构存储数据。同时, 基于这种模型, 提出了适合该模型的查询语句 (TPQL)。通过调用以上算法及查询语句实现了基于时态的次协调数据库的查询与操作功能。由于次协调数据库的概念直到近年来才被提出, 还需要将传统数据库中的关系演算、查询优化、范式、完整性约束等概念引入到次协调数据库中来, 以完善其整个系统。

参考文献:

- [1] R. Bagai, R. Sunderraman. A paraconsistent relational data model[J]. International Journal of Computer Mathematics, 1995, 55(1): 39-55.
- [2] R. Bagai. A Query Construct for Paraconsistent Databases[C]//Proceedings of the 7th International Conference on information Processing and Management of Uncertainty in Knowledge-Based Systems, Paris, France; IEEE, 1998: 428-434.
- [3] R. Bagai, M. A. Orgun. A Temporal Paraconsistent Relational Algebra for Incomplete and Inconsistent Information[C]//Proceedings of the 33rd Annual ACM Southeast Conference, 1995: 240-248.
- [4] 刘海, 汤庸. 时态数据完整性约束研究与实现[J]. 计算机科学, 2010, (37): 175-179.

(下转第 132 页)

字水印提取的困难,这一测试也反应出本文所设计的音频数字水印算法在鲁棒性上具有比较优良的表现,能够抵抗常见的各种音频水印攻击手段.但是在测试中发现对本文所设计的音频数字水印算法应用过程中如果对载有水印信息的文件进行部分采样点的剪切攻击之后会导致音频文件不全,此时这种攻击对数字水印的提取有比较严重的影响,在实验测试过程中发现当对原始的音频文件的采样点剪切攻击达到百分之二十之后所提取出来的数字水印信息基本上无法正确的反应原始的水印信息,这也表明本文所设计的水印算法在抗采样点裁剪攻击这一攻击形式时性能相对较弱.事实上造成这一因素的根源也是由本文所设计的音频数字水印嵌入算法而决定^[13],因为本文算法主要是利用音频中对人耳掩频效应的特征进行水印的嵌入,而没有更多的去研究和分析频率空间、音频信号的相关性.从而使得所嵌入的数字水印在抗剪切攻击时性能表现较差.

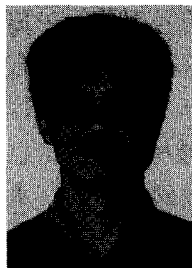
5 结束语

本文根据音频信号掩频效应这一特有的对听觉影响的效果,设计了基于掩频效应的音频数字水印嵌入算法.该算法充分利用了音频信号中的掩频效应对人耳听觉的影响,巧妙的将数字效应嵌入在音频的掩频区域中,从而实现了在音频文件中高质量的嵌入数字水印的这一应用需求.实验测试表明,本文所设计的音频数字水印算法在鲁棒性和不可见性等方面具有较优良的性能.

参考文献:

- [1] 陶智,赵鹤鸣,龚呈卉.基于听觉掩蔽效应和Bark子波变换的语音增强[J].声学学报,2005,(04):367-372.
- [2] 张敏,杨慧.基于混沌加密的盲音频水印算法[J].中国民航学院学报,2005,23(03):37-41.
- [3] 梁娟,朱忠军.基于心理声学模型的自适应音频数字水印算法[J].现代电子技术,2004,27(20):18-21.
- [4] 吴绍权,黄继武,黄达人.基于小波变换的自同步音频水印算法[J].计算机学报,2004,27(03):365-370.
- [5] 梁华庆,赵丽丽,钮心忻,等.一种基于心理声学模型的小波包域音频数字水印算法[J].石油大学学报(自然科学版),2003,27(06):112-115.
- [6] 谢志文,尹俊勋.音频掩蔽效应的研究及发展方向[J].电声技术,2002(12):4-7.
- [7] 李国富.基于正交拉丁方的数字图像置乱方法[J].北方工业大学学报,2001,13(01):14-16.
- [8] 王向阳,崔永瑞,赵红.基于整型小波的数字音频二维水印嵌入算法研究[J].辽宁师范大学学报:自然科学版,2005,28(03):297-301.

作者简介:

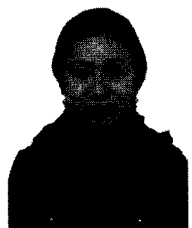


邱志宏 男,(1963—),副教授.研究方向为网络安全、信息安全.

(上接第128页)

- [5] 刘冬宁,汤庸.时态数据库时间轴的动态逻辑模型[J].软件学报,2010,(21):694-701.
- [6] 黄慧,毛宇光.一种支持次协调数据库的UcQL语言[J].计算机工程与应用,2006,(10):158-161.
- [7] 姜晓轶,蒋雪中,周云轩.时态数据库研究进展[J].计算机工程与应用,2005,(24):27-30.
- [8] 汤庸.时态数据库导论[M].北京:北京大学出版社,2004.

作者简介:



黄慧 女,(1979—),硕士,讲师.研究方向为数据库.