



复杂事件处理手册

复杂事件处理手册

在金融服务和其他行业中，如何使那些重要且具有战略意义的业务信息以高速数据流的方式到达企业变得尤其重要，而复杂事件处理(CEP)就是这一过程的代名词。在复杂事件处理中，数据是不断变化的，而“操作”是“静态”的。复杂事件处理具备了分析高速数据流并鉴别重要事件的能力，虽然对这些事件的鉴别过程是复杂的，但结果却是无价的。复杂事件处理能够帮助企业及时全面地洞察市场变化，降低风险和提高决策效率。下面我们就来介绍一下复杂事件处理。

复杂事件处理 CEP

想象一下你开着车，穿过大街，同时忙着弄东西吃，或者忙着干其他任何事情，如果你不及时的观察，处理和响应周围时时刻刻发生的事情，后果将不堪设想！同样，这种迅速辨识，捕捉和响应实时数据的能力，也正在成为很多业务领域一项至关重要的基本功。

- ❖ **实时分析应用：CEP 解决部署难 又快又省不是梦**
- ❖ **复杂事件处理平台（CEP）：更快 不贵 不复杂**
- ❖ **2012 年事件处理或成主流技术**

事件驱动处理 EDA

在业务系统环境里，“事件”这个术语意味着一个定义明确、分类清晰的业务条件发生了。那么 EDA 中事件是如何界定的？如何利用事件驱动系统降低变更成本？下面我们来具体看一下。

-
- ❖ EDA 中事件的界定和社交媒体 Twitter
 - ❖ 如何利用事件驱动系统降低变更成本？
 - ❖ 专家答疑：事件驱动处理

SOA 和 EDA

复杂事件是一种事件，通常多个更简单事件异步发生时预示着这种事件。一个人投球，一个人挥棒击球，另一个人跑垒，每个都是彼此孤立的事件，在一起组成了棒球比赛这个复杂事件。下面我们来讨论一下 SOA 和 CEP 如何一起工作。

- ❖ 如何用 SOA 创建业务利益的复杂事件处理（CEP）
- ❖ 如何使用“正好”的复杂事件进行 CEP？

实时分析应用：CEP 解决部署难 又快又省不是梦

环境感知度在我们生活的几乎各个方面都是不可或缺的。想象一下你开着车，穿过大街，同时忙着弄东西吃，或者忙着干其他任何事情，如果你不及时的观察，处理和响应周围时时刻刻发生的事情，后果将不堪设想！缺乏了解或心不在焉（而不具有好的环境感知度）将会导致错误的决定，甚至带来大祸临头。

同样，这种迅速辨识，捕捉和响应实时数据的能力，也正在成为很多业务领域一项至关重要的基本功。

许多组织机构，尤其是金融服务、电信、电子商务、政府的国防、情报和税收部门，其日益增长的需求已远非传统的智能业务应用所能满足。他们需要能够不间断获取情报的能力，从而能够分析大流量的实时事件，并迅速洞察事件原委，并即时——常常是自动化地——响应不断变化的世界。

在这些情景中，“实时”的含义已经发生了变化：分钟级或小时级的响应速度早就落伍了。即使是秒一级也略显不足了。现在，实时情报和响应越来越多的要求子秒级的响应速度——毫秒级或更快。在许多机构中，已经不再区分过去和将来——只有现在！

这种需要不间断数据处理的例子并不鲜见：比如在金融交易应用中，一毫秒的延迟可能会造成天壤之别——或者是巨大收益，或是损失惨重。同时你也会发现，在经历了最近的华尔街灾难之后，金融服务行业现在更加注重风险分析和抵御。电子商务公司，他们的客户买或者不买，其决定都在鼠标点击的一瞬之间，于是乎，他们开始研究用户行为模式和对实时商品信息的反应模式，从而增加交易的数量和利润。电信公司，面临激烈的市场竞争或巨大的流量变化，需要不间断的实时数据处理来监控和管理网络通信量来确保网络容量的最优利用和用户满意度的提升。在军事和国家安全领域，如果不能及时发现不良动

向的预兆和即将到来的威胁，其后果将不可估量，因此，不间断的情报获取和处理能力是获得成功的一个最起码的要求。

开发人员和客户都要面对的艰巨挑战

毫无疑问，要能够对持续大流量的实时数据进行分析，并能够快速响应，即使对技术最先进，最具创新能力的组织来说，也是一项令人望而生畏的挑战。

业务需求足够清晰，组织需要做的是：

- 减少应用存储数据（在分析数据之前）造成的延迟
- 能够持续，实时地分析多个数据流
- 能够关联不同数据流中的事件，从而发现新的相关情形
- 能够迅速响应新发现的危险或机会
- 能够迅速的将先前发现的规律应用到新的数据流分析模型中
- 能够利用已有的应用开发能力快速开放新的高性能，高扩展度的应用
- 确保应用和系统的连贯性。

批量分析已经不能提供足够快的实时智能处理。要满足日益苛刻的实时业务需求，这些特定的应用需要很多不同专业的编程人员来协作开发，包括网络程序员，事项驱动程序员，多线程程序员等等。这使得特定的应用开发成本昂贵，并且，开发周期太长。

如果能够买到现成的应用来满足组织机构的需求，将是一件多好的事啊。我曾经同很多组织很多人讨论过，但是他们都说，这样的应用根本就不存在，或者根本不能提供他们需要的功能。那么，当传统的方法——或者自己开发或者买——都不能奏效的时候，我们该怎么办呢？

(作者：Jeff Wootton 译者：李松 来源：TechTarget 中国)

原文链接：http://www.searchsoa.com.cn/showcontent_37557.htm

复杂事件处理平台（CEP）：更快 不贵 不复杂

尽管实时分析应用对比如说金融服务等机构来说并不陌生，但是，对那些需要从头开始创建这类应用的组织来说，实施起来所花的成本和时间，实在是一个繁重的负担。

很多处于不同行业的组织，本来可以从实时分析应用中获得巨大收益的，可是由于缺乏必要的资金支持和专业人员来构建他们自己的应用，也没有合适的现成产品可以购买，于是他们不得不在缺乏这项功能的情况下疲于应付。

然而，现在，我们有了第三个选择：复杂事件处理平台。

一个复杂事件处理（CEP）平台提供了专业开发应用的功能，现货供应软件的价格：从内在设计上，它使得组织能够以超过每秒一百万条消息的处理速度分析信息流，其延迟时间为毫秒级或更少，从而提供一个快速应用开发和实施的平台。就象许多其他平台一样，一个 CEP 平台能够加快开发，降低成本，消除对专业开发技巧的依赖，可以说，CEP 平台使得我们能够摆脱既不能自己开发又无处购买的二难境地。

很多曾经一度想自己从头开发实时分析应用的组织，现在发现，采用一套 CEP 平台，使得他们将开发和部署的时间从预期的 18 个月降低到了 4 个月！还有一些组织说，CEP 平台使得他们的开发时间和成本削减了多达 85%！

评价标准

当然，不是所有的 CEP 平台会产生同样的效果。因此，审慎的进行评估选择也是非常必要的。一个有效的 CEP 平台能够提供：

- 一个 CEP 引擎，能够连续处理进来的数据流，并同时可以提供类似 SQL 语言的查询
- 能够将 CEP 引擎集成到其运行环境的适配器
- 软件开发包 (SDK)能够支持流行的开发语言包括 C/C++、C# 、Java 等，从而使得客户可以开发一些自己可以定制的适配器和客户程序，以及一整套的工具来帮助编程人员和管理人员来开发，调试，部署和监视 CEP 的诸多应用程序。

在评价一个 CEP 平台时，你应该特别关注其性能和可扩展性，从而确保你选择的平台能够在处理你当前和预期的数据流时，其延迟时间足够低，并且足够稳定，从而可以满足你的业务需求。（平均延迟率的意义并不太大，因为只有负载运转时的延迟指标才真正影响你的业务能力）。同时，你也要确保该平台所提供的事项处理语言要直观，易于学习，并且通用，这样才能支持诸多的处理逻辑。

当你评价 CEP 平台时，你还会问其他一些问题，比如：

- 开发工具中的编程模型，是否能吸引众多不同类型的开发人员？
- 是否提供了高可用性和高安全性？
- 是否可以部署在多个使用阵列配置的服务器？
- 是否提供了监视功能？
- 其引擎是否可以易于扩展，从而来支持公司特有的高级的逻辑？
- 是否有恢复机制，即使一旦服务器故障，也可以防止数据丢失？
- 进出处理引擎的消息传递，其机制是否可靠？
- 它同数据库以及其他系统能否很好的集成？
- 这个平台是否支持连续型实时数据管理的各个生命周期，包括分析，存储，建模和一个新的实时分析应用的部署？

引人注目的使用效果

使用复杂事件处理来实现持续的实时智能处理，正成为越来越多的组织机构用来解决其面临的许多业务挑战的一个有效方法。

其好处是明显可见且引人注目。通过实施 CEP，你的组织能够：

- 明白眼下正在发生的事情
- 更迅速地响应变化的环境（包括对特定情景的自动响应）

能够基于完整和真正实时的信息，从而更好的决策。越来越多的组织发现，使用 CEP 平台：

- 降低了开发和部署实时分析应用所花的时间和精力
- 不再需要专业化的实时编程技巧，并且通过交付实时观察和授权即时响应增强组织敏捷性。

能够实时辨明情况，即时响应，提供了组织的敏捷度。随着不间断实时智能处理成为越来越多的机构的一个基本业务能力，目前的唯一问题是如何更好实现它。选用一个 CEP 平台或许是你和你的组织的一个最好的选择。

(作者：Jeff Wootton 译者：李松 来源：TechTarget 中国)

原文链接：http://www.searchsoa.com.cn/showcontent_37562.htm

2012 年事件处理或成主流技术

在业务系统环境里，“事件”这个术语意味着一个定义明确、分类清晰的业务条件发生了。在这个环境下的事件提供了恰到好处的信息，以便接收方能如期响应所发生的特定类型的事件。这些事件均以业务为焦点；比如说，新客户的增加，员工辞职，支票签发，项目支出超过了预算的 10%，等等。

挥别 2011，走进 2012，在企业架构领域您能预测一下下一个大事件是什么吗？

这个问题使我们思考未来的很好机会。现在所发生的是不是已经为我们描绘了 2012 年将会被广泛接纳的主流呢？如果你阅读过我的博客，对于我的回答一定不意外：事件。实际上，我在 2011 年初的一篇博客中就以《让 2011 年成为事件之年》为题。虽然我知道 2011 年肯定不会完全实现，但是我相信已经有很多力量聚集到一起，在未来十年，会对事件驱动系统产生重大影响。

第一个吸引人的是 node. js。Node. js 是一套服务器端 JavaScript 库，实际上它是对 GoogleV8 引擎（应用于 Google Chrome 浏览器）进行了封装。为什么这会是重要的事情？Node. js 使用事件驱动、非阻塞 I/O 模型。它是异步的。这也造就了良好的性能和可扩展性。今年八月份，LinkedIn 发布了彻底检修的移动应用，他们也声明能完全为 node. js 支持服务器端的功能。

尽管我所了解的大多数 node. js 的故事都来自初创公司或者技术社区，而不是公司 IT 部门，但是我希望 node. js 能够在那里也找到一席之地。我认为这件事不同于从 Java 到 PHP 或者 Ruby 的转移。这些语言仍遵循类似的 Web 应用请求响应范式。如果 node. js 继续得到关注，我们会看到那些理解基于事件编程的服务器端开发者的增长。

第二件事就是移动。移动设备的使用非常深远，但是仍旧处于生长阶段的事实是通告和移动定位服务。现在，大多数这些移动应用首先必须有用户开启，然后请求访问存储单元，然后用其做一些智能的操作。世界上并不缺少那些宣扬这种方法的人。手机的位置被检测，一些用用告诉用户，“嘿，瞧瞧我！我想给你看一些东西，最重要的是基于你在哪里才能看！”这就是一种基于事件的方法。

为了实现这个，很明显我们需要一个事件流。这件事的基础就是通过 Facebook 和 Twitter 来确定。这些就是事件流，等待有人采取行动来作出响应。任何事件流，都有大量信噪比要关注。我确信我们我都经历过这样的事情，在我们浏览 Facebook 或者 Twitter 流时，有人看过来说“为什么人们要发这些东西？”尽管这是件很棒的事情，处理“大数据”的过程中已经为我们提供了工具来处理这些数据。

最后，从业务观点出发，在很多案例中，我们在可用信息上已经取得进步，现在需要强调的是为手头的任务发现相关的信息。Gartner 在 2008 年的努力方向是正确的，他们引入了“内容驱动架构”概念。内容是事情必然相关的。它不是一个单一的信息片，相关是一个统计、时间、位置、历史活动、对等活动、近期活动等等的结合。我们可以通过追踪事件和接着发生的响应，为用户或者系统构建一个内容模型。过去，分析可能单独关注于结果（如事务日志），但是缺失了很多能够引导决策的事件。点击流和 HTTP 日志的分析改善了这件事，但是仍有很多事情要做。

除了这些还有很多内容，一些主流（如 BPM 引擎）和一些强大的壁龛应用（复杂事件处理系统），当然，所有的这些都需要强健的资源组合。但是在我考虑这件事所需要的内容时，很难找没有收到牵引的领域。可能我 2011 年初博客唯一的错误就是时间表。SOA 终其一年也没能成为主流，但是你能轻易地就说出从 2000 年以后的十年就是服务为主。现在，我们也预测 2010 年后的十年就是事件为主，事件和内容系统驱动系统和流程将成为主流，驱动着事件、移动、社交和大数据技术的成功利用。

(作者: Todd Biske 译者: 张培颖 来源: TechTarget 中国)
原文链接: http://www.searchsoa.com.cn/showcontent_56947.htm

EDA 中事件的界定和社交媒体 Twitter

在业务系统环境里，“事件”这个术语意味着一个定义明确、分类清晰的业务条件发生了。在这个环境下的事件提供了恰到好处的信息，以便接收方能如期响应所发生的特定类型的事件。这些事件均以业务为焦点；比如说，新客户的增加，员工辞职，支票签发，项目支出超过了预算的 10%，等等。

使用事件驱动方案时，作为消费一方的应用可监控特定的业务事件，作为响应，它可以选择忽略该事件，执行即时动作，为了更好的效率而延迟响应，或将信息与从其他事件中接收到的信息组合起来，形成对业务的集成视图。消费应用本身也可能产生新的业务事件，可被其他消费应用做进一步的处理。产生该事件的应用不需了解其消费者，也不需要明白事件是如何被消费的。结果，我们得到的是一个非常松耦合、高伸缩性、灵活且变更成本低的应用结构。

在事件驱动的环境里，不存在其他交互模型中出现的固定的消费者—供应者关系。相反，业务应用会通告某一特定业务事件的产生，任何侦听该事件的应用就成为了消费者。消费者仅接收刚好够用的信息来唯一识别出该事件，可自由地按需对事件进行处理。其他的方案导致传递庞大、复杂的数据集给所有消费者，而不管其是否需要整个数据集的数据，给伸缩性带来了负面影响，并增加了变更成本。

在这个事件驱动模型中，新的消费者可以在无需对现有业务系统进行任何改变的情况下被添加进来。与不得不对现有的时间—请求驱动型业务应用开发昂贵的新功能不同，事件驱动型应用无需冒稳定性和性能下降的风险，新功能可通过应急部署或在云上的应用提供出来。

社交网络领域面向事件环境的一个例子是 Twitter。一个人通知（tweets）了一个事件的产生，那些微博世界（twitterverse）里该人的追随者——人或者系统，就会知道该

事件的发生。该事件可以有任意数量的接收者，每个接收者均可选择响应、忽略、改变或者重新通告该事件。通过使用标签来传达元数据，事件得以分类。附加信息通常利用 URL 就可以表示出来，确保了原始的事件通告机制可以极端可伸缩化。解决方案被持续不断地添加到 Twitter 的生态系统里，却无需对 Twitter 的核心环境进行任何改变。

业务系统环境也可以从类似的松耦合、轻载、安全的事件驱动结构中获益。微博 (tweet) 的概念成为了一个定义清晰、简洁的业务事件，一个 URL 则提供了消费者所需要的获取与事件细节有关的更多数据的链接。业务应用充当事件的监控者，对所期望的事件进行响应，并根据额外的业务使用要求转发事件或改变该事件的用途。这样就能诞生出前所未有的创新的业务解决方案。由于这种方案在节约变更成本方面极其诱人，当机会来临时，其利用机会的能力将会令你的业务在竞争中占得先机。

SOA 怎么办？

最后，事件驱动方案并不意味着 SOA、WOA 等就没了一席之地。恰恰相反。将事件驱动方案对业务系统交互最好的部分与最好的支撑架构如 SOA 相结合起来，对于获得最高的商业价值来说是必不可少的。在你的业务系统架构演进（非变革）的道路上，采纳事件驱动方案应该成为你迈出的下一步。

(作者: Ryan Gloeckler 译者: 杨华军 来源: TechTarget 中国)

原文链接: http://www.searchsoa.com.cn/showcontent_34281.htm

如何利用事件驱动系统降低变更成本？

在今天这个高速运转、动荡变化的商业世界里，为了快速适应新需求、新规则及新机遇，业务系统的敏捷性至关重要。变更需要在数日而非数周或数月内实现，且不应动用一大群 IT 员工就能做到。而当前的业务系统结构却经常成为敏捷性的壁垒，变更成本代价又太过高昂。

业务系统通常是由许多独立的应急应用系统，或者是（正在不断增加的）云中的应用的组合。这些应用之间的联系也需要是敏捷的，要能快速适应新需求和机遇。本文讨论了业务系统通常是如何进行组合的，而事件驱动应用可以提高组织的灵活性及智能化程度，以便降低总的变更成本。以下是详解。

时间驱动及请求驱动交互是业务应用的两种常见的通讯类型。时间驱动型交互按照一个预定义的时间表管理应用向另一个应用请求服务或信息时的活动，比如说像轮询这样的场景。时间驱动交互的一个例子是每天从物资仓储管理系统下载获得最新的物资目录。

在请求驱动型交互里，应用根据需要向另一系统请求服务或信息，通常是以逻辑嵌入的形式体现在应用中，比如推-送场景。请求驱动型交互的例子是当向 CRM 系统添加一新客户时，CRM 会向记账系统提出将客户作为新的应收账款成员的请求。

在像每日的物资目录这类交互频率不高或交换信息量很少的情况下，时间驱动型交互可以工作得很好。随着业务要求更多接近于实时的信息，时间驱动的交互频率在不断增长。举个例子，与每日更新的供应商物资目录不同，购买者需要最新的价格和存货数。时间驱动型交互最终将会导致性能及伸缩性的下降，因为系统间的通讯经常会在不需要的时候发生，导致不必要的波动。

请求驱动型交互提供了一个更好的场景。应用对另一个系统的服务或信息请求是作为应用的逻辑而体现的。作为交互的形式，这多少要比时间驱动的办法要好一些，因为每一次交互均涉及应用中的一个特定需求。然而，最终的系统并不会像可能的那样灵活有效，而对新需求和机会的适应则需要应用内部进行修改。这一类变更一般都是比较昂贵的，需要花费数周或数月的时间。

第三种类型的交互正变得更为普遍—在这些交互中，业务系统指出何时有有意义的业务事件发生，允许任何消费者根据需要对事件进行处理。这些事件驱动型的交互正引领着新型的业务应用及解决方案的开发，使得业务能快速适应变化。事件驱动方案并非什么新鲜事物，但现在，它在降低变更成本至关重要的商业应用领域里正在获得应有的关注。

(作者: *Ryan Gloeckler* 译者: *杨华军* 来源: *TechTarget 中国*)

原文链接: http://www.searchsoa.com.cn/showcontent_34282.htm

专家答疑：事件驱动处理

同 Roy Schulte 一起，K. Mani Chandy 写作了关于设计事件驱动系统的书籍。Chandy 博士还获得了很多奖项，比如说，包括康联迈克尔逊奖(CMG Michelson Award)，IEEE 小林奖，以及巴贝奇奖。同时，他还是美国国家工程学院成员。在 Chandy 的新书《事件处理：敏捷公司的 IT 系统设计》(Event Processing: Designing IT Systems for Agile Companies) 出版后，我们同他讨论了事件处理相关的一些话题。最近，SearchSOA. com 网站编辑 Jack Vaughan 就此采访了 Chandy。

能否请你给我们简要地解释一下事件处理？

事件处理有四个部分。第一部分是你需要持续的获取数据，有很多种方法可以用来获取数据。其中一些方法是使用传感器。很多公司发现他们的生产和物流中的传感器已经太多，这些传感器产生了大量的数据，远远超过了他们的日常所需。因此，在第一部分获取数据之后，第二部分是数据的融合 — 将多个不同来源的数据关联起来。一旦你将这些数据融合起来，你就能够了解外界的情况以及外部世界是如何变化的。因此第三部分就是计划。你希望对这种改变做出响应。第四部分是执行这个响应计划。这四个部分的结合使得事件处理对于业务来说非常重要。

对于那些想要实施事件处理的读者，你有何建议？

我们的建议是，尤其是对于刚要准备开始的企业，一定要循序渐进，对每一步实施给业务带来的好处都进行评估。对重要的执行主管来说，一个重要的步骤是获取实时或接近实时的业务性能指示器。其首先关注的是我们前面讲过的第一部分的内容和第二部分的部分内容，也就是说，首先是数据采集，其次是非常简单的关联，在某种程度上能够将这些信息集中在一个表格中，这对执行主管最有用，因为他可以在一个屏幕之内看到所有这些信息。这就是所谓的业务活动监控(BAM)。

从这个相对简单的步骤出发，你可以确保这对公司有用：这能帮助执行主管更好的了解外部的发展变化，从而可能会作出更好的决策。一旦管理层看到这些益处，就可以实施下一步：进一步做更实时的关联，更实时的数据融合。开始挖掘数据之间的关联模式，这比展示给管理层一屏幕的数据更有价值。这样你就真正能够提取信息并将摘要提供给执行主管，这些信息也更加容易付诸实施。

此后才能够实施后续的步骤，进入到计划这一阶段。我认为，这些事情你需要一步一步地进行，首先考虑这样一个问题“我目前已经具有的软件系统中，哪一部分可以用来帮助我实现第一步？”

人们通常容易误以为，他们不得不丢弃他们的 SOA 系统并买一些全新的非常昂贵的东西。实际上他们可以在既有的系统之上开展事件处理的工作，如果有必要，再逐步地实行一些比较激进地动作。

在面向服务的架构中，事件会起什么作用？

我想分两部分来回答这个问题。服务通常被认为是由客户端引发的一些操作。客户请求一个服务，然后服务对客户的这个特定请求做出响应。客户说“我想让你根据这些参数去执行这个服务。而现在，事件在某种程度上改变了整个过程：客户所作的只是指定一个长期的设置。客户并不请求某个特定服务的执行。客户在说：“我对以下这些类型的事情感兴趣，并且我的兴趣会一直持续到相对长的一段时间”。

客户和服务器关于服务的契约一般来说是一个服务。当服务结束后，这个契约也就结束了。而客户和服务器对于事件处理的契约通常是长期的，非特定的。它更多的是一个兴趣的集合而不是某个带有特定参数的服务。这是服务和事件之间真正的区别所在。

说到实施，SOA 有时候被认为不仅仅是服务，正如 Roy Schulte 说讲，SOA 是一些模块化的准则。并且这些模块化的准则不仅仅适用于服务，也同样的适用于事件。因此，在这一意义上来说，SOA 意味这模块化，它毫无二致地适用于事件和服务。

事件处理在“现实世界”中的应用如何？

这就是生活中的东西。它甚至可以应用到象水这样基本的事物中，因为你需要在水中的传感器来确保水的正确流动和好的水质。还有越来越多的问题，比如水权之争，水是如何分布的等等。所有这些都需要持续的进行监控和响应。同样，对于食物“从农场到餐桌”的整个过程进行管理也是很重要的。

同样，对于医药品，也有管理的需要，比如说，要确保出现问题的药品批次能够召回，并且确保能够快速召回。因此，能够检测到某个问题的出现并迅速响应的能力是非常重要的。

还有，当下对于社会网络进行关注并对其发生的变化进行快速响应也很有意义。此外，对新闻故事和博客中的文字进行分析，并根据从中得到的信息及早响应，也是吸引了大量关注的一个应用。因此我认为这确实就是生活中活生生的例子。并且这也是我认为将是越来越重要的一种应用。

所以，当你看到生活中的这些实例时，你看到了事件？

我把这称作感知和响应。感知我们周围的这个世界并对其做出响应。事件是其中的一个部分。

最后，您还有其他的想法吗？

我还想要说的是，现在已经有了这个事件处理技术社团，它由一组公司构成，既有事件处理的厂商也有事件处理的用户，以及学术界和政府部门的代表。它被称作 EPTS，它有几个目标，其中之一是解释什么是事件处理，什么不是，尤其是它比较公平地谈论了事件处理的优势和弱点。同时它还帮助去开发一些标准。我们不会建议你采用某个特定的标准，但我们会告诉你如何去实现你的目标。

(作者: Jack Vaugh 译者: 李松 来源: TechTarget 中国)

原文链接: http://www.searchsoa.com.cn/showcontent_42071.htm

如何用 SOA 创建业务利益的复杂事件处理 (CEP)

复杂事件是一种事件，通常多个更简单事件异步发生时预示着这种事件。举一个非软件概念的例子：一个人投球，一个人挥棒击球，另一个人跑垒，每个都是彼此孤立的事件，正是这些事件在一起组成了棒球比赛这个复杂事件。在软件中，复杂事件处理 (CEP) 是识别复杂事件的一种能力，在一个软件系统中，个体业务事件的异步发生预示着这个复杂事件。服务间的消息交换经常暗示这个软件系统，并相应地应答。这些应答包括生成一个新事件，删除重复逻辑，和其他行为。

独立咨询师 Larry Fulton 最近给出了一则关于 CEP 和 SOA 的网络广播，讨论这两者如何在一起工作。Fulton 解释说，CEP 用业务流程帮助调整系统处理，通过翻译传入的业务事件并将它们简化成可以在 SOA 中的其他地方消费的新消息，从而减少了前阶段需要处理的工作量。反过来，适当地让 SOA 从组织的一个跨多个业务域，并且更容易实现的区域的 CEP 实现中获益。

下面三个观点是来自 Fulton 的网络广播，利用 SOA 来影响 CEP：

CEP 和 SOA 应当从两个角度来看。首先，应该把 CEP 看作 SOA 内的一种实现选择。CEP 可以用来实现新的业务能力——例如流程精简和关联——作为服务。其次，可以把 CEP 看作服务源。如果在适当的位置使用 CEP 系统，它们可能包含 SOA 中其他的用新方式来包装的服务。

如果你不确定是否需要 CEP，从理解你的业务事件开始来决定 CEP 是否合适。Fulton 推荐同业务人员对话来发现在流程中，什么时候，发生哪些事件，以及哪些事件对于业务目标来说最重要的。他还推荐分析靠近业务的底层事件，因为经常这么做会对业务整体产生新洞察。

CEP 将从消息的单一库中获益，而不是从可能存在很多重复标题或消息的很多更小的库中。Fulton 建议，通常来说，任何时候一个组织对消息共享的期望超过了基本的点对点交换，例如在 SOA 之中多个系统和服务之间拷贝或重编路由消息，应该考虑到为信息创建一个定义的桶容器。实现 CEP 可能需要透明化，但 CEP 与合适的库一起操作起来将更好。

(作者: *Mike Pontacoloni* 译者: 杨晓明 来源: TechTarget 中国)

原文链接: http://www.searchsoa.com.cn/showcontent_29192.htm

如何使用“正好”的复杂事件进行 CEP？

机遇和威胁都是有形的，但是威胁更多一些。企业全力以赴地去开拓机会，但是，威胁需要最先关注。这个方案在采用复杂事件处理软件的早期就发生了——早期，欺诈行为已经成为一种应用类型，排队得到 CEP 的处理。

当然，欺诈检测代表 CEP 的最主要的金融服务的外部入侵。为什么 CEP 在华尔街取得的成功是显而易见的呢：世界金融市场主要都是计算机化，存有大量的信息，可以合并为事件、分析并遵照程序获得利润。

即使面对 2008——2009 年的毁灭性的信贷危机，CEP 系统也能在华尔街给予方法，找到机会并赚钱。华尔街的机会比其他地方多，但是机会和欺诈是相辅相成的。它仍然是宇宙中最主要的地方，并且有大量的复杂的事件。

华尔街是一个特殊的情况，CEP 正慢慢向高级地区扩张。尽管如此，在主街的企业都想用现代化系统技术，开拓自己的机会。CEP 似乎是实现这一目标的手段。

华尔街之外

华尔街和缅因街的不同是显而易见的。在金融市场，向华尔街提供的大部分数据都是恰当地，如果你需要加上一个特殊的转变，它也许会找微调基金。

需要特殊的算法来分析事件数据吗？能够做到！一些成功的交易可以为火箭科学家支付运费，并创建分析过滤器。

但是，当你离开华尔街，事情就变得不同了。CEP 的实施能够威胁自身。这些是有危险的项目，难以量化的经济利益。在开发团队中，本想做复杂事件的人们可能会认为这是一个时机，建立他们自己的程序。

课程导航

CEP 应用程序包含大量高级技术，值得应用开发团队去思考。就数据点而言，你所拥有的关系数据、消息、HTTP Web 监听等聚集到“事件”。事件可以用软件对象、规则和 SQL-like 和 not-SQL-like 的程序语言进行处理。显然，“技术蠕变”是一个不断迫近的威胁，等待伏击项目。

最好的办法是基于应用常识——对不起，这里没有惊喜。当开始 CEP 一段时间的时候，围绕着“多少个 CEP 正好”的问题出现了。

CEP 架构师需要衡量多少个复杂的事件正好，什么样的更新速度正好，包含多少种新技术正好。在一天结束的时候，你用“正好的复杂事件”会做的更好。

(作者: *Jack Vaughan* 译者: 刘志超 来源: TechTarget 中国)

原文链接: http://www.searchsoa.com.cn/showcontent_43121.htm