

基于主动对象的全局 Web workflow 系统*

胡 华¹, 俞宏知²

¹(杭州商学院 计算机与信息工程系,浙江 杭州 310012);

²(浙江大学 网络中心,浙江 杭州 310027)

E-mail: huhua@yahoo.com

http://www.softlab.hzic.edu.cn

摘要: 在分布式主动对象的基础上给出了在一个 World Wide Web 上实现的全局 workflow 系统.首先根据 Internet 应用环境的特点与要求,对主动数据库中的 ECA 模型进行了时间和空间上的扩充,从而解决了主动对象在分布式环境下的时域和空域相关问题.其次,采用任务层次分解的方法建立起主动对象控制的 workflow 管理模型.最后,在相关模型和分布事件通信模式的基础上,给出了 World Wide Web 上的基于主动对象的全局 workflow 系统结构及其应用.

关键词: 主动;对象;EECA 模型;状态;任务;层次分解;Web;workflow;控制

中图法分类号: TP393 **文献标识码:** A

现代电子商务应用不仅包括了基于 Internet 网络上的在线商务交易,而且还包括了所有其他利用电子信息技术来解决问题、降低成本、增加价值和创造商机的商务活动.由于一个完整的电子商务活动过程往往涉及到企业内部和外部等诸多方面之间的有效协作,因此,在 Internet 环境下有效地实施能与企业各部分信息系统有效集成且平台无关的电子协作(electronic collaboration)和流程控制自然地成为电子商务应用的一项核心技术手段.

传统的工作流和计算机协同工作系统由于受到处理机制和网络规模的限制,其协作管理与组织运作的对象主要是小范围、被动式的企业内部流程控制管理和协调^[1~3].对于在 Internet 环境中进行的对时间和空间分布有着较高要求的跨平台式电子商务处理,研究和开发一种具有大范围时空协作处理能力的平台无关的全局智能 workflow 集成处理系统已成为现代计算机协同处理的一个重要研究方向^[4].史美林^[5,6]在对 Internet 环境下的 workflow 系统模式进行探讨、研究的基础上,根据 Web 服务器的工作特点给出一个基于 Web 的 workflow 管理控制系统;G. Cabri^[7]等人采用 Java 的 Servlet 方式设计实现了一个基于 Proxy 的 Web workflow 系统;Mahling^[4]提出了一个基于 Internet 的图书销售服务系统;Hull^[8]提出了一个在 Internet 上进行 workflow 描述的 workflow 描述模型;Peter^[9]提出了一个用于分布式 workflow 系统的层次对象协议.

我们根据现代网络技术和主动式数据库的发展特点及应用能力,在以 World Wide Web 为核心的 Internet 应用模式下,对分布式主动对象和 workflow 控制进行了时间和空间的体系结构扩充,并在此技术基础上,开发实现了一个在地理和功能上都广泛分布的面向主动对象的全局 Web workflow 集成控制框架(global Web workflow framework based on active object,简称 GWWFBAO).GWWFBAO 不仅通过 Internet 的全局开放性特点突破了传统 workflow 系统的时空局限性,而且还通过基于主动对象的软件构件包装技术解决了 workflow 系统中普遍存在的实时交互和智能协调处理问题.

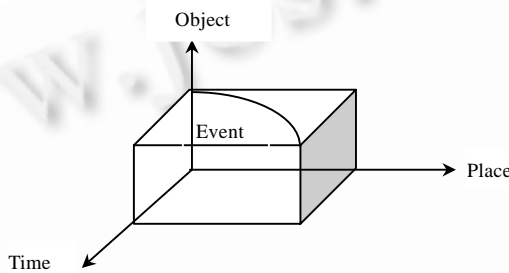
* 收稿日期: 2001-02-05; 修改日期: 2001-06-25

基金项目: 国家教育部优秀青年教师基金资助项目

作者简介: 胡华(1964 -),男,江西南昌人,博士,副教授,主要研究领域为数据库技术,信息系统;俞宏知(1965 -),男,浙江杭州人,讲师,主要研究领域为计算机网络,多媒体技术.

1 基于扩展 ECA 的主动对象

ECA(event-condition-action)模型是主动数据库的研究和应用过程中普遍使用的一种控制和规范主动数据库行为的规则模型.在该模型中,E 代表在数据库所在环境中发生且对数据库状态改变有触发作用的事件,C 代表主动数据库响应触发事件的条件,而 A 则代表主动数据库在条件满足的情况下对触发事件进行响应时采取的行动.通过把原本需要写入应用程序的系统管理和控制策略抽象和分离成为由 Event,Condition 和 Action 有机结合的 ECA 规则知识库,以 ECA 规则为核心的主动数据库系统能够有效地实现数据与知识的分离.尽管如此,ECA 模型却不能直接用于 GWWFBAO 的主动对象.这是因为,GWWFBAO 的主动对象作为一种能够使用主动规则模型来实施系统管理和系统控制运作对象实体,其分布式对象的本质属性,使得其感受环境、响应事件的行为与能力普遍受到随时间和空间进行变化的对象内部状态的制约和影响.正是因为主动对象的这种分布式对象处理所具有的时空相关性特点,极大地限制了 ECA 规则模型对主动对象的控制和描述能力.鉴于此,我们对传统的 ECA 模型进行了时空变化和对象状态描述方面的增强与扩充,使之能够有效地表达分布式环境下主动对象行为与语义.主动对象的时空特性如图 1 所示.下面我们给出扩展的 ECA 主动对象模型 EECA(extended-event-condition-action).



对象, 事件, 位置, 时间.

Fig.1 Time and space specialty of active object

图 1 主动对象的时空特性

设 $\alpha = \{e_0, e_1, e_2, \dots, e_n\}$ 为系统中的所有事件组成的集合, $\beta = \{\chi_0, \chi_1, \chi_2, \dots\}$ 为系统中所有对象的属性空间, $\gamma = \{\xi_0, \xi_1, \xi_2, \dots\}$ 代表对象所处环境(或系统)的属性空间.

定义 1. 主动对象响应的事件是一个三元组 $\{e, t, \rho\}$, 其中, $e \in \alpha, t$ 是如下结构为 *yyyymmddhhmmssms* 时间戳, ρ 为 URL 形式的系统定位信息.

定义 2. 主动对象的内部状态是一个五元组 $\{T, \chi_s, \chi_d, t, \rho\}$, 其中, T 是全局唯一的事务标志, $\chi_s, \chi_d \in \beta$ 分别代表主动对象的不变属性和可变属性, t 与 ρ 的含义同定义 1.

定义 3. 主动对象的环境状态是一个五元组 $\{T, \xi_s, \xi_d, t, \rho\}$, 其中, T 的含义同定义 2, $\xi_s, \xi_d \in \gamma$ 分别代表对象环境的不变属性和可变属性, t 与 ρ 的含义同定义 1.

如果将主动对象的响应事件集合记为 E , 内部状态组成的集合记为 S_{in} , 所处环境状态组成的集合记为 S_{out} , 那么我们定义扩展的 ECA 规则如下:

定义 4. 主动对象的扩展 ECA 规则 EECA 是一个三元组 $\{e, f(s_1, s_2), a\}$, 其中, $e \in E, s_1 \in S_{in}, s_2 \in S_{out}, f$ 是对 s_1 与 s_2 的断言, a 是主动对象应响应的操作的有序集合.

至此,我们可以将主动对象的语义模型定义如下:

定义 5. 一个主动对象类是一个五元组 $\{N, R, S_{in}, S_{out}, A\}$, 其中, N 是一个主动对象的全局命名系统, 它为每个该类主动对象分配一个全局唯一的对象标识, R 是该类对象拥有的扩展 ECA 规则集, S_{in} 是该类对象的内部状态集合, S_{out} 是该类对象的环境状态集合, A 是该类主动对象能够实施的操作集合.

定义 6. 一个主动对象是一个主动对象类的实例, 它可以表示成一个八元组 $\{I, R, T, S_{in}, S_{out}, A, Ta, a\}$, 其中, I 是一个全局唯一的对象标识, R 是该对象拥有的扩展 ECA 规则集, T 是对象经历过的状态的轨迹, $S_{in} \in S_{in}$ 是对象的当

前内部状态, $s_{out} \in S_{out}$ 是对象的当前环境状态, A 是主动对象能够实施的操作集合, Ta 是对象经历过的操作轨迹, a 是对象正在实施的操作轨迹.

此外,在主动数据库的 ECA 模型应用中,一个经常要解决的问题是避免因主动规则递归调用而产生规则执行风暴.对于这种情况,EECA 模型可以通过对对象的事务处理状态轨迹进行有效检验的方式来避免主动规则的递归调用.

2 基于有向图的工作流任务分解

作为一种以主动对象和 World Wide Web 为主体的流程控制系统,GWWFBAO 采用如图 2 所示的一种基于有向图的任务对象层次分解模式来进行工作流程的控制与分析.

根据任务对象的层次分解原理,在 GWWFBAO 中,一个完整的流程处理任务对象可以对应地分解为多个时间有序的子流程处理任务对象,每个子任务对象又可以按需要进一步细分为更细小的子任务对象.从有向图的角度分析,每个任务对象所属的各子任务对象按时间处理顺序组成一个流程处理有向图,图中的每个结点代表一个子任务对象,有向边则代表完成一个任务之后向下一个任务转换.

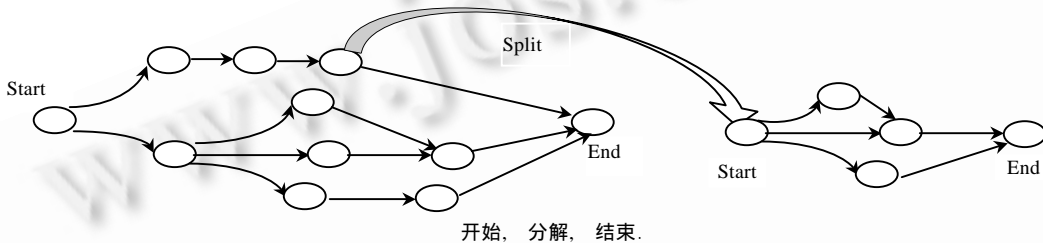


Fig.2 Task splitting on direct graph

图 2 基于有向图的任务分解

此外,考虑到 GWWFBAO 工作环境的分布式处理特点,每一个任务的工作状态应该包括该任务涉及的时间与地点信息.

定义 7. workflow 任务的状态是一个三元组 $\{s, t, l\}$, 其中, $s \in \{start, execute, wait, pending, delay, sleep, overtime, extend, stop, finish\}$ 是代表任务对象的当前执行状态, t 是时间戳, l 是任务的位置.

定义 8. 一个 workflow 任务类是一个七元组 $\{N, O, S, S_{init}, S_{end}, L, G\}$, 其中, N 是一个任务对象类的全局命名系统, 它为每个任务对象类分配一个全局唯一的任务类标识, O 是可以参加该类任务处理的主动对象类的集合, S 是该类任务对象的状态集合, S_{init} 是该类任务对象的初始状态, S_{end} 是该类任务对象的终止状态集合, L 是该任务类所能执行的最大时间跨度, G 是指向由该任务类分解出的子任务类有向图的指针.

定义 9. 一个 workflow 任务是一个由 workflow 任务类生成的任务对象实例, 它可以被表示成一个九元组 $\{I, O, S, S_{init}, S_{end}, T_{start}, T_{end}, G, B\}$, 其中, I 是一个全局唯一的任务标识, O 是参加该任务处理的主动对象的集合, S 是任务对象的状态集合, S_{init} 是任务对象的初始状态, S_{end} 是该任务对象的可能终止状态集合, T_{start} 是任务开始执行的时间, T_{end} 是任务结束的截止时间, G 是指向由该任务的子任务链表的有向图的指针, B 是指向由与该任务同级别的下一任务对象的指针(如果没有下一个同级子任务,该指针为空).

最后,将 workflow 处理抽象成层次的任务分解模式,还使得我们可以采用主动规则的模块化分层技术以缓解系统主动规则增加时产生的主动规则管理与控制上的困难.

3 GWWFBAO 的实现结构

根据前面所述的主动对象模型和流程分解原理,我们对 GWWFBAO 设计了如图 3 所示的系统层次协作结构.

根据系统中各部分的工作性质与处理目标,GWWFBAO 主要由用户支持、网络协议、管理引擎、资料管

理和数据库 5 个结构化模块组成.在这 5 个结构化模块之中,除网络协议外,其他模块都包含了多个相互协作的主动对象.

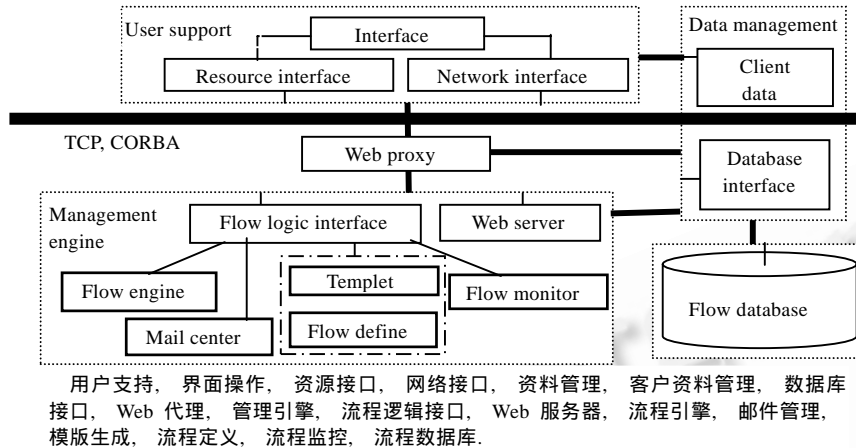


Fig.3 Architecture of GWWFBAO

图 3 GWWFBAO 的系统结构

3.1 用户支持

用户支持模块主要工作在流程控制管理系统的客户端,它由界面操作对象、资源接口对象和网络接口对象等 3 类相互牵制的主动对象组成.其主要功能是为流程处理的用户提供客户端的操作与管理能力.其中,界面操作对象工作在整个模块的最前端,它在提供用户处理的操作界面的基础上,接受、响应和转发用户与系统之间的各种处理和um制指令;资源接口对象工作于界面操作对象之后,其主要功能有两个方面,一方面是在用户界面对象和流程管理/控制资源之间的提供各种请求与处理的统一协调接口;另一方面是通过在线运行和邮件管理来感受流程服务器发给客户的流程处理状态和指令;网络接口对象在 TCP 协议的基础上,采用 CORBA 对象交互机制,为需要网络操作的客户端对象提供事件与信息的网络传输与包装支持.由于 GWWFBAO 是一个基于 World Wide Web 的控制与管理平台,用户支持模块的实现应与现行的各种 Web 浏览器兼容.

3.2 管理引擎

管理引擎模块主要工作在流程控制管理系统的服务器端,它由 Web 代理、流程逻辑接口、流程监督、流程引擎、流程定义、模板生成、邮件管理等一系列主动对象和 Web 服务器组成.管理引擎模块的主要功能是为流程处理提供驱动、控制与管理.其中,Web 代理对象工作在整个模块的最前端.它的工作可以分为系统输入和系统分发两个方面.在系统输入方面,Web 代理对象接受用户从网络发来的系统外部的处理请求,并根据请求的内容与性质,将用户的请求分别转发至 Web 服务器、流程引擎接口对象和数据库接口对象进行处理;而系统分发功能指的是 Web 代理对象接受来自于 Web 服务器、流程引擎接口对象和数据库接口对象等系统内部发来的数据信息和处理请求,并根据数据或请求的性质与内容转发至相应用户(通过网络)或系统对象.不仅如此,Web 代理还能够与客户资料管理对象一起协作完成用户的身份认证过程.流程逻辑接口对象是流程核心处理机制与外部处理环境交互的中介.流程监督对象根据一定的规则与逻辑自动经由流程逻辑接口对象实施对系统中的每一个执行流程进行监督与协调,不仅如此,该对象亦可为系统管理人员提供流程处理的人工监督与协调功能.流程引擎对象是流程运作执行的核心,该对象根据流程处理的当前状态,采用自动驱动和人工介入两种方式管理和控制流程的正确运行.其中,自动驱动方式是根据事先定义好的规则与模式管理驱动流程,而人工干预方式则为人工介入处理意外情况提供了接口与手段.邮件管理对象根据流程引擎的指示和系统内部的当前状态,为流程处理过程中的信息交换提供内部的邮件管理功能.最后,流程定义和模板生成对象主要提供流程模板的生成和管理功能.

3.3 资料管理

资料管理由客户资料管理对象和数据库接口对象两个对象组成,客户资料管理对象主要运行于 workflow 操作的客户端,其主要任务是在管理客户端的用户基本资料和信息的基础上,与 Web 代理和数据库接口对象一起完成对操作客户的身份认证处理.数据库接口对象为其他对象的数据库处理请求操作提供操作的一致性接口.

3.4 数据库

运行于服务器后台的数据库管理系统,其主要功能为流程处理所涉及的数据信息和资料提供存储与管理支持.

4 应用实现

我们采用 C++ 对 GWWFBAO 进行了实现,其中,工作于用户操作端的界面操作对象、资源接口对象、网络接口对象和客户资料管理对象采用 Web Browser 插件的形式实现,而其余工作于 Web 服务器端的对象则采用系统服务程序的形式实现.

在 GWWFBAO 的实际实现过程中,一个敏感而实际的问题是解决主动对象在分布式环境下的事件通信.为此,我们采用 CORBA 的对象通信方法建立一个对象通信协作的事件通道.对象通信协作的事件通道主要过程包括:事件接受者和事件发送者两个主体.

其工作模式的主要实现为(C++伪码):

```
class My_Notifier {
    void subscribe (Consumer & consumer, const char*filtercriteria) {
        insert *consumer into <consumer_set> with filtercriteria;
    };
    void unsubscribe (Consumer & consumer) {
        remove*consumer from <consumer_set>;
    };
    void push (Event & event) {
        foreach <consumer> in <consumer_set>
            if (event.topic matches consumer.filtercriteria) consumer.push (event);
    };
};
```

以下是客户在某个采用 GWWFBAO 实现的电子商务 Web 网站完成登录操作后实施交易的一个流程说明:

- 1) 客户在界面操作对象的辅助下,启动交易请求;
- 2) 界面操作对象经由资源接口对象和网络接口对象向 Web 代理对象发出交易处理请求;
- 3) Web 代理对象将请求发送至流程逻辑引擎;
- 4) 流程逻辑引擎经由 Web 代理和数据库接口对象装载一个交易流程实例,并进行相应的初始化;
- 5) 流程逻辑启动流程执行,并通知客户开始询盘操作;
- 6) 客户询盘后,流程引擎经由计算程序计算报盘数据后用邮件发送给客户;
- 7) 客户处理报盘邮件后进入定货(下一步)或放弃处理(结束);
- 8) 将合同发给客户;
- 9) 接受信用证并发货通知.

5 结 论

通过对 ECA 模型进行时间和空间上的扩充,我们建立了分布式环境下的主动对象应用模型.通过该模型,设计和实现了一个基于主动对象的全局 workflow 管理控制系统框架,有关的应用实践表明,通过该框架集成的 Web workflow 管理控制系统具有十分良好的环境独立性和时空适应性.

References:

- [1] Grudin, J. Computer-Supported cooperative work: history and focus. *IEEE Computer*, 1994,27(5):19~27.
- [2] Hollingsworth, D. The workflow reference model. In: *Workflow Management Coalition*. 1994.
- [3] Gaspari, M., Motta, E., Stutt, A. An open framework for cooperative problem solving. *IEEE Expert*, 1995,(6):48~54.
- [4] Mahling, D.E., Craven, N., Croft, W.B. From office automation to intelligent workflow system. *IEEE Expert*, 1995,(6):41~47.
- [5] Shi, Mei-lin, Yang, Guang-xin, Xiang, Yong, *et al.* WfMS: workflow management system. *Chinese Journal of Computers*, 1999,22(3):325~334 (in Chinese).
- [6] Shi, Mei-lin, Yang, Guang-xin, Xiang, Yong, *et al.* A Web-based workflow management system. *Journal of Software*, 1999,10(11):1148~1155 (in Chinese).
- [7] Cabri, G., Leonardi, L., Zambonelli, F. A proxy-based framework to support synchronous cooperation on the Web. *Software Practice and Experience*, 1999,29(14):1241~1263.
- [8] Hull, R., Lirbat, F., Simon, E., *et al.* Declarative workflow that support easy modification and dynamic browsing. *Software Engineering Notes*, 1999,24(2):69~78.
- [9] Peter, C.L., Walter, H.D. Object-Oriented protocol hierarchies for distributed workflow systems. *Theory and Practice of Object Systems*, 1995,1(4):281~300.

附中文参考文献:

- [5] 史美林,杨光信,向勇,等.WfMS:工作流管理系统.计算机学报,1999,22(3):325~334.
- [6] 史美林,杨光信,向勇,等.一个基于 Web 的工作流管理系统.软件学报,1999,10(11):1148~1155.

A Global Web Workflow System Based on Active Object*HU Hua¹, YU Hong-zhi²¹(Department of Computer and Information Engineering, Hangzhou University of Commerce, Hangzhou 310035, China);²(Network Center, Zhejiang University, Hangzhou 310027, China)

E-mail: huhua@yahoo.com

<http://www.softlab.hzic.edu.cn>

Abstract: A global workflow system framework on World Wide Web through the technology of distribute active object is presented. A model of EECA (extended event condition action) is set up for active object in Internet by adding time and space domain to ECA model in active database. Then a hierarchical task model is built for workflow with active object. Based on the models of active object and workflow, the architecture and the applications of the framework are given.

Key words: active; object; EECA (extended event condition action); state; task; level partition; Web; workflow; control

* Received February 5, 2001; accepted June 25, 2001

Supported by the Youth Foundation of the National Education Ministry