一种基于 RFID 数据集的物品工作流挖掘方法

顿海强[1,3],赵 文[2,3],邓鹏鹏[1,3],张世琨[2,3],王立福[2,3],谭 杰[4]
(1. 北京大学信息科学技术学院,北京 100871; 2. 北京大学软件工程国家工程研究中心,北京 100871;
3. 北京大学高可信软件技术教育部重点实验室,北京 100871; 4. 中国科学院自动化研究所 RFID 研究中心,北京 100190)

摘 要: 不同种类的物品在供应链中的移动形成不同的物品工作流,通过对这些物品工作流的挖掘,能够发现不同种类物品的流向和主要流转路径等信息,进而基于这些信息对供应链过程进行管理和优化. 本文提出了一种基于 RFID 数据集的物品工作流挖掘方法,其中定义了一种基于 Petri 网的物品工作流网,讨论了物品工作流网所支持的几种物品工作流模式,给出了基于 RFID 数据集的数据过滤和聚合算法,以及物品工作流网的挖掘算法,最后进行了必要的实验.

关键词: 物品工作流; 物品工作流挖掘; RFID 数据集; Petri 网
中图分类号: TP311 **文献标识码**: A **文章编号**: 0372-2112 (2008) 12A-086-08

A Commodity Workflow Mining Approach Based on RFID Data Sets

DUN Hai-qiang[1,3], ZHAO Wen[2,3], DENG Peng-peng[1,3], ZHANG Shi-kun[2,3], WANG Li-fu[2,3], TAN Jie[4]
(1. *School of Electronics Engineering and Computer Science, Peking University, Beijing* 100871, *China*;
2. *National Engineering Research Center for Software Engineering, Peking University, Beijing* 100871, *China*;
3. *Key Laboratory of High Confidence Software Technologies of Ministry of Education, Peking University, Beijing* 100871, *China*;
4. *RFID Centre, Institute of Automation, Chinese Academy of Sciences, Beijing* 100190, *China*)

Abstract: The movement of different kinds of commodities through the supply chain forms different commodity workflows. The flow trends and main paths of different commodities can be discovered by commodity workflow mining, and thus facilitate the management and optimizing of supply chain processes. On the basis of RFID data sets, an approach to mine commodity workflow is proposed. A Petri net based commodity workflow net is defined, some workflow patterns supported by the commodity workflow net are discussed, a cleaning algorithm and an aggregating algorithm based on RFID data sets are provided, and the mining algorithms of the commodity workflow net are presented. Finally, some essential experiments are conducted.

Key words: commodity workflow; commodity workflow mining; RFID data sets; petri net

1 引言

无线射频识别(Radio Frequency Identification, RFID)技术是 20 世纪 90 年代开始兴起的一种自动识别技术. 与其他自动识别技术相比,RFID 技术具有可非接触识别、可识别高速运动物品、抗恶劣环境、可同时识别多个物品等突出特点. 近年来,随着相关技术的不断成熟,RFID 技术引起全球学术界和产业界的普遍关注,在多个领域尤其是物流和供应链领域得到了广泛应用,成为目前发展比较迅速的自动识别技术之一.

在实际的供应链管理应用中,RFID 标签的读取会产生大量的数据,这些数据包含有物品本身相关信息以及物品在不同时刻所处的不同位置等信息. 对生产商、物流商、销售商以及一般客户来说,这些数据蕴含许多有价值的信息,例如某类物品在某段时间的销售情况和主要销售路径,以及同类物品在不同季节的销售变化趋势等信息. 基于这些数据能够挖掘出物品在供应链中的过程信息以及不同路径的执行概率信息,进而为供应链的管理和优化提供指导.

过程挖掘的基本思想在 1995 年由美国新墨西哥州立大学的 Cook 和 Wolf 提出,目标是从软件过程的事件日志中自动发现过程模型[1]. 1998 年,IBM Almaden 研究中心的 Agrawal 等人首先从工作流的角度提出工作流过程挖掘[2],他们的工作是从给定的工作流日志中发现一个工作流过程模型的图形表示,并且这个模型能够重现日志中记录的事件. 基于工作流网(Workflow Net,简称 WF-Net[3][4]),Aalst 提出了用于工作流挖掘的 alpha 算法[5],该算法在日志完备的情况下,可以发现良构的工

收稿日期:2008-07-16;修回日期:2008-11-16
基金项目:国家 863 高技术研究发展计划项目(No. 2006AA04A119,2006AA04A121);国家自然科学基金(No. 60803014)

作流网.但是上述几项工作都是基于工作流日志展开的,并不完全适用于基于 RFID 数据集的物品工作流挖掘.

Gonzalez 和 Han 提出了一种基于 RFID 数据集的物品工作流挖掘方法,定义了一种基于有限状态自动机的 RFID 工作流模型[6],给出了各个路径执行概率的计算方法.该工作流模型根据物品在位置的停留时间将每个位置散列到多个状态,能够描述各个状态下的物品终止个数和流转到其他状态的物品个数,进而基于这些信息计算不同路径的执行概率信息.但是,他们所提出的工作流挖掘算法没有考虑具体时间信息,不能够发现特定时间段内的物品流向信息,而且没有明确区分物品的类别,不提供特定种类物品的工作流模型的挖掘.

在实际的供应链管理应用中,物品的一般流转路径是从生产商发货,经过物流环节运抵销售场所,最终到达普通用户手中.基于以上假定,为了表示物品的流转路径、各个路径的执行概率以及物品在各个位置的停留时间等信息,参考 WF-Net,我们提出了一种基于 Petri 网的物品工作流网(Commodity Workflow Net,简称 CWN).CWN 对 WF-Net 进行了改进,具有一个开始节点和多个终止节点,任何其他中间节点都不能终止,此外,它扩展了变迁的执行概率和库所的停留时间等信息.本文主要对如何基于 RFID 数据集挖掘物品工作流网 CWN 进行了研究.

2　RFID 编码和 RFID 数据集

RFID 编码是 RFID 技术的核心,RFID 编码规则是产生 RFID 数据集的基础,是物品检索和物品工作流挖掘的依据,一个合理的编码规则能够有效地支持物品检索、物品工作流挖掘及 RFID 的其它推广应用.

2.1　RFID 编码

目前,在 RFID 编码及解析方面,美国的 EPCglobal 相关规范[7]和日本泛在中心的相关规范[8]均涉及这方面的内容,学术界和产业界也基本上都围绕着这两大类规范进行研发.不久前,我国推出了自己的 RFID 编码草案.在分析 EPCglobal 的标签数据规范和国内电子标签标准工作组提出的数据格式的基础上,结合我国国情,为了有效支持目前我国在行政管理、企业管理、产品质量检测与监督等方面实行的分地域、分级管理方式,863 课题"RFID 公共服务体系架构设计及应用服务关键技术研究与开发"项目组提出了一种可变长度的、并且兼容国际上现有标签数据规范的 RFID 编码规范 CID-V,如表 1 所示.

表 1　可变长 RFID 编码 CID-V 结构

码段	头部	有效位	分段标识位	国家、地区标识	城市区号	组织机构代码	产品分类码	序列号
长度(bits)	8	4	3	4 - 16	8 - 16	48	4 - 27	61 - 38

下面对 CID-V 的各个组成部分进行简单说明:

●头部:标识不同编码类型,对一种编码类型该段取值是固定的;

●有效位:共 4bits,前 2bits 表示国家地区标识码的长度,后 2bits 表示城市区号的长度;

●分段标识位:共 3bits,用来标识产品类别码和序列号的分割位置,而产品类别码和序列号总长度是固定的;

●国家、地区标识:采用全球区号码,长度 4-16bits,从头至尾每 4bits 表示一个十进制数字;

●城市区号:采用电信网电话区号;长度 8-16bits,从头至尾每 4bits 表示一个十进制数字;

●组织机构代码:标识一个组织机构(例如生产厂商),按照国家组织机构代码的编码规则,由 8 位数字或大写字母和一位校验码表示,这里去掉校验码,只用前 8 位,每一位由 0～35 表示(A～Z 由 10～35 表示),所以每位需要 6bits 来表示,共需 48bits;

●产品类别码:在同一个组织机构内部,唯一标识一个产品类别,其长度可变;

●序列号:在同一个产品类别下,唯一标识一个单品,其长度可变.

该编码方案的优点在于:(1)适应我国企业管理制度地域性强的特点,能有效满足按级别、按地域管理的要求;(2)兼容其它国内外编码方案,支持与其它编码的转换与相互查询.

2.2　RFID 数据集

贴有 RFID 标签的物品进入 RFID 读写器的覆盖区域后,RFID 读写器就会读取标签数据,生成 RFID 数据集.

RFID 应用所生成的原始数据集 RawData 可以表示成三元组(CID,Location,Time)的形式,其中 CID 是物品的唯一标识,采用的是上一节所介绍的 CID-V 编码结构,Location 是 RFID 读写器读取到物品的地点,Time 是 RFID 读写器读取到物品的时间.

为了形象地阐述物品工作流的具体构造过程,我们假定这样一个场景,一批货共 1000 个物品从生产地青岛分别发往北京、上海和广州,其中北京 600 个物品,上海和广州分别 200 个物品,发往北京的物品经分流中心、仓库和超市后最终到达用户手中,发往上海和广州的物品经分流中心和超市后到达用户手中.表 2 是该例所生成的原始数据集,其中 i 表示物品 ID,l 表示地点,r 表示时间,下标用来标识不同的物品、地点和时间.

在实际应用中,同一个物品在相同的地方可能被读取多次,因此原始数据集中就会包含多条该物品在

同一个地方的记录,从而造成原始数据集的规模过于庞大.对这类数据进行清洗可以消除冗余数据,有效减小数据集的规模.我们将清洗后的数据集记为 CleanData,其数据记录表示成四元组(CID,Location,Time_In,Time_Out)的形式,其中 CID 和 Location 的含义同前,Time_In 表示物品进入某个地点的时间,Time_Out 表示物品离开该地点的时间.表 3 是表 2 中的数据经过清洗后的结果,清洗后数据集的规模显著减少.

表 2 原始 RFID 数据集

RFID RawData(CID,Location,Time)
$(i_1,l_1,r_1)(i_2,l_1,r_1)\cdots(i_{1000},l_1,r_1)(i_1,l_1,r_2)(i_2,l_1,r_2)\cdots(i_{1000},l_1,r_2)\cdots(i_1,l_1,r_{10})(i_2,l_1,r_{10})\cdots(i_{1000},l_1,r_{10})(i_1,l_2,r_{15})(i_2,l_2,r_{15})\cdots(i_1,l_2,r_{20})(i_2,l_2,r_{20})\cdots(i_{1000},l_{15},r_{100})$

表 3 清洗后的 RFID 数据集

CID	CleanData(CID,Location,Time_In,Time_Out)
i_1	$(i_1,l_1,r_1,r_{10})(i_1,l_2,r_{15},r_{20})(i_1,l_5,r_{30},r_{40})$
i_2	$(i_2,l_1,r_1,r_{10})(i_2,l_2,r_{15},r_{20})(i_2,l_5,r_{30},r_{40})$
…	
i_{101}	$(i_{101},l_1,r_1,r_{10})(i_{101},l_2,r_{15},r_{20})(i_{101},l_6,r_{35},r_{50})$
…	
i_{1000}	$(i_{1000},l_1,r_1,r_{10})(i_{1000},l_3,r_{20},r_{30})(i_{1000},l_8,r_{35},r_{50})(i_{1000},l_{15},r_{90},r_{100})$

此外,在实际的供应链管理应用中,物品一般都是成批次运输,而且越在供应链的上游物品的批量越大,越往供应链的下游物品的批量越小.经过清洗后的 RFID 数据集中存在这样一些记录,这些记录的地点和时间相同,也就是说多个物品在相同的时间进入某个地点并且在相同的时间离开该地点,对这些记录进行聚合可以进一步减少数据集的规模.我们将聚合后的数据集表示为聚合表 AggregateData 和映射表 MAP 的形式.AggregateData 中的每条记录是四元组(SID,Location,Time - in,Time - out)的形式,其中 SID 表示的是其他的 SID 或者某些 CID 的集合,具体指向通过 MAP 表来表示,MAP 表是 SID 和 SID 之间、或者 SID 和 CID 之间的映射关系.以上面的表 3 为例,对相关数据进行聚合后数据集的规模进一步减少,聚合结果如表 4 和表 5 所示.

表 4 物品聚合数据表

SID	AggregateData(SID,Location,Time_In,Time_Out)
sid_1	(sid_1,l_1,r_1,r_{10})
sid_2	$(sid_2,l_2,r_{15},r_{20})$
sid_3	$(sid_3,l_3,r_{15},r_{20})$
sid_4	$(sid_4,l_4,r_{20},r_{30})$
sid_5	$(sid_5,l_7,r_{30},r_{40})$
sid_6	$(sid_6,l_8,r_{35},r_{50})$
sid_7	$(sid_7,l_5,r_{30},r_{40})$
sid_8	$(sid_8,l_5,r_{30},r_{50})$
…	…
sid_{17}	$(sid_{17},l_6,r_{35},r_{50})$
…	…
sid_{106}	$(sid_{106},l_{15},r_{90},r_{100})$

表 5 映射表

SID	SIDS/CIDS
sid_1	sid_2,sid_3,sid_4
sid_2	$sid_7,sid_8,\cdots,sid_{26}$
sid_3	sid_5,sid_6
sid_4	$sid_{27},sid_{28},\cdots,sid_{46}$
sid_5	$sid_{47},sid_{48},\cdots,sid_{66}$
sid_6	$sid_{67},sid_{68},\cdots,sid_{106}$
sid_7	$i_1,i_2,i_3,\cdots,i_{10}$
sid_8	$i_{11},i_{12},i_{13},\cdots,i_{20}$
…	…
sid_{17}	$i_{101},i_{102},i_{103},\cdots,i_{110}$
…	…
sid_{106}	$i_{991},i_{992},i_{993},\cdots,i_{1000}$

基于物品聚合数据表 AggregateData 和映射表 MAP 可以挖掘出单个物品或者一类物品的流转路径、物品在路径上各个位置所停留的时间以及每条路径的执行概率.本文将其建模为具有时间和概率的物品工作流的形式,从而为供应链过程的管理和优化提供相关信息.

3 物品工作流

作为一种广泛使用的系统建模工具,Petri 网[9~11]具有许多优良的性质,例如直观的图形表示、严格的数学定义以及丰富的分析方法和技术等.此外,目前许多计算机辅助 Petri 网设计分析工具也已经被开发出来并投入使用.综上,本文选用 Petri 网作为物品工作流的建模方法.

3.1 物品工作流网

工作流模型是工作流系统的核心,是对工作流的相关性质进行分析的基础.为了对工作流的进行建模,Aalst 提出了一种特殊的 Petri 网——工作流网(Workflow Net,简称 WF-Net)[3,4].在 WF-Net 中,一个库所对应过程中的一个条件,一个变迁对应过程中的一个可执行活动.

定义 1 一个有向网[9] $PN=(P,T;F)$ 称为一个工作流网(WF-Net)的充分必要条件是:

(1) 存在一个源库所 $i\in P$,使得 ${}^{\cdot}i=\varnothing$;

(2) 存在一个汇结库所 $o\in P$,使得 $o^{\cdot}=\varnothing$;

(3) 每个节点 $x\in P\cup T$ 都位于从 i 到 o 的一条路径上.

其中 ${}^{\cdot}x=\{y|(y,x)\in F\}$ 称为 x 的前集或输入集,$x^{\cdot}=\{y|(x,y)\in F\}$ 称为 x 的后集或输出集.

在实际的供应链过程中,每条路径的执行概率以及物品在各个位置的停留时间等信息至关重要,可以

用来对供应链过程进行管理和优化. 根据供应链过程中物品流动的特点以及实际的业务需求,本文对 WF-Net 进行了扩展,提出了一种包含变迁执行概率和物品在库所停留时间的物品工作流网 CWN.

定义 2　一个七元组 $CWN = (P, T; F, \rho, \theta, s, E)$ 称为一个物品工作流网的充分必要条件是:

(1) $(P, T; F)$ 是一个有向网;

(2) $\rho: T \rightarrow R$ 是一个从变迁 T 到实数 R 的映射函数,该函数值表示变迁 T 发生的概率;

(3) $\theta: P \rightarrow Duration$ 是一个从库所 P 到时间 $Duration$ 的映射函数,该函数值表示一个物品或者一类物品在一个库所的停留时间;

(4) s 是唯一的源库所;

(5) E 是汇结库所的集合.

3.2　物品工作流模式

模式是从不断重复出现的事件中发现和抽象出的规律,是解决问题的经验总结. Alexander[12] 给出的经典定义是:每个模式都描述了一个在我们的环境中不断出现的问题,然后描述了该问题的解决方案的核心. 通过这种方式,你可以无数次地使用那些已有的解决方案,无需再重复相同的工作. 工作流模式[13]可以用来描述过程逻辑,而物品工作流模式可以描述物品流转过程规约,对不同的物品工作流模式来说,变迁之间的发生满足不同的约束关系.

在供应链管理应用中,常见的业务模式主要有打包、拆包、再包装等,本节以顺序模式、物品打包模式、物品拆包模式为例分析变迁之间所满足的约束. 主要从以下几个方面进行分析:(1) 变迁之间的偏序关系;(2) 变迁发生的概率;(3) 变迁之间的条件概率关系. 为此定义如下几个关系符和运算符:

●≺关系:≺表示变迁之间的偏序关系,$t_1 \prec t_2$ 表示变迁 t_1 是变迁 t_2 的前驱节点,t_2 是 t_1 后继节点;

●变迁发生的概率 pr:一个变迁 t_i 具有发生权时得以执行的概率通过 pr_i 表示;

●条件概率 P: $P(t_2 | t_1)$ 表示在变迁 t_1 发生的前提下变迁 t_2 发生的概率.

物品工作流网所支持的顺序模式如图 1 所示,其中变迁 t_2 能否执行依赖于变迁 t_1 能否执行,它们之间满足以下条件:$t_1 \prec t_2 \wedge pr_2 = 1 \wedge P(t_2 | t_1) = 1$.

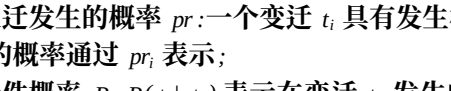


图1　物品工作流顺序模式

物品工作流网所支持的拆包模式如图 2 所示,各个变迁之间满足以下条件:

$$\sum_{i=1}^{n} pr_i = 1 \wedge (\forall 1 \leq i \leq n : 0 \leq P(t_i | t_0) \leq 1 \wedge t_0 \prec t_i).$$

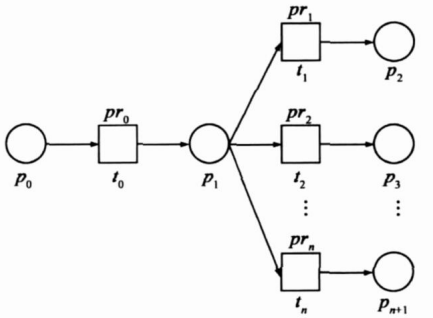


图2　物品工作流拆包模式

物品工作流网所支持的打包模式如图 3 所示,各个变迁之间满足以下条件:$pr_{n+1} = 1 \wedge (\forall 1 \leq i \leq n : 0 \leq P(t_{n+1} | t_i) \leq 1 \wedge t_i \prec t_{n+1})$.

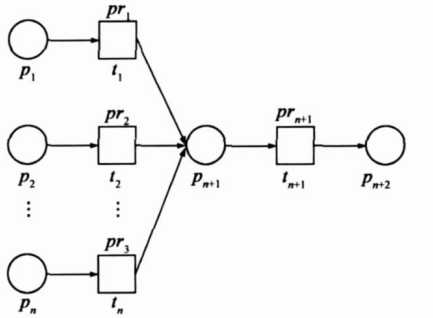


图3　物品工作流打包模式

3.3　时间及概率计算

对于 CWN 来说,时间和概率的计算是构造物品工作流网及进行其他相关分析的基础. 在给出它们的计算公式之前,首先对可能用到的一些相关函数和符号进行说明.

●路径 x:路径定义为库所的序列,一条路径 $x = x_1 x_2 x_3 \cdots x_n$ 表示从库所 x_1 开始,经 x_2, x_3 直到 x_n 结束的路径;

●路径长度函数 $length$:一条路径 x 的长度 $length(x)$ 表示该路径所包含的库所的个数;

●路径前缀函数 $prefix$:用来表示路径的前缀,$prefix(x, i)$ 表示路径 x 的前 i 个库所构成的一条路径. 其中 $prefix(x, 0) = \varnothing$,每条路径是其自身的一个前缀;

●路径物品计数 $count$:函数 $count$ 用来对通过路径的物品进行计数,$count(I, x)$ 表示通过路径 x 所有库所的某类物品 I 的数量,也就是通过最后一个库所的物品的数量;

●物品停留时长函数 $duration$:一个物品 i 在某个位置 l 所停留的时间为其离开该位置的时间与进入该位置的时间之差,$duration(i, l) = time\_out|_{i,l} - time\_in|_{i,l}$. 一类物品 I 在某个位置 l 所停留的平均时间为所有物品停留时间的总和与物品数目的商,即

$$duration(I,l)=\frac{\sum duration(i,l)}{count(I,l)}.$$

基于上述符号和函数可以计算一条路径执行的概率,它的值为该路径上各个位置被选择的概率的乘积,即 $P(x)=\prod_{k=1}^{length(x)} Poss(prefix(x,k))$,其中一个位置被选择的概率 $Poss$ 的计算公式如下:

$$Poss(x)=\frac{count(I,x)}{count(I,prefix(x,length(x)-1))}$$

根据上面的公式对第二节的数据集进行求解,可以得到图4所示的物品工作流网.该图中变迁上方的数字表示变迁具有发生权时发生的概率,库所上方的数字表示物品在该库所的平均停留时间.

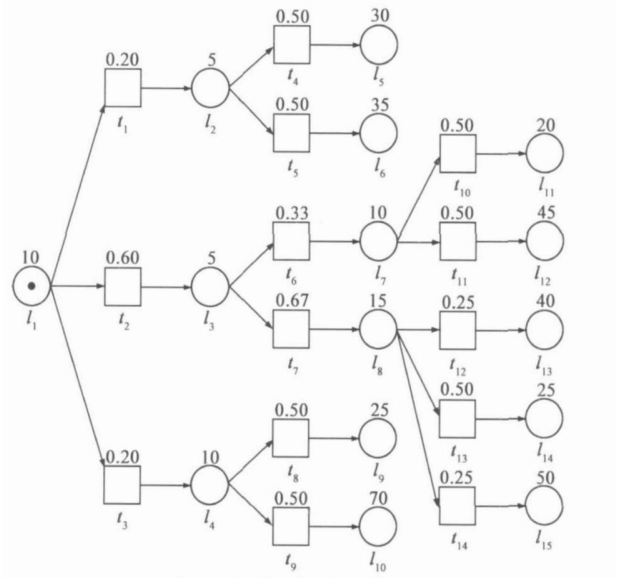


图4 物品工作流网实例

4 算法及性能分析

基于第三节的内容,我们设计、实现了原始数据集的预处理算法和物品工作流的挖掘算法,并进行了相关实验.

4.1 预处理算法

预处理算法包括数据清洗算法和数据聚合算法,其中数据清洗算法基于原始数据集,对某个位置同一个物品的不同时间记录进行清洗、合并,得到清洁数据集,具体实现过程见算法1.

算法1:原始数据集清洗算法
输入:原始数据集 RawData(CID,Location,Time)
输出:清洁数据集 CleanData(CID,Location,Time_In,Time_Out)
方法:

```
Table ←Empty HashTable;
// Table 用来记录一个物品在一个位置的进入和离开时间
FOR each Record in RawData
  IF Record.Time > Table[Record.CID + Record.Location].
    MaxTime THEN
      Table[Record.CID + Record.Location].MaxTime ←
      Record.Time; // 记录离开时间
  ELSE IF Record.Time < Table[Record.CID + Record.
    Location].MinTime THEN
      Table[Record.CID + Record.Location].MinTime ←
      Record.Time; // 记录进入时间
  ENDIF
ENDFOR
FOR each Record in Table
  Add(Record.CID,Record.Location,MinTime,MaxTime)
  to CleanData; // 生成清洁数据集
ENDFOR
```

数据聚合算法基于清洁数据集,对相同时间进入某个位置并且相同时间离开该位置的物品进行合并,得到物品聚合数据表和映射关系表,具体实现过程见算法2.

算法2:数据聚合算法
输入:清洁数据集 CleanData(CID,Location,Time_In,Time_Out)
输出:聚合数据表 AggregateData(SID,Location,Time_In,Time_Out),映射表 MAP(SID,SIDS/CIDS)
方法:

```
Table ←Empty HashTable;
// Table 用来记录相同地点、相同进入时间和离开时间的 CID 集合
FOR each Record in CleanData
  Add Record.CID to Table[Record.Location + Record.
  Time_In + Record.Time_Out];
  // 记录相同地点和时间的 CID 集合
ENDFOR
FOR each Record in Table
  Generate a corresponding SID; // 生成 SID
  Add(SID,Record.Location,Record.Time_In,Record.
  Time_Out) to AggregateData; // 生成聚合数据表
  Add(SID,CIDS) to MAP; // 生成映射表
ENDFOR
Intersect(MAP); // 对映射表进行变换操作,整理出 SID 之间的相互包含关系
```

4.2 物品工作流网挖掘算法

物品工作流网根据对象的不同可以分为单个物品的物品工作流网和一类物品的物品工作流网.其中单个物品的物品工作流网主要是标明单品的物品流向及其在各个位置的停留时间,它的挖掘算法直接基于物品的 CID 进行.一类物品的物品工作流网能够标识该类物品的不同流转路径、各个路径的执行概率以及物品在各个位置的平均停留时间等信息,其挖掘算法需要根据物品的类别判断某个物品是否属于该类,进而构造相应的物品工作流网.

单个物品的物品工作流网的挖掘算法如算法3所示,主要包括以下几个步骤:(1)在物品映射表中查找该物品所属 SID,查找物品聚合数据表,得到该 SID 所对应的位置以及每个位置的时间;(2)将位置作为库所,

位置上的时间作为库所的时间,增加一些变迁并按照时间顺序构造物品工作流网;(3) 查找 SID 所属的更高级别的 SID,查找物品聚合数据表,将位置作为库所,位置上的时间作为库所的时间,增加一些变迁并按照时间顺序构造物品工作流网,增加一个变迁,该变迁的后集为上一步骤生成的物品工作流网,该变迁的前集为本步骤生成的物品工作流网;(4) 递归运行步骤 3,直至找不到更高级别的 SID.

算法 3:单个物品的物品工作流挖掘算法

输入:聚合数据表 AggregateData(SID,Location,Time_In,Time_Out),映射表 MAP(SID,SIDS/CIDS),单个物品编码 CID

输出:物品工作流网 CWN

方法:

```
endpf ←True;//endpf 用来标记是否汇结库所
lastp ←Null;//初始化后继库所
SID′←the SID that contains CID in the MAP table;
WHILE(SID′≠NULL)
  FOR{each SID in AggregateData}
    IF(SID = SID′) THEN
      Add a place p to the set P;//添加库所
      θ(p) ←AggregateData[SID].TimeOut-AggregateData
      [SID].TimeIn;//计算库所停留时间
      IF endpf THEN
        endpf ←False;//修改汇结库所标记
      ELSE
        Add a transition t to the set T;
        Add a flow relation(p,t) to the set F;
        Add a flow relation(t,lastp) to the set F;
        ρ(t) ←1;//生成工作流网
      ENDIF
      lastp ←p;//记录后继库所
    ENDIF
  ENDFOR
  SID′←the SID that contains SID′ in the MAP table;
ENDWHILE
```

算法 3 的时间复杂度主要由两部分的和组成,第一部分是用来在 MAP 表中查找包含单个物品 CID 的 SID 的时间复杂度;第二部分又由两部分的乘积构成,其中一部分为在 MAP 表中递归判断是否存在更高层次 SID 的时间复杂度,该 SID 包含上一个循环中满足要求的 SID,另一部分是用来在 AggregateData 表中判断记录是否满足要求的时间复杂度与在 MAP 表中查找更高级别的 SID 的时间复杂度之和. 假定 MAP 表的记录规模为 m,AggregateData 表的记录规模为 n. 则第一部分的时间复杂度为 $O(m)$,第二部分的时间复杂度为 $O(m)O(m+n)$. 综上所述,算法 3 的时间复杂度为 $O(m)+O(m)O(m+n)=O(m^2+mn)$. 通常情况下,路径的长度远远小于 MAP 表的记录个数 m 的整数 k,此时算法 3 的时间复杂度为 $kO(m+n)=O(m+n)$.

一类物品的物品工作流网的挖掘算法如算法 4 所示,主要包括以下几个步骤:(1) 在物品映射表 MAP 中查找包含某类物品的 SID,得到该 SID 所包含的某类物品的数量;(2) 查找物品聚合数据表 AggregateData,得到该 SID 所对应的位置,确定位置对应的库所 p 或者生成新的库所 p,计算该类物品在库所 p 的数量和总时间;(3) 判断库所 p 是否存在后继库所 p',如果存在,增加一个变迁 t,该变迁 t 用来关联库所 p 和其后继库所 p';(4) 查找 SID 所属的更高级别的 SID,跳至步骤 2,直到找不到更高级别的 SID 为止;(5) 递归运行步骤 1 至步骤 4,直到找不到新的包含某类物品的 SID;(6) 计算物品在各个库所停留时间和每个变迁发生的概率,完成物品工作流网的构造.

算法 4:一类物品的物品工作流挖掘算法

输入:聚合数据表 AggregateData(SID,Location,Time_In,Time_Out),映射表 MAP(SID,SIDS/CIDS),物品种类 KID,开始时间 Time_Start,结束时间 Time_End

输出:物品工作流网 CWN

方法:

```
PMark ←Empty HashTable;
//PMark 用来标记是否新生成的库所
FOR {each SID in MAP table}
  IF(SID contains CID of KID kind) THEN
    SID′←SID;
    Gcount ←the number of CID;
    endpf ←True;//endpf 用来标记是否汇结库所
    lastpf ←False;//lastpf 用来标记是否有后继库所
    lastp ←Null;//初始化后继库所
    WHILE(SID′≠NULL)
      FOR{each SID in AggregateData}
        IF(SID = SID′∧Time_In > Time_Start ∧Time_Out <
        Time_End THEN
          //如果满足类别和时间约束
          IF(PMark[AggregateData[SID].Location] = NULL) THEN
          //如果是一个新的位置
          Add a place p to the set P;
          TotalTime(p) ←(AggregateData[SID].
          TimeOut-AggregateData[SID].TimeIn) ×Gcount;
          //计算库所的物品总停留时间
          TotalCount(p) ←Gcount;
          //计算库所的物品总数
          IF(not endpf) THEN
            //如果不是汇结库所
            Add a transition t to the set T;
            Add a flow relation(p,t) to the set F;
            Add a flow relation(t,lastp) to the set F;
          ENDIF
          lastp ←p;//记录后继库所
          lastpf ←True;//修改后继库所标记
        ELSE
          //如果不是一个新的位置
```

```
        p ←Aggregate[SID].Location;
        //取出相应的库所
        TotaTime(p) ←(AggregateData[SID].
        TimeOut-AggregateData[SID].TimeIn) ?
        Gcount + TotalTime(p);
        //计算库所的物品总停留时间
        TotalCount(p) ←Gcount + TotalCount(p);
        //计算库所的物品总数
        IF lastpf THEN
          //如果有后继库所
          Add a transition t to the set T;
          Add a flow relation(p,t) to the set F;
          Add a flow relation(t,lastp) to the set F;
        ENDIF
          lastp ←p;//记录后继库所
          lastpf ←True;//修改后继库所标记
        ENDIF
      ENDIF
    ENDFOR
    SID′←the SID that contains SID′in the MAP table;
    endpf ←False;//修改汇结库所标记
  ENDWHILE
 ENDIF
ENDFOR
Annotate(CWN);//根据相应公式计算变迁发生概率和库所停留时间,
标注物品工作流网;
```

算法 4 的时间复杂度取决于构造物品工作流网的迭代次数,而迭代次数是下面 3 部分时间复杂度的乘积:用来在 MAP 表中判断某个 SID 是否包含某类物品的时间复杂度;用来在 MAP 表中递归判断是否存在更高层次 SID 的时间复杂度,该 SID 包含上一个循环中满足要求的 SID;以及用来在 AggregateData 表中判断记录是否满足条件的时间复杂度与用来在 MAP 表中查找更高级别 SID 的时间复杂度之和. 假定 MAP 表的记录规模为 m,AggregateData 表的记录规模为 n. 则算法 4 的时间复杂度为 $O(m)\,O(m)\,(O(m)+O(n)) = O(m^2(m+n)) = O(m^3+m^2n)$. 通常情况下,路径的长度是远远小于 MAP 表的记录个数 m 的整数 k,此时算法 4 的时间复杂度为 $kO(m^2+mn) = O(m^2+mn)$.

4.3 性能分析

我们对基于 RFID 数据集的物品工作流挖掘方法进行了模拟实验,以验证相关算法的性能. 原型系统采用 Java 语言开发,实验的硬件环境为 P4 2.4GHz 的 CPU 和 1GB 内存,操作系统为 WindowsXP 专业版,Java 运行环境为 JDK 1.6.0_03.

实验一(预处理算法性能分析)

假定每个物品被同一个读写器读取 10 次,路径长度为 4,物品的批量大小分别为 100、50、20、10,图 5 是不同的数据规模下的空间消耗情况. 从该图可以看出,当原始数据集的规模较小时,几种数据集的空间开销都比较小,但是当原始数据集的规模越来越大时,清洁数据集和聚合数据集的空间开销的优势越来越明显. 当原始数据集的规模为 10 万条记录时,它的空间开销为 3.6M,清洁数据集和聚合数据集的开销分别为 460K 和 240K;当原始数据集的规模为 20 万条记录时,它的空间开销为 7.1M,清洁数据集和聚合数据集的开销分别是 900K 和 470K;当原始数据集的规模是 100 万条记录时,它的空间为开销 35M,清洁数据集和聚合数据集的内存开销分别是 4.4M 和 2.3M. 综上,数据清洗算法和数据聚合算法可以显著地减少数据集的规模,从而减少空间的开销.

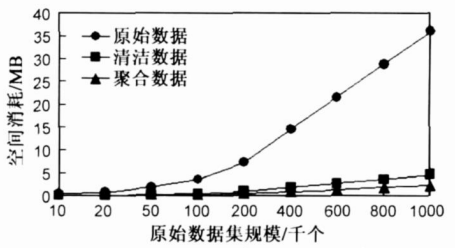


图5 预处理对空间开销的影响

物品批量的大小和物品在每个位置的读取次数都会对预处理算法的效果产生影响,进而影响空间开销. 通过分析和实验结果可以得知,物品批量和物品在每个位置的读取次数都和空间开销成反比关系. 在相同条件下,读取次数越多空间开销越小,读取次数越少空间开销越大.

实验二(挖掘算法性能分析)

通过实验可以发现单个物品的物品工作流网构造具有较高的效率,在此不再赘述,下面主要对一类物品的物品工作流网挖掘算法的性能进行说明. 假定原始数据集的规模分别为 5 千、1 万、2 万、5 万、10 万、20 万、40 万、60 万、80 万和 100 万条记录,且每个物品被同一个读写器读取 10 次,路径长度为 4,物品的批量大小分别为 100,50,20,10. 则得到的一类物品的物品工作流网构造的性能如图 6 所示. 从该图可以看出,当原始数据集的规模小于 10 万时,构造一类物品的物品工作流网的时间小于 50 毫秒,但是随着数据集规模的增加,物品工作流网构造的性能逐渐下降,当数据集的规模达到 100 万时,需要大约 780 毫秒的时间才能构造出相应的物品工作流网.

5 结论

本文提出了一种基于 RFID 数据集的物品工作流挖掘方法,定义了一种基于 Petri 网的物品工作流网,针对物品工作流网所支持的物品工作流模式,讨论了变迁之间所满足的条件和约束,给出了物品在库所的停

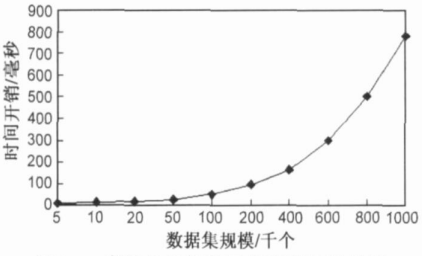


图6　一类物品的物品工作流网构造性能

留时间和变迁发生概率的计算方法,最后设计、实现了相关算法,并进行了实验.

物品工作流网能够表示物品的流转路径、各个路径的执行概率、以及物品在各个位置的停留时间等信息.基于物品工作流网,还可以发现物品流转过程中的异常节点信息,即在某个时间段内物品在这些节点对应位置的停留时间出现异常.上述各种信息可以用来为供应链过程的管理和优化提供指导.RFID 编码包含丰富的信息,下一步将对如何充分利用这些信息进行工作流挖掘,以得到语义更加丰富的物品工作流进行研究.此外,考虑利用 Petri 网分析上的优势,对物品工作流网的一些结构性质以及供应链过程的性能进行分析.

参考文献:

[1] Jonathan E Cook, Alexander L Wolf. Automating process discovery through event-data analysis[A]. Proceedings of the 17th International Conference on Software Engineering (ICSE '95)[C]. ACM press, 1995. 73 - 82.

[2] R Agrawal, D Gunopulos, F Leymann. Mining process models from workflow logs[A]. Proceeding of the 6th International Conference on Extending Database Technology (EDBT '06)[C]. Springer-Verlag, 1998. 469 - 483.

[3] W M P van der Aalst. The application of Petri nets to workflow management[J]. The Journal of Circuits, Systems and Computers, 1998, 8(1): 21 - 66.

[4] W M P van der Aalst, K M van Hee. Workflow Management: Models, Methods and Systems[M]. London: The MIT Press, 2002.

[5] W M P van der Aalst, A J M M Weijter, L Maruster. Workflow mining: Discovering process models from event logs[J]. IEEE Transactions on Knowledge and Data Engineering (TKDE), 2004, 16(9): 1128 - 1142.

[6] H Gonzalez, J W Han, X L Li. Mining compressed commodity workflows from massive RFID data sets[A]. Proceedings of the 15th ACM International Conference on Information and Knowledge Management[C]. ACM press, 2006. 162 - 171.

[7] EPCglobal. EPCglobal Tag Data Standard TDS Version 1.3.1. The EPCglobal Standards Development Process [DB/OL]. http://www.epcglobalinc.org/standards/tds/tds_1_3_1-standard-20070928.pdf. 2007 - 09.

[8] Ken Sakamura. Ubiquitous ID Technologies 2008. Ubiquitous id center [DB/OL]. http://www.uidcenter.org/pdf/UID910-W001-080226_en.pdf. 2008.

[9] 袁崇义. Petri 网原理与应用[M]. 北京:电子工业出版社, 2005.
Yuan Chongyi. Principals and Application of Petri Nets[M]. Beijing: Publishing House of Electronics Industry, 2005. (in Chinese).

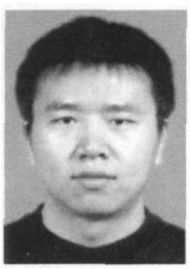
[10] C Girault, R Valk. Petri Nets for Systems Engineering: A Guide to Modeling, Verification, and Applications[M]. New York: Springer-Verlag, 2003.

[11] 林闯. 随机 Petri 网和系统性能评价(第二版)[M]. 北京:清华大学出版社, 2006.
Lin Chuang. Stochastic Petri Nets and System Performance Analysis (2nd edition)[M]. Beijing: Tsinghua University Press, 2006. (in Chinese).

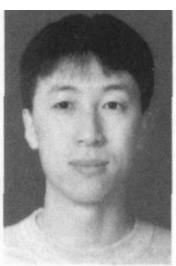
[12] C Alexander, S Ishikawa, M Jacobson, I Fiksdahl-King, S Angel. A Pattern of Language[M]. Oxford University Press, 1977.

[13] W M P van der Aalst, A H M ter Hofstede, B Kiepuszewski, A P Barros. Workflow patterns[J]. Distributed and Parallel Databases, 2003, 14(1): 5 - 51.

作者简介:



顿海强　男,1977 年生,河南长垣人,博士研究生,主要研究领域为软件工程、工作流技术和 Web 服务组合.
Email: alandun@gmail.com



赵　文　男,1967 年出生,辽宁大连人,博士,副研究员,主要研究领域为软件工程和工作流技术.