

Web 服务应用接口系统的开发研究

杨 东

(中国移动通信集团贵州有限公司 贵阳 550081)

摘 要 随着 Web 服务技术的不断成熟,其在异构系统集成应用中的优势逐渐显现,基于 Web 服务的应用集成技术成为企业信息系统研究的热点。使用方便、实用的 Web 服务应用接口,也成了互联网通信技术的发展趋势。文中对这一关键技术进行了研究,取得了初步理论与软件成果。另外,文中深入研究了 Web 服务的基本原理与实现技术、基于 Web 服务的应用系统集成技术、Web 服务应用接口系统的开发原理与技术,提出了一套 Web 服务开发方法、Web 服务应用接口系统的模型框架。在理论研究的基础上,基于 Microsoft.NET2003 环境,文中还实现了基于模式引导的应用接口包装器的生成与维护软件,其可为不同的应用系统提供 Web 服务的应用接口,方便客户端调用,还可以简化 Web 服务应用的编程过程,提高了应用系统开发效率。

关键词: Web 服务;应用接口;系统研发

中图法分类号 TP393.0

Development and Research of Web Service Application Interface System

YANG Dong

(China Mobile Group Guizhou Co.,Ltd.,Guiyang 550081,China)

Abstract With the continuous maturity of Web service technology, its advantages in the integration and application of heterogeneous systems have gradually emerged, and application integration technology based on Web services has become a hot spot in enterprise information system research. The easy-to-use and practical Web service application interface has also become the development trend of Internet communication technology. This key technology is studied in this paper, and preliminary theoretical and software results are obtained. In addition, this paper deeply studies the basic principle and implementation technology of Web services, application system integration technology based on Web services, and development principle and technology of Web service application interface system, and proposes a set of Web service development methods and model framework of Web service application interface system. On the basis of theoretical research, based on the Microsoft.NET2003 environment, this paper also realizes the generation and maintenance software of the pattern-guided application interface wrapper, which can provide the application interface of Web services for different application systems, which is convenient for client side calls. It can also simplify the programming process of Web service applications and improve the efficiency of application system development.

Keywords Web service, Application interface, System research and development

0 引言

随着企业信息化进程的不断推进,企业“信息孤岛”问题愈发严重。在这种情况下,企业应用集成(EAI)技术应运而生。EAI是将基于不同平台、用不同技术建立的异构应用系统集成起来的方法和技术^[1]。EAI技术从早期的点对点集成方式,到基于DCOM,CORBA等分布式计算技术的集成方式,一直发展为如今基于Web服务(Web Services)的企业应用集成技术。为了实现基于Web服务的EAI,必须解决以下关键问题:1)新型应用系统的开发模式;2)新系统与遗留系统的集成;3)现有分布式中间件的互操作性差、

用户应用集成困难等。针对这些问题,本文提出并实现了基于Web服务的应用集成中间件的技术解决方案,通过Web服务应用接口封装的生成系统来自动产生与目标应用相关的Web服务框架代码,并自动部署应用系统。

1 Web服务的原理

当前,Web服务不存在统一定义,其是一种不属于Web之上的组件或对象,拥有和对象技术相同的优势。建立在XML之上的Web规范技术,开放性更强。换言之,Web服务的实现需要浏览器、服务器、代理、缓存、网关等应用程序的支持,这些应用被公布在网络上,以便其他程序的调用,

收稿时间:2023-07-11

作者简介:杨东(1977—),硕士,中级工程师,研究方向为接口安全、数据安全。

也可经标准 Internet 协议访问。Web 服务不会对编程环境、操作系统或特定软件产生依赖,其本质是一种面对组件及对象的逻辑延伸,是一种新型分布式计算模型。Web 服务由标准与技术构成,包括 UDDI(发现 Web 服务)、SOAP(调用 Web 服务)和 XML(基石)^[2]。

其主要目标是为不同的异构平台创建与平台无关且通用的中间层,平台的不同层级借助该中间层能实现集成与连接,使公布式系统拥有广泛性及开放性。

2 Web 服务的开发与应用

Web 的开发周期包含构建、部署、运行及管理 4 个阶段,其服务流程与其他类型的应用程序创建过程十分相似。换言之,其是一种包含应用程序的设计、代码编写、部署、应用的周期。

2.1 跨防火墙通信

应用大量的分布式应用程序,会让服务器和客户之间的通信出现多种问题,这是因为两者之间存在代理服务器或防火墙。DCOM 等常规分布式计算技术的应用也会出现以上问题,解决该问题的重点在应用程序中间层。然而,多次编写中间层会导致应用程序开发受阻,无法对其进行有效的维护。借助 Web 服务转变中间层组件,可以使客户端经 SOAP 来直接调用 Web 服务,降低开发难度,缩短开发周期。

2.2 应用程序集成

企业中运行着多种不同的平台及语言软件,这些平台、软件之间需要实现数据交换及互通,若想集成它们,则会面临较大经济及人力压力。Web 服务可重新封装应用程序的数据与功能,直接无视开发语言的异构性或平台冲突等问题。

2.3 电子商务

大部分企业的电子商务平台使用的系统开发语言、部署平台、通信协议、对外交换数据的格式等都存在显著的差异。Web 服务可以集成和连接这些应用,实现动态电子商务。

3 Web 服务应用接口系统的原理及技术

3.1 使用 Web 服务的必要性

随着供应链管理(SCM)或企业资源规划(ERP)的出现,EAI 集成方案也能实现资源共享,但常规的 EAI 十分复杂,缺乏适应性及灵活性。Web 服务能借助公布式计算平台,在 Intranet 中使用标准的信息格式及 XML 协议,充分展现出商业应用服务。标准协议的使用能让发布者、Web 服务平台等呈现独立的状态,满足集成方案所需的条件,制定建设 EAI 的新方案。

语言及平台的独立性是 Web 服务的主要特征,企业与企业、企业内部等的资源共享的主要途径为 EAI。使用 EAI 集成方案,能实现 Web 服务应用等的系统集成,有效促进分布式系统(如网络化设计制造等)的合理应用。

3.2 基于 Web 服务的系统框架

各种 Web 服务适配器、Web 服务应用集成服务引擎(含事物管理引擎、SOAP 路由器及工作流引擎)、NET 应用程序服务器、Web 服务技术标准等,组成了应用集成系统框架。在该框架中,所有应用均能让相应的接口连接并被调用,具有快速、简单、灵活等优势。

4 Web 服务应用接口系统的开发原理及技术

4.1 系统开发原理

大部分开发工作者将分析对象作为首选,随即编写代码。而在新时代的背景下,分布式应用程序也涉及 Web 服务,并被广泛使用。其主要开发原理为契约先行,即 WSDL 文档的优先级较高。系统开发主要涉及数据建模、消息建模、操作及接口建模、生成代码框架、迭代契约及生成代码等 5 个步骤。通过契约先行的开发模式,能显著提高应用系统的开发效率,有效解决服务器及客户端的异构情况^[3]。

4.2 系统体系结构

Web 服务应用接口维护组件、部署组件和产生组件等构成了 Web 服务的应用接口系统。其中,Web 服务应用接口的维护组件以维护管理接口为主,部署组件以 UDDI 注册服务器或服务器上部署的服务器端 Web 服务接口为主,产生组件则可以封装相应的系统,按照实际要求形成服务器端的 Web 服务接口框架代码。

5 接口安全问题

开放 API 接口若未经过安全处理,则会影响对信息的收集,出现信息被篡改和泄露的现象。例如,一家公司开发了一个开放 API,而没有对其进行控制,黑客可以很容易地拦截该公司的客户数据。在黑客登录后,可以使用帐户名、密码等信息注册为客户,这使黑客可以轻松删除他们的客户信用卡,表单数据易于操作,若该部门开发的开放 API 不可篡改,则会导致业务数据受到破坏。当公司开发开放的 API 时,如果没有安全措施,黑客程序即爬虫程序可以使用开放 API 来查询客户数据的接口。此时公司无法有效地处理开放 API 接口,黑客无需调用客户信息接口来进行安全身份验证,其可以使用客户信息来查询 API,收集客户的数据。因此,软件开发的 API 接口安全至关重要。

6 安全问题的解决策略

非对称加密是一种常用的加密算法,与对称加密算法不

同,其只有一个密钥文件,但有两个密钥,即公钥和私钥。公钥是公共的,私钥是秘密,如果黑客只有公钥而没有私钥,不会对报文产生影响;当满足条件时,打开的 API 接口是使用 HTTPS 协议传输数据的接口,再使用 RSA 加密技术,可以更安全地传输 HTTPS。信息聚合算法的特征在于其不可逆性,字符串可以创建一个摘要,在大多数情况下,该算法用于汇总信息,也可存储用户的登录密码以及匹配信息。目前,信息摘要算法并不能真正解密。需要碰撞的数据量非常大,它是一个彩虹表,如许多汇总信息的算法。字符串与字符串匹配,在解密摘要算法时,其中会有大量的数据,包含加密的摘要算法,因此检索时间会相对较长,但会使信息聚合算法更加安全。应用程序接口安全性与用户的安全意识有着直接的关系,因此需加强对无意识泄露或滥用信息的人员的培训,并定期检查安全设备,提高所有员工的安全意识。

7 结语

本文通过对 Web 服务关键技术(如 UDDI、SOAP 或 XML)及基本原理的介绍,掌握了基于 Web 服务应用系统集成解决方式、Web 服务应用接口系统的开发原理及相关技术。另外,本文分析了 Web 服务应用接口的系统框架,具体包含 Web 服务接口管理工具、Web 服务接口发布/部

署向导工具及 WSDLEditor—WSDL 文档编辑器等。开发者及设计人员若能合理应用这些工具,在短时间内开发出 Web 服务接口,则能在降低开发成本的基础上,显著提高编程效率。同时,经过对 Web 应用开发方法的分析,明确了其主要的突出优势。客户端开发者和服务器端开发者,均能借助开发平台完成相关的开发工作,这能使用户在短时间内开发出相应的 Web 服务接口适配器。除此以外,本文还解决了服务器及客户端异构问题。目前,本文开发出的 Web 服务应用接口系统仅可产出服务器端的 Web 服务接口框架代码,还无法开发客户端的代理内容,同时,其在解析 WSDL 文档、映射代码或 WSDL 文档时,缺乏精确性和全面性,还需不断完善。

参考文献

[1] 肖利斌,郑向敏,黄文胜.国际旅游效率研究概况、热点及趋势——基于 Web of Science 核心合集的知识图谱分析[J].西南民族大学学报(人文社会科学版),2022,43(1):36-45.

[2] 田晨林,陈正宇.嵌入式系统 Web 服务器的移植与 CGI 的应用[J].金陵科技学院学报,2019,35(2):40-43.

[3] 傅荣鑫,李长云,李俊峰.Web 系统服务器集群部署策略研究[J].电脑知识与技术,2021,17(20):11-13,19.

[J].软件学报,2021,32(4):1082-1115.

[2] 邵浩,刘一烽.预训练语言模型[M].北京:中国工信出版集团,2021:10-22.

[3] 王琦,杨毅远,江季.Easy RL:强化学习教程[M].北京:人民邮电出版社,2022:36-52.

[4] 余同瑞,金丹,韩晓臻,等.自然语言处理预训练模型的研究综述[J].计算机工程与应用,2020,56(23):12-22.

[5] 闻东海,修瑞杰,孙玉婷.信息不完全状态下智能车辆突发情况感知框架研究[J].汽车博览,2022(9):87-89.

(上接第 184 页)

动指利用多模态模型来构建数字图片处理网络,再结合文本知识得到决策辅助信息。知识驱动以 ChatGPT 为军事信息处理模型,为后续推理和预测提供了支撑。本文还提出了基于 ChatGPT 和多模态模型的框架,为智能军事决策的发展提供了参考。

参考文献

[1] 王乃钰,叶育鑫,刘露.基于深度学习的语言模型研究进展