XML中的主动规则及其可终止性分析

李 晖 孔兰菊 洪晓光

(山东大学计算机科学与技术学院·济南250100)

摘 要 主动XML系统一般采用触发器,即"事件-条件-动作"(ECA)规则来提供主动行为。本文提出了一种新的事件监测机制,在XML系统中引入'主动节点',即把规则也融入节点,各节点上的ECA规则只需在节点修改时被激活并进行检查,提高了动作的执行效率,增强了系统的实时性。本文结合规则实例给出了分析规则终止性的静态判定算法,引入触发图、活化图、修改后的触发图、触发环等概念,对触发图中的简单触发环进行转换,产生一个对应于触发环的循环语句,对触发环中每一个被修改的节点产生一个递归等式。展开递归等式检验它的可满足性可以用来分析规则集的可终止性。这一算法提高了可终止性判定的精确性,降低了复杂度。

关键词 XML,OQL,ECA主动规则,可终止性,递归等式

Active Rules and thier Termination in XML

LI Hui KONG Lan-Ju HONG Xiao-Guang

(College of Computer Science and Technology, Shandong University, Jinan 250100)

Abstract XML is rapidly emerging as the most widely adopted technology for information representation and exchange in the applications. By means of active rules, it becomes possible to generate or manipulate the content of an XML document as a reaction to changes that occur to XML information. In this paper, we introduce a new rule execute model in active class in the XML environment. We discuss the rule analysis theory and propose an analysis approach, which can decide whether the rule sets can be terminated or not. Node, which forms the XML document, is instance of the active class. When the node is modified and the rule's condition is satisfied, the rule's action will be acted. There is no event listener with which applications often become inefficient and unmanageable when the number of triggers becomes large. In this paper we derive a recursion equation for each node that is modified in the cycle. Unfolding of the recursion equations and testing for the satisfiability are used to analyze the termination of the rule. Apart from the improved precision it can achieve, we are able to make statements about conditional termination properties of an ECA rule set.

Keywords XML, Active Rule, Termination, Equation

1 引言

日益普及的XML(eXtensible Markup Language),即可扩展的标识性语言,是一种新出现的互联网数据表示及数据交换的标准,其无模式、自描述的特点,能很好地描述网上数据;其数据结构、数据含义及数据本身又是可分析得到的,这就为我们使用传统数据库技术做好了准备。随着应用领域的不断扩大,越来越多的应用要求管理系统能够自动对XML文档内部与外界的状态进行监控分析,并做出实时响应。在没有用户干预的情况下,能够自动地对系统内部或外部所产生的事件做出反应的XML数据系统被称为"主动XML"。主动XML系统一般采用触发器,即"事件-条件-动作"(ECA)规则来提供主动行为。在本文中我们介绍了一种XML环境中的新的规则执行模型:主动节点,讨论了规则分析理论并提出了一种分析方法,它可以更好地断定规则集的可终止性。

2 主动规则运行机制

本文介绍了一种新的规则执行模型:主动节点。这种解决方案把规则当作节点的一部分,当节点被修改且满足规则条件时,执行规则的动作。它把规则与节点联合在一起管理,使得规则管理非常自然。

一个主动节点被描述为一个四元组:$AN=\{N,M,ECA,ANS\}$。

N是一个节点元素的集合,每个对象元素可以是复杂节点,这种复杂节点由简单对象以某种方式组成。M是一个可施行于N上的方法的集合。ECA表示在该节点类的节点收到消息时应主动激活执行的一个规则组。规则的事件被触发,规则的条件又得到满足,那么规则的动作将得到执行。

ANS是一个节点类名字的有限集合,它指明主动节点AN的每个超类。

由上述定义可见,各主动节点类按继承关系形成了一个具有多重继承关系的网状结构,对于节点类中的节点在被修改时都有可能激活一个事件监视器去检测一个规则组,从而主动激活执行一些预先设置的动作。主动节点区别于传统的节点,具有主动性。

2.1 主动规则表示方法

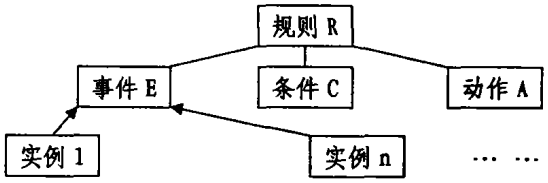

图1

规则、事件、条件和动作是预先定义的实例,该元类必须成为主动XML模式的一部分。

·事件:指要执行规则的操作。事件是跟发生变化的节点

联系在一起的。

·条件：由一个公共的抽象超类 Condition 来定义。类 Condition 主要包括条件的唯一名字，实现条件的方法名，实现条件的方法体，并且返回一个布尔值。

·动作：主要包括动作的唯一名字，实现动作的方法名，实现动作的方法体。动作可以视为发送节点的消息，为了创建动作，既要使用有关自身的信息，也要使用查询结果。

动作指要规则触发时所执行的动作。我们用 $\pm R \leftarrow \Phi$ 表示节点的插入与删除操作：Φ 是节点的操作表达式。数据状态的改变则可以表示为：插入时为 $\Phi \cup R$，删除时为 $\Phi - R$。

·规则：一个规则类将一个事件类、一个条件类和一个动作类聚集起来。

规则的定义包括规则的唯一名，它所基于的事件及条件 EC 耦合方式。一个规则一方面指定了事件实例化过程与条件计算间的耦合，另一方面也指定了事件实例化过程与动作执行间的耦合。规则包括了优先级信息来决定几个规则动作的执行顺序。

2.2 主动规则的可终止性

我们对触发环中每一个被修改的节点产生一个递归等式。展开递归等式检验它的可满足性可以用来分析规则集的可终止性。我们不仅仅得到了较高的精确度，而且可以得到一些判定可终止性的条件。

2.2.1 *主动规则处理方式* 规则处理采用的是一种迭代的行为方式：

(a)如果当前已经没有规则被触发，退出规则执行继续原来的事务处理。

(b)从当前规则集中选取优先级最高的规则。

(c)检验选定规则的条件是否满足。

(d)如果条件为真，执行选定规则的动作，返回到步骤(a)。

当执行规则动作时，如果该动作又触发了其他规则，则需要把被触发的规则加入当前规则集。如果当前规则所触发的规则中有优先级更高的规则，则当前规则被挂起，执行所触发规则中优先级最高的规则。最终，当没有规则再被触发时，算法就结束了；我们称它为规则运行的静止状态。然而，也可能出现规则不断的互相触发，没有尽头，永远达不到静止状态的情况。

2.2.2 *主动规则相关定义*

定义1(可终止性) 给定规则集，如果从任意数据状态触发的规则处理过程都肯定能终止，达到静止状态，则称规则集是可终止的，具有可终止性。

定义2(触发图) 任意规则集的触发依赖图定义如下：

(1)对于任意规则，在触发图中有节点与之一一对应。

(2)如果从 r_a 到 r_b 有一条弧，表示规则 r_a 的执行可能产生某事件，该事件触发 r_b。如果从 r_a 到 r_a 有一条弧，则称 r_a 为自触发规则。

(3)触发边用实线来表示。

定义3(触发环) 任意规则集触发图中的触发环定义如下：

图中从 v_0 到 v_k 的触发环是一系列顶点：$v_0, v_1 \cdots, v_k$，对于每个 $i(0 \leqslant i \leqslant k-1)$，有一条从 v_i 到 v_{i+1} 的弧，而且 $v_0 = v_k$。如果 $v_0, v_1, \cdots, v_k$ 各不相同，则该环为简单环。两个环等价是指对于两个环：$v_0, v_1, \cdots, v_k$ 和 $v'_0, v'_1, \cdots, v'_k$，对于 $i=0, \cdots, k-1$ 存在 j 使得 $v'_i = v_{i+j \bmod k}$。非简单环的环即复杂环。

定理1(修改后的触发图) 对触发图作以下一些修改：

(1)如果对于 r_x 所产生的同一事件可以同时触发规则 r_y、r_z，但是规则 r_y 的优先级高于规则 r_z 的优先级，则我们移去从 r_x 到 r_z 的弧。

(2)移去所有不属于任何环的顶点。

(3)在复杂环中合并只有一条入弧、一条出弧的环，用一个节点代替。

证明：修改(1)不会影响规则集的可终止性，因为 r_x 的优先级高于 r_z 的优先级 r_x 始终都只能触发 r_y。修改(2)不会影响规则集的可终止性，因为只有环才可能导致规则集的不可终止性。修改(3)不会改变规则集的可终止性。证毕。

对于给定触发图的环是否可终止，一个比较通用的认识如下：

·单调执行：对于环中的规则 r_i(a)规则条件中的节点数单调减少，或者(b)规则动作为删除时所选择的节点数单调减少，或者(c)规则动作为插入时所选择的节点数单调增加。

·有界执行：规则集开始执行后，无论文档在什么状态，所触发执行的规则不会超过一个固定的 k。

·非单调非有界执行。

我们注意到有界执行的实际结果等同于一个修改 XML 文档的有限序列。我们还注意到有界执行不一定单调。我们的算法在简单环里讨论单调执行和有界执行，在复杂环里讨论单调执行、有界执行和非单调非有界执行。

2.2.3 *主动规则集可终止性算法* 对于环 $v_0, v_1, \cdots, v_n$，假定规则 r_1 首先被触发，下面的程序可以表达其触发关系，可以用来分析其可终止性。(k 是一个给定的值)

```
tmp1=true
while tmp1 and i≤k
    tmp1= tmp2=…= tmpn=false
    for j=2 to n
        +tmpj←condition(rj-1) ∧ action(rj-1) ∧ tmpj-1
        if tmpj then exec(rj-1)
    end for
    +tmp1←condition(rn) ∧ action(rn) ∧ tmpn
    if tmp1 then exec(rn)
end while
```

定理2 假设我们有一个简单触发环，它已被翻译成循环程序，条件为 Φ。如果 $\Phi^i = \text{false}\ (i \leqslant k)$，则主动规则集可终止；如果 $\Phi^i = \text{true}(i > k)$，则它有可能不可终止。

证明：1. 正确性。用归纳法证明。

1)当 $n=1$ 时：如果 $\Phi^i = \text{false}(i \leqslant k)$，*i. e.* $condition(r_1) \wedge action(r_1) \wedge tmp_1$ 为假，则它不可能再次触发自身，因此可以终止使系统达到静止状态。定理1正确。

2)$n=t-1$，定理1是正确的，我们来证明 $n=t$ 时定理依然成立。由假设可知对于 $n=t-1$ 的情况，存在一个节点 $r_x(x < t-1)$ 使得 tmp_{x+1} 的值为假或者存在 r_{t-1} 使得 tmp_1 的值为假。如果 $x \leqslant t-1$ 我们得到规则不可能继续触发。如果 $n=t$ 我们有 $\Phi^i = \text{false}(i \leqslant k)$，也就是

$$condition(r_t) \wedge action(r_t) \wedge tmp_t \tag{1}$$

$$tmp_t = condition(r_{t-1}) \wedge action(r_{t-1}) \wedge tmp_{t-1} \tag{2}$$

我们用(2)式替换(1)中的 $tmpt$ 可得到

$$condition(r_t) \wedge action(r_t) \wedge tmp_t$$

$$= condition(r_t) \wedge action(r_t) \wedge condition(r_{t-1}) \wedge action(r_{t-1}) \wedge tmp_{t-1}$$

$$= \text{false}.$$

于是可以得到存在一个节点 $r_x(x < t)$ 使得 tmp_{x+1} 的值为假或者存在 r_t 使得 tmp_1 的值为假。因此没有规则能被触发，

系统状态不会继续发生变化，达到静止状态。定理1正确。

3)综上所述，定理2正确。

2. 算法终止性。假设循环不可终止。存在'i'($i \leqslant k$)使得 Φ' = false，也就是规则没有被触发，或规则的条件为假，或者规则的动作对系统状态没有影响。这就说明了没有规则可能被触发，循环状态不可能无限地持续下去，肯定是可终止的。如果 Φ' = true($i > k$)，规则集可能不可终止，但程序会因为 $i > k$ 而终止。

3. 算法复杂性。相对于以前所有节点的两两比较，程序的每次循环只需要计算 n 个节点，知道存在一个'i'使得 Φ' = false，或者 $i > k$ 退出循环，其复杂性为 $O(ni)$，比以前的 $O(n^2)$ 大大降低。证毕。

显然，循环次数越多，可满足性判定的代价就越高。如果两次以内都是可满足的，我们可以首先检验其单调性。如果循环中每个被修改的划分都是单调的，那么这个规则集是可终止的。我们还可以先利用有界性对比较复杂的触发环进行分析。如果 V 是一个节点的表达式。假设其子表达式 E 近似于有界区间$[a,b](a \in E \in b)$包含于表达式 V'。如果 V' 是不可满足的，那么 V 也是不可满足的。

到目前为止，我们都仅谈到了简单触发环。实际上，有很多触发图包含复杂触发环。对于这样的情况，我们的程序需要用到嵌套的循环语句。那些循环可以分别检测，然后进行组合。

结束语 我们提出了一种静态的分析 ECA 规则集可终止性的方法，把主动规则的执行翻译成一段循环程序。通过这种方法，规则集的可终止性变得比较清楚。我们就可以检查循环条件的可满足性及嵌套循环条件的可满足性来判定规则集行为的可终止性。对于复杂触发环我们可以首先进行单调性分析及有界性分析来协助可终止性分析。本文提出的算法可以获得更高的精度及更小的复杂度，相对于以前的节点对的比较，对于简单触发环，其复杂性由 $O(n^2)$ 降到了 $O(n)$。我们以后的工作将集中在两个方面：

1. 通过增量分析、施加应用限制来改进可满足性分析程序；

2. 对嵌套循环(复杂触发环)条件的可满足性作进一步的研究。

参 考 文 献

1 Baralis E, Bianco A. Performance Evaluation of Rule Semantics in Active Databases. In: 13th ICDE Conf. Proc. April 1997. 365～374

2 Baralis E, Ceri S, Paraboschi S. Compile-Time and Runtime Analysis of Active Behaviors. IEEE Transactions on Knowledge and Data Engineering, 1998, 10(3): 353～370

3 Bonifati A, Ceri S, Paraboschi S. Active Rules for XML : A New Paradigm for E-Services. In: 1st Workshop on E-Services (co-held with the 26th Very Large Data Bases Conference) Proc. Sep. 2000

4 Chakravarthy S, Le R, Dasari R. ECA Rule Processing in Distributed and Heterogeneous Environments. In: Intl. Symposium on Distributed Objects and Applications, 1999

5 Bonifati A, Ceri S, Paraboschi S. Pushing Reactive Services to XML Repositories using Active Rules. www 2001. 633～641

6 Bailey J, Poulovassilis A. Termination Analysis of Active Database Rules Techniques for Increased Precision: [Technical Report TR-98-06]. Department of Computer Science, King's College London, April 1998

(上接第92页)

Ⅰ 自反性：若 $Y \subseteq X$，则 $X \xrightarrow{R} Y$ 在 r 上成立。

Ⅱ 增广性：若 $X \xrightarrow{R} Y$ 在 r 上成立，且 $Z \subseteq U$，则 $XZ \xrightarrow{R} YZ$。

Ⅲ 传递性：若 $X \xrightarrow{R} Y$ 和 $Y \xrightarrow{R} Z$ 在 r 上成立，则 $X \xrightarrow{R} Z$ 在 r 上成立。

证明：(1)自反性：$\forall t_1, t_2 \in r$，因为 $Y \subseteq X \subseteq A$，所以 $\theta_x \subseteq \theta_Y$，即有 $\bigcap_{a \in X} \theta_a \subseteq \bigcap_{a \in Y} \theta_a$，根据定理3可知 $X \xrightarrow{R} Y$ 在 r 上成立。

(2)增广性：用反证法。假设$\exists\ t_1, t_2 \in r$ 违反 $XZ \xrightarrow{R} YZ$，即 $t_1[XZ] \overset{R}{=} t_2[XZ]$，但 $t_1[YZ] \overset{R}{\neq} t_2[YZ]$。

从 $t_1[XZ] \overset{R}{=} t_2[XZ]$ 可知，$t_1[X] \overset{R}{=} t_2[X]$，$t_1[Z] \overset{R}{=} t_2[Z]$。

从 $t_1[YZ] \overset{R}{\neq} t_2[YZ]$ 可知，$t_1[Y] \overset{R}{\neq} t_2[Y]$ 或 $t_1[Z] \overset{R}{\neq} t_2[Z]$。

如果 $t_1[Y] \neq t_2[Y]$，由于 $t_1[X] \overset{R}{=} t_2[X]$，则与已知的 $X \xrightarrow{R} Y$ 矛盾；如果 $t_1[Z] \overset{R}{\neq} t_2[Z]$，则与推导出的 $t_1[Z] \overset{R}{=} t_2[Z]$ 矛盾，因此假设不成立，所以Ⅱ是正确的。

(3)传递性：用反证法。假设$\exists\ t_1, t_2 \in r$ 违反 $X \xrightarrow{R} Z$，即 $t_1[X] \overset{R}{=} t_2[X]$，但 $t_1[Z] \overset{R}{\neq} t_2[Z]$。

如果 $t_1[Y] \overset{R}{\neq} t_2[Y]$，则与已知的 $X \xrightarrow{R} Y$ 矛盾；如果 $t_1[Y] \overset{R}{=} t_2[Y]$，则与已知的 $Y \xrightarrow{R} Z$ 矛盾，因此假设不成立，Ⅲ是正确的。

结论 本文中运用粗集理论的研究方法，改进了 RRDB 中的函数依赖定义，并给出了判断粗函数依赖是否成立的算法，使之更客观地反映 RRDB 数据的语义联系，体现现实世界不确定性信息的粗糙性和不完备性。本文用粗关系实例验证了粗函数依赖定义的优越性。在此基础上，探讨了粗函数依赖的推理规则，并得出了一些结论，这对 RRDB 理论的进一步研究奠定了基础。

当然，RRDB 数据依赖理论待研究的问题还有很多，比如：基于 Rough 关系数据库的数据约束和数据完整性、函数依赖与键的性质等，这些问题的研究是建立 RRDB 完善的理论体系的基础。

参 考 文 献

1 Pawlak Z. Rough sets[J]. International of Information and Computer Science, 1982, 11(5): 341～356

2 Pawlak Z. Rough sets-theoretical aspects of reasoning about data [M]. Dordrecht: Kluwer Academic Publishers, 1991. 68～162

3 Beaubouef T, Petry F, Buckles B. Extension of the relational database and its algebra with rough set techniques. Computational Intelligence, 1995, 11: 233～245

4 Beaubouef T, Petry F, Arora G. Information theoretic measures of uncertainty for rough sets and rough relational databases. Information sciance, 1998, 109: 185～195

5 Beaubouef T, Petry F. Fuzzy Set Quantification of Roughness in a Rough Relational Database Model. In: IEEE Intl. Conf. on Fuzzy Systems., 1994. 172～177

6 Nakata M, Murai T. Data Dependencies over Rough Relational Expressions. In: IEEE Intl. Fuzzy Systems Conf. 2001. 1543～1546

7 安秋生，徐久成，沈钧毅，等. Rough 关系数据库模型及其关系操作. 计算机科学，2002，29(7)：72～89

8 安秋生，徐久成，沈钧毅，等. 基于粗糙关系数据库的粗糙数据查询. 西安大学交通学报，2002，36(8)：859～862

9 施伯乐，丁宝康，周傲英等 编著. 数据库系统教程. 高等教育出版社，1999，12：120～148

10 Hailperin T. Probability Logic. Notre Dame Journal of Formal Logic, 1984, 25(3): 198～212