

22-25

工作流程管理

组件集成

协同工作

软件系统

⑤

计算机科学1998 Vol. 25 No. 3

面向协同工作的工作流管理中的组件集成技术的研究^{*})

On the Component Integrating Technology in Workflow Management Based on Cooperation

孙艳春 赵大哲 刘积仁

(东北大学软件中心 沈阳110006)

TP31

摘要 This paper introduces the component integrating technology in workflow management based on cooperation, then it provides with the distributed software component object model and component system function modules.

关键词 Workflow management, Cooperation, Component, Distributed object, CORBA, DCOM, RMI

1 引言

工作流程管理是近年来在计算机应用领域中发展最为迅速的几项新技术之一,其主要特征是实现人与计算机交互事件结合过程中的自动化。工作流主要涉及的内容是工作任务的整体处理过程、工作组成员间依据一组已定义的规则及已制定的共同目标所交换的文本文件、各种媒体信息或任务。工作流管理系统是一个用于分布式环境中工作任务进程间的协调及协作式处理的软件系统。

在当今的社会生活中,绝大多数工作都是在特定的群体环境中,由群体成员互相协作、共同完成的,人们的工作方式明显地具有群体性、交互性、分布性与协作性等基本特征。CSCW (Computer Supported Cooperative Work)正是以人们协同工作方式为背景,并且以计算机技术、通信技术、多媒体技术的发展为基础而形成的支持多学科合作的研究方向,它有广泛的应用领域。网络技术,尤其是 Internet 技术的飞跃发展,使信息技术产业从以计算机为中心过渡到以网络为中心的方向发展。用户对计算机应用的需求也随之发生了巨大变化,应用计算机环境的结构已从集中式发展到分布式的结构。面向群体协同工作并支持系统开放集成性的工作流管理系统是使企业的内部综合信息处理系统适应动态变化的市场需求的关键所在,人们主要采用工作流管理系统实现工作任务处理过程的重构设计及过程的优化。

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、并支持群体成员间协同工作的工作流管理环境。而构造工作流系统中的组件技术主要是实现两方面的功能,一是整个工作流系统的各部分功能将由组件构成,二是支持系统模拟仿真功能所调用的功能模块是组件。这样软件组件集成技术在工作流管理系统中是非常关键的技术之一。

2 分布式环境下的软件组件对象的标准

开放系统的发展可让用户透明地应用不同机型、不同运行平台组成的异构型计算资源,从而提出了应用集成与分布式处理的要求,即在千差万别的信息资源的基础上构造起信息共享的分布式系统,并能有效地实现应用系统与分布式处理的集成。分布式处理的关键是定义可管理的软件组件。就客户来说,希望这种软件组件能“即插即用”,即能从所提供的软件组件库中获取最合适的组件并可充分重用现有成熟的软件代码。就服务方来说,希望这种软件组件能便于客户裁减,维护和重构。

软件组件模型的确立是软件组件化的关键。软件组件模型详细描述了如何开发可重用的软件组件和这些组件之间如何相互通信,组件模型是一个体系结构,它给开发者提供了一整套 API 来定义软件组件,使用组件对象模型的软件开发者可以动态地组织现有组件来构成一个新的应用,以实现组件的重用。而每个软件组件必须具备以下几个特征:

^{*})本课题受国家“九五”科技攻关项目资助,孙艳春 博士生,主要方向为软件组件技术,CSCW 技术。赵大哲 博士,主要研究方向为软件开发环境,刘积仁 博士导师,主要研究方向为分布式多媒体、网络协议工程。

·组件是完成通用或特定功能的可重用的软件模块。

·组件必须能够进行自身描述,即组件必须能够识别在它构造过程中可更改的属性和它生成的事件。这些信息可以使开发环境与第三方软件组件无缝地结合起来。

·组件必须允许以图形方式编辑,其属性几乎只能通过控制面板来设置。组件通过界面输出其功能,外界环境也只能通过界面访问组件。

·组件必须可以被编程语言直接控制。因为组件也可以和脚本语言连接或者与从代码级访问组件的环境连接,这个特性使得软件组件可以在非可视化开发项目中使用。

·组件必须能产生一个事件或者其它让程序员从语义上连接的机制。这意味着程序员可以很容易地向按钮添加代码,这样,点中按钮就可以影响其它组件的动作。

·组件支持对象意义上的封装性、多态性及继承性,又称为组件对象。其遵循二进制代码标准,是一个可以在异构环境下被调用的支持互操作的软件模块对象。

随着 Internet 的飞速发展及当前人们对 CSCW 的需求,协作与系统集成显得越发重要,分布式对象计算正是从系统集成与分布式处理要求中提出的一种可行的解决方案。分布式异构环境下的软件组件技术是建立应用系统集成框架和标准构件的核心技术,也是我们实现的面向协同工作的工作流管理系统中功能模块实现与集成的关键技术。而当今国际上有三大分布式软件组件对象标准:一个是由 OMG (Object Management Group) 组织推出的 CORBA (Common Object Request Broker Architecture),即公共对象请求代理结构;一个是 Microsoft 推出的 DCOM (Distributed Component Object Model),即分布式组件对象模型;还有一个是 Sun 公司推出的用 Java 语言开发的分布对象模型 RMI (Remote Method Invocation),这三大分布式软件组件对象标准各有所长,下面我们就做一下对比。

(1)结构和规范性。CORBA 是一个通用的分布式对象的规范说明,它没有给出参考的实现方案,所以为实现提供了极大的灵活性,而 DCOM 和 Java 的 RMI 有明确的实现背景,规范严格细致,不利于优化。

(2)跨平台能力。目前 ActiveX/DCOM 基本用于微软操作系统—Windows95和 Windows NT,虽然已有支持 Unix 系统的 DCOM 实现的测试版本,但还需进一步完善。对于 RMI 来说,只要在客户端安

上 Java 虚拟机,就可以实现 RMI 的跨平台通信。而符合 CORBA 规范的产品支持广泛的平台,几乎可用在所有的操作系统上。

(3)跨语言支持能力。DCOM 实现中所用的编程语言几乎都是 C++,而对其它编程语言的支持有障碍。类似地,由于 RMI 依靠许多 Java 语言的内部特征,使得它一定要用 Java 语言编写,而不能用其它语言实现。与此相对比,CORBA 具有语言中立性,可以包容多种编程语言。在写规范时,OMG 已经采纳了用 C、C++、ADA 和 Smalltalk 语言使用 CORBA 机制的规范。所以 CORBA 具有强大的跨语言能力。

(4)安全性。所有的分布计算必然包括通信,如果分布计算是在分布式网络上,那么在传输数据时,数据的安全性和完整性都有危险。安全性必须保证用户不受破坏代码的侵害,DCOM 使用远程过程调用(RPC)在相距异地的对象间通信。它没有在分布式的数据网络如 Internet 上提供安全保证。使用 DCOM 实现的 ActiveX 控件不含严格的安全性检查或资源权限检查,控件具有其资源的所有权限,这样就缺乏固有的安全性。而 RMI 的安全性依赖于 Java。Java 的安全性包括严格的安全性和资源级安全性,严格的安全性用来保护用户不受那些由于不正确类型的映射和不合法内存存取而引起的错误;资源级安全性用来限制 Java 程序磁盘访问权限、网络访问权限等。Java 固有的安全性保证了 RMI 具有较强的安全性。与此相比,OMG 已经为基于 CORBA 的系统指定了广阔范围的安全服务,该服务不仅提供了保密性和认证机制,而且实现了非否认机制(用于确保参加者不能后来否认他们的许诺)。

基于 CORBA 在以上四方面的优越性,我们决定选择 CORBA 规范作为我们在工作流管理系统中分布式组件对象的标准。

3 CORBA 规范

CORBA 规范是由 OMG 组织提出的基于抽象的对象模型的分布式对象标准,它用于分布式异构环境下的基于对象的软件组件间的通信与系统集成,是表达应用系统、应用部件之间有效通讯的连接技术,OMG 于 1991 年发表了对象请求代理(ORB)的技术标准 CORBA1.0,1995 年发表了 CORBA2.0。

3.1 CORBA 规范的关系模型

理解 CORBA 设计结构的关键是关系模型,包括下列成分:

·Object Request Broker (ORB,对象请求代理):它使对象能在一个分布式环境下透明地建立和接收请求和回答,它是建立分布式对象应用的基础,也是在异构和同构环境中应用之间的交互性的基

础。

• Object Service (对象服务): 它是一个服务集合, 用于支持使用和实现对象的基本功能, 服务对于建立任何分布应用是必需的, 而且经常独立于应用域。例如, 生命周期服务定义了建立、删除、拷贝和移动对象的转化, 但没有说明对象在一个应用中是怎样实现的。

• Common Facilities (通用设施): 是一个服务集合, 许多应用共享此集合, 例如, 一个系统管理或电子邮件设备可以被划分为一个通用设施。

• Application Objects (应用对象): 是单个的供应商用于控制接口的产品, 应用对象遵从传统的应用观念, 因此它们不是由 OMG 来标准化。应用对象构成了关系模型的最高层。

在这四个部分中, 对象请求代理是关系模型的核心, 它象一部电话交换机, 提供创立和接收调用的基本机制。ORB 与对象服务相结合, 确保符合 CORBA 规范的应用组件间的有效通信。下面就对 CORBA 规范的内容及其采用的技术进行系统的介绍。

3.2 ORB 的体系结构

ORB 是 CORBA 规范的核心, 图1显示了 CORBA ORB 结构的主要组成部件, 并辅助说明了各部分的功能。

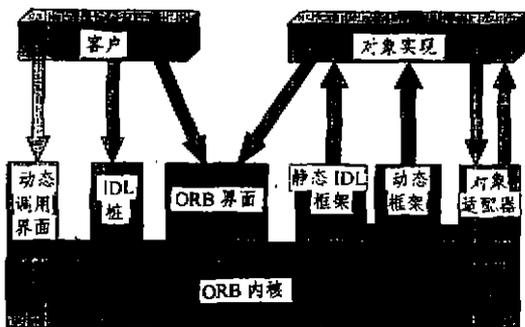


图1 ORB 体系结构

• Object Implementation (对象实现): 对象实现定义了实现 CORBA IDL 接口的操作, 它可以用 C、C++、Java、Smalltalk 及 Ada 等许多语言编写。

• Client (客户方): 这是调用对象实现上的操作的程序体。存取远程对象的服务对于调用者来说是透明的, 在图中的其它组成部分帮助支持调用的透明性。

• Object Request Broker (ORB 对象请求代理): ORB 为客户请求和目标对象实现之间提供透明的通信机制, ORB 通过把客户方与方法调用相分离而简化了分布式程序设计。这使客户请求好像是本地

的过程调用。当客户方调用一个操作时, ORB 负责找到对象实现, 如果需要时激活它, 传输请求给对象, 并把结果返回给客户方。

• ORB Interface (ORB 接口): 一个 ORB 是一个可以用多种方式 (如一个或多个进程或一个库集合) 实现的逻辑体。为了把应用程序与实现细节相分离, CORBA 规范为 ORB 定义了一个抽象的接口, 这个接口提供了多种功能如把对象实现转换成串及把串转换成对象实现, 创建由动态调用接口生成的请求的参数列表。

• IDL Stubs and Skeletons (IDL 桩和轮廓): IDL 桩是客户方和 ORB 之间的粘和剂, 相似地, IDL 轮廓是实现方和 ORB 之间的粘和剂。在 CORBA IDL 定义和目标编程语言之间的转换由 CORBA IDL 编译器来实现。

• Dynamic Invocation Interface (DII 动态调用界面): 这个接口允许客户直接访问由 ORB 提供的基本请求机制, 应用程序使用 DII 动态地发出请求给对象而不需链入 IDL 特殊接口桩。不象 IDL 桩仅允许 RPC 形式的请求, DII 也允许客户做非阻塞的延迟同步 (发送和接收相分离) 与只发送式的调用。

• Dynamic Skeleton Interface (DSI 动态轮廓界面): 这是在服务方的与客户方 DII 相类似的对应部分, DSI 允许 ORB 把请求传递给对象实现, 而该对象实现编译时并不知道它要实现的对象类型。提出请求的客户方不需知道对象实现是使用特殊类型的 IDL 轮廓还是使用动态轮廓。

• Object Adapter (对象适配器): 对象适配器帮助 ORB 把请求传递给对象以及激活对象, 更重要的是对象适配器把对象实现与 ORB 连接起来。可以指定对象适配器为某些对象实现类型提供支持。

CORBA 规范还定义了 ORB 之间互操作的机制, 使不同实现的 ORB 系统能相互识别和转换, 从而能够共存于同一个环境下。

4 工作流管理系统中的组件技术

在工作流管理系统中, 我们利用构造模型实现了事物处理过程从现实世界中的原形向计算机可处理的形式化定义的转换, 构造模型为工作流程中的每一个独立事件定义了必要的输入和输出信息、相关事件的运行顺序及必要的事件处理机制, 动态仿真采用组件库中提供的组件检测模型建立的正确性、实现实时事件处理过程控制和实时事件间的交互。因此, 仿真组件的生成与重用起着非常重要的作用。我们建立了支持组件生成、管理及重用的集成组件模型 (ICM—Integrated Component Model), 它采用 Agent 机制及面向对象程序设计方法并支持

OMG 的 CORBA 标准,基于此模型建立的工作流管理平台实现了组件间的协调及接口管理、组件间共享数据基管理和组件浏览及组件版本升级管理功能。系统具有信息模型易扩展、功能可动态重组、跨平台实现以及开放性等特点。

我们基于 CORBA 规范实现的面向协同工作的工作流系统中的分布式软件组件对象的模型如图2所示:

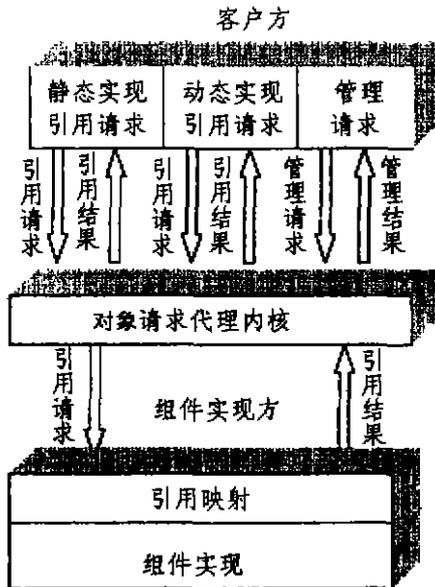


图2 分布式软件组件对象模型

用户基于系统层次化信息结构定义功能,可以对不同的组件加以划分,并由此构造出层次结构功能对象集合;系统功能重构框架根据应用系统信息模型对其所定义的系统对象功能进行选择 and 重组,用户可根据应用需求动态重构客户端功能界面;数据库服务代理在实现面向对象的结构化的组件存贮的基础上可根据应用需求对相关组件进行插入、删除、查询、重组等操作;系统管理对分布式环境中的功能组件进行动态管理和监控,并对各种用户请求进行权限检查;开放代理接口实现与其它组件标准的互联,实现功能对象的相互调用。在面向协同工作的工作流管理中的组件系统功能模块如图3所示。

如图3所示框架系统以网络环境和 Java 分布式技术封装为基础,在中间件基础之上实现远地和本地的组件查询和管理监控机制,功能重组框架则以预定义程序信息结构为基础,通过构造对象功能实现应用系统的具体功能组合,开放接口实现与其它标准中间件平台之间的相互操作,管理监控组件是一组相对独立的功能,它直接基于中间件平台实现对应用系统对象的监控功能,从而使整个系统不但灵活多变且易于管理。

目前,我们采用先进的分布式计算技术、网络技术编程实现了面向协同工作的工作流管理系统原型,下一步的工作将是对其进行功能扩充及进一步完善。

参考文献

- [1] The Common Object Request Broker Architecture and Specification, Revision 2.0, July 1995
- [2] Randy Otte et al., Understanding CORBA, Prentice Hall, 1995
- [3] Francies Chan, Research on OMG/CORBA, February 1996
- [4] The Java White Paper: Introduction to Java, Sun Microsystems Inc., 1995
- [5] Nat Brown, Charlie Kindel, Distributed Component Object Model Protocol-DCOM/1.0, Internet Draft May 1996
- [6] The System Object Model (SOM) and the Component Object Model (COM): a comparison of technologies from a developer's perspective, Object Technology Products Group, IBM Corp.

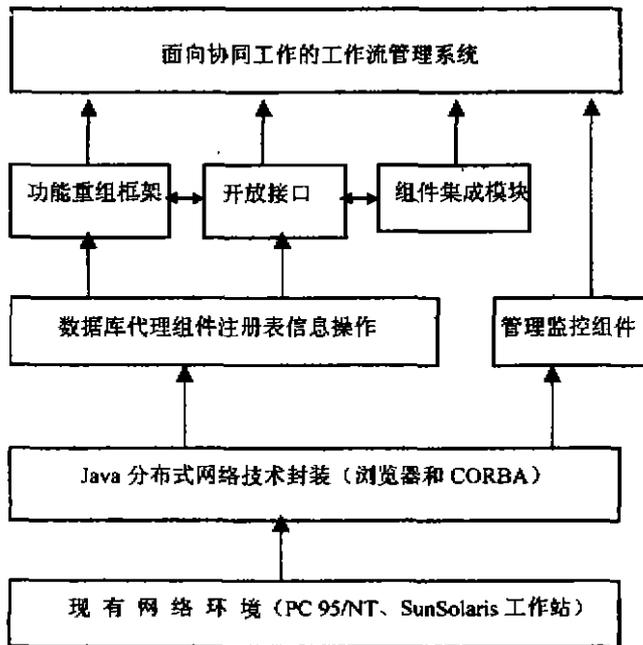


图3 面向协同工作的工作流管理中的组件系统功能模块