

基于有色 Petri 网的变电站监控数据版本控制建模与研究

时 谊¹, 余 维^{2,3}, 张 丹², 李江林¹, 叶阳东^{2,3}

(1. 许继电气股份有限公司, 河南 许昌 461000; 2. 郑州大学信息工程学院, 河南 郑州 450000;
3. 北京交通大学轨道交通控制与安全国家重点实验室, 北京 100044)

摘要: 在大型设备的状态数据库中, 数据版本的管理不善会造成系统的混乱甚至崩溃。针对这些问题, 通过对已有版本控制方法的分析比较, 设计了一种新的数据版本控制方法。该方法构建一种有色 Petri 网 (CPN), 模拟数据版本控制过程, 用不同颜色的托肯表示数据的不同版本, 通过限制变迁的触发序列模拟数据版本的更新过程, 应用 Petri 网的并发机制和 SQL Server 并发机制解决了版本更新过程中的并发、合并等问题。通过变电站监控系统的实例得出: 该方法可用于中、大型系统的数据版本控制, 有效提高了历史数据的管理水平。

关键词: 版本控制; Petri 网; 有色 Petri 网; 版本更新; 变电站

Modeling and research on substation SCADA data versions control based on colored Petri net

SHI Yi¹, SHE Wei^{2,3}, ZHANG Dan², LI Jiang-lin¹, YE Yang-dong^{2,3}

(1. XJ Electric Company Limited, Xuchang 461000, China; 2. School of Information Engineering, Zhengzhou University, Zhengzhou 450000, China; 3. State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China)

Abstract: In large equipment state database, the poor management of data versions can cause the system disorder and collapse. To solve the problems, through the analysis and comparison of the existing version control methods, this paper designs a new method which simulates data versions control process by building a colored Petri net. Different colors of tokens show different versions. The forming process of data version is simulated by limiting the firing sequence. Concurrent mechanism of Petri net and SQL server is adopted to solve concurrent and incorporative problems in the process of update. Through the simulation with a real example of substations supervision system, it is concluded that this method can be used in middle and large system and improve the management level of the historical data.

This work is supported by National High-tech R & D Program of China (863 Program) (No. 2011AA110501) and National Natural Science Foundation of China (No. 61170223).

Key words: version control; Petri net; colored Petri net; version update; substation

中图分类号: TM273 文献标识码: A 文章编号: 1674-3415(2014)04-0098-07

0 引言

在大型设备的状态数据库^[1]中, 尤其是数据经常需要更新的系统, 如果数据版本管理不善将会造成系统的混乱甚至崩溃, 因此数据版本控制^[2-4]尤为重要。数据版本控制技术是对每次修改都做详细记录, 尤其当多人同时开发一个数据库管理和应用系

统时, 采用合理的方法监控数据库中表、存储过程和视图等对象的变动非常重要。数据版本控制的核心是变更管理, 主要目标和任务是记录并保存每个安全可靠的历史版本记录, 并且保证每个客户端得到的都是最新版本。

目前数据版本控制的主要方法有: 差分编码、档案上锁、中央系统与分布式系统等^[5-7]。差分编码^[5]只保留档案相继版本之间的差异, 该方法可以有效地储存数个版本的档案, 节省大量空间, 但是采用该方法时, 版本恢复比较麻烦。档案上锁功能是高难度合并, 例如大幅更改大档案或档案群的许

基金项目: 国家 863 计划资助项目 (2011AA110501); 国家自然科学基金 (61170223); 国家重点实验室开放课题基金 (RCS2009K003); 河南省重点科技攻关计划项目 (122102210004); 湖南省科技攻关计划项目 (2012FJ3092)

多部分, 提供了一些保护, 但其他开发者可以绕过版本控制系统改变档案, 这本身就存在很大问题。所以档案上锁功能带来的功效与副作用一直饱受争议。中央式系统是当多个客户端同时要改变同一个档案时, 由中央权威管理存取权限“锁上”档案库中的档案, 一次只让一个开发者工作, 或是像分布式系统容许多个档案库开发者同时更动同一档案, 而不需经中央权威许可, 分布式系统仍然可以有档案上锁功能。

现有的几种版本关系模型中线性版本模型能够很好地描述版本顺序产生的过程, 但不能区分替换版本和修订版本的区别; 树状版本模型虽然解决了对对象模型的修订版本与替换版本之间的区别, 但是不能描述版本合并这一实际情况; 有向无环图尽管支持版本合并和变更的历史信息描述, 但只能用节点序号描述版本的产生层次和来源, 无法表示该版本的逻辑层次性, 而且容易产生歧义; 多色图的版本模型^[8]只考虑了同一版本的修订和变型关系。

立足以上模型的优点与不足, 针对需要频繁修改数据的变电站监控系统^[9-10], 本文运用 Petri 网^[11]强大的建模能力, 尤其是描述系统并发、异步性的优势, 对数据版本控制过程进行建模, 采用有色 Petri 网为不同性质的托肯赋予不同的颜色以表示数据的不同版本, 从而模拟数据版本更新的过程。该方法可解决由于多人并发修改导致的数据不一致问题, 并且可以实现版本的冲突检测和对冗余版本的合并。通过变电站监控系统的实例证明了该方法的有效性。

1 有色 Petri 网

有色 Petri 网 (Colored Petri Nets, CPN)^[12-13], 是定义在基本 Petri 网的基础上, 对库所中的托肯着以不同的颜色, 其实质就是对托肯进行分类, 以实现网系统的折叠。在一个实际系统的有色 Petri 网模型中, 托肯往往代表某种资源, 给托肯赋予一种称之为颜色的内禀属性, 即不同颜色的托肯代表不同类型的资源。库所是存放同一资源的地方, 变迁的发生表示对从输入库所取来的托肯进行加工, 变成新颜色的托肯后再传递给输出库所。CPN 的两大特点: 托肯是有颜色的; 变迁的发生可以改变托肯的颜色。CPN 大大减少了托肯和变迁的数量, 是普通 Petri 网的折叠^[14-16]。本文在文献^[15-16]基础上给出一种有色 Petri 网的形式化定义。

1.1 有色 Petri 网定义

定义 1 有色 Petri 网 (CPN) 模型是一个九元组 $\Sigma=(P,T,F,M,D,C,I,G,K)$, 其中

- 1) $P = \{p_1, p_2, p_3, \dots, p_n\}$ 是库所的有限集;
- 2) $T = \{t_1, t_2, t_3, \dots, t_m\}$ 是变迁的有限集;
- 3) $F = (P \times T) \cup (T \times P)$ 是网中有向弧集合;
- 4) D 为颜色集;
- 5) $C: P \cup T \rightarrow \Psi(D)$, $\Psi(D) = \{U | U \in D\}$ 是颜色集 D 的幂集, 其中 $\forall p \in P, C(p)$ 是库所 p 中可能的托肯色集合, $\forall t \in T, C(t)$ 是变迁 t 上可能的出现色集合;
- 6) $M: P \rightarrow D_{MS}$ 是网的标识, 标识 M_i 下的标识分量 $M_i(p)$ 表示库所 p 中的托肯颜色集上的多重集, M_0 是初始标识;
- 7) G : 变迁护卫函数, $\forall t \in T: Type(G(t)) = bool \wedge Type(Var(G(t))) \subseteq D$, $bool = \{true, false\}$, 当且仅当该变迁的护卫函数类型为真时变迁才可能触发;
- 8) I 为 F 上的权函数, $I: F \rightarrow D_{MS}$, 确定迁发生后托肯的变化;
- 9) K 为 P 上的容量函数, $K: P \rightarrow \{1, 2, 3, \dots\}$ 如果库所 P 没有定义容量函数, 则默认 $K(p) = \infty$ 。

其中: D_{MS} 表示定义在 D 上的所有有限多重集的集合^[6]; $Type(v)$ 表示变量 v 的类型; $Var(expr)$ 表示表达式 $expr$ 的变量集合^[12]。

库所中托肯的颜色用来描述对象的属性, 其属性值含有多个参数, $M(P) = (\langle X, Y, \dots \rangle, \langle X, Y, \dots \rangle \dots)$ 。 $\langle X, Y, \dots \rangle$ 代表 1 个含有不同参数的属性的托肯。例如: 表示某一版本的托肯其属性为 (版本号 num , 父版本号 $fnum$, 时间 $time$, 内容 $context$, 操作人 $name$)。若库所 P_i 中有一个该种属性的托肯, 则其标识可记作: $M(p_i) = (\langle v1.0.1, v1.0.0, 201212121210, 0, Jim \rangle)$, 时间用 12 位数字表示, 精确到分钟, 201212121210 表示 2012 年 12 月 12 日 12 时 10 分。

定义 2: CNP 的变迁触发规则: $\forall t \in T$ 当且仅当 $\forall p \in \bullet t: M(p) \geq I(p, t)$, 且 $G(t) = true$ 时, 变迁 t 触发, 记作 $M[t >$; 在 M 下是使能的变迁 t 触发, 可以得到后继表示 M' , 且

$$M'(p) = \begin{cases} M(p) + I(t, p), & p \in t \bullet - \bullet t \\ M(p) - I(p, t), & p \in \bullet t - t \bullet \\ M(p) - I(p, t) + I(t, p), & p \in \bullet t \cap t \bullet \\ M(p), & else \end{cases}$$

说明: 这里的加减运算指的是多重集上的加减。 $\bullet t$ 表示变迁 t 的前集或输入集, 且 $\bullet t = \{p | (p, t) \in F, p \in P\}$; $t \bullet$ 表示变迁 t 的后集或输出集, 且 $t \bullet = \{p | (t, p) \in F, p \in P\}$ 。

定义 3: CPN 中, $\forall t \in T, p \in \bullet t, p' \in t \bullet$, 变迁颜色处理函数, $H:I(p,t) \rightarrow I(t,p')$ 。

规则 1: 若 $K(p)=1$, 且 $|M(p)|=1$, 若又有变迁发生对该库所产生新的托肯时, 则覆盖已有托肯, 以保证库所的托肯是最新的。 $|M(p)|$ 表示库所 p 中托肯的数量。

规则 2: 若某变迁护卫函数初始值设为: $Type(G(t))=flouse$, 在经过某些判定后变为 $Type(G(t))=true$, 则该变迁触发一次后自动复位为 $Type(G(t))=flouse$; 若初始 $Type(G(t))=true$, 则无论该变迁触发多少次护卫函数保持不变。

1.2 简单的有色 Petri 网模型

根据有色 Petri 的定义, 对数据某一版本进行一次简单的更新操作而产生新的版本, 将这一过程进行建模。用库所表示版本存放库, 变迁表示更新操作, 托肯表示版本的差异信息, 由此建立一个简单的有色 Petri 网模型, 表示数据版本的变化和记录控制过程, 如图 1、图 2 所示。

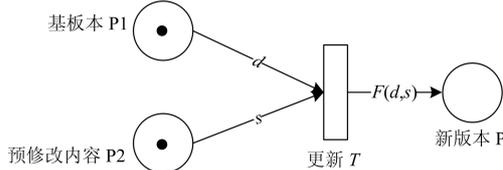


图 1 基本有色 Petri 网模型
Fig. 1 Foundation CPN model

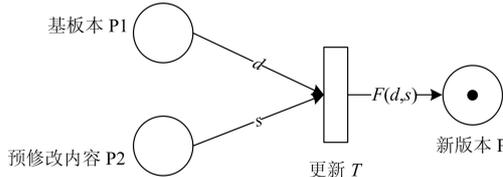


图 2 变迁 T 发生后有色 Petri 网模型
Fig. 2 CPN after T fired

其中: d 是 Num 类型的值, 表示一个整数, 即版本号; s 是 $Context$ 类型的值, 表示字符串, 即更新内容; $F(d,s)$ 函数表示: 若 $G(T)=true$, 则 $F(d,s)=\langle d++,s,T.time \rangle$, 否则 $F(d,s)=null$ 。其中 $time$ 为变迁 T 发生时获取的系统时间, $null$ 为空。

假设目前版本号为 1, 预修改内容为 $delete\ Table\ 1$, 此时 $G(T)=true$ 。初始化模型, 则有 $M(P1)=(1)$; $M(P2)=(delete\ table\ 1)$; 经过变迁 T 的触发, 产生新的复合色的托肯在 P 中, $M(P)=(\langle 2,delete\ table\ 1,201212151010 \rangle)$ 。库所 P 中的托肯就是产生的新版本, 复合色表示了版本号, 更新内容和时间。此例仅表示简单情况, 形象说明有色 Petri 网中托肯颜色的意义, 可以根据不同系统的需要修改复合色的内容。复合色既可以代表一种

属性也可以代表多种。

该模型简单, 形象且直观, 很好地模拟数据更新过程中版本控制管理的流程, 这也是基于有色 Petri 网建立数据版本控制模型的核心思想。该例仅表示了一次更新过程, 然而实际修改中还涉及到多人修改, 多次修改, 重复修改等情况, 因此系统数据版本的控制模型还需在该简单模型基础上优化完善。

2 基于有色 Petri 网的数据版本控制模型

2.1 数据模型和数据版本

数据模型^[17]是对现实世界数据特征的抽象, 用来描述数据, 组织数据, 对数据进行操作。建立数据模型的优点是能比较真实地模拟现实世界, 更容易为人所理解, 也便于在计算机上实现。

数据模型应用目的不同, 可以分为两类, 概念模型和逻辑模型。概念模型是按用户的观点来对数据和信息建模, 用于数据库设计; 逻辑模型包括层次模型, 网状模型, 关系模型, 面向对象模型和对象关系模型是按计算机系统的观点对数据建模。

面向对象数据模型克服了传统数据模型的局限性, 将具有复杂结构的一个逻辑整体视为一个对象, 提供了概念模型到逻辑数据模型以至物理模型的一致描述, 从而大大提高了管理效率, 同时也为版本管理、动态模式修改等功能的实现创造了条件。

数据版本是以对象为单位, 指数据库某一特定状态的数据记录, 创建数据库的某一版本就是选择了数据库的某一状态。通过这些版本可以任意比较不同版本之间的差异。数据库管理系统(DBMS)通过维护各个版本的状态信息使用户在各版本中进行独立事务处理而互不影响。

2.2 数据版本控制的有色 Petri 网模型

版本引用采用动态引用(静态引用是每个引用都固定指向一个版本), 即每次都是在最新的版本的基础上进行修改更新操作。版本存储用独立存储与差值存储相结合的方式。独立存储便于下次更新, 而差值存储节约空间, 采取两者结合的方式蕴含两者优点。可以在多个差值存储后进行一次独立存储, 根据差值和完整版本就可以得到最新的版本。

数据版本控制所要实现的功能是: 每次修改获取最新版本, 保存不同版本, 比较不同版本之间数据的差异, 对冗余版本进行合并。

Petri 网具有强大的建模能力, 尤其是描述系统并发、冲突、同步等关系。有色 Petri 网中的库所可以表示版本的状态, 变迁表示更新操作, 库所中托肯的不同颜色代表了不同版本的差异记录和相关信

息。由以上规则建立数据版本控制静态模型如图 3 所示。

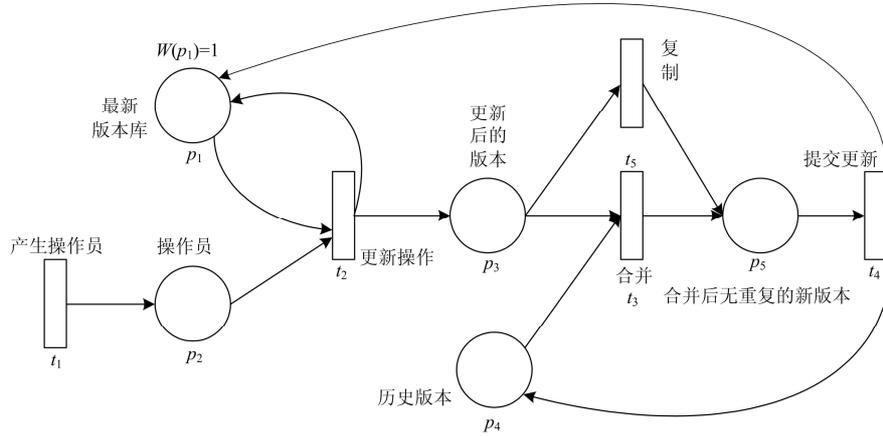


图 3 版本控制有色 Petri 网模型

Fig. 3 CPN model of versions control

2.3 模型分析

2.3.1 库所及托肯颜色的含义

该有色 Petri 网模型的库所分为两类： $\{p_1, p_3, p_4, p_5\}$ 和 $\{p_2\}$ 。其中 $\{p_1, p_3, p_4, p_5\}$ 表示版本库， $\{p_2\}$ 是操作员信息。表示版本库的库所中的托肯颜色为 $C(p_1) = \{c_1, c_2, \dots, c_n\}$, $c_1, c_2, \dots, c_n \in D$ ，其中 c_i 表示版本的差异信息，由含有多个参数的一元组组成。例如，属性若包括版本号，父版本号，修改时间，修改的内容，修改人，则 c_i 表示为 $c_i = \langle num, fnum, time, context, name \rangle$ 。操作员信息中托肯颜色为 $C(p_2) = \{m_1, m_2, \dots, m_n\}$ ， $m_1, m_2, \dots, m_n \in D$ ，其中 m_i 表示操作员信息，如操作员编号，操作员姓名，要更新的内容，则 $m_i = \langle mnum, mname, mupdate \rangle$ 。

2.3.2 变迁

t_1, t_2, t_3, t_4, t_5 为五类不同的变迁。

t_1 为源变迁，即 $\bullet t_1 = null$ ，提供更改数据的操作人员的信息。初始时 $G(t_1) = false$ 。当有操作员要对数据修改，且通过权限，即是否有资格对数据库修改，此时 $G(t_1) = true$ ，该变迁触发，产生一个带有该操作员信息的颜色为 m_i 的托肯，通过变迁处理， $m_i.mnum = t_1.mnum$, $m_i.mname = t_1.mname$, $m_i.mupdate = t_1.mupdate$ 。

t_2 为更新操作变迁，初始 $G(t_2) = false$ 。当 p_1 中有托肯，设颜色为 c_i ， p_2 中也有托肯，设颜色为 m_i ，且需要进行更新操作时，由 SQL Server 并发机制，对要修改的内容加锁，检测是否进行同一内容修改，从而避免造成脏数据和数据不一致。如无冲突， $G(t_2) = true$ ，变迁 t_2 触发，产生颜色为 c_i 和 c_k 两个托肯，分别存于 p_1 和 p_3 。 c_i 是原托肯的复制，便于下次更新操作； c_k 为新颜色的托肯，颜色的变化如

$c_k.fnum = c_i.fnum$ ， $c_k.time = t_2.time$ ， $c_k.update = m_i.mupdate$ 。若检测有冲突则 $G(t_2) = false$ ，变迁不触发。

对版本编号的设置 $V.X.Y.Z$ ，其中 X 表示主版本号， Y 表示子版本号， Z 表示修正版本号。初始版本号定义为 $V.1.0.0$ ，在进行局部少量修改后，修正版本号 Z 加 1，主版本号和子版本号均不变， $V.1.0.1$ ；当数据库中增加表信息时，子版本号加 1，主版本号不变，修正版本号复位 0， $V.1.1.0$ ；当数据库多处局部变化导致整体改变时，主版本号加 1，子版本号和修正版本号不变， $V.2.0.0$ 。

t_3 为冗余版本的合并。初始 $G(t_3) = false$ ，只有当 p_3, p_4 中托肯有冗余现象时， $G(t_3) = true$ 。 t_5 为版本的复制， $G(t_5) = true$ 恒成立。但是，当 p_3, p_4 中都有托肯，且 t_3 与 t_5 护卫函数都为真时， t_3 与 t_5 是冲突的，为避免冲突，增加一个控制条件，使得 t_3 的优先权高于 t_5 ， t_3 总是优先 t_5 触发。产生的托肯的颜色变化类似 t_2 。

t_4 是提交更新操作，初始 $G(t_4) = false$ ，当提交成功时 $G(t_4) = true$ ，将 p_5 中的托肯复制成 2 个，一个存放在 p_1 ，由于 p_1 的托肯容量为 1，根据规则 1，新产生的托肯将覆盖已有的托肯，以保证数据修改时得到的是最新版本，此托肯代表的版本用于下次更新，一个存放在 p_4 用于存储同时用于合并比较。

弧上的权函数 $I(t, p)$ 为正数， $I(p, t)$ 为负数) 见表 1。

ID 为恒同映射，表示参与的库所的托肯色与变迁的出现色一样； $F1$: if $G(t_1) = true$ then $\langle p_{1,r1} + m, p_{1,r1}, t_1.time, p_{2,r3}, p_{2,r1} \rangle$ else empty ; $F2$: if

$(p_{3r4} = p_{4r4})$ then $\langle p_{4r1}, p_{4r2}, p_{4r3}, p_{4r4}, p_{4r5} \rangle$ else empty。其中, p_{irk} 表示库所 p_i 的托肯复合色中第 k 种颜色的值。

表 1 弧上的权函数
Table 1 Weight function of arc

	t_1	t_2	t_3	t_4	t_5
含义	产生操作	更新操作	合并	提交更新	复制
p_1 最新、版本库		$-ID+ID$		ID	
p_2 操作员	ID	$-ID$			
p_3 更新后的版本		$F1$	$-ID$		$-ID$
p_4 历史、版本			$-ID$	ID	
p_5 合并后无重复的新版本			$F2$	$-ID$	ID

2.3.3 初始标识与终止条件

该网的初始标识为, p_1 、 p_4 有相同颜色的托肯各一个, 表示最初的基版本, 其他库所无托肯。当操作员开始对数据操作时操作员库所 p_2 产生托肯, 此时网按照变迁触发规则运行。库所 p_4 中托肯的信息相当于在数据库中建立了一个版本控制表, 记录一系列的版本变化信息, 可能包括数据库元素表、存储过程、视图、触发器的版本。 p_4 对版本的存储相当于差异存储, 为方便数据恢复等操作, 可以根据需要, 设置 p_4 的容量上限, 当 p_4 中托肯数量达到上限时终止网的运行。积累一定数量差异信息整理合成后产生完整的版本, 由此将差值存储与独立存储相结合。

该模型直观、灵活、容易理解且便于操作, 能有效管理版本和查阅备份版本, 实现了数据的一致性和完整性, 对冗余版本的合并避免了空间的浪费。

3 应用实例

在变电站监控系统^[18]中, 由于电网结构或运行方式变化、新建或备用线路投运, 以及技改项目实施等原因的影响, 都需要根据工作要求对数据库进行备份修改等操作。如馈线的三相电流、母线电压需要经常修改。但是如果修改次数繁多, 不能对修改的版本有效管理控制, 会造成不堪设想的后果。

以该变电站监控系统的数据库为例, 建立有色 Petri 网模型, 其网的初始状态为数据库的原始状态, 中间某状态为更新后的状态, 网的终止状态直到对数据的更新停止。

算例: 假设编号为 002 的操作员要对逻辑节点表 LD 中低压闭锁定值(U_{dybs})和过流 I 端保护时间值

(T_{dz1})进行修改, 初始状态如图 4 所示。

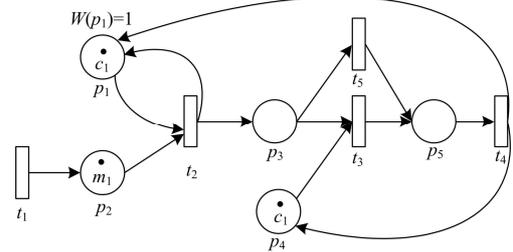


图 4 有色 Petri 网初始模型

Fig. 4 CPN model of initial state

版本库 $\{p_1, p_3, p_4, p_5\}$ 中托肯的属性值包含版本号, 父版本号, 时间, 内容, 操作人。 p_1 和 p_4 中有颜色为 c_1 的托肯。由于是初始版本, 除版本号外其他字段均为空, 表示为 $M(p_4) = \langle \langle V1.0.0, 0, 0, 0, 0, 0 \rangle \rangle$, $M(p_1) = \langle \langle V1.0.0, 0, 0, 0, 0, 0 \rangle \rangle$; 操作人员信息库 p_2 中托肯的属性包括操作人员的编号和姓名, 编号为 002 姓名为 sss 的操作员要对数据修改, 进行更新操作, 对逻辑节点表 LD 中低压闭锁定值(U_{dybs})和过流 I 端保护时间值(T_{dz1})进行修改。因为该操作员有权利对数据库进行修改, $G(t_1) = true$, 根据变迁触发规则, 变迁 t_1 触发, 库所 p_2 得到一个颜色为 m_1 的托肯, $M(p_2) = \langle \langle 002, sss, update LD set value = 0.21 where name = Udybs update LD set value = 51 where name = Tdz1 \rangle \rangle$ 。此时 p_1 、 p_2 中均含有 1 个托肯, 且由 SQL Server 并发机制得知该更新信息合法, $G(t_2) = true$, 变迁 t_2 可以触发, 得到后继网如图 5 所示。

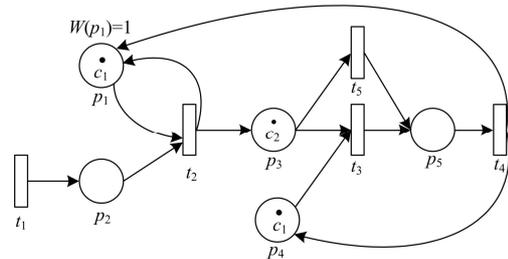


图 5 变迁 t_2 发生后有色 Petri 网模型

Fig. 5 CPN model after t_2 fired

变迁 t_2 发生同时获取系统时间, 此时, $M(p_1) = \langle \langle V1.0.0, 0, 0, 0, 0, 0 \rangle \rangle$, $M(p_3) = \langle \langle V1.0.1, V1.0.0, 201212151010, update LD set value = 0.21 where name = Udybs update LD set value = 51 where name = Tdz1, 002 \rangle \rangle$ 。此时, p_4 中托肯的第四种属性值即更新内容与 p_3 中的不同, 不需要合并去除冗余等操作, $G(t_3) = false$, 变迁 t_3 不发生, $G(t_5) = true$, 变迁 t_5 发生, p_3 中的托肯单纯的复制到 p_5 , 如图 6 所示。

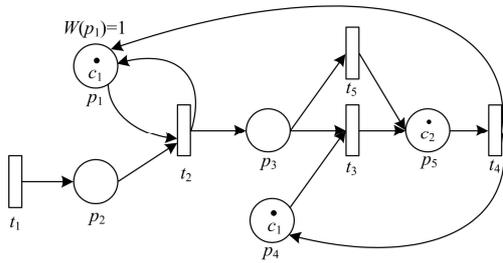


图 6 变迁 t_5 发生后有色 Petri 网模型

Fig. 6 CPN model after t_5 fired

此时, $M(p_5) = (< V1.0.1, V1.0.0, 201212151010, update LD set value = 0.21 where name = Udybs update LD set value = 51 where name = Tdz1,002 >)$, p_1 和 p_4 中的托肯保持不变。在此过程中, 若有其他操作员要对数据更新, 且拥有更新权限, 则变迁 t_1 继续触发, 实现了对数据的并行操作。

库所 p_5 有一个托肯, 且恒有 $G(t_4) = true$, 提交操作变迁 t_4 触发, 产生两个托肯分别存入库所 p_1 和 p_4 。因为 $W(p_1)=1$, 库所 p_1 中只能含有一个托肯, 由规则 1 保证每次更新的这一个托肯是当前最新的版本, 以备后继的更新操作。库所 p_4 存储所有的版本, 既方便查阅更新过程, 也可以在更新出错后快速恢复。 t_4 触发后网的状态如图 7 所示。

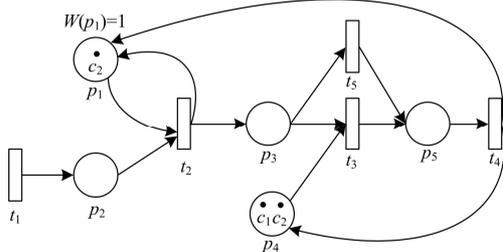


图 7 变迁 t_4 发生后有色 Petri 网模型

Fig. 7 CPN model after t_4 fired

此时 $M(p_5) = (< V1.0.1, V1.0.0, 201212151010, update LD set value = 0.21 where name = Udybs update LD set value = 51 where name = Tdz1,002 >)$, $M(p_4) = (< V1.0.0, 0, 0, 0, 0 >, < V1.0.1, V1.0.0, 201212151010, update LD set value = 0.21 where name = Udybs update LD set value = 51 where name = Tdz1, 002 >)$ 。

随着数据库不断更新, 历史版本库所 p_4 中的托肯越来越多, 状态标识 $M(p_4)$ 的矩阵越来越大, 每个托肯代表一次版本的更新操作。库所 p_4 相当于一个版本存储库, 既能快速查询数据, 又能在更新错误时快速恢复。

本文以变电站监控系统的数据库为应用背景, 该系统的数据版本控制完全依赖有色 Petri 网的控

制方法, 实验表明该方法实现了多人并发操作并控制了版本更新的过程, 保证了数据的一致性和完整性。

4 结语

该文提出运用有色 Petri 网对数据版本控制过程建模, 充分利用为托肯着色这一特点区分了各版本之间的差异。该方法用图的形式展示, 增加了直观形象性和易理解性。通过 Petri 网中托肯的不同颜色清晰地表示了各个版本之间的差异, 保证了每次修改都是在最新的版本基础上, 实现了多人并发修改, 并且保证数据的一致性和完整性。

变电站监控系统的数据库具有普适性, 代表了一类数据经常变动的中、大型数据库的特点。有色 Petri 网实现了对该系统数据的版本控制。以此推出该方法可以运用在所有数据需要经常修改的大型系统数据库中, 控制了因数据修改导致的数据不一致。保存的各版本之间的差异记录可以为操作人员对数据更新提供参考, 当数据出现问题时可以快速找出责任人, 并且及时恢复。

参考文献

- [1] 王军庄, 常鲜戎, 顾卫国. 基于 OCL 技术的 Oracle 数据库数据快速存取研究[J]. 电力系统保护与控制, 2009, 37(9): 53-56.
WANG Jun-zhuang, CHANG Xian-rong, GU Wei-guo. Research on quick access in Oracle database based on OCL technology[J]. Power System Protection and Control, 2009, 37(9): 53-56.
- [2] Ivan Milentijevic, Vladimir Ciric, Oliver Vojinovic. Version control in project-based learning[J]. Computers and Education, 2008, 50(1): 1331-1338.
- [3] Waltersdorfer F, Moser T, Zoitl A, et al. Version management and conflict detection across heterogeneous engineering data models[C] // Industrial Informatics (INDIN), 2010 8th IEEE International Conference on, Osaka, 2010: 928-935.
- [4] Armejach A, Seyedi A, Titos-Gil R. Using a reconfigurable L1 data cache for efficient version management in hardware transactional memory[C] // 2011 International Conference on Parallel Architectures and Compilation Techniques (PACT), Galveston, TX, 2011: 361-371.
- [5] 汪汇兵, 唐新明, 洪志刚. 版本增量式时空数据模型研究[J]. 测绘科学, 2006, 31(5): 133-136.
WANG Hui-bing, TANG Xin-ming, HONG Zhi-gang. Research on spatial-temporal data model in

- version-difference structure[J]. *Science of Surveying and Mapping*, 2006, 31(5): 133-136.
- [6] 阳万安, 李彦. 通用版本控制系统的研究和设计[J]. *计算机工程*, 2008, 34(12): 283-285.
YANG Wan-an, LI Yan. Research and design of general version control system[J]. *Computer Engineering*, 2008, 34(12): 283-285.
- [7] 李辉, 吴立言, 刘更, 等. 基于版本控制技术的机械仿真数据管理[J]. *机械设计与制造*, 2010(9): 236-238.
LI Hui, WU Li-yan, LIU Geng, et al. Management of mechanical simulation data based on version control technology[J]. *Machinery Design and Manufacture*, 2010(9): 236-238.
- [8] 李建民. 基于知识和多色图论的 PDM 版本管理研究[D]. 南京: 南京航空航天大学, 2008.
LI Jian-min. The research of PDM version management based on knowledge and multicolor-map[D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2008.
- [9] 朱意霞, 邓建慎, 邱俊宏, 等. Oracle 监控维护技术在变电站监控系统中的应用[J]. *电力系统保护与控制*, 2010, 38(11): 135-139.
ZHU Yi-xia, DENG Jian-shen, QIU Jun-hong, et al. Application on the monitoring and maintenance of Oracle in substation monitoring system[J]. *Power System Protection and Control*, 2010, 38(11): 135-139.
- [10] 张信权, 彭军林, 李少卿, 等. 基于 GOAHEAD、SVG 和 PRFIS 的变电站远程 Web 监控系统设计[J]. *电力系统保护与控制*, 2011, 39(12): 130-133.
ZHANG Xin-quan, PENG Jun-lin, LI Shao-qing, et al. A scheme of station long-distance web SCADA based on GOAHEAD, SVG and PRFIS subsystem[J]. *Power System Protection and Control*, 2011, 39(12): 130-133.
- [11] Biel J I L, Macias E J, Perez de la Parte M. Simulation-based optimization for the design of discrete event systems modeled by parametric Petri nets[C] // *Computer Modeling and Simulation (EMS) 2011 Fifth Uksim European Symposium on*, 16-18 Nov, 2011: 150-155.
- [12] 袁崇义. Petri 网原理与应用[M]. 北京: 电子工业出版社, 1998: 97-103.
YUAN Chong-yi. Petri net theory and application[M]. Beijing: Publishing House of Electronics Industry, 1998: 97-103.
- [13] Westergaard M, Maggi F M. Modeling and verification of a protocol for operational support using coloured Petri nets[C] // *Proceedings of the 32nd International Conference on Applications and Theory of Petri Nets, PETRI NETS 2011, Newcastle, UK, 2011*: 268-287.
- [14] 宫世豪, 杨吉江, 柴跃廷, 等. 基于有色 petri 网的经营过程建模[J]. *信息与控制*, 2000, 29(1): 1-5.
GONG Shi-hao, YANG Ji-jiang, CHAI Yue-ting, et al. Business process modeling based on colored Petri nets[J]. *Information and Control*, 2000, 29(1): 1-5.
- [15] 曹宜英, 秦现生, 张晓云. 基于多色 Petri 网的产品开发过程建模[J]. *系统工程理论与实践*, 2010, 30(5): 909-918.
CAO Yi-ying, QIN Xian-sheng, ZHANG Xiao-yun. Product development process modeling based on polychromatic Petri net[J]. *System Engineering Theory and Practice*, 2010, 30(5): 909-918.
- [16] ZHOU Guo-xiang, GAO De-ping. ECA rule and colored Petri nets based workflow modeling research[C] // *The National Natural Science Foundation of China 2010 IEEE*, 2010: 1-4.
- [17] 王山, 萨师焯. 数据库系统概论[M]. 北京: 高等教育出版社, 2006: 12-28.
WANG Shan, SA Shi-xuan. Introduction to database system[M]. Beijing: Higher Education Press, 2006: 12-28.
- [18] 杨襄, 黎绍发. 变电站资料管理的系统模型设计与研究[J]. *计算机技术与发展*, 2010, 20(8): 229-233.
YANG Xiang, LI Shao-fa. Design and realization of substation data management system model[J]. *Computer Technology and Development*, 2010, 20(8): 229-233.

收稿日期: 2013-01-25

作者简介:

时 谊(1977-), 男, 本科, 工程师, 研究方向为电力系统自动化和继电保护;

余 维(1977-), 男, 博士研究生, 副教授, 从事智能系统和 Petri 网理论的研究; E-mail: shewei1977@163.com

张 丹(1986-), 女, 硕士研究生, 从事智能系统和 Petri 网理论的研究。