

CORBA 3.0

组件

应用程序

Internet

计算机科学 2000 Vol. 27 No. 3

11-14

CORBA 3.0 新特性分析

Study on New Features for CORBA 3.0

夏承刚 严筠永

(中国纺织大学计算机科学与工程系 上海 200051)

TP317

TP393

Abstract As infrastructure of Distributed Object Computing, CORBA Specification is now used more and more widely. With development of distributed applications and other technologies, new specification is adopted to improve the performance and make it more easy-to-use. The Component Model, Portable Object Adapter and Asynchronous Messaging etc., which were waited for a long time, are committed in the new version. The detail will be examined to show how these technologies work and add new features to CORBA 3.0 Specification.

Keywords POA, Component model, Asynchronous messaging, QoS, Object by value

1 CORBA 3.0 框架

OMG 的 CORBA 标准作为分布对象处理的规范已有若干年了,随着分布系统的开发应用,原有规范也暴露出不足之处,同时为了适应越来越复杂的分布式应用程序,需要对规范作补充和改善以增加灵活性和适用性,满足不断增长的应用程序的要求。

到 1998 年底,OMG 已相继确定了 Portability 2.3 Rev、OBV 2.3 Rev 等一系列推荐模型,基本上确定了 CORBA 3.0 规范的整体框架。在 3.0 规范的先期版本 Pre-Release 中,新增了几个典型的特性,主要有支持可移植对象适配器(POA)、分布式组件模型(CORBA Component Model)与脚本语言、以值传对象(OBV)、服务质量控制(QoS)、异步消息传输协议支持以及和因特网技术的整合(Java/Internet)等。

新版本增加这些技术功能的主要目的是要提高 CORBA 的适用性,扩大规范的应用范围,并改善对网络需求的适应性。按 Dr. Soley 所言,3.0 版本的 COR-

BA 将简化 ORB 的使用,无论对有经验的分布计算设计人员还是一般的程序员而言都有新的特性和功能改善。

2 CORBA 组件模型

组件模型是新规范中增加的一个显著特性。OMG 通过组件模型定义了即插即用式 CORBA 组件的界面、机制和程序开发框架,组件模型封装了简单对象的创建、生命周期和事件等内容;允许客户方动态地访问组件对象的功能、方法和事件;借鉴并集成了现有的组件技术(JavaBean、ActiveX),提供了到这些组件的映射和互操作性。

CORBA 组件模型提供了运用组件组装应用程序的较完整机制,统一了简单程序的开发框架,显著减少了使用 CORBA 开发程序的难度,亦实现了和 Java 及其它组件技术的紧密集成。其目的就是要与微软的 ActiveX 和 COM 在可视化和简便性上相抗衡,争取更多的用户市场。

的层次上,以崭新的视角来观察和思考计算机科学理论和实践中正在发生的和将要发生的各种问题。相信它一定会对广大的计算机科学工作者的研究工作有所帮助,并产生出更多更深刻的理论。

参考文献

- 1 Stankovic J A. Misconceptions about real-time computing: A serious problem for next generation system. IEEE Comput., 1988, 21(10): 10~19

- 2 Shin K G. Real-Time Computing: A New Discipline of Computer Science and Engineering. Proceedings of the IEEE, 1994, 82(1)
- 3 Sinha A. Client-Server Computing. Comm. of The ACM, 1992, 35(7)
- 4 Sha L. Industrial Computing A Grand Challenge, Computer, Jan 1994
- 5 Brooks F P Jr. 软件工程之实质性困难和偶发性困难. 计算机科学, 1988(3)

2.1 CORBA 组件

OMG 在新规范中并没有具体确定组件的概念。组件并不是与某个 CORBA 界面或对象的一一对应,而是 CORBA 对象(包括界面)和相关功能的集合。组件具有属性和实例标识,支持事件通知机制,支持自省(Introspection),可映射到现有的商用组件模型(包括 JavaBean)。

组件可以有多个 IDL 界面,组件中的一个对象可以拥有多个视图,程序通过选择界面定义访问该界面中对象的函数和操作,从而提供组件在逻辑上独立的服务,提高编译的效率和程序的可维护性。

用户可以选择购买开发商提供的组件包,包含可视编程工具,建立起自己的应用程序。

2.2 CORBA 脚本语言

和其它脚本语言(如 VB Script)类似,CORBA 脚本语言是书写基于 CORBA 组件的应用程序的解释执行语言。

由于脚本语言自然简单,容易理解和使用,将有更多的程序员使用它来编写程序;由于不需要关心如内存分配和回收、指针及编译连接等复杂过程,开发者可以集中于逻辑上的开发整合;由于不需要经过编译就可以立即执行,可以加快程序特别是原型系统的开发。

脚本语言的出现将极大地扩展 CORBA 的使用者范围,减少开发应用程序的开销,并和其它对象模型一样使 CORBA 更具适用性。

2.3 以值传对象(Object By Value)

以“值”传对象是通过传值而不是传名(引用)的方法来传递对象。CORBA 允许对象引用作为参数传递,但对对象本身并不传到被请求方;客户程序为调用远端对象所发出的请求和远端对象所返回的结果,需要通过网络来回传输。通过以值传对象可以方便开发者更直接高效地传递对象信息;这种方法更接近于一般编程语言如 Java、C++ 等的相应机制,容易实现应用程序级的无缝连接。

新规范在 IDL 中增加一个新的“值类型”构件来支持以值传对象。值类型基本上是从界面派生出来的,且和结构结合,既包含界面的特点又包含结构的特点。值类型包含数据成员和操作,更像是 Java 或 C++ 中的类。同界面一样,值类型可以继承,也可以继承界面的操作。当值类型作为参数传递到远端操作时,就被复制到接收方的地址空间,这样当地进程对值类型副本中操作的调用便是本地的,也就是说,不同于对 CORBA 对象的调用,值类型调用不需要在网络上传递请求和答复。

把值类型的数据成员作为参数传递到远端应用程序,类似于传递其它数据类型,但传递操作就不那么容

易了。由于 CORBA 应用网络的异构性,客户方和服务方使用的编程语言和计算机平台都可能不同,将值类型操作的实现从客户方传到服务方或反之都会有困难。使用 C++/C 语言的话,接收方应用程序必须在编译时或通过其它方法获取值类型的有关信息。通常在广域网上,可以用 Java 开发客户方和服务方的实现,就能容易和安全地下载兼容的字节代码,实现对象以值传递。

值类型可以定义数据成员和操作,适合封装简单的数据,如链表等;但这是一种新的 IDL 类型,以值传对象规范并没有经过严密定义,仍显得粗糙。实现以值传递真正的 CORBA 对象还要解决对象定位、对象标识以及和对象适配器交互等问题,规范还需要不断改进才能做到既满足要求又简洁易用。

3 可移植对象适配器(POA)规范

CORBA 在 2.1 以前的版本中只定义了基本对象适配器(BOA),实现的是基本的功能。厂商和开发者在基本对象适配器的实现与使用中,发现了许多缺陷和不足,各厂商独立改进自己产品的功能,导致了不同厂商 ORB 实现间的不可移植性。

CORBA 2.2 定义了 POA,它是由 ORB 厂商和用户在使用 BOA 的经验基础上联合开发的。其主要目的是确保用编程语言实现的伺服程序(servant)及服务应用程序可以在不同厂家的 ORB 之间移植,它通过提供大量的灵活性实现了这个目标,满足许多不同类型 CORBA 服务应用程序的需要。POA 规范的引入对这个版本的标准具有较大的影响。

3.1 POA 的基本点

POA 介于 ORB 和服务应用程序之间,主要处理三类实体:对象引用、对象键与对象 ID、和用以具体实现对象功能的伺服程序。客户方通过对象引用向目标对象发出请求,对象引用中包含了用以标识对象的对象键部分。对象引用是在全局范围内唯一的;而对象键是在服务应用中唯一地标识目标对象。由于一个应用程序可以有多个 POA,对象键帮助 ORB 将请求分配到拥有目标对象的 POA。POA 使用对象键中的对象 ID 确定目标对象与伺服程序间的关联,将请求分配到伺服程序。对象 ID 的作用域仅仅是某个特定的 POA。由伺服程序执行请求所得结果,经 POA 和 ORB 返回客户方。POA 在三者之间起到了转译功能,把来自各地的对 CORBA 对象的请求派发给相应的伺服程序。

3.2 POA 支持的特性

3.2.1 持久与短期对象 CORBA 原来定义的对象可存活在创建或激活它的进程的有效时间以外,在多个服务进程的生命期之间持久存在,因而称为持

久对象;有时 CORBA 应用程序需要产生一些临时的对象,使其生命期受限于进程或创建它的 POA 的生命期之内,这种称为短期对象。POA 支持这种新的对象,由于 ORB 不必保留短期对象的定位轨迹,也不必用服务进程重新启动对象,CORBA 程序的开销就可以减少。

3.2.2 显式和按需激活 一个应用程序只有少量对象时,只需在开始侦听请求前创建这些对象的伺服程序,并在 POA 中注册。POA 支持这种显式的对象激活和伺服程序的注册。

另一种极端情况下,对于具有大量对象的应用程序,只需激活那些真正收到请求的对象。这种方法极大地减少了资源的使用,只有接收请求的对象才被激活。应用程序使用伺服管理器,在需要时创建伺服程序。

伺服管理器是本地的 CORBA 对象,用于管理伺服程序的生成,既可以为 POA 提供新创建的伺服程序,也可重用原有的伺服程序,负责将产生的伺服程序返回给 POA。

3.2.3 多线程支持 基本对象适配器(BOA)的一个不利点是它不能实现任何与多线程有关的应用程序的问题,而在服务程序中普遍使用多线程以便更容易地为并发请求服务。使用 POA 的应用程序可以创建两种不同线程模型的一种:ORB 控制模式,允许底层 ORB 实现多线程;和单线程模式,确保 POA 中所有对象的所有请求在单一线程中完成。

BOA 没有多线程定义;POA 的多线程模型是一个巨大的进步,这使程序更加灵活。POA 还能提供更好的多线程控制,将来的标准将扩展 POA 规范,使之更灵活。另外在 POA 中,还有许多对 BOA 作了改动的地方,如支持动态支架界面 DSI 等等,其实所有这些自由都来自创建 POA 时赋予的策略(如建立的是持久对象还是短期对象,多线程的还是单线程的)。为使应用程序能使用不同的 POA 策略集,应用程序可以使用多个 POA,其中有且仅有一个根 POA。

4 异步消息规范

几年来,缺乏异步消息通信一直被认为是 CORBA 规范中的明显漏洞。在庞大的分布式计算网络中,各通信系统不是所有时间都是可靠连接的,异步传输可以不依赖于客户方和服务方的紧耦合,因而适合应用于大规模分布系统。由于只提供了同步的调用方式,CORBA 被认为不能胜任大型分布系统。

CORBA3.0 在消息服务中新增了异步消息通信协议,并提出实时 ORB 概念,这是 OMG 采纳的实质性规范。异步消息支持 QoS 级别许可和异步请求调用,增强了 QoS 保证和通信效率,提供了分布对象系

统中异步消息的管理机制。

4.1 异步调用

原先的 CORBA 标准对请求提供了三种同步通信模式。(1)同步方式。客户方发请求,在响应到达前阻塞。(2)延迟同步。客户方调用操作,继续运行,以后回来查询或阻塞等待响应的返回,这种方式只能在 DII 方式下使用。(3)单工方式。客户方调用一个操作,没有响应返回,ORB 尽量保证请求的执行。这种做法是允许 CORBA 请求在不可靠的传输协议上(如 UDP)传递。新的 CORBA 消息规范在这三种请求方式以外,增加两种异步方式:

(1)回调(callback)方式。客户方在每个请求调用中提供一个额外的对象引用作为参数;当响应到达时,ORB 使用该对象引用将响应发回到应用程序。

(2)轮询(polling)方式。客户方调用一个请求,操作立即返回一个值类型(valuetype),可用于客户方程序独立地轮询或等待响应的结果。

客户方可以在 IDL 产生的静态桩中使用回调和轮询式这两种异步方式,而不象延迟同步方式必须在 DII 中使用,这是一个显著的优点因为静态调用显得更自然。

4.2 时间独立调用

为允许调用未激活或未连接的对象,CORBA 消息规范在标准的 GIOP 和 IIOP 协议上增加了中间路由代理。这些代理具备存储转发(store-and-forward)请求的能力,在对象未激活或不可用时保持消息;响应在发请求的客户方不可用或断开时也同样可以被这些代理保持。请求的生命期可以由其 QoS 策略控制。

4.3 QoS

CORBA 对客户应用程序隐藏了某些底层网络细节,特别是对对象定位、建立网络连接和发送请求的过程,给予 ORB 厂商的实现自由,但是,隐藏这些细节使 CORBA 应用程序难以控制底层 ORB 发送消息的质量。

新的 CORBA 消息规范矫正了这个不足,允许客户程序指定发送请求的质量控制(QoS)、消息队列和消息权限。程序使用 QoS 策略告诉 ORB 如何处理请求中各种情况(诸如对象找不到等)。ORB 通过选择不同的底层传输方法或者自动提高即将失效的请求的优先级等方法,确保通信的服务质量。QoS 策略的作用范围可以是 ORB 内的所有请求,或者是线程内的请求,还可以是向某个对象发送的请求。

4.4 实时 CORBA

实时 CORBA1.0 中扩展了一类新的 ORB,称为实时 ORB,允许开发者更直接地控制 ORB 的资源分配,保证端对端通信的可预测性和灵活性。虽然这个规

范不是很成熟,但给扩展实时 ORB 提供了稳定基础,为 CORBA 进入进程控制和事件严格响应的实时应用领域打开了大门。

5 与因特网技术的集成

目前因特网和 Java 技术应用日益普遍。为了显示 CORBA 与因特网紧密结合的能力,也促进了 Java 语言的进一步推广,OMG 从商业角度出发,吸收主要厂商成熟的系统资源和技术,在 CORBA3.0 中首次加入 Java 到 IDL 的映射标准,使许多原先使用了 Java 语言开发的系统可以产生 IDL 接口,提供给其它的二进制应用程序通过 IIOP 远程调用这些 Java 应用对象。

有了这个 Java 到 IDL 的映射标准,开发者可以完全使用 Java 来开发整个分布式应用系统,然后再从 Java 类自动生成需要的 IDL,方便了程序员,特别是熟悉 Java 的编程者,同时也为 OMG 的几个大厂商保护了原有系统的投资。

防火墙规范定义了 ORB 之间通过防火墙代理 (Proxy) 进行 IIOP 传输的界面,包含了防火墙过滤功能。使用防火墙可以控制 CORBA 程序有限地访问因特网和内联网,控制 IIOP 网络传输的安全,这方便了将来扩展 CORBA 在因特网上的安全性应用。

为使运行在 DCE 系统上的系统尽快地转移到 CORBA 平台上来,也便于使用 DCE 系统的组织扩展原有应用程序,OMG 在新规范中定义了 DCE/COR-

BA 互联的方法,通过将原有 DCE 应用程序集成到 CORBA 环境,保持两种系统的应用在程序级上互联工作,为使用 DCE 环境的组织提供了遗产环境支持。

结束语 总的说来,CORBA 在不断改进,吸收集成多种技术,以保持其领先的技术和活力,适应更广泛的应用程序,3.0 版本可以说在规范上迈进了一大步。当然,这些变化也不是都很好,不排斥其中有许多商业上的考虑,对现有技术并没有彻底吸收和改进,就补充到新的规范中去,一方面使规范越来越复杂和庞大,另一方面由于庞杂不能排除其中有问题的内容出现,只能不断更新,最终还使用户受累。在使用其中的先进技术的同时,也需要采取去芜存菁的方法才能完善自己的成果。

参考文献

- 1 Vinoski S. New Features for CORBA3.0. CACM,1998,41(10)
- 2 王芳兴. 公共对象请求代理结构(CORBA)的分析与研究:[硕士学位论文].(上海)中国纺织大学,1997
- 3 Siegel J. CORBA Enterprise Model. CACM,1998,41(10)
- 4 Object Management Group. Object Management Group Outlines CORBA 3.0 Features. www.omg.org/news/pr98/9-9.html,1998
- 5 Vinoski S,Schmidt D C. An Introduction to CORBA Messaging,SIGS C++ Report,1998,Object Interconnections, column 15

(上接第6页)

道德、社会和文化生活产生广泛的冲击,随着我们进入二十一世纪,随着信息技术更强大和更广泛的使用,这种趋势会日益加速。所以必须支持相应的社会科学研究。这种研究要求计算机科学家和社会科学家之间有更多的交流,而且,这种研究对各方面都是有帮助的。这个领域的研究主题包括:

- 面对 Internet,各种形式的非传统管理方式和行业自律,其效果如何;
- 信息技术对工作本质的冲击和影响,包括远程办公、工作质量以及“虚拟”团队的参与等;
- 人种、收入、民族和地理原因,在访问信息技术中造成不同的影响;
- 电子商务对市场结构、供应商-客户关系、生产率、价格稳定性和雇用关系等的冲击。

3.2 信息技术劳动力

目前,对于具有信息技术技能的劳动力,其需求快

速增长,政府、科学研究和教育部门必须为解决这个问题作出更多的贡献。

二十一世纪的工作要求劳动者具有所需的基本的计算机文化。在需要大量具有较高专业技能和较深知识的信息技术人员的同时,也有许多与信息技术有关的工作,不需要太专业的信息技术知识,例如网络管理员、初级程序员、问询处的操作员、系统分析员和网页设计员等,这类人员的需求会更多。

参考文献

- 1 Information Technology Research: Investing in Our future, Feb. 1999. Available at: <http://www.ccic.gov/it2>
- 2 Information Technology for the Twenty-First Century: a Bold Investment in American's Future, Jan. 1999. Available at: <http://www.ccic.gov/it2>
- 3 Networked Computing for the 21st Century, Mar. 1998. Available at: <http://www.ccic.gov/it2>