华晟经世ICT专业群系列教材

数据共享与数据整合技术

叶树江　耿生玲　谢　锟　郭炳宇　姜善永　主编

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

华晟经世ICT专业群系列教材

数据共享与数据整合技术

叶树江　耿生玲　谢　锟　郭炳宇　姜善永　主编

人民邮电出版社
北京

图书在版编目（CIP）数据

数据共享与数据整合技术 / 叶树江等主编. -- 北京：人民邮电出版社，2019.6
华晟经世ICT专业群系列教材
ISBN 978-7-115-50947-5

Ⅰ. ①数… Ⅱ. ①叶… Ⅲ. ①数据处理－教材 Ⅳ. ①TP274

中国版本图书馆CIP数据核字(2019)第044604号

内 容 提 要

本教材一共6个项目，项目1为SOA基础知识导入，主要介绍了SOA的基本概念、发展历程，与企业IT战略之间的关系；项目2介绍了Web服务的相关基础知识，包括Web服务的体系结构特性、服务规范、SOAP、WSDL、UDDI等；项目3介绍了ESB的相关知识，明确了ESB与EAI之间的关系，介绍了SOA思想针对实际问题的具体实现思路，重点讲解了iESB引擎和iESB设计器的安装配置方法；项目4至项目6比较了REST和SOAP两种WebService方式的差别，并通过模拟校园中常见的多个信息系统整合开发应用场景，介绍了iESB暴露出来的服务在多系统的整合当中是如何被调用的以及不同的系统之间如何通过iESB实现数据共享，从而加深了对数据共享与数据整合技术在实际应用中的解读。本书在内容上贯穿以“学习者”为中心的设计理念，教材内容以“学”和“导学”交织呈现，相信能以通俗易懂的方式为学习者呈现其所需的教学内容。

本书适合作为高等院校计算机、电子信息类相关专业，特别是软件工程、企业信息化等专业的研究生与高年级本科生教材；也适合作为信息技术领域的咨询和培训机构的参考资料与培训教材。

◆ 主　　编　叶树江　耿生玲　谢　锟　郭炳宇　姜善永
责任编辑　贾朔荣
责任印制　彭志环

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京市艺辉印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：20.5　　2019 年 6 月第 1 版
字数：485 千字　　2019 年 6 月北京第 1 次印刷

定价：68.00 元

读者服务热线：（010）81055493　印装质量热线：（010）81055316
反盗版热线：（010）81055315

前 言

在这样一个数据信息时代，以云计算、大数据、物联网为代表的新一代信息技术受到空前的关注。教育战略服务国家战略，相关的职业教育急需升级以顺应和助推产业发展。从学校到企业，从企业到学校，华晟经世已经为中国职业教育产教融合这项事业奋斗了15年。从最早做通信技术的课程培训到如今提供以移动互联、物联网、云计算、大数据、人工智能等新兴专业为代表的ICT专业群人才培养的全流程服务，我们深知课程是人才培养的依托，而教材则是呈现课程理念的基础。如何将行业最新的技术通过合理的逻辑设计和内容表达呈现给学习者，并达到理想的学习效果，是我们开发教材时一直追求的终极目标。

在这本教材的编写中，我们在内容上贯穿以“学习者”为中心的设计理念——教学目标以任务驱动，教材内容以“学”和“导学”交织呈现，项目引入以情景化的职业元素构成，学习足迹借图谱得以可视化，学习效果通过最终的创新项目得以校验，具体解释如下。

1．教材内容的组织强调以学习行为为主线，构建了“学”与“导学”的内容逻辑。“学”是主体内容，包括项目描述、任务解决及项目总结；“导学”是引导学生自主学习、独立实践的部分，包括项目引入、交互窗口、思考练习、拓展训练及双创项目。

2．情景化、情景剧式的项目引入。模拟一个完整的项目团队，采用情景剧作为项目开篇，并融入职业元素，让内容更加接近于行业、企业和生产实际。项目引入更多地还原工作场景，展示项目进程，嵌入岗位、行业认知，融入工作的方法和技巧，更多地传递一种解决问题的思路和理念。

3．篇章以项目为核心载体，强调知识输入，经过任务的解决与训练，再到技能输出。采用“两点（知识点、技能点）”“两图（知识图谱、技能图谱）”的方式梳理知识、技能，在项目开篇清晰地描绘出该项目所覆盖的和需要的知识点，在项目最后总结出经过任务训练所能获得的技能图谱。

4．强调动手和实操，以解决任务为驱动，做中学、学中做。任务驱动式的学习，可以让我们遵循一般的学习规律，由简到难、循环往复、融会贯通；加强实践、动手训练，在实操中获得的学习经验将更加直观和深刻；融入最新技术应用，结合真实应用场景来解决现实性客户需求。

5．具有创新特色的双创项目设计。教材结尾设计双创项目与其他教材形成呼应，体现了项目的完整性、创新性和挑战性，既能培养学生面对困难勇于挑战的创业意识，又能培养学生使用新技术解决问题的创新精神。

本教材一共6个项目，项目1为SOA基础知识导入，主要介绍了SOA的基本概念、发展历程，与企业IT战略之间的关系；项目2介绍了Web服务的相关基础知识，包括Web服务的体系结构特性、服务规范、SOAP、WSDL、UDDI等；项目3介绍了ESB的相关知识，明确了ESB与EAI之间的关系，介绍了SOA思想针对实际问题的具体实现思路，重点讲解了iESB引擎和iESB设计器的安装配置方法；项目4至项目6比较了REST和SOAP两种WebService方式的差别，并通过模拟校园中常见的多个信息系统整合开发应用场景，介绍了iESB暴露出来的服务在多系统的整合当中是如何被调用的以及不同的系统之间如何通过iESB实现数据共享，从而加深了对数据共享与数据整合技术在实际应用中的解读。

本教材由叶树江、耿生玲、谢锟、郭炳宇、姜善永老师主编。主编除了参与编写外，还负责拟定大纲和总纂。本教材执笔人依次是：项目1叶树江，项目2耿生玲，项目3谢锟，项目4杨慧东，项目5刘静，项目6李慧蕾。本教材初稿完结后，由郭炳宇、姜善永、王田甜、苏尚停、刘静、张瑞元、朱胜、李慧蕾、杨慧东、唐斌、何勇、李文强、范雪梅、冉芬、曹利洁、张静、蒋平新、赵艳慧、杨晓蕊、刘红申、黎正林、李想组成的编审委员会相关成员进行审核和内容修订。

整本教材从开发总体设计到每个细节都饱含了我们团队的协作和细心打磨，我们希望以专业的精神尽量克服知识和经验的不足，终以此书飨慰读者。

本教材提供配套代码和PPT，如需相关资源，请发送邮件至renyoujiaocaiweihu@huatec.com。

编　者

2018年7月

此为试读，需要完整PDF请访问：www.ertongbook.com

目 录

项目 1

SOA 基本概念初探

项目引入

我叫 Alphonse，是一名非资深 Java 程序员，刚刚入职软件公司，就职 IT 业务部。刚来公司，我就听主管 Edward 说，公司的信息化建设相对滞后，CIO 决定实施 SOA（面向服务的架构）改造，对现有种类繁多的信息服务系统进行重新治理。

Edward 仿佛从我的眼神中看出了一丝慌乱，告诉我不用着急："这个新闻你先看一看，了解一下 SOA 的重要性，然后咱们再慢慢熟悉 SOA 的基本概念。"

银行业反思：如果用 SOA 就不会发生金融危机

2009 年 09 月 02 日 13:15 中国计算机报

21 世纪初，在全球金融崩溃前的那段平静的日子里，我遇到了一位曾在花旗集团任职的朋友——斯克普·斯诺。他虽身任花旗集团企业架构部高级副总裁，但并不高兴。

在花旗集团的时候，他一直在企业内全力推行 SOA，但最终却在这场战争中输了。斯克普说，假如他或者其他金融巨头的 IT 系统架构师最终取得胜利的话，这场金融危机将不会发生。他表示，SOA 的应用能够很容易地对即将发生的金融风险进行预警。但可惜的是，企业的各个部门并不愿意在 SOA 的应用方面花费太多的精力。

IT 系统架构师看到了其中的好处，但是公司内各个部门由此能够得到什么好处呢？当时，这些部门可以很容易地赚到大把的钞票，因此，这些部门的领导人对 SOA 并没有太多的热情。事实上，公司采用的是相对固定的薪酬模型，即使在采用了 SOA 之后，也不会为员工带来额外的利益。

他说："无论是 IBM、Oracle 还是 HP，都希望保住自己在企业中已经占有的领地。如果应用 SOA，他们固有的利益怎么办呢？因此，他们对此并不感兴趣。"

这些大的供应商都想将你锁定在他们的私有体系内，而像花旗这样的大公司，其内部的多个系统也只能在有限的情况下互联互通。

这种缺乏整合的情形很具有讽刺意味。1999 年颁布的 Gramm-Leach-Bliley 法案消除了大萧条时期对于禁止同一家金融机构同时承担银行业务、投资银行业务以及保险服务。该法案使得抵押担保债券这一综合性的业务成为可能，但正是此种债券使全球的金融系统崩溃，因此该法案饱受谴责。因为原有的三大业务系统始终保持相对独立，很难整合到一起，因此产生了很多未知的风险。而这些新的必然都逃出了“监控雷达”的范围。

根据斯克普所说，花旗集团有多个未整合的风险管理系统，以及多种格式并不统一的账簿，管理人员无法将各种格式的数据统一到一张表格中。斯克普推测，如果公司的高层意识到了他们在抵押贷款的价值方面正处于急速下降的趋势，他们可能就会在崩溃之前摆脱这类资产。

看完这则新闻，我惊呆了，SOA 竟然这么重要，影响这么深远，那我可得打起百倍精神努力打好基础。

Edward 接着解释道：“SOA 战略 (包括业务流程管理和软件服务方法) 作为一种技术创新可以有效助力企业调整业务流程和削减经营开支。当企业需要减少经营成本时，必须要查看业务流程，也许有一些集成，也许有一些人工活动，我们也许会在那些不必要的业务流程中损失许多收入。目前软件行业也不像以前那样遍地黄金了，未来只有勇于变革、精细经营的企业才能渡过难关，而 SOA 在经济衰退期间能发挥非常重要的作用。从历史上看，强大的公司在这种动荡的时期将会变得更加强大，能够处于更有利的地位迎接下一次的经济繁荣。”

他列举了麦肯锡（Mckinsey）从 1982 年至 1999 年研究的 1000 多家公司的例子。通过对麦肯锡的研究，他发现，当行业领导者在经济衰退中退出的同时，新崛起的公司是那些在经济衰退期间在基础设施方面投资最多的公司。

“所以，这些公司不是仅守着现金和削减成本，实际上是在基础设施方面进行了技术创新，让自己等待着繁荣时期到来。”Edward 语重心长地说。

Edward 的话让我深深意识到 SOA 的重要性。嗯，接下来我要元气满满地投入到 SOA 基本概念的学习中去了，小伙伴们，一起加油哦！

知识图谱

项目 1 知识图谱如图 1-1 所示。

此为试读，需要完整PDF请访问：www.ertongbook.com

图1-1　项目1知识图谱

1.1　任务一：什么是 SOA

【任务描述】

刚来公司就可以接触改造 SOA 这么大的项目，我既兴奋又感到时间的紧迫，领导说了，对于我这个 SOA 新人来说，第一步，就是先熟悉 SOA 的基本概念。这就是我们要完成的第一个任务：了解什么是 SOA。

1.1.1　SOA的基本概念

SOA 的英文全称是“Service Oriented Architecture”，翻译成中文有很多种意思：“面向服务的体系结构”“以服务为中心的体系结构”和“面向服务的架构”，一般我们将其翻译成“面向服务的架构”。SOA 基本上可以分为两类，SOA 主要是一种架构风格；SOA 是包含运行环境、编程模型、架构方法和相关方法论等在内的一整套新的分布式软件系统方法和环境。第二类概括的范围更大，它涵盖服务的整个生命周期，即建模—开发—整合—部署—运行—管理，着眼于未来的发展，SOA 是分布式软件系统架构方法和环境的新发展阶段，所以我们更倾向于后者。

在 SOA 风格中，最核心的抽象手段是服务，业务被划分为一系列粗粒度的业务服务和业务流程。业务服务具有相对独立、自包含、可重用的特点，由一个或多个分布的系统实现，而业务流程则是由服务组装而来。一个“服务”定义了一个与业务功能或业务

数据相关的接口，以及约束这个接口的契约，如服务质量要求、业务规则、安全性要求、法律法规的遵循、关键业绩指标（Key Performance Indicator，KPI）等。接口和契约采用中立、基于标准的方式进行定义，它独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在不同系统中的服务可以以统一和通用的方式进行交互。除了这种不依赖于特定技术的中立特性，我们还可以通过服务注册库（Service Registry）加上企业服务总线（Enterprise Service Bus，ESB）来让系统支持动态查询、定位、路由和中介的能力，这样就使得服务之间的交互是动态的，位置是透明的。

技术和位置的透明性，这使得服务的请求者和提供者之间高度解耦。这种松耦合系统的好处有两点：一点是它适应变化的灵活性；另一点是当某个服务的内部结构和实现逐渐发生改变时，不会影响其他服务。而紧耦合则是指应用程序的不同组件之间的接口与其功能和结构是紧密相连的，因而当人们对应用程序的功能需求发生变化时，某一部分的调整会随着各种紧耦合的关系引起其他部分甚至整个应用程序的更改，这样的系统架构就很脆弱了。

例如，某学校需要开发一个移动办公系统，需要全校老师的用户信息，如果在系统中一一添加，不仅工作量巨大，还可能出现信息与学校人事管理系统中的信息不一致的现象。解决办法就是直接从人事管理系统中调用和读取用户信息。

从人事管理系统中提供用户信息，这就是一个服务，即人事管理系统的一个功能单元。那么通过什么方式把服务提供出来呢？这就涉及接口的概念，所谓接口就是服务方和使用方都能够识别的对接方式，采用标准的接口协议，例如 Socket 协议、WebService 协议等。

这种应用程序之间交互的方式采用的是接口，而不是直接调用程序，因此对于程序内部的业务逻辑变化，只要接口不变，就不会影响使用方的系统，这个就是松耦合的概念。

任何企业、组织都有各种各样的应用，应用之间都会有交互，如果应用之间直接调用对方的接口，就会形成蜘蛛网状。ESB 就是把各个应用提供的接口统一管理并将其暴露出来，所有的应用接口都通过 SOA 平台交互，避免了业务之间的干扰。SOA 对服务接口的统一管理如图 1-2 所示。

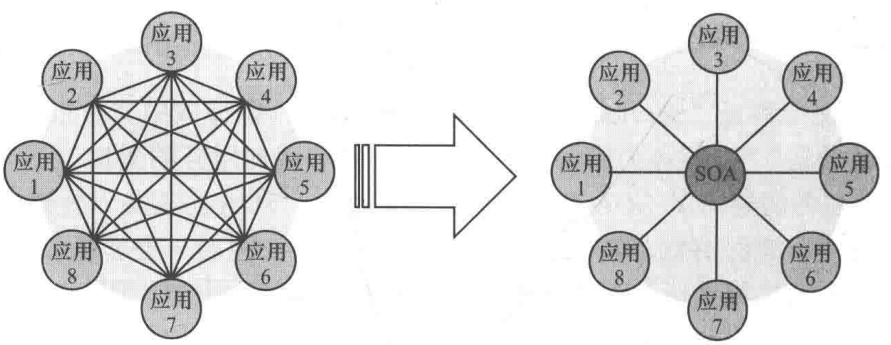

图1-2　SOA对服务接口的统一管理

SOA 带来的另一个重要观点是业务驱动 IT，即 IT 和业务更加紧密地对齐。以粗粒度的业务服务为基础来对业务建模会产生更加简洁的业务和系统视图；以服务为基础来实现的 IT 系统更灵活、更易于重用、能更好（也更快）地应对变化；以服务为基础，通过

显式地定义、描述、实现和管理业务层次的粗粒度服务（包括业务流程），提供了业务模型和相关IT实现之间更好的“可追溯性”，减小了它们之间的差距，使得业务的变化更容易传递到IT。因此，我们可以将SOA的主要优点概括为IT能够更好更快地提供业务价值、快速应变能力、资产重用。

从演变的历程来看，SOA在很多年前就被提出来了，现在SOA的再现和流行是若干因素的结合。一方面是多年软件工程的发展和实践所积累的经验、方法和各种设计/架构模式，包括EAI和中间件；另一方面是互联网的多年发展带来前所未有的分布式系统的交互能力和标准化基础。与此同时，企业越来越重视业务模型本身的组件化，以支持高度灵活的业务战略。但是现有的企业软件架构不够灵活，难以适应日益复杂的企业整合，难以满足随需应变的商务需要，因此与业务对齐、以业务的敏捷应变能力为首要目标、松散耦合、支持重用的SOA方法得到世界众多企业的青睐。

SOA容易混淆的几个基本问题如下。

第一，SOA是架构风格、方法，而不是具体架构、具体实现技术（如WebService）、具体架构元素（如ESB）。

经常有人认为只要用了WebService，就是SOA，这个观念是不对的，WebService只是实现服务的一种具体技术表现形式；同样，认为建设SOA，就是购买软件，建个ESB，这也是不对的，ESB只是SOA风格中的一部分。首先，ESB是一种从实践中总结出来的架构风格元素，即BUS（总线模式）；其次，ESB的主要功能是负责连通性和服务中介（Service Mediation），解耦服务的请求者和服务的提供者。

第二，SOA的首要目标是IT与业务对齐，支持业务的快速变化；其次是重用IT架构的灵活性和IT资产的重复利用。

业务对敏捷性的需要，是SOA最大的驱动力。一方面是业务在这方面的要求越来越高；另一方面是目前的IT技术还不够灵活，很难适应业务快速变化的需求。不仅仅是因为IT架构不灵活，更重要的是业务模型中的元素和IT系统的元素之间存在很大的差距，这种不对齐，导致业务人员和IT人员之间的沟通差异，业务的变化需要花费很大的代价传递到IT系统。这种业务和IT的对齐，需要在IT系统中实现更高阶的抽象元素，就是业务模型中的元素（服务、流程、业绩管理），并且需要满足业务需要的水平整合（将人、信息、应用和流程端到端地动态整合起来）。这样就形成了一个以服务为中心的、端到端整合的环境，首先使得业务变化可以在业务元素的层面上沟通得更加更容易、更准确地从业务传递到IT。其次，这种变化被隔离在需要变化的局部，而不扩散到系统的其他部分。这就需要整个IT架构本身是松散耦合的，一个服务的变化（功能、数据、过程、技术环境等）不影响其他服务。最后，我们希望这些反映业务元素的服务，是相对稳定、可以重用的，这对快速适应变化、减少成本是非常重要的。

第三，在工程上，SOA的重点是服务建模和基于SOA的设计原则进行架构决策和设计。

经常碰到客户提出这样的问题：SOA挺好，为什么好？怎么做才是SOA的方法？与过去的方法相比，和其他（如OO/CBD）有什么不同？有时候用一个J2EE服务器就好了，为什么那么复杂地建设SOA？

从建模和设计的角度来说，SOA 更多地侧重在业务层次上，也就是通过服务建模将业务组件化为服务模型，它是业务架构的底层，是技术架构的顶层，它负责承上启下，是灵活的业务模型和 IT 之间的桥梁，保证二者之间的“可追溯性”。从技术架构的顶层往下的具体设计，是基于已有的方法，比如从 OO/CBD 来进行的。从架构的层次上看，SOA 更多地侧重于如何将企业范围内多个分布的系统（包括已有系统 / 遗留系统）连接起来（ESB、Adapter/Connector），比如如何将它们的功能、数据转化为服务？如何通过服务中介机制（ESB、Service Registry）保证服务之间以松散耦合的方式交互？如何组装（集成）服务为流程？如何管理服务和流程等？从不同的服务之间的交互往下是对于实现各个服务的一个具体应用，它的架构、设计和实现可以基于已有的实践和方法应用，比如 J2EE 或 .NET。

有时，由于业务需求比较简单，所有这些东西都在一个 J2EE 的应用服务器上，有些要素不是那么突出，不过随着系统规模的扩大，要解决的业务问题更复杂、涉及范围更大时，SOA 的各种架构要素就会变得越来越重要。

1.1.2 SOA发展的驱动力

SOA 技术在 IT 界掀起巨大的狂潮，然而它不同于以前的模块化编程，它面向对象、Web 技术等技术变革，它们不论多难理解，总是能很快被大家接受，SOA 之所以让很多人觉得难以理解，是因为它不再单纯地从 IT 从业人员的角度理解 IT 系统，而是从业务人员的角度分析 IT 业务系统。

有两种现象相继呈现：一方面是企业 SOA 改造，精简企业业务流程，提升企业市场竞争与创新的能力，企业 IT 部门成为了企业管理的核心链条——“神经系统”；另一方面是很多企业觉得无从下手，SOA 太空无实，业务部门人员不愿支持，业务流程改造单靠 IT 部门难以完成，而企业内非 IT 部门，尤其是管理层对 SOA 了解得还很少。

1. 业务驱动

SOA 的概念早在 20 世纪就已提出，近两年在许多大公司相关产品与业绩推动下，它才进入了实际应用的黄金阶段。

SOA 的出发点是从业务角度重用应用系统的开发元素，最大限度地降低 IT 系统开发与维护的成本。很多企业的 CIO 都面临一个共同的问题：随着网络建设的浪潮，各种业务系统的开发如雨后春笋，一些大型企业，需要维护成百上千个业务系统是很常见的事，从机房配置、服务器管理、各种支撑系统的维护都让 IT 部门难以应付，更不用说被病毒攻击过后的清理系统与恢复业务工作，仅仅是查看各个业务系统的状态，就需要工作人员花很长时间，对于保障业务的持续性，更是繁复之至。业务系统的繁多与各自孤立，为新业务的上马带来更大困难，重复开发造成极大的浪费，信息不互通让每个系统都“麻雀虽小，五脏俱全”，企业失去了市场竞争的灵活性，这些弊端都极大地触动了企业管理者的神经。

很多大型公司开始推广 ERP 之类的大型企业软件系统，希望在一个庞大系统架构中，可以融合更多的业务流程，各个业务的信息可以交流，避免各个“业务孤岛”带来的管理弊端与效率低下。然而随着单个系统的庞大，开发的难度呈指数般提升，要考虑的因

素太多了，客户业务又千差万别，企业的管理成为极大瓶颈；另外“同制化”的设计模式恰恰抹杀了企业的创造力，而失去了“特点”的企业等于选择“自杀”。考虑到IT基础架构如何适应企业的“创造化”需求、新业务的开发如何快捷、如何降低IT支撑系统的管理成本并提供持续性的服务保障等需求，CIO们重新选择了SOA。

在这种情况下，SOA重新被提出来，SOA是一种IT技术框架，是一种最佳实践，而不是一种具体的技术，能实现SOA的技术很多，如何选择的关键是能否达到SOA提出的业务灵活度的目标。

对于SOA的思路，其实IT开发人员中有过类似的想法，我们回顾一下编程人员走过的历程：模块化编程提炼可重用的程序，方便调用，提高软件的结构性；后来发展到面向对象，把数据与程序封装在一起，让软件设计人员的思路逐渐接近现实的人类思维方式；B/S架构的推广将客户端的维护变得简单化，业务更适合于网络方式；Web2.0的发展解决了B/S体系的交互问题；中间件技术让跨平台、跨语言的业务开发变得容易，IT开发人员一直在探索、提炼可以重用的、优秀的软件模块，以便使我们的业务系统开发如搭积木一样容易。

虽然SOA是IT开发人员的思路，但推动SOA的是企业管理层，SOA是业务驱动发展的，而不是由技术驱动发展的。从新的视觉角度看，并非所有的企业CIO们对SOA都得心应手。

我们不再从IT开发人员的眼光看待要开发的业务系统，而是从业务的使用者角度看待要开发的系统，面向服务就是面向系统的实际使用者，“谁干谁说了算”，系统应该具备什么功能，应该做成什么样子，要看用户使用的效果。简单地说，就是用“敏捷”的开发思路，代替“闭门造车”的开发方式。所谓敏捷就是用户的参与，用户不懂你的专业“语言”，就需要快速的模型与界面展现，快速展现不重用是不现实的，而用户理解不从业务流程入手，是与用户没有共同语言的。

2. 计算环境的演变

（1）计算环境

计算环境由一组计算机、软件平台和相互联通的网络组成，这个环境能够处理和交换数字信息，允许外界访问其内信息资源。不同的计算环境有不同的计算风格和编程模型，由一些特定于该计算环境的技术来支撑。如何在一个计算环境中分割和部署计算能力、数据资源，如何让各个部分相互通信和协作，如何在概念上对问题域进行建模，然后映射到该计算环境，都会受到计算环境的影响和制约。因此，了解计算环境的历史，会帮助我们理解面向服务的计算环境是如何演变而来的。

（2）计算环境的演变历程

计算环境的演变经历了若干个阶段，在早期的主机时代，绝大多数的计算功能和系统的组成部分，都在一台机器里。在二十世纪八十年代，随着PC的繁荣发展，计算环境发生了很大的变化。通过局域网相互连接的计算设备构成客户/服务器计算环境，计算资源和数据资源被适当地分割，客户和服务器通过网络协议、远程调用或消息等方式相互协作，完成计算。

为了满足更高的可伸缩性需求出现了多层架构，计算资源和数据资源的分布多样化，

集成企业中原来已经存在的计算环境、尤其是主机及其遗留系统之间的集成也变得越来越重要。中间件迅速发展，开始出现分布式对象、组件和接口等概念，用于在计算环境中更好地分割运算逻辑和数据资源。计算环境中不同部分之间的交互，也从原有相对底层的网络协议、远程调用和消息机制的基础上，发展到支持分布式对象、组件和接口之间的交互，这种交互在名字服务（Naming Service）等的支持下，通常是位置透明的。但由于缺乏普遍的标准化支持，很难做到技术透明，系统是紧耦合的。

随着互联网的发展，开放和标准的网络协议被普遍支持，所有底层计算平台都开始支持这些标准和协议，这导致一个计算环境内部和各个计算环境之间交互的屏障被打破。数据和功能的表示与交互在 XML、Web 服务（WebService）技术与标准的基础上，保证了通用性和最大的交互能力，这使得计算环境发展到一个全新的阶段——基于标准、开放的互联网技术的计算环境。在这样的计算环境中，各个部分可以采用异构的底层技术，它们使用 XML 来描述和表示自己的数据和功能，采用开放的网络协议（如 Http）来握手，在此之上，基于 Web 服务来进行互操作和交换数据。在这里，一个很重要的新概念是服务，它是一个自包含的功能，使用者通过明确定义的接口（契约）来与一个服务交互，这个接口的描述基于 WSDL（WebService Description Language，网络服务描述语言）的开放标准。对象和组件重在表示一个事物本身的组成部分和相互关联（也就是 WHAT“THINGS”ARE 的问题），而服务则表示一个事物做什么（也就是 WHAT“THINGS”DO 的问题）。Web 服务是实现服务的技术手段，就如同各种编程语言中的对象是实现对象的技术手段，J2EE 中的 EJB 是实现组件的技术手段一样。这种基于标准、开放的互联网技术，以服务为中心的计算环境，我们称之为“面向服务的计算环境”。

（3）面向服务的计算环境

在面向服务的计算环境中，系统可以是高度分布、异构的。它一般包括服务运行时环境、企业服务总线、服务网关、服务注册库和服务组装引擎等，如图 1-3 所示。

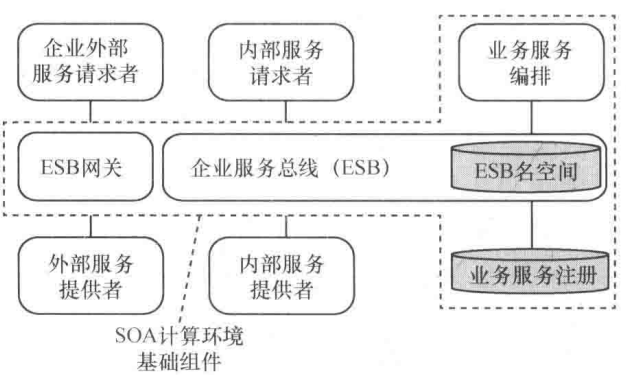

图1-3　SOA计算环境的组成要素

服务运行时环境提供服务（和服务组件）的部署、运行和管理能力，支持服务编程模型，保证系统的安全和性能等质量要素；服务总线提供服务中介的能力，使得服务使用者能够以技术透明和位置透明的方式访问服务；服务注册库支持存储和访问服务的描述信息，

此为试读，需要完整PDF请访问：www.ertongbook.com