

# 基于本体的发布/订阅系统的数据模型和匹配算法\*

汪锦岭<sup>1,2+</sup>, 金蓓弘<sup>1</sup>, 李京<sup>1</sup>, 邵丹华<sup>1</sup>

<sup>1</sup>(中国科学院 软件研究所 软件工程技术中心,北京 100080)

<sup>2</sup>(中国科学院 研究生院,北京 100049)

## Data Model and Matching Algorithm in an Ontology-Based Publish/Subscribe System

WANG Jin-Ling<sup>1,2+</sup>, JIN Bei-Hong<sup>1</sup>, LI Jing<sup>1</sup>, SHAO Dan-Hua<sup>1</sup>

<sup>1</sup>(Technology Center of Software Engineering, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

<sup>2</sup>(Graduate School, The Chinese Academy of Sciences, Beijing 100049, China)

+ Corresponding author: Phn: +86-10-62630989 ext 203, E-mail: jlwang@otcaix.iscas.ac.cn, <http://www.ios.ac.cn>

Received 2004-02-24; Accepted 2004-06-10

Wang JL, Jin BH, Li J, Shao DH. Data model and matching algorithm in an ontology-based publish/subscribe system. *Journal of Software*, 2005,16(9):1625–1635. DOI: 10.1360/jos161625

**Abstract:** The existing publish/subscribe systems can't match events with subscriptions based on the semantic of events, and they cannot support events with complex structure (such as graph structure). The Semantic Web technologies are introduced into the publish/subscribe system and an ontology-based publish/subscribe system is proposed. In this system, the concept model of events is represented as ontologies, the events are represented as RDF graphs, and the subscriptions are represented as graph patterns. The system can overcome the disadvantages of the existing publish/subscribe systems. Experimental results show that it has high matching efficiency.

**Key words:** publish/subscribe; ontology; RDF; matching algorithm

**摘要:** 现有的发布/订阅系统不能根据事件的语义来进行事件与订阅的匹配,且不能支持具有复杂结构(如图状结构)的事件.将语义 Web 技术引入发布/订阅系统中,提出一种基于本体的发布/订阅系统.该系统采用本体来表示事件的概念模型,采用 RDF 图来表示事件,采用图模式来表示订阅条件.它能较好地解决现有的发布/订阅系统的上述问题.实验结果表明,该系统具有较高的订阅匹配效率.

**关键词:** 发布/订阅;本体;RDF;匹配算法

中图法分类号: TP393 文献标识码: A

---

\* Supported by the National Natural Science Foundation of China under Grant No.60173023 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2001AA113010 (国家高技术研究发展计划(863)); the National Grand Fundamental Research 973 Program of China under Grant No.2002CB312005 (国家重点基础研究发展规划(973))

作者简介: 汪锦岭(1974 - ),男,安徽庐江人,博士,主要研究领域为分布式计算,中间件技术;金蓓弘(1967 - ),女,博士,副研究员,CCF高级会员,主要研究领域为分布式计算,软件工程技术;李京(1966 - ),男,博士,研究员,博士生导师,主要研究领域为软件体系结构,组合软件技术;邵丹华(1971 - ),男,工程师,主要研究领域为中间件技术.

Internet 的快速普及,极大地改变了分布式系统的规模.基于 Internet 的分布式系统可以包含上万个分布于世界各地的参与者,在系统的整个生命周期中,这些参与者的位置和行为都可能会改变.这就需要有一种更为灵活的通信模型和交互机制,以反映系统的高度动态和松散耦合的特性.而发布/订阅(publish/subscribe,简称 pub/sub)范型(paradigm)能够很好地满足上述需求,因此受到越来越多的关注.在 pub/sub 方式下,信息提供者以“事件”的形式,将信息发布到 pub/sub 系统中;信息使用者定义一个订阅条件,表示对系统中的某一特定种类的事件感兴趣;而 pub/sub 系统保证将所发布的事件及时、可靠地传送到所有感兴趣的订阅者.pub/sub 交互风格的优点在于,信息的生产者和消费者在时间、空间和控制流这 3 个方面都被完全解耦<sup>[1]</sup>,因而能够很好地满足大规模、高度动态的分布式系统的需要.

在不同的分布式系统中,各参与者之间所交换的信息的格式和含义往往是不一样的.为了使 pub/sub 系统成为一种通用的分布式计算基础设施,能够支持多种应用系统,它就必须具有很强的表达能力,能够:

- 支持各种具有不同语义和格式的事件;
- 提供一个强大的订阅语言,使得信息的接收者能够方便地表达它对哪些事件感兴趣.

虽然在 pub/sub 系统领域已经有了很多研究,但是现有的 pub/sub 系统在表达能力方面尚存在如下不足:

(1) 目前已有的系统基本上都是根据事件的结构信息来进行事件与订阅的匹配,而缺乏对事件本身语义的理解.如果 pub/sub 系统能够从语义上进行事件与订阅的匹配,必将会大大提高匹配的准确度,同时也会使订阅者能够更加方便地定义其订阅条件.

(2) 目前已有的系统只能支持关系数据结构(如“属性=值”对)和树形数据结构(如 XML),而某些应用场景可能要求事件具有更为复杂的结构,如图状数据结构.此外,不同的发布者所发布的事件可能具有不同的结构,例如有的为 XML,有的为图状结构.因此,需要有一种统一的机制,能够同时处理不同格式的事件.

为了解决上述问题,我们将语义 Web 技术引入 pub/sub 系统中,提出一种基于本体的 pub/sub 系统(ontology-based publish/subscribe system,简称 OPS).该系统将事件中所涉及到的各种概念整合到一起,建立起统一的概念模型,从语义和结构两个方面来解决订阅匹配问题.同时,在 OPS 系统内部,每个事件被表示为一个 RDF(resource description framework)<sup>[2]</sup>图,它是一种带标记的有向图(directed labeled graph,简称 DLG).Tim Berners-Lee 指出<sup>[3]</sup>,任何格式的数据都可以被表示为 DLG,进而可以被表示为 RDF.因此,本系统可以同时支持各种格式的事件,包括关系数据结构、树型数据结构以及图状数据结构等,其表达能力大大高于已有的 pub/sub 系统.当事件被发布时,系统首先将其转换成 RDF 格式,然后再对其进行进一步的处理.而对于事件接收者而言,所有的事件都是 RDF 格式的.

本文第 1 节介绍相关的研究工作.第 2 节介绍 OPS 系统的数据模型.第 3 节介绍其订阅语言.第 4 节介绍匹配算法.第 5 节对系统进行模拟实验,并对其效率作出评价.第 6 节对本文进行总结.

## 1 相关工作

Pub/sub 系统可以分为基于主题和基于内容两大类.在基于主题的系统(如 IBM MQSeries<sup>[4]</sup>),事件被划分为若干个固定的主题,每个事件都只能属于其中的某一个主题.发布者在发布事件时,必须指明该事件属于哪一个主题;订阅者则对某一主题下的所有事件进行订阅.在基于内容的系统中,订阅者根据事件的内部结构,设置一个订阅条件,所有满足该条件的事件都将被传送给该订阅者.与基于主题的 pub/sub 系统相比,基于内容的 pub/sub 系统提供了更强的表达能力,使订阅者能够以更细的粒度来定义其订阅条件.

目前的基于内容的 pub/sub 系统又可分为两类,一类是基于 Map 的,另一类是基于 XML 的.在基于 Map 的 pub/sub 系统中,事件的内容为多个“属性=值”的集合,订阅条件一般是各个属性之上的简单断言的连接,通常称为平面模式(flat pattern).较有影响的原型系统包括 SIENA<sup>[5]</sup>,Gryphon<sup>[6]</sup>,JEDI<sup>[7]</sup>,Keryx<sup>[8]</sup>,Elvin<sup>[9]</sup>等.在基于 XML 的 pub/sub 系统中,每个事件是一个 XML 文档,订阅语言通常是 XPath 或其变体,其中既包括对 XML 文档结构的约束,又包括对某些元素和属性的约束,通常人们称其为树模式(tree pattern).目前已有的原型系统包括 XFilter<sup>[10]</sup>,XTrie<sup>[11]</sup>,WebFilter<sup>[12]</sup>等.

与已有的 pub/sub 系统相比,OPS 系统的不同之处在于:

1) 已有的系统基本上都不能利用事件的语义信息进行订阅匹配,而 OPS 系统通过采用语义 Web 技术,能够根据事件的语义来提供更为准确的匹配。

2) OPS 系统中的事件被表示为一个有向图,从而该系统能够支持比 Map 和 XML 格式更为复杂的事件。

3) 在 OPS 系统中,订阅条件是一种图模式(graph pattern),其表达能力超过了平面模式和树模式。

近年来,也有一些对 pub/sub 系统事件语义的研究,如 S-ToPSS<sup>[13]</sup>和 CREAM<sup>[14]</sup>等。与这些工作相比,OPS 系统的不同之处在于,事件被表示为 RDF 图,订阅条件被表示为图模式,因而具有更强的表达能力。

在信息检索(information retrieval)领域的信息过滤(information filter)系统和选择性数据发送(selective dissemination of information)系统中,也有很多对订阅和匹配问题的研究<sup>[15-17]</sup>。在这些系统中,事件通常是非结构化的文本,订阅条件通常为关键词的集合。与它们相比,OPS 系统能够利用事件的语义信息和结构信息进行匹配,从而能够比基于关键词的方法提供更为精确的过滤。同时,信息检索领域的这些系统的关注重点通常是匹配的有效性,即如何找到最相关的信息,而很少关注匹配的效率。

在数据库领域的触发器和规则处理方面,也有一些对订阅和匹配问题的研究<sup>[18,19]</sup>。在这些系统中,订阅条件一般为平面模式,与基于 Map 的 pub/sub 系统类似。

在国内,对分布式系统消息通信机制的研究主要集中于消息队列(MQ)方式<sup>[20]</sup>,而较少有对 pub/sub 技术的研究。

## 2 数据模型

在 OPS 系统中,我们采用语义 Web 中的 RDF 语言和 DAML+OIL<sup>[21]</sup>语言来描述系统的数据模型。OPS 系统的数据模型包括如下两个方面:

1) 事件模型,它定义了事件内部数据的组织方式,采用 RDF 语言来描述;

2) 概念模型,它定义了所有事件中涉及到的各种概念及其相互关系,采用 DAML+OIL 语言来描述。

由于 RDF 和 DAML+OIL 主要是用来描述 WWW 上的信息的,因而它们使用 URI 作为各个实体的标识。而对于一个基于事件的系统而言,其关注的重点是正在传输中的事件,这些事件本身以及事件内部的各实体往往没有具体的 URI。因此,当我们采用 RDF 来表示事件的语义时,事件本身以及大多数事件内部实体都被表示为空白结点(blank node)。根据 RDF 规范,可以为空白结点赋一个以“\_”开始的 ID。

在下文中,为清晰起见,我们称图中的结点为“顶点”,称树中的结点为“结点”。

### 2.1 事件模型

在 OPS 系统内部,每个事件被表示为一个 RDF 图,称为事件图(event graph)。RDF 是一种按照(subject, property, object)三元组来表达事实的方式,每个三元组称为一个“语句(statement)”,其中 subject, property 为 URI, object 可以为 URI 或文本。RDF 数据可以用一种有向图来表示,图中的顶点表示各语句中出现的 subject 或 object,弧表示 property。每个弧的起点、终点以及弧本身构成了一个语句,其中弧的起点为语句中的 subject,弧的终点为语句中的 object。

例如,在一个网上拍卖系统中,设 Jinling Wang 要拍卖一个 IBM 公司生产的 Desktop PC,价格为 450 美元,其硬盘大小为 40G 且也由 IBM 公司生产,则相应的事件如图 1 所示。为简化起见,在图 1 中,我们省略了“daml:Thing”顶点、“rdf:Literal”顶点以及指向它们的弧。

RDF 规范中定义了一个属性“rdf:type”,其主语为某实体,宾语为该实体所属的类。在 RDF 图中,如果某顶点有一个“rdf:type”属性,则该顶点称为“带类型顶点(typed-vertex)”。

为了便于处理,在 OPS 系统中,我们对事件图作如下限制:

1. 图中有且仅有一个顶点称为“主顶点(home vertex)”。它描述整个事件的全局信息,例如事件的类型、发生时间等。我们规定该顶点的 ID 为“\_:H”。

2. 从主顶点到其他任何顶点都存在路径。

3. 每个顶点都为带类型顶点。用户可以为某顶点定义多个类,表示该实体同时属于多个类型。

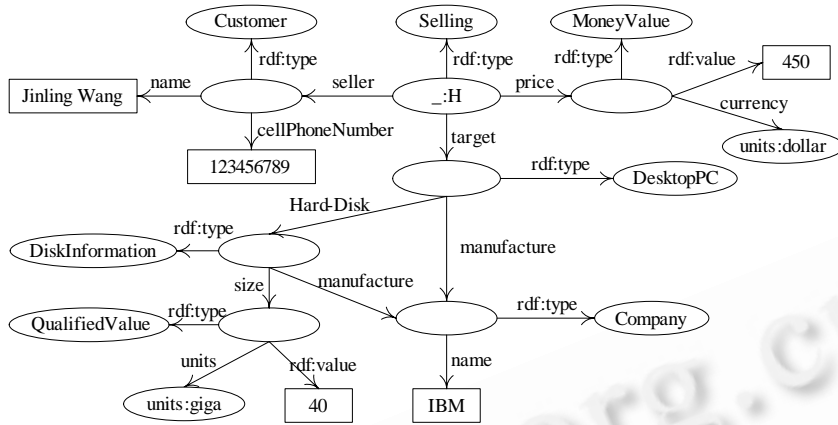


Fig.1 An example of event graph

图 1 事件图示例

## 2.2 概念模型

在 OPS 系统中,我们采用本体来表示事件的概念模型.一般认为,本体是一种对概念化结果的规范表示<sup>[22]</sup>.它描述了某一领域中的各种概念以及它们之间的关系和应满足的约束等.在语义 Web 领域中,目前影响较大的本体描述语言是 DAML+OIL.在 OPS 系统中,我们采用该语言来描述事件的概念模型.

事件的概念模型通常应由领域专家与计算机人员共同创建.目前,国际上也有一些组织致力于领域概念模型的标准工作,为某些特定领域提供通用的、统一的概念模型.

在 OPS 系统中,事件的概念模型主要由以下 3 部分组成:

1. 各 Class 及其层次关系描述.一个实体可以同时属于多个 Class.各 Class 之间可以存在多重继承,但是不允许出现循环的继承关系.例如,一个网上拍卖系统的一部分层次关系如图 2(a)所示.

如果订阅条件中有一个类为 A,而事件中某个实体的类为 B,且 A 为 B 的祖先,那么这两者可以匹配.例如,假设某人对所有价格小于 400 美元的计算机感兴趣,那么他可以根据“Computer”类来定义订阅条件,此后所有价格小于 400 美元的拍卖台式机或笔记本电脑的事件都会发送给他.而现有的 pub/sub 系统一般只能支持事件类型的层次结构,而不能支持事件内部实体的类层次结构.在这些系统中,如果某人对所有价格小于 400 美元的计算机感兴趣,那么他就必须针对“Computer”的每个子类分别设置订阅条件(即价格小于 400 美元的台式机、价格小于 400 美元的笔记本电脑),而这是非常麻烦的.

2. 各 Property 及其层次关系描述.各 Property 之间可以存在多重继承,但是不允许出现循环的继承关系.一个 Class 可以拥有多个 Property,一个 Property 也可以服务于多个 Class.例如,一个网上拍卖系统的一部分属性层次关系如图 2(b)所示.

如果订阅条件中有一个属性  $p_1$ ,而事件数据中有属性  $p_2$ ,且  $p_1$  为  $p_2$  的祖先,那么这两者可以匹配.例如,假设某人的订阅条件为 telephoneNumber="123456789",而某个事件中的数据为 cellPhoneNumber="123456789",那么该事件与该订阅条件是相匹配的.

3. 元语句(meta-statement).我们称一个三元组(SubjectClass,property,ObjectClass)为一个“元语句”.它表示对于一个给定的类(SubjectClass),允许有哪些属性(property),其属性的值分别属于哪个类(ObjectClass).

例如,一个网上拍卖系统可能包括如下一些元语句:

- (Selling, seller, Customer)
- (Selling, target, Product)
- (Product, manufacture, Company)
- (Customer, name, xsd:string)

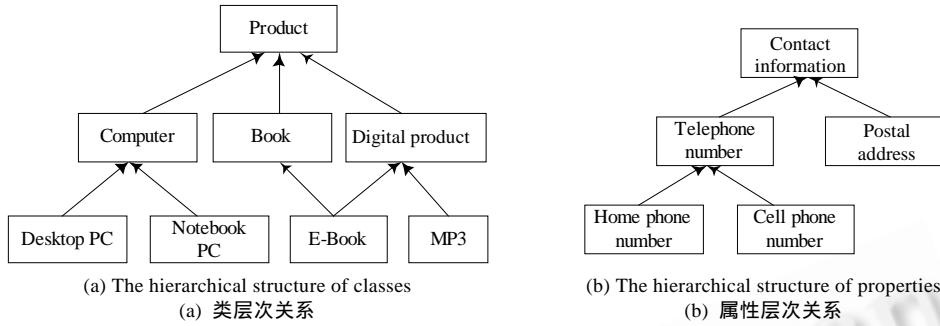


Fig.2 A part of classes and properties in an online auction system

图 2 网上拍卖系统的一部分类和属性

不同元语句之间也可以存在层次关系.对于两个元语句: $ms_1=(sc_1,p_1,oc_1)$ 和  $ms_2=(sc_2,p_2,oc_2)$ ,如果以下断言成立:

$$(sc_1 \text{ rdfs:subClassOf } sc_2) \wedge (p_1 \text{ rdfs:subPropertyOf } p_2) \wedge (oc_1 \text{ rdfs:subClassOf } oc_2),$$

则称  $ms_2$  是  $ms_1$  的“祖先(ancestor)”,记为  $ms_1 \sqsubseteq ms_2$ .其含义是,如果某语句(statement)满足  $ms_1$  的类型约束,那么它也一定满足  $ms_2$  的类型约束.

### 3 订阅语言

由于在 OPS 系统中事件被表示为 RDF 图,所以用户的订阅条件实际上就是一种建立在 RDF 图语法之上的图模式,其中规定了图的形状以及对某些顶点和弧的约束.我们在参考 SquishQL<sup>[23]</sup>,RQL<sup>[24]</sup>,RDQL<sup>[25]</sup>等 RDF 查询语言的基础上,设计了一种 OPS 系统的订阅语言.

在 OPS 系统中,用户的一个订阅条件由若干个“语句模式(statement pattern)”的“与”操作组成.每个语句模式描述事件图中的一个语句,其形式如下:

$$(\text{subject, object, meta-statement, [filter\_func(object)]}).$$

其中,subject 和 object 规定了一个语句中的 subject 和 object,它们可以是具体的值,也可以是变量,变量可以与任何具体的值相匹配.变量名以“?”作为前缀,如?1,?2 等.

语句模式中的 meta-statement 规定了一个语句应满足的类型约束,即令某语句模式中的 meta-statement 为  $(sc,sp,oc)$ ,若某语句(s,p,o)能与该语句模式匹配,则如下断言为真:

$$\begin{aligned} s & \text{ rdfs:type } sc \\ p & \text{ rdfs:subPropertyOf } sp \\ o & \text{ rdfs:type } oc \end{aligned}$$

当语句模式中 object 为变量且其类型为文本时,语句模式中可以有过滤函数 filter\_func(object),它是一个布尔表达式,用于进一步限制宾语变量的取值.过滤函数中允许的操作包括>, <, = 等关系运算和正则表达式运算等.

例如,在上述网上拍卖系统中,如果某人对所有拍卖计算机且价格小于 400 美元的事件感兴趣,则相应的订阅条件为

$$\begin{aligned} (& \_ : H, ?1, (\text{Selling, target, Computer})) \\ (& \_ : H, ?2, (\text{Selling, price, MoneyValue})) \\ (?2, & \text{units:dollar, (MoneyValue, currency, daml:Thing)}) \\ (?2, & ?3, (\text{MoneyValue, rdf:value, xsd:decimal}), ?3 < 400.00) \end{aligned}$$

每个订阅条件在系统内部用一个图来表示,称为订阅图(subscription graph).例如,上述订阅条件的订阅图如图 3 所示.

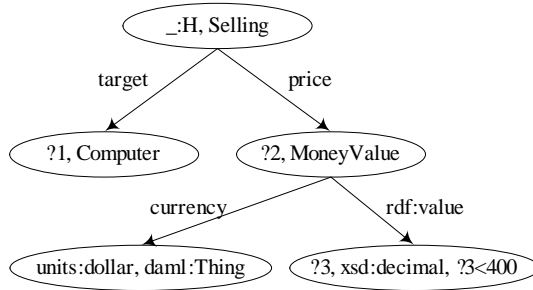


Fig. 3 An example of subscription graph

图 3 订阅图示例

在订阅图中,每个顶点上的标记为(id,type,[filter\_func(id)]),它对应于事件图中的一个顶点.其中id为变量名或事件图中的顶点的标记,type为id所对应的类.当对应的事件图顶点为文本顶点时,可以有一项filter\_func(id),表示id应满足的约束.每个弧上的标记为属性名,它和弧起点的type和弧终点的type一起,构成一个元语句.该元语句与弧起点的id、弧终点的id以及filter\_func(id)一起,构成了一个语句模式.

我们对用户的订阅条件作如下限制:

1. 在各语句模式中,至少有一个pattern的subject为“.\_:H”,即事件的主顶点.我们称订阅图中id=“.\_:H”的顶点为订阅图的“主顶点”.
2. 在订阅图中,从主顶点到任何其他顶点都存在路径.

#### 4 匹配算法

Pub/sub系统的匹配问题的本质在于,当到达一个事件以后,要能快速地找到所有与之匹配的订阅条件.从这一点上说,pub/sub系统与数据库系统相比,数据和查询(订阅)条件的角色正好颠倒过来了<sup>[26]</sup>.在数据库系统中,大量的数据被保存并建立了索引,以便当用户发起一个查询条件时,能够快速地找到所需要的数据.而在pub/sub系统中,大量的订阅条件被保存并建立索引,以便当到达一个事件(数据)时,能够快速地找到与之匹配的订阅条件.下面我们介绍本系统中采用的索引结构以及相应的匹配算法.

##### 4.1 索引结构

我们在用户所定义的元语句的基础上,根据类层次结构和属性层次结构,求出所有合法的元语句,将它们放到一个数组中,称为扩展元语句(extended meta-statement,简称EMS)数组.该数组是OPS系统的索引结构的基础,其中各元语句按字典顺序排序,以便于查找.

在EMS数组中,每一项包括两个链表:祖先链表(ancestor list)和待匹配项链表(waiting-pattern list).祖先链表中记录了此元语句的各祖先在数组中的序号,待匹配项链表中包括了与之相关的等待匹配的语句模式.初始时,各待匹配项链表中只包括各订阅条件中主语为“.\_:H”的语句模式.

例如,假设某系统中只有一个如图4(a)所示的订阅条件,则初始时EMS数组如图4(b)所示.

在图4(b)中,每一项后面的第1个链表为祖先链表(用实线表示),第2个链表为待匹配项链表(用虚线表示),nil表示空指针.为了书写简便,在图4以及后面的例子中,我们使用A,B,C等表示类名,使用 $p_1, p_2, p_3$ 等表示用户定义的属性名称,使用EMS( $i$ )表示EMS数组中第 $i$ 项的元语句.

##### 4.2 匹配过程和匹配树

当一个事件进入OPS系统以后,系统从该事件图的主顶点开始,按照广度优先的顺序,对图中的各弧进行遍历,以使得图中所有标记不等于“rdf:type”的弧都被遍历到,且仅被遍历一次.对于每个被遍历到的弧,系统对其生成一个或多个如下形式的三元组:

(subject, object, meta-statement),

我们称其为“带类型语句(typed-statement)”,其中 subject 是弧的起点的 ID,object 是弧的终点的 ID,meta-statement 是此弧所对应的语句的元语句.其生成规则为:令某语句为 $(s,p,o)$ ,所生成的元语句为 $(ts,tp,to)$ ,则  $ts$  为事件图中所指定的  $s$  的类, $p=tp,to$  为事件图中所指定的  $o$  的类.一个语句可能会对多个带类型语句.

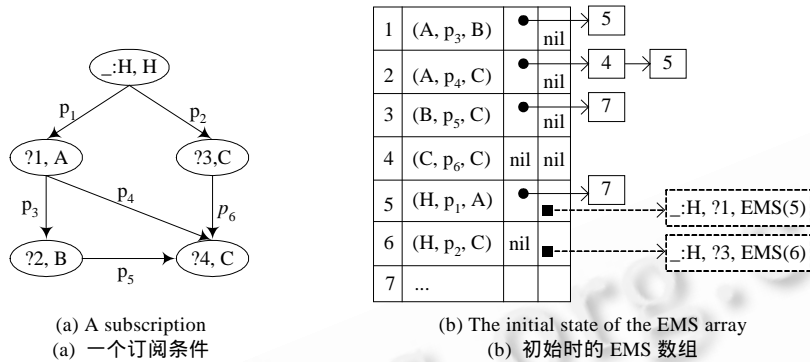


Fig.4 A subscription and the initial state of the EMS array

图 4 一个订阅条件及初始时的 EMS 数组

对于遍历事件内容时所生成的每个带类型语句,OPS 系统根据其中的元语句,找到 EMS 数组中的相应项,将其与待匹配项链表中的各语句模式进行匹配.然后,系统还应根据 EMS 数组中该项的祖先链表,对它的各祖先元语句中的待匹配项链表进行匹配.

令函数  $isVariable(k)$  表示断言“ $k$  为变量”.对于一个语句模式  $sp=(s_1,o_1,ms_1,filter\_func_1)$  和一个带类型语句  $ts=(s_2,o_2,ms_2)$ , $sp$  能与  $ts$  匹配的充要条件为

$$(s_1=s_2 \vee isVariable(s_1)) \wedge (o_1=o_2 \vee isVariable(o_1)) \wedge (ms_2 \sqsubseteq ms_1) \wedge filter\_func_1(o_2).$$

$sp$  与  $ts$  匹配的结果,是建立了两对顶点之间的映射: $s_1 \leftrightarrow s_2, o_1 \leftrightarrow o_2$ .

下面研究单个订阅图的匹配过程.匹配过程开始时,订阅图中以主顶点为起点的各语句模式已位于 EMS 数组中的待匹配项链表中.对于事件中的每个带类型语句,系统在 EMS 数组中的相应待匹配项链表中寻找与之匹配的语句模式.如果找到了,就形成了一个部分映射方案,其中记录了到目前为止的顶点映射.假设语句模式  $sp=(s_1,o_1,ms_1,filter\_func_1)$  和带类型语句  $ts=(s_2,o_2,ms_2)$  匹配成功,系统需要分如下两种情况进行处理:

1. 如果  $o_1$  不在从订阅图主顶点到  $s_1$  的当前路径上,那么系统在订阅图中找出以  $o_1$  为起点的其他各语句模式,将其中的变量代换成已有映射方案中的具体值,形成新的语句模式(称为派生语句模式),放入待匹配项链表中以供匹配.
2. 如果  $o_1$  已经存在于从订阅图主顶点到  $s_1$  的当前路径上了,那么就不再产生新的语句模式.

一个订阅图的匹配状态可以用一棵树表示,称为匹配树,如图 5 所示.订阅图的匹配过程可以表示为匹配树的创建和检验过程.

图 5(b)为图 5(a)所示的事件与图 4(a)所示的订阅条件进行匹配时所形成的匹配树.树中的圆圈结点表示部分映射方案,方框结点表示语句模式.根结点为圆圈结点,其中包含一个顶点对“\_:H” $\leftrightarrow$ “\_:H”(在图中,用“=”表示“ $\leftrightarrow$ ”).根结点的孩子为订阅图中所有主语为“\_:H”的各语句模式.一个圆圈结点下可以有多个方框结点,表示此次匹配所派生的各语句模式.只有当这些语句模式均匹配成功时,圆圈结点所代表的部分映射方案才是成功的.一个方框结点下也可以有多个圆圈结点,表示本语句模式有多种匹配方案.只要其中任何一个匹配方案成功,则此语句模式就匹配成功.由此可见,圆圈结点反映的是一种“与”关系,方框结点反映的是一种“或”关系,整个匹配树就是一棵与或树.

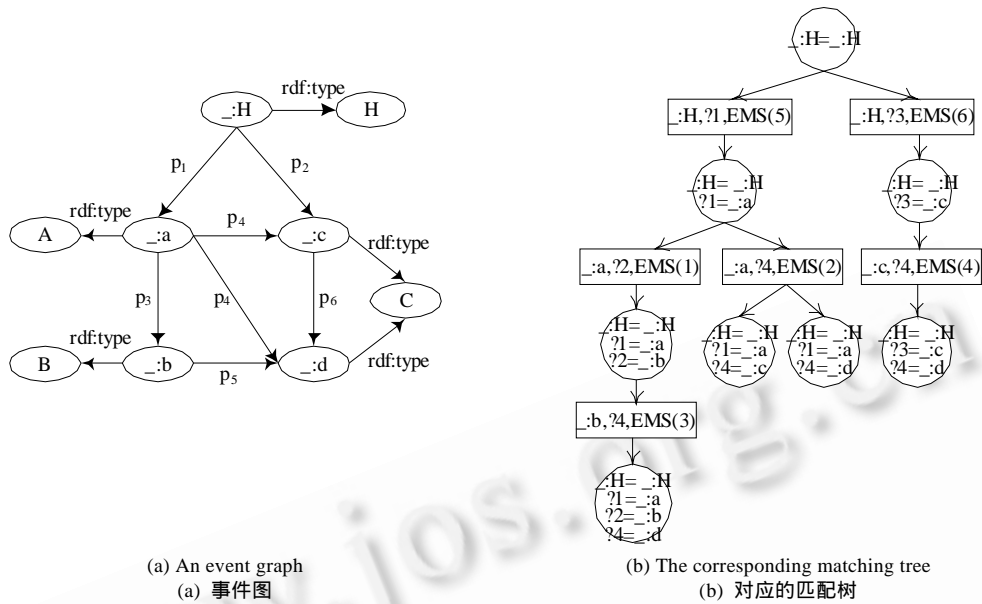


Fig.5 An example of matching tree

图 5 匹配树示例

### 4.3 匹配树的检验

当对事件内容遍历完成之后,系统就生成了各订阅图所对应的匹配树.然后,系统需要根据每个匹配树来判断相应的订阅图是否匹配成功,我们称这一过程为匹配树的检验(verification).OPS 系统采用两种方法来对匹配树进行检验:基于布尔表达式的检验方法(boolean expression based verification,简称 BEBV)和基于状态的部分检验方法(state based partial verification,简称 SBPV).

BEBV 方法对每个叶子结点赋一个布尔表达式,然后对整个匹配树进行表达式求值,以判断匹配是否成功.其规则如下:

1. 若叶子结点为圆圈结点,且其中的顶点映射为  $\{v_1^{SG} \leftrightarrow v_{x1}^{EG}, v_2^{SG} \leftrightarrow v_{x2}^{EG}, \dots, v_k^{SG} \leftrightarrow v_{xk}^{EG}\}$ , 则对应的表达式为  $(v_1^{SG} \leftrightarrow v_{x1}^{EG}) \wedge (v_2^{SG} \leftrightarrow v_{x2}^{EG}) \wedge \dots \wedge (v_k^{SG} \leftrightarrow v_{xk}^{EG})$ .
2. 若叶子结点为方框结点,则对应的表达式为 false.
3. 每个非叶子圆圈结点的表达式为其下各孩子结点的表达式的“与”运算,每个非叶子方框结点的表达式为其下各孩子结点的表达式的“或”运算.
4. 对于订阅图中的任意顶点  $v_i^{SG}$  和事件图中的两个顶点  $v_x^{EG}, v_y^{EG}$ , 表达式  $(v_i^{SG} \leftrightarrow v_x^{EG}) \wedge (v_i^{SG} \leftrightarrow v_y^{EG}) \wedge (v_x^{EG} \neq v_y^{EG})$  的值为 false, 即订阅图中的一个顶点不能同时映射到事件图中的两个不同顶点.
5. 对于事件图中的任意顶点  $v_i^{EG}$  和订阅图中的两个顶点  $v_x^{SG}, v_y^{SG}$ , 表达式  $(v_x^{SG} \leftrightarrow v_i^{EG}) \wedge (v_y^{SG} \leftrightarrow v_i^{EG}) \wedge (v_x^{SG} \neq v_y^{SG})$  的值为 false, 即事件图中的一个顶点不能同时映射到订阅图中的两个不同顶点.

根据上述规则对匹配树从底向上计算,如果根结点的值为 false,则表示匹配失败,否则,匹配成功.

然而,如果对每个匹配树都进行上述表达式求值,那么将是非常低效的.因此,我们又设计了一种匹配树检验方法——SBPV 方法,它能够以很低的成本检查出大部分匹配失败的匹配树,但不能最终确定一个订阅是否匹配成功.只有通过了 SBPV 检查后的匹配树,才利用 BEBV 方法来判断其是否匹配成功.

SBPV 方法对匹配树中的每个结点定义两种状态: checked 和 unchecked. checked 表示该结点已通过 SBPV 检验, unchecked 表示尚未完成检验.初始时,各结点的状态均为 unchecked.对于方框结点,只要它有一个孩子的状态变为 checked,其状态就变为 checked.对于圆圈结点,当其所有孩子的状态都为 checked 时,其状态才变为 checked.



每个圆圈结点中有一个 unCheckedChildren 字段,表示它所拥有的状态为 unchecked 的孩子总数.

在匹配树的创建过程中,当某个圆圈结点不能派生出新的语句模式时,系统将对匹配树进行回溯,以调整其上层结点的状态.其步骤如下:

- 1) 将当前结点的状态置为 checked.
- 2) 如果本结点为圆圈结点:
  - a) 若它是根结点,则回溯过程结束.
  - b) 若它不是根结点,检查父结点的状态是否为 checked.若是,回溯过程结束.若不是,则将父结点作为当前结点,递归调用本回溯过程.
- 3) 如果本结点为方框结点,则将父结点的 unCheckedChildren 减 1.如果该值变为 0,则置父结点为当前结点,递归调用本回溯过程,否则回溯过程结束.

当 SBPV 检查完成以后,如果匹配树的根结点的状态为 unchecked,则表示匹配失败.由于 SBPV 检查可以在匹配树的创建时同步完成,所以当系统对事件内容遍历完成之后,只需对那些根结点状态为 checked 的匹配树进行 BEBV 检查,从而可以大大提高处理的效率.

#### 4.4 算法分析

令  $TS$  表示平均每个事件中的带类型语句个数, $D$  表示 EMS 数组中平均每个元语句所能推导的元语句个数, $W$  表示平均每个待匹配项链表中的结点数, $SP$  表示各匹配树中平均每个圆圈结点下的语句模式个数,则由算法中的循环层次可知,本算法的时间复杂度为  $O(TS \times D \times W \times SP)$ .一般来说, $TS, D, SP$  的值都比较小,且不受订阅数量的影响,而  $W$  的值则可能会随着订阅数量的增加而增加.

设系统中的订阅数量为  $n$ ,EMS 数组中元语句数量为  $m$ ,平均每个匹配树中的语句模式数量为  $sp$ ,则  $W \approx (n \times sp) \div m$ .因此,本算法的时间复杂度与订阅数量呈线性关系.

在空间复杂度方面,本算法的空间占用主要包括以下两个方面:

- (1) 各订阅条件所对应的匹配树.匹配树中的结点个数由订阅图和事件图决定,与订阅条件数量无关.
- (2) EMS 数组(不包括其中的待匹配项链表,因为该链表中的结点同时也在匹配树中).其大小由事件的概念模型决定.

设匹配树的平均大小为  $mt$ ,EMS 数组中每一项的空间占用为  $e$ (不包括其中的待匹配项链表),则本算法的空间复杂度为  $O(n \times mt + m \times e)$ .因此,总的来说,本算法的空间复杂度与订阅数量呈线性关系.

## 5 实验

我们对 OPS 原型系统进行了模拟实验,以验证系统的正确性并对其性能进行评价.原型系统采用 Java 语言开发,实验所使用的机器是一台 CPU 为 Intel Pentium IV 1.6GHz,内存为 512M 的普通笔记本电脑,所使用的操作系统为 Windows 2000 Server,Java 运行环境为 JDK 1.4.1.

在模拟实验中,我们假设事件概念模型中有 10 个类和 10 个属性,每个类有两个属性,类之间和属性之间都不存在继承关系.

假设每个事件图中有 50 个空白顶点,它们之间有 55 个弧.每个订阅图中有 10 个顶点和 11 个弧.订阅图中的主顶点的 id 为“\_:H”,其他顶点的 id 均为变量.每个顶点的 type 都是随机产生的,各顶点中均不含有过滤函数.

图 6(a)显示了本系统在不同订阅数量下的匹配时间,其中订阅数量从 500 增加到 10 000.在这组实验中,平均匹配成功率为 3%.从图中我们可以看出,本系统的匹配时间与订阅数量基本上呈线性关系.当订阅数量为 10 000 时,匹配时间仅为 1.2s.

图 6(b)显示了本系统在不同订阅数量下的空间占用情况,其中  $Y$  轴表示平均每个匹配树中的结点数.OPS 系统中所需的空间主要用于创建各订阅所对应的匹配树.从图中我们可以看出,匹配树中的平均结点个数维持在 5 个左右,不随订阅数量的变化而变化.所以,本算法的空间复杂度与订阅数量呈线性关系.

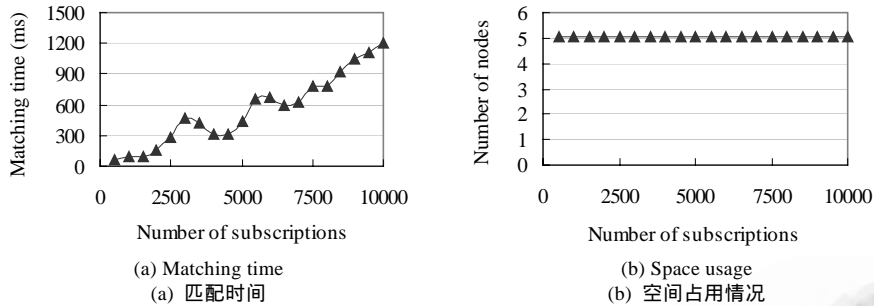


Fig. 6 The experimental result

图 6 实验结果

## 6 结 论

我们将语义 Web 技术与发布/订阅技术结合起来,提出了一种新型的发布/订阅系统.该系统能够利用事件的语义知识,提供更为准确的匹配,同时能够支持具有复杂数据结构(如图状结构)的事件.

在本系统中,事件概念模型是用 DAML+OIL 语言来描述的.该语言具有较强的描述能力,能够对类和属性进行多种语义约束,而本系统只利用了其描述能力的一小部分.我们下一步的工作将是进一步改进本系统的概念模型,使其能够包含更为丰富的语义知识,以进一步提高订阅匹配的准确度和效率.

### References:

- [1] Eugster PT, Felber PA, Guerraoui R, Kermarrec AM. The many faces of publish/subscribe. *ACM Computing Surveys*, 2003,35(2): 114-131.
- [2] Lassila O, Swick RR. Resource description framework (RDF) model and syntax specification. 1999. <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>
- [3] Berners-Lee T. Using XML for data. 2001. <http://www.w3.org/DesignIssues/XML-Semantics.html>
- [4] IBM. Internet Application Development with MQSeries and Java. Palos Verdes: Vervante Corporate Publishing, 1997.
- [5] Carzaniga A, Rosenblum DS, Wolf AL. Design and evaluation of a wide-area event notification service. *ACM Trans. on Computer Systems*, 2001,19(3):332-383
- [6] Aguilera MK, Strom RE, Sturman DC, Astley M, Chandra TD. Matching events in a content-based subscription system. In: *Proc. of the 18th ACM Symp. on Principles of Distributed Computing*. New York: ACM Press, 1999. 53-61.
- [7] Cugola G, Nitto ED, Fuggetta A. The JEDI event-based infrastructure and its application to the development of the OPSS WFMS. *IEEE Trans. on Software Engineering*, 2001,27(9):827-850.
- [8] Wray M, Hawkes R. Distributed virtual environments and VRML: An event-based architecture. In: *Proc. of the 7th Int'l World Wide Web Conf. (WWW7)*. Amsterdam: Elsevier Science Publishers, 1998. 43-51.
- [9] Fitzpatrick G, Kaplan S, Mansfield T, David A, Segall B. Supporting public availability and accessibility with Elvin: Experiences and reflections. *Computer Supported Cooperative Work*, 2002,11(3):447-474.
- [10] Altinel M, Franklin MJ. Efficient filtering of XML documents for selective dissemination of information. In: *Proc. of the 26th Int'l Conf. on Very Large Data Bases*. San Francisco: Morgan Kaufmann Publishers, 2000. 53-64.
- [11] Chan CY, Felber P, Garofalakis M, Rastogi R. Efficient filtering of XML documents with XPath expressions. *The VLDB Journal*, 2002,11(4):354-379.
- [12] Pereira J, Fabre F, Llirbat F, Jacobsen HA, Shasha D. WebFilter: A high throughput XML-based publish and subscribe system. In: *Proc. of the 27th Int'l Conf. on Very Large Data Bases*. San Francisco: Morgan Kaufmann Publishers, 2001. 723-724.
- [13] Petrovic M, Burcea I, Jacobsen HA. S-ToPSS: Semantic toronto publish/subscribe system. In: *Proc. of the 29th Int'l Conf. on Very Large Data Bases*. San Francisco: Morgan Kaufmann Publishers, 2003. 1101-1104.

- [14] Cilia M, Bornhoevd C, Buchmann AP. CREAM: An infrastructure for distributed, heterogeneous event-based applications. In: Proc. of the Int'l Conf. on Cooperative Information Systems (CoopIS). London: Springer-Verlag, 2003. 482–502.
- [15] Yan TW, Garcia-Molina H. The SIFT information dissemination system. ACM Trans. on Database Systems, 1999,24(4):529–565.
- [16] Cetintemel U, Franklin M, Giles CL. Self-Adaptive user profiles for large scale data delivery. In: Proc. of the IEEE Int'l Conf. of Database Engineering. Washington: IEEE Computer Society Press, 2000. 622–636.
- [17] Chinenyanga TT, Kushmerick N. Expressive retrieval from XML documents. In: Proc. of the 24th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. New York: ACM Press, 2001. 163–171.
- [18] Hanson EN, Carnes C, Huang L, Konyala M, Noronha L, Parthasarathy S, Park JB, Vernon A. Scalable trigger processing. In: Proc. of the 15th IEEE Int'l Conf. of Database Engineering. Washington: IEEE Computer Society Press, 1999. 266–275.
- [19] Hanson EN, Chaabouni M, Kim CH, Wang YW. A predicate matching algorithm for database rule systems. In: Proc. of the ACM SIGMOD. New York: ACM Press, 1990. 271–280.
- [20] Zhou ZH, Huang T, Li J. Design and implementation of message middleware manger. Journal of Computer Research and Development, 2002,39(3):318–323 (in Chinese with English abstract).
- [21] Harmelen FV, Patel-Schneider PF, Horrocks I. Reference description of the DAML+OIL (March 2001) ontology markup language. 2001. <http://www.daml.org/2001/03/reference>
- [22] Gruber TR. A translation approach to portable ontologies. Knowledge Acquisition, 1993,5(2):199–220.
- [23] Miller L, Seaborne A, Reggiori A. Three implementations of SquishQL, a simple RDF query language. In: Proc. of the 1st Int'l Semantic Web Conf. London: Springer-Verlag, 2002. 423–435.
- [24] Karvounarakis G, Alexaki S, Christophides V, Plexousakis D, Scholl M. RQL: A declarative query language for RDF. In: Proc. of the 11th Int'l World Wide Web Conf. New York: ACM Press, 2002. 592–603.
- [25] HP Labs. RDQL: RDF data query language. 2001. <http://www.hpl.hp.com/semweb/rdql.htm>
- [26] Yan TW, Garcia-Molina H. Index structures for selective dissemination of information under boolean model. ACM Trans. on Database Systems, 1994,19(2):332–364.

#### 附中文参考文献:

- [20] 周泽华,黄涛,李京.消息中间件管理器的设计和实现.计算机研究与发展,2002,39(3):318–323.