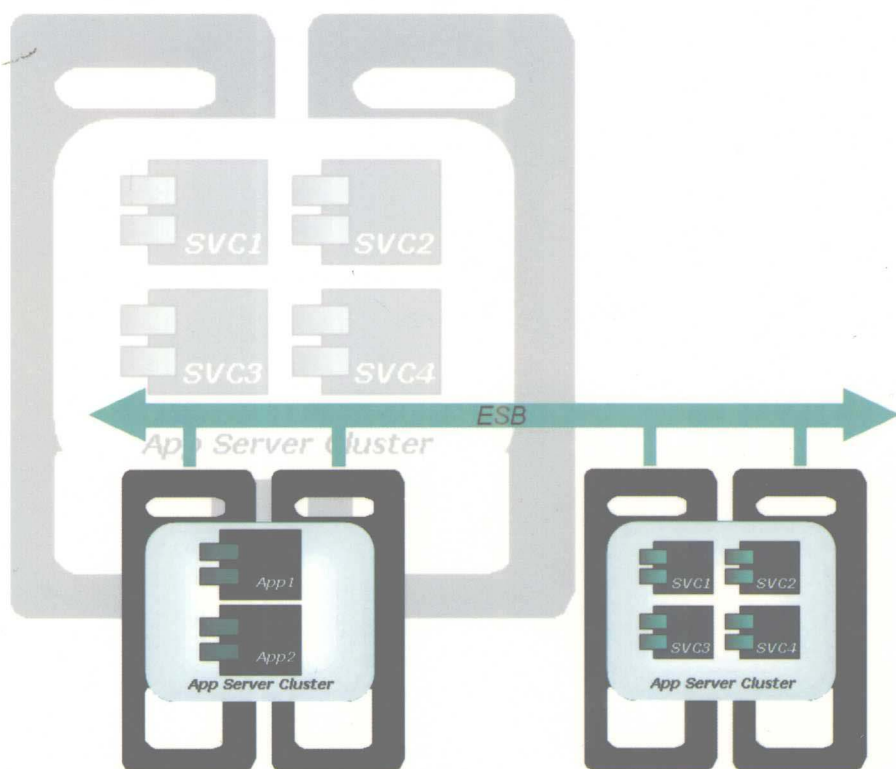


SERVICE ORIENTED ARCHITECTURE

电力企业 信息化SOA实践

费圣英 主编



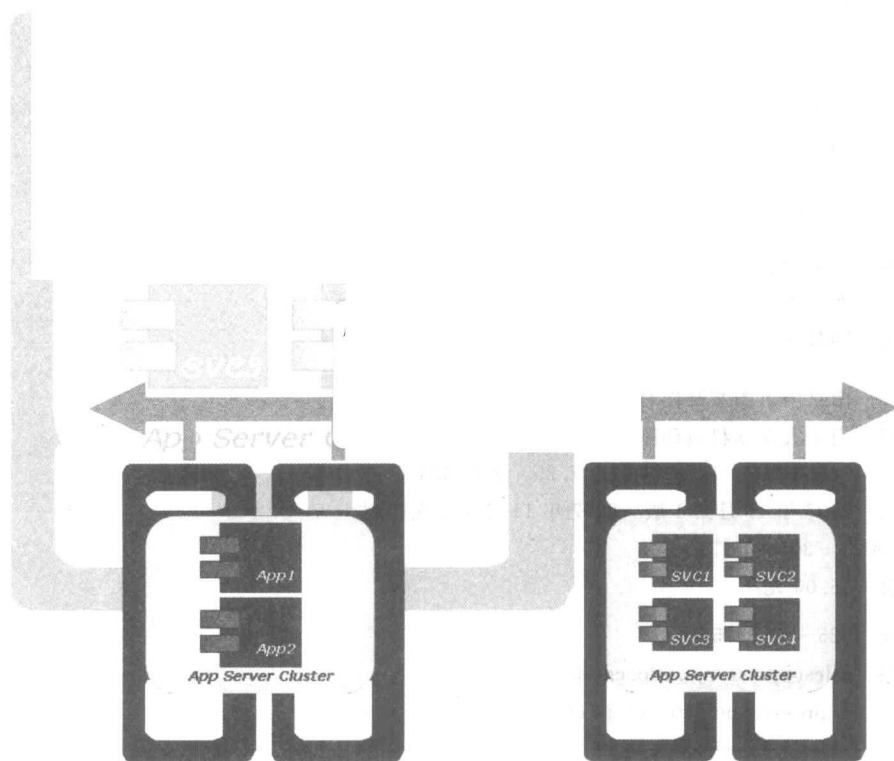
南京大学出版社

0307473

SERVICE ORIENTED ARCHITECTURE

电力企业 信息化SOA实践

费圣英 主编



南京大学出版社

内容简介

面向服务的架构(Service Oriented Architecture, SOA)是当前软件体系架构的最新发展方向。本书从 SOA 的起源开始,结合江苏电力的 SOA 实践,全面而系统地介绍了江苏电力基于 SOA 企业信息集成平台的建设全过程。

全书共 12 章,分三大部分。第一部分(第 1 章)介绍 SOA 的历史沿革、需求背景、核心技术以及当前应用现状;第二部分(第 2~4 章)介绍了江苏电力的 IT 应用现状,分析了电网企业信息化的特点,并详细阐述了江苏电力企业信息化集成平台建设的 SOA 实施框架;第三部分(第 5~12 章)详细介绍了平台建设的具体过程,包括:业务流程分析和设计、服务目录与服务总线的设计和部署、业务流程管理、企业门户展现、安全架构、集成平台测试和具体的运行管理平台。

本书适合企业信息主管、信息部门员工、软件工程师、计算机软件专业研究人员、高校计算机及相关专业的教师 and 研究生阅读。

图书在版编目(CIP)数据

电力企业信息化 SOA 实践/费圣英主编. —南京:南京大学出版社,2007.11

ISBN 978-7-305-05131-9

I. 电… II. 费… III. 信息技术—应用—电力工业—工业企业管理 IV. F407.61-39

中国版本图书馆 CIP 数据核字(2007)第 178221 号

出版者 南京大学出版社
社 址 南京市汉口路 22 号 邮 编 210093
网 址 <http://press.nju.edu.cn>
出版人 左 健
书 名 电力企业信息化 SOA 实践
主 编 费圣英
责任编辑 顾其兵 编辑热线 025-83593642
照 排 南京南琳图文制作有限公司
印 刷 南京大众新科技印刷有限公司
开 本 787×1092 1/16 印张 17.25 字数 434 千
版 次 2007 年 11 月第 1 版 2007 年 11 月第 1 次印刷
ISBN 978-7-305-05131-9
定 价 58.00 元
发行热线 025-83594756
电子邮箱 sales@press.nju.edu.cn(销售部)
nupress1@public1.ptt.js.cn

* 版权所有,侵权必究

* 凡购买南大版图书,如有印装质量问题,请与所购图书销售部门联系调换

《电力企业信息化 SOA 实践》

编 委 会

主 编：费圣英

编写组成员：马苏龙 鲁庭瑞 王海林 夏 俊
何大春 季 强 龚 冰 赵明奇
张爱民 许志龙 刘克智 崔恒志
沈红宇 魏 蓉 司为国 方 泉
张德进 许海清 庄哲寅 丁一新
张明明 赵俊峰 曹渝昆

前 言

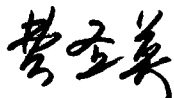
“十一五”开局之始，国家电网公司提出了在全系统实施“SG186 工程”的信息化建设规划，该工程将实现四大目标：一是建成“纵向贯通、横向集成”的一体化企业级信息集成平台，实现公司上下信息畅通和数据共享；二是建成适应公司管理需求的八大业务应用，提高公司各项业务的管理能力；三是建立健全规范有效的六个信息化保障体系，推动信息化健康、快速、可持续发展；四是力争到“十一五”末，国网公司的信息化水平达到国内领先、国际先进，初步建成数字化电网、信息化企业。该规划的出台对国家电网公司及其下属省网公司乃至整个电网系统的信息化建设都将会产生深远的影响。

江苏电力按照国家电网公司信息化建设思路和 SG186 工程“纵向贯通、横向集成”的指导思想，结合公司信息化工作的具体情况，确定了“十一五”期间公司将按照“硬件集中、软件集成”的总体工作思路开展信息化工作，有效整合信息资源，加强集约化管理，努力达到国家电网公司信息化建设 SG186 工程的要求。江苏电力经过多年的信息化建设，已初步建立起覆盖“SG186 工程”主要部分的信息化应用，但与国内外信息化领先企业相比仍然存在差距，总体水平仍有待提高。在企业级信息系统建设、信息资源有效利用、信息服务能力和支撑能力等方面亟需进一步加强。从业务系统建设方面来看，江苏电力在业务应用系统建设方面，大部分应用系统建设按部门职责条块分割，自成体系，较少考虑业务的跨部门特性，缺乏企业层面的信息化整体规划；下属各单位自行建设业务应用，未能实现企业级的统一管理；业务系统技术架构、数据架构、开发模式等缺乏规范性指导，不同业务系统间互相访问极为困难，导致“信息孤岛”现象的发生；相同的数据保存在不同应用系统中，缺乏必要的支撑手段保证数据一致性和准确性；信息系统集成度较低，未能实现企业信息系统的整合以及数据集成，缺乏有效的技术手段对企业的数据库资源进行深层次挖掘，无法为企业管理提供有效的辅助决策支撑。

针对江苏电力信息化存在的问题，江苏电力决定采用 SOA (Service-Oriented Architecture, 面向服务的架构) 架构对省公司的信息系统进行 SOA 改造，并建立基于 SOA 的企业信息集成平台，以达到信息技术支撑能力与业

务应用需求实现的同步,帮助企业根据业务需求,合理有效地动态配置各种资源目的。SOA 是一种架构模型,它可以根据需求通过网络对松散耦合的粗粒度应用组件进行分布式部署、组合和使用。SOA 可以为企业解决“信息孤岛”问题,并能够提供按需应变的服务。SOA 通过将原有的应用和资源转变为可共享的标准服务,企业只需根据自己的策略来定制流程,把应用作为服务“拿来就用”,而无需考虑底层的集成;从而实现信息技术和企业业务之间的同步。SOA 专注于考虑如何复用已有的服务。它鼓励使用可替代的技术和消息机制,通过把服务联系到一起来满足应用;这种基于消息的应用允许企业通过调整现有的服务模式而不是被迫进行大规模新应用代码的开发来满足业务需求。

本书是对江苏电力基于 SOA 的企业信息集成平台设计和部署方案,以及实施经验的总结,主要分为两大部分,前 3 章对 SOA 的基本概念、江苏电力的信息化现状和电网企业信息化特点进行了详细分析,后 9 章对整个项目的实施过程中的各个关键技术步骤、集成平台的核心组成部件和集成平台本身进行了详尽阐述。相信该书的出版将会对大型国有企业的 SOA 项目设计、部署和实施有所裨益和帮助,对电网企业的信息化建设起到积极推动作用。



2007 年 8 月

目 录

第一章 SOA 概述	1
1.1 软件体系结构	1
1.1.1 软件体系结构定义	1
1.1.2 软件体系结构的发展	2
1.2 SOA 的发展起源	3
1.2.1 产生阶段	4
1.2.2 推动阶段	4
1.2.3 成长阶段	5
1.2.4 SOA 定义	5
1.3 SOA 相关技术及核心部件	6
1.3.1 Web 服务	6
1.3.2 ESB	8
1.3.3 BPM	9
1.3.4 Portal	9
1.4 SOA 的特点	10
1.4.1 SOA 架构的技术特征和应用优势	10
1.4.2 基于 SOA 的信息系统开发方式特点	12
1.5 SOA 的应用现状	13
1.5.1 国际应用现状	13
1.5.2 国内应用现状	14
1.5.3 国际上主流 SOA 技术产品	14
第二章 江苏电力 IT 应用现状	16
2.1 软件应用现状	16
2.2 硬件应用现状	17
2.3 存在的问题	19
2.4 SOA 信息集成实施目标	20
第三章 电网企业信息化特点分析	23
3.1 电网企业的业务特点	23
3.2 电网企业信息化存在的问题	24
3.3 电网企业信息化建设面临的任务	25
第四章 江苏电力 SOA 实施框架	28
4.1 SOA 目标架构	28

4.1.1	目标架构视图	28
4.1.2	逻辑视图	32
4.1.3	功能视图	33
4.1.4	部署视图	37
4.1.5	非功能需求	44
4.2	SOA 信息集成平台概要设计	47
4.2.1	SOA 信息集成总体框架	47
4.2.2	集成平台架构	49
4.2.3	平台功能需求	52
4.2.4	平台非功能需求	54
4.2.5	集成平台逻辑架构	55
4.3	SOA 实施规划和组织保障	61
4.3.1	实施规划	61
4.3.2	组织保障	65
第五章	业务流程分析和设计	66
5.1	业务流程分析	66
5.1.1	分析步骤	66
5.1.2	跨部门需求细化	67
5.1.3	数据视图	70
5.1.4	流程视图	72
5.1.5	服务视图	76
5.2	业务流程设计	78
5.2.1	设计步骤	78
5.2.2	信息格式、配置文件和控件	78
5.2.3	流程定义	80
5.2.4	部署流程	81
5.2.5	流程详细设计模板	81
5.2.6	流程部署模板	84
5.2.7	业务流程设计举例	84
5.2.8	业务流程部署示例	93
5.3	流程之间的相互关系	93
第六章	服务目录与总线设计	97
6.1	服务目录	97
6.1.1	SOA 架构与服务目录	98
6.1.2	服务目录的设计要求	99
6.1.3	技术实现方式	100
6.1.4	服务目录结构视图	101
6.1.5	服务目录功能模块	102
6.1.6	信息存储模型	106

6.2	服务总线	110
6.2.1	服务总线技术实现	110
6.2.2	Package 概述	111
6.2.3	服务总线行为列表	113
6.3	服务请求和调用	114
6.3.1	服务调用原理	115
6.3.2	服务调用步骤	116
6.3.3	服务调用技术分类	116
6.3.4	服务接口规范	119
6.3.5	总线接口规范	122
6.3.6	接口访问安全	124
6.3.7	大数据量传输支持	126
6.4	服务的封装方法	127
第七章	服务目录与总线部署	130
7.1	服务注册	130
7.1.1	服务注册需要收集的信息	130
7.1.2	服务注册步骤	132
7.2	服务目录部署	135
7.2.1	服务目录部署方案	135
7.2.2	服务目录信息的缓存	137
7.2.3	服务目录信息改动通知	139
7.2.4	其他服务目录同步方案	140
7.2.5	服务目录部署物理需求	141
7.3	服务总线部署	142
7.3.1	服务总线架构	142
7.3.2	服务总线路由规则设计	143
7.3.3	服务总线物理部署需求	145
第八章	业务流程管理	146
8.1	业务流程管理概述	146
8.1.1	业务流程	146
8.1.2	业务流程管理	147
8.1.3	BPM 与 工作流	148
8.1.4	BPM 与 EAI	148
8.1.5	BPM 与 SOA	149
8.2	设计原则与决策	151
8.3	需求分析	153
8.3.1	系统使用者	153
8.3.2	系统边界	154
8.3.3	功能描述	156

8.4	架构设计	156
8.4.1	总体架构	156
8.4.2	逻辑架构	157
8.5	功能模块设计	159
8.5.1	功能模块结构图	159
8.5.2	流程映射 MOD	160
8.5.3	流程配置 CFG	162
8.5.4	流程跟踪写入 TRK	163
8.5.5	流程跟踪查询 MON	163
8.5.6	流程控制操作 CTL	166
8.5.7	流程催办 RMI	166
8.5.8	流程统计查询 STAT	168
8.6	非功能设计	168
第九章	企业门户展现	170
9.1	门户概述	170
9.1.1	企业门户定义	170
9.1.2	SOA 架构下的企业门户	171
9.1.3	门户分类	172
9.1.4	门户组件 Portlets	174
9.1.5	门户功能需求	174
9.1.6	门户非功能需求	176
9.2	部署构架	176
9.2.1	门户部署架构	176
9.2.2	集中式部署	179
9.2.3	集中加分布式部署	180
9.2.4	方案比较	181
9.3	门户实现方案	182
9.3.1	单点登陆	182
9.3.2	统一用户管理	184
9.3.3	统一用户认证	185
9.3.4	个性化	185
9.3.5	应用系统集成	187
第十章	集成平台测试	190
10.1	集成平台测试概述	190
10.2	集成平台测试计划	190
10.2.1	测试范围和提交文件	190
10.2.2	测试方案	191
10.2.3	测试问题及修改流程	192
10.3	测试用例	193

10.3.1	基础信息注册	193
10.3.2	服务目录管理	198
10.3.3	服务调用	202
第十一章	安全架构	207
11.1	SOA 架构安全性概述	207
11.1.1	SOA 架构的安全性挑战	207
11.1.2	SOA 安全性实施策略	208
11.2	SOA 的安全基础架构	209
11.2.1	认证	210
11.2.2	访问控制	212
11.2.3	加密解密	213
11.2.4	授权	214
11.3	江苏电力信息安全总体框架	215
11.3.1	基于 PKI 的信任服务体系	215
11.3.2	综合防护体系	216
11.3.3	集中监控体系	218
11.3.4	应用软件安全体系	219
11.4	服务调用安全控制	219
11.5	SOA 集成平台权限管理机制	220
11.5.1	权限定义	221
11.5.2	权限模型	222
11.5.3	角色	222
11.6	SOA 集成平台用户及权限管理实现	223
11.6.1	用户组管理	223
11.6.2	用户管理	224
11.6.3	角色管理	224
第十二章	运行管理平台	225
12.1	管理资源	225
12.1.1	需要管理的资源	225
12.1.2	服务总线	228
12.1.3	流程引擎	228
12.1.4	平台管理控制台	229
12.1.5	Weblogic Server 应用服务器	229
12.1.6	Weblogic Integration	229
12.1.7	Weblogic Portal	230
12.1.8	服务总线、服务目录和统计数据库	230
12.1.9	DB2 数据库服务器	230
12.1.10	存放在数据库中的数据	230
12.1.11	服务器硬件	231

12.2	管理机制	231
12.2.1	组件监控机制	231
12.2.2	服务统计机制	231
12.2.3	服务目录管理机制	233
12.2.4	服务总线管理机制	236
12.2.5	流程引擎管理机制	237
12.2.6	系统重启及恢复机制	242
12.2.7	备份与恢复机制	243
12.3	管理平台	243
12.3.1	监控统计	243
12.3.2	服务总线管理	248
12.3.3	服务目录管理	249
12.3.4	流程引擎管理	257
12.3.5	平台参数设置	257
参考文献		259

第一章 SOA 概述

随着大规模企业级应用的不断发展,信息系统的复杂度急剧提高,为了满足信息系统对异构性、互操作性、数据共享以及信息集成的要求,一种新型的软件体系架构应运而生——面向服务的架构(SOA, Service-Oriented Architecture),它是在面向对象技术和面向组件技术之上的更高级别抽象。本章从软件体系结构的演变历史入手,分析了面向服务的软件体系结构出现的必然性,然后通过对 SOA 架构发展由来的描述和对 SOA 基础概念的介绍,更深入地剖析了 SOA 的相关技术。最后,由对 SOA 的特点、应用场景和国内外 SOA 应用现状的分析,进一步阐述了 SOA 的发展趋势、应用潜力和可能的应用领域。

1.1 软件体系结构

1.1.1 软件体系结构定义

自 1968 年软件工程概念提出以来,软件工程界已经提出了一系列的理论、方法、语言和工具,解决了软件开发过程中的若干问题。但是,软件固有的复杂性、易变性和不可见性,使得软件开发依然存在开发周期长、代价高和质量低的问题。为了提高软件需求分析和软件设计的质量,各种需求分析工程技术和各种软件建模技术应运而生,但是在需求分析与软件设计之间仍存在一条很难逾越的鸿沟,缺乏有效的技术将需求转换为相应的设计。从机器语言、汇编语言、面向过程的程序设计语言、面向对象程序设计语言、形式化(半形式化)规格说明语言(如体系结构描述语言)发展过程中,可以发现:计算机语言越来越适合于开发人员的思维活动模型,代码复用的级别也在不断地提升。软件体系结构技术的研究,使软件复用从代码复用发展到设计复用和过程复用。

如同土木工程进入到现代建筑学一样,软件也从传统的软件工程进入到现代面向对象的软件工程,研究整个软件系统的体系结构,寻求构建最快、成本最低、质量最好的构造过程。结构化程序设计时代是以砖、瓦、灰、沙、石、预制梁、柱、屋面板盖平房和小楼,而面向对象时代是以整面墙、整间房、一层楼梯的预制件盖高楼大厦。预制构件怎样搭配才合理?体系结构怎样构造?重要构件有了更改后,如何保证整栋高楼不倒?每种应用领域需要什么构件(医院、工厂、旅馆)?哪些实用、美观、强度、造价合理的构件骨架建造出来的建筑(即体系结构)更能满足用户的需求?由于概念尚不统一、描述规范也无法达成一致认识,软件体系结构仍处于一个研究发展的阶段,迄今为止还没有一个为大家所公认的定义。下面给出了一个相对比较通用的软件体系结构定义。

IEEE 于 1995 年成立了体系结构工作组(AWG),综合了体系结构描述研究成果,并参考业界的体系结构描述的实践,起草了体系结构描述框架标准即 IEEE P1471。IEEE 1471

对软件体系结构定义如下：系统是完成某种特定功能的构件集合。软件体系结构是一个系统的基本组织，它体现在组件、组件之间的相互关系，以及组件与环境的关系中，还包括指导系统设计和进化的原则。

在大规模系统或具有类似需求和结构的软件产品开发中，必须从一个较高的层次来考虑组成系统的组件、组件之间的交互，以及由组件与组件交互形成的拓扑结构，这些要素应该满足一定的限制，遵循一定的设计规则，能够在一定的环境下进行演化。而且，软件体系结构应能反映系统开发中具有重要影响的设计决策，便于各种人员的交流，反映多种关注，据此开发的系统能实现其既定的功能，也能够满足其性能需求。

1.1.2 软件体系结构的发展

软件系统的规模在迅速增大的同时，软件开发方法也经历了一系列的变革。在此过程中，软件体系结构也由最初模糊的概念发展到一个渐趋成熟的技术。20 世纪 70 年代以前，尤其是在以 ALGOL 68 为代表的高级语言出现以前，软件开发基本上都是汇编程序设计。此阶段系统规模较小，很少明确考虑系统结构，一般不存在系统建模工作。70 年代中后期，由于结构化开发方法的出现与广泛应用，软件开发中出现了概要设计与详细设计，而且主要任务是数据流设计与控制流设计。此阶段软件结构已作为一个明确的概念出现在系统的开发过程中。

20 世纪 80 年代初到 90 年代中期，是面向对象开发方法兴起与成熟的阶段。由于对象是数据与基于数据之上操作的封装，因而在面向对象开发方法下，数据流设计与控制流设计则统一为对象建模，同时，面向对象方法还提出了一些其他的结构视图。如在对象建模 OTM(Object Modeling Technique)方法中提出了功能视图、对象视图与动态视图(包括状态图和事件追踪图)；BOOCH 首次提出了类视图、对象视图、状态迁移图、交互作用图、模块图、进程图；1997 年出现的统一建模语言 UML 则从功能模型(用例视图)、静态模型(包括类图、对象图、构件图、包图)、动态模型(协作图、顺序图、状态图和活动图)、配置模型(配置图)描述应用系统的结构。

90 年代以后则是基于构件的软件开发阶段，该阶段以过程为中心，强调软件开发采用构件化技术和体系结构技术，要求开发出的软件具备很强的自适应性、互操作性、可扩展性和可重用性。此阶段中，软件体系结构已经作为一个明确的文档和中间产品存在于软件开发过程中，同时，软件体系结构作为一门学科逐渐得到人们的重视，并成为软件工程领域的研究热点，因而文献【3】认为，“未来的年代将是研究软件体系结构的时代！”纵观软件体系结构技术发展过程，从最初的“无结构”设计到现行的基于体系结构软件开发，可以认为经历了 4 个阶段：(1)“无体系结构”设计阶段：以汇编语言进行小规模应用程序开发为特征；(2)萌芽阶段：出现了程序结构设计主题，以控制流图和数据流图构成软件结构为特征；(3)初期阶段：出现了从不同侧面描述系统的结构模型，以 UML 为典型代表；(4)高级阶段：以描述系统的高层抽象结构为中心，不关心具体的建模细节，划分了体系结构模型与传统的软件结构的界限，该阶段以 Kruchten 提出的“4+1”模型为标志。

软件体系结构的发展历史就是人们不断追求更高的抽象、封装和模块化的过程。图 1-1 展现了软件体系结构的发展里程碑。80 年代中期出现了 Client/Server 分布式计算结构，应用程序的处理在客户端(PC 机)和服务器之间分担；请求通常由关系型数据库处理，客户

端在接收到被处理的数据后实现显示和业务逻辑;系统支持模块化开发,Client/Server 结构因为其灵活性得到了极其广泛的应用。但对于大型软件系统而言,这种结构在系统的部署和扩展性方面还是存在着不足。Internet 的发展给传统应用软件的开发带来了深刻的影响。随着越来越多的商业系统被搬上 Internet,一种新的、更具生命力的体系结构被广泛采用,这就是“三层/多层结构”。

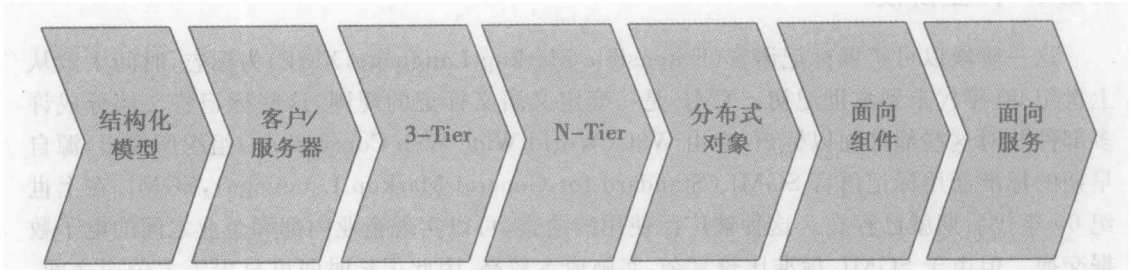


图 1-1 软件体系结构的发展过程

伴随着大规模企业级应用的不断发展,为了解决异构性、互操作性和不断变化的需求等问题,一种新型的软件体系架构应运而生——面向服务的架构(Service Oriented Architecture,SOA),它是在面向对象技术和面向组件技术之上的更高级别抽象。

SOA 将功能都封装为服务。服务都是独立的,服务的内部实现对于服务的使用者来说是透明的,服务的使用者只需通过服务提供的接口来调用服务,他所关心的不是功能如何实现,而是该服务是否能返回他所期望的结果。无论从技术角度,还是地理位置角度来看,服务的可用性并不局限在某一个特定的范围内,服务可能是在相同的应用程序中,也可能是在公司内部网不同系统上的不同地址空间中,还有可能是在合作伙伴的系统上的应用程序中。服务可能是由 C++ 语言开发,也可能是用 Java 或者 Visual Basic 开发的,甚至可能是任何一种已经存在语言开发的。

服务之间通过交互、协同作业从而实现完整的业务逻辑,通过服务总线和业务流程管理来连接服务和服务请求。业务流程管理定义好服务的执行序列,将它们按照适当的顺序调用来产生最后的结果。这种松耦合的体系结构使得各服务在交互过程中无需考虑双方内部的实现细节、部署在什么平台之上。应用系统的松耦合还提高了应用系统的灵活性和互操作性,这种灵活性和互操作性优于传统方法构建的紧耦合的、跨平台的应用系统。

1.2 SOA 的发展起源

SOA 最早是由 Gartner 在 1996 年提出的,由于当时的技术水平和市场环境尚不具备实施 SOA 的条件,因此当时 SOA 并未引起人们的广泛关注。伴随着互联网的浪潮,越来越多的企业将业务转移到互联网领域,带动了电子商务的蓬勃发展。为了能够将公司的业务打包成独立的、具有很强伸缩性、基于互联网的服务,人们提出了 Web 服务。

Web 服务开始流行以后,互联网迅速出现了大量的基于不同平台和语言开发的 Web 服务组件。为了能够有效地对这些为数众多的组件进行管理,人们迫切需要找到一种新的面向服务的分布式 Web 架构。该架构要能够使这些由不同组织开发的 Web 服务能够相互

交互,并兼顾服务的重用性和可管理性。由此,人们重新把注意力转向 SOA,并赋予其新的时代特征。到了 2002 年 12 月,Gartner 提出 SOA 是“现代应用开发领域最重要的课题”,并预计到 2008 年,SOA 将成为占有绝对优势的软件工程实践方法。回顾 SOA 发展历程,本书将其大致分为了三个阶段。下面将分别介绍每个阶段的重要标准和规范。

1.2.1 产生阶段

这一阶段以可扩展标记语言(Extensible Markup Language,XML)为标志,时间大致从上世纪 90 年代末到本世纪初。XML 是一套定义语义标记的规则,这些标记将文档分成许多部件并对这些部件加以标识;它由 W3C(World Wide Web Consortium)组织所提出,源自早期的标准通用标记语言 SGML(Standard for General Markup Language),SGML 在上世纪 60 年代后期就已存在。这种被广泛使用的语言,可以实现企业内部和企业之间的电子数据交换。但由于 SGML 标准比较复杂,实施成本较高,因此很长时间里只用于大公司之间,限制了它的推广和普及。

XML 和 HTML 都源自 SGML,它们都含有标记,有着相似的语法,HTML 和 XML 的最大区别在于:HTML 是一个定型的标记语言,它用固有的标记来描述网页,将网页的内容数据和网页的外观数据混合在同一个 HTML 文档中,因此 HTML 文档内部结构的条理性很差,是一种非结构化的描述形式。XML 则没有固定的标记,不描述网页具体的外观,它只是描述内容的数据形式和结构,XML 文档内部结构条理性较好,是一种结构化的描述形式,易于编写和阅读。XML 使用的是非专有格式的标记语言,允许不同的专业人员开发与自己的特定领域有关的标记语言,这使得 XML 成为在不同的应用领域间交换数据的理想格式。XML 的功能非常强大,对于人或是计算机来说,都容易阅读和编写,因而成为异构系统之间交换数据的首选数据描述语言。

通过 XML,开发人员摆脱了 HTML 语言的限制,可以将任何文档转换成 XML 格式,然后通过互联网传输。借助 XML 转换语言 XSLT(Extensible Stylesheet Language Transformation),接收方可以很容易解析和抽取 XML 文件中的数据。这使得企业既能够将数据以一种统一的格式描述和交换,同时又不必负担 SGML 那样高的成本。

在 SOA 架构中,XML 是 SOA 的信息交换标准,规定了服务之间以及服务内部数据交换的格式和结构。XSD(xml schema definition)规范保障了消息数据的完整性和有效性,而 XSLT 使得不同的数据表达能通过 Schema 映射而互相通信。

1.2.2 推动阶段

2000 年以后,人们普遍认识到基于互联网之上的电子商务具有极大的发展潜力,因此需要创建一套全新的基于互联网的开放通信框架,以满足企业对电子商务中各分立系统之间通信的要求。于是,人们提出了 Web 服务的概念,希望将企业的对外服务封装为统一的、标准的 Web 服务,实现异构系统之间的简单交互。这一时期,出现了三个著名的 Web 服务标准和规范:

- 简单对象访问协议(Simple Object Access Protocol,SOAP)
- Web 服务描述语言(Web Services Description Language,WSDL)
- 公共服务发现和集成协议(Universal Discovery Description and Integration,UDDI)

这三个标准可谓 Web 服务的核心,极大地推动了 Web 服务的普及和发展。短短几年之间,互联网上出现了大量的 Web 服务,越来越多的企业将其对外服务或业务接口封装成 Web 服务,有力地推动了电子商务和互联网的发展。Web 服务也成为互联网 Web 2.0 时代的一项重要特征。

1.2.3 成长阶段

从 2005 年开始,基于 SOA 架构的应用迅速升温。软件行业的从业人员都开始把目光投向 SOA。各大厂商也通过建立厂商间的协作组织共同努力制定中立的 SOA 标准。这一努力最重要的成果体现在 3 个重量级规范上:SCA(Service Component Architecture,服务组件架构)、SDO(Service Data Object,服务数据对象)和 WS-Policy(Web Service Policy,Web 服务策略)。SCA 和 SDO 构成了 SOA 编程模型的基础,而 WS-Policy 建立了 SOA 组件之间安全交互的规范。SCA 规范帮助简化服务的构建与整合,SDO 规范则关注对不同应用中多种格式数据的统一访问,WS-Policy 则用于自动配置代理服务和业务服务中的消息级安全。业界普遍认为,这三项规范的发布,标志着 SOA 已经进入了实质性阶段,将对 SOA 的发展发挥重要的作用。从整体架构角度看,人们已经把关注点从简单的 Web 服务拓展到面向服务体系架构的各个方面,包括安全、业务流程和事务处理等。

1.2.4 SOA 定义

目前,SOA 还没有一个公认的定义,许多组织从不同角度和不同侧面对 SOA 进行了描述。

Gartner 将 SOA 描述为:“一种客户端/服务器的软件设计方法,它由软件服务和软件服务使用者组成。SOA 与大多数通用的客户端/服务器模型的不同之处在于它着重强调软件组件的松散耦合,并使用独立的标准接口。”

W3C 将 SOA 定义为:“一种应用程序体系结构,在这种体系结构中,所有功能都定义为独立的服务,这些服务带有定义明确的可调用接口,可以按照定义好的顺序调用这些服务来组成业务流程。”

虽然不同厂商或个人对 SOA 有着不同的理解,但是仍然可以从上述的定义中看到 SOA 的几个关键特性:一种粗粒度、松耦合的服务架构,服务之间通过简单、精确定义的接口进行通讯,不涉及底层编程接口和通讯模型。也就是说,SOA 并不是一种具体的技术,而是一种架构和组织 IT 基础结构及业务功能的方法。

SOA 将应用程序的不同功能单元(称为服务)通过服务之间定义良好的接口和契约联系起来。接口采用中立的方式进行定义,它独立于实现服务的硬件平台、操作系统和编程语言,这使得构建在各种系统中的服务可以采用一种统一和通用的方式进行交互。SOA 通过服务的高重用性来降低开发成本;通过服务的粗粒度、松耦合特性来屏蔽复杂业务逻辑,从而降低业务流程复杂度;通过组织专业分工,使得软件开发的大规模化成为可能。

综上所述,SOA 是一种架构模型,它可以根据需求通过网络对松散耦合的粗粒度应用组件进行分布式部署、组合和使用。SOA 可以为客户解决“信息孤岛”问题,并能够提供随需应变的服务。SOA 将原有的应用和资源转变为可共享的标准服务,企业只需根据自己的策略来定制流程,把应用作为服务“拿来就用”,而无需考虑底层的集成,从而实现信息技术