**学 号：ZY1507210**





硕 士 文 献 综 述

“复杂事件处理及其关键技术研究文献综述”

作者姓名 文波

学科专业 工业与制造系统工程

指导教师 杨建军 教授

培养院系 机械工程及自动化学院

2016 年11月

要求

1. 文献综述题目必须是与论文选题相关方向的理论、技术、方法等，应该体现论文题目的核心词；一般不走过25字,可不与开题报告论文题目一模一样。
2. 每位学生在自己的学位论文开题前，必须至少阅读30篇以上与本人论文选题相关的技术文献，并满足其中有20篇以上近10年内的论文或技术报告，以及有15篇以上的外文论文资料的要求 ；
3. 文献综述报告字数为8000～10000字，通常10页左右。
4. 学生可到北航图书馆、国家图书馆或相关专业网站查阅有关文献。写文献综述一般经过以下几个阶段：即论文选题，搜集阅读文献资料、拟定提纲（包括归纳、整理、分析）和成文。

**【摘要】**

【摘要】不超过300字

XXXXXX

随着事件驱动架构的出现，商业活动的自动化以及信息监控系统的广泛应用，许多有关合法性、契约性以及可操作性的规则被添加到分布式系统上，这就导致了越来越多的事件的产生，同样也增加了事件的自动化管理和处理需求。复杂事件处理技术的出现很好地迎合了这样的需求。每当有事件出现时，复杂事件处理引擎自动地从这些低层事件获取高层的知识来识别系统中所有的事件。

复杂事件处理引擎的两个核心技术就是规则定义和事件检测。规则制定的难点在于参数的确定和适应不断变化的领域需求。而当前的解决方案都依赖领域专家不断地对规则做出调整，这使得规则制定成为了复杂事件处理技术应用的一大障碍。事件检测技术的关键在于准确以及快速地检测出事件。为了达到这个目的，许多检测模型以及相应的扩展模型已经应用到复杂事件检测中。

**【关键词】**

【摘要】不超过五个关键词

复杂事件、规则定义、事件检测、动作触发

# 一、阅读文献概述

【内容要求】对所述研究方向阅读文献的概述，包括对什么研究方向的那些分支搜集阅读那些文献资料、如何拟定文献综述提纲进行文献综述。

【写作要求和篇幅量】力求给出对研究方向的那些分支从那些方面对那些理论、方法、技术和系统，以及关键技术问题进行了那些包括归纳、整理和分析工作，论述资料的来源、完整性。广泛性、时效性和可信度等，通常这部分内容大半页左右。

【常见问题】不能针对研究方向相对系统地、全面地收集和分析文献资料，或没有搜集到足够，甚至根本没有搜集到对论文选题有直接或间接指导作用的情报资料。

XXXX

Gartner在2003年提出了事件驱动架构，指出它是一种有效的基于事件的系统应用开发方法[28]。EDA是一种软件架构类型，在EDA中各个组件之间是相对独立的，互相之间以事件作为消息进行通信和协作，是一种基于推（push）模式的异步通信模型。当有事件触发时基于EDA的系统会快速且自动进行响应，将事件消息推送给相关应用组件，因此EDA非常适合于异步的业务流和信息流处理，是一种适应性很强的实时处理技术框架。

复杂事件处理技术是由美国斯坦福大学D.Luckham所提出，一种以事件驱动为基础的运算模式[6]。CEP是一种面向实时数据处理的新技术，通过将数据信息抽象成不同类型的事件，并利用时间顺序、逻辑关系等语义建立事件关系模式库，进一步采用选择、过滤和聚合等技术对蕴含在事件间的各种逻辑关系进行模式匹配和语义推理，生成具有一定抽象层次，满足业务需求的高级事件。CEP技术被广泛地集成于事件驱动架构系统之中，用于构建和管理信息系统。

国内外许多学者围绕事件驱动以及CEP技术的理论和应用展开了深入研究。Rosenblum[29]通过分析大规模事件通知服务系统的特征和应用需求，提出了一种网络环境下EDA架构的一般设计思路，该设计思路中采用了对象模型、事件模型、命名模型等七个通用模型进行分析和设计，为事件驱动体系结构的系统设计打下了重要基础。加州大学伯克利分校的Eugene Wu等[30]基于CEP技术设计了SASE系统来满足 RFID 技术在监控领域的应用要求。Paschke等[31]提出了一种事件处理的一般应用参考架构，重点分析了系统中的共同要素以及各要素之间的逻辑关系。华南理工大学林侃[32]提出了一种基于XML的RFID复杂事件描述语言ECEDL（Extended Complex Event Description Language）来对高层的业务逻辑进行描述，并运用自动机模型和匹配树模型相结合的方式设计了一种解析ECEDL语言的处理引擎，实现复杂事件的实时检测和发布。

面向制造业，伴随着RFID等物联网技术的使用，事件驱动和CEP也逐渐被运用于RFID制造过程监控等技术应用场景。臧传真等[33]为了有效处理RFID等产生的大量数据，提出基于智能物件的制造企业复杂事件处理机制，详细描述了事件、事件操作符等的语法定义，给出了系统实现体系结构及检测算法实现，并在企业实际应用中证明了方法的有效性。黄毅等[34]为了解决RFID生产监控中的信息实时性和逻辑复杂性问题，提出一种基于复杂事件处理的RFID事件处理系统的设计与实现方法，并开发了原型系统，在应用中表现良好。杨屹等[35]针对云制造应用需要，结合复杂事件处理技术和事件驱动技术，利用开源软件处理制造过程中的RFID复杂事件，并详细给出了轴加工的实例分析。上海交通大学许斌[36]为解决制造企业中MES与PCS信息交换过程中的不一致性、扩展性不足以及系统异构等问题，提出了基于事件驱动架构的信息交换平台框架，并结合实际应用需求对平台进行了实施和验证。

随着研究的深入，基于事件驱动的复杂事件处理技术的应用也逐渐深入到许多领域，已成为物流、金融、制造业等众多应用领域中数据实时分析的关键技术[37]。面向物联网使能的智能制造，基于事件驱动的复杂事件处理技术可以有效地实现对大量多维度系统状态数据进行实时分析处理，并且在此基础上对处理结果给出快速响应，驱动业务流程。

# 二、研究方向的现状与发展趋势

## 研究方向主要情况

【内容要求】研究方向的若干（3～5个）分支

【写作要求和篇幅量】力求给出能够反映研究方向的全貌的重要（3～5个）分支，如研究方向所涉及的方面太多，可给出一个全貌，并重点论述重要（3～5个）分支。通常这部分内容大半页左右。

【常见问题】罗列、空洞的论述，导致与论文选题的脱节。

XXXX

基于事件驱动的复杂事件处理研究现状

CEP 系统参考架构

CEP 事件查询

## 研究的现状

【内容要求】每个分支的理论/方法/方案/技术研究的现状。

【写作要求和篇幅量】力求概要论述研究方向每个分支的理论/方法/方案/技术，明确理论/方法/方案/技术的内涵，代表性项目、机构或代表人，主要热点和难点问题，以及进展和前途。通常在2页左右。

【常见问题】主观和武断的评论，以及缺乏分析和客观的评论。

XXXX

事件处理的发展主要可分为三个阶段：数据库处理、数据流处理和复杂事件处理。

最初的时候，对于事件的处理通常采用的做法是先将事件转化为数据并存储到数据库中，然后从数据库中读取数据并执行查询来完成，这种方式处理事件的效率非常低【l 4I。一般将这种处理事件的系统称为流数据库，它们以数据存储为中心，提供有丰富的查询语言，但是不适合处理事件序列模式。比较典型的流式数据库有：以开源数据库PostgreSQL为基础并且进行体系结构改造以支持连续查询的TelegraphCQ[ISl；以及后来的提供历史查询功能的Truviso。但总体来说，基于数据库的事件处理的执行速度都不够理想。

采用数据流：处理技术的比较经典的系统包括Aurora[161、船嬲旧以及STREAM[is】等。它们都采用了发布／订阅模型，主要的缺点是查询语言的表现力有限，只能执行简单的选择操作；但它们的优势在于处理速度快。在数据流处理系统中，当数据流过系统时，通过简单的选择过滤而形成一个新的数据流。数据流处理系统通常是分布式的和高度并行的。尽管数据流处理系统查询的效率很高，但它们不能处理多个事件模式的查询，因此不适合处理复杂事件。

因此，复杂事件处理系统应运而生，复杂事件处理一般都采用非确定性有限自动机(Nondeterministic Finite Automata，NFA)变体模型来处理事件。典型的复杂事件处理系统主要有SASE[191，Cayuga例以及E驴er[211等。

SASE系统采用了一种基于本地序列操作符以及管道查询的数据流模型，使用关系运算符来定义随后到来的序列。跗舾采用了非确定性有限自动机来获取序列事件，在自动机执行期间使用分区主动实例栈算法来处理等值测试，实现了较好的可扩展性。与许多其他系统不同，SASE不仅会报告用户感兴趣的查询结果，而且会报告匹配此查询的所有事件，这在很大程度上增加了查询处理的复杂度。跗。她的主要局限性在于不能处理层状结构的复杂事件类型，也就是一个查询的结果不能用作另一个查询的输入。

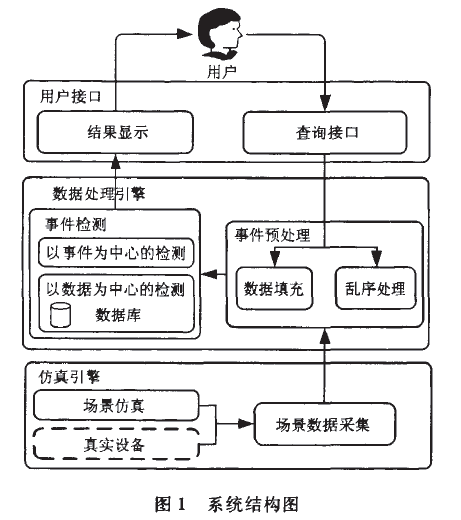
康奈尔开发的Cayuga则弥补了这个不足。由于Cayuga使用了传统的发布-订阅技术，它支持大量并发的订阅事件，并且提供了良好的可扩展性。Cayuga引擎使用单线程读取数据和利用自动机处理数据。自动机允许对输入数据进行存储，这使得新的输入可以与先前存储的事件做比较。Cayuga使用了类似流水线的处理模型，该模型需要每个查询结果实时输出给下个处理过程使用。通常事件检测的时间也就是最近接收的事件的时间。当处理查询序列时，可能出现多个查询期待同一个事件的情况，并且其中一些查询相比于其他查询能产生更有意义的输出。为了处理这样的问题，Cayuga采用了优先队列。Cayuga中还采用了查询优化技术，将多个拥有相同时间戳的具有等价状态的事件一起处理【221。然而，由于它的内核是单线程的，这些优化技术并没有显著提高Cayuga的性能。

Esper同样提供了复杂的查询匹配技术，包括时间窗El，不同事件流的连接，过滤以及排序等等。此外，它同样能够检测不相关事件序列和模式。与Cayuga不同的是，它使用多线程来处理事件。Esper的内部组成主要依赖于状态机和如砌网络，事件对象只关心数据的更改情况。Esper是一款开源软件，因此它的应用相对比较广泛。

### 国内：

谷峪, 胡小龙, 李传文,等. REvent:一种面向RFID应用的复杂事件处理系统[J]. 计算机研究与发展, 2010, 47(z1). 谷峪的邮件：[guyu@ise.neu.edu.cn](mailto:guyu@ise.neu.edu.cn)（Ps给他发了邮件，还没有回复。）

REvent的系统框架图

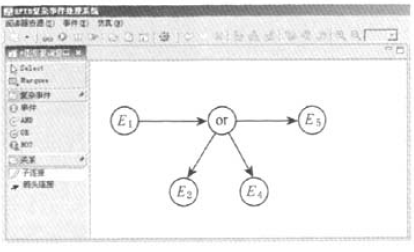


涉及到的关键技术：

**数据预处理技术**

**事件检测技术**

评语：从系统界面上能看出他们是用的java开发的一个小程序，从上面的系统结构图我们也可以看出他们的处理系统主要还是以人为中心，智能化处理程度还是偏低。



陈皓, 李瑜, 虎嵩林,等. 基于S4框架的并行复杂事件处理系统[J]. 通信学报, 2012(s1):165-169.

**关键技术：**

**基于操作符的负载分流优化方法**：当面对高速数据流时，如果能将负载均匀地分流到集群的各个节点，并行地处理事件流，则可以增加系统的吞吐量。然而复杂事件处理的特殊性在于事件往往不是孤立地存在，在处理中需要根据事件流上下文或其他事件流进行决策，单纯地拆分数据流到不同节点，会影响事件间的关系，导致处理出错。

**系统概述：**

S4（simple scalable streaming system）是一个去中心的、分布式的、可扩展的流式处理系统。类似于Storm 的消息分组策略，S4 框架通过对流事件的关键字进行散列计算，得到处理事件的节点编号，再将事件发送到对应节点进行处理。通过修改S4框架的事件分流机制，基于操作符的不同，使用不同的分流策略，实现了并行的通用CEP 系统。（从文章中未能发现他们所做的开发内容）

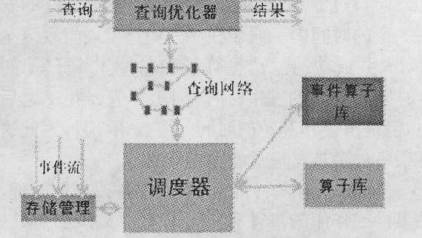
魏永超, 陈立军. 数据流上复杂事件处理系统Eagle的设计与实现[C]// 中国数据库学术会议. 2008.

涉及到的关键技术：

事件代数

事件操作符

聚集操作符  
系统架构：



**实验环境：**Inter酷睿E4400双核 2.00GHz Windows XP

根据文章资料可以推断数据库是用的Oracle，主要使用java语言开发的（和我实验室的应该差不多）。

### 国外：

Brenna L, Demers A, Gehrke J, et al. Cayuga: a high-performance event processing engine[C]// Proceedings of the 2007 ACM SIGMOD international conference on Management of data. ACM, 2007:1100-1102.

不断增加的数据量作为高速事件流到达。库存报价数据，网络流量数据，在传感器网络中收集的数据，RFID流只是这种流的几个示例。此外，随着网络上的动态内容（博客，新闻，拍卖）的数量迅速增长，网络正在成为许多有趣和复杂的事件流的来源。诸如医疗保健，环境监测，供应链管理和合规性检查等新兴企业级应用以及个性化新闻馈送和信息聚合等消费者应用都需要过滤，关联和聚合各个事件。所得到的复杂事件查询超出了发布/订阅，因为查询涉及多个事件之间的相关，而不是对单个事件的简单谓词。在康奈尔，我们建立了Cayuga系统，一个用于复杂事件处理的高性能系统[1,2]。 Cayuga将用于组成状态查询的简单查询语言与基于具有缓冲区的非确定性有状态自动机的可扩展查询处理引擎组合。 Cayuga的一个重要特征是它不仅可以扩展流中事件的到达速率，而且还可以扩展查询数量;可扩展性的后一维对于消费者应用尤其重要，其中数百万用户可以具有向系统注册的查询。在本演示中，我们将展示Cayuga的应用程序，用于监控来自Web Feed聚合器的事件流，用于个性化新闻聚合和通知。我们首先给出Cayuga系统的高级概述，介绍一些我们将在演示中显示的Cayuga查询（第2节）。然后我们详细描述实际演示（第3节）。

在Cayuga中，每个事件流都有一个固定的关系模式，并且流中的事件被视为关系元组。

每个事件具有两个时间戳，开始时间和检测时间，对事件可以具有非零但有限的持续时间的事实进行建模。 事件按其检测时间的顺序串行化; 具有相同检测时间的事件被认为同时发生，并且不能保证它们的序列化顺序。 几个有趣的理论问题导致了这种设计。

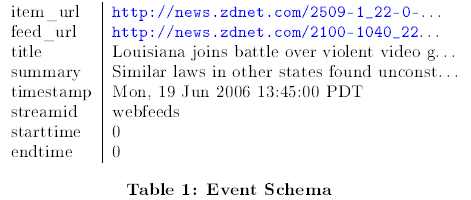
Cayuga查询语言是从事件代数派生的[1]; 它是代数运算符的简单映射一个SQL语法，在精神上类似于复杂事件SASE [4]中的语言。 每个查询具有以下形式：

SELECT <> attributes

FROM <> algebra\_expression

PUBLISH <> output\_stream

SELECT子句指定输出流模式中的属性，FROM子句指定Cayuga事件模式，PUBLISH子句给出输出流的名称。 事件模式可以用三个不同的运算符构建。 FILTER {？}运算符从满足谓词？的输入流中选择那些事件。



演示查询1：在演示中，我们将通过以下查询来说明FILTER运算符，其中包含由整个流中的Google新闻发布的所有新闻项目：webfeeds？ 具有表1中所示的模式。

SELECT \* FROM

FILTER {feed\_url ='http：//news.google.com/'}（webfeeds）

PUBLISH google\_news\_items

一个强大的结构允许我们随时间关联事件是排序运算符NEXT {？}。 当应用于两个输入流S1和S2作为S1 NEXT {α} S2时，运算符将来自S1的每个事件与S2中满足谓词α的下一个事件组合。 并且发生在S1中的事件的检测时间之后。

演示查询2.此查询包含某些网站发布的所有新闻项目，其后是一篇关于Google新闻的文章。 请注意，此查询将Query 1的输出作为其输入流之一。 用户定义函数包含执行子字符串匹配; $ 1和$ 2指的是NEXT运算符的两个输入流。

SELECT $ 2.summary，$ 1.item\_url

FROM（webfeeds）

NEXT {contains（$ 2.item\_url，$ 1.item\_url）= 1}（google\_news\_items）PUBLISH re？ed\_by\_google\_news

虽然NEXT允许我们关联两个事件，但是在许多情况下，我们需要遍历已知数量的事件，直到满足停止条件。 此功能由FOLD操作员提供。 直观地，FOLD是NEXT运算符的泛化，因为它查找包括两个或更多事件的模式。 我们将使用下一个例子来描述FOLD。

演示查询3.此查询发送通知

每当一个iPod在新闻中受欢迎（即，有

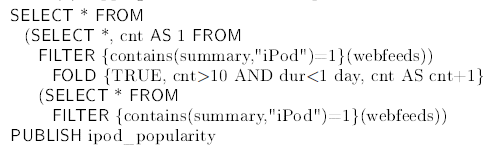
至少十篇文章在一段时间谈论iPod

持续时间）。 FOLD运算符包含三个表达式：

（1）迭代的条件

（2）迭代的停止条件

（3）迭代步骤之间的映射。



Cayuga在时代处理事件;在历元期间，处理具有相同检测时间的所有事件。查询被实现为具有如下工作的缓冲器和自循环的非确定有限状态自动机。自动机中的每个状态都分配有固定的关系模式。每个边缘，例如在状态P和Q之间，由三元组hS，α，fi标记，其中S标识输入流; ？是模式（P）×模式（S）的谓词;和f，模式映射是将模式（P）×模式（S）转换为模式（Q）的函数。 NFA操作如下。假设NFA实例处于具有存储数据x的状态P（注意x符合模式（P））。让事件e在流S上到达，使得满足α（x，e）。然后，机器非确定地转变到状态Q，并且存储的数据变为f（x，e）。 Cayuga支持重新订阅，一个类似于关系数据库系统中的查询计划的概念：来自一个查询的输出事件流可以用作一个或多个其他查询的输入流。重新订阅启用非常复杂的事件模式查询，并且它显着扩展了查询语言的表达性。该系统的更多细节可以在我们最近的出版物中找到[1,2]。

**3演示概要**

在本节中，我们提供了如何演示上面显示的查询的概述，以Demo Query 2为运行示例。 我们的演示包括以下三个组件：（i）基于Web的前端，供用户输入查询; （ii）轨迹可视化器，其显示Cayuga引擎的内部状态和基于自动机的查询表示; 和（iii）显示查询评价结果的输出可视化器。 在演示期间，我们将对来自Web Feed聚合器的事件流运行查询。 让我们现在通过查询2的执行来呈现上述每个组件。该查询是针对包含来自于2006年6月和10月之间记录的418个频道的225365个源项目的事件流来处理的。

**4 提交Cayuga查询**

基于Web的前端正在自定义的Python Web服务器上运行，基于AJAX的控件用于异步通信和浏览器中的用户友好界面。 Cayuga有一个基于Web的前端，用户可以输入持久性查询并使用正在运行的Cayuga引擎注册它们。为了方便起见，用户可以从下拉菜单中为所述查询语言选择预定义的模板。然后，他们可以根据自己的需要修改它们，或从头开始写一个。由于Cayuga查询可以轻松跨越多行，因此我们提供的多行输入框比通常在基于Web的查询界面中看到的多。图1中选择进行编辑的查询是显示查询2的查询前端的屏幕截图。请注意，此版本的查询是内联的，没有重新订阅。在提交查询时，查询将被编译并且在Cayuga之外。用户将收到一条带有查询ID或错误消息的消息，查询未编译，并且必须在更改后重新提交。用户可以选择批处理多个查询，然后再按EXECUTE向Cayuga注册查询。可以通过按清除来清除批次。

跟踪Visualizer

Cayuga可以输出其内部状态在事件之间如何变化的连续跟踪。此跟踪将写入文件，其中包含查询自动机的描述，后跟事件和状态实例。我们的可视化器读取跟踪文件，并使用基于Java Swing的GUI来显示事件如何匹配查询自动机中的谓词。

为了精确地查看加载的查询如何对输入流中的事件起作用，用户可以选择向前播放，在任何给定点暂停，或者逐步进行跟踪。这允许用户详细调查任何单个事件如何影响查询自动机的内部状态。在图2和图3中的可视化器的工具栏上看到的控件是（从左到右的顺序），用于放大和缩小自动机，调整与工具栏的垂直距离，以及用于回放。输入和输出流在动画中被命名和给定不同的颜色。在任何时候，如果用户点击自动机中的一个状态，将弹出一个文本框，以显示处于此状态的实例。单击边缘将显示该边缘上的谓词。图2示出了具有用其谓词注释的过渡边缘的查询2的自动机。在图3中，自动机已经处理了一些事件。流到目前为止产生了状态1的7个实例，状态2的4个实例。右边的文本框显示状态2的内容，向下滚动以查看最后一个实例达到这个状态。从文字中可以看出，http://news.zdnet.com在3月12日（星期三）06:00发布的故事是由http://news.google.com在当天的13:00链接到的。

**输出**

当生成见证事件（即查询结果）时，Cayuga当前将其转储到Web前端正在拖尾的文件中。 然后将这些文件放入一个BUYER中，其中频繁轮询AJAX函数可以获得它们并且在终端用户的web浏览器的HTML页面中显示，并按查询排序。 图4是屏幕截图，显示了查询2出现的第一个两个见证。

**结论**

Cayuga是一个成熟的事件处理系统，有趣处理模型。 我们相信它的功能演示将对数据库社区感兴趣。

**事件处理平台有很多：**

* 建立，被广泛使用的商用产品，如IBM的ODM和InfoSphere Streams的，SAP的ESP（以前的Sybase埃勒瑞），Software AG的Apama，TIBCO的业务活动和StreamBase，和其他人，是多方面的开发和运行软件套件，包括适配器与整合事件源，开发和测试工具，仪表板和报警工具和管理工具。
* 其他事件处理平台相结合，与功能，如查询，报表，互动分析，警报或​​关键绩效指标的工具，这是在运行智能应用专门针对。例子包括FeedZai脉冲，的SQLstream S-Server和Vitria Technology的作战情报。
* 新兴的DSCP，如Apache Samza，Spark和风暴，是没有全面的本地CEP分析功能和相关配件的通用平台。他们是高度可扩展的和可扩展的，因此开发人员可以添加逻辑来处理多种流处理应用程序。
* 这些所描述的DSCP，与蛾/ Nesper和RedHat的Drools的融合/ JBoss企业BRMS（两者均为CEP系统）一起，是开源产品。
* DataTorrent的RTS和IBM的InfoSphere流被划分为的DSCP或CEP系统，根据上下文和作者的观点。
* 有些事件处理平台捆绑在一起的规则引擎。例子包括IBM的ODM，RedHat的Drools的融合/ JBoss企业BRMS和TIBCO业务活动。

CEP使用量迅速增长，因为CEP，从技术意义上说，是让在实时或近实时事件流信息的唯一途径。该系统具有处理事件数据或多或少到达使得相应的操作可以快速地作出。在这篇文章中列出的子系统是通用开发和运行时工具，用于开发人员构建自定义，事件处理应用程序，而无需重新实现核心算法处理事件流。然而，这些子系统是唯一的几种方式来获得的CEP的功能之一，事实上，该实施CEP逻辑大多数应用程序不使用这些子系统。

最CEP是作为较大产品的一部分获得。公司收购打包应用程序或订阅已在幕后嵌入式CEP一个SaaS服务。公司正在购买这种情况，要求事件处理的溶液中，它可能没有意识到正在使用的CEP。例如，供应链可视化产品; 安全性信息，甚至管理（SIEM）产品; 某些类型的欺诈检测和治理，风险和法规遵从（GRC）的产品; 系统和网络监控系统; 业务活动监控（BAM）工具; 和许多其他类别的软件实现CEP逻辑的一些或大或小数额。在少数情况下，这些产品或SaaS产品的开发人员已经利用上面列出，以减少他们必须编写的代码量的通用事件处理平台。但在大多数情况下，开发人员实现的在新的代码事件处理算法的一个专门的子集，以满足他们的应用程序的目的。

一些用户企业都写有CEP逻辑的自定义应用程序，而不是利用一个现成的现成事件处理平台。这是20世纪90年代和21世纪初尤为常见上面列出的子系统广泛使用之前，一些开发商仍然选择写性能或成本的原因自己CEP逻辑。例如，大型银行及相关金融服务公司已经做了前端办公系统的资本市场有自己的嵌入式CEP逻辑交易。

每个公司都有它已经使用了幕后CEP实现一定量的应用程序或SaaS解决方案的地方。越来越多的公司需要的那种这里列出来支持高要求，高吞吐量，低延迟的应用中，事件处理逻辑是主要的业务问题，事件处理子系统。在这两种情况下，你越了解CEP，更好的将您的应用程序体系结构。

## 关键技术问题

【内容要求】该部分重点分析和罗列研究方向的若干分支有那些关键问题，关键问题己解决的程度与尚待解决的难点，关键问题解决的方向等，以及有重要参考价值的整体解决方案，如系统架构，模型框架，体系框架等。并给出解决这些难点的解决方案并分析其可行性。要解决你选题和解决方案的依据和参考系的问题。

【写作要求和篇幅量】该部分的技术性较强，论述要准确，技术概念要清晰。关键技术主要指技术层面上存在的、研究人员致力解决的热点和难点问题。论述时要用尽量采用技术性的语言、概念和有关术语。这部分内容是为你在开题报告中所选择要解决的关键技术问题的论据，并在更大的范围内说明关键技术的背景。通常篇幅在4页左右。

【常见问题】不是从研究方向及其分枝的存在问题角度，而不仅局限自己要解决的问题。实际上，自己选择要解决的问题可能只是这些关键问题的一个或一部分。

XXXX

## 未来发展的趋势

【内容要求】论述对研究方向的若干分支及其关键问题下一步的研究走向。

【写作要求和篇幅量】主要表现你对研究方向的若干分支及其关键问题下一步的研究走向的了解和情报资料掌握情况，是为论述你开题报告中总体技术方案和关键技术解决的先进性服务的。通常篇幅在一页左右。

【常见问题】情报资料掌握不够，理解不透彻，不能很好的归纳出研究方向的若干分支及其关键问题下一步的研究走向。

XXXXXXX

融合大数据处理

CEP 的主要思想如下：首先从大量数据中抽取原始事件；其次根据特定的规则，使用事件关联和事件聚合操作生成复合的业务事件；再对原始或复合事件进行事件处理，以获得它们的时间、因果、层次和其他的语义关系；最后响应可操作的业务信息。

事件处理语言支持事件模式（事件定义、选择和消费政策）的规范以及事件处理的规则。

CEP 通常用于处理大量事件流数据且根据事件的变化做出低延时反应的应用场景。由于CEP 应用如此广泛，必须根据不同的应用场景正确配置与定制CEP 系统。不同的应用领域，可能需要特殊的事件处理方法，如采用分布式还是集中式模式。CEP 进一步的研究课题包括处理不确定性的事件（例如，用概率论方法），检测一个预先未知的复杂事件（例如，事件流的数据挖掘）等。

# 三、结论

【内容要求】通常包括两个部分内容，一是对研究方向的若干分支及每个分支的理论/方法/方案/技术研究的现状和前途的小结；二是引出自己论文选题的背景、目标和必要性，即希望按那一种技术思路进行，方法、方案和技术有什么重要的改进，要解决那些关键问题，解决选题的必要性问题。

【写作要求和篇幅量】每一部分最好是一段文字，每段文字不要超过300字。

【常见问题】没有技术主见或技术观点模糊。

# 主要参考文献

参考文献是自己在进行一个课题研究或者项目开发之前，对当前或同领域情况的调研是否充分、深入的具体体现。

【引用规范】参考文献中只应列出正文中以标注形式引用或参考的有关著作和论文。参考文献的排列次序应与正文中引用或参考出现的次序相同。一篇论著在论文中多处引用时，在参考文献中只应出现一次，序号以第一次出现的位置为准。引用参考文献格式示例如下:

* 期刊

[序号]作者，文章题目，期刊名，年份，卷号，期号（若期刊无卷号，则为：年份、期号），起止页码

* 专著

[序号]作者，书名，版本（第一版不标注），出版地，出版者，出版年，起止页码

* 论文集

[序号]作者，题名，主编，论文集名，出版地，出版年，起止页码

* 学位论文

[序号]作者，题名，学位论文名（英文用Dissertation），保存地点，保存单位，年份

* 专利

[序号]专利申请者，题名，国别，专利文献种类，专利号，出版日期

* 技术标准

[序号]起草责任者，标准代号，标准顺序号—发布年，标准名称，出版地，出版者，出版年度

* 外国作者的姓名书写格式为：姓、名字或缩写。例如Johnson，A. J.

1 Etzion O. Complex Event Processing[M]. Springer US, 2010.

2 Luckham, David C. The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems[J]. Lecture Notes in Computer Science, 2008, 5321:3-3.

3 Wu E, Diao Y, Rizvi S. High-performance complex event processing over streams.[C]// ACM SIGMOD International Conference on Management of Data, Chicago, Illinois, Usa, June. 2006:407-418.

4 Wu, Eugene, Diao, et al. High-performance complex event processing over streams[J]. 2006:407-418.

5 Cugola G, Margara A. Processing flows of information: From data stream to complex event processing[J]. Acm Computing Surveys, 2012, 44(3):359-360.

6 Wu, Eugene, Diao, et al. High-performance complex event processing over streams[J]. 2006:407-418.

7 Margara A, Cugola G. Processing flows of information:from data stream to complex event processing[C]// ACM International Conference on Distributed Event-Based Systems, Debs 2011, New York, Ny, Usa, July. 2011:359-360.

8 Wang F, Liu S, Liu P, et al. Bridging Physical and Virtual Worlds: Complex Event Processing for RFID Data Streams[J]. Lecture Notes in Computer Science, 2006, 3896:588-607.

9 Gyllstrom D, Wu E, Chae H J, et al. SASE: Complex Event Processing over Streams[C]// Biennial Conference on Innovative Data Systems Research. 2007:407-411.

10 Luckham D C, Frasca B. Complex Event Processing in Distributed Systems[J]. Stanford University, 1998.

11 Adi A, Hadash K, Kerem O, et al. Definition of workflow patterns using complex event processing: US, US20060229923[P]. 2006.

12 Bates J, Smith G, Bentley R M, et al. Automated construction and deployment of complex event processing applications and business activity monitoring dashboards: US, US8276115[P]. 2012.

13 Zang C, Fan Y. Complex event processing in enterprise information systems based on RFID[J]. Enterprise Information Systems, 2007, 1(1):3-23.

14 Distefano M V. Complex event processing cloud: US, US8024480[P]. 2011.

谷峪, 于戈, 张天成. RFID复杂事件处理技术[J]. 计算机科学与探索, 2007, 1(3).

丁剑, 白晓民, 赵伟,等. 基于复杂事件处理技术的电网故障信息分析及诊断方法[J]. 中国电机工程学报, 2007, 27(28):40-45.

杨清, 徐建良, 高德欣. 基于复杂事件处理技术的RFID系统数据分析[J]. 微计算机信息, 2006, 22(9Z):179-181.

臧传真, 范玉顺. 基于智能物件的实时企业复杂事件处理机制[J]. 机械工程学报, 2007, 43(2):22-32.

阴晓加, 鞠时光, 王英杰. 基于复杂事件处理机制的RFID数据流处理方法[J]. 计算机应用, 2009, 29(10):2786-2790.

臧传真, 范玉顺. 基于智能物件的制造企业复杂事件处理研究[J]. 计算机集成制造系统, 2007, 13(11):2243-2253.

陈远, 李战怀, 陈群. 不可靠RFID数据上的复杂事件处理研究[J]. 计算机应用研究, 2009, 26(7):2537-2539.

项立锋, 孙建伶, XiangLifeng,等. 基于复杂事件处理的算法交易研究[J]. 计算机应用与软件, 2010, 27(12):154-157.

许婧, 白晓民, 徐得超,等. 基于复杂事件处理技术的连锁故障诊断[J]. 电力系统自动化, 2011, 35(3):5-8.

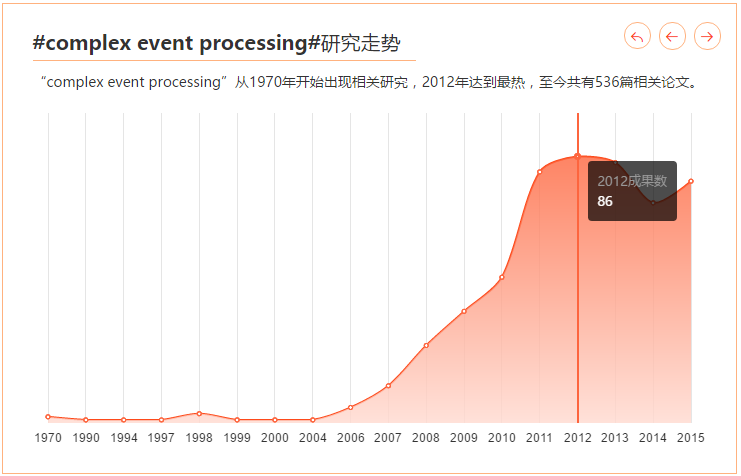
黃毅, 郑力, 向晴. 基于复杂事件处理的RFID辅助实时生产监控[J]. 清华大学学报(自然科学版), 2013(5):721-728.

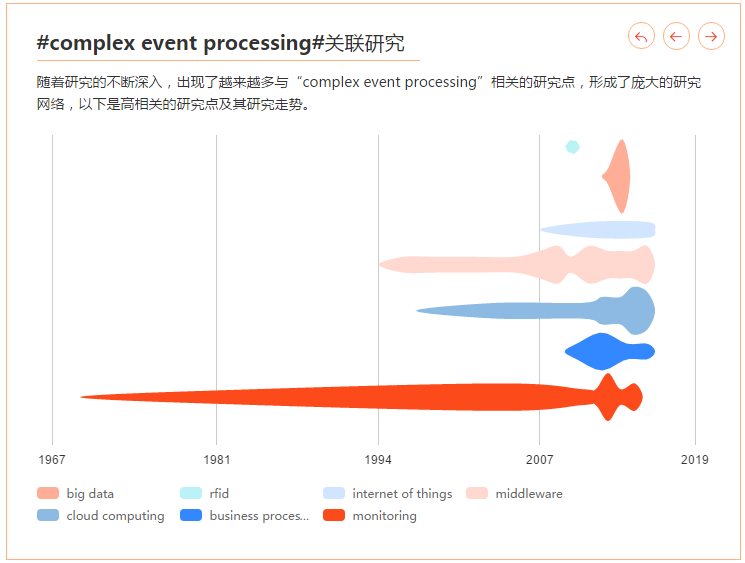
【常见问题】参考文献部分常常出现的问题是：①讲教材作为参考文献；②将网站内容作为参考文献。网站内容由于仅代表网站拥有单位或者发表者个人的观点，不具备学术性，因为不宜作为参考文献，需要引用时，可以采用页面底端的脚注方式说明观点的出处。

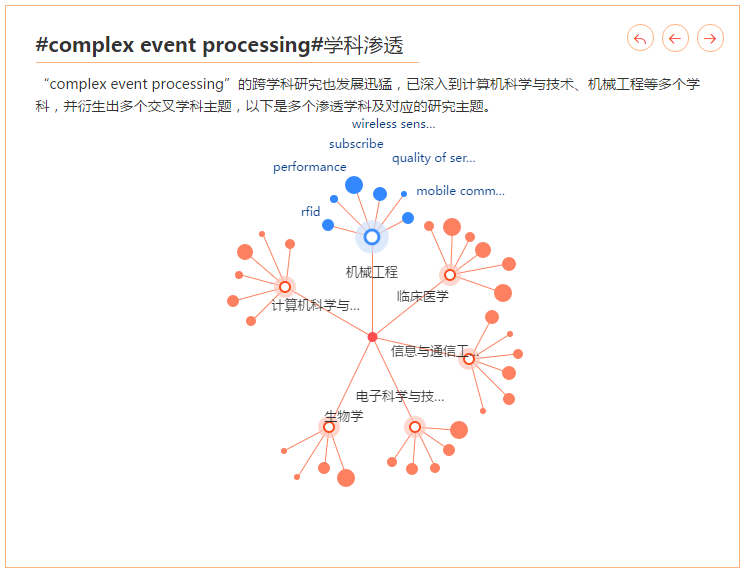
# 

# 附：主要参考文献综述简表（注：不少于20篇，其中外文不少于10篇）

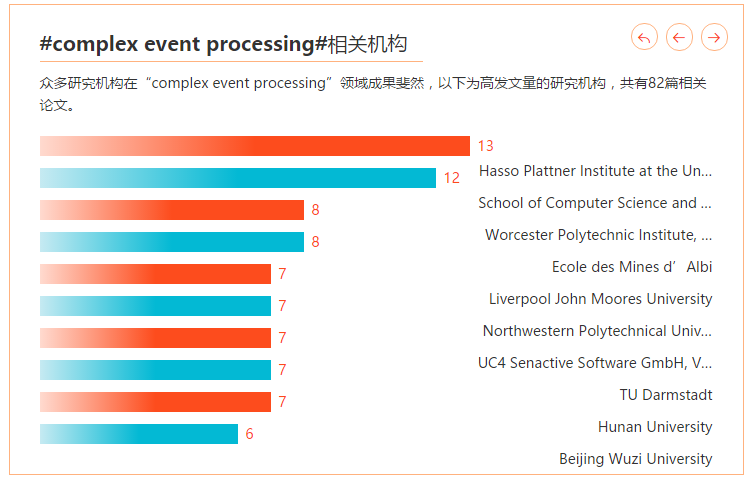
| **参考文献引用序号** | **参考文献主要学术观点和内容** | **对你论文工作有重要参考价值的内容** | **是否追溯系列文献** | **备注** |
| --- | --- | --- | --- | --- |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |
| 9 |  |  |  |  |
| 10 |  |  |  |  |
| 11 |  |  |  |  |
| 12 |  |  |  |  |
| 13 |  |  |  |  |
| 14 |  |  |  |  |
| 15 |  |  |  |  |
| 16 |  |  |  |  |
| 17 |  |  |  |  |
| 18 |  |  |  |  |
| 19 |  |  |  |  |
| 20 |  |  |  |  |
| 21 |  |  |  |  |
| 22 |  |  |  |  |
| 23 |  |  |  |  |
| 24 |  |  |  |  |
| 25 |  |  |  |  |
| 26 |  |  |  |  |
| 27 |  |  |  |  |
| 28 |  |  |  |  |
| 29 |  |  |  |  |
| 30 |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |











1 Etzion O. Complex Event Processing[M]. Springer US, 2010.

2 Luckham, David C. The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems[J]. Lecture Notes in Computer Science, 2008, 5321:3-3.

3

4 Wu, Eugene, Diao, et al. High-performance complex event processing over streams[J]. 2006:407-418.

5 Cugola G, Margara A. Processing flows of information: From data stream to complex event processing[J]. Acm Computing Surveys, 2012, 44(3):359-360.

6

7 Margara A, Cugola G. Processing flows of information:from data stream to complex event processing[C]// ACM International Conference on Distributed Event-Based Systems, Debs 2011, New York, Ny, Usa, July. 2011:359-360.

8 Wang F, Liu S, Liu P, et al. Bridging Physical and Virtual Worlds: Complex Event Processing for RFID Data Streams[J]. Lecture Notes in Computer Science, 2006, 3896:588-607.

9 Gyllstrom D, Wu E, Chae H J, et al. SASE: Complex Event Processing over Streams[C]// Biennial Conference on Innovative Data Systems Research. 2007:407-411.

10 Luckham D C, Frasca B. Complex Event Processing in Distributed Systems[J]. Stanford University, 1998.

11 Adi A, Hadash K, Kerem O, et al. Definition of workflow patterns using complex event processing: US, US20060229923[P]. 2006.

12 Bates J, Smith G, Bentley R M, et al. Automated construction and deployment of complex event processing applications and business activity monitoring dashboards: US, US8276115[P]. 2012.

13 Zang C, Fan Y. Complex event processing in enterprise information systems based on RFID[J]. Enterprise Information Systems, 2007, 1(1):3-23.

14 Distefano M V. Complex event processing cloud: US, US8024480[P]. 2011.

谷峪, 于戈, 张天成. RFID复杂事件处理技术[J]. 计算机科学与探索, 2007, 1(3).

丁剑, 白晓民, 赵伟,等. 基于复杂事件处理技术的电网故障信息分析及诊断方法[J]. 中国电机工程学报, 2007, 27(28):40-45.

杨清, 徐建良, 高德欣. 基于复杂事件处理技术的RFID系统数据分析[J]. 微计算机信息, 2006, 22(9Z):179-181.

臧传真, 范玉顺. 基于智能物件的实时企业复杂事件处理机制[J]. 机械工程学报, 2007, 43(2):22-32.

阴晓加, 鞠时光, 王英杰. 基于复杂事件处理机制的RFID数据流处理方法[J]. 计算机应用, 2009, 29(10):2786-2790.

臧传真, 范玉顺. 基于智能物件的制造企业复杂事件处理研究[J]. 计算机集成制造系统, 2007, 13(11):2243-2253.

陈远, 李战怀, 陈群. 不可靠RFID数据上的复杂事件处理研究[J]. 计算机应用研究, 2009, 26(7):2537-2539.

项立锋, 孙建伶, XiangLifeng,等. 基于复杂事件处理的算法交易研究[J]. 计算机应用与软件, 2010, 27(12):154-157.

许婧, 白晓民, 徐得超,等. 基于复杂事件处理技术的连锁故障诊断[J]. 电力系统自动化, 2011, 35(3):5-8.

黃毅, 郑力, 向晴. 基于复杂事件处理的RFID辅助实时生产监控[J]. 清华大学学报(自然科学版), 2013(5):721-728.



**制造系统复杂事件处理机制的研究**

复杂事件处理引擎及关键技术的研究