

基于分布式对象的软件构件

Software Component Based on Distributed Object

陈 翀 麦中凡

(北京航空航天大学计算机科学与工程系 北京100083)

分布式对象
软件构件

摘 要 With the development of network technology, software integration is transforming from single machine and centralized system to network and distributed application. It must ask for software component that is a bases of software integration to meet some new demands: supporting distributed application, open and flexibility and so on. In this paper, firstly, we provide an introduction to software component. Then, we discuss the inevitability of software component supporting distributed application and the advantage of adopting distributed object technology. After analyzing two main distributed object standards, we survey some important research problems demanding prompt solution and consider a few of development directions.

关键词 Software component, Software integration, Distributed object

设计

TP.311.5

自从60年代提出软件复用思想以来,就得到了 软件行业的普遍重视,它包括设计的复用(播种复用

$\text{binnode}::1;T$

$v;N$

$r;T$

$T=[\text{binnode}]$

这个例子有一点语法上的差别,在Z中,T是一个新类型,而在VDM中binnode是一个新类型,T则不是.附带一句,Spivey^[9]和Z Base Standard都不是成熟的递归结构,因此可以说,在Z中定义递归不是一件自然的事.

结束语 我们对Z与VDM的差异有了一定的了解,但差别还不止上面提到的这些.有些人认为VDM的工具较多,与对象语言(如PASCAL、C、ADA)等较接近,因此更适合开发一个大的规格;还有人认为Z表述清晰、规范,而且是建立在集合的基础上,因此Z便于使用等.这里牵涉个人的感受、实际应用范围等,我们就不详细讨论了.因为历史原因,使用Z的工业用户比较多一些也是不争的事实.我们希望有更多的人投入到规格说明语言的研究和应用中去,希望我们的工作可以为我国软件设计规格做一点贡献.

参考文献

- [1] C. B. Jones, Software Development: A Rigorous Approach, Prentice Hall Intl., 1980
- [2] Ian Hayes, editor, Specification Case Studies, Prentice Hall Intl., second edition, 1993
- [3] B. A. Sufrin, Notes for a Z handbook, Oxford University, 1988
- [4] P. H. B. Gardiner 等, A simpler semantics for Z, Z User Workshop, Springer-Verlag, 1991
- [5] J. C. P. Woodcock 等, W: A logic for Z, Z user Workshop, Springer-Verlag, 1992
- [6] J. P. Bowen, Select Z bibliography, 同[5]
- [7] A. Welsh, The specification, design and implementation of NDB, Master's thesis, University of Manchester, 1982
- [8] A. Walshe, NDB, Prentice Hall Intl., 1990
- [9] J. M. Spivey, The Z Notation: A Reference Manual, Prentice Hall Intl., second edition, 1992
- [10] 施小英等,一种面向对象的形式化规范说明技术--VDM++,上海交通大学学报,30(6)1995

陈 翀 硕士研究生,主要研究领域:软件工程。 麦中凡 教授,主要研究领域:软件工程,程序设计语言,面向对象方法学,数据库等。

“sowing”reuse)、现有模型复用、领域知识复用和软件构件(收获复用“harvesting”reuse)的复用^[1]。其中,使用软件构件(类似计算机硬件的各种元件)集成软件系统一直是软件业界梦寐以求的事情。自从80年代面向对象技术出现,软件系统的集成在面向对象思想的启示下,取得了卓有成效的进展,软件复用进入了新阶段,出现了以面向对象类库为核心的构件,构件的聚合性、独立性和复用性都有所提高。近年来,随着网络技术的发展,用户对软件系统的需求正逐渐由单机转向网络,甚至更大范围的分布式应用。软件集成的重点也相应地发生变化,对作为基础的软件构件提出了新的要求:支持分布式计算,具有更强的封装性以及开放性。分布式对象技术是近几年发展起来的以面向对象为基础支持分布式应用的软件技术。这种技术对软件构件的新要求具有天然的适应性。

1 软件构件的基本概念

软件构件的设想来自于硬件 IC:即插即用;高封装性,内部细节使用者不必过问;只根据外特性维护,升级重构,以此提高计算机系统的计算能力,减轻升级、维护费用。每个软件构件都是自主的,有其独特的功能,只能通过接口与外界通信。

1.1 软件构件的特征

软件构件不同于硬件 IC,有以下特征:①没有硬件 IC 的天然内聚性,其封装是人为约定的,因此构件粒度大小比较自由,且其升级、修改不一定要扔掉。②表达形式依赖。软件总是寓于某种指令形式之中,有源代码的,目标码的,甚至可以是规格说明层次上的。③运行环境依赖。软件最终要在硬件环境下运行才能体现其功能,愈是低层愈依赖于机器,难于移植。④软连接。软件构件需要一种智能的互操作机制将它们连接起来,这种连接也是软件,其灵活性大,表达能力强。正是由于连接的多样性和丰富表达能力,对标准化的要求更胜于硬件 IC。⑤多侧面性。软件构件表达的语义层次高,可以从不同的语义侧面连接它,完成不同项目的计算,外特性不唯一。

正是由于软件构件自身固有的特性,时至今日,尚无确切的定义。总的来讲,软件构件是可重用的用以构造其它软件的部件的软件单元,可以是被封装的对象类,一些功能模块,软件框架、软件系统模型、软件的文档,如可重用分析件、设计件等^[2]。在软件系统实现级,软件构件和软件本身并没有明确的界限,一个构件在某种情况下可以作为独立的软件运

行,在另一种情况下又可以和其他构件集成在一起构成一个新的软件系统。

1.2 软件构件的分类

软件构件有多种分类方式^[2]。按构件复用的方式,分为黑盒子构件和白盒子构件;按使用的范围,可分为通用构件和专用领域构件;按构件粒度的大小,分为小型(基本数据结构类构件)、中型功能构件、大型子系统级构件;按构件的结构,分为原子构件和组合构件;根据构件重用时的状态,也可以分为动态构件和静态构件。

此外,软件构件不是独立的技术,它同软件开发及程序设计方法等有密切的联系。在不同的时期,不同的发展阶段有不同的含义。从60年代末到80年代初,主要流行结构化模块化程序设计方法学,因此软件构件指构成软件的函数包或过程模块。这种构件层次较低,而且表达依赖和运行依赖强,扩充和复用能力差,对模块进行应用重组较为困难。进入80年代,面向对象程序设计受到重视,出现了以类库为核心的构件。此时的构件具有封装性和继承性,数据和操作集中在一个模块中增强了自治性。但在这一阶段,采用构件集成软件系统还有很多局限性,例如:这种源代码级的构件依赖于语言,开发平台;类库具有的继承性使构件之间耦合度较高,要求操纵构件者比较熟悉面向对象程序设计语言、类库组织及其实现细节。进入90年代,基于对象的脚本(Script)化程序设计逐渐流行,脚本是对目标码软件构件的抽象描述。这种构件的优点是封装性更强,所有的数据必须通过构件对外定义的接口访问;耦合度较高,构件的实现对于使用者透明,相应的脚本语言容易学习和掌握,因此使用者的范围更广。另外,由于这种构件具有动态链接的特性,因此集成后的软件可以做到动态局部升级或修改而不影响整个软件系统。

到目前为止,这三种构件形式都存在,在不同的层次为不同的使用者服务。总的来讲,类库构件发展较为成熟,但即使同一种语言的类库也缺乏一个统一的标准。

2 基于分布式对象的软件构件

近年来,随着网络技术的发展,特别是 Internet 的普及,人们对资源共享以及分布式协作提出更多的要求,例如:工程设计的合作、医疗会诊等。此外,工业生产的体系结构从树形变为网状,贸易的全球化,企业组织的分散化也使分布式系统成为计算机

系统发展的必然。从技术角度讲,分布式系统是由多种计算资源,以一定的互联方式组成的开放式、多平台、可互操作、合作的系统,能够为用户提供一定范围的服务^[1]。一个分布式系统内部资源的位置对用户来讲是透明的,任何用户只要具备关于所用系统最基本的知识,就可以在任何地点、任何时候访问并使用系统中的任何资源。

2.1 分布式软件集成对软件构件的要求

随着软件系统由单机集中式转向分布式,软件开发面临着一系列新的问题。由于分布式系统由多个分散的子系统构成,而且这些子系统之间可能存在着异构性问题(运行于不同的操作系统,采用不同的通信协议,用不同的语言开发等),因此传统的集中式开发模式已经不适用于分布式系统应用开发,分布式系统将这些子系统视为构件,利用这些构件加上临时开发的构件作系统集成,成为分布式软件系统开发的一种重要途径。软件集成需要解决的核心问题是异构资源之间的互操作问题。我们说两种或多种资源是可以互操作的,是指它们能交互地共同执行任务,互相提供服务^[2]。软件集成重心的转变,相应地对软件构件提出了新的要求:①软件构件必须遵循一个统一的开放标准,不同公司开发的构件可以实现互操作,②软件构件必须具有良好的连接性、可卸载性,便于客户裁剪,支持动态连接,③软件构件要求具有较强的封装性,以防止外来的非法入侵,④软件构件之间必须有强壮的互操作机制支持,⑤分布式环境中存在着多种构件,构件命名要有唯一性,并有构件资源的统一管理。

显然,前面提到的函数包、类库以及基于对象构件都不能满足要求。要达到这些要求,一种有效的途径就是在分布式系统中引入面向对象的思想,即采用分布式对象技术来构造软件构件。

2.2 分布式对象技术

分布式对象的主要思想是^[3,4]:在分布式系统中引入一种可分布的、可互操作的对象机制,并且把分布于网络上可用的所有资源看作公共可存取的对象集合,使得不同的对象可以集成在一起。此外,一个对象客户能够通过定义在分布对象模型上的接口来访问分布系统的可用对象。

把对象引入分布式系统是很自然的,对象是一个封闭具有离散特点的实体,对象之间采用消息传递作为唯一的通信方式,特别适合于分布式应用。一个分布式对象系统的逻辑模型如图1所示。其中,对象请求代理(ORB)是对象实现互操作的核心机制,

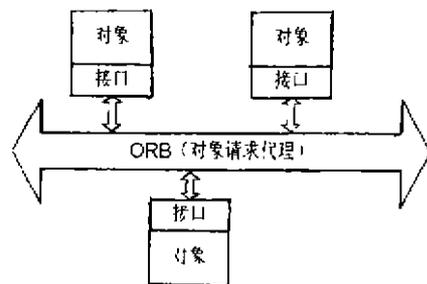


图1 分布式对象系统的逻辑模型

通过它达到分布式系统中资源(即各结点上对象)的访问透明性和位置透明性。对象通过接口定义语言(IDL)定义对外可见的接口,并且实现接口和对象的分离。客户要访问对象数据只能通过接口进行,实现了封装性。另外,虽然IDL本身不是一个可编程语言,但它为程序员提供了语言独立性。在分布式对象系统中对象之间采用客户/服务器的通信模式。一个对象在其生命周期中既可以是客户,也可以是服务器,或两者都是。客户/服务器通常有三种通信途径:对话、消息排队和远程过程调用(RPC)^[5]。其中,RPC是目前分布式处理中广泛采用的通信途径,使得一个客户进程可以调用远程过程为自己服务,形式上类似于传统的过程调用。除此之外,分布式对象技术还包括对象命名和对象管理等一些相关技术。

2.3 基于分布式对象软件构件的优势

分布式对象技术对软件构件的要求有天然的适应性,因为软件构件和分布式对象都同样代表系统可共享的资源。首先,分布式对象中所定义的对象具有较强的封装性,能够满足软件构件的要求。其次,前者对象是在目标码层次上,与整个系统存在着动态链接的关系,适合于构件的灵活卸载性。再次,分布式对象中定义对象接口IDL正是一种开放式标准,只要遵循这一标准,用不同语言开发的构件就可以有集成的可能。最后,对象请求代理机制可以作为实现异种平台构件互操作的核心,而对象的命名和管理也可以作为相应构件管理的可行方法。

总体上讲,基于分布式对象的软件构件是对基于对象构件的扩充。它除了支持分布式系统集成外,还有其他一些优点。首先,基于分布式对象的软件构件有利于软件工业化的形成。这主要是由于构件的标准具有开放性,开发者可以使用各种语言在不同平台上开发构件产品,而构件对使用者又是透明的,使用者只需知道构件的功能以及如何集成即可,因此开发者和使用者可以在不同的层次上,再加上粘

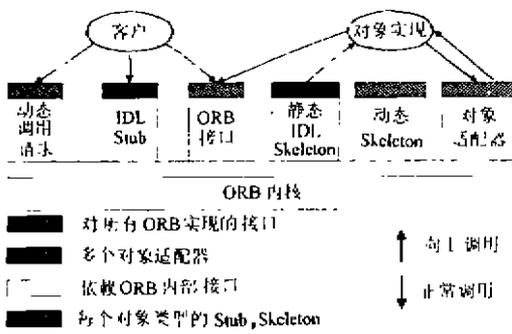
合构件的脚本较为简单,使构件的使用者范围更广,有助于专门从事构件开发的开发部门以及专门从事软件集成的集成部门的形成,层次分工的出现必然会加快软件工业化的速度。其次,基于分布式对象的软件构件利于分布式系统原型化开发,可以缩短发现软件需求语义断层^[1]的时间。

3 两种主要的分布式对象模型

在建立分布式对象标准方面,当前有两种被产业界普遍接收的标准,即对象管理组(OMG)的CORBA(通用对象请求代理结构)及Microsoft公司的COM/DCOM(构件对象模型和分布式构件对象模型),下面分别介绍。

3.1 OMG组织的CORBA标准

OMG是美国主要CASE制售商和一些学者组成的民间组织,成立于1988年,旨在协调厂家制定统一的对象标准。CORBA是OMG组织于1991年发布的构造分布式系统结构的模型,该组织在1995年又发布了CORBA2.0规范,对前一版本作了改进和补充。



CORBA标准实际上是一个规格说明,主要包括:ORB内核,IDL以及低层通信机制。其中ORB内核为CORBA中主要的控制结构,它提供映射(服务器)对象名的命名服务,调用参数的编组,发送请求到服务提供者,以及把翻译好的服务结果回送到请求的客户对象。ORB内核通过对象连接器(Skeleton)骨架与服务提供者对象发生交互作用。CORBA通过RPC提供底层通信机制,当分布式系统运行时,RPC有两种执行方式:动态机制和静态机制。在动态机制中,客户在编译时不必知道可用的服务器和接口信息,而是在运行时间内搜索可用服务器,找到服务器接口,构造请求并发送,最后收到应答;在静态机制中,用户必须在编译时就知道要

访问的服务器接口。CORBA既采用动态机制又采用静态机制是通过接口定义语言实现的。CORBA的IDL称为OMGIDL^[9],用以说明独立于语言的接口。OMGIDL是面向对象的,允许接口表示的抽象,多消息和接口继承,在语言层次上实现了互操作性。

由此可见,CORBA在客户和服务器之间提供了两层互操作性:首先,它将客户和服务器都抽象为对象,所有功能都封装在对象内部,对外提供简单的接口,这些接口被放在接口条目库中(Interface Repository),可以被其它对象以动态或静态方式调用。其次,对象之间通信是通过ORB代理实现,对象不必关心通信细节,由ORB定址发送请求。CORBA还定义了对象适配器^[6]。对象适配器构筑在ORB内核之上,是对象实现访问ORB服务的基本途径,例如:对象实现的注册。另外,对象适配器也使对象实现可以去访问ORB内核不提供的服务,即如果ORB提供了这种服务,适配器简单地对象实现提供一个接口,如果不提供服务,则适配器在ORB之上实现这一服务。在ORB上可以实现多个适配器,例如:基本对象适配器,库对象适配器,面向对象数据库适配器等。CORBA2.0版规格补充了管理对象引用的命名方案,增加了支持分布的ORB通信的网络传输协议(GLOP)^[4]。利用GLOP,不同厂商生产的ORB可以实现连接。

目前,大约有100多个公司(包括IBM,HP,SUN,NEC等)和十多个工业合作伙伴(包括开放软件基金会OSF,UNIX国际,X/OPEN等)表示支持CORBA标准,并且在各自的产品中实现了CORBA技术。

3.2 Microsoft的COM/DCOM

与OMG的CORBA标准类似,Microsoft定义了自己的构件软件体系结构标准。首先在1993年发布了COM,接着在1996年又发布了作为COM网络部分补充的DCOM。COM由一组规范和一个系统级的实现构成,COM定义的接口为一组语义相关的函数,并且同函数的实现体分离。在使用中,接口的定义采用COMIDL来描述^[7]。一个COM构件对象可以定义多个接口,COM正是采用多接口较好地解决版本问题^[9]。COM定义了一个所有构件都支持的特殊接口IUnknown,其他接口都是从这个接口继承得到。

考虑到一个分布式对象系统可能有几百万个需要唯一标识的接口和构件,COM使用全局唯一标识符(GUID)来解决这个问题。GUID与OSF DCE(分

布式计算环境)定义的通用单值标识(UUID)是一样的。此外,COM 还有一个关键部分——注册数据库,每个存在于给定系统的构件都在数据库中注册。

COM 的系统级实现称为 COM 库,这个库实现 ORB 功能,主要完成查找注册数据库,定位激活服务器以及返回对象指针的工作。COM 库可让客户透明地对对象进行通信,而不必考虑对象的位置。这种位置透明性机制是由 COM 服务控制机制 SCM 完成。如果对象位于进程内(In-process),客户可以直接调用它;如果对象在进程外(Out of process),则调用先到达 SCM 提供的客户端代理(Proxy),再由 Proxy 生成对服务对象的 RPC 来实现。在服务端也存在着一个桩程序(Stub),主要负责将接收的 Proxy 调用转化为对服务器的调用并将服务器的处理结果返回给客户代理。对象间通信过程如图3所示。

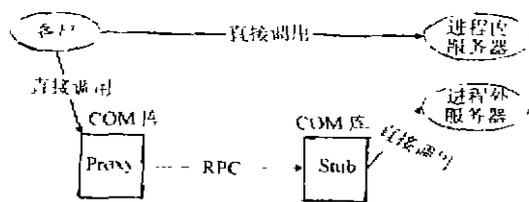


图3 对象间通信过程

DCOM 是对 COM 网络技术的改进,把 COM 局域网应用推向 Internet。主要包括两方面:①支持多种通信协议。DCOM 可以使异种协议的对象交互通信,其中包括 TCP/IP, UDP/IP, IPX/SPX, AppleTalk 等。②将 Java 虚拟机 JVM 集成进来^[6],使 Java Applet 能够和 Internet 上的其他构件通信。

3.3 CORBA 和 COM/DCOM 的比较

CORBA 和 COM/DCOM 有许多相似之处,例如它们都为分布式对象定义了接口并使用 IDL 描述,都通过 RPC 支持动态和静态的对象间通信,底层都有 ORB 支持等。但是,它们之间也有区别,这种区别主要表现在实现接口的方式上。COM/DCOM 规定了构件接口必须由 IUnknown 导出,CORBA 并不规定单一的基类,而是让规范的实现者自己选择。具体如下:①CORBA 是一个规格说明书,没有给出参考的实现建议。相反,COM/DCOM 既包括说明书还有一个实现。②关于实现继承性问题。实现继承性就是在面向对象意义下,通过类层次结构表现出来的继承性概念;而接口继承性则与类层次结构无关,是指对象之间接口重用的能力。Microsoft 认为:实现继承性在松耦合,非集中式对象系统中会带来产生的问题^[7],所以 COM 支持接口继承性而不支持实现继承性。而遵照 CORBA 标准的 SOM/DSOM 系

统却支持实现继承性。③CORBA 的许多实现支持跨平台网络的对象交互能力,而 COM/DCOM 目前只支持 Windows 系列及 Macintosh、Unix,其他几种平台的版本还处在开发阶段。

4 需要进一步研究的问题

基于分布式对象的软件构件作为分布式软件集成的基础,为有效地开发分布式应用提供了必要的前提条件,使传统的集中式软件开发向更广义上的协作式、分散式、层次化开发转变,有利于软件工业化的形成。但是,由于分布式对象技术属于新兴技术,发展还不成熟,软件构件的制作,软件系统的集成还存在着一些问题,有待于进一步研究解决。

首先要解决的是构件标准统一的问题。目前在计算机界存在着多种分布式对象标准,对于软件构件本身也没有统一的定义。因此,异种标准构件之间实现沟通成为亟待解决的问题,例如:基于 CORBA 标准的构件可以和基于 COM/DCOM 的构件通信。此外,不同层次的构件要达到互操作也必须统一在一个层次上,例如:面向对象类库构件编译成目标码后也要求遵循分布式对象标准。

其次,随着应用需求的不断增加,分布式软件系统开发也迫切需要分布式环境下软件集成开发工具的支持。这类工具至少应包括分布式环境中软件构件库的管理功能,构件的功能浏览和构件查询以及分布式应用的调试工具。

另外,由于软件构件的具体实现细节依赖于开发者个人,灵活性强,同一种构件的不同实现也就存在着运行效率、健壮性、资源消耗等多方面的差异,因此,需要相应的质量评测和质量保证方法来衡量构件的优劣。

最后,由于软件构件自身是对某一类计算功能的抽象,除了基本构件和通用构件外,大多数都与不同的应用领域有关。因此,不同领域内软件体系结构的研究就变得极为重要,直接关系到构件的生产质量和构件的可复用程度。简单地说,软件体系结构是对软件更高层次的抽象,主要研究的是软件系统的结构模型、风格和样本,注重全局而不是具体细节。这方面的研究也是实现软件集成的关键。

参考文献

- [1] 白光野、徐崇等,从软件工程的发展看软件自动化,软件学报,第六卷增刊,1995
- [2] 耿刚勇、仲萃豪,采用软件构件技术开发领域应用软件,计算机科学,24(1),1997

20
84-87
A

面向对象的可视化应用平台^{*}

Object-Oriented Visualization Application Platform

王兰波 蔡 勋 李林惠
(国防科技大学计算机系 长沙410073)

科学计算可视化
Visc. 平台

面向对象

TP391.41

摘 要 Visualization in Scientific Computing has been the important developing field of computer graphics. This paper argues the architecture and function of visualization software system using the principles of object-oriented technology, in the direction of system integration, to define a standard structure. So we could use visualization software system as an universal tool in scientific research.

关键词 Visualization, Object-Oriented, Architecture, Message flow

科学计算可视化(Visualization in Scientific Computing, ViSC)是把由数值模拟计算或实验获取的大量数据转化为人类视觉可直接观察的可视化图像,有利于加快数据处理速度,洞察多种物理现象的本质。大多数科学家与工程师因为缺乏编程经验,他们希望拥有高级工具来使他们的精力集中于自己的专业领域。随着可视化研究工作的进展,已出现了一些商用的或学术研究性的系统如 VAS 和 apE,然而科学可视化还处于探索研究的阶段,许多技术细节与标准还有待进一步的完善。本文提出一种面向对象的可视化应用平台的结构与规范,重点突出了其灵活、通用、开放和高可扩充特性,由此而探讨一些可视化系统的标准。我们的目的是使标准的可视化系统成为科学研究的通用工具。

1 可视化与可视化系统

可视化过程一般包括四个阶段:数据过滤、映射、绘制、显示。由于应用领域的广泛性,数据结构与形式有很大差异,这就需要我们对数据进行过滤预处理,

产生形式化的数据;第二阶段是从形式化的数据集中生成虚拟对象“抽象可视化对象”(AVO),也就是把模拟数据映射到 AVO 的属性上,我们可以把 AVO 看作是在空间和时间上产生扩充的虚构对象,它的属性域包括图元、时间、颜色、透明度、反射度、光照度、体素等等,AVO 的时间和几何域同模拟域中相似量可以不对应;第三阶段是绘制,一般由传统的计算机图形学技术实现画图画绘制,最后是图像以动画的方式显示。

可视化系统的一般结构如图1所示。可视化系统有别于一般的软件系统,也不同于传统的图形系统,这是由于使用对象和使用目的的特殊性。在概念上,我们对此应该有一个新的认识。可视化系统主要包括两方面:一个映射工具集和一个管理操作工具集,它一般应具有这样一些特点:①简单性,具有直接操作功能的用户界面。②完整性。③灵活性,对于不同领域的不同问题都可以在系统中构造一个有针对性的应用程序。④扩充性,能方便地引入新的算法。⑤适应性,能方便地建立在不同计算环境中的平台上。

[3] 蔡希尧、刘西洋等,分布系统与分布对象计算,计算机科学,22(3)1995
[4] 石祥滨、张斌等,基于 CORBA 的异构分布信息系统,小型微型计算机系统,18(2)1997
[5] 蔡希尧、边平定等,The Design and Implementation of Remote Procedure Calls, Proc. of the 3rd Pan Pacific Computer Conf., 1989
[6] OMG, The Common Object Request Broker: Archi-

ecture and Specification, Revision 2. 0. July 1995, Updated July 1996
[7] Microsoft Corp. and DEC, The Component Object Model Specification, Draft Version 0. 9, Oct. 24, 1995
[8] Kraig Brockschmidt, What OLE Is Really About, Microsoft Corp. Web Site, July 1996
[9] Microsoft Corp., Java & COM Integration, Microsoft Corp. Web Site, May 23, 1996

* > 银河巨型机软件开发项目