第 30 卷 第 1 期
2007 年 1 月
计 算 机 学 报
CHINESE JOURNAL OF COMPUTERS
Vol. 30 No. 1
Jan. 2007

一种基于 XML 的复合事件模型及其订阅语言

徐 罡 马建刚 黄 涛
（中国科学院软件研究所软件工程技术中心 北京 100080）

摘 要 基于内容的 Pub/Sub 系统能够为大范围软件集成提供松耦合的、可扩展的集成能力，越来越引起人们的重视. 现存的方法主要集中在对单一事件的匹配，缺乏对复合事件模型及其订阅语言的研究. 针对此问题，文章给出一种基于 XML 事件的复合事件模型及其订阅语言，该模型包括事件时序逻辑模型和事件复合模式，并扩展了 XML-QL 语言得到复合事件订阅语言 EXML-QL 支持事件复合模型和模式；最后，依据该订阅语言的特点构造复合事件匹配算法，给出该算法的分析和验证.

关键词 Pub/Sub 系统；软件集成；复合事件；订阅语言；XML
中图法分类号 TP311

A XML-Based Composite Event Model and Its Subscription Language

XU Gang MA Jiang-Gang HUANG Tao
(*Technology Center of Software Engineering, Institute of Software, Chinese Academy of Sciences, Beijing* 100080)

Abstract Content-based Pub/Sub system is gaining increasing popularity due to its loose coupling and scalable integration capability in the aspects of building large-scale software integrations. Unlike in the traditional address-based unicast or multicast, its core problem is how to match events by predicates on the contents of events. Currently, mostly existing approaches mainly limit at the "one-to-one" event matching and its matching predicates are composed of the conjunction and disjunction of non-semantic constraints. In fact, the composite event matching is required in many scenarios, typically including "one-to-many" or "many-to-one" matching. This kind of event matching enables application components to express more interest about the occurrence of complex patterns of events. A XML-based composite event model that consists of the temporal logical model and the event composite pattern is presented in the paper. The temporal logical model describes the temporal relations of events in distributed environments; the event composite pattern describes the different triggering and transferring patterns appearing the different application scenarios. Based on the composite event model, a subscription language called EXML-QL is introduced by adding the block-variables and the temporal logical operators in XML-QL language to support XML-based composite event patterns. Finally, according to the peculiarities of the subscription language, a composite event matching algorithm is presented and analyzed.

Keywords Pub/Sub systems; software integration; composite events; subscription language; XML

收稿日期：2005-05-31；修改稿收到日期：2006-06-06. 本课题得到国家"八六三"高技术研究发展计划项目基金（2001AA113010，2002AA413610，2003AA413010，2003AA115440）和国家"九七三"重点基础研究发展规划项目基金（2002CB312005）资助. **徐 罡**，男，1973 年生，博士，主要研究方向为软件工程、分布式计算、软件集成. E-mail：xugang@otcaix.iscas.ac.cn. **马建刚**，男，1977 年生，博士研究生，主要研究方向为分布式计算、中间件. **黄 涛**，男，1965 年生，博士，研究员，博士生导师，主要研究领域为软件工程、分布式计算.

1 引 言

建立大规模的软件集成需要中间件服务提供松耦合的、可扩展的集成能力，基于内容的 Pub/Sub 系统正是这样一种集成中间件. 利用 Pub/Sub 系统来集成分布软件系统只需要在 Pub/Sub 系统上订阅所需的事件，增加或减少被集成的软件系统对整个集成系统不会产生影响；其次，采用 XML 作为事件的表达形式，只需要相互间集成的系统能够处理 XML 文档，而对系统的构成没有额外的要求，可以说 Pub/Sub 系统具有很好的集成可扩展性.

与目前典型的集成中间件和集成方法相比，如远程方法调用、消息中间件、元组空间、传统的事件系统，在集成柔性和表达能力方面存在显著优势. 在集成柔性方面，基于内容的 Pub/Sub 系统能够根据事件的内容选择不同的转发路径，而消息中间件一般在消息代理中使用别名机制实现消息路由，别名本身仅仅是 IP 地址的一种表示方法，传统的事件系统使用主题或渠道来分类事件，事件发布者给每个事件标记特定渠道或主题名称，事件订阅者接受渠道中的所有事件或特定主题事件；在集成表达能力方面，基于内容的 Pub/Sub 系统具有更小粒度的订阅表达能力[1].

尽管基于内容的 Pub/Sub 系统具有很多优点，但是目前的研究只限于对单一事件模式的支持，即“一对一”的转发模式，缺乏对事件复合模式的研究，并且事件多采用信息集合的表述形式，而非 XML 事件. 复合事件较单一事件具有更多的事件表达能力. 例如，在企业应用集成中，企业中的客户关系管理系统与财务系统和销售系统需要实现数据集成，同时销售系统又与下游的库存管理系统和生产计划系统集成，如图 1 所示. 其中，集成服务器采用基于内容的 Pub/Sub 系统实现数据路由和集成. 客户关系系统中的客户数据由销售系统和财务系统中的数据共同生成，销售系统中的订单数据又需要分裂为库存系统和生产计划系统中的数据，集成服务器中基于内容的 Pub/Sub 系统需要支持这种复杂的事件复合需求. 其次，XML 也逐步成信息表示和交换的标准，事件采用 XML 格式已成为一种发展趋势. XML 是一种自描述语言，在软件集成中只要求被集成方能够处理 XML 文档，而不对其组件构成加以限定，极大地增强了集成可扩展性，其主要不足是集成效率. 如果在基于 XML 的复合事件中允许只传递订阅者感兴趣那一部分内容，减少事件订阅者接受无关信息的可能性，是一种增强 XML 集成效率的可行途径.

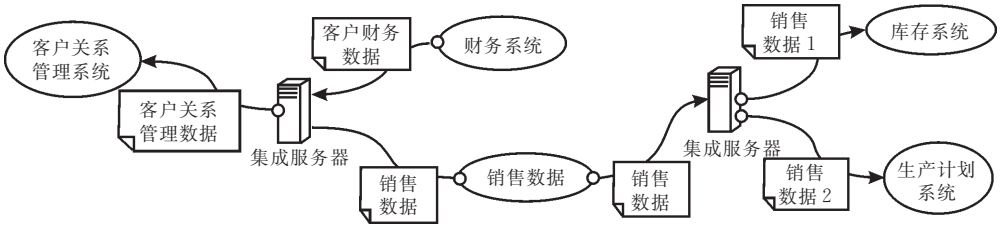

图 1 企业应用集成中一个基于内容的复合事件

针对这些问题，本文采用 XML 作为事件的表达形式，提出一种基于 XML 的复合事件模型及其订阅语言. 第 2 节给出复合事件模型，该模型包含事件时序逻辑模型和事件复合模式，时序逻辑模型论述了在分布环境下事件间的时序逻辑关系，复合模式论述了四种不同的事件复合模式；第 3 节依据复合事件模型，扩展 XML-QL 语言得到一种面向 XML、支持复合事件模式的 EXML-QL 订阅语言；第 4 节根据复合事件订阅语言的特征，构造了复合事件匹配算法并分析了该算法；第 5 节对已有的相关工作做了比较；第 6 节总结了全文并指出进一步的研究方向.

2 复合事件模型

对于基于内容的 Pub/Sub 系统，我们使用事件时序逻辑模型和事件复合模式来构成复合事件模型，时序逻辑模型描述在分布环境下事件在时间序列上的关系；事件复合模式描述多种事件转发和触发模式.

2.1 时序逻辑模型

对于基于内容的 Pub/Sub 系统，事件是信息发布者发布的信息，具有原子性和瞬时性. 在分布环境下，由于网络延时等原因导致事件发布时间不等同于 Pub/Sub 系统接受事件时间，事件时间是一个时

间间隔. 我们依据时间间隔特性，给出时序逻辑关系如下.

定义 1.　$t=[t^s,t^f]$表示一个事件的时间间隔，t^s表示事件发布者发布事件的时间，t^f表示 Pub/Sub 系统接受事件的时间. 时序逻辑关系 T-Logic$\in\{<,\prec,\supset,\cup\}$，其中，“$<$”表示完全顺序关系，“$\prec$”表示部分顺序关系，“$\supset$”表示时间包含，“$\cup$”表示时间合并.

对于事件 e_i 和 e_q，它们的时间间隔时间分别为 t_i 和 t_q，时序逻辑关系满足如下条件：$t_i<t_q\Leftrightarrow t_i^f<t_q^s$，$t_i\prec t_q\Leftrightarrow(t_i^s<t_q^s<t_i^f)\vee(t_i^f<t_q^f)$，$t_i\supset t_q\Leftrightarrow(t_i^s<t_q^s<t_i^f)\vee(t_q^f<t_i^f)$，$t_i\cup t_q\Leftrightarrow[\min(t_i^s,t_q^s),\max(t_i^f,t_q^f)]$.

图 2 给出 t_1,t_2,t_3,t_4 共 4 个事件的时间间隔. 这里，$t_1<t_2$，$t_1<t_3$，$t_1<t_4$，$t_2<t_4$ 而 $t_2\not\prec t_3$；如果使用部分顺序关系“$\prec$”，则 $t_2\prec t_3$；对于 t_4 和 t_3，还存在 $t_4\supset t_3$.

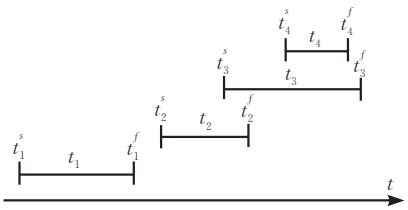

图 2　时间间隔事件顺序关系

2.2　事件复合模式

在基于内容的 Pub/Sub 系统中，事件复合模式说明在典型应用场景下所需要的事件匹配转发方式. 复合事件可以采用多种匹配转发模式，支持对 XML 事件的不同处理. 依据所匹配内容与转发内容的关系及事件的入度和出度，我们定义事件复合模式如下.

定义 2.　复合模式操作符 Composite-Operator$\in\{\xrightarrow{=},\xrightarrow{\approx},\xrightarrow{\supset},\xrightarrow{\subset}\}$，其中，“$\xrightarrow{=}$”表示“转发模式”，它针对“一对一”的事件转发，匹配得到的事件就是需要转发的事件，“$\xrightarrow{\approx}$”表示“触发模式”，它是由多个输入 XML 事件依据一定的时间逻辑关系复合而成，其转发的事件可以是任意事件，“$\xrightarrow{\supset}$”表示“分裂模式”，它针对“一对多”的转发，需要将匹配的事件分裂为多个输出事件，“$\xrightarrow{\subset}$”表示“聚合模式”，它针对“多对一”的转发，需要匹配多个输入 XML 事件，选择多个 XML 事件的部分内容，聚合生成新的 XML 事件，并转发该复合生成的事件.

如果 E 表示事件的集合，$E=\{E_I,E_O\}$表示事件划分为输入事件集合 E_I 和输出事件集合 E_O 两种类型. 如果使用函数 $C(e)$表示事件 e 的内容，那么复合模式操作满足如下关系：

$E_I\xrightarrow{=}E_O$：$\{(E_I,E_O)\mid\exists e_i\in E_I,\ \exists e_q\in E_O,\ e_i=e_q\}$；

$E_I\xrightarrow{\approx}E_O$：$\{(E_I,E_O)\mid\exists e_i\in E_I,\ i=1\rightarrow n,\ \exists e_q\in E_O,\ q=m\rightarrow j,\ (1\le m\le j\le n)\vee(n<m\le j)\}$；

$E_I\xrightarrow{\supset}E_O$：$\{(E_I,E_O)\mid\exists e_i\in E_I,\ \exists e_q\in E_O,\ q=1\rightarrow n,\ \bigcup C(e_q)\le C(e_i)\}$；

$E_I\xrightarrow{\subset}E_O$：$\{(E_I,E_O)\mid\exists e_i\in E_I,\ i=2\rightarrow n,\ \exists e_q\in E_O,\ \bigcup C(e_i)\ge C(e_q)\}$.

如果将事件复合模式按事件所处的状态划分，那么可以分为检测状态、转发状态、触发状态、分裂状态和聚合状态，输入的 XML 事件分别处于不同的状态，事件不同状态的组合构成事件复合模式，如图 3 采用状态图的方法描述了 4 种复合模式.

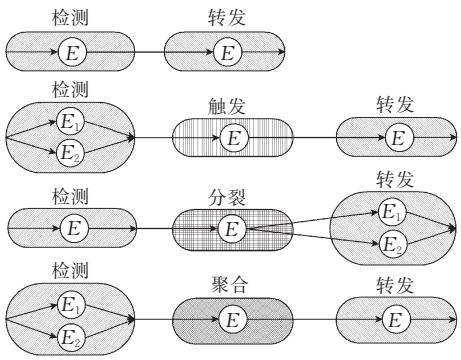

图 3　事件复合模式

3　复合事件订阅语言

在基于内容的 Pub/Sub 系统中，信息发布者以事件的形式发布信息，订阅者使用订阅语言订阅事件，匹配到的事件被及时地转发给订阅者. 其中，事件和订阅语言是一个相互作用的整体，采用何种形式的事件也就决定了其订阅语言的基本形式，反过来，订阅语言的形式也限定了所能订阅的事件. 对于 XML 事件，其本身不同于其它格式的事件，它的内容是以层次树组织的，除包含元素和属性信息外，还包括结构信息.

不失一般性，给出 XML 事件的定义如下.

定义 3.　XML事件是一个三元组：XML-Event=(element, attribute, structure). 其中，element 和 attribute 本身具有名称、类型和值，attribute 依附于特定的元素，structure 限定元素间的结构关系，包括父子、兄弟、祖先和子孙四种结构关系.

例如，下面是一个简单的 XML 格式的事件 e_1.

```
<enterprise>
  <name>DELL</name>
  <product>PC AT, 1.2GHz, 512KB RAM</product>
  <price>5000</price>
</enterprise>
```

这里，e_1 描述为 e_1(name)="DELL"，e_1(product)="PC AT, 1.2GHz, 512KB RAM"，e_1(price)=5000. 我们可以把事件作为一种特殊的状态函数 f，它是一个建立在常量和变量之上的非布尔型表达式. 可以使用 $s[f]=f(\forall v: s[v]/v)$ 表示从状态到值的映射，其中，$f(\forall v: s[v]/v)$ 表示由状态 s 获得的事件 e 的值. 那么，单个的事件匹配 p 是建立在状态函数 f 之上的布尔表达式，$[p]$ 表示从状态函数到布尔值的映射，对于特定状态 $s[p]$ 值为"true"或"false". 对于复合事件订阅语言，每个输入的 XML 事件可以看作为一个处理原子，多个基于时序逻辑的原子事件组合构成了事件触发和转发条件，其订阅语言是时序逻辑操作符附加在值为"true"的 $s[p]$ 的运算，可以表示为 T-Operator[True[$s[p]$]] → {true, false}.

定义 4. 时序操作符 T-Operator ∈ {":", ";", "‖", "⊥", "▽"}，其中：

":"表示串联事件，串联事件表示事件遵循部分顺序关系，在时间上有重叠，如 $E_1:E_2$；

";"表示顺联事件，顺联事件表示事件遵循完全顺序关系，在时间上没有重叠，如 $E_1;E_2$；

"‖"表示并行事件，检测并行发生的两个事件，在时间上相互重叠，对发生的次序没有要求，如 $E_2 \parallel E_1$；

"⊥"表示定时事件，定时检测在特定的时间间隔内(如事件 E_1 和 E_2 间)是否发生特定事件 E_3，如 $(E_1 \perp E_2)E_3$；

"▽"表示交替事件，检测事件 $E1$ 或 $E2$，如 $E_2 \triangledown E_1$.

在无约束的上下文环境下，基于内容的 Pub/Sub 系统可以有多个满足匹配条件的输入事件，但不是所有满足匹配条件的事件对系统是有意义的. 例如，在顺联复合事件($e_1;e_2 \xrightarrow{\approx} e_3$)中，$e_1$ 发生后和 e_2 发生前，就可能有多个 e_3 发生，在这种情况下，需要定义上下文规则指定使用哪个 e_3 事件. 我们使用参数化上下文方法定义面向基于内容的 Pub/Sub 系统的参数化上下文规则，其与主动数据库系统中的方法的不同在于基于内容的 Pub/Sub 系统是基于内容的匹配转发事件，由复合事件构成的事件具有"唯一确定性". 定义参数化上下文规则如下.

定义 5. Parameter-Context ∈ {Recent, Exclusive, Globe}，其中

Recent 表示只有最近发生的事件被使用；

Exclusive 表示使用初始事件和终止事件检测后的事件；

Globe 表示使用初始事件后的所有事件.

我们说"时序逻辑操作"与"参数化上下文规则"构成了复合事件时序判定的"完备性"，即可以唯一地确定一个需要转发的匹配事件.

Pub/Sub 系统的复合事件构成的"唯一确定性"是完备性判定的前提条件. "唯一确定性"是指对于由时序逻辑操作符构成的复合事件，其构成的子事件是唯一确定的，否则将无法判定. 参数化的上下规则 Globe 和 Exclusive 将时序空间划分为两个主要部分，即初始事件开始以后的时间和终止事件结束以后的时间，Recent 规则在两个时间域进一步确定优先规则. 对于五种时序逻辑关系，组合应用 Recent、Exclusive 和 Globe 规则可以得到不同逻辑的唯一性判定，因此也就构成复合时序判定的完备性. 例如，应用 Recent 和 Exclusive 规则于顺联复合事件(如 $E_1;E_2 \xrightarrow{\approx} E_3$)，要求 E_1 和 E_2 都是最近发生的，并且满足顺联时序逻辑关系，唯一确定所触发的事件 E_3 是 E_2 发生后最近发生的事件.

3.1 复合事件订阅语言 EXML-QL

复合事件订阅语言 EXML-QL 是一种基于 XML-QL 语言，通过增加时间操作符、使用"块变量"替代"In 子句"扩展得到的面向 XML 事件的订阅语言. 首先，EXML-QL 语言能够表达所要匹配的事件信息，包括 XML 的元素、属性和结构，支持事件转发模式；其次，EXML-QL 语言能够区分多个不同事件，依据时序逻辑模型构造不同事件间的时序逻辑关系，支持事件触发模式，再者，EXML-QL 语言支持对所匹配内容的选择，由所选择的部分内容生成一个或多个新的 XML 事件，支持复合事件的分裂和聚合模式.

XML-QL 语言本身是一种面向 XML 的查询语言，它的显著能力在于能够在已有的 XML 文档中抽取数据并进一步构造为新的 XML 文档，具有语法简单、关系完整等特点，并能和其他查询优化方法相结合. 在构造 XML 方面，XML-QL 利用混合嵌套功能支持 XML 数据的层次构建，利用 Skolem 函数来规划元素的生成. 图 4 给出一个 XML-QL 的例子.

```
WHERE〈$〉〈author〉〈firstname〉$fn〈/〉
                〈lastname〉$ln〈/〉
            〈/〉
            〈title〉$t〈/〉
        〈/〉IN"www.a.b.c/bib.xml"
CONSTRUCT〈OutputXML〉
              〈person ID=PersonID($fn, $ln)〉
                〈firstname〉$fn〈/〉
                〈lastname〉$ln〈/〉
                〈publicationtitle〉$t〈/〉
               〈/〉
            〈/〉
```

图 4　一个 XML-QL 例子

在图 4 的例中，$ 前缀表示变量，WHERE 和 IN 子句表示在“www.a.b.c/bib.xml”中需要匹配和获取的内容，CONSTRUCT 子句构造一个以 person 元素为根元素的 XML 文档. 其中，PersonID 是 Skolem 函数，它的目的是为每个新的（$fn，$ln）值产生一个新的标识.

尽管 XML-QL 在由多个 XML 文档构造新的 XML 文档方面具有优势，但是在表达事件订阅方面存在不足：

(1) 基于内容的 Pub/Sub 系统是以内容匹配为事件识别的基础，因此无法在事前指定 XML 事件名称；

(2) 在 XML-QL 中，如果忽略 IN 子句，那么就无法区分不同对象的 XML 匹配语句；

(3) XML-QL 没有引入时间概念，不支持时序逻辑判定；

(4) XML-QL 不支持事件复合模式.

我们在利用 XML-QL 已有功能的基础上，进一步扩展得到 EXML-QL 事件订阅语言，定义如下.

定义 6.　EXML-QL 是一种基于 XML-QL 的扩展语言：EXML-QL = {#Block-variable} ∪ {T-Operator} ∪ {Composite-Operator} ∪ {XML-QL}. 其中，# 为块变量前缀，T-Operator 为时间操作符，Composite-Operator 为复合操作符，XML-QL 为 XML 查询语言.

图 5 为一个简单的 EXML-QL 例子. 在该例子中，我们使用了两个块变量 #P 和 #T，表明〈person〉元素和〈orderpayer〉元素分别来自不同的 XML 事件. 在 WHERE 子句的末端，我们引入时序操作符并利用块变量指代不同的事件建立事件间的时序逻辑关系，该例是顺联事件（#P 弱跟随 #T）. 在 CONSTRUCT 子句中，使用 #P 中的内容作为构建输出 XML 的元素. 其匹配过程为在匹配到满足元素〈pay〉等于 finish、元素〈deliver〉等于 ok 并且元素〈pn〉与 #P 事件具有相同的 personnumber 值的事件，在时序上两者遵循弱跟随关系，那么构造一个根元素为〈result〉，具有 $n 值并包含元素〈pn〉的 XML 事件.

```
WHERE〈person〉
          〈name〉〈/〉ELEMENT_AS $n
          〈personnumber〉$pn〈/〉
        〈/〉#P
        〈orderpay〉
          〈pay〉finished〈/〉
          〈deliver〉ok〈/〉
          〈pn〉$pn〈/〉
        〈/〉#T
#T; #P
CONSTRUCT〈result〉〈pn〉$pn〈/〉$n〈/〉
```

图 5　一个 EXML-QL 例子

我们进一步利用语法规则来规范 EXML-QL 事件订阅语言的操作.

语法规则 1.　如果不存在 CONSTRUCT 子句，那么 EXML-QL 处于转发状态. 条件 1：不存在时序操作符，必然不存在块操作符，WHERE 子句所订阅的内容来自单一的 XML 事件，否则 EXML-QL 事件订阅语言将陷入“事件转发混乱”；条件 2：如果存在时序操作符，那么必然存在两值触发操作符“$\xrightarrow{\approx}$”指明所要转发的 XML 事件.

语法规则 2.　如果存在 CONSTRUCT 子句，那么 EXML-QL 处于构造状态. 条件 1：如果不存在时序操作符，存在块操作符，必然遵循“Recent”环境参数，否则 EXML-QL 事件订阅语言将陷入“事件选择混乱”；条件 2：如果存在时序操作符，那么必然不存在两值触发操作符“→”指明所要转发的 XML 事件，否则导致“事件转发冲突”.

下面给出 EXML-QL 事件订阅语言规则的必要性和正确性证明：

如果不存在规则 1，那么 EXML-QL 语言将产生“事件转发混乱”.

证明.　“事件转发混乱”是指存在多个满足内容匹配条件的事件导致系统无法选择转发唯一事件. 对于规则 1，因为不存在 CONSTRUCT 子句，WHERE 子句除须指明所要匹配的事件内容，也应隐含地指出所要转发的事件. 对于不存在时序操作符的 WHERE 子句，如果存在两个或多个块操作符，表示匹配转发多个 XML 事件，即便应用“参数化上下文规则”，系统也无法判定需要转发的 XML 事件，因此，产生 EXML-QL 语法错误；如果是单一

块变量并应用“Recent”环境参数，系统可以判定最近匹配的 XML 事件即为唯一需要转发的 XML 事件. 对于存在时序操作符的 WHERE 子句，必然存在两值触发操作符“$\overset{\approx}{\longrightarrow}$”来限定所要转发的 XML 事件，如果不存在两值触发操作符“$\overset{\approx}{\longrightarrow}$”，同样，即便应用“参数化上下文规则”，系统也无法判定需要转发的 XML 事件.

如果不存在规则 2，那么 EXML-QL 语言将产生“事件选择混乱”和“事件转发冲突”.

证明. “事件选择混乱”是指系统无法判定依据哪个 XML 事件来构造新的 XML 事件. 对于包含 CONSTRUCT 子句的 EXML-QL，“事件转发冲突”是指构造的 XML 事件与指定转发的事件冲突. 在 EXML-QL 处于构造状态时，可以通过使用多个块变量表明多个 XML 事件，如果不存在时序操作符，表明多个 XML 变量不存在时序逻辑关系，而新构造的 XML 事件由这多个 XML 事件共同组成，那么这些多个 XML 事件是构造新的 XML 事件的必要条件，因此，如果不应用“Recent”环境参数，那么可能发生同时匹配到多个 XML 事件，系统将无法判定需要使用的 XML 事件. 对于 EXML-QL 处于构造状态，存在时序操作符，必然不再使用复合模式操作符，因为复合模式操作符的右值表明所要转发的 XML 事件，而 CONSTRUCT 子句表示的事件为转发事件，两者矛盾.

4 复合事件匹配算法

本节介绍基于 EXML-QL 语言的复合事件匹配算法，该算法完成由 EXML-QL 订阅语言匹配复合 XML 事件并创建新 XML 事件的功能，主要包括以下几步：

(1) 将 WHERE 子句分解为多个独立的 XML 事件匹配语句，完成 XML 事件的匹配；

(2) 依据块变量建立的时序逻辑，对匹配的 XML 事件进行事件时序逻辑判定；

(3) 将满足时序逻辑判定的 XML 事件转换为固定的 XML 文档，EXML-QL 相应地等价变换为 XML-QL，由 XML-QL 解析器生成新的 XML 事件.

4.1 匹配 XML 事件

在 EXML-QL 事件订阅语言中，WHERE 子句包含多个输入 XML 事件并以块变量区分不同的 XML 事件. 对于每个块变量，其内容和结构为一个连续的 XML 文档片断，与传统的基于约束集合的订阅语言相比，使用块变量匹配 XML 事件引入如下新问题：

(1) 在块变量中，元素是具有层次结构的，而且包含元素统配符“*”和变量操作符“$”，但不包含子孙间隔操作符“//”；

(2) 在块变量中，元素是有序的，对于同一层次的元素具有先后顺序，并且在同一层次可以存在多个相同元素；

(3) 元素节点可以包含零个或多个匹配谓词、匹配元素及其属性的值.

为了解决这些问题，我们采用近似有限状态机的方法来构造 XML 事件匹配算法. 每个块变量分解为多个元素，元素对应于状态，元素的结构关系转换为状态间的转换映射，由 XML 解析器解析的事件驱动状态机的执行，当状态机的最后状态到达时，认为块变量与输入 XML 事件相匹配.

首先，将块变量分解为元素节点的集合，采用数组结构表示元素，其结构如图 6 所示. 其中，包含元素名称、元素在 XML 事件中的结构信息及元素的匹配谓词. 对于元素统配符“*”不产生元素节点，对于变量操作符“$”不作谓词判定.

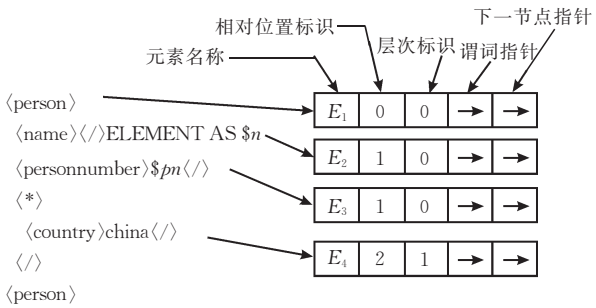

图 6 WHERE 子句的分解

图 6 中的各项描述如下：

(1) 元素名称. 元素名称是元素节点的表示，其等同于 XML 事件中的元素名称.

(2) 相对位置标识. 相对位置标识是一个整数，描述在文档层次结构中的该元素位置与上一个元素位置间的相对距离. 对于第一个元素，该值为 0，其他元素的值为 1 加上在该元素与上一个元素间的元素统配符“*”的个数.

(3) 层次标识. 层次标识是一个整数，表示被检查的元素节点在 XML 文档中的层次. 由于在 XML 文档中不限制元素出现在多个文档层次，并且块变量表示一个相对位置的匹配，因此在转换块变量时，不可能分配一个预定的值，该值是在 XML 事件解析的

过程中更新. 在层次标识初始化阶段，如果相对位置标识的值大于 1，则层次标识设定为 1；否则为 0.

(4) 匹配谓词. 元素节点可以包含零个或多个匹配谓词，匹配谓词作为元素节点的属性，由元素节点引用.

(5) 下一节点指针. 每个元素节点都包含指针，指向在块变量中需要匹配的下一个元素节点.

对于基于内容的 Pub/Sub 系统，匹配效率是其主要目标之一. 我们采用基于主元素名称建立多个订阅的索引，其匹配过程基于“推”式的方法，即使用 XML 事件匹配订阅. 所谓的主元素是指块变量中的第一个元素. 对于具有相同名称的主元素，其位置标识、相对位置标识、层次标识具有相同的值，通过合并具有相同名称的主元素，也就是进一步合并多个匹配为一个单一匹配，降低了需要检索的订阅数量，XML 事件订阅索引结构如图 7 所示. 使用 SAX XML 解析器解析输入的 XML 事件，SAX 解析器提供基于事件的 XML 解析界面，它使用回调机制报告解析的事件. SAX 解析器为每个解析事件传递元素及其在 XML 文档中的层次. 例如，当 SAX 解析器遇到元素开始标签时，它向回调函数传递元素的名称、元素的层次以及元素的属性和值. 对于每个解析事件，我们实现三种函数分别处理元素的开始标签、元素的结束标签和元素的内部数据. 首先，检查匹配索引中的主元素名称表，找到所匹配的主元素名称；其次，对于其余的元素名称节点，执行三种类型的检查：名称检查、层次检查和谓词匹配检查. 名称检查的目的是判定所匹配的元素名称一致；层次检查的目的是利用元素解析时所更新的层次标识和相对位置标识，判定元素在 XML 文档中出现的层次位置是否满足块变量的层次要求；谓词匹配检查主要判定元素的值及其属性值是否满足匹配谓词. 匹配算法描述如下.

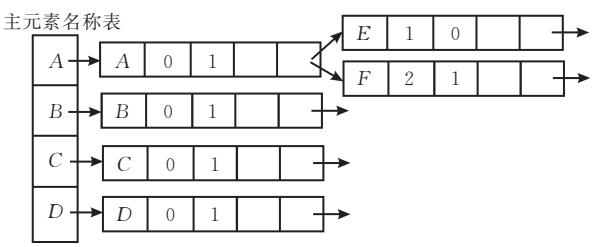

图 7　XML 事件匹配索引结构

算法 1.　*Match-XMLEvent*(D, MT, NC).

输入：D 表示输入的 XML 事件、MT 主元素名称表、NC 表示元素节点链

输出：R 表示与 XML 事件 D 匹配的块变量集合

```
1. Initialize R to be empty, DN[i], i=0, flag=false
   //DN[]表示 XML 事件的元素名称组成数组，i 为
   //数组指针，flag 为查找主元素名称表的成功标志
2. Repeat
3.   en=DN[i++]
4.   flag=Search(MT, en)
5.   if(flag=false) then goto 3
6.    else
7.     temp=NC.nextnode(en)
8.     if(temp=null) then
9.      insert Block-id into R //匹配成功，将块变量
                               //id 插入 R
10.     else if (MatchElementName(temp,NC,DN)=
            false) then goto 3
          //MatchElementName()为元素名称匹配
11.      else if (CompareHRN(temp,NC,D)=false)
            then goto 3  //元素层次匹配
12.       else if(ComparePredicate(temp)=true)
            then goto 7  //谓词匹配
13.        else goto 3
14. Until(D has been completely parsed)
15. return R
```

4.2　时序逻辑判定

对于包含时序逻辑的 WHERE 子句，我们构建事件图来判定复合事件的时序逻辑，事件图结构如图 8 所示. 它是一个非循环有向图，顶点具有两个入度和一个出度，标记为时序操作符，整个事件图的输出“true”或“false”，表示是否触发 XML 事件或构造 XML 事件. 此外，对于每个顶点，我们添加属性约束作为时序判定指定的“参数上下文规则”. 下面给出一个在“Recent”参数化上下文规则下的“串联事件”和“交替事件”复合事件判定算法.

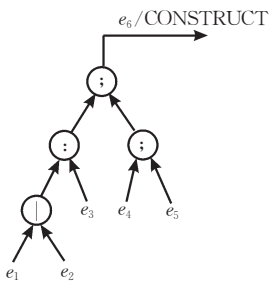

图 8　事件图

算法 2.　在“Recent”参数化上下文规则下的复合逻辑判定算法.

1. 对于事件图中的每个输入事件维护一个独立的输入列表 $List\_e_i$

2. if (e_i is arrived)

3.　if node is “串联事件”

```
4.   if e_i is left_event  //left_event 表示时序操作符的
                           //左值输入事件;
5.     add e_i to List_e_i
6.   if e_i is right_event //right_event 表示时序操作符的
                           //右值输入事件;
7.     if (e_i+1 in List_ e_i+1) and (e_i_t^s < e_i+1_t^f < e_i_t^f)
8.     Pass <e_i, e_i+1> to the parent_node
9.   else delete e_i in List_e_i
10. if node is "交替事件";
11.   Pass e_i to the parent_node.
```

4.3 构建 XML 事件

在构建新的输出 XML 事件方面，我们主要利用现存的 XML-QL 解析器，做如下处理：

(1) 转换 EXML-QL 表达为 XML-QL 表达；

(2) 将所匹配的并且符合复合逻辑判定的块变量作为新命名的 XML 文档输入 XML-QL 解析器；

(3) 由 XML-QL 解析器生成新的输出 XML 事件.

图 9 给出与图 5 中的 EXML-QL 的例子对应的 XML-QL.

```
WHERE <person>
        <name></>ELEMENT_AS $n
        <personnumber> $pn</>
      </>IN "P.xml"
      <orderpay>
        <pay>finished</>
        <deliver>ok</>
        <pn> $ pn</>
      </>IN "T.xml"
CONSTRUCT<result><pn> $pn</> $n</>
```

图 9 一个 EXML-QL 与 XML-QL 转换的例子

4.4 算法分析与实验

在事件匹配阶段，可以通过计算在匹配过程中所访问的元素节点个数来预测算法的时间复杂度，假设算法在每个元素节点执行简单的元素名称、层次和匹配谓词的判定，其消耗时间可以认为是固定常量，那么匹配 XML 事件所消耗的时间正比于所访问节点个数是合理的. 相应地，事件匹配时间是块变量数量和大小的函数，同时也是特定输入 XML 事件的函数，不同的输入 XML 事件导致不同的匹配时间. 那么，匹配一个 XML 事件所花费的时间，即便在最坏的情况下，也线性小于元素节点的个数. 设 K 为块变量中元素节点的个数，S 为块变量的集合，$C(s)$ 表示在块变量集合 S 中匹配 XML 事件所花费的时间，t 为在每个节点匹配所消耗的时间. 由于每个块变量采用线性数据表示，多个线性数组的集合构成了块变量集合 S，那么 $C(s) \leq K|S+m(n-1)|t$，其中，m 为具有相同主元素块变量的个数，n 为主元素的重复数量.

在时序逻辑判定阶段，需要判定的数目等于时序逻辑表达式中所包含的时序操作符的数目，依据事件图，判定数目等于事件图的顶点数目. 由于“Recent”规则是操作符判定的唯一“参数化上下文规则”，而其他规则为事件输出规则. 并且，在事件图的每个定点只作一个时序比较，那么，我们可以说逻辑判定所消耗的时间正比于事件图的顶点数目. 设 N 为事件图的顶点数目，t 为在每个顶点判定所消耗的时间，$E(G)$ 表示判定一个事件图所消耗的时间，那么，$E(G) \leq kNt$，其中 k 为常量.

目前，我们在 CUP 2.8GHz、512MB RAM 的普通 PC 机上，采用 Java(JDK1.3) 建立了一个模拟环境，模拟了“顺联事件”的匹配率，匹配率是指已匹配信息的数量与信息总数量之比. 对于匹配率计算，我们限定订阅语言 EXML-QL 的时序逻辑表达式只包含一个“顺联事件”，依据“顺联事件”构造一个 XML 输出事件，固定总共 500 个订阅，得到 1000 个块变量，使用 500 个输入 XML 事件模拟匹配，这 500 个输入事件由不同的 10 组 XML 输入事件构成，其中，满足匹配条件的输入事件为 200 个，采用随机方法输入 XML 事件，其实验结果如表 1 所示. 可以看出匹配率与信息输入的先后顺序密切相关，随着输入信息的积累，匹配率(表示为匹配成功的个数)逐步提高.

表 1 匹配率实验结果

事件数(个)	匹配数(个)
100	3
200	7
300	9
400	14
500	21

5 相关工作比较

目前在基于内容的 Pub/Sub 方面，较有影响的系统包括 Gryphon，Siena 和 Elvin，在这些系统中采用的事件表达格式与订阅语言彼此相似，其基本构成都是采用单一的语义约束，不支持复合事件，只是单一事件转发[2-4]. Gryphon 的订阅语言只能包含“与”关系，这就导致了表达订阅语言的复杂性. Siena 的订阅语言除支持“与”关系，与其它方法不同的是引入模式的概念，允许在不同的事件中建立联系. Elvin 也使用 BDD(Binary Decision Diagram) 方法，

具有更大的表达能力，并且支持对字符串操作，支持“或”关系，但没有再进一步的扩展. 基于 XML 事件的 Pub/Sub 系统主要有 XFilter[5]、YFilter[6]、XTrie[7]和 WebFilter[8]，这些系统更确切地说应属于信息检索领域，它们面向的是在 Internet 中发布的 XML 信息，采用 XPath 作为 XML 选择语言，在信息选择粒度和表达能力方面优于基于关键字和字包(word-bag)的信息获取方法以及 XML-QL 语言，但 XPath 不支持对多个 XML 文档复合操作，也不支持新 XML 文档的生成. XFilter、YFilter、XTrie 和 WebFilter 在检索算法方面各有特点，XTrie 采用一种基于子串的方法处理 XPath 语句，并基于子串构造了匹配算法，这种方法较适合于 XPath 语言，不太适合 XML-QL 语言. YFilter 系统将 XPath 语言转化为有向图，多个 XPath 语句中相同的元素合并为一个节点，对于 Pub/Sub 系统，我们认为元素的等同取决于多方面，在多个订阅语言中，存在相同元素的几率并不高，因此没有借鉴. 与本文的方法相比，本文中的块变量的数据结构与 XPath 语句数据结构表示相类似，不同的是在本文中维护一个基于主元素的索引表，允许建立一对多的连接，具有更高的查询效率；此外，其他算法只是对 XML 文档的整体选择，没有涉及对部分内容的获取及构造新的 XML 文档.

对于复合事件时序逻辑的研究多集中在主动数据库领域，如 SAMOS 数据库使用 Petri Nets 检测复合事件，可以表示并行行为和管理具有参数的复杂数据. 文献[9]设计了一种称为 Snoop 的语言支持复杂的时间表达，并使用树形结构来反映复合事件的检测. 目前，对 Pub/Sub 系统中的复合事件时序逻辑模型的研究较少，文献[11]论述了在分布环境下的复合事件检测，与本文相比，它更注重于对分布环境下的复合事件检测的拓扑分布结构的研究，并且它们都不支持分裂与聚合复合模式，而本文更侧重于复合模式及其订阅语言.

6　总结及进一步的工作

在软件集成系统中，XML 已逐步成为分布软件系统的信息交换标准，基于内容、面向 XML 的 Pub/Sub 系统为有效地路由、聚集和转换来自不同信息源的 XML 信息提供了可行的机制，与其它集成中间件相比，具有更小的订阅表达粒度、更大的路由柔性、集成可扩展性及灵活的 XML 事件处理. 目前现存的事件系统主要是集中在对单一事件匹配方法的研究，缺乏对复合事件模型及其订阅语言的研究. 针对该问题，本文提出一种基于 XML 事件的复合事件模型及其订阅语言，时序逻辑模型论述了在分布环境下事件间的时序逻辑关系，复合模式论述了四种复合事件模式，包括支持 XML 事件的分裂与聚合. 在此基础上，通过引入块变量、时序逻辑操作符扩展 XML-QL 语言得到面向 XML 的复合事件订阅语言 EXML-QL，支持如选择多个输入 XML 事件构造输出新的 XML 事件的复杂匹配操作. 最后，依据订阅语言的结构特征，构造了复合匹配算法并加以分析与验证.

在下一步的工作中，我们将重点放在对 XML-QL 语言的进一步改造上面，研究使用 XPath 语言代替块语言，使 XML-QL 语言具有更大的订阅表达能力，以及如何在订阅语言中添加数据转换原语，进一步满足集成需求.

参　考　文　献

[1] Antonio Carzaniga, Alexander l Wolf. A benchmark suite for distributed publish/subscribe systems. Department of Computer Science, University of Colorado, Colorado: Technical Report: Colorado 80309-0430, 2002

[2] Banavar G, Chandra T, Mukherjee B. Matching events in a content-based subscription system//Proceedings of the 18th ACM Symposium on Principles of Distributed Computing (PODC'99). Atlanta USA, 1999: 53-61

[3] Carzaniga A, Rosenblum D S. Wolf: Design and evaluation of a wide-area event notication service. ACM Transactions on Computer Systems, 2001, 19(3): 332-383

[4] Gough J, Smtih G. Efficient recognition of events in a distributed system//Proceedings of the 18th Australasian Computer Science Conference(ACSC18). Glenolg South Australia, 1995: 55-65

[5] Altinel M, Franklin M J. Efficient filtering of XML document for selective dissemination of information//Proceedings of the 26th International Conference on Very Large Data Bases. Cairo Egypt, 2000: 53-64

[6] Diao Yanlei, Fisher Peter, Franklin M J. Raymond To. Yfilter: Efficient and scalable filtering of XML documents//Proceedings of the 18th International Conference on Data Engineering(ICDE'02). San Jose CA, 2002: 341-342

[7] Chan C Y, Felber P, Garofalakis M, Rastogi R. Efficient filtering of XML documents with XPath expressions. The VLDB Journal, 2002, 11(4): 354-379

[8] Pereira J., Fabret F., Llirbat F., Jacobsen H A, Shasha D. WebFilter: A high throughput XML-based publish and subscribe system//Proceedings of the 27th International Conference on Very Large Data Bases. Roma, Italy, 2001: 723-724

[9] Chakravarthy S, Mishra D. Snoop: An expressive event specification language for active databases. Department of Computer and Information Science, University of Florida, Florida USA: Technical Report: UF-CIS-TR-93-007, 1993

[10] Pietzuch Peter R, Shand Brian, Bacon Jean. Composite event detection as a generic middleware extension. IEEE Network, 2004, 18(1): 44-55

XU Gang, born in 1973, Ph. D.. His research interests include software engineering, distribution computation, software integration.

MA Jian-Gang, born in 1977, Ph. D. candidate. His research interests include distribution computation, middleware.

HUANG Tao, born in 1965, Ph. D., professor, Ph. D. supervisor. His research interests include software engineering, distribution computation.

Background

The issue discussed in the paper belongs to integration middleware in the area of software integration. The paper presents a XML-based composite event model and its subscription language to extend the traditional "one-to-one" single event matching to "many-to-one" or "one-to-many" composite event matching. The XML-based composite event model includes the temporal logical model and the event composite pattern that induces some typical event composite patterns and presents the some temporal logical relation between events. The composite event subscription language is introduced by adding the block-variables and the temporal logical operators in XML-QL language to support XML-based composite event patterns. Based on the both, the composite event problem in Pub/Sub systems can be further resolved.

Compare with the paper, the existing Pub/Sub systems focus on the "one-to-one" single event matching pattern, not composite event patterns. Furthermore, the events usually adopt the single semantic constraint sets, not XML-format. For example, based on supporting the single event matching, Siena tries to uses the "pattern" concept to define relations between different events. In the aspects of XML, XFilter adopts XPath as subscription languages to performing efficient filtering of XML documents for large-scale information dissemination systems, but XPath has not ability to construct new XML document and does not contain the time concept so no way to support the composite event subscriptions. The research in the paper is beneficial to promote the progress of Pub/Sub systems, especially as a integration middleware.

This research is supported by the project of "the Research of Net-component Software Middleware Platform Model and Framework" as the fifth sub-project under the National Basic Research Program (973 Program) of China project-"The Research of Agent-Based Software Middleware Theory and Method under the Internet Environment", and the National High Technology Research and Development Program (863 Program). The fifth sub-project mainly focus on the middleware platform supporting the running of net-component under Internet, and researches the common and basic services needed by net-component, such as platform model, framework and prototype. In the past years, the authors have researched and developed Web Application Server, Data Integration Middleware, Workflow, Transaction Monitor, Information Portal, and so on. As the important part of the fifth sub-project, the research of the paper can be regarded as an intelligent routing mechanism between net-components and ultimately builds a middleware-based integration and collaboration model and method.