

电子病历中不确定时态数据的时态过滤运算^①

Temporal Filtration Operation of Uncertain Temporal Data in Electronic Medical Record

黄雄波 (佛山职业技术学院 计算机工程系 广东 佛山 528000)

陈章 (广东商学院 信息学院 广东 广州 510320)

摘要: 在以电子病历(Electronic Medical Record)为核心的 HIS 医院信息管理系统中,病历数据的管理是其核心内容。介绍了电子病历中不确定时态数据的概念及表示方法,进一步,在 Delphi 编程环境中深入讨论了不确定时态数据的瞬时过滤、期间过滤的设计原理及编程方法,从而有效地展示了电子病历中不确定时态数据在过去某个时刻或期间的医疗情况。

关键词: 不确定时态数据 时态过滤 电子病历 Delphi

随着计算机、通讯网络及数据库技术的迅猛发展,我国医疗信息化的重点已从医院管理信息化向临床管理信息化转变,而电子病历(Electronic Medical Record, EMR)则是临床信息管理现代化的重要标志。所谓电子病历实质是指计算机化的病历,它不仅包括了患者纸张病历的原有内容,而且还反映了患者整个医疗的过程,储存了患者全部的医疗信息(病史、各种检查表格和影像资料等多媒体信息),是个人医疗信息及其相关处理过程综合化的体现。

客观现实世界是一个四维的世界,每个事物都有其时间维,伴随着时间的流逝,事物的信息流也就包含着众多的时态信息,即事物在不同的历史时刻其数据值是不同的^[1]。在电子病历中其医疗数据具有明显的时态特性,如患者的病情记录、会诊记录、用药记录等,均随着时间的推移而有所改变。然而,至今为止,还没有出现成熟通用的 TDBMS(Temporal DBMS,时态数据库管理系统),且当前主流的 Oracle、DB2 及 MS-SQL Server 等 RDBMS 均属于非时态的 DBMS,即对时态数据没有专门的处理措施。因此,在实际应用中,一般是在非时态 DBMS 的基础上通过 SQL 扩展或中间件技术来处理时态数据^[2,3]。针对医疗信息系统的需求,本文介绍了电子病历中不确定时态数据的概念,并结合 Delphi 开发工具对其时态过滤进行分析与设计。

1 电子病历中不确定时态数据的表示方法

1.1 不确定时态数据的概念

时态数据库模型是在传统数据库模型的基础上增加时间维来刻画某个时刻的数据,并反映历史和揭示将来。目前众多的时态数据库模型都是建立在有效时间的开始点和结束点均确定的基础上,但由于现实世界是动态变化的,在许多情况下,我们并不能准确地获取事件的确切发生时间和结束时间。在电子病历的病情记录中,患者出现疾病症状和病情开始好转的时间一般来说是不确切的。如某患者反映大概在 1999/12/08 09:32 ~ 1999/12/08 10:22 期间内出现了感冒症状,到医院确诊且经过治疗后,大概在 1999/12/14 06:00 ~ 1999/12/15 19:00 期间内感冒症状逐步减轻并痊愈,显然,该患者的病情记录是一条不确定时态数据,这里的时间粒度选取为分钟。

1.2 不确定时态数据的表示方法

确定性时态数据一般是在其历史数据表中添加两个 DateTime 类型的数据来表示事件的开始和结束时间,即元组中的确定时态属性可用(Value, < t₁, t₂ >)的形式来表示。以患者的主诊医生时态数据为例,

① 基金项目:广东省自然科学基金资助项目(06023960);佛山职业技术学院校级科研基金资助项目(01028)

Value 表示的是主诊医生的工号,而 t_1, t_2 是两个确定的时刻,表示该主诊医生在 $[t_1, t_2]$ 期间内负责这名患者的诊断和治疗。

不难发现,只要把确定时态数据的表达式中的两个确定时刻 t_1 和 t_2 分别扩展为时间期间,便可得到不确定时态数据的表示方法,即 $(Value, <t_{es}, t_{is}>, <t_{ef}, t_{if}>)$ 。这里, t_{es}, t_{is}, t_{ef} 和 t_{if} 分别表示不确定时态属性值的最早发生时间、最迟发生时间、最早结束时间和最迟结束时间。以前述患者的感冒症状为例,由于它是属于一条不确定时态数据,故可用如下形式来表示: $(感冒, <1999/12/08 09:32, 1999/12/08 10:22>, <1999/12/14 06:00, 1999/12/15 19:00>)$ 。

综上所述,我们可以利用 MS - SQL Server 2005 非时态 DBMS 设计出如表 1 所示的病情记录的历史数据表,其中,“病历号”、“疾病名称”和“最早发病时间”三个属性为数据表的联合主键。

表 1 电子病历中病情记录的历史数据表

病历号	疾病名称	最早发病时间	最迟发病时间	最早痊愈时间	最迟痊愈时间
990125	感冒	1999/12/08 09:32	1999/12/08 10:22	1999/12/14 06:00	1999/12/15 19:00
990125	肝炎	1999/12/10 11:09	1999/12/13 09:00	2000/01/22 08:00	2000/01/29 18:30
026383	肺炎	2003/08/06 07:22	2003/08/09 12:09	2003/09/13 09:25	2003/09/19 20:39
026383	高血压	2004/01/12 11:08	2004/01/17 15:21	2004/02/23 12:00	2004/03/01 08:00
.....					

2 时态过滤的定义

设 $Tt = [t_s, t_e]$ 为考察的时间区间,当 $t_s = t_e$ 时,该时间区间便退化为一时间点(即时刻)。若 T_k 为某元组属性 A_i 取 m 种值的对应的的时间区间,则 $T_k = [[A_i = a_k]] (a_k \in Dom(A_i), k = \{1, 2, \dots, m\})$ 。

定义:所谓时态过滤(Temporal Filtration)^[4,5],即求得元组的时态属性在要考察的时间区间内取何种数值的问题。以表 1 中的病情记录不确定时态数据为例,其时态属性的取值可分为以下四种情况: {没有病情记录,出现“xxx”疾病的症状期间,确诊为“xxx”疾病

的期间、“xxx”疾病的痊愈期间}。

例 1:若 $T_1 T \subseteq_k$,则经过时态过滤后,该属性取值为 a_k 。如考察表 1 中病历号为“990125”的患者在 2000 年 1 月 1 日 12:00 这一时刻的病情记录情况,可得到如下结果:

病历号	病情记录的时态过滤运算结果
990125	确诊为“肝炎”疾病的期间

例 2:若 $T_1 \not\subseteq T_{k1}, T_1 T_{k2}, \dots, T_1 T_{kn}$,但 $T_{1T} k1 \cup T_{k2} \cup \dots \cup T_{kn}$,经过时态过滤后,该属性取值为 $a_{k1}, a_{k2}, \dots, a_{kn}$ 。如考察表 1 中病历号为“026383”的患者在 2003 年 9 月 18 日 12:00 至 2004 年 2 月 20 日 12:00 这一期间的病情记录情况,可得到如下结果:

病历号	病情记录的时态过滤运算结果
026383	[2003/09/18 12:00 ~ 2003/09/19 20:39] “肺炎”疾病的痊愈期间;
	[2003/09/19 20:39 ~ 2004/01/12 11:08] 没有疾病记录情况;
	[2004/01/12 11:08 ~ 2004/01/17 15:21] 出现“高血压”疾病的症状期间;
	[2004/01/17 15:21 ~ 2004/02/20 12:00] 确诊为“高血压”疾病的期间;

需要指出的是,这里的时态过滤运算只考虑了病情记录这一种医疗情况,至于多种医疗情况的时态过滤运算,我们将另文论述。

3 电子病历中不确定时态数据的时态过滤

3.1 电子病历中不确定时态数据的瞬时过滤

所谓电子病历的不确定时态数据的瞬时过滤实质就是要获取患者在特定时刻里的医疗情况。继续以表 1 的不确定时态数据为例,瞬时过滤的程序实现过程是,首先在历史数据表中用 SQL 语句分别在[最早发病时间,最迟发病时间]、[最迟发病时间,最早痊愈时间]和[最早痊愈时间,最迟痊愈时间]区间内检索出满足查询条件的不确定时态数据,然后把这些查询结果映射成上述所介绍的四种不确定时态属性的取值情况,最后把时态过滤运算结果显示在一个 StringGrid 控件^[6]中。

在 Delphi 编程环境中,StringGrid 控件是一种通用的栅格,主要用于在行和列中显示各种数据。StringGrid 控件的相关属性有:

ColCount:指定栅格的列数。

RowCount:指定栅格的行数。

Cells[ACol, ARow]: 该属性是一个二维数组, 通过它可以存取特定的行和列上的数据。其中 ACol 是列的序号, ARow 是行的序号, 序号都是从 0 开始。

图 1 是电子病历中病情记录不确定时态数据的瞬时过滤的程序运行示意图。在图 1 中, 病历号为“990125”的患者在 1999 年 12 月 11 日 12 时 00 分这一时刻的过滤运算结果出现了两个记录, 这表明该患者当时出现了并发症。

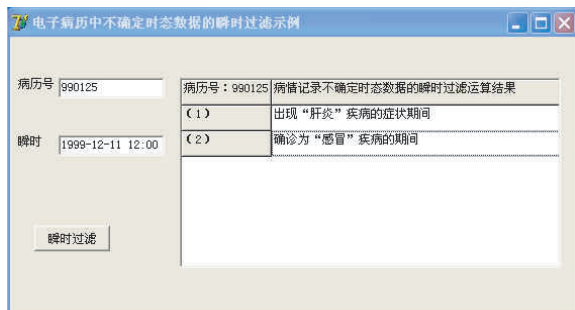


图 1 不确定时态数据的瞬时过滤的程序运行示意图

主要源码如下:

```
{ 栅格初始化, 设置其列数和行数 }
StringGrid1. ColCount := 2;
StringGrid1. RowCount := 2;
{ 构造 < 出现“xxx”疾病的症状期间 > 的 SQL 查询语句 }
SQLStr := 'select 疾病名称 from 病情记录历史数据表 where (病历号 = :var1) and (最早发病时间 < :var2) and (最迟发病时间 > = :var2)';
ADOQuery1. Active := False;
ADOQuery1. SQL. Clear;
ADOQuery1. SQL. Add( SQLStr );
{ 设置 SQL 查询的动态参数 }
ADOQuery1. Parameters[0]. DataType := ftString;
ADOQuery1. Parameters[0]. Value := Edit1. Text;
ADOQuery1. Parameters[1]. DataType := ftDateTime;
ADOQuery1. Parameters[1]. Value := StrToDateTime(Edit2. Text);
ADOQuery1. Active := True;
{ 把查询结果映射为 < 出现“xxx”疾病的症状期间 > 的不确定时态属性取值情况, 并显示在栅格中 }
```

```
while not ADOQuery1. Eof do
begin
StringGrid1. Cells[1, StringGrid1. RowCount - 1] := '出现' + ADOQuery1. FieldByName('疾病名称'). AsString + '疾病的症状期间';
ADOQuery1. Next;
{ 调整栅格的行数 }
StringGrid1. RowCount := StringGrid1. RowCount + 1;
end;
.....
```

3.2 电子病历中不确定时态数据的期间过滤

所谓电子病历的不确定时态数据的期间过滤实质就是要获取患者在某一特定期间内的医疗情况。为了增强数据显示的直观性, 期间过滤在设计实现上应以多个独立的时间子区间来展示医疗信息^[7], 如例 2 的期间过滤就把患者的医疗信息分成四个时间子区间来考虑。

不确定时态数据的期间过滤比其瞬时过滤要复杂得多, 而求出期间过滤的各个时间子区间则是其关键。这里, 我们设计了一种如图 2 所示的不确定时态数据期间过滤的时间子区间提取算法。该算法的主要思想是, 首先用 ADOQuery 控件检索出患者在时间上与过滤期间有重叠的所有不确定时态数据, 然后在一个直到型循环结构中遍历这些不确定时态数据, 借助 Record_Array 和 ResultTemp_Array 两个日期时间型动态数组, 用二路归并排序法^[8]把落在过滤期间的每条记录的四个日期时间型字段值(最早发病时间、最迟发病时间、最早痊愈时间及最迟痊愈时间)按升序的格式写入至 Result_Array 日期时间型动态数组中。显然, Result_Array 数组中两两相邻的两个元素就是所求的各个独立的时间子区间的起点和终点。

算法的实现要点如下:

(1) 判断每条不确定时态数据 t_1 [最早发病时间, 最迟痊愈时间] 与过滤期间 t_2 [过滤期间的起点, 过滤期间的终点] 是否存在时间重叠的方法为: 若 $\forall (\neg ((\text{过滤期间的起点} > \text{最迟痊愈时间}) \vee (\text{最早发病时间} > \text{过滤期间的终点})))$, 则 t_1 与 t_2 存在时间重叠。

(2) 动态数组的使用: 为了应用程序的资源分配更合理, Borland Delphi 开发工具允许程序员在程序设

计过程中使用动态数组技术。其中, SetLength 函数用来调整动态数组的长度, 而利用 High 函数可获取动态数组的长度。

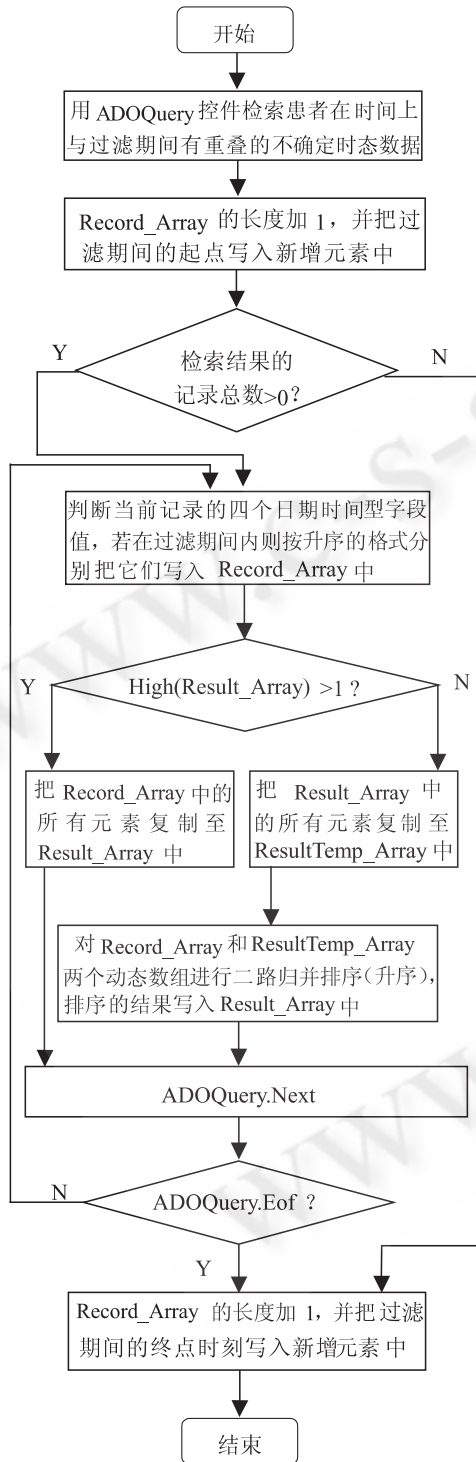


图 2 不确定时态数据期间过滤的时间子区间提取算法流程

(3) 二路归并排序法: 对两个已经有序的源数组来说, 若希望生成一个包括这两个源数组中所有元素的数组, 并保证新数组中的元素有序, 则可通过二路归并排序法来完成, 这是一种 $O(n \log(n))$ 的算法。它的工作原理是, 首先对两个源数组的第一个元素进行比较, 将小者复制到目标数组中, 并使用计数器来维护各个数组的元素指针 (即被复制的源数组和目标数组的计数器同时递增 1), 继续以此方式处理, 直到某个源数组已处理完毕, 然后把另一个源数组的剩余元素复制到目标数组中。

限于篇幅, 这里仅给出二路归并排序法的主要源码:

```

var
    { 声明三个时间日期类型的动态数组 }
    Record_Array, ResultTemp_Array, Result_Array : array of TDateTime;
    { 动态数组的长度变量 }
    ArrayLength : integer;
    { 三个计数器变量置 0 }
    inx1 : = 0; inx2 : = 0; inx3 : = 0;
    ArrayLength : = 0;
    { 用二路归并排序法对 Record_Array 和 ResultTemp_Array 两个动态数组进行升序排序, 排序结果写入至 Result_Array 中 }
    while (inx1 < High ( ResultTemp_Array ) + 1) and
        (inx2 < High ( Record_Array ) + 1) do
    begin
        ArrayLength := High ( Result_Array ) + 1;
        { 调整动态数组的长度 }
        Setlength ( Result_Array, ArrayLength );
        { 从两个有序源数组中找到较小的元素, 并将该元素复制到目标数组中; 令 inx1 或 inx2 和 inx3 索引递增 }
        if ResultTemp_Array [ inx1 ] < Record_Array [ inx2 ] then
        begin
            Result_Array [ inx3 ] := ResultTemp_Array [ inx1 ];

```

```

    inc( inx1 );
end
else
begin
    Result_Array [ inx3 ] := Record_Array
[ inx2 ];
    inc( inx2 );
end;
inc( inx3 );
end;
{ 当某个有序动态数组处理完毕后,循环结束;
若第一个源表中还有剩余的元素,则予以复制}
if ( inx1 < High( ResultTemp_Array ) + 1 ) then
for i := inx1 to High( ResultTemp_Array ) do
begin
    ArrayLength := High( Result_Array ) + 2;
    Setlength( Result_Array, ArrayLength );
    Result_Array[ inx3 ] := ResultTemp_Array
[ i ];
    inc( inx3 );
end
else
for i := inx2 to High( Record_Array ) do
begin
    ArrayLength := High( Result_Array ) + 2;
    Setlength( Result_Array, ArrayLength );
    Result_Array[ inx3 ] := Record_Array[ i ];
    inc( inx3 );
end;

```

求出各个独立的时间子区间后,它们所对应的不确定时态信息便可参照上述的瞬时过滤方法来获取,这里就不再重复,图 3 是电子病历中病情记录不确定时态数据的期间过滤的程序运行示意图。

4 结论

构建医院的电子病历信息系统是一项艰巨而复杂

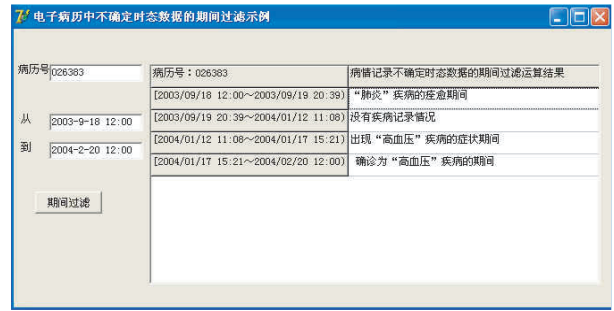


图 3 不确定时态数据的期间过滤的程序运行示意图

的工作,涉及的内容及技术很多。本文以电子病历的病情记录为例,介绍了不确定时态数据的概念及表示方法,并在 Delphi 编程环境中深入讨论了不确定时态数据的时态过滤运算的设计原理,从而有效地增强了医护人员查阅患者医疗信息的直观性。

参考文献

- 何新贵. 特种数据库技术. 北京: 科学出版社, 2000. 1 - 36.
- 汤庸. 时态数据库导论 [M]. 北京: 北京大学出版社, 2004. 1 - 121.
- 蔡启先. 数据的时态性及其在非时态 DBMS 上的处理. 计算机应用, 2000, 20(12): 23 - 27.
- 蔡启先, 唐新来. 基于历史关系模型的时态过滤运算. 广西工学院学报, 2003, 14(3): 16 - 20.
- 蔡启先. Key - Time 法及其用于时态过滤运算的研究. 广西工学院学报, 2005, 16(2): 34 - 38.
- 郑城荣, 曾凡奎. Delphi 运行时间库 RTL 和组件库 VCL 技术参考. 北京: 人民邮电出版社, 1999. 239 - 245.
- 黄雄波. 电子病历中时态数据库的分析与设计. [硕士学位论文], 广州: 华南理工大学 2007.
- Julian Bucknall 著; 林琪, 朱涛江译. Delphi 算法与数据结构. 北京: 中国电力出版社, 2003. 109 - 148.