

# 医疗信息系统分布式技术平台研究

叶枫

(浙江工业大学计算机信息系统研究所 310014)

吕旭东 段会龙

(浙江大学生物医学工程研究所 310014)

摘要: 分布式对象技术为解决医疗信息系统的集成提供了一条崭新的途径。本文在分析比较当前主流的分布式对象技术的基础上, 提出了医疗信息系统分布式对象技术平台, 介绍了该平台的软件总线层和对象管理层的设计与实现。

关键词: 医疗信息系统 分布式技术 CORBA EJB

## 1 引言

医疗信息系统的对象是与人体相关的医疗信息, 而医疗信息的复杂性和多学科性决定了医疗信息系统具有不同于其他类型信息系统的特点。采用通用的分布式技术平台不能从根本上解决医疗信息系统的集成问题, 需要进行适合医疗特点的分布式技术平台设计。

本文在比较当前主流分布式对象技术平台解决方案的基础上, 结合医疗信息的特点提出了医疗信息系统分布式对象技术平台, 并给出了它的总体设计, 最后对该平台的应用和结果分析进行了介绍。

## 2 医疗信息系统分布式对象技术平台

医疗信息系统属于迄今为止世界上现存的分分布式信息系统中最为复杂的一类, 这是由医疗机构本身的目标、任务和性质决定的。它不仅要和其他所有信息系统一样处理伴随人流、财流、物流所产生的管理信息, 提高整个医疗机构的运作效

率, 而且还应该支持以病人医疗信息记录为中心的整个医疗、教学、科研活动。它有以下特点:

(1) 信息复杂。病人信息是以多种数据类型表达出来的, 不仅需要文字与数据, 而且时常需要图形、图表、影像等等。

(2) 信息的安全、保密性要求高。病人医疗记录是一种具有法律效力的文件, 它不仅在医疗纠纷案件中, 而且在许多其他法律程序中均会发挥重要作用, 有关人事的、财务的、病人的医疗信息有严格的保密性要求。

(3) 数据量大。任何一个病人的医疗记录都是不断增长着的、图文并茂的书, 而一个大型综合性医院拥有上百万份病人的病案是常见的。

(4) 高水平的信息共享需求。一个医务工作者对医学知识、病人医疗记录的需求可能发生在他(她)所从事的全部医、教、研的活动中, 可能发生在任何时间和地点, 而一个病人的医疗记录也可能被各有关临床科室、医技科室、行政管理部门所需要。

(5) 系统异构性。整个医疗机构中存在着大

量异构、自主且分布的数据/影像设备, 数据库、信息系统、智能化诊断/治疗应用程序。

这些特点决定了整个医疗信息系统应用环境是一个完全分布式的结构。根据这些特点, 我们提出医疗信息系统分布式对象技术平台, 见图1。在软件总线层采用CORBA总线技术, 基于CORBA规范实现了ORB核心(包括GIOP/IIOP)、DII、接口库、实现库、DSI等部件, 它屏蔽了系统层的异构性, 为运行在其上的软件对象提供一个彼此交互的总线机制。在对象管理层基于EJB的思想构建支持多语言环境的对象管理器, 使之既具备EJB技术所具有的优秀对象管理性能, 又能适用于多种语言开发的软件对象, 通过CORBA软件总线和对象管理器的结合, 可以实现分布式技术平台的总体解决方案。

该平台具有以下特点:

① 它具有很好的可扩展性, 对功能的增加或修改只需添加或更新相应的对象即可, 开发人员只需关心业务逻辑。

② 易于进行医疗环境中存在的大量异构系

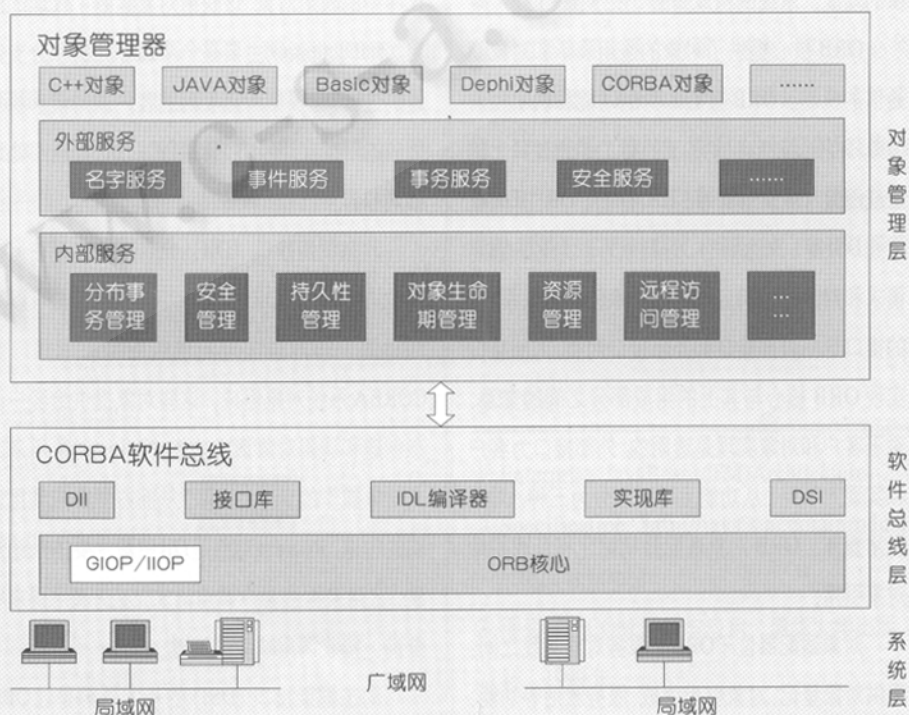


图1 医疗信息系统分布式对象技术平台总体设计

统的集成。由于CORBA的平台无关性、语言无关性和生产商无关性,医疗信息系统只要提供对象化接口,就能方便地被其他系统调用,实现功能集成。

③ 它具有很好的安全性。CORBA本身提供的安全服务以及OMG组织专门为医疗行业指定的医疗安全服务,对于保证医疗数据的安全性十分有效。

### 3 软件总线层设计

软件总线层基于CORBA技术实现,CORBA总线系统由ORB核心(包括IIOP协议)、IDL存根、动态调用接口、IDL框架、动态框架接口、对象适配器、接口库、实现库和IDL编译器构成,其中IDL存根和IDL框架由IDL编译器自动得到。当客户程序发出一个调用请求时,客户程序可以使用动态调用接口(DII),也可以使用IDL存根。对象实现通过IDL框架或者动态框架接口(DSI)的回调来接收客户的调用请求。在处理调用请求时,对象实现使用对象适配器和ORB核心所提供的功能。系统的构成如图2所示。

ORB核心提供了对象的通用表示和对象间通信的机制。ORB核心本身的功能和机制相对简单,对不同对象机制的支持能力是通过其上的组成部分(对象适配器、IDL存根、DII、IDL框架和DSI等)来达到的,这样就可以为客户调用请求和对象实现响应的传输带来较高的效率。它的接口包括内部接口和外部接口两类。内部接口实现ORB核心与其上的组成部分之间的交互,对于客户和对象实现是透明的;外部接口为客户和实现方所见,从功能上可以划分为三种:对象引用操作、ORB对象适配器初始化和获取初始对象引用。

对象适配器位于ORB和服务对象实现之间,提供对象登记、对象引用生成、服务激活等功能。每一类对象可以有一个相对应的对象适配器,如

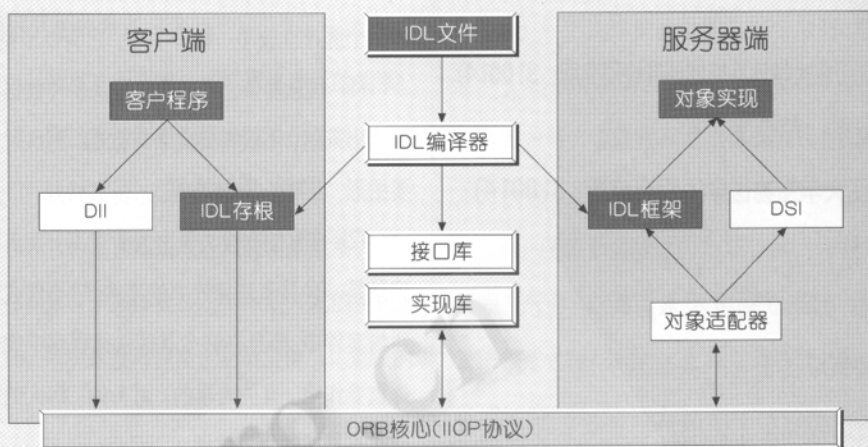


图2 软件总线层设计

基本对象适配器(BOA)、程序库对象适配器和面向对象数据库适配器等。

IDL语言是一种独立于编程语言、下层网络和具体实现的数据类型描述语言。IDL到具体编程语言的映射通过IDL编译器来实现,它把IDL文件自动生成具体编程语言的IDL存根和IDL框架。IDL存根建立了目标服务对象在客户方程序中的映射对象,该映射对象起到了代理的作用;而IDL框架则主要是一个循环判断和分支处理的过程,它根据所接收到的客户调用请求找到相应的对象实现及方法,执行该方法并返回处理结果和异常信息。

动态调用接口(DII)允许对象在运行时动态地加入CORBA环境中,并使客户可以顺利地调用该对象实现提供的服务。它由一组CORBA系统对象组成,这些对象提供给客户动态创建和调用对象请求功能,NVList对象用来定义对象操作的参数的列表,从而方便客户调用请求的创建,Request对象是DII中最主要的系统对象,它用来构造客户调用请求、发送调用请求和查询、接收调用响应的关键。

动态框架接口(DSI)的基本思想是让ORB核心/对象适配器对所有的对象调用请求都通

过向上调用同一组接口例程来调用实际对象实现中的操作。DSI的主要目的是为了支持ORB间互操作机制中不同域间的转接桥,因为这些转接桥在编制时无法预知需支持的目标对象的接口。为实现DSI,必须实现一个或者多个具有Dynamic Implementation接口的对象,并向对象适配器登记由该对象处理某种IDL接口的调用请求。对象适配器在接收到调用请求后,通过已登记的Dynamic Implementation对象将请求传递给DSI,DSI通过ServerRequest对象来获取调用的操作名、调用参数列表以及返回值存放位置等信息,从而可以直接调用目标对象的具体操作。

接口库和实现库是为了协助ORB接收处理请求和响应而产生的,主要用于存放IDL接口描述、定义信息以及保存激活对象所需的信息,是ORB系统的辅助部件。接口库用来存储、发布、管理相关对象接口定义的集合,客户通过动态地查询接口库,可以获得目标对象的类型描述,在DII和DSI方式下十分有用。实现库存储供ORB核心和对象适配器定位和激活对象实现所需要的信息,往往和接口库作为整体一起实现。

## 4 对象管理层设计

CORBA软件总线为不同软件对象间的交互提供了一个底层机制,但是如何组织并管理这些对象,使之更高效地运行和配置,则是对象管理器的任务。这些任务主要分为两类:一类是内部服务,即由对象管理器本身调用的服务,用于进行对象运行环境的管理,如对象生命期管理、对象位置透明性管理等;另一类是外部服务,即需要由用户显式调用的服务,如名字服务、事件服务等。在具体实现时,对象管理器其实就是特殊类型的CORBA对象,它对运行在其中的所有服务对象和用户对象进行管理。

内部服务对用户对象来说是不可见的,它由对象管理器直接调用,配合对象管理器共同完成对象运行环境的建立。包括后台分布事务管理、后台安全管理、后台资源管理和对象生命期管理、后台持久性管理、后台远程访问管理和后台多用户支持等。每种服务在实现时都涉及一个复杂的工作机制,以生命期管理服务为例,它的工作机制如图3所示。一个用户对象在对象管理器中共有四种状态:停止、激活、事务、空闲。停止状态即表示用户对象尚未启动;激活状态即表示用户对象的某个实例正被调用;事务状态表示用户对象实例的正在进行事务操作;空闲状态表示用户对象正在等待调用。这四种状态之间的转

换可以分为两种情况。一种情况是由其他对象调用该对象的某些方法时,触发对象管理器自动进行的。另一种情况完全是对象生命期管理服务自动控制的。当用户对象在激活状态下超过一定的时间未被调用,对象生命期管理服务就自动调用它的`srvPassive()`,使之转换为空闲状态下,直至下一次调用时才被激活。

外部服务是供用户对象显式调用的功能服务,如名字服务、事件服务、事务服务、安全服务等。这些服务为用户对象提供了一系列通用的功能服务,便利了系统的开发。它们基于CORBA的COSS规范实现。以名字服务为例,它使得对象能够通过“名称”进行注册和查找对象。在名字服务中,名称是由一个序列构成的,该序列中的最后一个部分表示被注册或查找的对象,其余每个部分都用于定义一个名称上下文,一个名称和一个对象之间的联系称作名称绑定。解析名称就是在某个名称上下文范围中确定名称所对应的对象,绑定名称就是在所给定的名称上下文中为A对象创建一个名称绑定。名字服务的基本运行过程如图4所示。

名字服务包括两个接口: `NamingContext` 和 `BindingIterator`, 它们主要包括七部分的功能: 绑定名称、解析名称、解除绑定、创建名称上下文、删除名称上下文、列出名称绑定和遍历所有名称绑定。



图4 名字服务运行过程

## 5 总结和展望

本文通过对目前流行的三种分布式对象技术平台解决方案进行分析,提出了医疗信息系统分布式对象技术平台。它既能为运行在其上的软件对象提供一个与硬件平台、操作系统、通信网络和编程语言都无关的中性平台,使异构的软件对象彼此之间能够交互,又能提供一个具备优越性能的对象管理器,使软件对象在运行时获得各种自动化的服务,令对象交互更为便利。

医疗信息系统分布式对象技术平台完整、具体的实现工作是一个复杂、庞大的工程。为了能够给出一个实现实例,我们采用了成熟的市场产品,即基于CORBA和EJB相集成的方案。这种方案虽然能基本达到要求,但仍不理想。主要体现在两个方面:首先,实现时必须采用RMI到CORBA的映射,对运行效率产生了一定的影响;其次,非Java对象不能获得与Java对象一样优越的对象管理性能。进行医疗信息系统分布式对象技术平台的完整实现是我们今后努力的方向。 ■

## 参考文献

- Enterprise JavaBeans(TM)Architecture--2.0 Specification, <http://java.sun.com/products/ejb/2.0.html>.
- 潘爱民, COM原理与应用, 清华大学出版社, 1999年。
- OMG编著, CORBA系统结构、原理与规范, 电子工业出版社, 2000年。

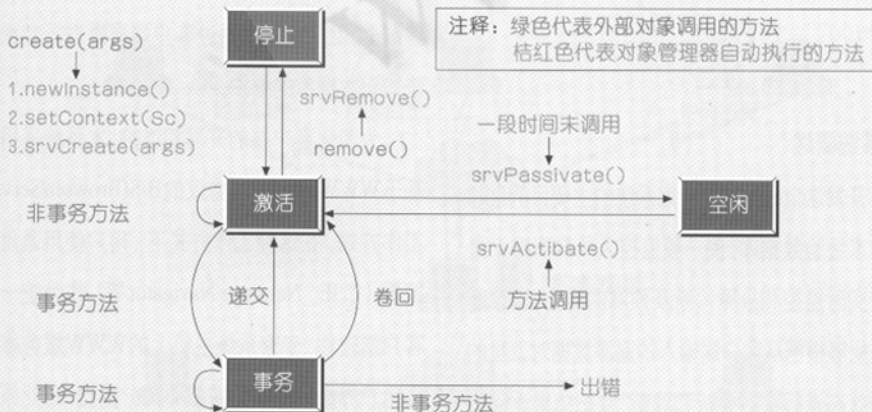


图3 对象生命期管理工作机制