第30卷 第5期 计 算 机 工 程 2004年3月
Vol.30 №5 Computer Engineering March 2004
·软件技术与数据库· 文章编号：1000—3428(2004)05—0090—03 文献标识码：A 中图分类号：TP393

从面向对象的应用到Web服务的迁移

杨 滨，赵文耘
（复旦大学计算机与信息技术系，上海 200433）

摘 要：通过对Web Service和再工程的研究，提出了一套从面向对象的应用（遗产工程）迁移到Web服务的通用的再工程过程，针对其核心技术——域包的抽取，提出了解决的方法。

关键词：面向对象；Web Service；再工程

Migration from Object-oriented Application to Web Service

YANG Bin , ZHAO Wenyun
（Department of Computer and Information Technology, Fudan University, Shanghai 200433）

【Abstract】 This paper provides a process for migration from object-oriented application to Web Service. It gives a solution on domain package extraction technology. The technology can be used to analyze object-oriented applications.The paper also gives a solution to assemble new Web Service.

【Key words】 Object-oriented; Web service; Reengineering

1 概述

许多商业应用面临着与其它程序互操作性的问题。因为不是所有的应用程序都是使用COM或.NET语言编写的，并且都运行在Windows平台上，事实上大多数商业数据仍然在大型主机上以非关系文件(VSAM)的形式存放，并由Cobol语言编写的大型机程序访问。而且，目前还有很多商用程序继续在使用C++、Java、Visual Basic和其他各种各样的语言编写。除了最简单的程序之外，所有的应用程序都需要与运行在其他异构平台上的应用程序集成并进行数据交换。这样的任务通常都是由特殊的方法，如文件传输和分析、消息队列，还有仅适用于某些情况的的API，如IBM的“高级程序到程序交流(APPC)”等来完成的。在以前，没有一个应用程序通信标准是独立于平台、组建模型和编程语言的。然而Web Service的出现完全改变了这种状况，它提供了一种跨平台、跨语言、跨操作系统的分布式计算技术，使得不同的操作系统、编程平台和对象模式的框架结构之间具有互操作性。Web Service 是目前分布式计算发展的方向。

面向对象方法从认知学的角度来看，符合人们对客观世界的认识规律；开发的软件系统易于维护，其体系结构易于理解、扩充和修改；继承机制有力支持软件的复用，所以面向对象方法的出现很快受到计算机软件界的青睐，并成为自20世纪90年代延续至今的主流开发方法。随着应用系统越来越大、越来越复杂，遗产系统越来越多，当前的软件开发已经从传统的需求、开发、测试、维护发展到通过面向对象的复用技术，通过再工程技术将现存的遗产系统重新构造为新的应用这样一种模式上来。

1.1 面向对象方法

面向对象方法是一种把面向对象的思想应用于软件开发过程中，指导开发活动的系统方法，简称OO方法，它是建立在对象概念（对象、类和继承）基础上的方法。它的主要概念有：

·对象：现实世界中某个具体的物理实体在计算机逻辑中的映射和体现。

·类：是一种抽象的数据类型，是同种对象的集合与抽象。属于类的某一个对象则被称为类的一个实例。

OO方法的特点是抽象、多态、封装、继承，它具有可重用性、可扩展性、可管理性等优点。

1.2 再工程

再工程是一个工程过程，它将逆向工程、重构和正向工程组合起来，将现存系统重新构造为新的形式,如图1所示。

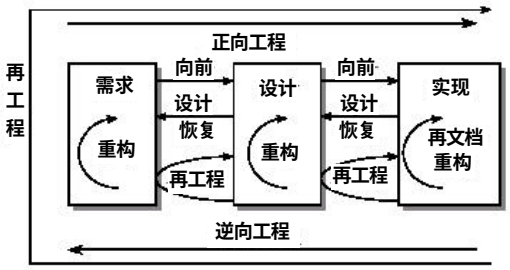

图1 再工程定义

1.3 Web Service

Web Service是一种可以接收从Internet或者Intranet上的其它系统中传递过来的请求的技术，是轻量级的独立的通信技术。这种技术允许网络上的所有系统进行交互。随着技术的发展，一个Web服务可以包含额外的指定功能并且可以在多个B2B应用中协作通信。

对于外部的使用者，Web服务是一种部署在Web上的对象/组件，它具有完好的封装性、松散耦合、使用协约的规范性、使用标准协议规范和高度可集成能力等优点。

无论是国内还是国外的研究，都是基于自己的Application Server平台，提出了迁移的工具或者方法，缺乏一个通用的迁移过程，本文提出了一个通用的再工程过程，用于分析传统的面向对象方法设计的系统，将它们部署到Web Service上。

基金项目：上海构件库及其应用研究基金资助项目(025115014)
作者介绍：杨 滨（1979—），男，硕士，主研方向为软件工程、软件构件技术；赵文耘，教授
收稿日期：2003-06-18 E-mail：Yang9724@etang.com

2 迁移过程概述

我们提出了一个通用的过程，可以把面向对象的应用程序组织成具有内聚功能的软件包，这些软件包可以通过Web服务访问。迁移过程如图2所示。

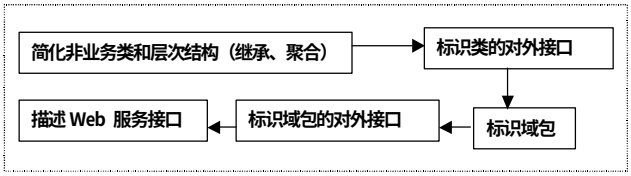

图2 迁移过程

2.1 简化非业务类和层次结构（继承、聚合）

为了标识面向业务的Web服务，首先需要对现有的面向对象的应用结构进行抽象和简化。对于一些持久的或者系统的类，这些对象存在并且重要，但是由于简单化的原因，而且由于我们将精力集中在业务方面，因此可以认为它们是理所当然应该存在的，并且可以将它们忽略掉。这些类中，持久的类包括硬编码的SQL、数据访问类（如Java数据对象（Java Data Object,JDOs））、一个服务（如EJB的容器管理持久性（container-managed persistence，CMP））或者一个或多个Web服务等；对于系统的类，可以把它们作为一个API、包装器类集合来实现。

为了标识服务者，常常可以简化继承和聚集层次结构。对于继承层次结构，如果一个子类没有添加其它新的对外接口，那么可以忽略它，并且通常可以把一个类层次结构看作单个类。对于聚集层次结构，可以忽略与聚集层次结构外部的其它类不相关联的任何局部类。通过分解聚集和继承层次结构，我们简化了传统的面向对象模型，在定义子系统时使其更易分析。例如图3中F类除了继承E类外，没有添加其它新的对外接口，所以相对于E类F类就可以被忽略掉；B、C、D类除了与A的聚集关系外，没有与其它类相关，成为了“局部类”，所以相对于类A，B、C、D也可以忽略掉。

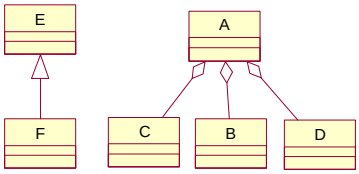

图3 继承和聚集的例子

2.2 标识类的对外接口

对外接口是其它对象可以请求的一个对象的任何服务或者行为。换句话说，它是一个直接响应来自其它类的消息的操作。为了使得抽取Web服务的方法更加简单，可以忽略所有不是类的对外接口的操作（也就是类内部的操作），因为它们对分布在不同包中的类之间的通信不起作用。

2.3 标识域包

一旦标识了类的对外接口，就可以开始标识它们提供的可能的域包和服务了。域包是一个互相协作以支持可以被认为是一个黑盒的对外接口的内聚集合的类集。基本思想是每个域包将提供一个或者更多的Web服务给其它域包、应用程序，甚至可能是外部系统，当然像高内聚、低耦合这样的标准可以被用来衡量标识的合理程度。从表面上看来，域包好象很简单，但是因为封装了许多类的行为，所以它们的内部通常相当复杂。

关键目标之一是把设计组织进几个包，这样可以减少它们之间的信息流量。包之间传递的任何信息(以Web服务调用的形式或者以来自那些调用的结果集的形式)都代表着可能的网络流量。因为希望最小化网络流量以改善应用程序的响应时间，所以域包的标示应有比较具体、完备的方案。

2.4 标识域包的对外接口

域包的对外接口是那些被域包以外的类访问的接口。域包接口组成了潜在的由包提供的Web服务。这一步的完成需要对业务熟悉，并且有经验的技术人员配合进行。为了减少Web服务的数量，需要把包内在业务上最相关的接口“拼”起来，组成一个接口。

2.5 描述Web服务接口

最后需要描述“拼”完后的接口。Web Service描述语言(WSDL)是一种基于XML的语言，用于描述Web Service及其函数、参数和返回值。因为是基于XML的，所以WSDL既是机器可阅读的，又是人可阅读的，这有一个很大的好处。目前已有很多开发工具既能根据Web Service生成WSDL文档，又能导入WSDL文档，生成调用相应Web Service的代码，这些工作能够自动地完成对接口的描述工作。当然为了简化Web服务的接口，返回结果和参数都应该只包含最少的必需的元素。

3 域包的抽取

在整个迁移过程中，域包的抽取是比较关键的一步。设计域包的原则是使大部分信息流在包内发生而不是在包之间发生。对域包抽取，分两步进行。

3.1 根据协作关系初步划分域包

为了确定一个类是否属于一个域包，需要分析它确定分布类型所涉及到的协作。服务器类接收消息但是不发送它们；客户机类发送消息但是不接收它们；客户机／服务器类既发送又接收消息。一旦标识了每个类的分布类型，就可以开始标识可能的域包了。可以遵循以下的规则：

(1) 服务器类属于一个域包并且常常会形成它们自己的域包，因为它们是应用程序里消息流的最后一站。

(2) 如果有一个域包，这个域包是唯一的一个其它类或者域包的服务器，可以决定合并这两个域包。

(3) 客户机类并不属于一个域包。客户机类仅仅生成消息但是不接收它们，然而域包的目的就是响应消息；所以，客户机类没有什么可以添加到包所提供的功能里去。

(4) 一个相关的启发就是客户机／服务器类属于一个域包，但是可以有几个它可能属于的域包。这就是需要考虑额外问题的地方，例如进出类的信息流以及添加新类如何影响每个组件的内聚度（事物的各部分在一块有意义的程度）。一个基本的设计规则就是，一个类只要存在并去完成一个域包的目标，那么它就应该是该域包的一部分。

3.2 根据内聚度划分域包

上面划分的域包是根据协作关系划分的，还比较粗，当交互涉及到大对象(作为参数传递或者作为返回值接收)时，这样的划分就不够了。为了细化域包的划分(对得到的域包进行可能的再次划分)，那些高度耦合的类要合在一块。

提出域包内聚度的标准如下：

假设域包包括N个类，类之间互相引用，如果每个类都互相引用时，引用数为N*(N-1)，称之为完全引用。在完全引用的情况下，域包内聚度为1。域包内聚度是域包内各类的实际引用关系占完全引用关系的百分比，也即域包内聚度=包内类实际引用数/完全引用数=包内类实际引用数/(N*(N-1)). 有了这个标准，就可以判定域包的聚合程度，显然域包内聚度越大，域包的聚合程度就越大，越满足“高内聚”的

标准。这个标准将引申来评价构件抽取的妥当程度。

制定标准a：对于某类A，A的引用中来自当前域包的比例。这个标准表达了类A对当前域包的专属程度。如果初始制定了一个临界比例，那么当专属程度大于这个临界比例时，可以认为当前域包包含类A。

对系统按照内聚度划分域包的步骤如下：

(1) 选定系统中可能的业务初始的类作为入口类（需要业务人员参与）。

(2) 建立两个空的集合：S和H，S表示抽取出的域包，H表示原有的系统E中除去S中的类后剩下的类。用户选取入口类，将入口类s加入S中，初始时，S={s}，H=E-{s}.

(3) 依照引用关系，逐个考察H中的被S中的类引用的类。如果满足标准a,那么把这个类加入S中，并从H中去掉这个类。

(4) 如果S在这一轮处理后没有变化，那么算法结束，S就是抽取出的子系统；否则转（2）。

通过算法，原有的系统被划分为多个部分，每个类的内聚度都是很大的，而耦合度就比较小。

下面举一个简单的例子,如图4。

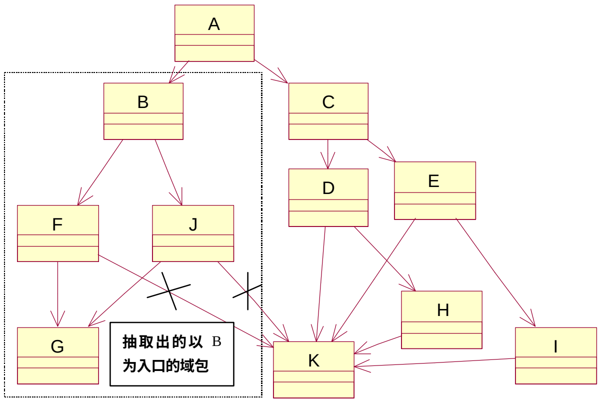

图4 域包的抽取

根据我们的抽取算法，抽取以B类为入口的域包。假定类属于域包的归属度要大于75%。

以B为入口，S={B},H={A,C,D,E,F,G,H,I,J},B引用了F，F没有其它类引用它，归属度为1，满足条件a，所以F加入S中；再考虑G，G被F和J引用，J还没有加入S中，所以归属度为1/2=50%，不满足75%的要求，所以暂时不能加入；接着考虑J，J可以加入。所以S={B,F,J},H={A,C,D,E,G,H,I}。

S变大，继续执行算法，这次可以加入G,G是满足条件的，再考虑K,K的归属度只有1/3,所以不满足条件。算法结束，S={B,F,G,J}，这就是我们抽取出的以B为入口的域包。

4 改进的方向与结语

如果希望减少域包与其它域包（类）之间的关联，可以在图4打“X”的地方切断引用的连线，方法是将引用到的类K的方法“复制”一份到F和J中，这样F和J就能够在没有K的情况下工作了，得到了更大的独立性。具体的方法有Ritsumeikan大学的Katsuhisa Maruyama提出的通过使用基于块的切片进行“方法”提取的自动化重构。对域包接口的定义，最好能在设计阶段就进行，虽然这样做有时候是比较困难的，但这样就可以把精力集中在包的内部，而不用考虑对包内类的改变有可能带来的对其它域包的影响了。

基于Internet的大规模电子商务的不断发展产生了建立这样一种分布式计算的需要，这种分布式的计算要能够跨平台、跨语言、跨协议地提供服务，Web Service满足了这样一种需要，它提供了商业用户互相发现、发布、引用对方服务的能力。大型企业迫切地希望将自己原先的系统（遗产工程）迁移到Web Service上来。本文提供了一种通用的再工程的过程，使得遗产工程能够顺利地迁移到Web Service上来，也使得用户在原有基础上扩展自己的业务成为可能。

参考文献

1 Wirfs-Brock R, Wilkerson B, Wiener L. Designing Object-oriented Software. Prentice Hall, 1990

2 Ambler S W. Building Object Applications That Work.Cambridge University Press, 1997

3 朱三元, 钱乐秋, 宿为民编. 软件工程技术概论. 北京: 科技出版社，2002-01

4 柴晓路. Web服务架构与开放互操作技术. 北京: 清华大学出版社，2002-06

5 Hendricks M, Galbraith B. Java Web 服务编程指南. 北京: 电子工业出版社, 2002-10

6 Maruyama K. Automated Method-extraction Refactoring by Using Block-based Slicing.Symposium on Software Reusability. New York: ACM Press, 2001

※※

（上接第76页）

及请求进行评估。策略判决点从策略管理点动态地获取策略集，并且向上下文处理器返回一个决策结果。上下文处理器将这个结果转发给WS代理，代理实施这个决策，或者让来自WS客户端的访问请求通过或者拒绝它。

4 结论和将来的工作

已经给出了一个将XACML访问控制实施于WS的系统模型。通过对PMI的语义描述，也解决了如何集成PMI的问题。模型最大的特点是对XML元数据技术的充分使用。这不仅仅体现在WS本身使用的SOAP、WSDL和UDDI技术上，也包括引入XACML作为访问控制的基础，以及对PMI的无缝整合上。

要真正实现Web Service的安全机制，不仅仅在于访问控制，也包括身份认证机制以及软件的安全实现方面。将来的工作将集中在提供一个完整的适用WS的安全体系上。

参考文献

1 Chadwick D W. An X.509 Role-based Privilege Management Infrastructure. Business BriefingGlobal Infosecurity, http:// www. permis.org /, 2002

2 张 纲, 李晓林, 游赣梅等. 基于角色的信息网络访问控制的研究. 计算机研究与发展，2002,39(8)

3 OASIS. Extensible Access Control Markup Language(XACML) Version 1.0. OASIS Standard, http://www.oasis-open.org/xacml/,2002-02-18

4 Schimdt T. Security Architecture for an Extended Enterprise Using Web Services. XML Europe, http://www.xmleurope.com/, 2003

5 (美) 莫里森. XML揭密[M]. 北京: 清华大学出版社，2001-06