

# 企业级分布计算的基本需求及中间件技术<sup>\*</sup>

Basic Needs for Enterprise Distributed Computing and Middleware Technologies

高全泉

(中国科学院数学研究所 北京 100080)

**Abstract** In this paper, we'll sum up basic needs of Enterprise level distributed computing technologies according to enterprise computing trends at present. Furthermore, we concrete these needs to some characteristics and functional supports which can be used as fundamental criteria to measure middleware technologies/products. Some middlewares which can satisfy these needs to a certain degree are also introduced.

**Keywords** Enterprise computing, Distributed computing needs, Middleware technologies

## 1 引言

近年来,随着网络技术,特别是 Internet 的飞速发展与普及,对分布计算技术提出了很高的要求。企业作为计算机应用的主体,需要使各自的业务系统具备与网络时代相适应的特点,因而必须运用分布计算技术建造新的或改造已有的应用系统。然而,人们通常对分布计算技术的期望过高,已超过了传统的软件环境/平台所能奉献的能力,于是,中间件作为开发和运行分布环境下的多层结构应用系统的一种综合型技术应运而生。形象地说,中间件是处于应用系统及它宿主的系统平台中间以其特有的方式工作的软件,故此得名。功能上,它包括了为实现应用系统的功能所必需的但并不属于业务逻辑的各种基础结构和服务,实现了将应用程序与它所依附的软件平台的较低层细节和基础结构的复杂性之间的隔离。

对企业来说,选用现成的中间件而不是从头搭建无疑是解决分布计算问题的有效途径。问题是,当前哪一种中间件最适合企业的短期和长期需求?若不将企业的一般需要明显地映射到中间件技术的长处,则不会有明确的答案,而且,当前某些厂商的市场宣传和强调各自产品的与众不同往往有片面或夸大之嫌,不利于搞清这一问题。有比较才有鉴别,只有依据业界认同的标准,不仅从理论方面分析,而且运用有说服力的证据做专业和全面的分析比较,才能做出正确的回答。企业级应用面向实际工业和企业领域,支持如:用户大量并发请求、跨网络和异构系统平台、关键业务的联机交易(事务)处理等,在时效及安全性、方便性等方面满足

实用要求,其规模和复杂性远非部门级应用可比。根据国际惯例,能解决实际问题的企业级应用才是计算机应用的高境界,能支持企业级应用的中间件才算名副其实。

在企业应用中,当业务发生变化时,IT 的需求和一般中间件的特点之间具有以下典型的对应关系:

企业规则	IT 需求	中间件长处
通过提供一致的 Views(外观),增加客户友好性	集成客户、部门、厂商、及供应者	业务应用集成
紧随市场变化,调整业务的成本	增量增长 IT 能力	基于组件, N 层环境
通过降低业务成本增加效益	保留并利用已有的 IT 资产	异构 IT 资源的共存支持
将服务尽快地发送到变化的市场	缩短应用开发时间	可共享的服务基础结构

显然,由企业的业务规则所对应的中间件长处应是衡量和评估现有的中间件技术/产品的基本准则。应该说,当今一些主要的中间件技术在不同程度上具备了这些特长,但他们的功能支持及性能内涵不尽相同,甚至存在一些根本性的区别。比如说,对异构 IT 资源的共存支持能力就涉及跨异构/同构平台、跨语言等性能特点,对业务领导者,他们并不特别在意分布计算技术的细节,而是着眼于发现不同中间件技术/产品的真实差异,并将其长处与紧迫的业务需求匹配。如前所述,若不做特别的技术比较,仅凭市场化的宣传,则很难将一些中间件技术的重要特点区别开来。

<sup>\*</sup> 国家自然科学基金项目,批准号 60072006。高全泉 研究员,研究方向为人工智能、知识工程、软件体系结构、中间件技术等。

公共对象请求代理体系结构 CORBA 和分布式组件对象模式 DCOM 是人所共知的一类面向对象的中间件,已被广为采用。有关此二者在企业级分布计算性能方面的评价众说纷纭,莫衷一是,不得要领者不在少数。为了有助于回答前而提出的问题,国际上的 META 组织在相关的技术报告<sup>[1]</sup>中,从企业当前需求出发,对 CORBA 和 DCOM 在支持企业需求方面的性能做了较全面、系统和详尽的比较,有理有据,其评估结果颇令人信服。报告也探讨了当前正在形成的中间件技术,如 CORBA/Java 和 DCOM/Java 之间的协同作用等。这是一种专业性很强的特别比较,将二者最根本的与不为常人所知的区别公诸于世,并分别比较对照,尽现“庐山真面目”。比较采取的基本做法是:(1)将企业需求的功能对应到分布计算技术;(2)描述并分析支持这些基本需求的 CORBA 和 DCOM 的关键元素;并且(3)客观地评价这些元素对于企业部署的长处和不足。

CORBA 从提出至今已有 10 余年的历史,DCOM 要晚一些。应该说,迄今为止,针对企业实际需求对它们做专门的、客观和全面的综合比较并不多见,META 的咨询报告是很好的—例。它评估了这些标准以及与之相关的跨一定范围环境的实现,注意了区分主导中间件风格和产品间的区别,特别强调企业“需求”,编纂了该组织认为是度量中间件技术的“企业至关重要的”特性的现实集。

信息时代,为能做到与国际上的企业级计算技术“环球同此凉热”,必须掌握、运用、发展领先的中间件技术。笔者在相关实践中深感中间件技术对企业级应用的重要性,因其技术特色反映了当今国际上企业级分布计算技术的主流发展趋势。META 组织这些基于深入调查研究的结论,对以电子商务为代表的企业级应用,具有明确的指导意义和现实的参考价值。事实上,如此先进和实用的技术也是所有软件研发人员所应了解和认真学习的,否则何以保证创新。通过对主流中间件技术的基于企业需求准则的特别比较,搞清它们的真实区别,可避免盲目性和走弯路,有助于作出最适合自身实际情况并具有战略前瞻的抉择,构建具有一定预见能力和富有生命力的应用系统。

根据 META 组织这一报告及相关文献<sup>[2,3]</sup>,结合笔者的一些见解,我们形成了内容相关又各有侧重的两篇文档。上篇,即本文,首先讨论企业级计算的基本需求及企业评估中间件技术的准则;然后,介绍可在一定程度上满足这些需求的有代表性的一些中间件技术(包括 CORBA 与 DCOM);下篇,即,基于企业需求性能的 CORBA 与 DCOM 之特别比较<sup>[4]</sup>,以企业准则的具体特性为标准,介绍、分析和比较主流中间件 COR-

BA 与 DCOM 性能的各个方面,总结二者的最关键的差别,给出了相关的结论和前瞻。这种松耦合的结构既为保证内容的系统性和完整性,也为了浏览的便利。

## 2 当今企业分布计算技术的发展趋势

我们先来关注当今企业计算的发展趋势,因为这些趋势直接揭示当前企业计算最迫切的需求。为满足这些需求所必需的软件功能支持及特性构成了中间件技术的基础和核心,因而中间件方案在解决企业问题中具有战略意义。

1)趋势之一:强调基础结构 对企业而言,效率总是第一位的,尽快部署新系统是业务的重要方面。业务决策者往往强调通过“应用购买”来构建自己的应用。然而,购买的应用只提供同等的竞争机会,而不能通过它带来有区别的领先优势。基于 IT 的基本竞争战术包括:

- 加快所购买的应用的部署速度(增加竞争所要实现的首要的利润回报);
- 降低内部开发的应用之实现成本与风险(产生更可靠和可持续变迁的应用系统);
- 削减分布计算成本(使得业务能将资金分配到 IT 以外的用场)。

不难看出,这些战术的共同核心是基础结构的优化。META 组织认为,IT 产生业务的可持续战略领先的惟一正确的途径是在战略应用对象的支持中集中开发基础结构。传统的解决 IT 基础结构的方法、公共的 IT 硬件、软件及支持已被作为一种纯成本的核心——即,一种支持并将当前的业务应用文件夹集成在一起的战术性响应。为了响应专门的应用需求就需要购买或开发相应的基础结构。基础结构涉及低层的系统级如 OS、通信协议等方面的编程,不仅复杂,而且在应用系统中所占比例极大。国际上的 Standish 组织曾对 100 个任务应用系统作过调查,对其编码中的业务/应用部分的代码量和基础结构部分的代码量进行评估。评估结果表明:只有少于 30%的代码与应用/业务有关,而其余部分均归属于基础结构。当业务环境发生变化以及应用业务的改变,都将引起包括基础结构在内的整个系统的改变。理论上,环境和业务改变不会或不应引起基础结构方面的变化,因为基础结构具有较好的完备和稳定特性。问题在于,在一般应用系统中,基础结构与应用代码紧耦合在一起,因此,无论何种改变,都不可避免地引起整个系统的改变,可谓牵一发而动全身。由此可见,基础结构若不分离出来,必然限制业务驱动的改变。

在以往的 IT 环境中,要将上述那些不断发展的应用拼接在一起事实上是不可能的,因为这些脆弱的基

基础结构与应用高度结合(即紧耦合),很难将其分离出来。企业为了获得竞争先机或需要尽快完成向新功能的扩展,必须将基础结构视为一种不只简单地适应,而且能预料业务中的关键的改变的战略财富。如此一种预见性的解决方法隐含了多种战术响应,包含了能促进业务的成长而不是束缚它的新的处理策略,并使得企业能够获得跨多个分布式应用的代码的最大限度的利用,并且建立长期的灵活性以响应业务中即将到来的变化。

对这种需要,IT 必须开发一种阶段性的战略,以便既交付近期的应用服务,也交付长期的战略性的公共技术服务。这种战略通过将公共的IT 财富与可用于单个应用的服务之比极大化来完成。我们可以这样理解,公共服务不只限于技术性的基础结构(如消息传输,TCP/IP 传输服务),而且也包括业务服务(如工作流,日历/进度表,销售跟踪等)。换言之,基础结构包含技术的基础结构加公共服务。这些公共服务构成中间件的基础。那么,由应用开发者自行开发中间件的功能是否可取?答案是否定的,因中间件实现技术以复杂性著称,远非一般应用开发人员所能胜任。Standish 市场组织的相关调查表明,由用户自行开发中间件功能的项目失败率高达90%以上。该组织对项目失败的定义是:项目被取消或没有完成预期的预算、交付使用时间以及业务需求等目标。

由于基础结构在应用系统中所起的作用重要,所占比例极大,将它们逻辑上独立出来成为服务于所有应用体系的中间件技术是当今企业计算发展主流趋势之一。中间件相对于传统的软件环境有了一大飞跃,它大大简化了应用的开发模式、系统体系结构和复杂性,便于业务的管理和扩展。由于提供了应用必需的基础结构和公共服务,用户可以将精力集中于应用业务逻辑的开发。用知识工程的术语来说,用户只需提供业务知识即可,其余事情都由中间件代理了。

2)趋势之二:注重与原有系统的集成 企业计算的另一个主要的趋势则是需要对原有应用系统的利用与集成,这也是中间件技术产生的重要背景和强大推动力。对企业来说,数以百万行的程序代码和兆级别的数据表示了驱动企业当前和将来继续生存的业务智能,这些资源不啻是企业的命脉。由业务的连续性可知,大批地置换所有IT 资源既不现实也不经济。通过扩展原有应用的生命时间,将先前的投资发挥最大效益是至关重要的要求。同样重要的另一个因素是,需要以一种对操作环境影响最小的方式来控制由老的应用向新的应用变换的进度。这样,就要求新老应用之间的互操作能力。

企业在提供其IT 资源之间的互操作性方面面临

实际的风险。可实施的中间件标准提供在这些资源之间的一个“互操作能力合约”,有助于预防因一个厂商的产品的每个小的改动而引起大量的置换的需要。跨平台和跨语言支持都是使得能够在一个富遗留软件环境中达到递增地变换到中间件的关键因素。

3)趋势之三:采用多层体系结构 传统的企业应用软件基本采用客户机和服务器直接连接的二层结构。二层结构在当今的Internet/Intranet 环境下,存在以下固有的不足:①以单一服务器和局域网为中心,系统的可伸缩性差;②将企业信息资源服务直接发送到每个客户带来的复杂性;③在每台机器上安装和维护业务逻辑引起的管理性问题;④软硬件组合及集成能力有限。

为了解决以上问题,人们提出三层及多层结构的模式。三层结构中,将原来的客户层分为表示逻辑层(客户层)和业务逻辑层(应用服务器层),第三层是资源管理器(数据库服务器)。通过将企业服务实现成多层应用可避免二层结构中的问题。多层系统提供增量开发、集成和访问性,是当今企业应用不可缺少的重要特性。中间件采用多层结构开发模式,是构建多层应用的理想基础平台,其最主要的功能是负责客户机和服务器间的连接和通信,并提供一个具体的多层结构的应用开发和运行平台,屏蔽了OS、网络协议等差异,能极大地降低应用开发、部署、管理的复杂性。

### 3 衡量中间件技术的企业准则

事实上,前文提到的企业对分布计算的基本需求即是衡量中间件技术的基本标准,只不过还需将它们映射为更专门和具体的功能支持和度量准则,才利于准确分析、比较并检验现有的中间件。由于中间件技术发展的特殊性,对它们的比较不应只限于标准的比较。技术标准固然重要,能体现对企业明显的可比较的价值,特别在互操作性范畴。然而,对许多需求而言,只比较标准并无太大意义,因为只有这些标准的“活着的”实现,才能够产生有意义的性能和可量测性的数据。

虽然中间件因其强大的分布计算支持风靡于世,但对企业应用而言,不允许选择一种尚未被证明或不能支持最小准则集的中间件技术。为了确实能支持企业计算的需求,中间件必须满足一定的基本要素。企业级中间件事实上是以先前的一系列应用所需求过的那些相同的功能性为基础的,包括:互操作性(处于企业同等的,代表现实混合的系统环境的基础之上);性能和可靠性(构成企业据此来评估中间件的最基本的特性);市场可行性(作为中间件生命周期内,度量厂商支持的产品的成熟性及其长久性的最终检验依然有效)。

这些作为中间件基础的功能性,每一种又可细化

为更专门的特定功能支持和可以反映该特性水平的度量准则,它们之间的包含关系如下:

中间件性能	功能支持和度量准则
跨互操作性	跨平台支持,跨语言支持,网络通信,公共服务
可靠性	事务(交易),消息传递,安全性,目录,容错
性能	可伸缩性
可行性	产品成熟性,厂商前景

#### 4 一些主要的中间件技术概况

我们知道,中间件技术是为适应现代企业分布计算的发展趋势,根据多层系统开发的需要,提供多数应用软件必备的基础结构和公共服务,能方便用户开发、部署和管理的综合型技术。这项技术始于二十世纪八十年代中期,经过十几年的不断发展和完善,中间件已成为当今集分布计算技术和众多现代软件开发技术之大成者。其在现实中,特别是作为电子商务的开发和运行平台发挥着不可替代的作用。它为应用提供了一个灵活的体系结构,能够在此结构之上集成业务应用,台算地构建应用并能灵活地将其扩展。中间件在灵活性方面有着巨大潜力,能够适应众多的设计风格和实现策略,开发最大限度的重用能力,并具备互操作性,即将业务应用连接在一起的粘黏能力,因此,已被作为分布式企业环境中,基于组件的开发和部署的一种不可缺少的基本成分。

当今,具有代表性的中间件有以传统的分布事务监控见长的 TUXEDO,以跨异构平台通信见长、体现分布式计算与面向对象计算结合的公共对象请求代理体系 CORBA,以桌面 Wintel 同质平台对象跨应用使用见长的 IXOM,以及最近提出的 Internet 环境下集组件模式、跨异构平台,兼备事务处理、数据库连接等企业需求的诸多基础结构服务的 Enterprise Java 平台等等,不一而足。以下,对这几种笔者认为有代表性的中间件做一简介。限于笔者水平及知识面问题,这里难免挂一漏万或不得要领,姑妄言之,欢迎批评指正,本意是试图引起对中间件技术的关注。中间件技术在国际上大行其道,在给企业带来极大便利和效益的同时,从事研制和发展中间件的厂商也从市场获得了滚滚财源,这不能不说是中间件的魅力之一。

##### 4.1 事务处理监控器 TUXEDO

最早的开放平台交易中间件当属 BEA 和 TUXEDO,系统于 1984 年在 AT&T 贝尔实验室开发成功。BEA 享有全球最大的独立中间件厂商之美誉,TUXEDO 是最早的也最成熟的中间件产品。当前市场占有率最高,并得到几乎所有的硬件和数据库厂商的支持。BEA TUXEDO 以开放的操作系统 UNIX 为背景,是

一个成熟、稳定、可靠的产品,被绝大多数计算机厂商选为 TPC(事务处理性能委员会)测试的标准平台,足见其功能及性能之好。它具备丰富的开发和管理工具,除具有一般中间件的基本功能外,还具有一些独到之处。如,域管理,将多个服务器组成的应用按功能或结构划分成不同的域,每个域独立地完成域内的操作,域间操作由域网关完成,以提高各个域和整个系统的运行效率。其它技术特点包括:广播通信和时间触发,数据依赖路由,不同系统平台和编程语言数据格式的自动转换,网上数据压缩与加密、动态伸缩及基于 Web 的浏览器管理功能。笔者曾使用过 TUXEDO,印象不错,用它开发 client 和 server 都相当简单,基本上可视为 C 编程的扩展,能大大简化分布系统构建的复杂性。其运行效率及可靠性无可挑剔,是大型企业应用,特别是电子商务的较好的平台。

由于 TUXEDO 提出较早,而不是对象或组件型的,难以获得面向对象或组件的优越性。在传统的函数或过程一级不易把握分布应用的粒度,其向 Internet 的扩展也需借助其他相关技术。人们将 TUXEDO 称为传统的 TP monitor(事务处理监控器)。当然,TUXEDO 只是 BEA 的产品之一,且最近又有新的发展,已经可与许多其它中间件产品协同作用。限于篇幅,本文不做进一步的讨论,可参阅相关文献做深入了解。

##### 4.2 EJB 及企业 Java 平台技术

近年来,Sun Microsystem 公司提出的基于 JavaBean 组件技术的 EJB 及 Enterprise Java 平台是中间件技术的后起之秀。Java 开创了网络计算的新时代,相关的组件技术和企业解决方案迅速发展,受到广泛关注。Internet 普遍存在的特性需要平台无关的组件解决方案。为了达到这一目标,Java 及其 JavaBean 组件模式已经作为基于 Internet 计算的占有统治地位的技术之一,JavaBean 是 Java 向组件方面的发展。组件单位叫 Bean 或 JavaBean,这种组件模式是用 Java 编写的可移植的、平台独立的组件模式。JavaBean 使得开发者能够编写可重用的组件一次,便可到处运行。可被用于编写 applets、Java 应用及其它 JavaBean。JavaBean 的突出特点是组件在 Web 浏览器中图形化可视和可从 Internet 下载运行。在分布计算模型中,可用来构建客户端程序。

EJB(Enterprise JavaBean)于 1998 年提出,是服务器方的组件模式,定义了开发和部署可重用的 Java 服务器组件的模式和 API 规范,EJB 模式是用于开发和部署多层结构的、分布式、面向对象的 Java 应用系统的跨平台的组件体系结构。EJB 规范扩展了 JavaBean 组件模式,提供事务和企业级的服务。在 Java-

aBean 和 EJB 体系结构之间,一个最明显的区别是事件的处理。JavaBean 是为紧耦合的桌面环境而设计,因此使用局部的事件模式(事件源,监听器和事件对象)。它的通信通过同一机器上的 Java 事件模式。显然,这种机制对不安全的网络,等待时间长的连接以及遗留系统的集成是不适合的。EJB 体系结构对分布式的事件提供灵活的支持,允许远程客户通过远程方法调用 API 访问服务器方的功能。为适应分布式对象体系结构中的多层应用,设计了由 EJB 提供资源的自动调节功能。EJB 组件体系结构使得能够快速地产生产于其它事先构建的小的组件的顾客的应用。与之兼容的应用代码是鲁棒的、可伸缩的和安全的,而且具有事务能力。EJB 兼容的程序能够透明地访问在分布环境中的所有的基本接口和服务,并不需要做冗长的编程。

EJB 还不是一个完整的平台,其提供的服务也较有限。于是,Sun Microsystem 又在 EJB 的基础上,提出能适应广泛企业计算的 Enterprise Java 平台技术。Enterprise Java 平台是一个综合型并可互操作的计算环境。该平台的设计旨在满足企业及其多层应用计算环境需求的需要,为容易地开发和管理跨多层环境的所有层提供一个开放的体系结构以及能加速应用部署的组件模式。Enterprise Java 平台的设计建立在“包装和包含”及可重用的组件基础之上。基本的前提是 Enterprise Java 平台可以建立在已有的基础结构服务和已经存在的系统之上,如事务处理,名字和目录,系统管理,安全性,数据库访问和 CORBA 服务。用于与有关不同的基础结构服务连接的主要的 API 是:□用于数据库连接的 JDBC;□用于企业名称和目录服务的 JNDI;□用于与 CORBA 互操作的 Java IDL;□用于 CORBA 事务服务的 Java Transaction Services;□用于 Web 服务的 JavaServer Pages;□用于系统和网络管理的 Java Management API 等。

用 Enterprise Java 平台及其接口编写的应用可以安全地执行,可在服务器之间容易地移动,并能与每个平台上的基本的服务互操作。因此,可移植性、安全性和集成性是 Enterprise Java 平台的三个重要的特点。除了上述一组接口外,企业 Java 平台也提供应用封装和重用的组件模式。其应用模式包括客户方的 JavaBean 组件模式以及服务器方的 EJB 体系结构。

在 Enterprise Java 基础上,Sun Microsystem 又在 1999 年提出了企业 Java 平台第二版 J2EE,可谓好戏连台。有关这方面的研究与发展反映了 Java 系列技术在当今分布计算解决方案中的无可比拟的优势,对其核心技术,我们将另文介绍。Java 技术与其它中间件的协作正在飞速发展,前景看好。对前文所讨论的企业计算准则来说,Java 技术的组件模式和跨平台能力实

属难能可贵。

然而,也应看到,现实中存在大量遗留的非 Java 应用系统,它们可能早于 Java 许多年,与这些资源的互操作和在一个大环境中共存是不容回避的现实问题。仅从语言角度来看,Java 本身是一个羽翼丰满的语言,同时也是用于构建分布式组件的服务集。这样,它被定位为既是一种直接地体现分布的语言,同时也是具备一组基础结构服务的综合性计算平台,身兼二职。Java 的这种跨平台但不跨语言特性,在解决与遗留的非 Java 应用的互操作性方面需要吸收和借鉴 CORBA 及其它技术。这在 EJB 和 Enterprise Java 平台中已得到体现。以下,我们会看到,CORBA 是一种分布和集成技术,而不是一种通用开发语言,这与 Java 截然不同。它假设,各种通用语言将继续被使用和集成,被定位到为已有的语言和应用提供分布特性。在许多重要的技术方面,CORBA 和 Java 技术恰好可以互补,因此,二者的协同结合能产生更理想的效果。事实上,当前此二种技术正互相借鉴融合,取得了很好的结果。

#### 4.3 公共对象请求代理体系结构 CORBA

本节剩余部分介绍 CORBA 及 DCOM 的基本情况,了解这些中间件技术是怎样浮出水面的是重要的。CORBA 和 DCOM 的初始目标提出了一些至关重要的见解,这在很大程度上决定了它们当前的长处和不足,为其未来发展方向奠定了基础。如今 CORBA 已经推出 3.0 版,DCOM 也有相应发展,但本文不可能全都提及,以下主要概括它们在 1998 年以前的相对成熟的技术性能。在本资料的下篇文[4],我们将对此二者作综合性的特别比较。

公共对象代理体系结构 CORBA 由对象管理组织 OMG 于 1991 年提出。OMG 由不同国家、公司等组成,是一个非赢利组织。较主要的成员有 DEC、HewlettPackard、HyperDesk、NCR、对象设计公司 and SunSoft 公司等。该协会的初衷是将错综复杂的网络编程从单独的组件和应用中移走,OMG 成员早在十年前就已认识到,要使对象在企业级是可用的,必须克服几个障碍。

首先,编程语言、操作系统、平台及网络之间存在差异,并且,这种差异在可以预见的未来可能继续存在。在组织内,个别计划希望能够自由选择最适合他们需要的工具来获得最佳效果,此即需要选用不同的工具。然而,各计划之间的差别并非仅此而已,IT 文件夹继续随客户、打包的主框架以及客户/服务器应用一起而变化。将这些各不相同的系统集成在一起的需求在不断增长,然而,在当时此种需求不可能得到满足,因为还不存在这样一种综合的体系结构,它能够屏蔽跨

计划间的差异,使得能够共享和重用每个计划内的软件财富。

其次,若不克服这种跨计划间共享和重用软件资源的复杂性,面向对象的好处在企业级不可能得到实现。面向对象的关键原则,如封装(隐藏)性必须被运用到这种“中间”区域,才能减少由于缺乏基础结构硬件(平台和网络)厂商和应用软件(语言和组件)之间的协定而引起的复杂性障碍。

OMG 提出 CORBA 旨在解决这些问题。之后,在组件互操作能力支持方面,许多重要的贡献源于 OMG,包括:

- OMG 接口定义语言 IDL。IDL 于 1991 年提出,是 CORBA 将语言差别隔离的基本工具。已被一些机构作为软件接口的通用标准,并已由 ISO(国际标准组织 DIS 14750)完全采纳。

- CORBA 互操作性。1996 年的 CORBA 2.0 规范定义了 ORB 间协议 GIOP 作为 ORB 到 ORB 之间通信的基础。在 1995 年的旧金山对象世界展示会上,OMG 的 CORBA net 的成功示范揭开了经过 Internet 的 ORB 到 ORB 的互操作性的序幕。

- GIOP 协议。此协议要求使用基于 TCP/IP 的 Internet ORB 间协议 (IIOP) 作为基本的公共通信机制,该通信机制是所有 CORBA 2.0 兼容的 ORB 所必须支持的。也允许使用其它协议,如 DCE。

OMG 通过发展 CORBA 服务、CORBA 域及 CORBA 设备,在横向和纵向扩展其风格。横向方面,当前对象管理体系结构 (OMA) 在如下四个高层 (CORBA 设备) 范畴上定了 15 个基本 CORBA 服务:

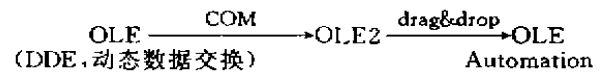
- ◇ 信息管理,如 Query (查询), Persistence (持久性)
- ◇ 任务管理,如 Transaction (事务), Concurrency (并发)
- ◇ 系统管理,如 Lifecycle (生命周期), Trader (交易者)
- ◇ 基础结构,如 Security (安全性), Messaging (消息传递)

纵向方面,CORBA domain 的定义也更加深入,并且处理如 Internet 和各种工业领域,如金融、医药、制造和电信等。这些领域又被深入到子域。例如,在金融领域中的银行和保险业。整个体系结构形成了一个关于分布式业务对象应用和基础结构的综合型模式。

#### 4.4 关于分布组件对象模式 DCOM

从 1991 年起,借助于对象连接和嵌入 (OLE), 分布式组件对象模式从一些已派上用场的技术演变而来。OLE 使用一种叫做动态数据交换 (DDE) 的技术,作为一种简单的桌面剪切与粘贴机制开始其生命历

程。之后,OLE 扩展了微软的组件对象 (COM) 导引而成为 OLE2,为的是提供跨微软应用间的通信和文档嵌入。OLE 也被扩展到支持“drag and drop (拖拽和落下)”及描述能力,使得一个应用能够执行另一个应用 (OLE Automation) 的简单的工作。这一发展简历可用下图示意:



几乎是与此同时,微软的可视 BASIC 产品正在获得作为将可视组件 (Visual Basic custom controls--VBX) 组装到一起的通用模式方面的信誉。VBX 的局限性开始显现出来。VBX 体系结构不是“开放的”,微软也不透露有关 VBX 控制的标准。由于缺乏这种标准,所以厂商们在他们的供给中很难将 VBX 结合进来。这促使微软提供一种关于应用间通信和控制的更一般化的基础结构。

通过将 COM 和 OLE 控制结合定义了这一框架。OCX 成为支持多个应用的 COM 的第一个应用级实现。换句话说,COM 是系统级标准,而 OLE 是构建在 COM 标准上的应用服务 (即,OLE 与 COM 的关系类似于某个专门的 ORB 服务与 CORBA 的关系)。当然,这种解释并不完全,因为 COM 也是用于连接组件的实际的通信技术。

分布式 COM (DCOM) 的出现是为了弥补 COM 在支持远程组件方面的不足。微软曾将 DCOM 描述成“COM with a long wire (带有一根长线的 COM)”。Roger 会议对 DCOM 与 COM 作出了恰当的评价,“二者之间的划分更多的是历史的偶然而不是技术上的深谋远虑”。微软仍将 COM (最近的是 COM+), OLE Automation (现简作 Automation), 以及 DCOM 区分成单独的实体。

Active X 是 OCX 的后继者,被描述成“能用于 Internet 的 COM”。有时,作为覆盖许多如 DCOM 及 Automation 技术的基本技术的总的术语,Active X 也描述一个 Automation 对象,这种对象既可以是可视的,也可以是不可视的。

最后,进入微软的一些产品有: Viper, Falcon 及 Wolfpack。这些产品的目的是将微软的 Microsoft NT 由一个简单的文件和打印服务器,扩展成一个更鲁棒的应用服务器。基本情况如下:

- ✓ Viper, 微软现在的事务服务器 (MTS), 已作为一个 ActiveX 组件的通用服务器环境出现。

- ✓ Falcon, 微软消息队列服务器 (MSMQ), 是微软的应用之间可靠的消息传递的服务器。

- ✓ Wolfpack, 微软基于 NT 簇技术的代码之名

(下转第 14 页)

我们提出的移动 Agent 系统模型的移动机制是可行的。这为今后实现网上购物机器人提供了一个基础模型。其中用到了我们设计的移动 Agent 结构模型和服务环境结构模型,体现了理论与实践的紧密结合,对深入研究移动 Agent 技术提供了实验环境。本系统模型结构清晰,各模块协调良好,设计上是成功的。整个系统用 Java 语言实现,保证了 Agent 能够跨平台运行。

### 参考文献

- 1 Shoham Y. Agent-oriented programming. In: Artificial Intelligent, 1993, 60: 51~92
- 2 Milojkic, et al. MASIF: The OMG Mobile Agent System Interoperability Facility. In: Proc. of the Second International Workshop on Mobile Agents LNCS 1477. Stuttgart, Springer-Verlag, Germany, September 1998. 50~57
- 3 Russell S, Norving P. Artificial Intelligent: A Modern approach. Prentice Hall, 1995
- 4 Maes P. Designing Autonomous Agents. MIT Press, Cambridge, MA, 1990
- 5 Sargent P. Back to school for a brand new ABC. In The Guardian, 12, March, 1992
- 6 Ovum Report Intelligent agent: the new revolution in software. 1994
- 7 Coen M. <http://www.ai.mit.edu/people/sodabor/skideshow/total/p001.html>
- 8 Hayes-Roth. An Architecture for Adaptive Intelligent Systems. AI: Special Issue on Agents and Interactive, 1995, 72: 329~365
- 9 Jennings N P, Sycara K, Wooldridge. A Roadmap of Agent Research and Development. In Autonomous Agents and Multi-Agent System, 1998, 1: 275~306
- 10 Muller J P. The Design of Intelligent Agents. A Layered Approach. Springer, 1996
- 11 Rao, Georgeff. Modeling rational Agents within a BDI-architecture. Proc. of KR-91, San Mateo, CA, USA, 1991
- 12 陆汝钤. 人工智能(下册). 科学出版社, 1996. 423
- 13 Wooldridge M. Time, Knowledge, and Choice (Preliminary Report) M Wooldridge, Muller, J. P. Tamble, editors. Intelligent Agents - Agent Theory, Architectures, and Language (LNAI1037), Springer-verlag: Heidelberg, Germany, 1996
- 14 Renaud P E 著. 客户/服务器系统入门必读. 唐涛等译. 电子工业出版社, 1995. 129
- 15 Nwana H S, Ndumu D T. An Introduction to Agent Technology. Nwana H S, Azarmu N, eds. Software Agents and Soft Computing. Towards Enhancing Machine Intelligent. Springer, 1997
- 16 Jennings N R, Sycara K, Wooldridge M. A Roadmap of Agent Research and Development. J. of Autonomous Agents and Multi-Agent Systems, 1998, 1(1): 7~38
- 17 Jennings N R, Wooldridge M. Applications of Intelligent Agents. In Jennings N R, Wooldridge M, eds. Agent Technology: Foundations, Applications, and Markets. 1988. 3~28
- 18 Bradshaw M. An Introduction to Software Agents. In: Software Agents. Springer, 1996
- 19 Wooldridge M, Jennings N R. Intelligent Agents: theory and practice. The Knowledge Engineering Review, 1995, 10(2) 115~152
- 20 Fisher M. Representing and Executing Agent-based System. In: Proc. of ECAI-94 Workshop on ATAL, Amsterdam Netherlands, 1995
- 21 张润彤, 朱晓敏 著. 电子商务. 北京出版社, 1999. 7
- 22 OMG. Mobile Agent Facility Specification. January 2000
- 23 宋辉, 江峰, 等. Java 服务器程序设计. 清华大学出版社, 1999. 6

(上接第 20 页)

称,这些技术旨在改善可用性、可管理性和可伸缩性。

将 COM, DCOM, OLE 及 ActiveX 这些服务结合起来,形成微软创造的分布式网络互联体系结构 DNA (Distributed interNet Architecture)。

**结束语** 从企业分布计算发展趋势不难看出中间件技术在企业计算中的作用和地位。面对当今各种中间件技术,如何作出最佳选择?我们需要将企业计算的基本需求映射到中间件技术的具体特性。对中间件每种性能或特性,都可进一步具体化为功能支持和度量准则加以衡量,从而能对中间件技术/产品作出客观的比较和评估。为此,首先需要了解一些中间件技术的概

况。这些是本文讨论的内容。如前所述, CORBA 和 DCOM 是当前具有代表性的主流中间件,有关它们企业性能的特别比较在本资料下篇专文讨论。

### 参考文献

- 1 META Group Consulting. CORBA vs. DCOM: Solutions for Enterprise, March 1998
- 2 Sun Microsystem. The Enterprise Java Platform——A Java Adoption White Paper for Developers, Dec. 1998
- 3 BEA Systems. BEA TUXEDO 6.4 Application Development Course, Nov. 1998
- 4 高全泉. 基于企业准则的 CORBA 与 DCOM 之特别比较. 计算机科学, 2001, 28(8)