计算机科学1999Vol.26№.12

基于信息传送和事件驱动的分布式工作流系统

Distributed Workflow System Based on Message Transaction and Event Driven

郑白桦　袁潜龙

（浙江大学人工智能研究所　杭州 310027）

Abstract　Distributed_WorkFlow_Management is a software system used to control and coordinate the PE that operates in environments distributed across multiple organizational entities. In this paper we introduce two typical distrubuted_workflow_systems.

Keywords　Distributed workflow, Message transaction, Event driven

1　引言

传统的工作流（如 Actionworks Metro 与 XSoft's InConcert）都建立在客户机/服务器的模型上，由服务器提供整个系统绝大多数的功能性能。这种结构有它的优点，如集中式的管理、简单的同步机制，整个系统设计复杂度的降低等。但是科学技术的飞速发展，使得企业的业务流程所牵涉的信息资源从以往的以计算机为中心过渡到以网络为中心，工作环境也从单个工作组上升到整个企业乃至多个企业，这就要求工作流程能跨越广域网上的多个服务器和客户端，通过并行计算、资源共享和分布操作等手段来完成整个工作流，传统的工作流模型很明显不能适应这种发展趋势，并成为复杂流程实现的瓶颈所在，所以人们要求一种全新的分布式工作流模型来支持逐渐复杂的工作流程。就目前的分布式工作流模型而言，集中式的服务器已演变为分散在广域网上的多个服务器，每个服务器只需对本地的客户机提供帮助，服务器之间通过网络技术来联系。每一客户机的功能都具有一定的独立性，若干个客户机可以实现一个子流程或某一步骤，力求降低客户机对服务器的依赖程度，减少客户机为获取必要的信息而与服务器进行交流。随着业务流程的复杂化，分布式工作流将成为工作流研究领域中日益热门的话题。

2　研究情况简介

目前工作流管理的目标是研究与开发具有分布式开放体系结构的、组件集成化特征的、并支持群件成员间协同工作的工作流管理环境。借助分布式计算技术、多媒体技术、通信技术以及 Internet 网络技术，工作流模型正逐步向群体性、交互性和协作性的分布式工作流管理系统发展。随之也提出了一系列可行的解决方案，如分布式对象计算、分布式异构环境下的软件组件技术等。

在世界范围内，较著名的研究机构和相应的研究方向主要包括以下几个：

● IBM 公司：在已有产品 FlowMark，MQSeries 的基础上，着手研究基于"信息传送"的分布式工作流系统 Exotica/FMQM；

● 瑞士苏黎士大学：将工作流建立在事件和服务的基础上，并提供了支持分布式工作流的 Broker/Services 模型以及它的实现平台 EVE，还定义了相应的语言用于描述工作流，以保证工作流实现的正确性；

● 美国麻萨诸塞州大学：主要研究 CREW（Correct & Execution of Workflow）项目，努力解决并发事件对共享资源的影响，提供了错误处理机制，力图使工作流能协调进行；

● 北美卡罗莱纳州大学：主要研究工作流的管理，在分布式工作流方面主要致力于并发调度，并定义了一套辅助的形式化语言；

● Matshushita 实验室：设计了一个 INCAS 模型，将具体执行步骤中所牵涉的信息资源保存在一个对象中，即信息载体。

同时，1993年在欧洲还成立了工作流管理联盟（Workflow Management Coslition），推行工作流管理的标准化。

3　分布式工作流系统的实现机制

分布式工作流的研究主要是提供一个开放系统，使用户能透明地应用由不同机型、不同运行平台组成的异构型计算资源，借助信息共享的分布式技术完成一个业务流程。不同的研究机构由于研究方向和依靠

技术的不同，他们所提出的具体实现系统也有所不同。IBM 公司的基于"信息传送"的分布式工作流系统[4]和苏黎士大学的基于"事件驱动"的系统[5]颇具典型性和可行性，在以下的篇幅中将详细介绍。

3.1 基于"信息传送"的分布式工作流系统

Exotica/FMQM (FlowMark on Message Queue Manager)系统是美国 IBM 公司提供的一个分布式工作流管理系统。它借助于一系列 IBM 现有产品(FlowMark 和 MQSeries)，以"信息传送"为实现机制。

FlowMark 模型中定义了一系列相关活动，每个活动有各自的输入、输出存储器，主要用于保存数据，对于一个特定的输出存储器还定义了一个数据流，用于说明输出数据的流向；相关活动之间由控制流来联系，每个控制流通过自身的状态(true 或 false)来决定下一个活动是否该开始运行。

FlowMark 可以在多个平台上运行，如 AIX、OS/2 和 Windows 等。服务器之间以及服务器和客户机之间可以通过不同的网络协议来传送信息，如 TCP/IP、NetBIOS 或 APPC 等，协议的选择主要是依赖于不同的平台。但是近几年来，人们开始要求不同的平台之间能进行交流，而不管彼此的网络协议是否相同，Message Passing/Queuing 便是一个提供这种服务的最普遍的方法。它通过在本地建立一个信息队列，应用程序可以在队列中存入或取回信息。当一信息被存入队列时，通讯系统将负责将之交付给位于远方的某一机器的队列中。由于传送只局限于队列，所以它可以独立于网络协议进行通讯。目前，一系列的相关产品已经问世，如 DEC 公司的 MessageQ、Transarc 公司的 Recoverable Queuing Services、Novell 公司的 Tuexdo/q、IBM 公司的 MQSeries 等，面向信息的中间件联盟机构正在努力使该机制标准化。

Exotica/FMQM 系统便是 FlowMark 模型与 Message Passing/Queuing 的有机结合，它拥有一系列自主的客户机，每个客户机有相对独立的功能，能单独实现流程中的一个或多个步骤，而不用与服务器进行频繁的信息交换，从而克服了传统的工作流实现技术由于对服务器的依赖性而产生的性能瓶颈问题。客户机之间的交流是为了确定某个步骤的实现与否，主要靠信息来传送。该系统将工作流的执行过程分为建模和执行两个阶段，分别由建模客户机、运行客户机和运行服务器来负责。

●建模阶段 系统为建模客户机提供一个友好的用户界面，即用户可以借助该图形界面来定义他所需要的工作流。然后该客户机将用户定义的具体流程进行编译，即将流程划分为几个部分，每个部分又包括若干个具体的活动。通常每个运行客户机都负责一个部分，多个运行客户机之间通过客户机管理者进行信息交换。当编译工作结束后，运行客户机从建模客户机上获取必要的静态信息(工作流的具体说明)，然后由管理者建立一张"流程表"，用于记录该客户机与工作流程的关系、诸如流程和活动的标识符、控制流和出口条件等。然后一个"流程线程"便开始运行，它主要负责在一个流程的执行过程中各个运行客户机间的协调关系。每个运行客户机都有一信息队列，实现控制信息和数据的交换。因此"流程线程"首先要为该运行客户机建立该队列，然后检查该队列所拥有的信息是否满足"流程表"中规定的条件，从而决定是否进入执行阶段。

●执行阶段 当一个流程的前期工作准备完毕，便开始进入执行阶段。该阶段，每个运行客户机都有一"实例表"，主要记录一个流程的多个实例在运行时各个活动的运行情况，包括诸如进程标识符、实例标识符、活动标识符、实际输入输出数据、活动状态以及开始条件的值。

按照 FlowMark 模型，多个部分之间的前后次序是由控制流决定的，所以首先应执行那些不受控制流控制的部分(即流程的开始部分)。然后将输出数据和相应的控制信息按"流程表"的定义送入相应的队列。当某一运行客户机的"流程线程"发现队列中有新的信息，便创建一个"活动线程"，开始一个具体的活动，同时在"实例表"中建立相关项目，用于记录该活动的详细执行情况。该"活动线程"通过检查开始条件来决定是否可以开始运行。若条件还不能计算，即有些控制信息仍未接收到，那么便在"实例表"之该线程的开始条件中标明已取得的控制信息，然后该线程便处于"不活动的"睡眠状态，直到"流程线程"因发现新的信息而将它唤醒。若条件失败，则该活动便被认为已经终止，同时"死路"的信息被传送到相应的队列。若条件成功，便执行该活动。当执行完毕时，检查是否符合出口条件，若不符合则再次执行，直到符合为止。最后将输出数据送入相关的队列，同时将自身队列中的信息和"实例表"中的相应项目删除。

3.2 基于"事件驱动"的分布式工作流系统

苏黎士大学将一个工作流看成由若干个服务组成，每个服务都通过若干事件的完成来提供，而事件的实现可由客户机来负责。如果某个客户机在执行过程中需要得到其它客户机的帮助，它可以发出服务请求事件，然后由服务器通知相应的客户机，该客户机就负责提供这一服务并在事件完成后发出服务应答事件。如此反复，多个客户机协调合作来完成一个工作流。为了实现这一事件驱动机制，他们提出了 Brokers/Services 模型和 EVE 平台。

Brokers/Services 模型是将一个工作流实现过程

中的若干个事件指定给特定的经纪人(Broker)来负责实施。多个经纪人之间通过不同的服务要求来取得联系，从而完成一个工作流程。在该模型中，所建议的一个完整的工作流管理模型(WfMS)应包括三部分：部件(components)：代表该工作流程中的处理实体；任务(tasks)：代表由部件执行的任务；规则(rules)：代表在何时或什么条件下，部件可以或必须执行某一任务。

Broker 即以上所介绍的"部件"，它代表一个特定的处理实体，拥用特定的功能，通过自身功能的实现给别的经纪人提供服务。当它需要某一服务时，则可以通过"服务请求"来要求别的经纪人提供。WfMS 的一系列任务用 Services 表示，每个服务都有各自的签名，包括服务名称、所需的参数以及可能导致的异常与应答。每个经纪人的行为由"事件-条件-行为"(ECA)规则来规定。ECA 规则的具体格式如下：

```
ON
service_request_event(service_parameters)
[IF condition]
DO execute service provision actions;
    generate reply event;
```

该规则表示，如果有一要求服务的事件发生并且条件允许，那么提供该服务的事件就被执行，并产生相应的应答事件。

一个事件可以是单一的，也可以是合成的。某一特定的服务可由一个或多个经纪人来完成。一个服务在执行过程中，如果产生了应答或异常，则该服务被认为是终止了。下面通过一个实际的"要求支付健康保险"的例子以说明如何用该模型来描述工作流。

如图1所示，保险代理机构首先接收一份要求支付健康保险的申请单，建立一个包括病情诊断结果、治疗方法以及所需费用的文件(活动1：接收)，然后审核顾客所申请的保险金额(活动2：审核)，如果金额少于某一定值并且申请项目在保险范围之内，该申请就被接受同时打印支付支票(活动3：打印支票)；否则就将该申请转交给保险公司的票据交换所来处理，主要由子工作流程 SW1 负责，它包括两个并行的活动：在公司的数据库中检查申请单上所注明的治疗是否都在保险范围之内(活动5：检查范围)；由公司的某位医学专家来判断该治疗方法是否适应该病情(活动6：专家判断)。如果医学专家判断失败，则在该顾客的历史记录中作好登记(活动8：不合理申请)，再打印出申请拒绝单(活动4：拒绝)；否则也需在该顾客的历史记录中作好相应的登记(活动7：合理申请)，然后接受该申请打印出支付支票(活动3：支付)。

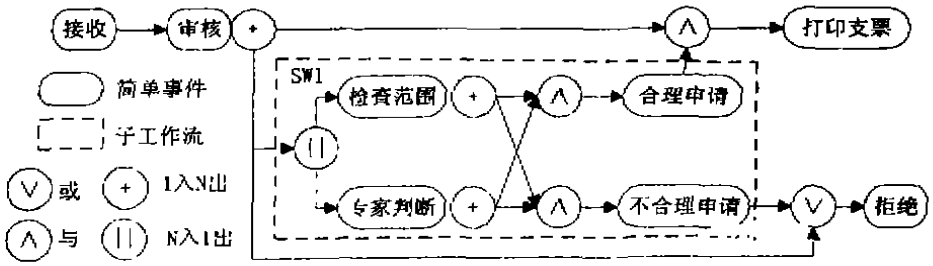

图1 工作流实例

该实例中，我们定义了以下几个经纪人：

●经纪人 Meler 和 Muller，代表保险代理人，可以处理活动1：接收；

●经纪人 HICEF，代表一实际应用，可以处理活动2：审核，并创建一新的 HIC 文件；

●经纪人 DBCL，代表数据库客户机的应用程序，可以处理活动5：检查范围、活动7：合理申请和活动8：不合理申请；

●经纪人 Johnson，代表公司的一位医学专家，用于处理活动6：专家判断；

●经纪人 LPF，代表打印机，可以处理活动3：打印支票和活动4：拒绝；

通过以上的例子，我们可以得知 Brokers/Services 模型的大概。它较适合分布式工作流系统，一台客户机或客户机上的某一实际应用程序就可以模拟一个经纪人，工作流中的各个活动可用事件来代替。而多个经纪人之间的合作可借助 EVE 平台来实现。EVE 即事件驱动机器(Event engineer)，主要负责发现事件，通知相应的经纪人来执行相关事件等。为了适合分布式工作流的特点，EVE 提供了以下几个功能：

●分布：由于分布式工作流所牵涉的资源往往处于不同的地点，EVE 将位于同一地点的经纪人用局域网连接，而局域网之间的经纪人可以通过"适配器"(EVE-Adapters)与局域网的服务器取得联系，多个局域网的服务器又借助网络来联系，形成一个广域网；

●知识库(Runtime Repository)：用于保存工作流

的某一实例在运行过程中所需的信息，如工作流的类型（诸如名称，开始与结束事件，活动的与已结束的实例等），事件的类型，经纪人的信息（诸如负责的服务器等）和有关的 ECA 规则；

●事件的发现和通知：负责发现新的事件，将事件分配给某一经纪人并通知经纪人开始该新事件，事件分配策略包括任意选择、最佳选择、最少选择和综合策略；

●事件历史的管理：以时间为线索，一系列的事件就记录了过去发生的一切，而这些事件记录就组成了事件历史，可以提供有关经纪人活动和工作流执行过程的关键信息，是管理、分析以及优化工程流的数据库。它一般分布于 EVE 的各个服务器上，每一事件在被发现后应作好登记，以便保存在事件历史中。

现在介绍一下 EVE 的分布式工作流实现机制。由于当前所讨论的工作流是由事件驱动的，所以如何发现事件并通知相关的经纪人便成为首要任务。通常由位于 EVE 上的"Event Detector"（ED）负责发现事件。事件分为简单事件类型和合成事件类型。简单事件类型包括服务请求事件类型、请求证实事件类型、应答事件类型和异常事件类型；合成的事件类型则包括连接事件类型、排斥事件类型、串行事件类型、并发事件类型、否定的事件类型以及反复事件类型，实际应用中的一个事件，则是以上所定义的事件类型的一个实例，当该工作流开始运行后，负责启动的经纪人就通知服务器它将开始启动事件，服务器接到通知后就将该信息转交给 ED 来处理，然后 ED 就将该事件的有关信息记录下来。在 ED 中，用事件图来代表相应的事件。简单事件用单一结点来代表，合成事件用结点树来代表，树中结点的父子关系就代表了事件的合成关系。如一合成事件定义为((ED1 and ED2))and ED3，如图2所示。

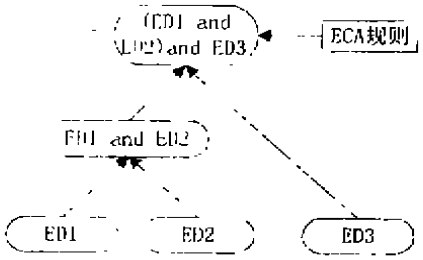

图2　规则结点树

只有简单事件被发现后，合成事件才有可能产生。同时，在 ED 的事件图中，事件结点还带有相应的 ECA 规则，当一个事件结点被发现后，与之相应的 ECA 规则被执行。通常 ED 是按照时间先后顺序来执行 ECA 规则。当 ECA 规则被执行后，新的服务要求就产生了，然后 ED 利用事件分配策略来找到一个合适的经纪人负责提供该项服务，并通知该经纪人。于是，该经纪人接到通知后，就开始执行某事件，ED 则再次执行以上所介绍的步骤，直至结束。而整个 EVE 的运行原理也基本如此，可参照图3。

由于存在合成的事件类型，则一个合成事件可以由位于不同地点的简单事件组成。但是，每个地点可能都拥有一个自身的时钟系统，那么 ED 在执行 ECA 规则时就可能遇到无法判断先后的问题。为了实现整个系统的协调合作，EVE 提供了一个公共的时间环境即时间戳。针对不同地点 S 的当地时间，系统提供了一个时间转换功能，负责将一个局部的时间转换为一个公共的全局时间。一个事件的时间戳即该事件发生的全局时间。这样，ED 就可以顺利执行它的任务了。

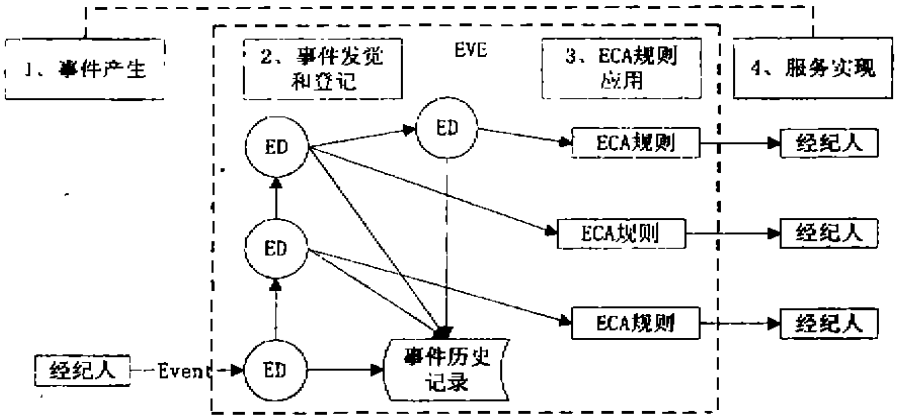

图3　EVE 运行原理

3.3 两者之比较

以上介绍的基于"信息传送"的系统和基于"事件驱动"的系统都是一个可行的分布式工作流系统，但两者各有特点。

IBM 公司的基于"持续性信息传送"机制的一个最大优点是增加了系统的错误恢复能力。它强调了每个结点的功能独立性，使得一个结点的失败只影响到与该结点有关的相应工作流实例或某一工作流步骤，从而将影响范围缩小到最小范围，使得系统构造方面有了更好的升级性和适应性。但是它没有提供一个类似于"事件历史"的历史记录，所以该机制现在正在研究如何使用日志文件来管理工作流，并尝试用非持续性的信息传送来提高性能。

苏黎士大学所研究的基于"事件驱动"的机制主要是开发了 Brokers/Services 模型和 EVE 平台，从而将分布式工作流的实行分为三步[3]：首先由高层的图形界面提供工作流建模工具，然后由中层的 Brokers/Services 模型执行工作流，最后由底层的 EVE 平台提供分布式工作流的实行框架，主要负责事件管理、历史记录和工作流执行者之间的交流。B/S 模型提供了一个较完善的语言定义机制，保证建模人可以了解工作流执行者的实际行为，组成工作流说明的工作流执行过程可以被正确定义、工作流执行过程的准确性得到保证，工作流执行完毕后的分析也成为可能。但是，对活动的工作流或正在运行的组成部件进行修改所带来的影响还没有得到很好的解决，这将是以后的研究方向。

结束语 虽然目前的研究取得了一定的成绩，分布式工作流的实现机制在某种程度上也开始发挥作用，但是这些模型还是十分理想化的，如现今的 EVE 平台还不能处理异常的恢复，在某些长时间运行的工作流应用例子中，工作流类型的发展变化问题也仍未得到解决。所以我们还要继续努力，利用现有的理论研究成果进行深入研究并逐渐完善之，力求开发出一个能够真正支持分布式工作流的系统。

参 考 文 献

1 Alonso G, et al. Exotica/FMQM: A Persistent Message-Based Architecture for Distributed Workflow Management

2 Feppert A, Tomtros D. Event-based Distributed Workflow Execution with EVE: [Tehnical Report] 96.05

3 Tombros D, Geppert A. Semamtica of Reactive Comopoment in Event-Driven Workflow Execution, Institut Informatits Universitat Zurich

（上接第30页）

与社会发展中的应用。

4 小波分析的发展前景

小波分析是科学家、工程师和数学家们共同创造的，反映了大科学时代学科之间的综合、渗透的优势，小波理论来自傅里叶分析，思想也来源于傅里叶分析，它不能完全取代傅里叶分析，它是傅里叶分析的新发展。小波理论与傅里叶分析的互补优势和相辅相成的良好效果已被科研实践所证实。小波分析的发展一方面需要从理论上提高和丰富，尤其是三维和三维以上的小波理论（因为它们还很不成熟），另一方面需要在应用中提出更多的研究课题，使小波应用的深度和广度得到进一步拓展。

尽管小波分析形成了目前国际研究热点，但我们不能认为小波分析是包打天下的有效工具。例如，在进行数值分析时，采用小波展开的优点是能够较准确地算出 f(x) 的精细结构，但付出的代价是代数方程阶数增大，条件数变坏，因此小波分析仅仅作为一个基来使用是很不够的，更重要的是作为数值计算后处理的分析手段。

由于小波理论处理问题的特殊技巧和特殊效果，小波分析不仅为纯数学与应用数学提供新的强有力的工具，而且是多媒体、信息高速公路某些核心技术的理论保证。

参 考 文 献

1 李建平. 小波分析与信号处理——理论、应用及软件实现. 重庆：重庆出版社，1997

2 李建平. 矢量积小波变换及小波分析的理论与应用研究：[博士学位论文]. 重庆：重庆大学，1998.6

3 李建平、唐远炎. 小波分析方法的应用. 重庆：重庆大学出版社，1999