



# SOA概念、技术与设计

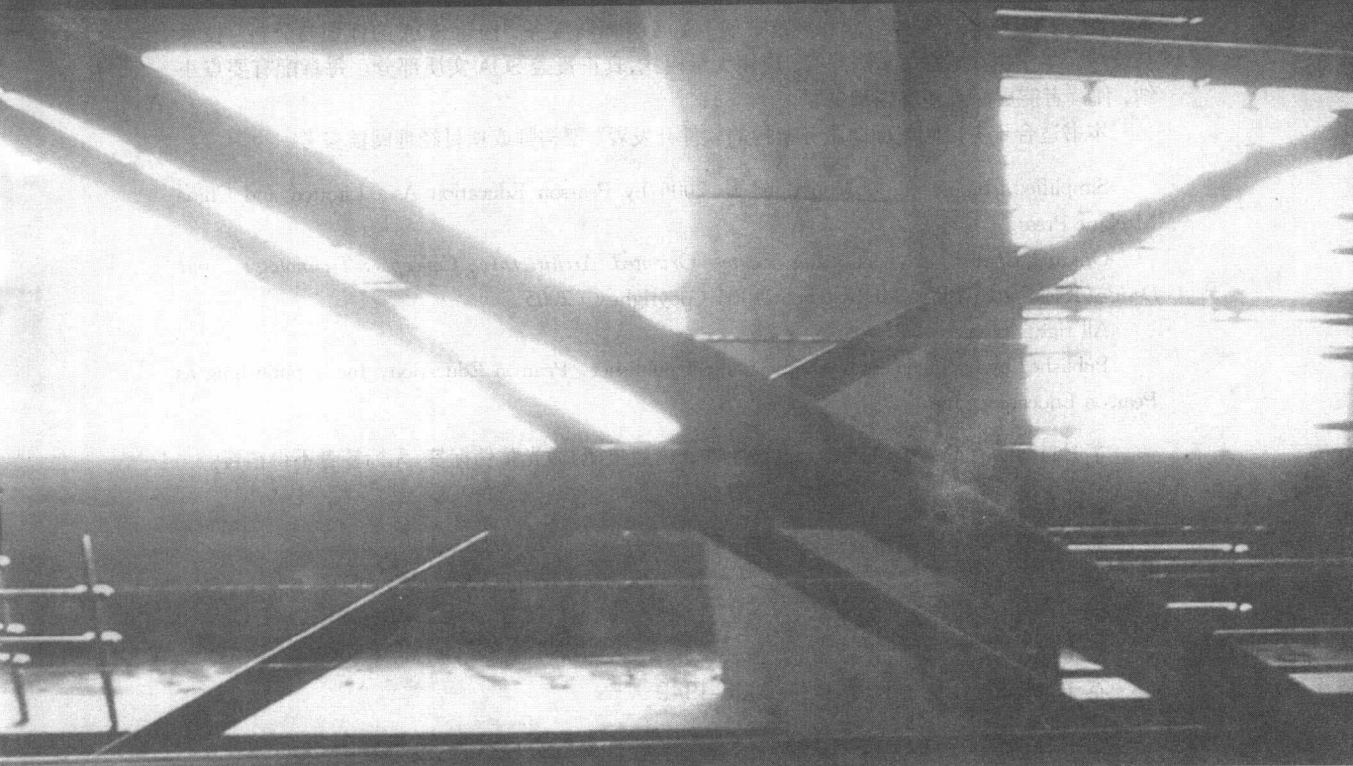
Service-Oriented Architecture

Concepts, Technology, and Design

(美) Thomas Erl 著  
王满红 陈荣华 译



机械工业出版社  
China Machine Press



# SOA概念、技术与设计

Service-Oriented Architecture

Concepts, Technology, and Design

(美) Thomas Erl 著

王满红 陈荣华 译



机械工业出版社  
China Machine Press

本书系统介绍 SOA 概念、技术与设计。全书共分五部分,分别介绍 SOA 与 Web 服务的基本原理,SOA 与第二代 Web 服务规范扩展,SOA 与面向服务,构建 SOA 的计划与分析、技术与设计。本书由浅入深,示例翔实,应用大量模式,真正覆盖 SOA 实质部分。每章配有要点小结,使读者能更深入理解该章主题。

本书适合于考虑实施面向服务架构的软件开发者、架构师或项目经理阅读参考。

Simplified Chinese edition copyright © 2006 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Service-Oriented Architecture: Concepts, Technology, and Design* (ISBN:0-13-185858-0) by Thomas Erl, Copyright © 2005.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Pearson Education, Inc.

本书封面贴有 Pearson Education(培生教育出版集团)激光防伪标签,无标签者不得销售。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

**本书版权登记号: 图字: 01-2006-1940**

**图书在版编目(CIP)数据**

SOA 概念、技术与设计/(美)伊尔(Erl,T.)著;王满红,陈荣华译. —北京:机械工业出版社,2006.10

书名原文: *Service-Oriented Architecture: Concepts, Technology, and Design*  
ISBN 7-111-19727-5

I.S… II.①伊… ②王… ③陈… III.互联网络—网络服务器 IV.TP368.5

中国版本图书馆 CIP 数据核字(2006)第 091946 号

机械工业出版社(北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 陈冀康

北京牛山世兴印刷厂印刷·新华书店北京发行所发行

2007 年 1 月第 1 版第 1 次印刷

186mm×240mm·26 印张

定价: 56.00 元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换  
本社购书热线:(010)68326294

# 对本书赞誉

在当今 IT 界，面向服务架构是一个热门主题，但却经常被误解。托马斯清晰地描述了面向服务与 Web 服务背后的概念、规范 and 标准。针对要采用 SOA 的企业，给出了面向服务的分析、计划与设计的细节性建议。这是一本必读书籍！

——亚历克斯·林奇，微软企业服务部首席咨询师

在设计中应用 SOA 的一个主要目的是为我们的解决方案提供业务价值。正确理解面向服务的分析、设计与开发方案至关重要。托马斯做了一件漂亮的事情，他实际上用本书揭开了 SOA 的神秘面纱。

——瑞克·韦夫，IBM 公司资深咨询师、SW I/T 认证专家

这是一本实用的 SOA 原则、策略与最佳实践的指南，它揭示了在复杂的企业环境中采用 SOA 的基本框架。

——萨米尔·泰基，Sun 公司高级工程师

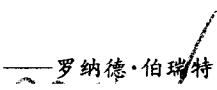
对于这一新兴领域来说，本书非常及时，十分必要。通过对该领域的原则与细微差别的澄清，作者提供了 SOA 关键方面的综合处理方法：从分析、计划到标准，涉及从 WS-规范到 BPEL。我要向计划采用 SOA 原则的客户及合作伙伴推荐本书。

——瑞维·佩尔浦，Rogue Wave 软件公司资深领域架构师

总之，这是基于在实际生产环境中的实践经验而写的一本 SOA 书籍。太多的 SOA 书籍陷入了有关 Web 服务标准的技术细节中，或者只是在重复厂商的宣传。而本书真正覆盖了最实质的部分：复杂的过程计划、设计以及适合组织目标的面向服务架构实现。它对于考虑实施面向服务架构的任何软件开发者、架构师或项目经理而言，都是良师益友。

——普里西拉·沃姆斯利，Datypic 公司管理总监

本书对于面向服务架构的介绍已经尽如人意。作者用一卷书覆盖了所有主题，从理论到现实中的技术细节。示例相当丰富，行文极其清晰。

—— 罗纳德·伯瑞特，《XML and Databases》一书的作者

一本获得现实答案与示例的 SOA 书籍。可跟随作者进入 SOA 的真实世界尽情畅游。从设计到行业标准，本书都写得相当不错，并能够作为日常参考之用。当你自己开始面向服务探险时，你会把这本书装入你的行囊。

——克拉克·赛尔，CSell 公司副总裁

现在，那些致力于改进现有的面向服务架构而不是仅仅停留在简单 Web 服务的机构终于有了可用的专家级资源。在引领通往真正面向服务的企业之路方面，作者通过细节性的实践讨论与案例研究揭开了复杂的开放 WS-I 标准的面纱。本书描述深入清晰，在领域指导方面的工作做得近乎完美。

——凯文·P·戴维斯博士，软件架构师

对于正在或打算从事 Web 服务或者面向服务架构解决方案的架构师、开发者与管理者而言，本书是一本绝佳的指南。本书分为四个部分，其中第一部分对 XML 的基础技术、Web 服务与面向服务架构做了详细描述，并对新兴标准给予关注。本书写得相当好，覆盖了所有主题。我极力向所有关注企业级服务架构的人推荐这本书。

——亚当·霍赛克，Broadstrokes 公司总裁与 CTO

更多的赞誉请参考：[www.soabooks.com/reviews.asp](http://www.soabooks.com/reviews.asp)。

# 译者序

面向服务架构(SOA)是当今 IT 界倍受关注的主题,也是未来的发展趋势。Gartner Group 预测,到 2008 年,SOA 将成为占绝对优势的软件工程技术。主流企业,特别是软件企业,现在就应该了解和应用 SOA 的概念和技术。

但是,与先前的任何技术或概念相比,SOA 又是一个宏大的主题,这是 IT 技术发展积累的结果,也是由它所担负的使命所决定的。同时,与 J2EE、.NET 等具体技术相比,SOA 更具厂商中立性,到目前为止,还没有哪家厂商能宣称其“拥有 SOA”。因而,SOA 主题的研究既要继承历史,立足现实,又要面向未来;既要涉及技术,又要顾及市场;既要保持中立,又要考虑既有的厂商技术。所有这些,本书都进行了完整而深入的阐述。读完本书,犹如享受了精美的 SOA 盛宴。

本书主要包括三个方面的内容,SOA 的概念、相关技术和规范、设计流程。书中对于 SOA 的原则及其所带来的效益进行了深入细致的探讨,这甚至成为贯穿全书的一条线索,从理论到实践,从原则到产品支持,不厌其烦。我想,这自有它的道理,因为任何一种技术,尤其是它刚出现的时候人们很容易会为了赶时髦而使用它,而使用者往往忽视了任何一种技术都是为解决某种问题而存在的,即技术因市场需求而生,是“果”而非“因”。只有深刻理解 SOA 的效益,才能享受它所带来的价值而规避其局限性。

本书的另一大特色还在于其丰富而系统的案例研究,作者不仅虚构了两家有代表性的大型和小型企业,逐步地帮它们建立起 SOA 的企业环境,而且对于抽象的概念还辅以“简言之”的类推场景,从日常的生活常理中对晦涩的概念给予深入浅出的解释。这样的示例占据了近乎全书三分之一的篇幅,从而使得本书极具实践意义。当然,不要指望照抄案例就能构建自己的 SOA 环境,它只是对理论理解的一种辅助手段,而不是 SOA 的“教程”。

本书可以帮助读者建立起一整套关于 SOA 的理论体系以及构建 SOA 环境所需的技术和指导,带领我们走入 SOA 的神圣殿堂;同时,它也可以作为一本必备的 SOA 参考手册,在你走入技术迷途时为你指引方向。

很幸运,能够有机会翻译此书。虽然“人行”近 10 年,但对于翻译则是“新手”,加之本书涉及面广、词汇量大,更有一些新术语需要“创新”,令译者颇感费力。翻译过程中参考了大量的现有译法,但由于译者水平所限,加之时间仓促,错误之处在所难免,恳请广大专家、读者批评指正。

本书第 10~17 章由陈荣华负责初译,王满红对其余部分进行翻译、并负责统稿和审校。

感谢陈冀康先生,没有他的慧眼推荐、极力督促和指导,本书中文版不会得以快速面世。同时感谢他对本人的信任,这给了我们极大的信心和动力。还要感谢我的朋友温昱,他的鼓励和引荐促成了翻译工作的启动。

机械工业出版社的编辑们认真负责的态度也令我感动,他们的工作保证了本书的可读性和准确性。最后,特别感谢我的妻子马晓艳,她坚定不移地支持以及对家庭和儿子的全力照料,使我能够夜以继日地工作并最终按时完稿。

您对本书有任何建议或意见,欢迎发送电子邮件 [manhong.wang@gmail.com](mailto:manhong.wang@gmail.com) 或 [unixcrh@gmail.com](mailto:unixcrh@gmail.com)。

王满红

2006 年 5 月 15 日

## 译者简介

**王满红** 任普元软件产品总监,主要从事软件产品开发、管理及设计工作,10 年以上软件从业经验。对面向构件的中间件及大型企业级软件的开发有深入研究,与黄柳青合著《构件中国:面向构件的方法与实践》(清华大学出版社,2006 年 5 月)。

**陈荣华** Family Healthcare 技术总监,全面负责产品调研、架构设计、开发、质保和工程实施,10 年以上软件开发和管理经验。专注于 OOA/D 及网络与通信技术,对 BSD 系统内核有深入研究。在此之前曾任东信冠群软件有限公司研发总监,从事电信 IP 数据网络和业务管理系统的开发和管理工作的。

# 前言

本书的创作几乎用了一年的时间,包括写作、研究以及对于所关注问题的范围及重点的不断扩展。尽管本书大部分章节集中于从厂商中立的角度来讨论面向服务架构,但要完成从这个视角的精确阐述,还需要花费大量精力去评估支持 SOA 的所有主流厂商平台。作为这次研究活动的一部分,我与一百多位资深 IT 专家进行了交谈,有的是通过直接的面谈,另一些是我参与平台的评判工作时的意外收获。

本项目中最有趣的因素已经在以 Web 服务为背景的面向服务中体现。然而要把面向服务的各个组成作为模式来研究时,我才开始意识到它们就植根于已有的创新中。但又有根本的区别,因为这是一个混合传统概念与新概念的独特架构模型。

尽管 SOA 是以崭新的面貌出现的,但在基础层面却是基于一个陈旧的既有思想学派。面向服务,作为与逻辑单元相隔离的一种方法,是一个非常通用的概念。随着书中章节的进展,我开始注意到这在日常生活中更为常见。我们日常接触的个体、人和组织要么提供某种形式的服务,要么是一个服务的参与者。尽管如此,一旦应用到技术架构,面向服务却只关注面向服务世界的一个特殊部分:业务自动化。

用最小的付出获得最大的效益,这是竞争性业务对公司策略目标的要求,持续浪费资源的低效组织将会落后。组织的业务自动化方式是决定其运营效率的根本因素,从而也是对其敢于冒险的奖励。

这就是 SOA 的价值所在。通过面向服务形成自动业务逻辑,能够保护现有投资,有助于智能化地表达业务,进而表现其内在的自动敏捷性。当与 Web 服务技术平台联合时,SOA 提供了一个重要的潜在现实效益,从而改变组织的技术与发展前景。我的目标是帮助读者探索、理解并实现这个潜能。

## 致谢

在本书的写作期间,我有幸遇到一个强大的团队,包括技术评审、高级编辑、制作人及营销专家。我感谢你们所有人不辞辛劳的努力,特别感谢我的家庭对我的理解和坚定不移的支持。

## 作者简介

托马斯·伊尔是 SOA 系统公司(SOA Systems Inc.)的创始人,([www.soasystems.com](http://www.soasystems.com)),这是一家特别提供 SOA 咨询及培训服务的企业解决方案提供者。他的前一本书,《Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services》,是 2004 年 Web 服务及 SOA 类别的畅销书。该书处理了诸多集成问题,并为转向 SOA 提供了策略及最佳实践。

托马斯是 OASIS 的投票成员,并积极进行相关研究,譬如 XML 与 Web 服务集成框架(XWIF)。他是私人 and 公开会议的演讲者和讲师,且已经发表诸多论文。

更多信息请参见 [www.thomaserl.com/technology/](http://www.thomaserl.com/technology/)。

## 关于 SOA 系统公司

SOA 系统公司(SOA System Inc.)是一家积极参与面向服务架构、面向服务、XML 及 Web 服务标准与技术研究及开发的咨询公司。通过研究和企业解决方案项目, SOA 系统公司已经开发出一整套经过验证的、用于集成和实现面向服务的概念、技术及架构的主流方法。

更多信息请访问 [www.soasystems.com](http://www.soasystems.com)。

SOA 系统提供的一个咨询服务是广泛的 SOA 迁移计划, 以及厂商技术产品的目标评估。

更多信息请访问 [www.soaplanning.com](http://www.soaplanning.com)。

本书的内容是由 SOA 系统公司所开发和提供的一系列 SOA 基础讲座和专题研讨的基础。

更多信息请访问 [www.soatraining.com](http://www.soatraining.com)。

## 关于照片

我希望你喜欢本书封面及插页所用的图片。这些照片是我最近参观不同城市时自己拍摄的, 这些城市包括布拉迪斯拉发(Bratislava)、布拉格(Prague)、维也纳(Vienna)和华盛顿(Washington)。有些图片在一定程度上是和特定主题相关的一种象征, 而另一些图片只是用一种无法言传的方式表达了一个主题。收编这些图片使我意识到, 它们与服务很相似, 独特并抽象地表达了世界的不同部分。

# 目 录

译者序

前言

|                                 |    |
|---------------------------------|----|
| 第 1 章 引言 .....                  | 1  |
| 1.1 本书的重要性 .....                | 1  |
| 1.1.1 虚假的 SOA .....             | 1  |
| 1.1.2 理想的 SOA .....             | 2  |
| 1.1.3 现实的 SOA .....             | 2  |
| 1.2 本书的目标 .....                 | 2  |
| 1.2.1 理解 SOA、面向服务和 Web 服务 ..... | 2  |
| 1.2.2 学习如何用 Web 服务构建 SOA .....  | 3  |
| 1.3 本书为谁而写 .....                | 3  |
| 1.4 本书不涉及的内容 .....              | 3  |
| 1.5 本书的组织结构 .....               | 4  |
| 1.6 附加信息 .....                  | 10 |
| 第 2 章 案例研究 .....                | 11 |
| 2.1 如何使用案例研究 .....              | 11 |
| 2.1.1 样式特征 .....                | 11 |
| 2.1.2 与抽象内容的关系 .....            | 11 |
| 2.1.3 代码示例 .....                | 11 |
| 2.2 1 号案例背景: 瑞克有限公司 .....       | 12 |
| 2.2.1 历史 .....                  | 12 |
| 2.2.2 技术基础架构 .....              | 12 |
| 2.2.3 自动化方案 .....               | 12 |
| 2.2.4 业务目标与障碍 .....             | 12 |
| 2.3 2 号案例背景: 交通岸线系统公司 .....     | 13 |
| 2.3.1 历史 .....                  | 13 |
| 2.3.2 技术基础架构 .....              | 13 |
| 2.3.3 自动化方案 .....               | 13 |
| 2.3.4 业务目标与障碍 .....             | 13 |

## 第一部分 SOA 与 Web 服务基础

|                    |    |
|--------------------|----|
| 第 3 章 SOA 简介 ..... | 16 |
| 3.1 SOA 基础 .....   | 16 |
| 3.1.1 面向服务类比 ..... | 16 |

|                                       |    |
|---------------------------------------|----|
| 3.1.2 服务如何封装逻辑 .....                  | 17 |
| 3.1.3 服务如何关联 .....                    | 17 |
| 3.1.4 服务如何通信 .....                    | 18 |
| 3.1.5 服务如何设计 .....                    | 18 |
| 3.1.6 服务如何构建 .....                    | 19 |
| 3.1.7 基本 SOA .....                    | 19 |
| 3.2 当代 SOA 的共同特征 .....                | 20 |
| 3.2.1 当代 SOA 是面向服务计算平台的核心 .....       | 21 |
| 3.2.2 当代 SOA 可提升服务质量 .....            | 21 |
| 3.2.3 当代 SOA 是根本上自治的 .....            | 22 |
| 3.2.4 当代 SOA 基于开放标准 .....             | 22 |
| 3.2.5 当代 SOA 支持厂商多样性 .....            | 22 |
| 3.2.6 当代 SOA 促进发现 .....               | 23 |
| 3.2.7 当代 SOA 鼓励内在互操作性 .....           | 23 |
| 3.2.8 当代 SOA 促进联邦 .....               | 23 |
| 3.2.9 当代 SOA 提升架构组合能力 .....           | 24 |
| 3.2.10 当代 SOA 鼓励内在复用性 .....           | 24 |
| 3.2.11 当代 SOA 强调可扩展性 .....            | 25 |
| 3.2.12 当代 SOA 支持面向服务的业务建模模式 .....     | 25 |
| 3.2.13 当代 SOA 实现抽象层 .....             | 26 |
| 3.2.14 当代 SOA 促进整个企业的松散耦合 .....       | 26 |
| 3.2.15 当代 SOA 促进组织敏捷性 .....           | 27 |
| 3.2.16 当代 SOA 是一个构件 .....             | 27 |
| 3.2.17 当代 SOA 是一场变革 .....             | 28 |
| 3.2.18 当代 SOA 还有待成熟 .....             | 28 |
| 3.2.19 当代 SOA 是可实现的理想 .....           | 28 |
| 3.2.20 定义 SOA .....                   | 28 |
| 3.2.21 可分离的具体特征 .....                 | 29 |
| 3.3 常见的 SOA 误解 .....                  | 29 |
| 3.3.1 使用 Web 服务的应用就是面向服务 .....        | 30 |
| 3.3.2 SOA 仅仅是重建 Web 服务品牌的一个市场术语 ..... | 30 |
| 3.3.3 SOA 仅仅是利用 Web 服务 .....          |    |

|                                          |    |                                 |    |
|------------------------------------------|----|---------------------------------|----|
| 重建分布式计算品牌的一个市场术语 .....                   | 30 | 4.3.2 比较 SOA 与客户端 - 服务器架构 ..... | 45 |
| 3.3.4 SOA 简化分布式计算 .....                  | 30 | 4.3.3 比较 SOA 与分布式互联网架构 .....    | 48 |
| 3.3.5 使用了 WS- * 扩展的 Web 服务应用就是面向服务 ..... | 30 | 4.3.4 比较 SOA 与混合 Web 服务架构 ..... | 53 |
| 3.3.6 只要你理解 Web 服务, 构建 SOA 就不成问题 .....   | 30 | 4.3.5 面向服务与面向对象 (第一部分) .....    | 54 |
| 3.3.7 一旦你走到 SOA, 一切都具备了互操作性 .....        | 31 | 第 5 章 Web 服务与基本 SOA .....       | 56 |
| 3.4 SOA 常见的切实效益 .....                    | 31 | 5.1 Web 服务框架 .....              | 56 |
| 3.4.1 改良的集成 (及内在互操作性) .....              | 31 | 5.2 服务 (作为 Web 服务) .....        | 57 |
| 3.4.2 内在的复用 .....                        | 31 | 5.2.1 服务角色 .....                | 58 |
| 3.4.3 改进的架构与方案 .....                     | 32 | 5.2.2 服务模型 .....                | 65 |
| 3.4.4 保护遗留资产 .....                       | 32 | 5.3 服务描述 (用 WSDL 语言) .....      | 68 |
| 3.4.5 建立标准的 XML 数据表示 .....               | 32 | 5.3.1 服务端点与服务描述 .....           | 70 |
| 3.4.6 聚焦通信基础设施投资 .....                   | 33 | 5.3.2 抽象描述 .....                | 70 |
| 3.4.7 “最优供给”可替代 .....                    | 33 | 5.3.3 具体描述 .....                | 70 |
| 3.4.8 组织敏捷性 .....                        | 33 | 5.3.4 元数据与服务契约 .....            | 71 |
| 3.5 采用 SOA 的常见失误 .....                   | 33 | 5.3.5 语义描述 .....                | 72 |
| 3.5.1 构建类似传统分布式架构的面向服务架构 .....           | 33 | 5.3.6 服务描述广告与发现 .....           | 72 |
| 3.5.2 非标准化的 SOA .....                    | 34 | 5.4 消息 (以 SOAP 规范) .....        | 74 |
| 3.5.3 没有建立迁移计划 .....                     | 34 | 5.4.1 消息 .....                  | 75 |
| 3.5.4 没有以 XML 基础架构开始 .....               | 34 | 5.4.2 节点 .....                  | 78 |
| 3.5.5 不理解 SOA 性能需求 .....                 | 35 | 5.4.3 消息路径 .....                | 80 |
| 3.5.6 不了解 Web 服务安全性 .....                | 35 |                                 |    |
| 3.5.7 没有保持与产品平台和标准开发的联系 .....            | 35 |                                 |    |
| 第 4 章 SOA 的演变 .....                      | 37 |                                 |    |
| 4.1 SOA 历史 .....                         | 37 |                                 |    |
| 4.1.1 XML 简史 .....                       | 37 |                                 |    |
| 4.1.2 Web 服务简史 .....                     | 38 |                                 |    |
| 4.1.3 SOA 简史 .....                       | 38 |                                 |    |
| 4.1.4 SOA 如何改造 XML 与 Web 服务 .....        | 39 |                                 |    |
| 4.2 SOA 的持续演变 .....                      | 40 |                                 |    |
| 4.2.1 比较“标准”、“规范”与“扩展” .....             | 40 |                                 |    |
| 4.2.2 标准组织对 SOA 的贡献 .....                | 40 |                                 |    |
| 4.2.3 主流厂商对 SOA 的贡献 .....                | 42 |                                 |    |
| 4.3 SOA 与过去架构的比较 .....                   | 44 |                                 |    |
| 4.3.1 什么是架构 .....                        | 44 |                                 |    |
|                                          |    | 第二部分 SOA 与 WS- * 扩展             |    |
|                                          |    | 第 6 章 Web 服务与当代 SOA:            |    |
|                                          |    | 活动管理与组合 .....                   | 85 |
|                                          |    | 6.1 消息交换模式 .....                | 86 |
|                                          |    | 6.1.1 基本 MEP .....              | 87 |
|                                          |    | 6.1.2 MEP 与 SOAP .....          | 91 |
|                                          |    | 6.1.3 MEP 与 WSDL .....          | 91 |
|                                          |    | 6.1.4 MEP 与 SOA .....           | 92 |
|                                          |    | 6.2 服务活动 .....                  | 92 |
|                                          |    | 6.2.1 基本与复杂的服务活动 .....          | 93 |
|                                          |    | 6.2.2 服务活动与 SOA .....           | 94 |
|                                          |    | 6.3 协调 .....                    | 95 |
|                                          |    | 6.3.1 协调者组合 .....               | 96 |
|                                          |    | 6.3.2 协调类型与协调协议 .....           | 97 |
|                                          |    | 6.3.3 协调语境与协调参与者 .....          | 97 |
|                                          |    | 6.3.4 激活与注册流程 .....             | 97 |
|                                          |    | 6.3.5 完成流程 .....                | 98 |

|                                       |     |                          |     |
|---------------------------------------|-----|--------------------------|-----|
| 6.3.6 协调与 SOA .....                   | 98  | 7.2.5 可靠消息传递与寻址 .....    | 130 |
| 6.4 原子事务 .....                        | 100 | 7.2.6 可靠消息传递与 SOA .....  | 130 |
| 6.4.1 ACID 事务 .....                   | 101 | 7.3 关联 .....             | 132 |
| 6.4.2 原子事务协议 .....                    | 102 | 7.3.1 抽象关联 .....         | 132 |
| 6.4.3 原子事务协调 .....                    | 102 | 7.3.2 MEP 与活动关联 .....    | 132 |
| 6.4.4 原子事务处理 .....                    | 103 | 7.3.3 协调关联 .....         | 133 |
| 6.4.5 原子事务与 SOA .....                 | 103 | 7.3.4 编排关联 .....         | 133 |
| 6.5 业务活动 .....                        | 105 | 7.3.5 寻址关联 .....         | 133 |
| 6.5.1 业务活动协议 .....                    | 106 | 7.3.6 可靠消息传递关联 .....     | 133 |
| 6.5.2 业务活动协调者 .....                   | 106 | 7.3.7 关联与 SOA .....      | 133 |
| 6.5.3 业务活动状态 .....                    | 106 | 7.4 策略 .....             | 134 |
| 6.5.4 业务活动与原子事务 .....                 | 107 | 7.4.1 WS-策略框架 .....      | 135 |
| 6.5.5 业务活动与 SOA .....                 | 107 | 7.4.2 策略断言与策略抉择 .....    | 135 |
| 6.6 编排 .....                          | 109 | 7.4.3 策略断言类型与策略词汇 .....  | 135 |
| 6.6.1 业务协议与流程定义 .....                 | 111 | 7.4.4 策略主题与策略范围 .....    | 135 |
| 6.6.2 流程服务与伙伴服务 .....                 | 111 | 7.4.5 策略表达式与策略附件 .....   | 135 |
| 6.6.3 基本活动与结构化活动 .....                | 112 | 7.4.6 你必须真正知道什么 .....    | 135 |
| 6.6.4 序列、流程与链接 .....                  | 112 | 7.4.7 协作策略 .....         | 136 |
| 6.6.5 编排与活动 .....                     | 112 | 7.4.8 编排与编导策略 .....      | 136 |
| 6.6.6 编排与协调 .....                     | 112 | 7.4.9 可靠消息传递策略 .....     | 136 |
| 6.6.7 编排与 SOA .....                   | 112 | 7.4.10 策略与 SOA .....     | 136 |
| 6.7 编导 .....                          | 114 | 7.5 元数据交换 .....          | 137 |
| 6.7.1 协作 .....                        | 115 | 7.5.1 WS-元数据交换规范 .....   | 138 |
| 6.7.2 角色与参与者 .....                    | 115 | 7.5.2 获取元数据请求与响应消息 ..... | 138 |
| 6.7.3 关系与通道 .....                     | 115 | 7.5.3 获取请求与响应消息 .....    | 138 |
| 6.7.4 交互与工作单元 .....                   | 115 | 7.5.4 选择性的元数据恢复 .....    | 139 |
| 6.7.5 可复用性、可组合性与成熟度 .....             | 115 | 7.5.5 元数据交换与服务描述发现 ..... | 140 |
| 6.7.6 编排与编导 .....                     | 115 | 7.5.6 元数据交换与版本控制 .....   | 140 |
| 6.7.7 编导与 SOA .....                   | 117 | 7.5.7 元数据交换与 SOA .....   | 140 |
| 第 7 章 Web 服务与当代 SOA: 高级               |     | 7.6 安全 .....             | 142 |
| 消息、元数据和安全 .....                       | 119 | 7.6.1 识别、认证与授权 .....     | 143 |
| 7.1 寻址 .....                          | 121 | 7.6.2 单点登录 .....         | 144 |
| 7.1.1 端点引用 .....                      | 121 | 7.6.3 机密性与完整性 .....      | 145 |
| 7.1.2 消息信息报头 .....                    | 122 | 7.6.4 传输级安全与消息级安全 .....  | 145 |
| 7.1.3 寻址与传输协议独立 .....                 | 123 | 7.6.5 加密与数字签名 .....      | 146 |
| 7.1.4 寻址与 SOA .....                   | 124 | 7.6.6 安全与 SOA .....      | 146 |
| 7.2 可靠消息传递 .....                      | 125 | 7.7 通知与事件 .....          | 148 |
| 7.2.1 RM 源、RM 目的地、应用源与<br>应用目的地 ..... | 126 | 7.7.1 发布与订阅摘要 .....      | 148 |
| 7.2.2 序列 .....                        | 127 | 7.7.2 一个概念、两个规范 .....    | 149 |
| 7.2.3 确认 .....                        | 127 | 7.7.3 WS-通知框架 .....      | 149 |
| 7.2.4 交付保证 .....                      | 128 | 7.7.4 WS-事件规范 .....      | 150 |
|                                       |     | 7.7.5 WS-通知与 WS-事件 ..... | 152 |

|                        |     |
|------------------------|-----|
| 7.7.6 通知、事件与 SOA ..... | 152 |
|------------------------|-----|

### 第三部分 SOA 与面向服务

|                                |     |
|--------------------------------|-----|
| 第 8 章 面向服务的原则 .....            | 156 |
| 8.1 面向服务与企业 .....              | 156 |
| 8.2 面向服务架构剖析 .....             | 158 |
| 8.2.1 Web 服务框架的逻辑组件 .....      | 159 |
| 8.2.2 自动化逻辑的逻辑组件 .....         | 159 |
| 8.2.3 SOA 组件 .....             | 160 |
| 8.2.4 SOA 中组件如何内部关联 .....      | 161 |
| 8.3 面向服务的一般原则 .....            | 162 |
| 8.3.1 服务是可复用的 .....            | 163 |
| 8.3.2 服务共享一个正式契约 .....         | 164 |
| 8.3.3 服务是松散耦合的 .....           | 165 |
| 8.3.4 服务抽象底层逻辑 .....           | 166 |
| 8.3.5 服务是可组合的 .....            | 168 |
| 8.3.6 服务是自治的 .....             | 169 |
| 8.3.7 服务是无状态的 .....            | 171 |
| 8.3.8 服务是可发现的 .....            | 172 |
| 8.4 面向服务原则如何相互关联 .....         | 173 |
| 8.4.1 服务复用性 .....              | 174 |
| 8.4.2 服务契约 .....               | 174 |
| 8.4.3 服务松散耦合 .....             | 175 |
| 8.4.4 服务抽象 .....               | 176 |
| 8.4.5 服务可组合性 .....             | 176 |
| 8.4.6 服务自治 .....               | 177 |
| 8.4.7 服务无状态 .....              | 177 |
| 8.4.8 服务可发现 .....              | 178 |
| 8.5 面向服务与面向对象 (第二部分) .....     | 179 |
| 8.6 原生 Web 服务对面向服务原则的支持 .....  | 180 |
| 第 9 章 服务层 .....                | 182 |
| 9.1 面向服务与当代 SOA .....          | 182 |
| 9.1.1 起源映射和具体 SOA 特征的支持源 ..... | 183 |
| 9.1.2 不支持的 SOA 特征 .....        | 184 |
| 9.2 服务层抽象 .....                | 185 |
| 9.3 应用服务层 .....                | 187 |
| 9.4 业务服务层 .....                | 188 |
| 9.5 编排服务层 .....                | 190 |

|                                                        |     |
|--------------------------------------------------------|-----|
| 9.6 无关服务 .....                                         | 191 |
| 9.7 服务层配置场景 .....                                      | 192 |
| 9.7.1 1 号场景: 仅有混合应用服务 .....                            | 192 |
| 9.7.2 2 号场景: 混合与工具应用服务 .....                           | 193 |
| 9.7.3 3 号场景: 以任务为核心的业务服务与工具应用服务 .....                  | 193 |
| 9.7.4 4 号场景: 以任务为核心的业务服务、以实体为核心的业务服务与工具应用服务 .....      | 194 |
| 9.7.5 5 号场景: 流程服务、混合应用服务与工具应用服务 .....                  | 194 |
| 9.7.6 6 号场景: 流程服务、以任务为核心的业务服务与工具应用服务 .....             | 194 |
| 9.7.7 7 号场景: 流程服务、以任务为核心的业务服务、以实体为核心的业务服务与工具应用服务 ..... | 194 |
| 9.7.8 8 号场景: 流程服务、以实体为核心的业务服务与工具应用服务 .....             | 195 |

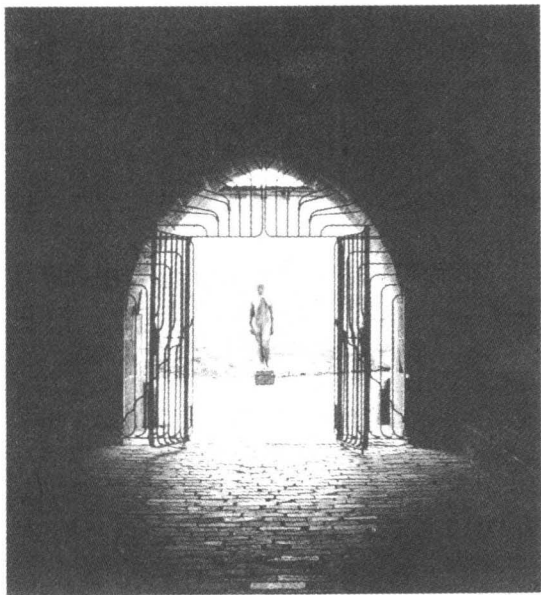
### 第四部分 构建 SOA (计划与分析)

|                            |     |
|----------------------------|-----|
| 第 10 章 SOA 交付策略 .....      | 198 |
| 10.1 SOA 交付周期的阶段 .....     | 198 |
| 10.1.1 SOA 交付周期的基本阶段 ..... | 198 |
| 10.1.2 面向服务的分析 .....       | 199 |
| 10.1.3 面向服务的设计 .....       | 199 |
| 10.1.4 服务的开发 .....         | 199 |
| 10.1.5 服务的测试 .....         | 199 |
| 10.1.6 服务的部署 .....         | 200 |
| 10.1.7 服务的管理 .....         | 200 |
| 10.1.8 SOA 的交付策略 .....     | 200 |
| 10.2 自顶向下策略 .....          | 201 |
| 10.2.1 过程 .....            | 201 |
| 10.2.2 利与弊 .....           | 202 |
| 10.3 自底向上策略 .....          | 202 |
| 10.3.1 过程 .....            | 203 |
| 10.3.2 利与弊 .....           | 203 |
| 10.4 敏捷策略 .....            | 204 |

|                                                |     |                                                |     |
|------------------------------------------------|-----|------------------------------------------------|-----|
| 10.4.1 过程                                      | 204 | 12.2.10 分类服务建模逻辑                               | 231 |
| 10.4.2 利与弊                                     | 206 | 12.2.11 分派适当的建模资源                              | 231 |
| 第 11 章 面向服务分析：介绍                               | 207 | 12.2.12 创建和发布业务服务建模<br>标准                      | 232 |
| 11.1 概述                                        | 207 | 12.3 服务模型逻辑分类                                  | 232 |
| 11.1.1 面向服务分析的目标                               | 207 | 12.3.1 SOE 模型                                  | 233 |
| 11.1.2 面向服务的分析流程                               | 208 | 12.3.2 企业业务模型                                  | 233 |
| 11.2 以业务为核心 SOA 的效益                            | 210 | 12.3.3 构件与服务模型                                 | 233 |
| 11.2.1 业务服务构建敏捷的业务<br>模型                       | 210 | 12.3.4 基本建模构件                                  | 234 |
| 11.2.2 业务服务是为编排所准备的<br>流程                      | 211 | 12.4 服务建模方法对比的示例                               | 235 |
| 11.2.3 业务服务促成复用                                | 211 |                                                |     |
| 11.2.4 只有业务服务才能实现面向<br>服务的企业                   | 211 | <b>第五部分 构建 SOA (技术与设计)</b>                     |     |
| 11.3 源于业务的服务                                   | 212 | 第 13 章 面向服务设计：介绍                               | 244 |
| 11.3.1 哪些业务服务能够作为源泉                            | 212 | 13.1 概述                                        | 244 |
| 11.3.2 业务服务的派生类型                               | 215 | 13.1.1 面向服务的设计目标                               | 244 |
| 11.3.3 业务服务与编排                                 | 217 | 13.1.2 “设计标准”与“行业标准”的<br>对比                    | 245 |
| 第 12 章 面向服务分析：服务建模                             | 218 | 13.1.3 面向服务的设计过程                               | 245 |
| 12.1 服务建模(循序渐进的过程)                             | 218 | 13.1.4 先决条件                                    | 245 |
| 12.1.1 服务与服务候选                                 | 218 | 13.2 WSDL 相关的 XML Schema 语言<br>基础              | 246 |
| 12.1.2 过程描述                                    | 219 | 13.2.1 schema 元素                               | 247 |
| 12.2 服务建模指导                                    | 228 | 13.2.2 element 元素                              | 247 |
| 12.2.1 考虑所封装逻辑潜在的跨流程<br>复用性(以任务为核心<br>的业务服务候选) | 228 | 13.2.3 complexType 与 simpleType<br>元素          | 248 |
| 12.2.2 考虑被封装逻辑(以任务为<br>核心的业务服务候选)<br>在流程内复用潜能  | 229 | 13.2.4 import 与 include 元素                     | 248 |
| 12.2.3 与流程相关的依赖因素(以任务<br>为核心的业务服务候选)           | 229 | 13.2.5 其他重要元素                                  | 248 |
| 12.2.4 跨应用复用的模式(应用服务<br>候选)                    | 229 | 13.3 WSDL 语言基础                                 | 249 |
| 12.2.5 对未来分解需求的思索                              | 230 | 13.3.1 definitions 元素                          | 249 |
| 12.2.6 以清晰的边界识别工作逻辑<br>单元                      | 230 | 13.3.2 types 元素                                | 250 |
| 12.2.7 避免逻辑界线蔓延                                | 230 | 13.3.3 message 与 part 元素                       | 251 |
| 12.2.8 不使用编排时的模拟流程服务<br>(以任务为核心的业务<br>服务候选)    | 230 | 13.3.4 portType、interface 与<br>operation 元素    | 251 |
| 12.2.9 锁定一个平衡的模式                               | 231 | 13.3.5 input 与 output 元素(与<br>operation 一起使用时) | 252 |
|                                                |     | 13.3.6 binding 元素                              | 252 |
|                                                |     | 13.3.7 input 与 output 元素(与<br>binding 一起使用时)   | 253 |
|                                                |     | 13.3.8 service、port 与 endpoint<br>元素           | 253 |

|                                       |     |                                                          |     |
|---------------------------------------|-----|----------------------------------------------------------|-----|
| 13.3.9 import 元素 .....                | 254 | (循序渐进的过程) .....                                          | 297 |
| 13.3.10 documentation 元素 .....        | 254 | 15.5 服务设计指导 .....                                        | 306 |
| 13.4 SOAP 语言基础 .....                  | 254 | 15.5.1 使用命名标准 .....                                      | 306 |
| 13.4.1 Envelope 元素 .....              | 255 | 15.5.2 应用适当接口粒度级别 .....                                  | 306 |
| 13.4.2 Header 元素 .....                | 255 | 15.5.3 将服务操作设计为天生可扩展 .....                               | 307 |
| 13.4.3 Body 元素 .....                  | 256 | 15.5.4 识别已知的和潜在的服务请求 .....                               | 308 |
| 13.4.4 Fault 元素 .....                 | 257 | 15.5.5 考虑使用模块化的 WSDL 文档 .....                            | 308 |
| 13.5 服务接口设计工具 .....                   | 257 | 15.5.6 谨慎使用命名空间 .....                                    | 308 |
| 13.5.1 自动生成 .....                     | 257 | 15.5.7 使用 SOAP 文档和文本属性值 .....                            | 309 |
| 13.5.2 设计工具 .....                     | 258 | 15.5.8 即使不需要遵守 WS-I 也可以<br>使用 WS-I 概要 .....              | 310 |
| 13.5.3 手工编码 .....                     | 258 | 15.5.9 有元数据的文档服务 .....                                   | 310 |
| 第 14 章 面向服务设计: SOA 组合指导<br>原则 .....   | 260 | 第 16 章 面向服务设计: 业务流程<br>设计 .....                          | 312 |
| 14.1 组合 SOA 的步骤 .....                 | 260 | 16.1 WS-BPEL 语言基础 .....                                  | 312 |
| 14.1.1 第一步: 选择服务层 .....               | 261 | 16.1.1 BPEL 4WS 与 WS-BPEL 简史 .....                       | 313 |
| 14.1.2 第二步: 定位核心标准 .....              | 261 | 16.1.2 先决条件 .....                                        | 313 |
| 14.1.3 第三步: 选择 SOA 扩展 .....           | 261 | 16.1.3 process 元素 .....                                  | 313 |
| 14.2 选择服务层的考虑 .....                   | 262 | 16.1.4 partnerLinks 与 partnerLink<br>元素 .....            | 314 |
| 14.3 定位核心 SOA 标准的考虑 .....             | 263 | 16.1.5 partnerLinkType 元素 .....                          | 315 |
| 14.3.1 行业标准与 SOA .....                | 263 | 16.1.6 variables 元素 .....                                | 315 |
| 14.3.2 XML 与 SOA .....                | 264 | 16.1.7 getVariableProperty 与<br>getVariableData 函数 ..... | 316 |
| 14.3.3 WS-I 基本概要 .....                | 264 | 16.1.8 sequence 元素 .....                                 | 316 |
| 14.3.4 WSDL 与 SOA .....               | 265 | 16.1.9 invoke 元素 .....                                   | 317 |
| 14.3.5 XML Schema 与 SOA .....         | 265 | 16.1.10 receive 元素 .....                                 | 317 |
| 14.3.6 SOAP 与 SOA .....               | 265 | 16.1.11 reply 元素 .....                                   | 318 |
| 14.3.7 命名空间与 SOA .....                | 266 | 16.1.12 switch、case 与 otherwise<br>元素 .....              | 318 |
| 14.3.8 UDDI 与 SOA .....               | 266 | 16.1.13 assign、copy、from 与 to 元素 .....                   | 319 |
| 14.4 选择 SOA 扩展的考虑 .....               | 267 | 16.1.14 faultHandlers、catch 与<br>catchAll 元素 .....       | 319 |
| 14.4.1 选择 SOA 特征 .....                | 267 | 16.1.15 其他 WS-BPEL 元素 .....                              | 319 |
| 14.4.2 选择 WS-* 规范 .....               | 268 | 16.2 WS-协调概览 .....                                       | 320 |
| 14.4.3 WS-BPEL 与 SOA .....            | 268 | 16.2.1 CoordinationContext 元素 .....                      | 321 |
| 第 15 章 面向服务设计: 服务设计 .....             | 271 | 16.2.2 Identifier 与 Expires 元素 .....                     | 321 |
| 15.1 概述 .....                         | 271 | 16.2.3 CoordinationType 元素 .....                         | 322 |
| 15.1.1 设计标准 .....                     | 272 | 16.2.4 RegistrationService 元素 .....                      | 322 |
| 15.1.2 过程描述 .....                     | 272 | 16.2.5 指定 WS-业务活动的协作类型 .....                             | 322 |
| 15.1.3 先决条件 .....                     | 272 | 16.2.6 指定 WS-原子事务的协作类型 .....                             | 322 |
| 15.2 以实体为核心的业务服务设计<br>(循序渐进的过程) ..... | 273 |                                                          |     |
| 15.3 应用服务设计(循序渐进的<br>过程) .....        | 286 |                                                          |     |
| 15.4 以任务为核心的业务服务设计                    |     |                                                          |     |

|                                                                         |     |
|-------------------------------------------------------------------------|-----|
| 16.3 面向服务的业务流程设计<br>(循序渐进的过程) .....                                     | 322 |
| 第 17 章 基础 WS-* 扩展 .....                                                 | 340 |
| 17.1 WS-寻址语言基础 .....                                                    | 340 |
| 17.1.1 EndpointReference 元素 .....                                       | 341 |
| 17.1.2 消息信息报头元素 .....                                                   | 342 |
| 17.1.3 WS-寻址的复用性 .....                                                  | 344 |
| 17.2 WS-可靠消息传递语言基础 .....                                                | 345 |
| 17.2.1 Sequence、MessageNumber 与<br>LastMessage 元素 .....                 | 346 |
| 17.2.2 SequenceAcknowledgement 与<br>AcknowledgementRange 元素 .....       | 347 |
| 17.2.3 Nack 元素 .....                                                    | 347 |
| 17.2.4 AckRequested 元素 .....                                            | 348 |
| 17.2.5 其他 WS-可靠消息传递元素 .....                                             | 349 |
| 17.3 WS-策略语言基础 .....                                                    | 349 |
| 17.3.1 Policy 元素与通用策略断言 .....                                           | 350 |
| 17.3.2 ExactlyOne 元素 .....                                              | 350 |
| 17.3.3 All 元素 .....                                                     | 351 |
| 17.3.4 Usage 属性 .....                                                   | 352 |
| 17.3.5 Preference 属性 .....                                              | 352 |
| 17.3.6 PolicyReference 元素 .....                                         | 352 |
| 17.3.7 PolicyURIs 属性 .....                                              | 352 |
| 17.3.8 PolicyAttachment 元素 .....                                        | 353 |
| 17.3.9 策略断言的附加类型 .....                                                  | 353 |
| 17.4 WS-元数据交换语言基础 .....                                                 | 354 |
| 17.4.1 GetMetadata 元素 .....                                             | 354 |
| 17.4.2 Dialect 元素 .....                                                 | 355 |
| 17.4.3 Identifier 元素 .....                                              | 356 |
| 17.4.4 Metadata、MetadataSection 与<br>MetadataReference 元素 .....         | 356 |
| 17.4.5 获得消息 .....                                                       | 357 |
| 17.5 WS-安全语言基础 .....                                                    | 358 |
| 17.5.1 Security 元素 (WS-安全) .....                                        | 358 |
| 17.5.2 UsernameToken、Username 与<br>Password 元素 (WS-安全) .....            | 359 |
| 17.5.3 BinarySecurityToken 元素<br>(WS-安全) .....                          | 359 |
| 17.5.4 SecurityTokenReference 元素<br>(WS-安全) .....                       | 359 |
| 17.5.5 编写 Security 元素内容<br>(WS-安全) .....                                | 359 |
| 17.5.6 EncryptedData 元素<br>(XML-加密) .....                               | 360 |
| 17.5.7 CipherData、CipherValue 与<br>CipherReference 元素<br>(XML-加密) ..... | 360 |
| 17.5.8 XML-签名元素 .....                                                   | 361 |
| 第 18 章 SOA 平台 .....                                                     | 363 |
| 18.1 SOA 平台基础 .....                                                     | 363 |
| 18.1.1 基本的平台构件 .....                                                    | 363 |
| 18.1.2 通用 SOA 平台层 .....                                                 | 364 |
| 18.1.3 SOA 层与技术之间的关系 .....                                              | 365 |
| 18.1.4 基础服务技术架构 .....                                                   | 365 |
| 18.1.5 厂商平台 .....                                                       | 372 |
| 18.2 J2EE 中的 SOA 支持 .....                                               | 372 |
| 18.2.1 平台概览 .....                                                       | 372 |
| 18.2.2 基本 SOA 支持 .....                                                  | 379 |
| 18.2.3 面向服务原则的支持 .....                                                  | 380 |
| 18.2.4 当代 SOA 支持 .....                                                  | 381 |
| 18.3 .NET 中的 SOA 支持 .....                                               | 383 |
| 18.3.1 平台概览 .....                                                       | 383 |
| 18.3.2 基本 SOA 支持 .....                                                  | 389 |
| 18.3.3 面向服务原则的支持 .....                                                  | 389 |
| 18.3.4 当代 SOA 支持 .....                                                  | 390 |
| 18.4 集成性考虑 .....                                                        | 392 |
| 附录 A 案例研究: 总结 .....                                                     | 394 |
| 附录 B 服务模型参考 .....                                                       | 399 |



# 第 1 章

## 引 言

### 1.1 本书的重要性

我很喜欢引用在一次研讨会上准备演讲时无意听到的一段对话。当时两位 IT 专家正在讨论各自的环境,其中一个人问另一个人,他们的团队是否正在构建面向服务架构。得到的回答是:“我们的架构师认为它是面向服务的,开发者坚持它是面向对象的,而分析师却希望它能更面向业务。我所能告诉你的是,在我们开始构建 Web 服务之前它什么也不是。”

这种坦率的陈述是一个时代的信号。面向服务架构(Service-Oriented Architecture, SOA)已成为 IT 行业的焦点,然而人们对它却缺乏充分的理解。本书旨在填补这个知识空缺,以帮助你达到以下目标:

- 理解 SOA、面向服务和 Web 服务。
- 学习如何用 Web 服务构建 SOA。

我们先从识别采用 SOA 时所遇到的最常见障碍开始介绍。

#### 1.1.1 虚假的 SOA

我想不出还有哪个术语比“面向服务”更含糊。这种含糊导致厂商、IT 专家和媒体各执一词,当然,这使得人们更难理解那些贴上“面向服务”标签的技术架构。

SOA 作为一个抽象模式,过去代表了一种不涉及具体实现的基础分布式架构。但是相对而言,这个模型仅代表 SOA 最常见的当代形式中的一个子集。

与 Web 服务和一系列公认的面向服务原则一起,SOA 已发展为与其前辈截然不同的架构平台。它引入了由精选概念所支持的新概念,明显增强了传统的分布式计算平台的特征,从而导致了面向服务环境需要经常重新定义 IT 基础设施的面向服务环境。

当代 SOA 的多变性已经引起广泛关注。它被宣传为变革企业环境的平台能力,这有赖于在 Web 服务技术的进步、组织联盟及敏捷性的加强,以及对跨平台协调性的追求之间进行权衡。

多数观点认为,面向服务的技术架构只是简单地由 Web 服务组成。这是个普遍而危险的假设,它会导致计划采用 SOA 的组织犯下弥天大错,即认为只要对 Web 服务平台深入投资就能获得当前主流 SOA 所