

基于有色网的 RFID 复杂事件 检测模型的研究与分析*

王 钊, 马志锋

(合肥工业大学 计算机与信息学院, 合肥 230009)

摘要: 针对 RFID 数据的特性以及目前几种复杂事件检测模型的不足, 在原有基于 Petri 网模型的基础上, 通过扩展和改进, 提出了一种基于有色网的 RFID 事件检测网 EDN, 对事件操作符进行了形式化和图形化的描述, 建立了复杂事件表达式的层次结构模型; 并分析了复杂事件表达式中的冲突算法, 给出了 RFID 事件的资源回收机制, 提高了 RFID 事件的处理能力。通过实验证明了该模型的正确性和有效性。

关键词: 射频识别技术; 复杂事件处理; 有色网; 事件检测; 事件共享机制

中图分类号: TP311 **文献标志码:** A **文章编号:** 1001-3695(2011)04-1425-04

doi:10.3969/j.issn.1001-3695.2011.04.063

Research and analysis of RFID complex event detection model based on CPN

WANG Zhao, MA Zhi-feng

(College of Computer & Information, Hefei University of Technology, Hefei 230009, China)

Abstract: Because of the characteristics of RFID data and shortcomings of current several complex event detection models, by expanding Petri nets, this paper proposed a RFID event detection network EDN. It gave the event operator a description of formal and graphical, established complex event expressions hierarchy model, analyzed the conflict algorithm in the event expression and event recycling mechanism. Experimental results demonstrate the correctness and validity of the model.

Key words: radio frequency identification(RFID); complex event processing; colored Petri nets; event detection; event sharing mechanism

无线射频识别^[1](RFID)实现了非接触式的数据自动获取和交换,是物联网应用的关键技术之一,受到业界的广泛关注。RFID 数据具有以下几个特性:a)动态性、关联性和时效性;b)丰富的隐含语义;c)批量性和海量性;d)冗余性和异构性。这种 RFID 数据的特点使发现有效信息变得非常困难,而复杂事件处理技术^[2]是 RFID 数据管理的重要方法之一。RFID 复杂事件处理的主要工作是事件清洗和事件检测两个方面。在事件检测方面,相关研究提出了不少方法,如基于有限自动机、匹配树、有向图和 Petri 网的检测方法。ODE^[3]是最早提出使用自动机模型进行复杂事件检测的系统,每当一个参与复杂事件发生的基本事件到达,自动机就会从一个状态跳跃到下一个状态,当到达终止状态后说明发生了复杂事件。但是,这种方法需要额外的数据结构来保存这些基本事件的相关信息,必须设计专门的自动机模型。文献[4]提出了一个基于查询计划的复杂事件检测技术,给出了 SASE 事件描述语言和六种操作符,并且对 SEQ(顺序操作符)扫描和构造操作符进行了优化,提高了事件检测的效率,而对事件模式匹配之间的共享没有研究,也缺乏对复杂事件层次性的建模验证。文献[5]提出了通过建立共享池和复杂事件分类表来提高检测效率的方法,却没有把事件检测和中间结果共享机制统一起来,加大了系统的开销。文献[6]提出了基于 Petri 网的事件检测方法,并且给出了网结构的定义,其缺陷是对于事件表达式的描述所产生的网结构过于复杂,没有层次化且不容易理解。文献[7]对 Petri 网进

行了扩展,提出了(timed Petri nets, TPN)检测方法,但是也没有考虑中间检测结果的共享。文献[8]探讨了基于有色网定义的 RFID 事件检测模型,给出了事件检测的构造规则,并对复杂层次的复合事件进行了分析和验证,同样没有给出中间结果的共享机制。

针对目前事件检测模型检测效率不高的问题,可以利用有色网描述复杂事件的优势,建立基于有色网的事件检测模型,通过对共享机制的分析,提出了 EDN 事件检测冲突算法。

1 RFID 事件的相关定义

RFID 事件是指在 RFID 应用系统中的一个具有语义的规则约束,一般可以分为原子事件和复杂事件。

定义 1 阅读器对电子标签的一次读取操作称为原子事件。

原子事件采用(EPC, reader, timestamp)三元组表示。其中:EPC 表示存储在电子标签中的代码;reader 表示阅读器;timestamp 表示阅读器读取 EPC 的时间点。

定义 2 复杂事件是由原子事件通过事件操作符组合而形成的具有某种语义的事件模式,记为 CE (complex event) [operator₁(component₁), operator₂(component₂), ..., operator_n(component_n)] | [E_{begin}, E_{end}]。其中:operator₁ 和 operator₂ 是事件操作符,component₁ 和 component₂ 表示原子事件或者复杂事件的语句。复杂事件通常发生在一个时间段内,用时间窗口

收稿日期: 2010-10-14; 修回日期: 2010-11-25 基金项目: 安徽省自然科学基金资助项目(104413264)

作者简介: 王钊(1958-),男,安徽桐城人,高级工程师,主要研究方向为软件工程;马志锋(1983-),男,山东潍坊人,硕士研究生,主要研究方向为复杂事件处理技术(mzfhfut@163.com)。

$[E_{begin}, E_{end}]$ 表示。

事件操作符表示了 RFID 事件之间的逻辑关系,具有时间约束和属性约束,表示符号约定如下:

- a) 否定操作符 negation。记为 $\neg A$, 表示事件 A 没有发生。
- b) 合取操作符 conjunction。记为 $A \& B$, 表示 A 和 B 都发生才发生的复合事件(不考虑发生的顺序)。
- c) 顺序操作符 sequence。记为 $A \rightarrow B$, 表示 A 发生完之后 B 才发生,即时间约束为 $a_{end} \leq b_{begin}$ 。对于表达式 $A \rightarrow A \rightarrow \dots \rightarrow A$, 记为 A^n , 其中, n 是 A 发生的次数, A 按照顺序依次发生。
- d) 析取操作符 disjunction。记为 $A \mid B$, 表示当事件中的任何一个发生时它就发生。
- e) Within。记为 $w(A, n)$, 表示 A 发生,且 $a_{end} - a_{begin} \leq n$ 。

属性约束可以用布尔表达式来表示,加在事件表达式的后面,这样的事件操作符被称为事件操作符变元。例如, $A \rightarrow B$ ($\text{getProperty}X(A) = B, \text{property}X$) 中的事件操作符就是 $A \rightarrow B$ 的一个变元操作符。其中, $\text{getProperty}X(A)$ 是一个全局函数,取得事件 A 的 X 属性值, X 属性同时也是事件 B 的一个属性。这个事件操作符的变元不仅表示为 A 发生之后 B 发生,而且还要满足后面布尔表达式的关系,只有布尔表达式成真才能检测到事件的发生。

上面定义的五组事件操作符分别对应的事件实例为 ($\neg a$), ($a \& b$), ($a \rightarrow b$), ($a \mid b$), $w(a, n)$ 。基本上能够满足描述各种所有的复杂事件,而且它们的优先级是依次递减。其中操作符变元所对应的事件实例为 $a \rightarrow b$ ($\text{getProperty}X(a) = b, \text{property}X$)。

在复杂事件检测过程中,带有 negation 事件操作符的事件实例有时是不可检测的,带有 negation 操作符的这种复合事件被称为被动事件。相对应地,不含 negation 事件操作符的复杂事件被称为主动事件。例如 $\text{sequence}(A, B, \text{negation}(C), D)$ 是一个被动事件, $\text{negation}(C)$ 是一个非事件。这个复杂事件表达式通过分析原子事件之间的关系来检测非事件 C 的发生。而在一些表达式中是不能检测出非事件的发生,需要人为产生一种事件来帮助检测非事件的发生,称之为伪事件。

原子事件是复杂事件处理引擎的基本处理单元,经过复杂事件引擎处理以后生成具有逻辑语义的复杂事件。下面举一个复杂事件的例子,超市监控商品被盗的一个业务逻辑,转换为复杂事件表达式为 $w(\text{shelf} \rightarrow \neg \text{counte} \rightarrow \text{exit}, 1)$, 其中 shelf 、 count 和 exit 都是 EPC 事件,分别表示货架事件类型、柜台事件类型和出口事件类型。这个事件表达式表示的业务逻辑为:如果在 1 h 内阅读器读到货架事件发生且柜台事件没有发生而出口事件发生了,说明物品不经过柜台被带出超市,应该发生报警。

2 RFID 事件检测模型

复杂事件可以看做是符合某种模式下的通过事件操作符连接多个事件的匹配组合,形式化的事件检测模型是验证事件检测正确性的一个重要基础,逐步探索一个能够识别系统中的原子事件有序队列表示的模型是重要目标之一。Petri 网具有图形化和可执行化的特点,适合表示复杂事件的检测,其层次结构在模拟复杂事件表达式上有着很大的优势,可以直观地反映复杂事件的层次结构。

2.1 RFID 事件检测网

定义 3 一个有色网^[9] (CPN) 是一个六元组 $\{P, T, \text{Pre},$

$\text{Post}, C, \text{cd}\}$ 。其中:

- a) P 是库所的有穷集合;
 - b) T 是变迁的有穷集合,且与 P 不相交;
 - c) C 是颜色类的集合;
 - d) $\text{cd}: P \cup T \rightarrow C$ 是颜色域的映射;
 - e) Pre, Post 是网的向前和向后关联矩阵。
- 定义 4 EDN(event detection network) = $\{P, P_s, P_e, T, N, A, U, C, CF, E, G, M, B\}$ 是一个 13 元组。其中:
- a) P 为库所的有穷集合,存放的是事件类型;
 - b) $P_s \in P$ 为输入库所的有穷集合,模拟事件流的简单事件;
 - c) $P_e \in P$ 为输出库所的有穷集合,模拟输出的复杂事件;
 - d) T 为变迁的有穷集合,存放的是事件操作符;
 - e) N 描述了库所的颜色;
 - f) $AC(P \times T \cup T \times P)$ 为连接库所和变迁的弧集合;
 - g) $U: p \rightarrow N^+$, 描述库所中托肯的失效时间(N^+ 是一个正整数);

h) C 表示颜色集合,描述事件类型,可以是复合事件类型也可以是 EPC 事件类型;

i) $CF: P \rightarrow C$ 托肯函数,将每个库所映射到一个属于 C 的托肯类型;

j) E 为弧函数,将有限弧映射到一个表达式(称为弧表达式),描述弧上消耗和产生的托肯和托肯颜色;

k) G 表示哨函数,描述了事件的条件约束如时间约束和属性约束等,是与输入库所颜色相关的布尔表达式;

l) M 是事件检测网的标志, M 为 $P \rightarrow \text{Token}$ 的函数,其中托肯表示事件类型的实例,它是具有某一颜色即某一事件类型的具有属性的对象;

m) B 表示一个变迁表达式,描述计算托肯的属性值。

对上述元组中的元素作以下几点补充说明:

a) 除了存放事件的库所颜色模拟事件类型变迁模拟事件操作符,其中变迁 t 描述了事件之间的约束关系,包括时间约束和属性约束,这些约束存放的是事件类型哨函数描述。

b) 对于每个变迁 t , 输入库所定义为形如集合 $\{p \mid \langle p, t \rangle \in A\}$, 输出库所形如 $\{p \mid \langle t, p \rangle \in A\}$, 输入弧形如 $\{a \mid a \in P \times T \text{ 且 } p: a = \langle p, t \rangle\}$, 输出弧形如: $\{a \mid a \in T \times P \text{ 且 } \exists p: a = \langle t, p \rangle\}$ 。

c) 每个库所可以包括一个或多个托肯,EDN 的实际状态就是由分配给它的托肯数量决定的,每当一个库所被标记为一个托肯,符合规定的约束条件,托肯就可以从一个库所移动到另一个库所。

2.2 EDN 检测网变迁的授权

设 RFID 事件检测网 $\text{EDN} = \{P, P_s, P_e, T, N, A, U, C, CF, E, G, M, B\}$, M_0 表示检测网的初始状态($M_0 = \emptyset$), M_n 表示第 n 个时间关注点的状态。设 M_n 是目前检测网的稳定标志,此时没有任何变迁,称之为实时稳态。 M_{n+1} 是第 $n+1$ 个时间关注点的状态。如果满足触发规则,则发生 $M_n \rightarrow M_{n+1}$ 。 $M_n \rightarrow M_{n+1}$ 表示实时稳态从第 n 个关注点进入到第 $n+1$ 个关注点。

EDN 变迁触发规则:在 $n+1$ 的时间点,某个 RFID 原子事件进入到 EDN 检测网中,输入库所 P_s 中托肯被标志,此时网系统是不稳定的;当变迁的局部条件满足且 G 哨函数成真,则变迁触发;触发以后,按照 B 给出的操作生成的托肯存放在相

应的库所中。

2.3 事件操作符的图形化表示

上述定义的五种事件操作符是组成事件表达式的关键元素,利用 EDN 检测网对这五种事件操作符进行建模,也是整个检测网的基本逻辑结构网。如图 1 所示 negation 的结构,其中库所用圆圈表示,变迁用矩形表示,有限弧用箭头表示。其事件检测网可以描述为 $EDN = \{P', P_s', P_E', T', N', A', U', C', CF', E', G', B', M'\}$ 。其中: $P' = \{p_1, p_2\}, P_s' = \{p_1\}, P_E' = \{p_2\}, T' = \{t\}, A' = \{\langle p_1, t \rangle, \langle t, p_2 \rangle\}, U' = \{(p_1, 0), (p_2, 0)\}, C' = \{A, E\}, CF' = \{(p_1, A), (p_2, E)\}, E' = \{(\langle p_1, t \rangle, (1, A)), (\langle t, p_2 \rangle, (1, E))\}, G' = \emptyset, B' = \{E. begin = A. begin, E. end = A. end\}, M_0 = \emptyset$ 。

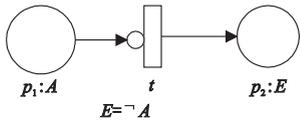


图 1 $E = \neg A$ 图形化表示

图 2 为 conjunction、sequence 的结构,从图可以看出两个图的构造基本相同,sequence 结构上增加了一个哨函数 $G': A. end \leq B. begin$ 来表示时间约束。Conjunction 的事件检测网可以描述为 $EDN = \{P', P_s', P_E', T', N', A', U', C', CF', E', G', B', M'\}$ 。其中: $P' = \{p_1, p_2, p_3\}, P_s' = \{p_1, p_2\}, P_E' = \{p_3\}, T' = \{t\}, A' = \{\langle p_1, t \rangle, \langle p_2, t \rangle, \langle t, p_3 \rangle\}, U' = \{(p_1, 0), (p_2, 0), (p_3, 0)\}, C' = \{A, B, E\}, CF' = \{(p_1, A), (p_2, B), (p_3, E)\}, E' = \{(\langle p_1, t \rangle, (1, A)), (\langle p_2, t \rangle, (1, B)), (\langle t, p_3 \rangle, (1, E))\}, G' = \emptyset, B' = \{E. begin = \min(A. begin, B. begin), E. end = \max(A. end, B. end)\}, M_0 = \emptyset$ 。Sequence 的检测网在 conjunction 的结构上加了哨函数。因此,除了 G' 不一样,其他元组都一样,即 $G' = A. end \leq B. begin$ 。如果有属性约束可以与哨函数进一步组合,如属性限制为 $(A. y = B. y)$,则整个约束描述为 $G' = A. end \leq B. begin \& (A. y = B. y)$ 。Sequence 有一种特殊结构即为 $E = A^n$,如图 2(c) 所示,其事件检测网的描述与 sequence 类似。

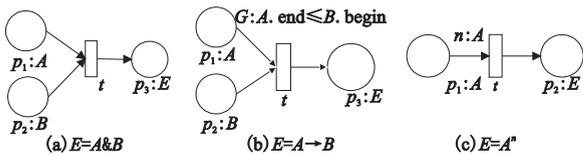


图 2 $E = A \& B$ 和 $E = A \rightarrow B$ 的图形化表示

图 3 为 within 操作符的图形化表示,其表示式为 $w(A, n)$ 表示在 n 时间内 A 有效,超过规定的时间 n 之后事件 A 就会失效,因此 $U(p) = n$ 。图 3(b) 给出了 $w(A | B, n)$ 的网结构。其检测网的描述与上面操作符的监测网描述类似。以图 3 为例,不同的就是 $U' = \{(p_1, n), (p_2, n), (p_3, 0)\}$ 。

2.4 RFID 事件表达式的网状结构

复杂事件可以看做是符合某个模式的多个原子事件加上事件操作符的任意组合。因此,多层次 RFID 复杂事件检测网可以分解为上述定义的事件操作符的层次组合,以及操作符上各种约束的组合。五种事件操作符分别代表了事件表达式中的五种基本网结构。建立一个复杂事件表达式的层次网结构的过程,其实是一个已构造出来的网结构的输出库所和输入库所的融合过程。例如上述一个监控超市物品被盗的例子,其中 shelf、count 和 exit 分别用 A, B, C 表示,涉及到的操作符分别为 sequence、negation 和 within,事件表达式为 $\text{within}(\text{sequence}(A,$

$\text{negation}(B), C), 1)$,如图 4 所示。

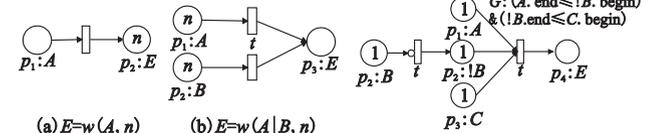


图 3 Within 和 $\text{Within}(A|B, n)$ 的图形化表示

图 4 事件表达式的层次结构

2.5 事件表达式的冲突算法

复杂事件表达式中的多层次结构是由基本逻辑结构网逐层建立的,但是复杂事件表达式存在着一些相同的子事件,所以在建立事件监测网的过程中可以共享子事件的检测,这样可以降低事件检测的冗余性,提高事件检测的效率。但是这可能会导致在事件检测的过程中产生资源冲突。冲突是 Petri 网很常见的一类现象,反映了过程对共享资源的竞争。在图 5 所示的 RFID 事件检测中,两个事件表达式 $E_1 = \text{sequence}((A | B), C^n)$ 和 $E_2 = \text{sequence}((A | B), D^n)$ 中 $(A | B)$ 可能会引起资源冲突。

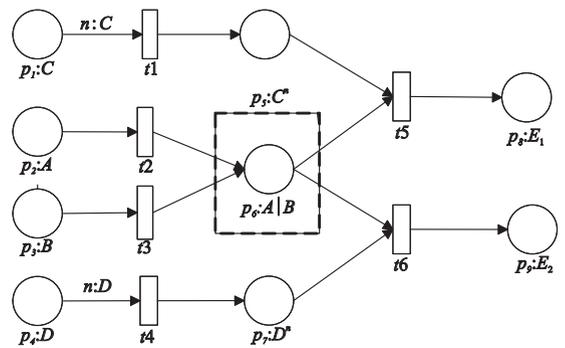


图 5 复杂事件检测的冲突

一般地,对于冲突的判断需要几个条件:a) Petri 网的层次结构模型中是否存在冲突,这是构成冲突的必要条件;b) 库所中的托肯数量;c) 冲突结构中各个相关变迁的激活情况。在图 5 中, p_6 是一个共享库所,同时也是决策库所。当 t_5 和 t_6 的激活条件满足且决策库所 p_6 的托肯数量小于等于 1 时,产生冲突。在事件检测网 EDN 中,处于某一种状态的监测网可以用系统对应的 Petri 网 $\Sigma_M = (N_M, M)$ 进行描述, M 表示 Σ_M 中的状态。

定理 若检测网 $\Sigma_M = (N_M, M)$ 中存在两个不同变迁 t_1 和 t_2 ,则在某一时刻陷入冲突条件为 $({}^*t_1 \cup t_1^*) \cap ({}^*t_2 \cup t_2^*) = \emptyset$ 。其中: *t 是变迁 t 的输入库所, t^* 是变迁 t 的输出库所。

系统首先记录该共享决策库所中的标志和输出变迁的个数,并检测共享库所中所含的托肯数量,然后根据以下条件解决冲突:

- a) 如果决策库所中的托肯数量为 0,则等待。
- b) 当决策库所中有托肯出现时,分为两种情况讨论,即托肯事件数量为 1 和托肯事件数量大于 1 的情况。
- c) 如果决策库所中的托肯数量为 1 且满足约束条件,则发生资源冲突。根据输出事件的优先级调用变迁上的权函数,从而判断输出变迁的激活顺序。如图 5 所示,如果 E_1 是一个实时事件, E_2 是一个普通的查询事件,那么输出事件 E_1 比 E_2 的优先级高, t_5 先于 t_6 被激活。
- d) 如果决策库所中的托肯数量为 1 且满足约束条件,但是输出事件的优先级别一致,则根据文献[8]中提出的 Chronicle 语义,选取构成复杂事件的最旧事件的初始事件作为该复杂事件的初始事件。如图 5 所示, E_1 和 E_2 的优先级别一样,如果 p_6 含有一个托肯, p_5 含有托肯而 p_7 不含托肯,变迁 t_1 先

于 t_4 激活,从而激活变迁 t_5 ,反之激活 t_6 。

e)如果输出事件的优先级一致且决策库所中的托肯数量为 1, p_5 和 p_7 中含有托肯,在这种情况下会出现争夺 p_6 中的托肯导致死锁。为了预防死锁的现象,在事件检测过程中预先记录此检测网的状态 M ,如果满足这种状态,对决策库所中的托肯事件进行缓存预处理来同时以满足变迁 t_5 和 t_6 的激活。

f)当决策库所中的托肯数量大于 2 且满足约束条件时,系统不会发生冲突。图 5 中, t_5 和 t_6 同时被激活。

g)当某个代表事件的托肯通过变迁到终端节点时,代表系统感兴趣的事件被检测到,此时触发相应的动作执行。在图 5 中, E_1 和 E_2 分别代表终端节点即输出事件。

3 检测网 EDN 的空间回收机制

在基于 Petri 网的检测过程中,中间库所所含的托肯一直存储在内存中,用来与新产生的托肯进行组合匹配。但是随着检测的进行,大量的中间结果被存储起来可能导致系统的不稳定。Within 操作符就表示事件的组合符合一定的时间约束,只有在某一时间范围内才有可能构成复杂事件。因此,在事件的检测过程中不必保存那些过期的事件,可以用空间回收机制将事件从存储区中删除,这样既可以释放空间又可以降低中间结果,从而提高检测的效率。空间回收机制为每个检测网 EDN 的存储区库所中的托肯计算有效的的时间约束,并随时删除过期的事件。在回收空间机制为存储区计算时间约束条件时,会出现图 5 中的情况,即事件检测网的共享,从而导致决策库所与不同库所计算的时间不一致。为此,当决策库所出现时,空间回收机制需要保存根据其他库所的托肯与决策库所的托肯激活的条件,然后计算各自满足的时间约束条件。

4 实验结果与分析

使用 Java 语言实现了基于 RFID 事件流的检测网,支持五个事件操作符的检测,并且兼容 ALE^[10] 标准。从内存消耗和事件处理时间两个方面来测试提出的 EDN 网检测算法。测试环境如下:Windows XP 操作系统, Petium IV 2.8 GHz, 2 GB RAM。利用开源的 RFID 仿真平台 Rifi^[11] 搭建的一个仿真环境,产生实验所需要的仿真数据。定义了十个基本事件类型分别为 A、B、C、D、E、F、G、H、I、J,这些事件类型通过上文定义的事件操作符任意组合,其中至少有一个具有一定语义的复杂事件输出,并且组成复杂事件的表达式中的操作符必须有两种或者两种以上,如 $E_1 = \text{sequence}((A|B), C^n)$ 。将 EDN 事件检测方法 with Esper3.5.0^[12] 进行实验结果分析比较,Esper 是一个事件流处理和复杂事件处理的系统,目前受到了广泛的关注。

图 6 表示 EDN 事件检测方法和 Esper 事件检测中内存使用的情况。从图 6 中可以看出,在事件数量比较少的时候,两种方法使用内存的效率差距不大。当事件数量达到 6 000 时,EDN 事件检测技术明显比 Esper 事件检测技术消耗内存少。这是因为基于 Petri 网的事件检测内存回收机制会把过期的事件从内存中删除。事件的处理时间如图 7 所示。从图 7 可以看出,EDN 比 Esper 的事件处理花费的时间少。这是由于 EDN 算法的中间结果共享机制可以避免中间结果的重复检测,降低复杂事件表达式的冗余,从而减少了事件的处理时间。

同时又因为内存回收机制把无用的事件回收,提高了事件检测的效率,从而也减少了事件的处理时间。

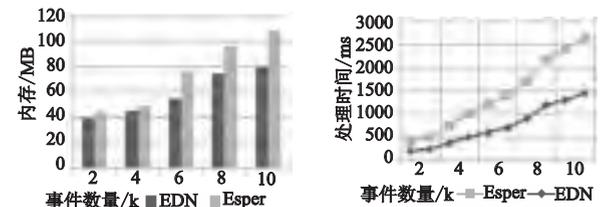


图6 两种算法的内存使用比较

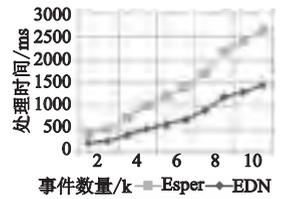


图7 两种算法的事件处理时间比较

5 结束语

为了提高 RFID 事件检测效率,首先分析了 RFID 事件检测中存在的不足,提出了一种基于有色网的事件检测模型 EDN,对事件操作符给出了形式化和图形化的表示。在此基础上建立了复杂事件表示式的层次模型,并讨论了复杂事件表达式中子事件的冲突算法。在 RFID 复杂事件检测中,RFID 事件之间也存在时间约束。未来的工作包括将事件的时间因素加入到检测网中,进一步研究复杂事件构成的时间约束、空间回收和时间约束之间的存在关系和对非事件检测的支持,从而提高事件的检测效率。

参考文献:

- [1] 董丽华. RFID 技术与应用[M]. 北京:电子工业出版社,2008.
- [2] WANG Fu-sheng, LIU Shao-rong, LIU Pei-ya. Complex RFID event processing[J]. The VLDB Journal-The International Journal on Very Large Data Bases,2009,18(4):913-931.
- [3] GEHANI N, JAGADISH H. Ode as an active database: constraints and triggers[C]//Proc of VLDB. San Francisco, CA: Morgan Kaufmann Publishers Inc,1991:327-336.
- [4] WU E, DIAO Yan-lei, RIZVI S. High-performance complex event processing over streams[C]//Proc of the 2006 ACM SIGMOD International Conference on Management of Data. New York: ACM,2006: 407-418.
- [5] ZHAN Chuan-zhen, FAN Yu-shun, LIU Ren-jing. Architecture, implementation and application of complex event processing in enterprise information systems based on RFID [J]. Enterprise Information System,2007,1(1):3-23.
- [6] GATZIU S, DIRTRICH K R. Detecting composite events in active databases using Petri nets[C]//Proc of the 4th International Workshop on Research Issues in Data Engineering: Active Data System. 1994:2-4.
- [7] JIN Xing-yi, LEE Xiao-dong, KONG Ning, et al. Efficient complex event processing RFID data stream[C]//Proc of the 7th IEEE/ACIS International Conference on Computer and Information Science. Washington DC:IEEE Computer Society,2008:75-81.
- [8] 叶蔚,黄雨,赵文,等. 基于 Petri 网的 RFID 中间件复合事件检测研究[J]. 电子学报,2008,36(B12):1-8.
- [9] GIRAULT C, VALK R. 系统工程 Petri 网—建模、验证与应用指南[M]. 王生原,余鹏,霍金键,译. 北京:电子工业出版社,2005.
- [10] EPCglobal. The application level events (ALE) specification version 1.1.1 [S/OL]. (2009). http://www.epcglobalinc.org/standards/ale/ale_1_1_1-standard-core-20090313.pdf.
- [11] FLOERKEMEIER C, FLEISCH E. RFID applications: interfacing with readers[J]. IEEE Software,2008,25(3):67-70.
- [12] Esper reference documentation [R/OL]. (2010). http://esper.codehaus.org/esper-3.5.0/doc/reference/en/pdf/esper_reference.pdf.