

# RFID 信息服务中发布/ 订阅系统 若干关键技术问题研究

刘殿兴<sup>1,3</sup>, 赵 文<sup>2,3</sup>, 邓鹏鹏<sup>1,3</sup>, 张世琨<sup>2,3</sup>, 王立福<sup>2,3</sup>

(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. 北京大学信息科学技术学院软件研究所 高可信软件技术教育部重点实验室, 北京 100871)

**摘 要:** 本文将描述逻辑中基于个体的推理方法引入到 RFID 信息服务的发布/ 订阅系统中, 利用时间本体来描述事件之间的时序关系, 通过判断事件断言集与订阅的一致性来解决语义匹配以及与时序相关的复合事件与复合订阅的匹配问题, 并给出了订阅语言和匹配算法. 实验结果表明, 匹配算法的效率能够满足实际应用的需要.

**关键词:** 发布/ 订阅; RFID; 描述逻辑; 时间本体

**中图分类号:** TP393 **文献标识码:** A **文章编号:** 0372-2112 (2008) 12A-094-08

## Research on Key Techniques of Publish/ Subscribe System in RFID Information Services

LIU Dian-xing<sup>1,3</sup>, ZHAO Wen<sup>2,3</sup>, DENG Peng-peng<sup>1,3</sup>, ZHANG Shi-kun<sup>2,3</sup>, WANG Li-fu<sup>2,3</sup>

(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 (Ministry of Education), School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)

**Abstract:** We introduced individual-based reasoning method in description logics into the publish/ subscribe system of RFID information services, and the temporal relations between events were depicted using ontology of time. Semantic-based matching and the matching between composite events and composite subscriptions which have temporal relations were solved by judging the consistency between event assertion set and subscription. Also, a subscribing language and a matching algorithm were proposed. The results of testing show that the efficiency of the matching algorithm is able to satisfy the need of application.

**Key words:** publish/ subscribe; Radio Frequency Identification (RFID); description logics; ontology of time

### 1 引言

RFID (Radio Frequency Identification) 作为一种高效的非接触自动识别技术是当前国内外热门的研究课题之一, 世界各国目前都在投入巨资来研究 RFID 开环应用技术, 而在 RFID 开环应用方面我国目前还处于起步阶段. 要建立起 RFID 跨地区、跨行业乃至全社会的规模化应用, 需要建立一套可靠高效的 RFID 公共服务基础设施和信息共享机制, 目前 EPCglobal 组织已经提出了这样一套体系结构框架<sup>[1]</sup>. 在这个体系结构框架中, RFID 信息服务处于捕获应用 (Capturing Application) 与上层访问应用 (Accessing Application) 之间, 为上层访问应用提供查询业务事件信息和物品信息的标准接口.

RFID 信息服务以轮询和发布/ 订阅两种方式为用

户提供查询服务<sup>[1]</sup>, 本文以 RFID 信息服务中的发布/ 订阅系统作为主要研究内容. 以物流领域为例, 关注产品流通的用户可以按照发布/ 订阅的方式向物流各环节上的 RFID 信息服务发出订阅, 订阅自己感兴趣的产品流通信息, 生产商在产品出厂时为产品贴上 RFID 标签, 当产品流通到物流中的某个环节时, 读写器读取产品上的标签数据并结合业务逻辑信息来产生业务事件, 如果该业务事件符合用户的订阅要求, 该物流环节上的 RFID 信息服务就会自动通知该用户, 从而达到用户实时跟踪物品流通的目的.

在实际应用中, 要求 RFID 信息服务中的发布/ 订阅系统具有一些新的特性. (1) 需要根据事件的业务语义来进行事件与订阅的匹配. 以物流领域为例, 货物的业务状态分为 selling 和 transporting, 而 transporting 包括

shipping, airlifting, landCarrying, 并且 landCarrying 包括 roadCarrying 和 railCarrying 等多种. 用户有如下订阅需求:在某个物流环节,当一批货物的业务状态变为 transporting 时向用户发出通知. 用户该如何表达订阅? 如果该物流环节的 RFID 信息服务中产生了业务状态是 railCarrying 的业务事件,系统如何知道该事件匹配了用户的订阅? (2) 需要支持与时序相关的复合事件与复合订阅的匹配. 例如某企业在出货操作发生时需要检查货物的状态是否完好,企业的管理人员有如下的订阅需求:如果某批货物在出货之前被解包装则向管理人员发出通知(或是如果某批货物在出货之前没有经过包装则向管理人员发出通知). 管理人员该如何表达上述订阅? 系统又该如何检测是否产生了这样的复合事件? 针对这些实际应用需求,本文给出了表达用户订阅的订阅语言以及匹配算法.

## 2 RFID 信息服务及相关研究工作介绍

### 2.1 RFID 信息服务

在 EPCglobal 提出的体系结构框架中,RFID 信息服务位于捕获应用与上层访问应用之间,如图 1 所示. RFID 中间件负责将读写器读取到的标签数据进行收集、过滤和分组,并形成应用层事件 (Application Level Event 简称 ALE),它是不包含业务语义的<sup>[2]</sup>. ALE 形成以后,捕获应用 (Capturing Application) 按照预先设置的业务逻辑向 ALE 中添加业务语义最终形成业务事件 (Business Event 简称 BE),并调用 RFID 信息服务提供的下层捕获接口存入 RFID 信息服务中. 上层的访问应用可以按照“推”、“拉”两种模式访问 RFID 信息服务中的信息.

RFID 信息服务中有四类业务事件<sup>[3]</sup>,分别是 AggregationEvent、QuantityEvent、TransactionEvent 和 ObjectEvent. 其中 AggregationEvent 刻画了伴随着物品包装与解包装的业务逻辑(这里的包装既可以指货物移上托盘,也可以指物品的装箱操作);QuantityEvent 刻画了

伴随着某类物品数量的业务逻辑;TransactionEvent 刻画了伴随着物品交易的业务逻辑;ObjectEvent 刻画了除前三种之外的其他业务逻辑. 这四类事件包含的部分属性及含义如表 1 所示:

表 1 事件的部分属性及含义

名称属性	含 义
eventTime	发生时间
recordTime	记录时间
eventTimeZoneOffset	相对 UTC 的时区补偿
action (枚举型)	{ADD, OBSERVE, DELETE}
bizStep	业务步骤
disposition	分拣状态
readPoint	逻辑读写器名称
bizLocation	业务位置

### 2.2 相关研究工作介绍

RFID 公共服务是一个相对较新的研究领域,而 RFID 信息服务是 RFID 公共服务的一个重要组成部分,最早的关于 RFID 信息服务的标准于 2007 年 4 月由 EPCglobal 提出<sup>[3]</sup>,最新的标准于 2007 年 9 月发布<sup>[4]</sup>,该标准还在不断的更新和完善中. 由于研究领域较新,目前可以参考的其他资料较少. 国内外在 RFID 信息服务方面的研究主要集中在系统原型的实现上,包括瑞士圣加仑大学 Auto-ID 实验室开发的称为 Accada 的系统原型和 MIT Auto-ID 实验室开发的系统原型. 这两个系统原型中的发布/ 订阅系统是基于 Map 的,不支持语义匹配和与时序相关的复合事件同复合订阅的匹配. 国内目前中科院软件所和北京大学也在从事 RFID 信息服务的相关研究,并成功开发出了系统原型.

从发布/ 订阅系统的研究来看,针对语义匹配的问题,近年来有如下的一些相关研究:S- ToPSS<sup>[5]</sup>是 Toronto 大学的一个研究成果,该系统在基于 Map 的发布/ 订阅系统的基础上,利用 DAML + OIL (DAML 是 DARPA Agent Markup Language 的简写, OIL 是 ontology interchange language 的简写) 语言,建立各属性之间的关系,从而可以在一定程度上支持语义异构的事件;CREAM<sup>[6]</sup>是在基于主题的发布/ 订阅系统的基础上,利用事件适配器结合系统中存储的本地信息将原始事件转变成一种 MIX 结构,最终根据事件的内容得到其所属的主题,并将事件发送给主题的订阅者;中科院软件所提出了一种 OPS 系统<sup>[7]</sup>,该系统中的事件用 RDF 图来表示,订阅条件被表示成图模式,采用遍历事件图的方式实现事件与订阅的语义匹配. 在支持复合订阅的研究方面,有如下的一些相关研究:SIENA<sup>[8]</sup>和 Rebeca<sup>[9]</sup>允许顺序事件的检测;文献[10]给出了能够表达与、或、顺序和循环的复合订阅语言;文献[11]给出了基于事件评估树的复合订阅匹配算法;文献[12]给出了按照复合订阅生成自动机来检测复合事件的方法.

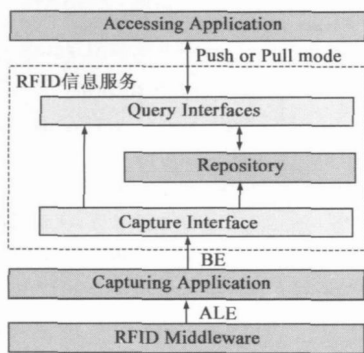


图1 RFID信息服务的边界

与这些研究工作相比,本文结合 RFID 的实际应用需求,将描述逻辑中基于个体的推理方法引入到 RFID 信息服务的发布/订阅系统中,利用时间本体来描述事件之间的时序关系,通过判断事件断言集与订阅的一致性来解决语义匹配以及与时序相关的复合事件与复合订阅的匹配问题.

### 3 描述逻辑与时间本体

#### 3.1 描述逻辑

描述逻辑<sup>[13]</sup>作为一种知识表示方法已经在信息科学的多个领域得到了广泛的应用,特别是描述逻辑中基于知识的推理方法能够让系统理解语义,使之成为当前的研究热点.描述逻辑中的推理分为基于概念的推理和基于个体的推理两大类,其中基于个体推理的一个重要内容就是实例(个体)检查.所谓实例检查就是根据现有的知识库来判断某个个体  $a$  是否满足概念  $C$  的约束.本文将用户的订阅看作是概念,每一个事件(包括原子事件和复合事件)看作是一个个体,判断事件  $e$  是否匹配订阅  $Sub$ ,就是根据现有的知识库来判断个体  $e$  是否满足概念  $Sub$  的约束.

基于描述逻辑的知识库由两部分组成,分别是 TBox 和 ABox. TBox 是公理的集合,代表了系统所具有的领域知识;ABox 用于存储断言,表示系统中存在的事实.断言有两种形式,分别是  $C(a)$  (概念断言)和  $R(b, c)$  (角色断言,  $R$  是 Role 的简写,也可以理解为属性),  $C(a)$  表示个体  $a$  满足概念  $C$  的约束,  $R(b, c)$  表示个体  $b$  的属性  $R$  的一个取值是  $c$ .

#### 3.2 时间本体

时间本体是 DAML 工程中的一个重要的内容,它是对时态实体以及时态实体之间联系的形式化说明.在时间本体中有且仅有两种时态实体(temporal entity)<sup>[14]</sup>,它们分别是:Instant 和 Interval.每个 Interval 必须存在开始时刻,但是可以没有结束时刻,而每个 Instance 由一个唯一的时刻  $t$  来表示,也可以看作是开始时刻和结束时刻相等的特殊的 Interval.本文将事件看作是一种存在结束时刻的 Interval,其开始和结束时刻分别用  $e.begin$  和  $e.end$  来表示,这样我们就可以利用时间本体中表达 Interval 之间时序关系的谓词来表达事件之间的时序关系.

在时间本体中,总共有 13 个表达 Interval 之间关系的二元时态谓词(temporal predication),它们分别是 *intBefore*、*intMeets*、*intOverlaps*、*intStarts*、*intFinishes*、*intDuring*、*intEquals*、*intAfter*、*intMetBy*、*intOverlappedBy*、*intStartedBy*、*intFinishedBy* 和 *intContains*.其中,后六个时态谓词可以用前六个时态谓词来定义(例如  $intAfter(T_1, T_2) = intBefore(T_2, T_1)$ ),因而前六个时态谓词加上 *intEquals* 构

成了互不相交且联合完备的最小子集.本文就选择这 7 个时态谓词作为表达事件之间时序关系的谓词,为了运算的需要,又分别给出了这 7 个时态谓词的否定含义,如图 2 所示.

<i>intEquals</i> ( $e_1, e_2$ )	( $e_1.begin = e_2.begin$ )	( $e_1.end = e_2.end$ )
<i>intBefore</i> ( $e_1, e_2$ )	$e_1.end < e_2.begin$	
<i>intMeets</i> ( $e_1, e_2$ )	$e_1.end = e_2.begin$	
<i>intOverlaps</i> ( $e_1, e_2$ )	( $e_1.begin < e_2.begin$ )	( $e_2.begin < e_1.end$ )
	( $e_1.end < e_2.end$ )	
<i>intStarts</i> ( $e_1, e_2$ )	( $e_1.begin = e_2.begin$ )	( $e_1.end < e_2.end$ )
<i>intFinishes</i> ( $e_1, e_2$ )	( $e_2.begin < e_1.begin$ )	( $e_1.end = e_2.end$ )
<i>intDuring</i> ( $e_1, e_2$ )	( $e_2.begin < e_1.begin$ )	( $e_1.end < e_2.end$ )
<i>intUnEquals</i> ( $e_1, e_2$ )	( $e_1.begin \neq e_2.begin$ )	( $e_1.end \neq e_2.end$ )
<i>intUnBefore</i> ( $e_1, e_2$ )	$e_2.begin \leq e_1.end$	
<i>intUnMeets</i> ( $e_1, e_2$ )	$e_1.end < e_2.begin$	
<i>intUnOverlaps</i> ( $e_1, e_2$ )	( $e_2.begin \leq e_1.begin$ )	( $e_1.end \leq e_2.begin$ )
	( $e_2.end \leq e_1.end$ )	
<i>intUnStarts</i> ( $e_1, e_2$ )	( $e_1.begin \neq e_2.begin$ )	( $e_2.end \leq e_1.end$ )
<i>intUnFinishes</i> ( $e_1, e_2$ )	( $e_1.begin \neq e_2.begin$ )	( $e_1.end \neq e_2.end$ )
<i>intUnDuring</i> ( $e_1, e_2$ )	( $e_1.begin \neq e_2.begin$ )	( $e_2.end \leq e_1.end$ )

图 2 时态谓词及其否定形式的含义

### 4 订阅语言

结合描述逻辑与时态谓词的订阅语言用文法的形式描述如图 3 所示,其中 ASub 表示原子订阅,CSub 表示复合订阅,  $C, D$  表示原子概念,“ $\wedge$ ”和“ $\vee$ ”是描述逻辑中的概念运算符,“ $\cup$ ”、“ $\&$ ”和“ $\mid$ ”是构成复合订阅的运算符.

$\mid$
$\mid \& \mid$
$op > \mid < \mid \mid = \mid$
<i>Equals</i>   <i>Before</i>   <i>Meets</i>   <i>Overlaps</i>   <i>Starts</i>
<i>Finishes</i>   <i>During</i>   <i>UnEquals</i>   <i>UnBefore</i>   <i>UnMeets</i>
<i>UnOverlaps</i>   <i>UnStarts</i>   <i>UnFinishes</i>   <i>UnDuring</i>
Sub ASub   CSub
ASub ASub <sub>1</sub>   ASub <sub>1</sub> ASub <sub>1</sub>   ASub <sub>1</sub> AExpr
AExpr ASub <sub>1</sub>   AExpr AExpr
AExpr (ASub <sub>1</sub> ASub <sub>1</sub> )
ASub <sub>1</sub> C   $\forall R. RoleExpr$   $\exists R. RoleExpr$
$\forall R. (type\ op\ value)$   $\exists R. (type\ op\ value)$
RoleExpr C   (C D)   RoleExpr
(C RoleExpr)   (RoleExpr C)
CSub ASub   CSub( $t$ )   C(Expr( $t$ ))   CSub CSub
CSub CExpr   CExpr CSub   CExpr CExpr
CExpr (CSub CSub)

图 3 订阅语言

对订阅语言的几点说明:

(1) 原子订阅刻画了一类原子事件,它们的属性满

足原子订阅的要求;复合订阅刻画了一类复合事件,每个复合事件中的原子事件不仅满足复合订阅中对应的原子订阅要求,而且满足复合订阅要求的复合关系。

(2) 复合订阅  $ASub_1 \wedge ASub_2$  规约了  $[e_1 e_2]$  这样的复合事件,满足  $ASub_1(e_1) \wedge ASub_2(e_2) \wedge int(e_1, e_2)$ , 其中  $int$  对应图 2 中的时态谓词。

(3) 复合订阅  $ASub_1 \& ASub_2$  规约了  $[e_1 e_2]$  这样的复合事件,满足  $ASub_1(e_1) \wedge ASub_2(e_2)$ ;

(4) 复合订阅  $ASub_1 | ASub_2$  规约了  $[e_1 e_2]$  这样的复合事件,满足  $ASub_1(e_1) \wedge ASub_2(e_2)$ ;

(5) 订阅语言没有引入描述逻辑中的数量约束,这是由于 RFID 信息服务中只有 QuantityEvent 涉及到数量约束,并且这种数量约束只是针对 RFID 标签的数量,不存在基于数量的语义匹配,不引入它可以大大提高匹配的效率;

(6) “type op value”实质上是对某个概念进行更加详细的规约,我们可以把它看作一个新的概念参与匹配;

(7) 订阅  $CSub(t)$  (或  $CExpr(t)$ ) 刻画了“满足订阅  $CSub$  (或  $CExpr$ ) 的事件没有发生”这样一类事件,这样的事件不会触发系统的匹配过程,必须要系统去主动检测,时间  $t$  就规约了检测的时间间隔。

利用上面给出的订阅语言,我们可以表达引言中给出的订阅需求,如图 4 所示。其中,  $ASub$  订阅了业务步骤是 transporting 的交易事件,  $CSub$  订阅了在出货之前货物被解包装的复合事件,其中  $ASub_1$  规约了解包装的业务事件,  $ASub_2$  规约了出货的业务事件。为了简化表达,订阅中都省略了对具体某批货物的限制。

$ASub = TransactionEvent$	$\exists bizStep. transporting$
$ASub_1 = AggregationEvent$	$\exists Action (ActionType = DELETE)$
$ASub_2 = TransactionEvent$	$\exists disposition. readyForSale$
$CSub = ASub_1 \text{ Before } ASub_2$	

图 4 订阅举例

### 5 匹配算法

本文将用户的订阅看作是概念,每一个事件看作是一个个体,采用描述逻辑中基于个体推理的实例(个体)检查方法作为匹配算法的理论基础,定理如下:

**定理**  $A \models Sub(e)$  当且仅当  $S = A \setminus \{Sub(e)\}$  是不一致的。其中  $A$  表示事件断言集,  $Sub(e)$  表示事件  $e$  匹配订阅  $Sub$ 。

该定理与描述逻辑中基于个体推理的实例检查的判定定理相似,证明从略。“一致性”是描述逻辑中的术语,由于篇幅所限就不再详细解释,关于一致性的问题可参见文献[13],本文在 5.2.2 和 5.3.2 节中给出了判断一致性的准则。事件断言集  $A$  中的断言刻画了事件  $e$  的所有属性取值。

该定理给出了根据事件断言集  $A$  来判断事件  $e$  是否匹配订阅  $Sub$  的方法。具体来说,匹配算法从  $S = A \setminus \{Sub(e)\}$  开始,将  $S$  按照一致性保持规则(见图 6、8)进行变换,直到不能变换为止(称为  $S$  的完全形式),如果  $S$  的完全形式是不一致的(当且仅当  $\forall A_i, S$  是不一致的),则  $S$  是不一致的,由定理可以得出事件  $e$  匹配订阅  $Sub$ , 否则相反。 $S$  的完全形式可以通过如下方式得到:当一条规则应用于  $S$  中的某个集合元素  $A_i$  时,用变换的结果来替换  $A_i$ , 如此迭代下去,直到  $S$  中的所有集合元素都不能应用一致性保持规则为止。所谓一致性保持规则就是指,对于任意  $A_i$ , 且  $A_i$  是一致的,如果应用规则后的结果中有一个是一致的就称该规则保持了一致性。本文提出的规则是具有一致性的,由于相对简单,验证从略。

#### 5.1 TBox 初始化

为了提高匹配算法的效率,需要将一些相对固定的计算放到匹配之前来完成,包括对 TBox 的初始化和对用户订阅的初始化。对 TBox 的初始化工作包括以下两个部分:

(1) 对公理进行完全扩展;

(2) 对于用枚举方式定义的原子概念,将其中的元素按照字母顺序进行排序。

概念在 TBox 中以概念符号的形式来表示,概念符号分为两类:分别是名称符号(name symbols)和基本符号(base symbols)。出现在公理表达式左边的符号叫做名称符号,只出现在公理表达式右边的符号叫做基本符号。所谓对 TBox 进行完全扩展就是通过迭代使 TBox 中所有公理表达式的右侧只包含基本符号。

假设有图 5 的 TBox,显然 bizState 和 transporting 是名称符号, TBox 是其完全扩展后的结果:

$TBox = \{bizState \text{ selling } transporting,$
$\text{transporting shipping airlifting landCarrying}\}$
$TBox = \{bizState \text{ selling shipping airlifting landCarrying},$
$\text{transporting shipping airlifting landCarring}\}$
其中, $landCarrying = \{railCarrying, roadCarrying\}$

图 5 完全扩展举例

由于公理通常用概念定义(形如  $C \sqsubseteq D$ )和概念特化(形如  $C \sqsubseteq D$ )两种方式来表达,而概念特化可以转换成概念定义的形式(例如,公理  $C \sqsubseteq D$  可以转换成  $C \sqsubseteq \bar{C} \sqcup D$ , 其中  $\bar{C}$  表示性质与  $C$  相反的一个新概念),因而本文讨论的 TBox 中的公理都是概念定义的形式。

由于 TBox 中的公理代表了系统所具有的知识,这些知识是相对稳定的,因而上面两项工作在系统初始化时只需做一次(只有当领域知识改变的时候才须重做),不会对匹配工作带来额外的负担。

#### 5.2 原子事件与原子订阅的匹配

### 5.2.1 原子订阅的初始化

原子订阅初始化主要是为了得到  $S = \{\{ \text{ASub}(e)\}\}$  (这里的  $e$  只是一个占位符) 的完全形式, 这样可以加快匹配过程中计算  $S = \{A \mid \text{ASub}(e)\}$  的完全形式的速度, 分为以下几步:

(1) 结合 TBox 中的公理, 计算 ASub 的完全扩展形式 (ASub 中的每个概念符号都是基本符号);

(2) 利用德摩根定律, 计算 ASub (使用步骤 1 的结果) 的否定一般式 (否定符号只出现在概念符号前面的表达式), 并应用图 6 所示的一致性保持规则, 计算  $S = \{\{ \text{ASub}(e)\}\}$  的完全形式。

- 规则
条件: $(C_1 \sqcap C_2)(x) \sqsubseteq A, C_1(x) \sqsubseteq A, C_2(x) \sqsubseteq A$
操作: $A = A \sqcap \{C_1(x), C_2(x)\}$
- 规则
条件: $(C_1 \sqcup C_2)(x) \sqsubseteq A, C_1(x) \sqsubseteq A, C_2(x) \sqsubseteq A$
操作: $A = \{C_1(x)\}, A = \{C_2(x)\}$
∃- 规则
条件: $(\exists R. C)(x) \sqsubseteq A, A$ 中不存在个体 $z$ , 满足 $C(z) \sqsubseteq A, R(x, z) \sqsubseteq A$
操作: $A = A \sqcap \{C(y), R(x, y)\}$ , 其中 $y$ 是没有在 $A$ 中出现的个体
∀- 规则
条件: $(\forall R. C)(x) \sqsubseteq A, R(x, y) \sqsubseteq A, C(y) \sqsubseteq A$
操作: $A = A \sqcap \{C(y)\}$

图 6 断言展开的一致性保持规则

### 5.2.2 匹配算法

判断原子事件  $e$  是否匹配原子订阅 ASub 的算法分如下几步:

(1) 系统为原子事件  $e$  建立一个原子事件断言集  $A$ , 并通过断言生成算法生成  $A$  中的断言;

(2) 应用图 6 所示的一致性保持规则, 计算  $S = \{A \mid \text{ASub}(e)\}$  的完全形式. 由于订阅初始化中已经得到  $S = \{\{ \text{ASub}(e)\}\}$  的完全形式, 因而只需从  $S = \{A_i \mid \forall A_k, S, A_i = A \sqcap A_k\}$  开始计算;

(3) 判断  $S$  的完全形式是否一致, 从而得到  $e$  是否匹配 ASub.

判断  $S$  是否一致, 就是判断  $S$  中的每个集合元素  $A_i$  是否一致, 当且仅当满足下面两种情况之一时,  $S$  中的集合元素  $A_i$  是不一致的:

$$\{C(x), \neg C(x)\} \subseteq A_i;$$

$$\{C(x)\} \subseteq A_i \quad (x \notin C)$$

原子事件与原子订阅的匹配算法如图 7 所示.

### 5.2.3 匹配示例

下面以引言中的订阅需求来举例说明匹配过程, TBox 采用图 5 给出的例子.

用户的订阅需求: 如果某批货物的业务状态变为

```

//计算用户订阅的完全扩展形式
void createSubscriptionExtension(AtomicSubscription ASub) {
  for each concept C in ASub
    replace C with its expression in TBox;
}
//计算用户订阅的否定一般式
void createNegationNormalForm(AtomicSubscription ASub){
  for each element ele in ASub from left to the right {
    switch (ele) {
      case "¬" ://注:case语句后的参数不能为字符串, 这里只是伪码
        flag = ~flag; // initial value is false
      case "(" :
        if (ele.last == "¬")
          s.push(~flag); //s is a stack
        else
          s.push(flag);
        result = result.add("(");
      case ")" :
        result = result.add(")");
        flag = s.pop();
      default:
        if (flag == true)
          //logicalOpposite用于得到ele的逻辑相反值, 例如“∩”的
          //逻辑相反值为“∪”, 概念“A”的逻辑相反值为“¬A”
          result = result.add(logicalOpposite(ele));
        else
          result = result.add(ele);
    }
  }
  ASub.setContent(result.getContent());
}
//计算断言集的完全形式. 在订阅初始化和匹配过程中都要用到.
void createCompleteForm(AssertionSet S) {
  for each element A in S {
    if Ai satisfy ∃-rule {
      replace Ai with A' in ∃-rule
      createCompleteForm(A');
    }
    if Ai satisfy ∀-rule {
      replace Ai with A' and A'' in ∪-rule
      createCompleteForm(A');
    }
  }
  createCompleteForm(A'');
  if Ai satisfy ∃-rule {
    replace Ai with A' in ∃-rule
    createCompleteForm(A');
  }
  if Ai satisfy ∀-rule {
    replace Ai with A' in ∀-rule
    createCompleteForm(A');
  }
}
//创建事件断言集
void createEventAssertions(Event e) {
  for each property p in e.property_list {
    if p is specified using concept C
      A = A ∪ {C(e)};
    if p is specified using role R
      A = A ∪ {R(e, p.value)};
  }
}
//判断断言集是否一致
boolean isConsistent(AssertionSet S) {
  for each element Ai in S {
    //按照概念符号前面是否有“¬”将概念符号分成两组g0和g1
    group Elements of Ai By Negation to form g0 and g1
    for each group g {
      //按照概念符号字符串进行排序
      sortElementsByNames(g);
    }
    for each C(x) in g0 {
      if (∃¬C(x) ∈ g1) { flag = true; break; }
    }
    if (flag) { flag = false; continue; }
    for each element C(x) in Ai {
      if (x ∉ C) { flag = true; break; }
    }
    if (flag) { flag = false; continue; }
  }
  return true;
}
return false;

```

图 7 原子事件与原子订阅的匹配算法

transporting 时向用户发出通知.

用户的订阅表达为:

$A_{Sub} = \text{TransactionEvent} \quad \exists \text{bizStep. transporting}$

条件:系统中产生了一个交易事件,且该事件的

bizStep 取值是 railCarrying.

目标:判断该事件是否匹配订阅  $A_{Sub}$ .

订阅初始化的结果:  $S = \{ \{ \text{TransactionEvent}(e) \}, \{ \forall \text{bizStep. ( shipping airlifting landCarrying)}(e) \} \}$  匹配过程:

(1) 根据断言生成算法得到  $A = \{ \text{TransactionEvent}(e), \text{bizStep}(e, \text{railCarrying}) \}$

(2) 计算  $S = \{ A \setminus \{ A_{Sub}(e) \} \}$  的完全形式,结果如下:

$S = \{ \{ \text{TransactionEvent}(e), \text{bizStep}(e, \text{railCarrying}) \}, \{ \text{TransactionEvent}(e), \text{bizStep}(e, \text{railCarrying}), \text{shipping}(\text{railCarrying}), \text{airlifting}(\text{railCarrying}), \text{landCarrying}(\text{railCarrying}) \} \}$

(3) 判断  $S$  的一致性: $S$  中的第一个集合元素是不一致的,因为该集合元素包含  $\text{TransactionEvent}(e)$  和  $\text{TransactionEvent}(e)$ ;由于  $S$  中的第二个集合元素包含  $\text{landCarrying}(\text{railCarrying})$ ,显然也是不一致的,因而,  $S$  是不一致的,所以由定理可得事件  $e$  匹配订阅  $A_{Sub}$ .

### 5.3 复合事件与复合订阅的匹配

#### 5.3.1 复合订阅的初始化

每个复杂的复合订阅(包含两个以上的原子订阅)都可以看作由两个订阅成分通过复合订阅运算符复合而成,而每个订阅成分可能是一个原子订阅也可能是一个复合订阅.例如复合订阅  $C_{Sub} = (A_{Sub_1} \&A_{Sub_2}) \text{Before } A_{Sub_3}$  可以分解为  $C_{Sub} = C_{Sub_1} \text{Before } A_{Sub_3}$ ,  $C_{Sub_1} = A_{Sub_1} \&A_{Sub_2}$ . 我们可以按照这种方式将复合订阅分解为一系列的订阅,而每个订阅只包含两个订阅成分.这样处理以后,匹配过程只需考虑仅由两个订阅成分构成的复合订阅,而复杂的复合订阅的匹配是建立在其订阅成分被匹配的基础之上,从而使匹配算法得以简化.通过以上分析,我们只考虑仅包含两个订阅成分的复合订阅.对复合订阅的初始化包括如下两部分工作:

(1) 分解复合订阅  $C_{Sub}$ ,找出构成  $C_{Sub}$  的订阅成分,这些订阅成分构成一个集合  $C$ ,为  $C_{Sub}$  建立一个复合事件断言集  $CA$ ,并为每个订阅成分建立一个事件全序列列表  $eventList$ .  $CA$  用于记录订阅成分的匹配情况,  $eventList$  用于记录匹配该订阅成分的事件(事件按照发生时间的先后在  $eventList$  中进行排列),初始时  $CA = \{ \text{Sub} \mid \text{Sub} \in C \}$ ,  $eventList$  为空;

(2) 计算  $C_{Sub}$  的否定一般式(否定符号只出现在订阅成分的前面),并应用图 8 所示的一致性保持规则,计算  $S = \{ \{ C_{Sub} \} \}$  的完全形式.(由于复合订阅的匹

配中不涉及到事件的属性,因而省略了占位符  $e$ )

<p>&amp;- 规则                  条件: <math>(Sub_1 \&amp;Sub_2 \ A, Sub_1 \in A \ Sub_2 \in A)</math>                  操作: <math>A = A \setminus \{ Sub_1, Sub_2 \}</math></p> <p>  - 规则                  条件: <math>Sub_1 \mid Sub_2 \ A, Sub_1 \in A \ Sub_2 \in A</math>                  操作: <math>A = A \setminus \{ Sub_1 \}, A = A \setminus \{ Sub_2 \}</math></p> <p>- 规则                  条件: <math>Sub_1 \ Sub_2 \ A, Sub_1 \in A \ Sub_2 \in A \ \text{int}(Sub_1, Sub_2) \in A</math>                  操作: <math>A = A \setminus \{ Sub_1, Sub_2, \text{int}(Sub_1, Sub_2) \}</math></p>
--

图 8 复合订阅展开的一致性保持规则

#### 5.3.2 匹配算法

当一个事件匹配  $C$  中的某个订阅成分  $Sub$ ,或是到达  $C_{Sub}$  中某个否定订阅成分的检测时间点时进行如下匹配过程:

- (1) 对  $CA$  和对应的订阅成分的  $eventList$  进行更新;
- (2) 计算  $S = \{ CA \setminus \{ C_{Sub} \} \}$  的完全形式,即  $S = \{ A_i \mid \forall A_k \ S, A_i = CA \setminus A_k \}$ ;
- (3) 判断  $S$  的完全形式是否一致,从而得到是否产生了匹配  $C_{Sub}$  的复合事件.

当且仅当满足下面两种情况之一时,  $S$  中的集合元素  $A_i$  是不一致的:

$$\{ Sub, \ Sub \} \subseteq A_i$$

$$\text{int}(Sub_1, Sub_2) \ A_i \ (\text{isConflict}(\text{int}(Sub_1, Sub_2), \text{eventList}) = \text{true}),$$

其中  $\text{isConflict}$  用于判断  $\text{int}(Sub_1, Sub_2)$  与  $\text{eventList}$  记录的时序关系是否冲突.

复合事件与复合订阅的匹配算法如图 9 所示.事件的消费语义采用 Chronicle<sup>[15]</sup>.对于没能消费掉的事件可以采用一定的超时策略进行删除,由于篇幅所限,就不再赘述.

#### 5.3.3 匹配示例

下面仍以引言中的订阅需求来举例说明复合事件与复合订阅的匹配过程:

用户的订阅需求:如果在出货前货物被解包装则向管理人员发出通知.

用户的订阅表达为:  $C_{Sub} = A_{Sub_1} \text{Before } A_{Sub_2}$ ,其中  $A_{Sub_1}$  规约了货物的解包装,  $A_{Sub_2}$  规约了出货,具体的表达式见图 4.

条件:系统中产生了匹配  $A_{Sub_1}$  和  $A_{Sub_2}$  的原子事件  $e_1$  和  $e_2$ ,并且  $e_1$  在  $e_2$  之前发生(即  $e_1.\text{end} < e_2.\text{start}$ ).

目标:判断  $[e_1 e_2]$  是否匹配  $C_{Sub}$

订阅初始化的结果:

$C = \{ A_{Sub_1}, A_{Sub_2} \}$ ,  $CA = \{ A_{Sub_1}, A_{Sub_2} \}$ ,  $A_{Sub_1}$  对应的  $eventList_1$  和  $A_{Sub_2}$  对应的  $eventList_2$  均为  $\phi$ ,  $S = \{ \{ C_{Sub} \} \} = \{ \{ A_{Sub_1} \}, \{ A_{Sub_2} \}, \{ \text{intUnBefore } A_{Sub_1}, A_{Sub_2} \} \}$



个复合订阅中只包含两个原子订阅。

图 10(a) 显示了在不同复合订阅数量下的匹配时间,其中订阅数量从 500 增加到 10000。从图中可以看出,本系统的匹配时间与订阅数量基本上呈线性关系。

图 10(b) 显示了在不同复合订阅数量下的空间占用情况,其中空间占用既包括保存所有订阅初始化的结果所占用的空间也包括每次匹配时需要的额外空间。空间的单位以概念或属性的个数来度量。从图中可以看出,空间占用与订阅数量基本呈线性关系。

## 7 结论

本文在分析 RFID 信息服务对发布/订阅系统要求的基础上,将描述逻辑中基于个体的推理方法和时间本体引入 RFID 信息服务的发布/订阅系统中,解决了原子事件与原子订阅的语义匹配问题以及与时序相关的复合事件与复合订阅的匹配问题。本文提出的算法目前已经应用于 863 项目“RFID 公共服务体系架构设计及应用服务关键技术研究”以及“区域 RFID 信息公共服务平台关键技术研究”中的 RFID 信息服务中,由于本文提出的订阅语言还未能更多的实际应用中表达复杂的业务逻辑,所以我们后续的研究重点是在分析其他关于 RFID 信息服务发布/订阅系统的基础上,在兼顾匹配效率的情况下来提高订阅语言的表达能力,并在实践中进行检验。

## 参考文献:

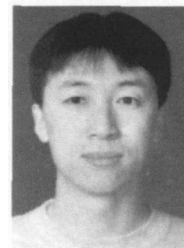
- [1] EPCglobal. The EPCglobal Architecture Framework[DB/OL]. <http://www.epcglobalinc.org/standards/architecture/architecture.1.2-framework-20070910.pdf>, Sep 2007.
- [2] EPCglobal. The Application Level Events (ALE) Specification, Version 1.1[DB/OL]. <http://www.epcglobalinc.org/standards/ale/ale.1.1-standard-core-20080227.pdf>, 2008-02.
- [3] EPCglobal. EPC Information Services (EPCIS) Version 1.0 Specification[DB/OL]. <http://www.epcglobalinc.org/standards/epcis/epcis.1.0-standard-20070412.pdf>, 2007-04.
- [4] EPCglobal. EPC Information Services (EPCIS) Version 1.0.1 Specification[DB/OL]. <http://www.epcglobalinc.org/standards/epcis/epcis.1.0.1-standard-20070921.pdf>, 2007-09.
- [5] Petrovic M, Burcea I, Jacobsen HA. S-ToPSS: Semantic Toronto publish/subscribe system[A]. Proceedings of the 29th VLDB Conference on very Large Data Bases[C]. San Francisco: Morgan Kaufmann Publishers, 2003. 1101-1104.
- [6] Cilia M, Bornhövd C, Buchmann AP. CREAM: An infrastructure for distributed, heterogeneous event-based applications[A]. Proceedings of the International Conference on Cooperative Information Systems[C]. London: Springer-Verlag, 2003. 482-502.
- [7] 汪锦岭, 金蓓弘, 李京, 邵丹华. 基于本体的发布/订阅系统的数据模型和匹配算法[J]. 软件学报, 2005, 16(9): 1625-1635.
- [8] Wang JL, Jin BH, Li J, Shao DH. Data model and matching algorithm in an ontology-based publish/subscribe system[J]. Journal of Software, 2005, 16(9): 1625-1635. (in Chinese)
- [9] Carzaniga A, Rosenblum DS, Wolf AL. Design and evaluation of a wide-area event notification service[J]. ACM Transactions on Computer Systems, 2001, 19(3): 332-383.
- [10] Fiege L, Gärtner FC, Kasten O, Zeidler A. Supporting mobility in content-based publish/subscribe middleware[A]. Proceedings of the 4th ACM/IFIP/USENIX International Middleware Conference[C]. Rio de Janeiro: Springer-Verlag, 2003. 103-122.
- [11] Li GL, Jacobsen A. Composite subscriptions in content-based publish/subscribe systems[A]. Proceedings of the 6th ACM/IFIP/USENIX International Middleware Conference[C]. Grenoble: Springer-Verlag, 2005. 249-269.
- [12] Samani MM, Sloman M. GEM: A generalized event monitoring language for distributed systems[J]. Distributed Systems Engineering, 1997, 4(2): 96-108.
- [13] Pietzuch P, Shand B, Bacon J. A framework for event composition in distributed systems[A]. Proceedings of the 4th ACM/IFIP/USENIX International Middleware Conference[C]. Rio de Janeiro: Springer-Verlag, 2003. 62-82.
- [14] Baade F, Calvanese D, et al. The Description Logic Handbook: Theory, Implementation, and Applications[M]. Cambridge University Press, 2003. 47-100.
- [15] Hobbs JR, Pan F. An ontology of time for the semantic web[J]. ACM Transactions on Asian Language Information Processing, 2004, 3(1): 66-85.
- [16] Chakravarthy S, Krishnaprasad V, Anwar E, Kim S-K. Anatomy of a composite event detector[R]. Technical Report UF-CIS-TR-93-039, University of Florida, 1993.

## 作者简介:



刘殿兴 男, 1980 年出生, 辽宁抚顺人, 北京大学博士研究生, 主要研究方向为软件工程、发布/订阅系统, RFID 相关技术。

E-mail: ldxhenty@amail.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、工作流技术和 RFID 相关技术。