

执行SOA

SOA 实践指南

Executing SOA

A Practical Guide for the Service-Oriented Architect

(美) Norbert Bieberstein 等著
王海鹏 等译



机械工业出版社
China Machine Press

developerWorks®

执行SOA

SOA 实践指南

Executing SOA

A Practical Guide for the Service-Oriented Architect

(美) Norbert Biebers

王海鹏 等译



机械工业出版社
China Machine Press

本书细致全面地描述了 SOA，从架构视图直到实际的实现，展示了如何克服成功实施 SOA 的关键障碍，并确定了针对所有方面的最佳实践。本书关注的问题包括：引入服务原则，支持协作和信息过程共享；利用已有的技术资产和策略来集成服务；为新的工具选择正确的角色；文化、治理和架构方面的转变；为整个组织机构的生命周期带来更大的敏捷性。

本书内容详实，结构清晰，可作为管理与技术人员的参考用书。

Simplified Chinese edition copyright © 2009 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Executing SOA: A Practical Guide for the Service-Oriented Architect, 1E* (ISBN 978-0-13-235374-8) by International Business Machines Corporation, Copyright © 2008.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as IBM, 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：01-2009-1360

图书在版编目（CIP）数据

执行 SOA：SOA 实践指南 /（美）拜伯斯汀（Bieberstein, N.）等著；王海鹏等译. —北京：机械工业出版社，2009.5

（SOA 技术丛书）

书名原文：Executing SOA: A Practical Guide for the Service-Oriented Architect, 1E

ISBN 978-7-111-26682-2

I. 执… II. ①拜… ②王… III. 互联网络—网络服务器—指南 IV. TP368.5-62

中国版本图书馆 CIP 数据核字（2009）第 045175 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：陈佳媛

北京京北印刷有限公司印刷

2009 年 5 月第 1 版第 1 次印刷

186mm × 240mm · 12.25 印张

标准书号：ISBN 978-7-111-26682-2

定价：35.00 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

译者序

几年前，为了尝试 JDK 1.5 中的并发包，我写了一个多线程的网页爬虫程序，利用线程池来抓取和分析页面。

并发 200 个线程，每个线程从待爬 URL 队列中取得一个 URL，取回网页，进行分析，找出其中的 URL 链接，再放入待爬队列。开发过程很正常，但在测试中遇到了问题。在爬了 7 万多个网页之后，程序开始越来越慢。凭感觉判断，有一些线程“死”掉了。

多线程的调试并不是件容易的事。这个问题很“难”再现。这不是普通意义上的难再现，它每次都会出现。但要跑到 7 万多 URL 时，才会出现。也就是说，再现这个问题的代价很大。我试过将线程池的大小退化到 1，想找出什么样的 URL 会导致线程死掉，但是行不通，因为速度太慢。当时的 IDE 也缺乏对多线程调试的一些支持。而且即便有支持，可能也不太适合这种情况。后来因为种种原因，那个程序就不了了之了。

这本书中 SOA 治理的思想给了我一些启发：我们需要关注服务执行的健康状况，包括服务执行的时间。例如，我们可以进行这样的改动：

在每个线程领取 URL 时，记录一个时间戳。在它完成这个 URL 处理时，再记录一个时间戳。再利用一个线程，对未完成的 URL 定时检查它的健康程度。如果在很长的一段时间内它还没完成，那么它就有问题。这样我们可以找到嫌疑 URL。我们可以对这种 URL 单独测试，看看是否因为程序的原因，不能处理这样的 URL。或者，我们可以把对应的线程任务杀掉，直接跳过这些有问题的 URL。

如果您和我一样，是一名开发人员，学习一些 SOA 的思想是很有帮助的。我们可以在程序中设计一些机制，支持运营维护和故障分析，这正是 SOA 的一部分内容。

IT 运维部门需要 SOA。业务部门需要 SOA。企业高层需要 SOA。设想一家经营固话业务的电信公司，通过兼并和重组，拿到了一个移动网络。公司最需要的是什么？就是 SOA。

这个移动网络上跑着多少应用？多少中间件？多少数据库？多少操作系统？多少

服务器？它们的使用状况如何？它们由谁提供技术支持？它们是什么配置和版本？它们有哪些参数可以调整？它们支持着怎样的业务流程？它们支持着怎样的业务数据模型？它们提供怎样的 QoS？它们在安全性和可伸缩性方面存在哪些风险？

SOA 参考框架帮助我们提出这些问题。提出问题比解决问题更重要，真的。企业应该认真考虑向 SOA 迁移。

参加本书翻译工作的人员除封面署名外还有：王海燕、李国安、周建鸣、范俊、张海洲、谢伟奇、林冀、钱立强、甘莉萍。在这本书的翻译过程中，我学到了很多，因此郑重地向大家推荐它。如果这本书对于您有所帮助，我将十分高兴。

王海鹏

己丑年春日于上海

序

面向服务的架构（Service-Oriented Architecture, SOA）已经不是新东西了。实际上，随着技术成熟曲线的发展，它经历了一些紧缩和倒退，有了不少失败尝试的案例。为什么会这样？如果这个方面是如此紧迫，为什么一些人转向怀疑和公然反对？主要原因在于我们都没有理解这种转变的困难，在业务和 IT 基础设施方面，它都需要训练有素的人员、深刻的理解和积极的行动。在协作和跨团队的过程中，我们需要训练有素的人员，这样我们才能够进行更大范围的、关于组织结构的思考，而不只是个人的、局部的“快速修复”或局部应用。因此，许多计划尝试失败的原因是多种多样的：

- 未能建立有效的治理，或者甚至未能意识到这种治理必须改变现状，通过交付共享的、有用的和有效的服务来获得持久的好处。
- 试图在组织机构的独立部门中引入服务的原则和方法，而没有大幅度地改变共享的文化和信息处理的过程。这种疏忽绝对会导致失败。
- 未能理解如何将有关服务的新思维与原有的技术资产和方向进行集成。
- 相信单项技术或工具将实现期望的结果。
- 相信速度来自敏捷，而敏捷则简单等同于“快速交付”独立的、互不相干的项目，没有考虑到组织机构范围的生命周期。这也被称为无结构的混沌。

在我看来，追求 SOA 的基本动机或理由在今天尤为普遍，超过以前任何时候。通过更快的速度和创新的业务模式来进行竞争的压力没有改变。对于企业、生态系统和供应链全球化的压力正在加剧。依赖于 IT 系统来扩张业务的需求持续增长，技术遍及商业和日常生活的方方面面。

在所有业务模式的转变中，许多扩张问题都来自于因特网所提供的途径和选择，因特网正变得越来越无所不在。因特网正在改变和演进。带宽不断增长，导致的网络效应正为新的公司和业务模式提供机会，同时也对已有的公司和业务模式形成破坏性的冲击。我们只要看看音乐行业中发生的例子（实际上是整个娱乐行业）。电信服务提供商（TSP）是典型的例子，其中行业的垂直集成和账单模式已经被基于 IT 的服务和设备的引入所击碎。两者的交集（娱乐业和 TSP）变化更是戏剧性的，并且对消费电

子、汽车等行业产生了影响。不断增长的选择和由此导致的全球供应链、供应源的转变对一切都产生了影响，从制造的产品，到知识产权，再到人才。没人能置身事外。

IT 系统必须支持或引领这种潮流，否则，与之相联系的业务将在竞争中失败，沦为淘汰者。支持灵活性和规模伸缩的关键就在于面向服务的架构原则。前面曾提到，SOA 中没有魔法。在很大程度上，它是古老的“模块化”和结构化分解概念的演进版本，强调在全球化的、开放的环境中更好地实现 IT 和业务的融合。但是，要在因特网上成功部署基于服务的架构，实现模块化和共享，真正支持全球化的业务，一些基本的规模伸缩问题以前还没有人尝试过。

我对这个伸缩性问题的体验来自于在 IBM 中转变 Rational。我曾经尝试利用 SOA 来影响 Rational 的业务和 Rational 的技术中文化和技术的演进。Jazz 的方向和架构是这种策略转变的主要体现，这种策略不仅源自于基于因特网的架构，也源自于对开发工具和平台底层的业务和技术的重新思考。这让我们开始创造新的产品，从已有的产品演进，并利用新的业务模式（定价和打包）在交付客户价值方面的优点。正如本书所介绍的，这要求文化的转变，治理模型的转变，涉及组织机构、技术假定、信息架构、协作服务等。这要求我们重新思考软件开发的工作，将它建模为一系列的业务过程，这些业务过程需要建立起来，并通过动态的测量指标监控服务来实现报告，这些服务可以放在全球分布的内部网上。现在说我们已经将愿景变成现实还为时尚早，但我比以前任何时候更坚信，这样做是对的。

出于这种精神，我推荐您阅读这本书，进一步了解我们和客户在过去 4 年中各种基于 SOA 的工作经验所带来的好处。前一本书展示了舞台，带给我们许多关键的问题和思考。这一本书深入探讨了治理、服务管理，特别是生命周期的挑战等关键问题。它关注了我一开始提到的许多失败原因。作者们经验丰富，而且提炼出了最佳实践，为实现成功的 SOA 转变提供了实际可行的、深思熟虑的路线图。好好享受这本书。毕竟，假如事情很容易，那么每个人都能做好！

Daniel Sabbah
GM, Rational Software
IBM Software Group

致 谢

我们要感谢 IBM 的管理团队，允许我们花时间来编写这本书。所有在 IBM 写的书都得到了管理层的支持。我们感激这种支持，以及允许我们在编写本书时访问必需的资源。谢谢 IBM Software Group 和 IBM Global Services 的直接支持主管，特别是 IBM 全球咨询服务的总经理 Robert LeBlanc。

我们要感谢 IBM Press 和 Pearson Education 的工作者，这些出版人和员工帮助完成市场方面的工作，并完成了这本书的出版工作。我们要感谢 IBM Press 的 Tara Woodman 和她的团队。我们要感谢 Pearson Education 的 Greg Wiegand，他在选题阶段给予我们帮助。我们也要感谢 Pearson Education 的 Katherine Bull，她是高级编辑，让我们著作能符合标准。我们要感谢开发编辑 Ginny Bess Munroe 和拷贝编辑 Keith Cline，他们对可理解性提出建议，帮助我们更好地表达我们的想法。我们要感谢 Sue Outtersson，她提供了技术审读，指出了不一致的地方。我们也要感谢 Pearson 的产品和市场团队，他们的帮助让这本书变成现实。

我们还想感谢所有那些令人吃惊的才华横溢的人们，他们在 IBM developerWorks (<http://www.ibm.com/developerWorks>) and IBM Redbooks (<http://www.ibm.com/redbooks>) 上发表了大量的文章，让公众分享。在第 5 章“利用可复用的资产”为例，它包括了这些人的贡献：Alan Brown、Mikko Kontio、Dr. Tracy Gardner、Larry Yusuf、John Lord、Eoin Lane、Clive Gee、John Medicke、Feng-Wei Chen、Margie Mago、Scott Linehan、Kevin Williams、John Ganci、Amit Acharya、Jonathan Adams、Paula Diaz de Eusebio、Gurdeep Rahi、Diane Strachan、Kanako Utsumi、Noritoshi Washio 和 Grant Larsen。

最后，我们想感谢对本书的各章做出贡献的许多人。Clive Gee 来自 IBM UK，他在治理方面与客户合作了多年。他对什么有效、什么有意义的深刻见解，让我们的 SOA 治理和 SOA 组织部分的内容受益非浅。Randy Iangel 来自 IBM U.S.，告诉了我们关于 SOA 治理的业务方面和 SOA 治理的子集方面的内容。Bruce Hawken 来自 IBM Australia，展示了他的洞见和专注，他让 SOA 治理成为现实并令人鼓舞。我们感谢他们 3 位对 SOA 治理部分的贡献。我们也要感谢 Dr. Ali Arsanjani，他是 IBM 杰出工程

师，在过去数年中，与我们多次讨论了 SOMA，我们在这个主题上获益良多。我们也要感谢 Patrick Haren，IBM 执行架构师，他的关键复查和建议让我们最终完成了第 4 章的工作。我们感谢 Marc Fiammante，他是 IBM 杰出工程师，不管多忙，他总是抽出时间来，为我们提供急需的意见和指导。我们感谢 Rosalind Radcliffe，来自 IBM Tivoli 的 STSM，他为 SOA 架构的产品提供了关键的意见。我们感谢 Sankar Singha，他是 IBM 高级架构师，他帮助我们将第 6 章的内容组织在一起。最后，我们要感谢 Thomas Schaeck，他是 IBM 杰出工程师，参与开发了 Lotus Quickr 和 WebSphere Portal Web 2.0。他在协作式解决方案方面提供了深刻见解，这种解决方案能够扩展基于 SOA 的 IT 在企业中的好处。

作者简介

Norbert Bieberstein 为 IBM 的 SOA 高级技术部门工作，在全球范围内发布和沟通 SOA 相关的主题。他从客户项目那里取得了第一手的经验，这些客户项目属于不同行业，正努力向基于 SOA 的解决方案迁移。Norbert 发表了一些关于 SOA 的文章，与 IBM Systems Journal issue 44-4 在 SOA 方面进行合作，他也是《Service-Oriented Architecture Compass》(IBM Press, 2005) 一书的第一作者。在 2007 年，“SOA for Profit”一书出版，Norbert 是三位作者之一。他还参与编写了两本 IBM 红皮书：“Introduction to Grid Computing with Globus”和“Enabling Applications for Grid Computing with Globus”。在 1993 年，他出版了自己的第一本书“CASE-Tools”。1989 年，Norbert 作为一名软件工程顾问加入了 IBM 软件开发实验室。他在信息技术和计算机科学方面的经验超过 27 年。在他的职业生涯中，曾经在一个较小的软件供应商那里担任应用开发者，也曾在 Aachen University of Technology (RWTH) 担任科学计算程序员，他在这所大学取得了数学和地理学硕士学位。在 2006 年，他从英国 Henley 的 Henley Management College 的一个合作 MBA 课程毕业。

Robert G. Laird 是 IBM 的 IT 架构师，属于 SOA 高级技术组，自 2006 年 5 月以来，为 IBM 全球范围的客户提供咨询，主要领域是 SOA 治理和 SOA 架构。他是行业 TO-GAF (The Open Group Architecture Framework) SOA Governance 工作组的成员。

Robert 有超过 20 年电信行业的经验，曾在 MCI 和 Verizon Business 工作。他曾是 MCI 的首席架构师，领导企业的架构师团队并负责整个从订单到现金的应用套件。他领域开发了基于 SOA 的单栈策略，以简化多个网络和多个应用库。Rob 推动了 MCI 的产品开发的决策、计算和执行，领域涉及合同中心、IP/VPN、VoIP、IM 和受管理的服务。对于 OSS，他曾成功领导并实现了自动化的网络提供、网络恢复和网络管理。

在加入 MCI 之前，Robert 曾在 American Management Systems (AMS) 和 Ideation, Inc 担任咨询师。他的计算科学硕士学和学士学位是在 Purdue 大学获得的，他在电信领域拥有 2 项专利。他曾在不同的行业论坛发表演讲，为 SOA Magazine 撰写文章，并被 CIO Insight、Telecommunications、InfoWorld 和 Computerworld 引用。

Dr. Keith Jones 目前是 IBM 执行 IT 架构师，属于 SOA 高级技术组。主要关注为前沿客户提供面向服务的架构和实现。他在 IT 行业有 30 年的经验，曾担任系统工程师、软件架构师、战略制定者，也是多本中间件图书的作者。Keith 的主要兴趣集中在构建支持事务的、面向消息和面向服务的中间件基础设施，在各种企业环境中支持业务过程。最近，他的工作涉及在主要金融服务、零售服务、汽车制造、在线媒体和拍卖公司创建基础设施。Keith 拥有化学博士学位，和他的家人一起住在美国科罗拉多州的 Boulder。

Tilak Mitra 是高级认证执行 IT 架构师，属于 IBM 全球服务部。他为 IBM 在全球提供企业架构方面的咨询，帮助客户实现 SOA 战略，从愿景直到设计和实现。

Tilak 在零售、银行、媒体和娱乐、健康和交通行业有超过 10 年的行业经验，曾担任各种领导职位，工作内容涉及业务和 IT 转换，包括解决方案实现和交付。Tilak 与 IBM SOA Center of Excellence 密切合作。在合作中，他对许多 IBM SOA 解决方案做出了贡献，参与撰写了白皮书和技术文章，这些文章成为了 IBM SOA 资源。他目前关注的是建立一些资产和技术，支持快速简单地开发基于 SOA 的复合应用，并能够在不同供应商的平台上执行（如 IBM WebSphere 和 SAP NetWeaver）。

Tilak 拥有印度 Indian Institute of Science (IISc) 的电子工程学硕士学位和 Presidency College 的物理学学士学位。他是 Java Developers Journal (JDJ) 的特约记者，并且经常为 IBM developerWorks、JDJ 和 WebSphere Developer's Journal 编写文章。他还在美国多所大学演讲，介绍 SOA 各方面的内容。

对本书的评价

SOA 是一种架构风格选择。但是，实现 SOA 会带来一种雪崩效应，其中一些后果目前还没有发现。通向 SOA 的方法不只一种。我们必须在做中学，积累经验和最佳实践。这本书对应用 SOA 的后果提供了有价值的深刻见解。它基于经验，对完整的服务生命周期提供了方法、原则和指导。这本书是每个企业架构师必备的。

Martin van den Berg

Lead Enterprise Architect

Sogeti Netherlands

现在许多关于面向服务的架构（SOA）的出版物只关注技术和工具。本书提供了更广阔的视图。针对 SOA 对业务的影响、SOA 的治理和 SOA 架构师的方法，都分别提供了整章的内容。

一些精彩观点包括：

1. SOA 首先关注的是业务敏捷性，而非 IT。
2. SOA 是一种架构风格，并不等同于 Web 服务这样的技术。
3. SOA 的原则并不新，并不是和这个名词一起出现的。

我们需要更多类似这样的书籍！

Prof. Dr. Bernhard Humm

全面而实用的一本书。它深刻地描述了 SOA 治理，给出了服务的全面视图，从架构视图直到实际的实现，我觉得这一点对我很有价值。对于准备面对 SOA 中艰难部分的企业架构师来说，这本书是很有用的。关于如何实现资产复用、SOA 中人员的方面、描述如何适用工具的那些章节，让这本书很值得一读，而且很实用。通过对其他材料的扩展引用，它也是为进一步阅读提供了向导。

Erik van Ommeren

Director of Innovation

Sogeti USA LLC / VINT

Washington D. C. , USA

目 录

译者序	
序	
致 谢	
作者简介	
对本书的评价	

第1章 SOA 简介	1
1.1 SOA 回顾	1
1.2 要考虑的新问题	4
1.3 这本书有何不同	5
1.4 这本书写给谁	6
1.5 这本书包含哪些内容	6
1.6 developerWorks 的文章链接	7
1.7 参考资料	8
第2章 揭示好处	9
2.1 为什么业务部门应该 关心 SOA	9
2.2 架构	13
2.3 聚焦业务架构	14
2.4 业务过程	15
2.5 业务组件	16
2.6 揭开面纱	17
2.7 developerWorks 的文章链接	18
2.8 参考资料	18

第3章 SOA 治理	19
3.1 SOA 战略的治理	21
3.2 针对 SOA 进行组织	29
3.3 SOA 治理的考虑	32
3.4 小结	47
3.5 developerWorks 的文章链接	47
3.6 参考资料	47
第4章 服务建模和设计方 法学	49
4.1 一个 SOA 参考架构	50
4.2 面向服务的建模和架构	54
4.3 小结	70
4.4 developerWorks 的文章链接	70
4.5 参考资料	71
第5章 利用可复用的资产	72
5.1 什么是资产	72
5.2 服务复用	72
5.3 什么让 SOA 服务变得可复用	73
5.4 复用模式	75
5.5 让遗留系统可复用：从遗留的 整体应用中获取可复用组件	79
5.6 小结	86
5.7 developerWorks 的文章链接	86
5.8 参考资料	87

第6章 服务实现	89	8.2 Web 2.0 与 SOA	151
6.1 实现 SOA 生命周期	89	8.3 构建 SOA 协作环境	161
6.2 SOA 中的建模前活动	90	8.4 从 SOA 到企业运营的好处	164
6.3 SOA 中的服务建模	93	8.5 小结	167
6.4 SOA 中的服务组装	105	8.6 developerWorks 的文章链接	168
6.5 SOA 中的服务部署	108	8.7 参考资料	169
6.6 SOA 中的服务管理	110	第9章 SOA 的未来	171
6.7 SOA 编程模型	117	9.1 复合业务服务与复合应用 程序	171
6.8 架构和设计考虑	122	9.2 行业模型标准化与 行业范围内支持 SOA	172
6.9 小结	126	9.3 打包的应用程序转变为单点 解决方案	172
6.10 developerWorks 的文章链接 ...	127	9.4 SOA 和 EDA 的杂合架构 方法	173
6.11 参考资料	128	9.5 SOA 方法学演进	174
第7章 信息服务	131	9.6 业务过程和 SOA 涉及的人	174
7.1 数据和信息服务	131	9.7 SOA 测量指标	175
7.2 数据, SOA 与松耦合	132	9.8 企业中无所不在的 SOA	175
7.3 从数据源到消费者	133	9.9 全球使用 SOA	176
7.4 数据质量	136	9.10 SOA 打开了业余软件服务 市场	177
7.5 数据处理	137	9.11 小结	177
7.6 数据服务提供者的逻辑模式 ...	138	9.12 developerWorks 的文章链接 ...	178
7.7 复合服务逻辑	143	9.13 参考资料	178
7.8 语义互操作性	143		
7.9 小结	145		
7.10 developerWorks 的文章链接 ...	145		
7.11 参考资料	146		
第8章 在 SOA 下协作	147		
8.1 SOA 对人们意味着什么	147		

第 1 章 SOA 简介

“又是一本面向服务的架构（SOA）的书”您可能会这样想。书店里已经有几百本这样的书在卖了。在 4 年的成功应用之后，SOA 已经形成了强大的市场价值，您几乎可以买到任何“基于 SOA 的某种产品”。市场人员很快发现了这种强烈的趋势，所以将产品重新命名或描述为 SOA 兼容的、基于 SOA 的、为 SOA 而生的，以及一切与 SOA 有关的。虽然出了大量介绍 SOA 的书，但有一些问题仍没有涉及。所以，在这本书中，我们将讨论这些“漏掉的问题”。

SOA 的原则不是新的，不是和这个缩写同时发明的，很多供应商“觉得很有理由”宣称他们的产品是基于 SOA 的。当然，如果您检查一下 IT 解决方案，您会发现 SOA 的原则在几十年以前就实现了。例如，在一些财务服务公司的 IT 部门内部开发的基于大主机的解决方案，已经很聪明地考虑到了将来的复用和变更要求，以一种松耦合的方式来构建。在某些情况下，已经采用了我们今天所谓的企业服务总线（ESB）的架构。这些单元虽然没有以这样的方式来标识，但它们确实是按 SOA 的要求来运行的。架构原则不是最近才发明的，您可以认为它们是 SOA 的基础。

在我们深入细节之前，让我们进一步了解一下 SOA 的历史。SOA 是如何产生的？这个问题的答案将很快揭示成功执行 SOA 的关键要素。但是，只了解历史是不够的，早期基于服务的架构要向 SOA 转变，还必须考虑一些新的问题。

1.1 SOA 回顾

SOA 的大量采用可能源自于它在 IT 和业务之间提供的联系。它承诺在两个阵营之间的断层上架起一座桥梁。企业中的相关各方希望打破围墙，不再是业务部门将需求抛过围墙，然后 IT 部门在围墙后面试图弄明白业务人员想要什么。

在 20 世纪五六十年代，当 IT 还是新事物时，接触它的人是一些专家，他们说着神秘的语言，知道如何操作神奇的业务机器。这些机器（以及它们的第一个应用程序）能够更快地计算出利润/税款，并使某些业务过程自动化。一组高薪的专家可以很容易

地了解简单的业务需求，然后将它们实现为在这些机器上运行的软件。随着个人计算机的兴起，人们知道了如何进行编程，而不只是在绿屏幕的某个位置输入数据，这形成了一个更精通计算机的用户群体，他们知道如何对 IT 部门提出更复杂的需求。在 20 世纪 70 年代，越来越多的软件供应商在市场上出现，为许多行业提供操作自动化的解决方案。

市场让企业能够比较和评估不同的软件解决方案，通常是针对他们自己内部的 IT 部门。于是，公司开始寻求 IT 外包，因为 IT 变得过于复杂，内部的团队不能够控制那些没能实现的需求，而这些需求需要马上实现，以击败竞争对手。战略外包成为 IT 行业的一个重要市场。实际上，通常整个 IT 部门会被 IT 服务公司收购，由 IT 服务公司来提供一定水平的服务，并承诺消除原有的所有 IT “壁垒”。“关注于核心业务”很快变成了今天的格言。

让“专家”来接管 IT，可以确保服务水平、升级过程、条件和要求等多方面需求。除此之外，这些外包的服务最终比内部 IT 部门所支持的企业日常业务要走得更远。服务提供者在客户的要求和竞争压力下，需要提供尽可能低的费用以获取长期的合同，从而确保稳定的收入。实现这一点的途径之一就是，服务提供者实现标准化，并复用已有的 IT 架构和解决方案。

在 20 世纪 90 年代，兼并和收购的规模越来越大，所以一些不兼容的 IT 解决方案需要进行集成。例如，这些扩大了的公司必须整合各种资产和要求，运营成为一个问题。标准化的软件包代表了这个问题的一种解决方案，但是，原有的企业资源计划（ERP）系统已经在很大程度上进行了定制，在某些情况下，ERP 解决方案提供者已经被他们的竞争对手兼并，需要集成他们自己的包。有时候，软件供应商针对某个行业的特定运营单位的特殊需求，发布不同产品线的软件解决方案，这些解决方案一开始并没有考虑到集成。这些系统缺少作为广泛接受的标准行业业务模型，使共享成为困难。

这些所谓的“坑子解决方案”，每一个都针对具体的业务主题和数据集（以及在数据上的相应操作功能）。在不同企业单元的独立业务考虑的驱使下，这些“坑子解决方案”成为了在每个企业单元内实现操作优化意识的范例。由于优化业务最有效的方式就是关注于业务单元的内部操作，总体大局观就丧失了。

前面我们曾提到，在企业单元之间进行集成的一种驱动力量是兼并和收购，结果导致企业需要通过复用一些企业单元来实现预想的成本降低，如 HR、发票、采购和销售管理。最初的尝试是有重复的部门之间创建集成自动化，这样用户可以对同样的数

据只输入一次,就适用于不同的系统。或者,利用一个批处理过程从一处系统向另一个系统传输信息。企业应用集成(EAI)系统似乎为建立重复应用之间的点到点的集成提供了成熟的支持。

为了让集成自动化,人们开发出不同的中间语言和适配器;通常,标准软件提供商提供一些应用编程接口(API),目的是更好地将它们的解决方案与其他不同目的的方案进行集成。这些最初的尝试将水平的信息流和贯穿企业的业务过程对应起来,产生了集成软件和服务的市场,但是语言和转换规则仍然必须单独定义,并且它们经过剪裁才能用于特定的业务过程。

对于企业IT系统更快速集成的要求,标准软件包仍不能满足的特殊业务单元需求的不断增长,以及基于因特网的新业务之间协作的需求(经常跨越边界实现互操作),这些都驱动了对IT行业的需求。在Web服务的旗帜下[⊖],世界各地的大型软件供应商和不断成长的小型IT提供商定义并支持了一组标准。



在这些定义中,提出了首批互操作性术语。大家普遍认为,任何通过互联网提供的服务都可以进行描述(如何发现服务、如何调用服务、什么是标准格式,等等)。但是,这还不是一种IT架构定义,只有提供者-消费者注册表这一基本概念,它的定义在如何公开找到web服务、如何宣传、如何由消费者访问等方面提供了标准。

在大多数EAI系统中占主导地位的“集中-发言”架构引发了迈向SOA的另一步。这些系统提供了一个集中控制单元,确保了对已注册应用的正确功能的调用,并处理其他问题,如访问权限、格式转换、确保交付等。将这一思想进行抽象,让它在企业范围内应用,ESB的概念就诞生了。IBM的SOA参考架构的创建就是为了展示这种方法。在参考文献Bieberstein et al. (2006)中,您可以找到关于ESB的深度讨论,以及它如何支持IBM的SOA参考架构。

这些服务要求人的干预,可以配置成整个系统,并根据客户的需求立即进行操作。SOA参考架构于是变成了一个企业运营的清晰框架。这一框架最终可以为最终用户构建的前端和根据Web 2.0的服务定义构建的应用程序提供一个基础,如混搭(mashup)等,并快速部署用AJAX或类似的透明开发工具编写的面向用户的应用程序。

SOA曾被视为避免IT部门未能及时交付的手段,因为它是服务于多种业务需求和IT需求的架构。在这个系列的第一本书《Service-Oriented Architecture (SOA) Compass, Bieberstein et al. (2006)》中,我们讨论了各种SOA的定义,它们在早期曾被使

⊖ “参考资料”小节列出了两个与SOA和Web服务有关的标准化组织。