计算机科学 1997Vol.24 №.3

分布对象技术综述

陈 莹 樊惠娟 王能斌
(东南大学计算机系 南京 210096)

摘 要 分布对象技术是解决当前分布异构环境计算的主要方法,本文详细介绍分布对象技术产生的背景,分布对象模型和组件概念、模型,同时介绍了几种主要的分布对象标准及其实现情况。

关键词 分布异构环境 分布对象 组件 复合文档

1. 引言

计算机支持下的工作正在转移到一个复杂的分布异构计算环境,它主要有以下特点:·场地分布,由 LAN 或 WAN 支撑,存在多种网络协议。·数据分布,各种形式的数据分散在各节点,以各种形式(文件、数据库、电子表格等)存在。·硬件平台多样化,从台式机,工作站到大型主机,从单处理器、对称多处理器(SMP)到大规模并行处理(MPP)。·操作系统多样化,如 Windows NT,Netware,各种 Unix 以及 VMS 等。·应用平台多样化,包括来自不同开发组织的各种应用软件、中间件(middleware)和开发工具。

分布异构计算环境的出现是计算机技术发展和市场需求驱动的必然结果,是信息随处可得这一人类期盼已久目标的基础。同时它也给计算机界带来了新的挑战,用何种方法支持这种环境下的工作?用户在这种环境下以何种方式获取信息?面对这些问题,急需新模型支持分布异构环境下的协同工作,这个新模型应提供以下主要措施:·场地透明机制,屏蔽本地服务和远程服务对应用造成的差异。·平台独立机制,支持各种主流机型和操作系统。·统一的编程模型,这种统一性应体现在软件体系的不同层次,如操作系统、网络,特别是应用层。·互操作能力,不同的应用程序之间可以互相调用。

针对这个目标,各标准化组织、各计算机厂商和联盟作出了许多努力,在分布式环境下针对不同层次、不同用途给出了解决方法,制定了诸如互连性、远程过程调用、数据库调用接口(CLI)、GUI、标准的操作系统调用界面等众多标准,如 ISO OSI、X/Open POSIX、OSF DCE、ANSI SQL、MIT XII 等等,但众多标准仍缺乏一个统一框架,使得在此框架下能充分利用各种标准技术,真正提供给用户(最终用户和应用开发者)一个在分布式环境下的统一视图。

对象技术(OT)和客户/服务器(C/S)计算模式的逐渐兴起和被广泛接受使建立统一的分布式异构计算环境标准的努力出现了曙光。

2. 背景

对象技术(OT)出现后迅速应用到了计算机软件的各个方面,面向对象的软件工程(OOSE),程序设计语言 OOP,面向对象数据库 OODB,面向对象的操作系统 OOOS,以及流行的各种面向对象的可视化开发工具。这些都大大提高了软件开发效率和使用的简便性,对象技术的原则和方法,如封装、继承、多态等,显然已成为软件界的通行法则。但 OT 目前存在一个不支持异构环境下互操作性的重要缺陷,主要表现在来自不同厂商和开发组织,甚至同一厂商的对象很难协作,形成了所谓对象孤岛(islands of objects),它严重阻碍 OT 的发展,使其很难在分布式计算中发挥作用。

另一方面,与 OT 几乎同时开始大规模应用的多媒体技术大大改善了计算机的交互和处理能力。通过使用多媒体工具可以编辑多媒体对象,如声音、图象、动画等,使人们从单调的文字中解脱出来,信息表达方式更加自然。基于 GUI 的多媒体文档(Multimedia Docuement,MMD)概念应运而生。MMD 中包含各种媒体对象,使文档丰富多彩。MMD 的应用同时也给用户维护文档带来了困难,因为不同种类媒体需要不同的编辑工具处理,用户不得不脱离文档编辑去分别编辑众多种类的媒体,再通过转送机制,一般利用文件系统或剪贴板(clipboard),将媒体信息嵌入文档,使用户难以将工作集中于文档内容本身,这样的过程增加了工作的繁琐

程度，文档一致性难以维护，在文档较为复杂的情况下，该问题显得更加突出。人们迫切需要一种新的集成文档工作平台，支持以文档为中心的(document-centric)工作方式，并且具有高度的灵活性、可管理性，以及在分布式环境下的跨平台文档管理功能。复合文档(Compound Document，CD)的概念应运而生。

CD是一种包含不同种类数据的文档，它集成异构数据并能对它们进行有效管理。CD不仅能包括多媒体数据，还有其它如电子表格，直方图等，并且能对这些数据进行网络环境下跨机器、跨平台的管理，真正提供给用户统一的文档视图，这点与不包含管理机制的MMD不同。CD提供了一种以文档为中心的工作模式，提供可视化工作环境，用户不需离开当前工作环境就可以操作分布在网络上的文档，且管理维护方便。CD支持的主要技术包括：

·链接与嵌入。链接指文档组件本身不存放数据，而仅存放一个指向数据的指针。这种方法可以较好维护数据一致性，外部数据变化可以立即反映到文档中，且数据可以在多个文档之间共享，节约数据存储空间，适用于数据内容经常发生变化的情况。缺点是当文档位置发生变化时，指针则会失效。嵌入是指数据整个嵌入到文档中，因此没有链接中位置变化问题，缺点是数据一致性难以维护，数据在外部的修改不能自动反映到文档中，且多拷贝占用存储空间大，一般嵌入方式用于数据变化不大的情况。CD应同时提供这两种方法。

·嵌套包含。文档组件嵌套是实现"树"型文档结构的基础。它指文档组件内可以包含子件，构成层次关系，使文档更容易组织、管理和共享。

·本地激活(In-Place Activation)，又称可视编辑。指在文档内直接激活对象的编辑程序而不需脱离当前应用，在一个典型的图形窗口环境中，这意味着用户不需要激活另一个窗口编辑数据，复合文档自动激活编辑数据的应用，该应用附着在当前窗口，用户只需在本窗口内编辑，它使用户的焦点集中在文档，同时提高了工作效率。

·拖放操作。文档间(或任意两个支持拖放的应用)复杂数据的交换复制变为简单的鼠标操作，拖即取得数据源，放即将数据置于目标文档，不需要通过剪贴板的中间过渡，更加直观化，简化了操作。

·操作自动化。对支持复合文档技术的应用可提供自动化接口，使用户或另一应用无需启动该程序就可以通过向该应用发送简单指令得到所要求的服务，使用户具有扩展的灵活的控制方式，开发较为复杂的复合文档。如对一电子表格内的数据自动进行计算。

CD提供的功能涉及到多对象、跨平台，因此需要一种分布对象模型支持。

为了解决上述问题并且满足分布式计算的需求，80年代末至90年代初，分布对象模型(Distributed Object Model，DOM)的概念逐渐形成，它以对象技术为基石，将OT融入分布环境中，支持在分布环境下对象之间的互操作和协作，最终建立开放的分布式系统。DOM已逐渐成为公认的分布异构环境下的主流计算模型。

3. 概念

3.1 分布对象模型DOM

DOM的概念非常适合于分布式异构环境，以DOM的观点，分布式环境中的一个自治系统或一个应用都可视为对象或一组对象，分布式系统中可共享的软硬件资源，如文件、数据库、打印机、硬盘空间等等，都被操作它们的对象所封装。对象对外提供操作界面，操作由消息激活，在协调机制参与下，对象之间通过传递消息完成共同的任务。

DOM还提供管理维护工具，它包括大量对象的创建、浏览、复制、注册等，各种数据库(存放对象的元信息)的维护，如一致性、完整性，以及各种安全控制，如鉴别(Authentication)、授权(Authorization)等。特别是在大型的分布式环境下，这些功能尤为重要。

DOM为分布式环境下的资源提供了统一的视图，对象界面和消息传递屏蔽了异构性和分布性，协调机制解决了互操作性，使分布式系统的设计、实现和管理得以大大简化。DOM中的一个重要概念是组件。

3.2 组件

组件(component)是在DOM基础上提出的，是分布对象技术的一个重要概念。组件是一块独立可重用的二进制代码，它具有特定功能，支持灵活的即插即用，可以被方便地插入到网络、语言、应用、工具、操作系统中工作。例如，一个拼写检查组件可以嵌入到不同的字处理应用中。它类似于集成电路(IC)，来自不同厂商的IC可以组合起来形成一定功能的模块，组件可看作软件IC。它有如下特点：

·组件是完成通用或特定功能的可重用软件模块。

·组件遵循二进制标准，其实现不依赖于某种高级语言，它可以由对象语言或非面向对象语言实现。

·组件通过界面输出其功能，外界仅能通过界面访问组件。

·组件支持对象意义上的封装、多态和继承，所以又称组件对象。

·组件是一个支持互操作的对象，它可以在跨越地址空间、网络、语言、操作系统的异构环境下被调用，或和其它组件协同工作。

·组件不是一个完整的应用程序，多个组件可以通过组合构造一个应用程序。

由组件的特点可以看出，它是实现分布式计算的基础单元。

3.3 组件系统模型

组件系统模型(Component System Model，CSM)是一个基于组件技术的分布对象系统模型，其中存在多个组件对象，它们相互通讯、协作，共同完成某项工作。CSM 中的一个重要概念是对象总线(Object Bus，OB)，又称软件总线(Software Bus)或对象请求代理(ORB)，其主要作用相当于 DOM 中协调机制，为组件间或组件使用者与组件间提供透明的通讯通道。CSM 见图 1。

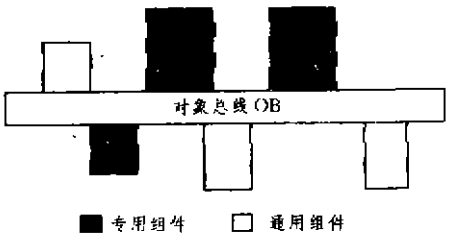

图 1 组件系统模型示意图

组件分为两种基本类型：专用组件和通用组件。专用组件实现某一领域内的特定功能，如字处理、电子表格等；通用组件实现大多数组件所需的通用功能，如名字查询，并发和事务服务，安全服务等。

CSM 中组件的交互通过 C/S 方式，OB 负责本地或网络上服务对象的定位，该过程对客户是透明的，客户请求通过 OB 转发至服务对象的界面，服务返回的结果再由 OB 返回客户。在整个过程中，客户只需了解服务组件的界面语义，无需关心其实现细节，包括编程语言，运行平台(硬件和操作系统)，位置等，实现了透明调用。服务组件只要实现其公布的界面功能，不需要关心请求来自何处，通过何种方法而来。CSM 为用户提供了一个统一的编程接口。

CMS 的实现涉及到许多重要的技术问题：

·命名管理。大量的组件必须有统一的命名管理机制，保证在大型的分布环境中甚至是全球范围内可以定位特定的组件。

·版本管理。组件的版本演化不应影响使用它的客户。

·界面管理。提供界面信息的维护手段，包括界面信息浏览，客户调用一致性检查等。

·远程调用。提供透明的远程调用机制，屏蔽各种网络协议的差异和数据表示差异。

·统一的数据交换机制。组件系统内涉及到跨越网络的多进程，数据交换是保证协作的重要环节，该机制保证进程间和网络上的交换数据格式的统一，提供数据格式协商机制。

·动态激活。提供客户在运行时刻动态构造请求的方法。

·事务管理和并发控制。在多组件协作环境下，应能提供事务管理机制和并发控制，维护应用的完整性。

·安全性。提供安全机制保证数据的保密性和服务访问的合法性。

·OB 互操作。保证来自不同组织的 OB 之间的互操作，实现建立在不同 OB 之上的组件间的相互访问。

解决上述问题的方法有些已存在相应的成熟技术或相关标准，CSM 应能充分采用这些技术或标准，将它们嵌入组件中或制成通用组件。

3.4 CSM 下的重用机制

如上所述，重用是组件的主要特点，在 CSM 中，重用有两种主要方式：一是基于代码的重用，用于构造对原始组件的功能扩充和增值服务，这种方法通过白盒或黑盒重用技术利用原始组件基本功能，是一种适用于各种领域的方法。二是基于框架的重用，主要针对特定应用领域，在构造系统时，利用该领域特有的组件和通用组件，根据实际情况进行优化组合(customization)并开发增值服务。这种方法能很快构造出企业级的信息系统，典型情况是 80%的功能可以从特定组件获得，再开发 20%的增值服务即可。

3.5 组件软件工业

正象 IC 工业一样，新的组件工业正在形成。组件概念的提出和实现为组件供应组织、应用开发者

和最终用户带来了利益。应用开发者可以方便地开发和发布应用。组件提供了从单进程到企业级网络的可伸缩性(scalbility)和重用(代码重用和框架重用),使应用开发者可以自由组合生成适合的应用,这些大大提高了生产率并降低开发费用。组件只需面对单一的模型,在设计与其它应用接口时可以方便地增加和替换应用中的组件,并且组件升级不影响客户应用,充分发挥重用的特点。最终用户将面临更多的选择。他们可以选择适合自己的组件插入到应用程序中,也可以按应用需求裁剪(tailored on-demond)。

4. 分布对象模型及标准

以下简要介绍几种主要的分布对象和复合文档标准。

4.1 共同对象请求代理结构 CORBA

CORBA(Common Object Request Broker Architecture)是对象管理集团(Object Management Group,OMG)在其对象管理结构(OMA)框架之下、以对象请求代理(ORB)为核心制定的分布对象标准,它定义了对象之间通过ORB透明地发送请求和接收响应的机制,保证了在分布异构环境下对象之间的互操作性。OMG是一个拥有660个成员(包括IBM、HP、SUN、DEC等)的国际性组织,是目前世界上最大的软件开发集团。OMA中定义了OMA参考模型(OMA Reference Model)和对象模型(Object Model),它们同时也是OMG其他标准的基础。

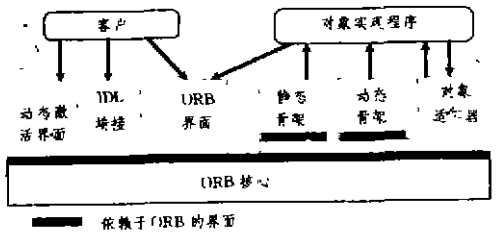

图2 ORB结构

OMA的对象模型支持面向对象的主要特征,其中重要的一点是对象通过界面输出其功能,界面是客户唯一访问对象获得服务的方法,同时也是实现透明访问的基础之一。界面定义使用界面定义语言(Interface Definition Language,OMG IDL),界面信息可加入界面仓库(Interface Repository,IR),IR提供在运行时对这些信息的访问。

CORBA1.1规范于1991年制定,包括界面定义语言(IDL)以及在特定ORB下客户和服务方交互的API,随后有CORBA1.2规范。CORBA2.0规范于1995年制定,主要增加了不同ORBs的互操作协议。

ORB是CORBA的核心,其结构如图2。客户的请求通过ORB传送至对象实现,返回值也通过ORB传给客户,在这个过程中,ORB负责对象实现的定位,启动等工作,除对象实现的界面外,客户无需知道对象实现的细节,如位置、语言等,因此ORB提供了分布、平台等透明性。在发出请求时,客户可以使用静态激活界面(Static Invocation Interface,SII)或动态激活界面(Dynamic Invocation Interface,DII)。SII又称IDL端接(IDL Stubs),它是由IDL编译器预先生成,定义了客户如何激活相应的服务。与此相反,DII允许客户在运行时刻发现服务,获得界面定义,动态构造请求。SII和DII满足相同的请求语义。此外,客户还可以使用ORB界面提供的服务。

对象实现程序,即服务方通过静态骨架界面(Static Skeletons Interface,SSI)或动态骨架界面(Dynamic Skeletons Interface,DSI)接收请求。SSI如同SII,由IDL编译器为服务方的每个界面生成。而DSI如同DII,实现了运行时刻束定(binding)服务的机制,DSI检查消息中的参数,决定目标对象和方法,它可以接受SII和DII。DSI是CORBA2.0引入的,它有助于实现不同ORB之间桥接(bridge)。请求来自SSI或DSI对服务方是透明的。此外,服务方也可以使用ORB界面提供的服务,还可以使用对象适配器提供的服务。对象实现程序可以根据服务要求选取对象适配器。对象实现程序的信息在安装时存入实现仓库(Implementation Repository)供将来使用。

CORBA2.0规范中通过引进互操作结构(Interoperability Architecture)、ORB桥(ORB Bridge)、GIOP(Global Inter-ORB Protocol)、IIOP(Internet Inter-ORB Protocol)、ESIOP(Enviroment Specific Inter-ORB Protocol)等支持不同ORB之间的互操作。

4.2 组件对象模型 COM/DCOM

组件对象模型(Component Object Model,COM)是Microsoft的组件软件方案。COM是一个标准规范,也是一个基于对象的编程模型,旨在推动基于组件的互操作性。COM中定义并提供了应用程序(客户)和服务对象(software object)的连接机制,

连接建立后客户和对象可直接通讯而不需 COM 介入。DCOM(Distributed COM)使用分 DCE 的远程过程调用(DCE RPC)机制在网络环境下为它们提供了场地透明性和安全性等机制。此外,Microsoft 还在 COM 基础上定义了 COM 的扩展结构。

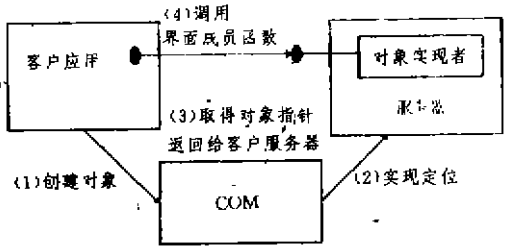

图 3 COM 的客户/服务器模型

同 CORBA 相似,COM 中对象通过界面提供服务,在 COM 中,对象和对象的使用者之间的交互是基于客户/服务器模型。其中有三种实体参与:

• 客户(client),指持有界面指针并使用对象服务的一段代码(不一定是一个应用)。

• 对象实现者(object implementor)是实现对象界面的代码。

• 服务器(server)是构造对象并且赋与其 COM 类标识符(CLSID,GUID 的一种)的代码。

客户通过传递 CLSID 请求 COM 获得对象,COM 的服务控制管理器根据 CLSID 加载运行服务器代码请求创建对象,然后建立客户与新对象的连接,如图 3 所示。注意,服务器并不是对象,而是一个构造服务的代理。

COM 的一个重要特点是它不支持传统 OOP 的实现继承(implementation inheritence),而采用黑盒重用技术,使用包含/委托(Containment/Delegation)、聚集(Aggregation)等机制实现代码重用。

4.3 系统对象模型 SOM/DSOM

SOM 是 IBM 提出的组件解决方案,其主要目的是建立一个语言中立(language-neutral)的二进制对象标准,从而实现对象的重用,它包括面向对象的模型及其实现。SOM 对象模型支持继承、封装和多态等重要面向对象特征,并且提出了元类(metaclass)等概念。DSOM(Distributed SOM)是 SOM 中实现分布对象访问的一组类库,它符合 CORBA 标准(CORBA-compliant)并做了扩充。

4.4 对象连接与嵌入 OLE

OLE(Object Linking and Embedding)技术是 Microsoft 提出的复合文档标准,它的实现基于 COM 及其扩展结构。在其中,对象以两种形式包含在容器类文档(Container Document)中,第一种方式是连接,容器中只包含指向对象的指针,而对象体本身存放在别处;第二种方式对象内容整体包含在容器中。OLE 在两种不同机制下实现了复合文档所要求的功能。在 OLE 基础上,又定义了 OLE 控件(OCX)、Active X 控件等组件。

4.5 开放文档 OpenDoc

OpenDoc 是组件集成实验室(Component Integration Laboratory,CILab)提出的复合文档标准,其主要支持厂商包含 Apple、IBM、Novell 等。它的底层对象模型基于 COM/DSOM。OpenDoc 包含文档、零件(parts)、容器应用(container application)、零件编辑器(parts editor)和零件浏览器(parts viewer)等。它实现了 CD 的主要技术,其重要特点是支持不规则形状的对象并且可以同时激活操作多个对象。

5. 分布对象技术实现及应用

分布对象概念的提出,标准的建立和技术的逐渐成熟为构造分布式异构环境下的计算提供了坚实的基础。它的观点和技术已经被广泛采用于软件系统构造的各个领域,许多运用分布对象技术的产品陆续推出,其中遵循 CORBA 标准(CORBA-complaint)的产品有 DEC ObjectBroker,Iona Orbix,IBM SOM/DSOM,Expersoft XShell,SunSoft DOE,HP ORB Plus 等,它们都支持 CORBA1.2 和主流的系统平台,如 SOM/DSOM、OpenDoc 及其开发工具已经实现于 OS/2、AIX、Windows、Machintosh 多种平台之上,ObjectBroker 产品也已经实现于 Windows 和各种 UNIX 平台。目前厂商正在相应的产品上实现新的 CORBA2.0 的互操作标准,如 IBM 正在 DSOM 中加入 IIOP,符合 CORBA2.0 标准的产品正在出现。

OLE2.0 支持 Windows95/NT 平台,目前大多数该平台下的应用软件和各种开发工具都支持 OLE 及 OLE 开发,已经成为 Windows 平台分布对象的事实工业标准,Microsoft 正计划通过 Win32 API 在主流的 UNIX 平台上支持 OLE。Network OLE,又称分布 OLE(Distributed OLE),基于 DCOM,在 WindowsNT 的新版本 Cario 中将支持它。同时,Microsoft 和 DEC 合作,使 Network OLE 支持在 Sun Solaris、IBM AIX、HP-UX、DEC UL-

TRIX、DEC OpenVMS 和 Digital Unix 等 UNIX 平台下的互操作。由于市场占有率及使用广泛性，OLE 对分布对象技术标准的影响不可忽视。

其它软件厂商也在分布对象的潮流下纷纷推出自己的组件解决方案，如 Sybase Visual Component，Informix Datablade 以及各种基于组件技术的开发工具，如 Microsoft Visual C++ 4.2、Borland Delphi 等。

6. 展望

目前，围绕分布对象标准之争日益激烈，主要是在 OLE/COM 和 OpenDoc/CORBA 之间，它们都有各自的优势，使用户和开发组织很难决定支持哪一个，同时也给这两个标准技术的使用者间的互操作带来了困难。有些厂商则选择两者都支持，如 Oracle 的 OCA (Open Computing Architecture) 同时支持 OLE 和 CORBA。OMG 正在制定的 CORBA/COM 的互联标准，以解决两者对象的互操作问题。

近年来 Internet 技术的发展及 WWW (World-Wide Web) 的出现，为大型分布式系统 (例如全球性的数据检索服务) 的构造提供了基础，同时也给分布对象技术提出了新的要求，即如何将分布对象技术运用于 Internet 的服务，如 WWW、WAIS (Wide Area Infomation Search)、Gopher 等，它们拥有大量非结构化的数据，利用分布对象技术加以封装、抽象，提供给客户端一个标准界面，同时也利用 Internet 上的大量资源来充实对象服务。Internet 一项重要的技术是 Java 语言，它是一个面向对象的程序设计语言，主要特点是具有平台独立性 (platform-neutral)，因此运用 Java 作为分布对象系统的语言具有天然优势，SUN 和 OMG 制定了 OMG IDL 到 Java 的映射标准，SUN 正在实现 CORBA/Java 项目，Netscape 在 ONE (Open Network Enviroment) 中实现了基于 Java 的 ORB (Java-based ORB) 并支持 I-IOP，因此，Internet 技术和分布对象技术的相互结合已成为必然并会有很大的应用前景。

随着计算机技术的发展，分布对象技术将不断成熟和完善，以支持分布式的协同工作 (DCW)、各种异构数据和资源的集成、透明的桌面环境等。同时，分布对象技术也将和其它开放技术和标准不断融合，如 ISO ODP (Open Distributed Processing)，最终形成一套完整的分布式系统的体系标准和技术，真正改善人类的计算环境。

参考文献

[1] "Component Object Model Specification", Microsoft Corp., 1993

[2] "The Common Object Request Broker: Architecture and Specification Revison 2.0", OMG, July 1995

[3] "The Javet(TM) White Paper: Introduction to Java", Sun Microsystem Inc., 1995

[4] "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.

[5] Adler, Richard M., "Emerging standards for componet software", Computer, 28(3), 1995

[6] Nat Brown, Charlie Kindel, "Distributed Component Object Model Protocol-DCOM/1.0", Internet Draft, May 1996

更名启事

经报请中央机构编制委员会办公室批准 (中编办字[1997]51 号)，中国科技信息研究所重庆分所现正式更名为"国家科委西南信息中心"。

本刊主办单位中国科技信息研究所重庆分所，自本期起更名为"国家科委西南信息中心"。

《计算机科学》杂志社