

# 一种面向项目管理信息系统集成的复杂事件处理方法

李青 王东 王卓昊 张志刚

(中国科学技术信息研究所 北京 100038)

**摘要** 大型项目管理信息系统中实时产生的海量事件,对深度挖掘系统的隐含特征具有重要意义。通过分析科技计划项目管理的业务逻辑和事件结构,提出项目管理事件模型。为描述事件之间的关系,定义项目管理事件操作符,实现事件的逻辑、算数和时间的计算。基于复杂事件处理技术,提出一种面向项目管理信息系统的事件处理算法PMEP,执行事件处理规则,分析系统产生的海量事件,得到深层次的项目管理复杂事件。实验表明,PMEP算法可实现项目管理信息系统中海量事件的实时处理。

**关键词** 复杂事件处理 科技管理业务模型 事件处理算法 科技项目管理信息系统

中图分类号 TP3

文献标志码 A

DOI:10.3969/j.issn.1000-386x.2024.09.054

## A COMPLEX EVENT PROCESSING METHOD FOR PROJECT MANAGEMENT INFORMATION SYSTEM

Li Qing Wang Dong Wang Zhuohao Zhang Zhigang

(Institute of Scientific and Technical Information of China, Beijing 100038, China)

**Abstract** The massive events generated in real time in large project management information system have great significance for the deep mining of the hidden characteristics of the system. By analyzing the business logic and event structure of science and technology project management, the project management event model is proposed. In order to describe the relation among events, the project management event operators were defined to achieve the logic, arithmetic and time calculation of events. Based on complex event processing technology, an event processing algorithm called project management event processing (PMEP) for project management information system was proposed, which executed event processing rules and analyzed large amounts of events generated by the system to obtain the complex events of deep project management. Experimental results show that PMEP algorithm can effectively achieve the real-time processing of massive events in project management information system.

**Keywords** Complex event processing Science and technology management business model Event processing algorithm Science and technology project management information system

## 0 引言

随着中央财政科技计划(专项、基金等)管理改革的深入推进<sup>[1-3]</sup>,项目申报、评审与管理流程不断演进,为支撑多计划、多专项、多年度、多批次、多阶段的科技计划项目管理,政府部门和科研院所使用的科技项目管理信息系统集成度与复杂度不断提高。为支撑科技计划项目管理复杂的业务环节和动态多变的业务

流程,科技项目管理信息系统在建设过程中通常采用SOA(Service Oriented Architecture)<sup>[4]</sup>架构,利用门户技术<sup>[5]</sup>集成项目申报、专家评审、过程管理、数据查询等应用模块,目前,一般的管理系统集成应用模块已达数百个。

科技项目管理信息系统的集成环境中,数百台服务器与应用模块,连续不断地产生着海量信息,涵盖复杂业务流程、应用系统操作、数据资源使用、底层服务状态等。然而,这些信息尚未得到及时有效的处理,存

在项目管理业务状态更新不及时、服务压力负载不均衡、资源监控不全面等问题。随着科技项目管理信息系统集成规模的增长,海量的底层信息(原子事件)应被及时处理,通过一定的规则形成高层的系统通知(复杂事件),及时反映出各类系统的运行情况、业务状态、异常问题,对科技项目管理信息系统的业务活动管理、用户访问控制、服务状态监控都具有重要的意义。

目前,CEP(Complex Event Processing)复杂事件处理技术主要应用在分布式环境下海量信息流处理问题,可对大规模事件流进行逻辑、算术、时间计算。复杂事件处理技术可应用在基于 SOA 架构或 EDA 架构的系统中,复杂事件处理系统架构如图 1 所示。复杂事件处理系统一般由复杂事件处理引擎、预定义的事件规则、事件代理和复杂事件反馈模块组成<sup>[6]</sup>。事件规则由业务用户预先定义,是面向多源事件的一系列计算与操作的组合规则,通常由特定的事件处理语言描述,常见的事件处理语言有 pseudo<sup>[6]</sup>、EPL<sup>[7]</sup>、LAIPL<sup>[8]</sup>等。复杂事件处理引擎是复杂事件处理系统架构的核心模块,依据事件处理规则,将原子事件转换成复杂事件,复杂事件处理引擎有 Esper<sup>[7]</sup>、LAIPE<sup>[8]</sup>、CEP-Wizard<sup>[9]</sup>、LiSEP<sup>[10]</sup>等。事件代理在原始信息流或事件流中发现原子事件。复杂事件反馈模块将复杂事件处理引擎产生的复杂事件反馈给定制用户或系统。



图1 复杂事件处理系统架构

复杂事件处理技术已被广泛应用在各行各业<sup>[11-15]</sup>,一定程度上解决了各种业务领域复杂事件的实时汇聚、分析、处理问题。然而,在科技项目管理信息系统中,复杂事件处理技术应用较少,大量的系统事件没有得到及时处理和充分利用。本文面向科技项目管理信息系统的集成环境,采用复杂事件处理技术,通过事件规则发现海量原始事件中隐含的重要信息,研究门户、申报、评审、管理等应用系统的海量事件逻辑表达与实时处理方法,提升科技项目管理信息系统的实时信息处理能力。

## 1 国内外研究现状

### 1.1 项目管理信息系统

项目管理信息系统(Project Management Information System, PMIS)通常集成了人员、单位、权限、信息等项目要素,有效地管理各类系统模块、数据资源,调度用户请求与系统响应,实时监控用户业务操作与资源使用情况<sup>[16]</sup>。项目管理信息系统通常由项目管理、评审管理、信息维护 and 安全管理四个部分子系统组成,每个子系统又可以划分为更小的功能模块。国内规模较大的项目管理信息系统包括国家科技管理信息系统、科学基金网络信息系统、中国科学院资源规划项目。

国家科技管理信息系统<sup>[17]</sup>是国务院要求科技部建设的项目管理信息系统,面向国家科技计划项目管理业务流程与数据资源,收集、存储、处理、管理各类科技计划项目数据,对需求征集、指南发布、项目申报、立项和预算安排、监督检查、验收结果等进行全过程信息管理。

国家自然科学基金委员会建设了科学基金网络信息系统(Internet-based Science Information System, ISIS)<sup>[18]</sup>,为广大科研工作者和科研管理人员提供协同创新的平台,支撑从申报、受理到评审等重要环节,有效地支持国家自然科学基金项目的全过程精细化管理,辅助科学决策。

中国科学院资源规划项目(Academia Resource Planning, ARP)<sup>[19]</sup>,是实现中国科学院资源规划的信息系统工程,在科研活动管理中综合运用信息技术和管理理念,以科技计划与执行管理为核心,对中国科学院人力、资金、科研基础条件等资源配置及相关管理流程进行整合与优化,构建有效的管理服务信息技术平台。

### 1.2 复杂事件处理

复杂事件处理是由斯坦福大学的 Luckham 等<sup>[20]</sup>提出,是一种实时处理连续信息流的技术,可用于处理高速输入数据流,获取数据流内实时信息,增强业务系统迅速响应能力<sup>[15]</sup>。复杂事件处理技术已被广泛应用于大量行业领域,例如智能交通<sup>[11]</sup>、RFID 系统<sup>[12]</sup>、在线教育<sup>[13]</sup>、实时监控<sup>[14]</sup>、其他自动化系统<sup>[15]</sup>。

文献[11]提出把复杂事件处理技术应用在我国高铁车身信息系统的多模式事件处理中,通过一种复杂事件描述模型和多模式事件监测模型(Multipattern Event Processing, MPEP),有效提升高铁信息系统中的复杂事件检测能力。文献[12]针对 RFID 系统中的事件处理问题,提出一种基于时间事件模型和语言的复

杂事件处理方法,利用状态机实现事件的实时处理,有效解决了 RFID 系统中时间事件的描述和发现等问题。文献[13]提出一种服务大规模在线课程管理的复杂事件处理功能模型,实时处理不同来源产生的异构数据,并对各类数据进行语义分析,实现用户活动的实时、有效管理。文献[14]对复杂生产活动的业务流程异常状态变化问题进行研究,通过复杂事件处理技术和相关的事件内容模型,将不同阶段的业务活动状态迁移进行建模,利用复杂事件处理引擎实现多阶段业务活动中异常状态监控。

复杂事件处理技术通常用于处理海量事件流,根据预定义的事件模型与事件规则来标识各类事件,进而发现海量事件中隐含的重要信息。本文通过复杂事件处理技术的优势,将复杂事件处理技术应用在项目管理领域中,根据项目管理业务活动特征提出面向项目管理的事件模型和事件查询规则,通过复杂事件处理引擎执行事件查询规则并处理原始事件,实现大型项目管理信息系统中海量事件的实时处理。

## 2 项目管理事件建模与处理方法

### 2.1 项目管理业务模型

根据国发[2014]64号文件<sup>[2]</sup>要求与国家科技计划历史情况<sup>[21]</sup>,科技计划分为国家自然科学基金、国家科技重大专项、国家重点研发计划、技术创新引导专项、基地和人才专项、973计划、863计划、科技支撑计划、火炬计划等100多个类别,每一类科技计划的管理流程都具有各自特点,互不相同。综合考虑科技部诸多科技计划管理过程,可将项目管理划分为多个阶段,包括申报、评审、立项、执行、验收等。

文献[22]分析了一般项目管理信息系统的信息流和功能模型,面向伊朗政府项目管理部门提出一种基于Web的项目管理功能模型,提高伊朗政府的项目管理与决策支撑能力。本文结合文献[17]所述管理业务体系和通用科技项目管理信息系统架构,提出一种项目管理业务模型,如图2所示。

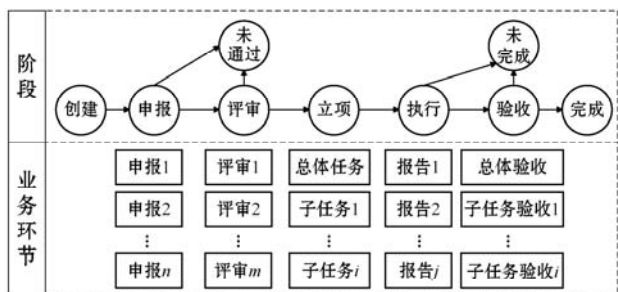


图2 项目管理业务模型

本文提出面向管理的项目管理业务模型,将项目全生命周期划分为创建、申报、评审、立项、执行、验收、完成、未通过、未完成9个阶段,每一个阶段对应多个业务环节,适用于各类科技计划(专项)项目全过程管理。项目管理业务模型中的项目阶段,随着项目管理全过程中业务环节的推进而变迁,每一个通过验收的项目都需要经历创建到完成7个阶段;未立项的项目在申报或评审阶段未通过审核则进入未通过阶段;已立项项目在立项、执行或验收阶段存在问题不能通过验收则进入未完成阶段。

### 2.2 项目管理事件模型

复杂事件处理技术通常将事件定义为“在特定系统或领域内,已经发生或正在进行的事情”<sup>[20]</sup>。在项目管理信息系统中,存在很多业务活动或系统操作,例如用户登录系统、项目信息提交、申报材料合成、课题验收通过、某台服务器的CPU使用率、某个应用系统服务中断、一次数据查询等。任何一个业务活动、系统操作或服务状态都是一个事件,每个事件都在某个时间,携带特定的信息或状态,在底层系统产生。结合文献[8]的研究,考虑项目管理业务模型中项目全生命周期的9个阶段,本文提出一种项目管理事件模型,包含原子事件模型和复杂事件模型。项目管理事件模型相比其他事件模型<sup>[8,11,13-14]</sup>,是一个阶段化的事件模型,满足项目管理的业务活动需求,可以灵活表示项目全生命周期的业务活动与系统服务情况。

**定义1** 原子事件(Primitive Event, PE)指在某个时间点发生的一个事情,定义为如下六元组:

$$P_E = \{ T, O, P, S, A, T_{timestamp} \} \quad (1)$$

式中: $T$ 表示事件的类型,例如用户登录、项目提交、服务器运行监控、数据下载等; $O$ 表示事件来源,一般是服务器IP地址或应用系统id; $P$ 表示科技计划或专项的类别,即前文所述100多种科技计划类别; $S$ 表示项目管理阶段,对应项目管理业务模型中的9个阶段(创建、申报、评审、立项、执行、验收、完成、未通过、未完成); $A$ 表示事件属性的值,一般是文本或数字; $T_{timestamp}$ 表示事件发生时刻的时间戳,格式为YYYYMMDD HH:MM:SS。

以用户提交项目的事件为例,原子事件  $p_e^{project\_submit}$  表示一个青年资助计划申报阶段的编号为P2018YOU0123456的项目,在申报系统中完成提交,可形式化表达为:

$$p_e^{project\_submit} = \{ Project\ Submit, Declare\ App, Youth$$

Assistance Plan, Declare, project id:P2018YOU0123456, 20210122 18:39:23}

**定义 2** 复杂事件(Complex Event, CE)是多个原子事件或复杂事件根据特定事件处理规则组合而成的事件,定义为式(2)所示六元组。事件处理规则描述复杂事件生成条件,分为算数操作、逻辑操作和滑动窗口操作。

$$C_E = \{T, O, P, S, A\{A_1, A_2, \dots\}, T\{t_{begin}, t_{end}\}\} \quad (2)$$

与原子事件不同,复杂事件由多个事件组成,其属性是多个事件属性的集合,即 $\{A_1, A_2, \dots\}$ ,其时间是一个时间段,即 $\{t_{begin}, t_{end}\}$ ,其中 $t_{begin} \leq t_{end}$ 。实际上,复杂事件模型继承了原子事件模型的各种特征,当 $t_{begin} = t_{end}$ ,且 $A_2 = \text{null}, A_3 = \text{null}, \dots$ ,复杂事件模型可用于表示原子事件。

**定义 3** 事件类是具有相同结构和语义内容的事件对象的集合,每一个事件对象都是事件类的一个实例。事件类可以表示事件生产者产生的原子事件,也可以表示事件处理引擎产生的复杂事件。本文用 $C$ 表示事件类, $T$ 表示事件类型, $O_i$ 表示事件类的来源, $U_1^n C_{O_i}^T$ 表示 $n$ 个事件类型为 $T$ ,来源分别是 $O_1, O_2, \dots, O_n$ 的事件类的集合,简称为 $C^T$ 。

$$C^T = \{C_{O_1}^T, C_{O_2}^T, \dots, C_{O_n}^T\} = U_1^n C_{O_i}^T \quad (3)$$

例如,应用系统的在线人数事件类 $C^{\text{Online\_users}}$ ,由每一个应用系统作为事件来源对应的在线人数事件类 $C_{\text{System}_i}^{\text{Online\_users}}$ 组成。然而每一个 $C_{\text{System}_i}^{\text{Online\_users}}$ 是由其对应的所有复杂事件 $c_e$ 和原子事件 $p_e$ 组成,用 $e$ 表示任意一个事件(原子事件或复杂事件),事件类 $C^{\text{Online\_users}}$ 可表示为:

$$\begin{aligned} C^{\text{Online\_users}} &= \{C_{\text{System}_1}^{\text{Online\_users}}, C_{\text{System}_2}^{\text{Online\_users}}, \dots, C_{\text{System}_x}^{\text{Online\_users}}\} = \\ &U_1^x C_{\text{System}}^{\text{Online\_users}} = \{p_{e_1}^{\text{Online\_users}}, p_{e_2}^{\text{Online\_users}}, \dots, \\ &p_{e_n}^{\text{Online\_users}}, c_{e_1}^{\text{Online\_users}}, c_{e_2}^{\text{Online\_users}}, c_{e_m}^{\text{Online\_users}}\} = \\ &\{U_1^n p_e^{\text{Online\_users}}, U_1^m c_e^{\text{Online\_users}}\} \quad (4) \end{aligned}$$

### 2.3 项目管理事件处理操作符

事件处理操作符是用于描述事件之间的关系,也可以表示出复杂事件组成关系。根据文献[12, 23]的事件处理操作符介绍,本文结合科技计划项目管理的业务特征与事件模型,对事件处理操作符进行扩展,提出适用于项目管理过程的事件处理操作符,如表1所示。按照事件处理操作符的类型,分为基础操作、高级操作和窗口操作三类,可单独或组合使用来描述复杂事件的检测过程。

表1 项目管理事件处理操作符

类型	操作符	表达式	描述
基础操作	$\wedge$	$e_1 \wedge e_2$	$e_1$ 与 $e_2$ 都发生,不考虑其发生的先后顺序
	$\vee$	$e_1 \vee e_2$	$e_1$ 与 $e_2$ 至少有一个发生,不考虑发生的先后顺序
	$-$	$\bar{e}_1$	$e_1$ 没有发生
	Sequence	Sequence( $e_1, e_2$ )	$e_1$ 与 $e_2$ 先后发生
	$<$	$e_1 < e_2 \vee e_3$	复杂事件 $e_1$ 是由 $e_2$ 和 $e_3$ 聚合而成
高级操作	Average	Average( $e_1, e_2$ )	$e_1$ 与 $e_2$ 属性的平均值
	Sum	Sum( $e_1, e_2$ )	$e_1$ 与 $e_2$ 属性求和
	Max	Max( $e_1, e_2$ )	$e_1$ 与 $e_2$ 属性的最大值
	Min	Min( $e_1, e_2$ )	$e_1$ 与 $e_2$ 属性的最小值
	Count	Count( $e_1$ )	$e_1$ 在输入事件中发生的次数
窗口操作	Within	$e_1$ Within( $t_1, t_2$ )	$e_1$ 在 $t_1$ 至 $t_2$ 时刻发生
	Window	$e_1$ Window( $t$ )	$e_1$ 在时间段 $t$ 内发生
	Batch	$e_1$ Batch( $n$ )	$e_1$ 在事件窗口长度为 $n$ 范围内发生

表1中的事件处理操作符可以用来定义复杂事件的聚合规则,从而实现在事件流中实时获取有效信息的目标。例如,式(5)表示90 s内事件类 $T$ 的所有事件的平均值。

$$\text{Average}(C^T) \text{ within } (\Delta t) (\Delta t = 90 \text{ s}) \quad (5)$$

式(6)表示 $e_1, e_2$ 顺序发生,然后 $e_3, e_4$ 同时发生,并且 $e_3, e_4$ 的属性之和大于300。

$$\text{Sequence}(e_1, e_2, e_3 \wedge e_4) \text{ And } \text{Sum}(e_3, e_4) > 300 \quad (6)$$

### 2.4 事件处理语言

事件处理语言(Event Processing Language, EPL)是一种类SQL的语言,面向事件处理提供过滤、关联、约束、组合等操作,为复杂事件处理提供语言和语法支撑。通过事件处理语言中各类操作,可以实现从事件流中获取信息、事件聚合等,在复杂事件处理引擎中对于描述事件规则和定义筛选条件具有重要作用。事件处理语言支撑的保留字与数据库SQL语言类似,包括SELECT、FROM、WHERE、GROUP BY、HAVING、ORDER BY等。此外,事件处理语言通常还提供模式、操作符、函数和视图等功能。

一般的事件处理语言查询至少包含一个选择关键字和一个事件流。复杂的事件查询语句的WHERE关键字包含多种不同的查询条件,SELECT关键字描述需要查询的事件属性,FROM关键字描述事件流,WHERE关键字为满足事件输出要求设置查询条件,

GROUP BY 关键字将输出结果划分为多个分组,HAVING 关键字根据 GROUP BY 的限制条件定义事件的选择或丢弃,OUTPUT 关键字定义输出事件产生的频率,ORDER BY 关键字根据属性对所有输出事件排序。本文采用的事件处理语言的语法如下:

```
select select_list
from stream_def[ as name][ , stream_def[ asname]][ , ...]
[ where search_conditions]
[ group by grouping_expression_list]
[ having grouping_search_conditions]
[ output output_specification]
[ order by order_by_expression_list]
```

下面对本文提出的项目管理事件模型、事件操作符,以及本文采用的事件处理语言,进行举例说明。

**查询条件 1** 所有服务器操作系统连续 90 s 中内存的平均值。

根据项目管理事件模型,内存平均值的复杂事件记作  $c_e^{\text{Memory\_usage}}$ ,原子事件记作  $p_e^{\text{Memory\_usage}}$ ,利用平均值和时间操作符,内存平均值的复杂事件可定义为:

$$c_e^{\text{Memory\_usage}} < \text{Average}(p_e^{\text{Memory\_usage}}.A) \text{ within } (\Delta t)$$

$$(\Delta t = 90 \text{ s})$$

$$c_e^{\text{Memory\_usage}} < \frac{\sum_{i=1}^n p_{e_i}^{\text{Memory\_usage}}.A}{n} \Delta t$$

$$(1 \leq i \leq n, \Delta t = 90 \text{ s}) \quad (7)$$

根据本文采用的事件处理语言,可将上述内存平均值复杂事件表述为如下事件处理语句:

```
select avg(Memory_usage) from PrimitiveEvent.
win:time (90 sec)
```

**查询条件 2** 青年资助计划申报周期内项目完成提交的情况。

依据科技项目申报系统技术处理逻辑,申报用户登录申报系统提交项目申报材料后系统会执行申报书 PDF 文件合成,当 PDF 合成成功表示一个项目提交完成。将项目提交复杂事件记作  $c_e^{\text{Project\_submit\_finish}}$ ,利用顺序和时间操作符,项目提交复杂事件可定义为:

$$c_e^{\text{Project\_submit\_finish}} < \text{Sequence}(p_e^{\text{Project\_submit}}, p_e^{\text{Project\_file\_create}})$$

$$\text{and } (p_e^{\text{Project\_submit}}.P = p_e^{\text{Project\_file\_create}}.P =$$

$$'YouthAssistancePlan') \text{ and } (p_e^{\text{Project\_submit}}.A =$$

$$p_e^{\text{Project\_file\_create}}.A) \quad (8)$$

根据本文采用的事件处理语言,可将项目提交复杂事件表述为如下事件处理语句:

```
select * from PrimitiveEvent AS pe
[ every pe[i].S = 'Project_submit',
pe[j].S = 'Project_file_create']
```

Where pe[i].P = pe[j].P = 'Youth Assistance Plan' And pe[i].A = pe[j].A

本文采用上述事件处理语言,描述项目管理事件的处理规则,且应用在事件处理算法中,实现项目管理事件的表达。

## 2.5 事件处理算法

在处理海量事件实时处理问题上,文献[23]提出一种 LAIP 算法,使用自动状态机实例执行事件处理规则完成事件处理。本文在项目管理事件模型和事件处理语言的基础上,提出一种项目管理事件处理算法(Project Management Event Processing, PMEP),在复杂事件处理引擎中以规则实例的形式运行事件处理规则,实现对项目管理事件的实时处理。通过 PMEP 算法,可以适配项目管理事件模型对应的原子事件和复杂事件,支撑各类事件处理操作符,满足项目管理信息系统中各类事件的实时处理。事件处理规则使用事件处理语言定义,在 PMEP 算法中被转换成一个规则实例,实时处理输入事件。当事件处理规则产生匹配的复杂事件后,PMEP 算法不仅对事件进行存储,还创建一个通知消息,用于通知系统管理员。规则库是 PMEP 算法的事件处理规则来源,统一存储并维护事件处理规则,为 PMEP 算法提供事件处理规则读取服务。系统管理员管理规则库,拥有创建、修改、删除、浏览规则的权限,负责配置并查看每一条规则的启停状态。

### 算法 1 PMEP 算法

```
Load_EPL_Rules();
WHILE (Queue_Input.size > 0)
  PrimitiveEvent pe = Queue_Input.shift();
  IF (pe != NULL)
    FOR (i = 1; i <= Count_EPL_Rules; i++)
      ComplexEvent ce;
      IF (Rule[i].enable == True)
        ce = Rule[i].excute(pe);
      IF (ce != null)
        Action(ce);
      EventStorage.add(pe, ce);
```

PMEP 算法执行前,初始化阶段加载所有项目管理事件处理规则,每个规则都是一个实例,在初始状态等待输入事件。所有输入事件都被存在队列 Queue\_Input 中,PMEP 算法循环检查原始事件队列 Queue\_Input 的大小,当存在未处理事件时对原始事件进行处理。创建原子事件对象  $p_e$ ,初始化为 shift() 函数在队列 Queue\_Input 中取出队首事件,当  $p_e$  不为空时,遍历每个事件处理规则实例对  $p_e$  进行处理。在每个规则实例中,当

规则实例  $\text{Rule}[i]$  可用时,规则实例  $\text{Rule}[i]$  处理  $p_e$ ,从而产生复杂事件  $c_e$ 。如果  $c_e$  不为空,则表示  $c_e$  为有效复杂事件,根据  $c_e$  创建通知消息,同时在事件记录中存储  $p_e$  和  $c_e$  的信息。PMEP 算法的详细处理过程如图 3 所示。

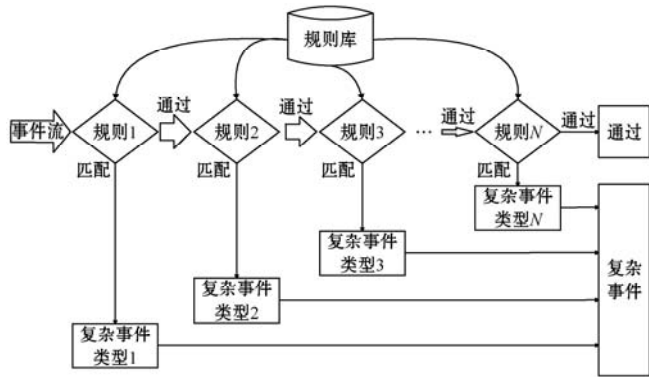


图3 PME P算法处理过程

### 3 实验结果

为了验证项目管理事件模型及 PME P 算法,检验复杂事件处理技术应用在项目管理信息系统中的预期成效,本文设计一个基于 Java 语言的事件处理原型系统,采用开源 CEP 引擎 Esper<sup>[7]</sup> 作为原型系统的事件处理引擎。在原型系统中,开发一个事件模拟器,可以根据配置产生各类原子事件,为原型系统提供海量输入事件。实验设备为一台 Acer 笔记本,其配置为 8 GB 内存、2.50 GHz Intel Core i7-6500U CPU,操作系统为 Windows 7 64 位。

本文实验有两个目标,首先是证明面向项目管理的事件模型、事件操作符与事件处理算法的可用性,其次是验证 PME P 算法的事件实时处理能力和产生复杂事件。实验分为两个部分,第一部分通过配置不同数量的事件处理规则,比较处理相同数量原子事件时,PME P 的时间消耗,通过事件处理时间验证算法执行效率。第二部分实验在第一部分的事件处理规则基础上,比较处理相同数量原子事件时产生的复杂事件数量。

第一部分实验结果如图 4 所示。在原型系统中,配置 10 个事件处理规则时,输入原子事件数量从 10 000 增长到 100 000,事件处理时间从 0.16 s 增长到 1.45 s;当配置 20 个事件处理规则时,输入原子事件从 10 000 增长到 100 000,事件处理时间将从 0.43 s 增长到 2.13 s。因此,事件处理规则数量直接影响事件处理算法的效率,尤其是当事件处理规则中包含逻辑操作和时间操作时,对事件处理算法效率的影响更大。

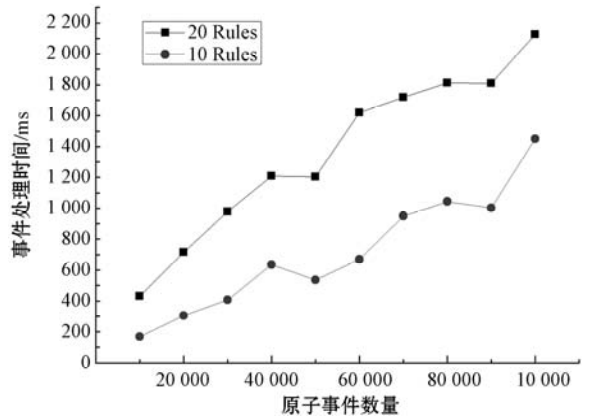


图4 事件处理时间消耗比较

第二部分实验结果如图 5 所示。在原型系统中,配置 10 个事件处理规则时,输入原子事件数量从 10 000 增长到 100 000,产生的复杂事件从 70 增长到 703;当配置 20 个事件处理规则时,输入原子事件从 10 000 增长到 100 000,产生的复杂事件则从 126 个增长到 1 270 个。这个实验可以看出,随着输入事件的增长,产生复杂事件的数量随之增长,同时,当事件处理规则增加时,产生复杂事件的数量也会相应增长。

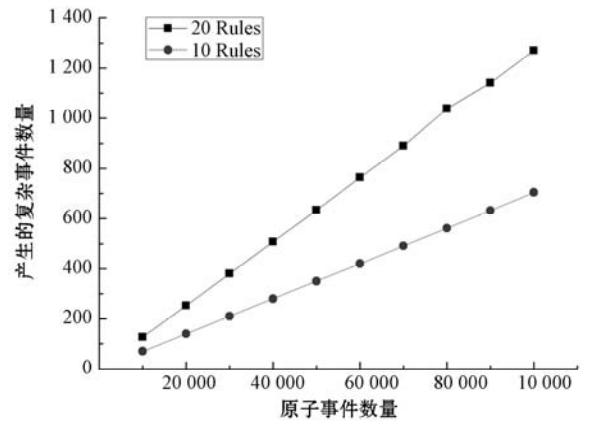


图5 产生复杂事件数量比较

### 4 结语

针对科技管理信息系统大量事件的实时处理问题,提出一种面向项目管理信息系统的复杂事件处理方法,即项目管理事件处理算法(PME P)。首先,定义项目管理业务模型和项目管理事件模型,采用事件处理操作符描述事件之间的关联关系;然后,通过 Esper 的 EPL 语法规则,定义项目管理复杂事件,形成事件处理规则;最后,提出 PME P 算法,在复杂事件处理引擎中以规则实例的形式运行事件处理规则,实现对项目管理事件的实时处理。实验结果表明,复杂事件处理技术具有较强的事件处理能力,可应用在项目管理信息系统中,有利于提升各类项目管理事件的实时处理能力。

## 参 考 文 献

- [1] 国务院. 国务院关于改进加强中央财政科研项目和资金管理的若干意见[EB/OL]. (2014-03-12). [http://www.gov.cn/zhengce/content/2014-03/12/content\\_8711.htm](http://www.gov.cn/zhengce/content/2014-03/12/content_8711.htm).
- [2] 国务院. 国务院印发关于深化中央财政科技计划(专项、基金等)管理改革方案的通知[EB/OL]. (2015-01-12). [http://www.gov.cn/zhengce/content/2015-01/12/content\\_9383.htm](http://www.gov.cn/zhengce/content/2015-01/12/content_9383.htm).
- [3] 国务院. 国务院关于优化科研管理提升科研绩效若干措施的通知[EB/OL]. (2018-07-24). [http://www.gov.cn/zhengce/content/2018-07/24/content\\_5308787.htm](http://www.gov.cn/zhengce/content/2018-07/24/content_5308787.htm).
- [4] Alwadain A, Fiel E, Korthaus A, et al. Empirical insights into the development of a service-oriented enterprise architecture[J]. *Data and Knowledge Engineering*, 2016, 105: 39-52.
- [5] Gao Q, Xing W. A design on single sign-on system of enterprise information portal based on security domain[J]. *Intelligent Computer and Applications*, 2013, 3(5): 5-13.
- [6] Bruns R, Dunkel J, Offel N. Learning of complex event processing rules with genetic programming[J]. *Expert Systems with Applications*, 2019, 129: 186-199.
- [7] Espertech. Esper[DB/OL]. [2021-03-01]. <http://www.espertech.com>.
- [8] Wang D, Zhou M, Ali S, et al. A novel engine: Complex event processing with LAIPE[J]. *Journal of Computers*, 2017, 12(4): 317-334.
- [9] Shin Y, Yoon S, Trirat P, et al. CEP-Wizard: Automatic deployment of distributed complex event processing [C]//2019 IEEE 35th International Conference on Data Engineering, 2019: 2004-2007.
- [10] Zappia I, Paganelli F, Parlanti D. A lightweight and extensible complex event processing system for sense and respond applications[J]. *Expert Systems with Applications*, 2012, 39(12): 10408-10419.
- [11] Ma M, Wang P, Chu C, et al. Efficient multi-pattern event processing over high-speed train data streams[J]. *IEEE Internet of Things Journal*, 2015, 2(4): 295-309.
- [12] Zhu W, Sun S. Supporting interval time complex events processing in an RFID system[J]. *Procedia Computer Science*, 2019, 147: 324-330.
- [13] Arbia S, Alaoui N, Bennani S. Functional semantic complex event processing model for massive open online courses[C]//2017 16th International Conference on Information Technology Based Higher Education and Training, 2017: 1-4.
- [14] Lu T, Zha X, Zhao X. Multi-stage monitoring of abnormal situation based on complex event processing[J]. *Procedia Computer Science*, 2016, 96: 1361-1370.
- [15] Higashino W, Capretz M, Bittencourt L. CEPsim: Modeling and simulation of complex event processing systems in cloud environments [J]. *Future Generation Computer Systems*, 2016, 65: 122-139.
- [16] Morozov V, Kalnichenko O, Liubyma I. Proactive project management for development of distributed information systems [C]//2017 4th International Scientific-Practical Conference Problems of Infocommunications Science and Technology, 2017: 25-28.
- [17] 胡少华, 王卓昊, 宋学鹏, 等. 国家科技管理信息系统建设与应用[M]//2017 中国科研信息化蓝皮书. 北京: 电子工业出版社, 2017: 125-138.
- [18] 李东, 郝艳妮, 彭升辉, 等. 国家自然科学基金委员会网络安全现状与展望[J]. *网络与信息安全学报*, 2022, 8(6): 92-101.
- [19] 及俊川. 中国科学院资源规划系统发展规律的探究[J]. *科研信息化技术与应用*, 2013(2): 3-8.
- [20] Luckham D C, Frasca B. Complex event processing in distributed systems[D]. Stanford University, 1998.
- [21] 中华人民共和国科学技术部. 中华人民共和国科学技术部官网[EB/OL]. (2015-10-15). <http://www.most.gov.cn/kjjh/>.
- [22] Taghavi M, Patel A, Patel A, et al. Web base project management system for development of ict project outsourced by Iranian government [C]//2011 IEEE Conference on Open System, 2011: 267-272.
- [23] Wang D, Zhou M, Ali S, et al. A novel complex event processing engine for intelligent data analysis in integrated information systems[J]. *International Journal of Distributed Sensor Networks*, 2016, 12(3): 1-14.

## (上接第390页)

- [14] 邵尹池, 穆云飞, 余晓丹, 等. "车-路-网"模式下电动汽车充电负荷时空预测及其对配电网潮流的影响[J]. *中国电机工程学报*, 2017, 18(10): 102-108.
- [15] Low S H. Convex relaxation of optimal power flow—part II: Exactness [J]. *IEEE Transactions on Control of Network Systems*, 2014, 1(2): 177-189.
- [16] Sutton R S, Barto A G. Reinforcement learning: An introduction[M]. 2nd ed. Cambridge: MIT Press, 2018.
- [17] Wei W, Wu L, Wang J H, et al. Network equilibrium of coupled transportation and power distribution systems [J]. *IEEE Transactions on Smart Grid*, 2018, 9(6): 6764-6779.
- [18] 李晋津, 孙二鑫, 陈宗娟. 自动驾驶车辆循迹路径生成算法研究[J]. *汽车实用技术*, 2020, 45(18): 16-21.
- [19] Silvr D, Lever G, Heess N, et al. Deterministic policy gradient algorithms [C]//31st International Conference on Machine Learning, 2014: 387-395.
- [20] Ruelens F, Claessens B J, Vandael S, et al. Residential demand response of thermostatically controlled loads using batch reinforcement learning[J]. *IEEE Transactions on Smart Grid*, 2017, 8(5): 2149-2159.