

Agent

分布式集成框架

设计
双高晓

10

一种基于 agent 的分布式集成框架的设计与实现

Design and Implementation of an Agent-Based Distributed Integration Framework

42-45

史殿习 王怀民 邹 鹏 吴泉源

(国防科技大学计算机系 长沙410073)

TP18

摘 要 An object frameworks is a group of related classes that can be used to build a specific type of application. An object framework can integrate well with other frameworks and create a distributed integration framework. First, this paper promotes event-process centric agent framework. Then, we build the integration framework of the distributed Client/Server computing environment based on multi-agents CSE/MA, discuss the components of the CSE/MA and their functions, describe the mechanisms and executive process of CSE/MA. Last, we design and implement the framework of the service agent(SA) and the request agent(RA) to support the development of the distributed Client/Server applications.

关键词 Object Framework, Integration Framework, Client/Server, Agent

对象框架技术已成为当今分布异构环境下建立应用系统集成和软构件的核心技术,为支持分布式客户/服务器应用的开发和集成,我们开展了分布式集成框架 CSE/MA 的研究和开发,目的在于提供一种支持分布式客户/服务器应用开发的分布式集成框架,方便用户开发和集成分布式客户/服务器应用。本文论述对象框架的概念及其工作原理,分布式集成框架的组成及特性,讨论分布式集成框架对分布式客户/服务器应用的支持,对 agent 的概念进行研究,提出 agent 的框架模型及 CSE/MA 的框架结构,以及 RA 和 SA 框架的设计思想。最后,对 CSE/MA 的特性进行概括。

1 对象框架

1.1 对象框架的含义

对象框架技术是在对象、类以及类库等技术基础之上逐步发展起来的。一个对象框架是一组提供特定功能的类,这些类之间相互关联并且可以被用来建立某一特定类型的应用,协助应用类共同完成特定的任务。从这一点上看,框架结构的技术含义包括两个方面:

- ① 框架结构相对于目标系统来说是未完成结构;
- ② 框架结构实际上是建立了一个开发环境,支持以其为基础的再开发,进而建立目标系统。

实际上,框架好比一块主板,当人们购买主板时,总是买一块完整的板,而不是购买每一个芯片,这一个个芯片就相对于一个个对象,硬件主板本身为这些芯片的协同工作提供了支持机制。不过与硬件主板不同的是,“软件插板”(框架)可以被扩展和进一步的剪裁,以便满足应用的需要。

1.2 对象框架的工作原理

对象框架的工作原理与过程调用的工作原理不同(如图1所示)。当应用开发者使用对象框架时,只需告诉对象框架其想处理的主要事件,并提供处理这些事件的代码即可。当这些事件发生时,框架将自动调用相应的代码,应用程序无需关心框架的结构、执行的流程等方面的内容,框架将自动完成上述工作,应用代码所要做的只是等待被框架调用。

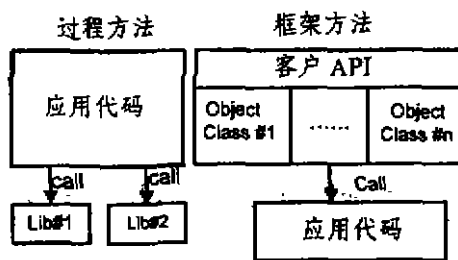


图1 过程方法和框架方法工作原理的比较

对象框架具有很强的可扩展性,应用开发者可以根据需要对框架进行修改,可以用自己的类来代替对象框架中的某些类,从而达到剪裁对象框架的目的。这可以提供多重继承和方法重载来实现。

2 分布式集成框架

2.1 分布式集成框架的含义

在分布式系统中,要支持分布式应用的开发和集成,仅有应用于某类应用开发的对象框架是不够的,还需要有将多个对象框架集成起来的支持机制,这就是分布式集成框架。一般而言,分布式集成框架是一种支持系统集成的通用结构与机制,它为应用系统中各个独立成分之间实现相互作用提供公共通道。例如,OMG的对象管理结构是一个分布式集成框架,该集成框架在各种抽象层次上提供了丰富的服务,以支持分布式对象计算。通常,一个分布式集成框架由如下两个基本机制组成:

- 框架接口** 提供一组 APIs,该组 APIs 是对每个框架所提供的服务的简单的抽象。框架接口作为框架之上的一层。

- 一个连接各个框架的对象请求代理** 该对象请求代理相当于“软件总线”,其作用是跨越地址空间传递请求。

分布式集成框架技术是推动分布式对象技术的理想技术,其好处在于它将实现对象框架之间关系的责任交给集成框架来完成,从而使应用开发者将主要精力集中在应用对象的开发上,应用类从框架类中继承框架类的行为的同时,也继承了框架类之间的关系。使用集成框架,应用开发者既可以重用已存在的功能建立一个客户方的应用,也可以建立提供服务的服务器应用。当客户请求某一服务时,集成框架调用相应的服务器功能为客户提供服务。

2.2 分布式集成框架的技术基础及特性

分布式集成框架技术逐步成为分布式对象计算的核心技术,这主要得益于面向对象的设计原则及其使能技术和网络技术的强有力的支持。一方面,面向对象的设计原则及其使能技术在几个方面对集成框架提供了支持:●面向对象的分析和设计方法学(OOAD);●支持 CASE 工具;●面向对象的程序设计语言及支持环境;●OMG 所提出的 ORB 机制及其相关的标准;以及●OLE 在桌面系统中的成功应用。另一方面,网络技术通过局域网(LAN)、广域网(WAN)及 Internet 技术支持真正的分布式计算。通

过网络应用开发者可以将其应用作为服务器为网络上的其它的用户提供服务。

将网络技术与面向对象方法和技术结合起来,应用开发者便可以设计出分布式应用,但要将这些分布式应用集成起来成为一个系统,就必须有相应的支持技术,其中,分布式集成框架技术就是一种非常好的技术。要支持分布式客户/服务器应用的集成,分布式集成框架必须具有如下的特性:

- 支持高层次的接口定义;

- 提供请求激活、名字服务及事件处理等支持服务的基础结构;

- 允许用面向对象开发环境建立新的对象服务及客户应用;

- 提供将应用功能的接口和功能的实现相分离的能力等。

利用分布式集成框架的支持技术和其提供的支持服务,应用开发者便可以在集成框架中方便地建立分布式客户/服务器应用。

3 分布式集成框架对分布式客户/服务器应用的支持

分布式集成框架的目标是如何跨越异构平台建立分布式客户/服务器应用。为实现这一目标,分布式集成框架为建立分布式客户/服务器应用提供了相应的基础结构和系统管理服务(如 OMG 的对象管理结构)。其中,基础结构为应用开发提供一组支持机制,如事件处理机制和通信服务机制等,事件处理机制提供事件的注册及引发等服务,通信服务机制在框架的各个应用之间提供消息传递服务。系统管理服务对系统的管理提供支持,如跟踪、监控系统的状态、浏览系统内的某些信息等。

由于分布式集成框架为开发分布式客户/服务器应用提供了相应的支持机制,从而屏蔽了通常需要应用开发者了解的许多细节问题,并负责对网络和应用的管理,协调运行在分布异构平台上的各种应用,所以大大降低了开发分布式客户/服务器应用的复杂性。对最终用户来说,一个分布式应用系统看起来就象一个运行在本地的应用系统;对应用开发者来说,只需将注意力集中在如何开发自己的应用上,而无需关心其它方面的内容。因此,集成框架为开发分布式客户/服务器应用带来很多的好处,如为应用开发者提供了即插即用的能力、减少应用开发代码、降低复杂度及开发费用、减少维护以及提高可

重用性和灵活性等。

4 CSE/MA——一种基于 agent 的分布式集成框架

在对对象框架和分布式集成框架研究的基础上,为了统一地划分分布式计算系统中的计算节点、服务器、客户机及进程组等各类独立自主的行为实体,我们开展对 agent 的研究,引入了 agent 的概念和机制,提出了一种基于 agent 的分布式集成框架 CSE/MA,以支持分布式客户/服务器应用的开发。

4.1 agent 的框架结构

Agent 是指在分布式系统中持续自主地发挥作用的计算实体。例如,通常分布式客户/服务器计算机系统中的各类服务器、SNMP 中的 Manager 和 Agent 都是我们这里所说的 agent。Agent 的概念与“对象框架”的概念有密切的联系,从内部结构上看:

Agent=对象结构+行为引擎
简单形象地讲,agent 是有“生命”的对象,其中,行为引擎是 agent 的“心脏”,是体现 agent“生命力”的部分;对象结构是 agent 的“大脑”,是体现 agent“智力”的部分。从外部特征上看,agent 具有:

●独立性。即 agent 是分布式计算系统中边界明确、可被独立地引用的行为实体,支持数据、过程和通信设施进行封装;

●自主性。即 agent 是分布式计算系统中自主的行为实体,支持局部与自身的行为控制,能够根据其内部状态或外部事件来决定和控制自身的行为;

●交互性。即 agent 支持与其它 agent 灵活多样的交互机制,能够有效地与其它 Agent 共同工作。

为此,我们提出了以事件处理为中心的 agent 模型。在该模型中,agent 由事件处理系统、方法集和内部状态集三个主要成

分构成。其中,事件处理系统是 agent 的行为引擎,Agent 的活性表现为它的事件处理系统在该 agent 的生命期内始终持续自主地工作。在该模型中,事件是指与 agent 有特定关联的状态,事件处理系统及事件感知、事件适配和事件处理分发三个环节的活动。Agent 的事件感知器捕捉所关注的事件是否发生,并根据事件状态的类型启动相应的事件适配器工作;事件适配器获取有关事件信息作识别,并将识别的结果提交给相应的事件处理分发器。事件处理分发器根据事件适配结果启动有关的事件处理方法执行。Agent 的方法集是体现 agent 事件处理能力的成分,描述了 agent 处理相关事件的方法。方法的执行可能影响 agent 的内部状态,从而导致新的事件的发生。Agent 的内部状态集是表现该 agent 当前状态的成分,其中包括表征事件的状态。

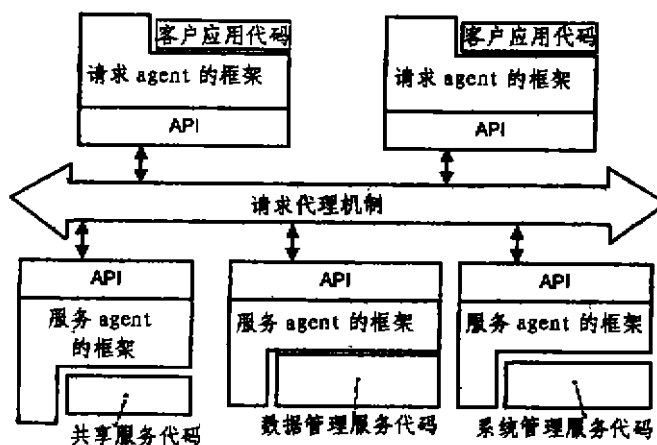


图2 分布式集成框架 CSE/MA 的结构

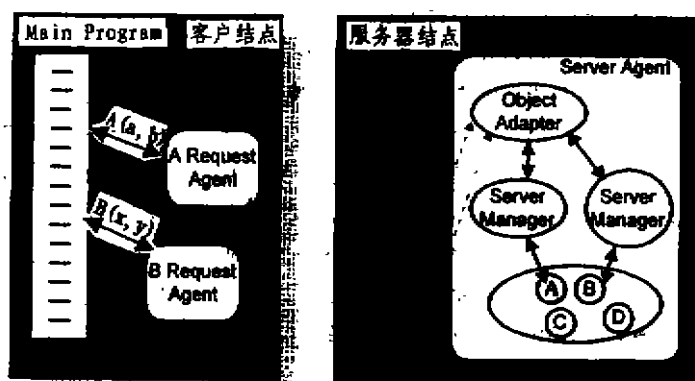


图3 CSE/MA 的工作原理

4.2 CSE/MA 的框架结构

CSE/MA 的基本目标是为应用开发者提供开发客户/服务器应用的通用的集成框架(如图2所示),支持应用开发者灵活方便地建立各种客户/服务器应用。CSE/MA 的框架可理解为“软件总线”,其核心是基于 agent 的服务请求代理机制,各类服务 agent (简记为 SA)可作为“软部件”插接到该框架上。客户应用则通过本地称为服务请求代理(简记为 RA)的对象访问 SA。RA 的存在使得客户应用所需的异地服务如同在本地一样。

从客户/服务器计算角度来看,CSE/MA 分为客户环境和服务器环境两个部分。CSE/MA 客户环境不仅提供了典型 SA 的 RA,而且提供了 RA 的构造框架及相应的功能简洁的 API,支持应用开发者建立所需的 RA。服务环境中服务可分为基本系统服务、公共应用服务和特殊应用服务。基本系统服务是保证系统正常工作的系统服务,如注册管理服务 Register 使得客户方的 RA 可以通过 SA 的名字在运行时刻动态地确定 SA 的物理位置;公共应用服务涉及客户应用所需的常规服务(如数据共享服务和知识处理服务等);特殊应用服务(如产品设计数据管理服务)为特定客户提供针对性的服务。CSE/MA 提供了建立 SA 的框架,应用开发者采用这一框架可方便地建立新的 SA,加入到服务环境中,为客户提供服务。

4.3 CSE/MA 的工作原理

利用 CSE/MA 建立客户/服务器应用时涉及 RA、SA 和系统服务 agent Register。CSE/MA 的工作过程(如图3所示)是:当客户应用欲请求一个 SA 为其服务时,即可以向 SA 在本地的代理 RA 发出请求,RA 负责向系统管理 agent Register 查询其所代理的 SA 的地址信息,并与其所代理的 SA 建立连接;SA 接收到 RA 发来的请求后,则进行相应的处理,并将处理后的信息传递给对象适配器,对象适配器进行对象的适配,如果适配成功,则将有关信息传

递给服务管理器,服务管理器负责激活相应的服务方法,并将结果传递 RA;最后,RA 将服务结果传递给客户应用。

结束语 CSE/MA 的主要技术特点是采用90年代先进的分布对象技术,提供了灵活开发客户/服务器应用的通用框架(软件总线)以及基本的分布管理,我们与航天工业总公司二院合作建立航天产品数据管理(PDM)的 Framework,利用 CSE/MA 的通用集成框架,在短短的两周时间内将 CAD 开发工具 ProE 及有关的工具集成为一个支持 CAD 应用开发的分布式计算环境,大大缩短了应用开发周期,同时提高了 CAD 应用产品的开发效率。CSE/MA 的开发与应用已经体现出基于 agent 的分布式集成框架的优势。

参 考 文 献

- [1]Lewie T. G., Where is client/server software headed? Computer, 28(4)1995
- [2]Khanna R. (editor), Distributed Computing, implementation and management strategies, Prentice Hall, 1994
- [3]Mowbray T. J. and Brando T., Interoperability and CORBA-based open systems, Object magazine, Sept.-Oct. 1993
- [4]Wang H. M., Wand P. and Gao H. K., Decentralized computing environment for agent-oriented programming, In: IFIP 13th World Computer Congress 94 Vol. 1, Germany, 1994
- [5]Corbin J. R., The Art of Distributed Applications, Programming Techniques for Remote Procedure Calls, Spring-Verlag, 1991
- [6]Robert Orfali and Dan Harkey, Client/Server Survival Guide with OS/2, Van Nostrand Reinhold, 1994
- [7]Mark Betz, An Object by any other, Object magazine, 1996, March
- [8]Peter Hennessey et al., An integration framework for distributed system, Same to [7]

从1999年

《计算机科学》杂志改成年刊