

开放系统

互操作

软件组件

计算机科学2000 Vol. 27 No. 10

面向对象

开放系统中互操作技术的发展和前景*

Development and Future Perspective of Interoperability Technologies for Open Systems

27-30, 35

刘锦德

唐雪飞

TP31

(电子科技大学计算机学院 成都610054)

Abstract In this paper, role and importance of the interoperability for open systems are clarified. Then, the essential of its capability is explored and classification for the interoperability is given. The concept of Web interoperability is presented first time. Finally, alternative solutions and their future are fully discussed for the computing resource-oriented interoperability.

Keywords Open system, Interoperability, Web interoperability, Middleware

经历九十年代的持续实践,采用“开放系统”的思想来集成大型计算机系统,已成为技术界不争的事实。然而,完善的开放系统的实际形成有赖于“互操作性”的实现。本文将互操作性为对象,探讨近十年来人们对其认识的深化,为有关挑战性难题所作的突破,以及未来发展的前景。

1 开放系统的描述

八十年代中叶计算机技术界出现了“开放系统”这一名称。在随后的商战中,开放系统日显其威力。高举此旗的厂商日趋兴旺,而反其道者日益衰落,终于在九十年代初出现,计算机界所有厂商均自称产品属于开放系统性质。此时用户所面临的困惑问题是:何为开放系统?如何组建自己单位即将构造的开放系统?国外用户先求助于专家,但发现各说不一^[1];又求助于中立研究机构给出开放系统的定义,也被婉言谢绝^[2],因为存在着某种社会压力和制约。

在我国,九十年代初已出现了采用开放系统的社会需要,因此也开始了同样问题的探索。由于并不存在社会压力和制约,我国技术界成功地完成了开放系统描述的研究^[3-5],其研究成果可简明归纳为:

1. 一个开放系统可以用其“轮廓(Profile)”来勾画。
2. 轮廓由七个成分所组成,它们是:系统管理(A)、用户界面(U)、安全性(S)、编程服务(P)、互操作服务(I)、通信服务(C)、信息实体(E);七者可用

“AUSPICE”一词概括、简称。

3. 在构造系统时对其轮廓的七成均应采用适当的标准,其结果将形成开放系统。

该成果使人们能简明地说清:什么是一个开放系统和如何来组成一个开放系统。此一澄清使“开放系统”变成为一个在实际中可操作的名称,解决了长期来困扰用户的问题,使开放系统得到了更为广泛的认同和采用。

2 开放系统的开放程度

多年的实践使人们对开放系统的认识深化,即现有的开放系统与过去的专用系统并非1与0的关系。现实的开放系统在开放程度上大体处于1与0之间。具体的系统究竟处于其间何处,可以利用表1予以判断。

由表1知,一个实际系统若能对AUSPICE七个成分都选用适当的标准,将使此正在建设的系统有可能成为高层次的开放系统(即开放程度近于1);反之,胸无成竹地选用标准,有可能使未来推出的系统成为一低档次的开放系统(即开放程度近于0)。

同时由该表可见,互操作性在开放系统中具有十分重要的地位,因为它是高层次开放系统的标志。实际上,当今现实的开放系统在扩充和发展之际(经常体现为规模变得更大和组成成分变得更为丰富和复杂),系统中必然会出现异质成分,若此时缺少了互操作功能,则这一系统的升级将面临极大的困难,甚至不可能高效和经济地实现。

*)本文工作得到国防军事预研项目的资助。刘锦德 教授,博士生导师,四川省学术带头人,主要从事开放系统和中间件技术,网络多媒体技术等。唐雪飞 副教授,博士,主要从事开放系统和中间件技术,网络技术。

表1 开放系统的开放程度及其判据

开放程度		选用标准的成分							为开放系统实现解决了什么问题
量化表示	模糊表示	A	U	S	P	I	C	E	
0 ↑ X ↓ 1	没有开放								
	初级开放		✓		✓				1 (实现了软件可移植性) 1+2 (实现了用户可移植性)
	中等开放		✓		✓			✓	1+2+3 (实现了互联互通,即为远程合作打下基础) 1+2+3+4 (为实现信息共享、异种机上数据透明动用创造了条件)
			✓	✓	✓	✓		✓	✓
高层次开放	✓	✓	✓	✓	✓	✓	✓	①+②+③+④+⑤+⑥+⑦ (实现了互操作性,即异种机上服务的透明动用)	

3 互操作性的本质功能和分类

由于互操作性在开放系统中的重要性,人们较早就开始对其实现作了试探性的研究^[6],但对其作较系统的理论性研究则尚是九十年代中期的事,在这之前,技术界对互操作性的认识不一,其中有代表性的为^[7]:

1) ISO 的 SPAG: 当几个产品相互合作,协同地为自己所面临的用户提供某个指定服务时,所表现出的能力称之为互操作性。

2) 美国 IEEE: 互操作性是指,两个或多个系统相互使用已被交换了的的信息的能力。

3) IT 经理: 互操作性是指跨越不同的专用系统传递信息,从而使人们在公司的任何部位都能把它们提取出来的能力。

他们各抒己见,互有出入。由于均未触及本质,难以统一而形成令人信服的定义。

国内在这方面所开展的研究工作,成功地抽象出了开放系统中互操作性的本质^[8],即“从异质环境(异种体系结构、异种操作系统、异种网络等)中获得资源的透明动用能力”,于是可以从资源动用的角度,对互操作性给出一个科学的工作定义如下:

在一个由异质实体构成的网络环境中,当应用在网络的节点上运行时,它可以透明地动用网中其它节点上的资源,并借助这些资源与本节点上的资源共同来完成某个或某组任务,这种能力称之为互操作性。

由于掌握了本质,因此人们得以合理地对互操作性的研究进行分类。根据透明动用的资源的不同,可将开放系统中的互操作性分为表2中所示的两大类。其中面向计算资源的互操作性,是指对于网络环境中异构机上的处理功能的透明动用能力。面向信息资源的互

操作性是指对于网络环境中各异构信息源中信息的透明动用能力。

九十年代前期网络规模较小,网中信息源多数为数据库,因此人们所关注的信息源性的互操作性是对于异种数据库的透明访问,习惯上称之为数据库互操作性。九十年代中后期网络规模日趋庞大,网中信息源更重要的是 Web,因此人们开始关注的信息源将是品种五花八门和数量众多的 Web,同时关注的信息源的互操作将是异种 Web 的透明访问。对此学术界尚无专称,根据我们的逻辑可将其定名为“Web 互操作性”。

表2 互操作性的分类

互操作性	}	1) 面向计算资源
		的互操作性
互操作性	}	2) 面向信息资源
		的互操作性
		{ a) 数据库互操作性 b) Web 互操作性

4 面向计算资源的互操作性的解决方案

在历史上技术界首先关注的是两类互操作性中的第一类;也正因为如此习惯上将面向计算资源的互操作性就简称为互操作性(以下内容按此习惯)。实现这一互操作性要借助于互操作中间件(以下简称为中间件)。中间件是一种支撑性软件^[9]。作为一中间件,应具有以下两部分:

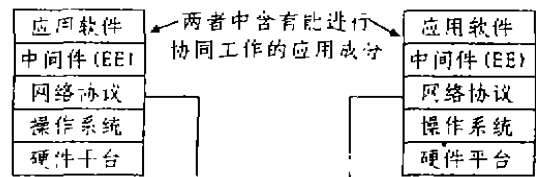


图1 中间件作用的示意图

1) 执行环境软件(简称 EE 软件) 网络上各节点安装了 EE 软件(见图1),各节点上的应用软件之间就可实现协同工作 这时允许各节点可为不同的计算机和操作系统,亦即 EE 软件使各节点下层的设备对应用软件而言变成透明的,所以 EE 软件是实现互操作功能的关键,是中间件的主体部分。

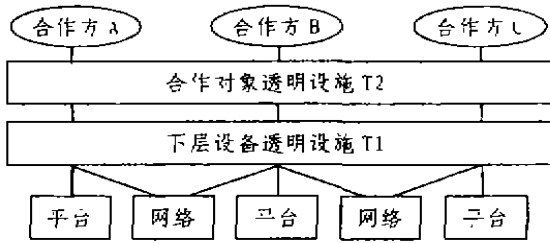


图2 理想的中间件模型

2) 应用开发工具(简称 AD 工具) 应用软件要能透明地动用合作方的资源,该软件中应有作出此种动用的指示,为此必定要有一组工具,备有专用语言和有关的编辑器,它们可用来开发含“透明动用对方成分”的应用软件。所以 AD 工具是一个完善的中间件必备的部分。

一个理想的中间件对于互操作(或协同工作)应作出两点贡献:①提供下层设备透明设施(图2中 T1):有了它,合作一方可不必关心合作的另一方面所用的节点平台(计算机和操作系统)与本节点的差异,②提供合作对象透明设施(图2中 T2):有了它,合作一方不必知道合作的另一方为谁和它在何处,只要说明自己需要怎样的服务,T2就能为其物色一合适的合作方。

十年的努力,技术界已为互操作性的解决,研究和开发出了三种技术,它们是:OSF 的 DCE 技术,ISO/ITU 的 ODP 技术和 OMG 的 CORBA 技术,在此应予以指出:

1) 以上三种技术既能为实现开放系统服务,也能为构造分布系统服务。这是因为互操作性是这两类系统的公共问题,如图3所示。

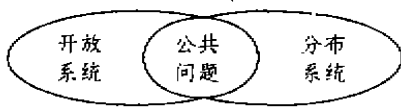


图3 公共问题—互操作性

2) 人们在开发以上三种技术时既重点瞄准了互操作性这一难题,同时也照顾了分布系统实现的种种方面,因此这些技术的名称中常出现“分布式”的定语。

4.1 DCE 技术^[10,11]

DCE(Distributed Computing Environment) 技术在三者中出现最早(90年),当时 OSF 的主力成员 IBM、DEC 和 HP 等积极贡献各自的技术,让 OSF 使其融为一体,企图突破异种机互操作难题,同时也为用户构造分布系统提供框架。该技术由6部分组成:RPC 服务、目录或命名服务、安全性服务、分布式文件服务、时间服务、线程服务,形成了 DCE 中间件。92年 DCE 产品初出问题甚多(难于使用,RPC 设计不周,应用开发工具薄弱,系统管理工具不全等),得不到用户的认同,随后风云变化,DCE 失去财政资助,无力改进和发展,技术停滞不前,生存产生问题。直到96年 OSF 与 X/Open 合并形成 Open Group,DCE 才得到转机,并于97年初推出新版 1.2.2(改进了过去的不足)。DCE 出现较早,因为未采用面向对象技术,再加多年的搁浅,使其在技术上处于明显的劣势(因后来者均采用了面向对象技术),因此今后作强势发展已属渺茫,目前只有凭其有特色的安全性服务(Kerberos)吸引一些用户。

在此应指出,DCE 虽已妥善提供了下层设备透明设施,但在提供合作对象透明设施上层次不高,它只能提供白页服务(靠 CDS 和 GDS 构成目录服务),而不能提供黄页服务,因此从互操作性的角度来评价,DCE 只能算作下乘技术。但不管如何,DCE 是一个可以解决互操作性的方案。

4.2 ODP 技术^[12,13]

DCE 前期的挫折,促使技术界另探途径,于是出现了 ODP(Open Distributed Processing)技术,当时英国剑桥大学设立了 ANSA 项目,以异构分布式处理环境为对象,研究建立这种环境所面临的问题及其解决方案,93年他们成功地理清了问题,建起了实验模型。该模型展示了崭新的解决互操作性的方案——借助于 ANSA 交易器,后者可成功地实现“合作对象透明设施”(即图2中 T2)。这一成就引起了国际技术界的兴趣,特别是受到了一些国际性标准化组织的关注。

ISO 和 ITU 在八十年代末已认识到:“过去 ISO 所制定的 OSI 参考模型,只支持了系统之间的相互连接和沟通,有其局限性,因此有必要另行创建一个面向应用的参考模型来对付未来将常见的“应用的分布”,这一参考模型将保证分布着的应用之间能实现互通和互操作。”于是启动了 ODP 标准的建立,但其实际进展缓慢。ANSA 的成功和示范,使 ISO/ITU 见到了样本,有了参考,于是标准的制定得到加速,95年推出了 ODP 标准的草案,96年形成了正式的国际标准:ISO 10746/ITUX. 900。标准一出,使 ODP 技术得到了技术界的公认,同时 ANSA 的成果也得到了发扬。

设计一个特大的分布系统所面临的问题成堆,将

其梳理清楚实属不易。ODP 技术采用视点 (Viewpoint) 的方法,将一大系统从五个侧面(企业、信息、计算、工程、技术)来阐明;同时用引入多种透明性(最多可达8种)的方法,来将分布系统种种复杂性加以屏蔽。这些方法使系统设计者既能洞察问题,又能深知在何处引用标准。在实现分布系统时,ODP 技术采用已定义了的4类24种功能来构造出 ODP 系统。在这些功能中有着交易器功能,它就是以 ANSA 交易器为蓝本。

因此,ODP 技术实际上是一种大型的异构分布式系统的设计技术,但在最终实现的难点上也提供了具体的解决方案——交易器技术(以另一标准《ODP 交易功能》的形式出现)。由于交易器为系统所提供的是黄页服务,所以从互操作性的角度来评价,ODP 可以称为上乘技术,是优秀的解决方案。

4.3 CORBA 技术^[4,15]

今天 OMG 已宣称自己的目标为“将 CORBA (Common Object Request Broker Architecture) 变成无处不在的中间件”而奋斗。但当年 OMG 成立时的初衷则是,在软件行业中推广面向对象技术的使用,并建立相应的规范,促进人们对软件组件概念的认同,使软件集成和软件重用变成行业的习惯。为此它在90年推出了参考模型 OMA,OMA 中的核心成分 ORB(对象请求代理)是起着“插板”的作用;人们将若干精选的软件组件插入“插板”,就可使它们形成一个集成的软件。90年代初欧洲成功地利用面向对象技术解决了互操作性,启发了美国技术界,于是 OMG 也扩大了其目标,将解决异质网络中应用之间互操作性也作为了己任。新的任务要求 OMA 增强原有的 ORB 功能,于是 OMG 提出了 CORBA 以替代 OMA 中原有的 ORB。

根据目前规范,CORBA 的组成成分有:ORB 核心、ORB 界面、IDL 存根、IDL 框架、动态调用界面(DII)、动态框架界面(DSI)、对象适配器(OA)、界面仓库、实现仓库和 ORB 间互操作作用 GIOP/IOP。在 CORBA 中 ORB 核心起着通信总线的作用,利用它客户可以透明地访问服务方对象所提供的服务。实现这一透明的关键是利用 IDL 语言,它可用来定义对象所能提供的服务。根据 IDL 定义,可用 IDL 编辑器产生 IDL 存根和 IDL 框架,将 IDL 存根置于客户与下层平台之间,同时将 IDL 框架置于服务方对象与下层平台之间;这将使客户调用远程对象犹如作本地调用一样,同样服务方对象为远程客户服务也象是作本地服务一样。IDL 存根和框架所提供的是静态调用,它有执行速度快特点。然而,也可不用静态调度而改用有灵活特点的动态调度,这时客户通过 DII 直接动用 ORB 核心所提供的请求机制,动态地向对象发出请求,而不必事

先准备好 IDL 存根(当然这时客户应提供有关操作和参数类型的信息)。由于目前的 ORB 对于静态和动态两种调用方式是公用的,所以这一增强型 ORB 被称为 CORBA,在此的“C”来自“公用”(Common)。

应该指出,CORBA 中间件的设计意图是:依靠静态调用机制来实现白页服务,而动态调用来完成黄页服务,但在实际中发现,用动态调用机制进行黄页服务必须借助于界面仓库,后者的介入带来了开销,导致动态调用的有效执行速度远比静态的慢(有数量级之差)^[15]。所以从互操作性的角度来衡量,CORBA 提供的并非上乘技术,因为它的黄页服务性能很差,为此 OMG 只能在上述组成成分之外,另行引入 ODP 交易器技术来克服这一缺点。

由于经历不同,CORBA 技术的策划者呈现出在多方面逊色于 ODP 技术的策划者。后者显得善于高瞻远瞩和长于疏而不漏,所以开发的 ODP 技术既适用于大型、甚至极大型分布系统,又未见其频频补漏,反之对于 CORBA 技术,则一再发生:异厂商 CORBA 之间无法沟通(后来靠引入 GIOP/IOP 予以解决),对多媒体信息无支持(正在靠加入 AV 流机制来解决),黄页服务差(将借助引入交易器来改善)等。所以 CORBA 要成为完善的中间件,尚有一个不断的改进和充实过程。

但也应注意到,OMG 是一个善于联系社会且工作十分高效的机构,CORBA 依托于它,将会从产业界、用户界和技术界得到广泛支持;再加 CORBA 规范日趋完善,具有活力的开发厂商如 IONA 等十分活跃,所以预期上述改进和充实将顺利进行,CORBA 中间件最终将获得宽广的发展前景。

结束语 本文展示了互操作性在开放系统中的关键作用和重要地位,因为它是决定一个开放系统开放程度高低的技术成分,由于互操作本质含意是“从异质环境中获取资源的透明动用能力”,所以互操作性可分为两大类:面向计算资源的和面向信息资源的,面向信息资源的互操作性又分为两种:数据库互操作性和 Web 互操作性,后者是在因特网和其它远程网大发展的今天,新呈现出来的技术研究领域。

经九十年代的研究和开发,使(面向计算资源的)互操作性这一技术前沿有了突破,解决难题的工具是中间件。当前的主要解决方案有三种方案:DCE 技术,ODP 技术和 CORBA 技术。

DCE 是一种可实用的互操作性解决方案,但由于它属于第一代中间件(注:DCE 与后两者相比,虽然在出现时间上相去不远,但由于它未采用面向对象技术,因而与后两者在技术上有了“隔代”之差),因此未来发展空间有限,只有在需要利用它的特色性能(安全性)

(下转第35页)

$$N_{max} = \left[\frac{21\bar{D}_i - d_i}{d\lambda(2\bar{D}_i - d_i)} \right] \quad (2)$$

实际上,交易器的所有操作中,负载信息收集的时间最长(约3ms,其中约2ms的时间用在通信上,对交易器所在主机没有特别的要求,一般的主机即可),令 $d=0.003s$, $D_a=0.1s$, $\lambda=0.9$,通过(2)式可以计算出 $N_{max}=364$,即一个交易器可以支持364个结点。对于更大规模的分布式系统,可以采用以下几种方法扩大交易器的伸缩性:①交易器使用多线程的方式,特别是负载信息的收集可以采用单独的线程完成;②可以使用联合交易的方式(这是交易器的灵活表现之一),将系统划分为不同的域,由不同的交易器处理不同域内的请求,同时交易器之间通过联合交易的方式,处理跨域之间的请求;③由于CORBA的IIOP协议使用的是TCP协议,通信成为交易器的伸缩性的瓶颈,采用更快的网络也是扩大规模的方法之一。

从上面的分析可以看出,集中式的方式也可以支持大规模的分布式系统,[6]中也得到了类似的结论。

结束语 随着CORBA应用范围日益扩大,负载平衡成了其应当考虑的问题。本文首先对CORBA负载平衡所使用的各种策略、机制进行了详细的讨论。在此基础上,提出了一个基于交易服务的动态负载平衡

模型。本模型对于交易器的扩展,对用户透明,从透明性和性能两方面考虑,我们建议使用集中式算法,文章的最后对该模型的伸缩性进行了分析,分析指出它可以支持大规模的分布式系统。由于采用集中式的方式,交易器的可靠性对整个分布式系统至关重要,如何提高交易器的可靠性是我们下一步的工作。

参考文献

- 1 Object Management Group. Common Object Request Broker: Architecture and Specification, 2.2 ed., February 1998
 - 2 Zhou S. A Trace-Driven Simulation Study of Dynamic Load Balancing. IEEE Trans. on Software Engineering, 1988, 14(9): 1327~1341
 - 3 Slama D, et al. Enterprise CORBA. Prentice Hall, 1999
 - 4 OMG. CORBA services. Common Object Services Specification. February 1998
 - 5 Shivararri G, et al. Load Distributing for Locally Distributed Systems. IEEE Computer, 1992, 25(Dec.): 33~44
 - 6 Lin Hwa-Chu, Raghavendra C S. A Dynamic Load Balancing Policy with a Central Job Dispatch (LBC). IEEE Trans. on Software Engineering, 1992, 18(2): 148~157
-
- (上接第30页)
- 之处会有用武之地。ODP技术是由国际标准化组织精心策划下所开发的解决方案,它面向大型和特大型的分布式应用系统。随着时间的推移,它将受到愈来愈多的人的重视和采用。有人预见,它的ODP参考模型将成为OSI参考模型的有效补充。CORBA技术是一个正在蓬勃发展的解决方案,虽然CORBA中间件需要不断地改进和充实,但由于存在极有实效的技术社会的支撑,因而它的发展前景十分宽广。
- ### 参考文献
- 1 Gill P J. What Is Open System? Uniform Monthly, 1992, 12(1)
 - 2 Hach D. Forcing an Open Systems Definition. Open Computing, 1994, 11(9)
 - 3 刘锦德,唐雪飞. 军用计算机开放系统轮廓描述的研究: [电子科技大学研究报告]. 1992
 - 4 Lau Jinde. Open System Connotation. 19th Intl. Computer Conf, Hong Kong, 1996
 - 5 刘锦德. 对于开放系统内涵的澄清. 计算机应用. 1997, 17(6)
 - 6 Millikin M D. DCE--Interoperability Now! Network Monitor, 1990, 5(1)
 - 7 Gray P. Open Systems and IBM. McGraw-Hill, 1993
 - 8 唐雪飞. 开放系统中互操作性的研究: [电子科技大学博士论文]. 1996
 - 9 苏森,唐雪飞. 开放系统中的互操作性. 计算机应用, 1997, 17(6)
 - 10 Gill P J. DCE Status Report. Uniform Monthly, 1995, 15(3)
 - 11 De Nitto K. Richer Feature Set for DCE. Uniform Monthly, 1997, 17(1)
 - 12 Farooqui K, et al. The ISO Reference Model for Open Distributed Processing and Standards. Computer Network and ISDN Systems, 1995, 271: 1215~1229
 - 13 ISO/IEC & ITU. Reference Model of ODP, Part 1 Overview. May 1995
 - 14 刘锦德,苏森. CORBA技术的新发展. 计算机应用, 1999, 19(5)
 - 15 OMG. The Common Object Request Broker: Architecture and Specification, Revision 2.2. Feb. 1998
 - 16 Orfali R, et al. Instant CORBA. John Wiley & Sons, 1997