

# 基于 CORBA 的多数据库系统\*

Multidatabase System on CORBA

石祥滨 张斌 于戈 郑怀远

(东北大学计算机系 沈阳 110006)

**摘要** As the distributed object management method appears, CORBA, which is a standard about distributed object management system, is suggested by Object Management Group (OMG) and has become a de facto industry standard. CORBA makes a important impact on the architecture and implementation of multidatabase system—a special case of distributed system. After introducing the concepts of distributed object management system and CORBA briefly, this paper puts forward a suggestive architecture for a multidatabase system on CORBA and discusses some problems about its implementation.

**关键词** Distributed system, Multidatabase system, Distributed Object Management

面向对象技术在分布系统中的使用,形成了一种构造分布系统的新方法,即分布对象管理方法。这种新方法的主要思想是:在分布网络上可用的所有资源被作为公共可存取对象的集合,一个基于对象模型的分布对象管理系统(DOMS)负责管理这些分布对象,并能以任意方式组合这些对象,以提供新的信息处理能力,这样的分布对象管理系统把网络中的资源像面向对象数据库中的对象一样对待。一个客户不仅能够通过定义分布对象模型上的接口来存取分布系统的可用对象,而且还能在更高的级别上表达对分布系统的请求。

分布对象管理方法的一个最重要基础是分布对象管理系统,它为构造面向对象的分布系统提供一种面向对象的环境,可用以自然地解决分布系统的互联问题及较低级的互操作,并为高层互操作提供基础,所以分布对象管理系统的标准化是一个很重要的问题。很多标准化组织包括官方组织(ISO 和 CITT)、工业联合体(如 OSF、OMG 等)在这方面做出了努力,并已取得了相当大的成果,其中最著名的是 OMG 的 CORBA<sup>[1]</sup>。目前 CORBA 已成为一种事实上的工业标准并已出现了很多 CORBA 产品,如 IONA 的 Orbix, Sun DOE 的 DOMF, HP 的 ORB Plus, IBM 的 DSOM, DEC 的 Object Broker, Ex-

pertssoft 的 PowerBroker 等。很多组织已使用或正在使用 CORBA 产品开发不同类型的分布系统,如 MIND<sup>[2]</sup>, OASIS<sup>[3]</sup>。

多数据库系统作为分布系统的一种特例, CORBA 对它的设计和实现产生了重要的影响。我们认为 CORBA 对多数据库的影响主要表现在两个方面:1)多库系统的体系结构。多数据库系统提供的功能体现为 CORBA 对象提供的服务,功能模块作为 CORBA 对象的对象实现;2)系统实现。与实现相关的很多问题如功能模块(对象实现)的构成方式、进程之间的通讯方式等都需要重新考虑。多库系统的其它原理性问题如模式转换机制、模式集成机制、查询处理策略、事务处理策略等仍可沿用原有的理论和方法,有关这些问题的讨论详见文[8-10]。

本文简单介绍分布对象管理系统与 CORBA, 讨论了基于 CORBA 的多库系统的体系结构和实现一个基于 CORBA 的多数据库系统应考虑的一些问题。

## 1 分布对象管理系统与 CORBA

### 1.1 分布对象管理系统

一个分布对象管理系统(DOMS)是一个面向对象的环境,在这个环境中可以集成自治的、异构的局

\* 国家 863 计划资助。石祥滨 博士生,副教授,研究方向:数据库理论及应用,信息系统集成。张斌 博士,副教授。于戈 博士,教授。郑怀远 教授。

部系统,并且可以开发非传统的应用<sup>[3][6]</sup>。

一个 DOMS 的体系结构中由任意数量的分布节点和客户组成,节点构成了系统的计算资源,每个节点可以支持一个或多个应用程序、数据库系统、面向对象系统中的对象等,客户要求这些计算资源完成指定的操作,客户可以是应用程序、软件工具、面向对象系统中的对象等。一个或多个分布对象管理器作为客户和资源之间的中件服务,它们使系统的计算资源作为对象出现,允许客户提出包括放在系统任何位置上的资源的请求,客户不需要知道资源的位置和实现细节,客户和资源通过软件接口连接到分布对象管理器上,通过这些接口传递请求及把结果转换到局部系统接受的形式。

一个最小的 DOMS 体系结构中包括如下成分:

- 对象实现的集合,是一个实体的集合,每个实体包括数据(状态)及在数据上可执行的操作集。
- 客户接口:用于让客户请求对象(或分布式对象管理器)完成相应的操作、提供操作的参数和接收结果。
- 消息管理(由分布对象管理器提供):传递客户请求到指定的对象。
- 对象接口:在支持客户请求时,让分布对象管理器引发对象的实现。
- 一个分布的计算机系统集合:提供以上部件的运行环境和连接它们的通讯功能。

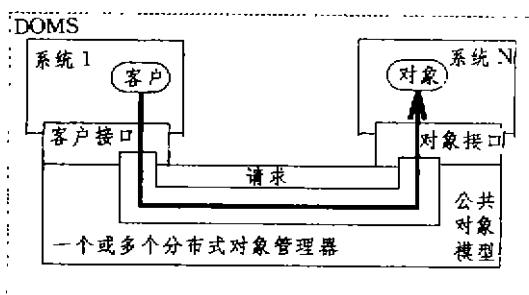


图 1 分布对象管理系统体系结构

图 1 给出了 DOMS 的体系结构,它包括一个公共对象模型,以提供为所有分布对象管理器所理解和支持的抽象的集合;对象的实现必须支持 DOMS 所希望的对象接口,开发者能够借助于称为适配器的软件或 DOMS 提供的对象构造功能,通过开发与公开的接口规定相一致的任意模块来建立这样的实现。

## 1.2 CORBA

OMG 制订 CORBA 标准的目标就是对如上所

述的分布对象管理系统的体系结构及体系结构中的相应成分做出规定。它的基本原则就是实现接口与实现的分离,并且通过类型化的接口实现存取透明性和定位透明性。CORBA 解决了异构分布系统的互联及简单互操作问题,并为高层互操作提供了基础。

CORBA 规定包括如下成分:对象需求代理器(ORB),相当于分布对象管理器;接口定义语言(IDL);静态调用接口(SII);动态调用接口(DII);动态轮廓接口(DSI)。CORBA 的核心是 ORB。这些成分基于一个 CORBA 公共对象模型,CORBA 公共对象模型强调以下特征:封装,任何客户请求的服务(操作),不管是 CORBA 提供的服务,还是由应用系统提供的服务都必须由对象(与应用系统相对应的目标对象和与 CORBA 服务相对应的虚对象)代理,并且通过良好定义的接口存取服务;抽象,具有相同行为特征(操作集)的对象属于一个类(接口类型),允许一个类的一个操作有不同的实现(数据和程序),这一点不同于面向对象数据库类中的概念;多型性,CORBA 的多型性是包含多型性,即一个类的操作对于它的所有子类也是可用的。

## 2 基于 CORBA 的多数据库系统的体系结构

我们认为一个基于 CORBA 的多数据库系统通常具有图 2 所示的体系结构。图中的局部数据库系统可以是一个数据库系统,如关系数据库系统、面向对象数据库系统,也可以是非数据库系统(如文件系统);客户可以是一用户程序,也可以是一查询语言接口;局部模式存储中包含模式转换信息、模式输出信息;集成模式存储主要存储集成模式信息。为了讨论方便,假定公共数据模型采用了面向对象的数据模型。

在图 1 所示的体系结构中,存在两种对象:CORBA 对象和共享对象。CORBA 对象是为客户提供系统服务的实体,由 CORBA 负责管理。实例对象表示多数据库系统提供给用户共享的信息实体,当成员系统是一个面向对象的数据系统时,由局部系统管理;当成员系统是一个非面向对象的数据系统(如关系数据库系统,文件等)时,由包装器(Wrapper)管理。

有三种主要的 CORBA 对象:协调器(Mediator)对象、集成器(Integrator)对象、包装器对象。

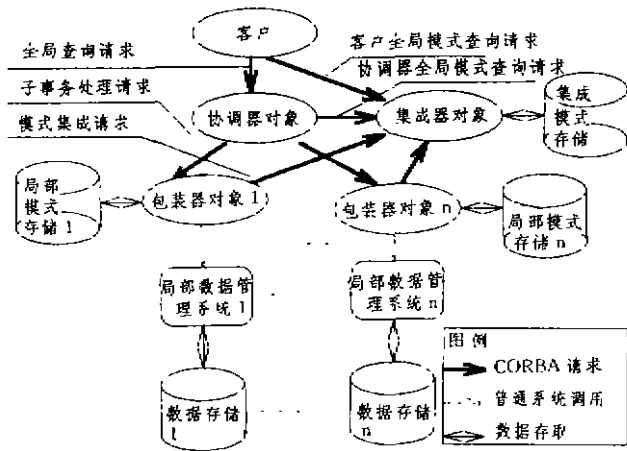


图2 基于CORBA的多数据库系统体系结构

协调器对象负责全局信息共享的协调,直接响应客户的全局查询请求,包括三类对象:全局查询处理(GQPO)对象、全局事务管理对象(GTMO)、全局查询后处理对象(GQPPO)。GQPO负责全局查询语言的语法分析、查询分解、查询优化,最终形成一组针对各个局部系统的全局子查询。在查询处理的过程中,GQPO向集成器对象发出协调器全局模式查询请求获取全局模式信息,从这个角度看,协调器对象是集成器对象的客户。全局事务管理对象负责全局事务的管理,根据一定的规则将全局子查询划分为全局子事务,并负责这些子事务的调度,为了完成子事务的执行,它向包装器对象发出全局子事务处理请求,从这个角度看,协调器对象又是包装器对象的客户。全局查询后处理对象负责对全局结果的最后组装及不能分解到局部场地执行的方法的执行。

集成器负责全局模式的管理,它包括两类对象:模式集成对象(SIO)和模式查询对象(SQO)。模式集成对象根据成员系统提供的输出模式生成全局模式;模式查询对象为客户和协调器对象提供模式查询服务。

包装器为多数据库系统的客户提供一个成员系统的面向对象的接口,它包括三类对象:模式转换对象(STO)、查询转换对象(QTO)、对象数据管理对象(ODMO)。模式转换对象在成员系统为一个有模式的数据库系统时,采取结构映射的方法将局部模式转换为公共面向对象模型表示的成员模式;在成员系统为无模式的文件系统时,采取操作映射<sup>[10]</sup>的方法建立成员模式;在已经建立成员模式后,模式转换对象向集成器对象发出全局模式集成请求,将成员系统的输出模式提供给集成器对象。查询转换对

象在成员系统为一数据库系统时,将全局查询语言表示的子查询转换为成员系统支持的查询语言;在成员系统为文件系统时,将全局查询语言转换为对基本操作的调用。对于不支持查询优化的成员系统(如网状数据库系统、文件等),查询转换对象负责局部查询的优化。对象数据管理对象在成员系统为非面向对象的系统时,负责将成员系统中的实例转换为公共对象模型支持的对象,并负责这些对象的管理。

### 3 系统实现的一些问题

这部分讨论实现一个基于CORBA的多数据库系统时,实现策略的选择所面临的一些问题,有些问题不仅是构造多数据库系统所面临的问题,也是构造其它类型的分布系统所面临的问题。

#### 3.1 公共数据模型的选择

公共数据模型(CDM)的最佳选择是ODMG<sup>[9]</sup>的对象模型,ODMG的对象模型是OMG对象模型的扩充。这样的决定减少了在体系结构方面和模式体系结构方面因模型的不一致可能引起的概念的混淆和不相容。ODMG和OMG对象模型的主要特点是区分类和类型的概念,类型定义对象的接口,类定义对象的实现,允许同样的对象接口有不同的对象实现。这个特点为由于成员系统所采用的局部数据模型和公共全局模型在建模结构上的差异的解决提供了方便条件,特别是为集成无模式的成员系统提供了方便,即可以使用操作映射<sup>[10]</sup>的方法。

#### 3.2 服务的抽取和定义

在设计阶段,需要考虑的一个最重要的问题是服务的确定和组织。服务的确定和组织应基于以下原则:

- 自治性原则。局部系统的持有者有权决定为分布系统的用户提供哪些服务和客户以何种方式调用这些服务。

- 可联编原则。服务必须提供能与对象实现联编的接口。

- 粒度划分原则。通常服务或对象的粒度可大可小,没有一个统一的限定。如当一个参加集成的局部系统是一个面向对象数据库时,可以将数据库中的一个具体对象提供的功能(方法)做为服务,即把数据库中的一个对象做为一个CORBA可管理的对象,也可以将整个数据库提供的功能作为服务,即把整个数据库做为一个CORBA可管理的对象。但服

务或对象的粒度太小,要求 CORBA 产品必须支持如查询、并发控制等公共服务,否则服务是不可用的。最常用的选择是将每个局部系统封装为一个对象,局部系统所提供的服务是查询服务。

服务确定后,下一个问题是如何组织服务。服务按语义相关性原则组织或定义成 CORBA 接口和模块(MODULE)。为实现程序的重用,减少程序的开发量,应尽可能将接口组织成继承层次。

### 3.3 服务的实现

服务的实现即根据服务的定义(模块、接口、操作)编写程序(对象实现),应着重考虑以下问题。

- 对象实现的构成。构造对象实现时不仅与 CORBA 提供的公共服务有关,而且与局部系统服务的定义及局部系统的分布情况有关,不但一个接口可以有不同的实现,而且一个接口的操作所对应的方法也可以分布在不同的网络节点上。

- 实现信息的存储。CORBA 本身只能提供有限数量的持久化存储,包括接口仓(Interface Repository),实现仓(Implementation Repository)等。对象实现相关的其它信息(如局部模式转换信息)如需持久化,则由对象实现的设计者在对象实现中自行管理。

- 对象实现的激活和撤消策略。对象实现的激活和撤消与所采用的适配器(Object Adapter)有关,由于基本对象适配器(BOA)支持很大范围的各种各样的对象实现,通常情况下足以满足我们构造多数据库系统的需要。基本对象适配器提供四种激活策略:分享服务器、非分享(独占)服务器、每个方法一个服务器、持久化服务器。通常最常用的是前两种。分享服务器策略允许一个给定实现的多个激活对象分享相同的服务器,而非分享服务器策略规定在一个服务器上一个给定实现每次只能有一个激活对象。一个对象实现同时满足多个客户需求时,如采用非分享服务器策略,则需建立多个操作系统进程,其优点是实现简单,但容易造成进程数量的爆炸,降低并发程度;而采用分享服务器策略,优点是减少了进程数量,提高了并发程度,但实现起来麻烦一些,因为对象实现一般需采用多线程来实现并发处理。

- 通讯方式的选择。服务方通讯方式的选择是对操作的执行语义的规定,可以是 Best-effort 或 At-most-once。具有 Best-effort 方式的操作不能返回任何结果,而且操作的请求者(客户)不能与操作的实现同步。具有 At-most-once 方式的操作允许客

户以同步或异步的方式调用这个操作。因此,At-most-once 方式是最常用的一种方式,而 Best-effort 方式只适用于那些不需要回答,并且不要求请求必须被成功传输的事件公告操作。

- 接口选择。服务方可以选择基于特定类型的静态轮廓接口或选择动态轮廓接口来联编对象实现。动态轮廓接口实现了服务方程序的动态联编,具有动态联编程序的各种优点,但不容易掌握和使用,用来开发满足特殊需要的基于解释的交互软件工具,调试器等。

- 安全问题。CORBA 只能确保客户需求的正确传输,它并没有加强任何形式的安全性管理,因而安全性管理只能由对象实现来完成。

### 3.4 客户的实现

CORBA 中客户与服务器的协同方式不是传统客户/服务器技术的主/从方式,而是生产者/消费者方式,客户和服务器已无明确的区分,一个客户也可以是其它客户的服务器。因而,客户可以是只提出请求的用户程序,也可以是同时具有客户成分和服务成分的对象实现。

- 接口的选择(动态、静态)。客户方可以选择基于存根的静态调用接口或动态调用接口。尽管 SII 是最简单、常用的方法,但在有的 CORBA 产品(如 DEC 的 ObjectBroker<sup>[7]</sup>)中,SII 不支持异步通讯方式。DII 允许客户方动态构造客户请求,即在编译时客户不需要了解对象类型。在所有 CORBA 产品中 DII 支持同步和异步两种通讯方式。

- 通讯方式的选择。客户方通讯方式的选择即应答方式的选择,客户可以选择同步或异步通讯方式。同步通讯方式是客户向对象实现发送一个请求后,等待一个回答;异步通讯方式是客户向对象实现发送一个请求后,继续其它的工作,在以后的某一时刻接收回答。

结论 目前,我们正在以本文提出的体系结构实现一个基于 CORBA 的 CIMS 信息集成平台,我们认为用 CORBA 实现一个多数据库系统具有如下优点:

1. 由于客户方发送给服务方的消息仅取决于服务方的接口,服务方能够独立地和透明地被改变,而维持接口不变。这个优点使得我们能实现软件的即插即用。

2. CORBA 的程序设计语言的映射方便了用户程序与分布系统提供的功能的联编。

3. CORBA 提供的继承机制实现了程序重用,

# 群体决策支持系统 GDSS: 模型和分类

Group Decision Support Systems: Models and Classification

吴朝晖 莫振饶 潘云鹤

(浙江大学人工智能研究所 杭州 310027)

44-49

TP399

**摘要** 群体决策支持系统 GDSS 作为计算机支持的协同工作(CSCW)的分支,与传统的决策支持系统 DSS 有着较大的差别,前者将面临多个参与者且决策权力是分层的,本文介绍了集中式与分布式群体决策支持系统及其群体决策模型,并展望了未来的工作。

**关键词** 群体决策支持系统,集中式,分布式,群体决策模型

GDSS 模型 分类

群体决策支持系统(GDSS),指的是“一个基于计算机的系统,通过让一组决策者们以群体的形式一起工作,使非结构化的、难以决策问题更易于解决”;它作为计算机支持的协同工作(CSCW)的分支,是一门较新的、具有巨大发展前途的研究领域。

GDSS 是从传统的决策支持系统(DSS)发展而来的,但和 DSS 又有着较大的差别。DSS 的对象是单个用户,且决策具有较高自动化;而 GDSS 面临着多个参与者,并将决策的权力分层次地留给了参与决策者们。

GDSS 主要包含了两方面的技术:一是,包含信息表达和处理技术,例如,相关信息的收集、整理、表示及分析。二是,还需要解决群体协同的计算技术,如通讯,冲突的解决,一致性控制,和数据访问的权

限制控制。所以,群体决策所涉及的领域相当广,还涉及社会学,心理学,计算机硬件,软件,网络技术,以及人工智能。

有了 GDSS,人们就能通过计算机举行会议、进行决策,而不受时间和地点的限制。它和人工的会议、决策相比,应该具有如下优点:更能将人们的注意力集中在相应的问题上;能帮助决策者们理解复杂的问题和环境;更能促进决策者之间的相互交流,增进他们之间的了解和信任;能获得更佳的决策;能达成更高的一致程度。

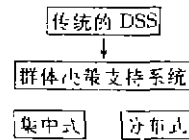


图 1 GDSS 分类

可减少编写程序的工作量。

4. CORBA 支持同一场地和不同场地上的程序间的通讯,因而可将程序的通讯交给 CORBA 完成,减少程序的复杂性。

5. 利用 CORBA 的异常处理功能可方便地处理网络异常和与服务相关的各种异常。

### 参考文献

[1] OMG, The Common Object Request Broker: Architecture and Specification, revision 2.0, July, 1995  
 [2] A. Dogac et al., A Multidatabase System Implementation On CORBA, 6th Intl. Workshop on Research Issues in Data Eng., Feb. 1996  
 [3] J. R. Nicol et al., Object Orientation in Heterogeneous Distributed Computing System, IEEE Computer, 26 (6)1993  
 [4] Edmond Muebrobian et al., OASIS: An Open Architec-

ture Scientific Information System, 同[2]  
 [5] OMG, The Object Database Standard, ODMG-93, release 1.2, 1996  
 [6] F. Manola et al., Distributed Object Management, Intl. J. of Intelligent and Cooperative Information System, 1(1), World Scientific, 1992  
 [7] ObjectBroker, System Integration Guide, Digital Equipment Corp., Aug. 1994  
 [8] 王国仁, 基于自治模块和柔性联盟的大规模数据库技术的研究及一个面向对象的数据库集成系统的设计与实现, 东北大学博士论文, 1995  
 [9] P. Evaggela, B. Omran, Object Orientation in Multidatabase Systems, ACM Computing Surveys, 27(2) 1995  
 [10] E. Berrino et al., Integrating Heterogeneous Database Applications through an Object-oriented Interface, Inf. Syst., 14(5)1989