

# 基于 SOA 的事件协同感知模型

刘思中<sup>1</sup>, 曹健<sup>2</sup>

(上海交通大学计算机科学与工程系, 上海 200240)

**摘要:** 在现有面向服务架构(SOA)实现中, Web 服务之间无法直接协同。为了完成协同工作, SOA 的各个构件必须具有对系统全局的感知能力。该文采用一个事件信息组件, 在 SOA 中引入事件驱动架构, 搜集各个 Web 服务的调用信息, 利用复杂事件处理技术对其进行整理后, 发送给 SOA 构件订阅者, 达到协同感知的目的。

**关键词:** 面向服务架构; Web 服务; 协同感知; 复杂事件处理; 事件驱动架构

## Event Collaborative Awareness Model Based on SOA

LIU Si-zhong<sup>1</sup>, CAO Jian<sup>2</sup>

(Department of Computer Science & Engineering, Shanghai Jiaotong University, Shanghai 200240)

**【Abstract】** In the current SOA implementation, Web services can not collaborate with each other. In order to work collaboratively, the component in the SOA must have the awareness about the activities in the system. This paper adopts an event infrastructure, which introduces the Event Driven Architecture(EDA) into the SOA. This event infrastructure collects execution information of the Web service, uses the complex event processing technology to process this information, and sends it to the SOA component to realize collaborative awareness.

**【Key words】** Service Oriented Architecture(SOA); Web service; collaborative awareness; complex event processing; event driven architecture

### 1 概述

激烈的市场竞争和多变的客户需求使协同工作成为企业专业化信息应用领域之一。全球化趋势决定企业与企业之间必须协同工作, 才能在市场上发挥竞争优势。面向服务架构(Service Oriented Architecture, SOA)是一种面向服务的 IT 架构, 互联网的发展促使其被各个企业广泛采用。

现有 SOA 实现不完全具备支持协同工作的能力, 主要表现在以下 3 方面<sup>[1]</sup>: Web 服务缺乏自主性和相互协同能力, Web 服务缺乏语义表达能力, 服务过程模型不能满足动态业务环境要求。

协同感知是协同系统的核心技术之一, 在企业应用中, 用于感知各个活动的进展、各种信息和环境的变化, 是实现快速决策行动和跨组织协同工作的基础<sup>[2]</sup>。事件机制能自动捕捉各个协同应用产生的事件信息, 对其进行处理并通知订阅方, 为协同感知提供技术基础<sup>[3]</sup>。SOA 环境为统一采集各种信息奠定了基础, 通过在 SOA 中植入获取和处理信息的组件, 可以为基于 SOA 的系统增加协同感知功能。

为了实现 SOA 中的协同感知, 本文介绍了一个基于 SOA 的事件协同感知模型, 它可以将服务的调用映射成事件, 通过对事件的收集、处理和发布, 使 SOA 的各个构件具有全局协同感知能力。

### 2 协同感知与事件驱动架构

目前, 协同感知研究的 3 个重要内容如下: 协同感知信息源, 协同感知信息管理和协同感知的信息效果表示。

事件驱动构架(Event Driven Architecture, EDA)用于发现系统或周围环境中发生的特殊事件<sup>[4]</sup>, 并快速发布通知给对此事件感兴趣的订阅方(可以是操作人或智能组件), 从而使订阅方能快速作出反应。

在事件驱动架构中, 事件处理流程按逻辑被分为 4 层<sup>[5]</sup>:

事件源, 事件通道, 事件处理和下游事件驱动活动。架构完整地覆盖了协同感知模型、事件源、事件处理和下游事件驱动活动分别对应协同感知模型中的信息源、信息管理和信息效果表示。事件通道支持架构整体的松耦合特性, 使 EDA 可以轻易集成在在分布式 SOA 系统中。

### 3 SOA 事件协同感知模型

SOA 事件协同感知模型如图 1 所示。

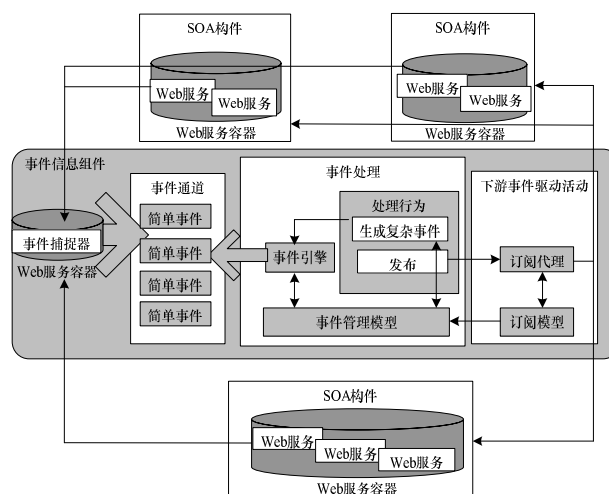


图 1 SOA 事件协同感知模型

本文描述的事件信息组件采用事件驱动架构, 在 SOA 环

**基金项目:** 国家“863”计划基金资助项目(2006AA04Z152); 国家自然科学基金资助项目(60503041)

**作者简介:** 刘思中(1983—), 男, 硕士研究生, 主研方向: SOA 中的协同系统; 曹健, 副教授

**收稿日期:** 2008-10-26 **E-mail:** sizhong.liu@gmail.com

境下,收集 Web 服务执行时产生的基本信息,并把处理后的信息发布给订阅方。部署于 SOA 中的各个构件,根据所收到事件的内容,可以得知协同系统中的全局信息,从而根据各自的协同协议作出反应,完成协同工作。

在系统模型中,事件源即 SOA 环境中的各个 Web 服务容器。为了在每次 Web 服务被执行时发送调用信息给事件捕捉器,每个 Web 容器必须植入相应代码。事件捕捉器的功能是根据从其他 Web 服务容器处得到的信息生成格式统一的简单事件,并将其放入事件通道中。事件捕捉器本身是一个 Web 服务,它提供统一接口供其他 Web 服务容器调用。

事件通道通常是一个消息传递网络,在事件源和事件处理引擎之间传递格式标准化的事件消息。

事件处理是事件驱动架构的核心,实现协同感知中的信息管理。它利用一个复杂事件处理引擎,处理搜集到的基本信息,从而得到更有价值的或用户感兴趣的信息。

事件感知模型中的下游事件驱动活动主要负责将得到的复杂事件信息发送给各个订阅者。订阅代理从订阅模型中获取与复杂事件相符的订阅者名单。订阅者必须实现某个接口,以接收订阅代理发送过来的复杂事件。

事件管理模型模块主要负责对事件元数据的管理。事件元数据包括事件定义、事件模式匹配条件和处理规则。

订阅模型模块主要记录复杂事件与订阅者的对应关系,并负责告知订阅代理,该向哪个订阅者发送复杂事件信息。

#### 4 系统核心的实现

事件信息组件是按 EDA 架构思想设计的,是一种松耦合分布式结构。其中各个模块部署于不同节点中,以完成事件捕捉、处理和订阅通知等系列工作。

图 2 描述了事件信息组件的 5 大模块:事件捕捉器,事件通道,事件引擎控制,元数据 XML 操作,订阅管理。上述 5 个模块相对独立,以松耦合为目标。

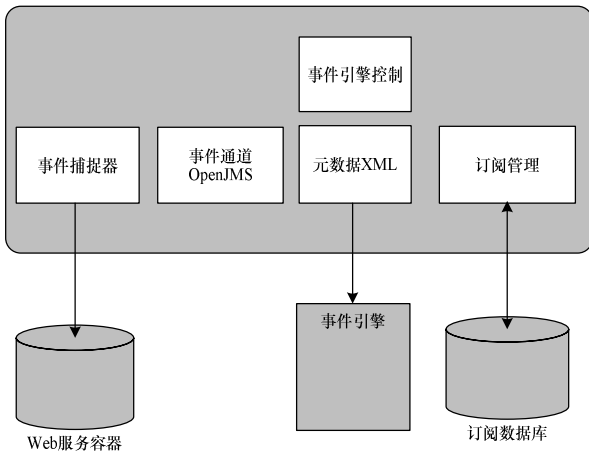


图 2 事件信息组件的模块结构

当 Web 服务被执行时,一组具体 Axis Web 服务容器的插入代码负责通知事件捕捉器,它会生成最基本的原子事件 ServiceEvent,并把它放入事件通道中。

事件通道由 OpenJMS 实现,OpenJMS 是 JMS 的一个开源实现,能模拟 ESB 中的消息传递,并可以在扩展中被其他消息中间件取代。

事件引擎的管理是事件组件的核心。本文设计的事件信息组件采用 Esper 事件引擎,并通过一系列元数据 XML 文档对引擎进行配置。在其上层的事件引擎控制模块中,通过修

改元数据 XML 文档可以达到控制事件引擎的目的,从而提高复用性。

事件引擎整理并生成复杂事件后,订阅管理模块将搜索订阅数据库,并把这些复杂事件通知相应的订阅者。订阅者以 Web 服务的形式在数据库中注册,通知订阅者的具体行为可以理解为调用此 Web 服务。

事件信息组件不是一个单节点组件,而是部署在整个 SOA 环境中的组件实现,需要与原有 SOA 组件进行交互。在具体的交互点,即事件捕捉、事件通道、事件订阅这 3 个模块上,通知以 SOA 实行的标准,如 Web 服务或 SOA 能够兼容的形式(如消息传递模式)实现。这为事件信息组件与 SOA 的应用整合打下了良好基础。

#### 5 应用案例

##### 5.1 案例介绍

通过一个简单案例来具体认识现代电子商务中的协同工作。该案例以某公司的购货业务为需求场景。为了进一步规范购货业务过程需求,图 3 描述了一个以 Web 服务为背景的电子商务案例流程。其中共有 3 个参与流程的角色:客户,公司和银行。每个角色提供了一些 Web 服务以完成购货业务。

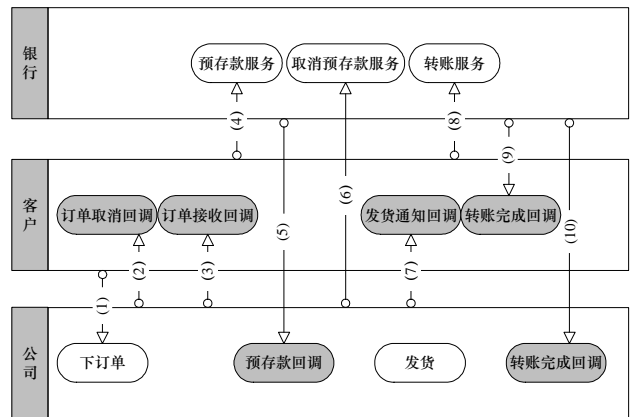


图 3 电子商务案例流程

由图 3 可以看出,各个角色可能分布于不同地点,拥有各自的服务和操作规范。在现有实现中,一般利用 BPEL4WS 技术将整个流程封装成一个组合服务并统一发布。例如在本案例中,BPEL4WS 被部署在公司或第三方服务中。BPEL4WS 方式存在以下不足:

(1)其实质是通过一个中心协调者对整个流程进行控制。各个 SOA 构件无法自主地根据自身逻辑对流程进行改进和控制。

(2)其异步回调 Web 服务的方式破坏了 Web 服务封装的逻辑完整性,增加了 SOA 构件之间的耦合性。

(3)以一种硬编码方式进行服务组合。在过程制订时,无法动态地快速部署、满足 SOA 的敏捷性需求。

##### 5.2 事件信息组件的应用

为了解决 BPEL4WS 的不足,SOA 各个构件的工作必须以一种对等协作的方式展开。作为其基础,SOA 各个构件必须对其他参与协同工作的构件具有全局感知能力,以了解协同工作的进行状态。

事件信息组件通过收集各个构件中 Web 服务的执行情况,获取关于协同工作的状态信息,并将其发布给参与协同工作的其他构件。图 4 描述了电子商务案例中事件信息组件的应用。

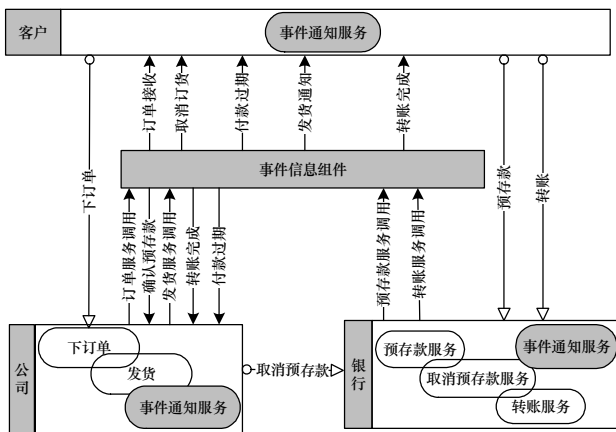


图4 电子商务案例中事件信息组件的应用

由图4可以看到,各个SOA构件保持着独立的控制逻辑,一系列Web回调服务被取消,使用一个统一的事件通知接口接收来自事件信息组件的事件信息。

当各个SOA构件中的Web服务被调用时,调用信息被发送到事件信息组件中。根据这些调用信息,事件信息组件可以提炼包含在信息中的语义含义,并以事件形式发送给参与协同的其他SOA构件,协助其执行协同协议框架下的流程事务。

与BPEL4WS的方式相比,引入事件信息组件具有以下优点:(1)流程的执行不是以一个组合服务的形式出现,各个SOA构件能包含在一个协同协议的框架内,客户端能个性化定制内在流程。(2)取消了Web回调服务,减少了各个SOA构件之间的耦合性。(3)事件中的语义信息支持SOA构件自主地进行个性化定制、Web服务发现编排等智能化工作。

## 6 结束语

本文所述事件感知模型能收集SOA环境下Web服务执行时产生的信息,并根据复杂事件处理的方式,对事件进行

处理,得到有价值的信息。利用带有语义的事件信息,SOA构件可以减少相互间的耦合度,并了解协同工作状态,从而完成协同工作。

事件感知模型的处理过程基于复杂事件,是一个递归运算,各个信息之间不是相互独立的,因此,能更有效地析取有价值的信息。

下一步工作将分2步进行:(1)改进事件信息组件的各个模块;(2)利用现有事件信息组件实现来完成一个分布式事件处理模型,解决在开放式SOA环境下的事件处理和订阅问题。

本文设计并实现的事件信息组件为EDA与SOA的结合提供了一个尝试性方案,并为SOA进一步的扩展、应用与研究奠定了基础。

## 参考文献

- [1] 曹健,李明禄,张申生.基于多Agent协商的服务流程定制[J].计算机学报,2006,29(7):1116-1121.
- [2] Dourish P, Bellotti V. Awareness and Coordination in Shared Workspaces[C]//Proceedings of CSCS'92. Toronto, Canada: ACM Press, 1992.
- [3] Li Yushun, Gong Neng, Shi Meilin. A New Collaborative Awareness Model and Its Application[C]//Proceedings of the 8th International Conference on Computer Supported Cooperative Work in Design. Beijing, China: [s. n.], 2004.
- [4] David C L, Brian F. Complex Event Processing in Distributed Systems[Z]. Computer System Lab, Stanford University, 1998.
- [5] David S R, Alexander L W. A Design Framework for Internet-scale Event Observation and Notification[C]//Proceedings of the 6th European Conference Held Jointly with the 5th ACM SIGSOFT International Symposium on Foundations of Software Engineering. Zurich, Switzerland: [s. n.], 1997: 344-360.

(上接第9页)

## 参考文献

- [1] Ohta N. Optical Flow Detection by Color Images[C]//Proc. of IEEE International Conference on Image Processing. Singapore: [s. n.], 1989: 801-805.
- [2] Ohta N. How Much Does Color Information Help Optical Flow Computation[J]. IEICE Transactions on Information and Systems, 2006, 89(5): 1759-1762.
- [3] Lai J, Gauch J, Crisman J D. Computing Optical Flow in Color Image Sequences[J]. Innovation and Technology in Biology and Medicine, 1994, 15(1): 76-87.
- [4] Golland P, Bruckstein A M. Motion from Color[J]. Computer Vision and Image Understanding, 1997, 68(3): 346-362.
- [5] Barron J, Klette R. Quantitative Color Optical Flow[C]//Proc. of the International Conference on Pattern Recognition. Vancouver, Canada: [s. n.], 2002: 251-255.
- [6] Horn B K P, Schunck B G. Determining Optical Flow[J]. Artificial Intelligence, 1981, 17: 185-203.
- [7] Lucas B D, Kanade T. An Iterative Image Registration Technique with an Application to Stereo Vision[C]//Proc. of International Joint Conference on Artificial Intelligence. Vancouver, Canada: [s. n.], 1981: 674-679.
- [8] Andrews R J, Lovell B C. Color Optical Flow[C]//Proc. of Workshop on Digital Image Computing. Brisbane, Australia: [s. n.], 2003: 135-139.
- [9] Bruhn A, Weickert J. Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optic Flow Methods[J]. International Journal of Computer Vision, 2005, 61(3): 211-231.