国际信息工程先进技术译丛

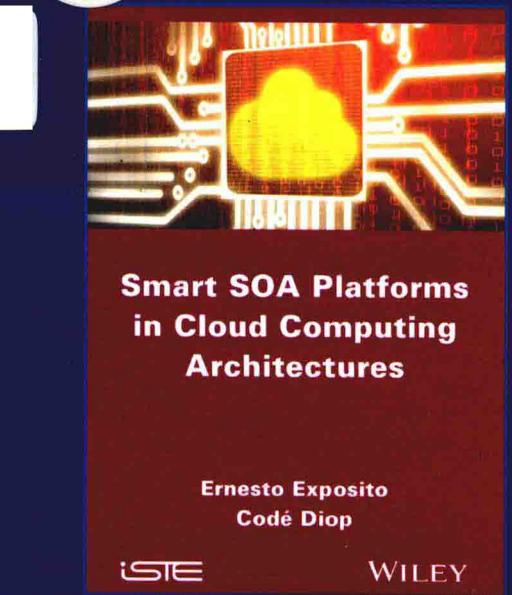
ISTE WILEY

云计算体系架构中的智能SOA平台

Smart SOA Platforms in Cloud Computing Architectures

[法] 埃内斯托·埃斯波西托（Ernesto Exposito）
科德·迪欧普（Codé Diop） 著

谭励 冀赛赛 译



机械工业出版社
CHINA MACHINE PRESS

国际信息工程先进技术译丛

云计算体系架构中的智能SOA平台

[法] 埃内斯托·埃斯波西托（Ernesto Exposito）
科德·迪欧普（Codé Diop） 著

谭 励 冀赛赛 译

机 械 工 业 出 版 社

图书在版编目（CIP）数据

云计算体系架构中的智能 SOA 平台/（法）埃斯波西托，（法）迪欧普著；谭励，冀赛赛译. —北京：机械工业出版社，2016.5

（国际信息工程先进技术译丛）

书名原文：Smart SOA Platforms in Cloud Computing Architectures

ISBN 978 - 7 - 111 - 53312 - 2

Ⅰ.①云… Ⅱ.①埃…②迪…③谭…④冀… Ⅲ.①互联网络 - 网络服务器 - 研究 Ⅳ.①TP368.5

中国版本图书馆 CIP 数据核字（2016）第 061549 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

策划编辑：顾 谦 责任编辑：顾 谦

责任校对：刘怡丹 封面设计：马精明

责任印制：常天培

北京京丰印刷厂印刷

2016 年 4 月第 1 版第 1 次印刷

169mm × 239mm · 11.5 印张 · 235 千字

0 001—2 800 册

标准书号：ISBN 978 - 7 - 111 - 53312 - 2

定价：49.90 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务 网络服务

服务咨询热线：010 - 88361066 机 工 官 网：www.cmpbook.com

读者购书热线：010 - 68326294 机 工 官 博：weibo.com/cmp1952

010 - 88379203 金 书 网：www.golden-book.com

封面无防伪标均为盗版 教育服务网：www.cmpedu.com

本书介绍了在云计算架构中的智能 SOA（面向服务的体系结构）平台，同时为了展示如何在现代企业应用程序的开发中有效地应用 SOA 的概念，书中采用了一个基于项目的学习方法，即 yPBL。另外，通过对 eBay 系统的案例研究，本书介绍了构建智能 SOA 平台所有必要的组件，并引入了分布式系统设计与开发的基础方法。书中解释说明了如何使用一个实用的、基于手册的方法来搭建能够满足大量非功能性需求的智能 SOA 平台。书中具有十分详细的操作步骤，对于读者来说，可以根据手册和菜单的内容一步步实现智能 SOA 平台的搭建。

本书适合网络领域的专家以及任何对分布式系统开发感兴趣的读者阅读，主要面向对面向服务、事件驱动以及云计算架构感兴趣的网络与软件工程方面的学生、工程师以及研究人员、专业技术人员。本书从企业的层次上，为感兴趣的读者提供了分布式应用程序的解决方案。

原书前言

最初分布式应用程序开发的复杂度被认为是原始的传输控制协议（TCP）/互联网协议（IP）网络模型。这个复杂度随着网络传输和网络服务的演变一直在快速增长。因此，直接在应用程序编程接口（API）工作的开发者们需要更高水平的专业知识。这些专业知识包括由大量的传输协议提供的所有服务，以及这些各种各样的服务如何与提供 IP 网络层的底层服务结合起来的技术。

例如，一个 Web 应用程序的开发者想要在一台移动终端机上进行操纵，他应该掌握如何选择一个用于不同网络环境的适当的传输协议，包括应用程序逻辑在内。也就是说，开发者不仅应该考虑在静态环境下，Web 应用程序通过连接 WiFi 或蜂窝网络进行的操作，也应该实现足够的逻辑来应对从一个网络移到另一个网络的变化，甚至是终端断开连接的情况。

今天，分布式应用程序的开发人员需要专注于应用程序逻辑，以便使开发更有效率，同时减少开发延误和实现过程中的一些错误。为了达到这个目标，他们需要将通信系统的所有细节做一个抽象。这意味着他们不需要直接应对如何选择和配置网络传输和网络服务，以及如何处理在应用程序分布式组件之间数据传输的细节。

读者

本书适合网络领域的专家阅读，但实际上它也适合其他读者，包括任何在学习开发分布式系统时，对能够将传输层和网络层的服务做一个抽象的解决方案感兴趣的人们。本书将从企业内部和企业之间的层次上，为感兴趣的读者提供分布式应用程序信息技术（IT）解决方案。

特别的，本书将介绍各种中间件通信层的演进，所谓的中间件通信层即为了隐藏应用程序组件的分布复杂性而被设计的一个适应层。本书将把精力集中在深深影响了这类系统设计的主要范式上，这类主要范式是面向服务的体系结构（SOA）范式。即使这不是一个新的或革命性的概念，经过在这一领域数年的工作，人们已经意识到 SOA 范式所涉及的复杂性太高以至于通常不好理解。SOA 通常演变为 Web 服务（WS）的基本概念，它失去了范式迫切需要构建敏捷和灵活的分布式系统的巨大优势。这在云计算架构领域是特别重要的，在这个领域遵循面向服务的架构、平台或软件是一项基本原则。

方法

本书的目标是根据实际使用情况进行演示，演示在开发当前和下一代企业应

用程序时，如何有效地应用云计算平台和 SOA 的概念。为了避免一条条地罗列各种理论的定义和概念，本书决定提出一个实践性的实用方法，命名为基于项目的学习。这种方法是基于概念的合理化和在实际项目的设计和开发基础上的能力提高。

我们选择一个案例，研究的灵感来自众所周知的易贝（eBay）在线市场。基于这个 C2C（客户到客户）分布式系统用例，将介绍所有必要的组件来构建智能 SOA 平台。最初这个平台的实现是在实现 SOA 支柱的基础上完成的，SOA 的基本支柱即企业服务总线（ESB），旨在保证可集成性、互操作性和可扩展性等非功能性的需求。本书将考虑这个初始阶段作为基本 SOA 概念的开发，它将作为平台的第一个版本解决方案，被命名为 1.0 智能 SOA 服务平台（SSOAPaaS 1.0）。基于额外的非功能性需求，主要是关于一个更好的解耦或减少分布式组件之间的直接依赖关系以及主动性需求属性，将会以 SSOAPaaS 2.0 之名开发一个新版本的平台。SSOAPaaS 2.0 包括第二个基本支柱，丰富了前一个平台，第二个基本支柱通常称为基于事件的事件驱动架构（EDA）系统。在最后的开发中，额外的非功能性需求，就可管理性和可伸缩性方面而言，将在 SSOAPaaS 3.0 中给予解决。SSOAPaaS 3.0 解决方案的扩展与自我管理的属性将使智能平台实现本书的目标：为基于 SOA 的系统设计和开发一个智能平台。

本书需要提供一个专门的 IT 架构以便很容易地安装和部署不同版本的 SSOAPaaS 所需的组件，将开发基于虚拟化和云技术的解决方案，这个解决方案促进了一个有效的、便携的和可扩展的架构的设计和实现。这个平台名为智能 PaaS（SPaaS），它提供所需的架构管理功能以支持 3 个不同版本的智能 SOA 平台。

本书结构

本书包括 6 章主体、概述以及总结和展望 3 个部分：

在概述部分（第 0 章），提供一个总体的介绍，包括本书的主要目标和所采用的基于项目的学习方法。

第 1 章主要内容是案例研究，介绍了提出三代 SOA 平台的主要需求。

第 2 章旨在介绍 SOA、EDA、云计算和自主计算的基本概念。它将展示通信层中间件、企业应用程序集成和 SOA 解决方案的基本概念，并将介绍 WS 和 ESB 技术。企业界重要的企业集成进化描述所代表的面向消息和 EDA 范例会在其后介绍。此外，关于虚拟化和云计算架构的介绍将展示非功能性需求，诸如如何使用先进的虚拟化和云计算策略来实现可管理性和可伸缩性。本章将说明手工管理 SOA 平台的复杂性，接下来介绍用来实现智能平台解决方案的自主计算框架。

第 3 章主要介绍了包含一系列关于 SPaaS 解决方案开发内容的第一本手册，它旨在开发智能 IT 基础设施，需要安装和部署不同版本 SSOAPaaS 平台所需的组件。

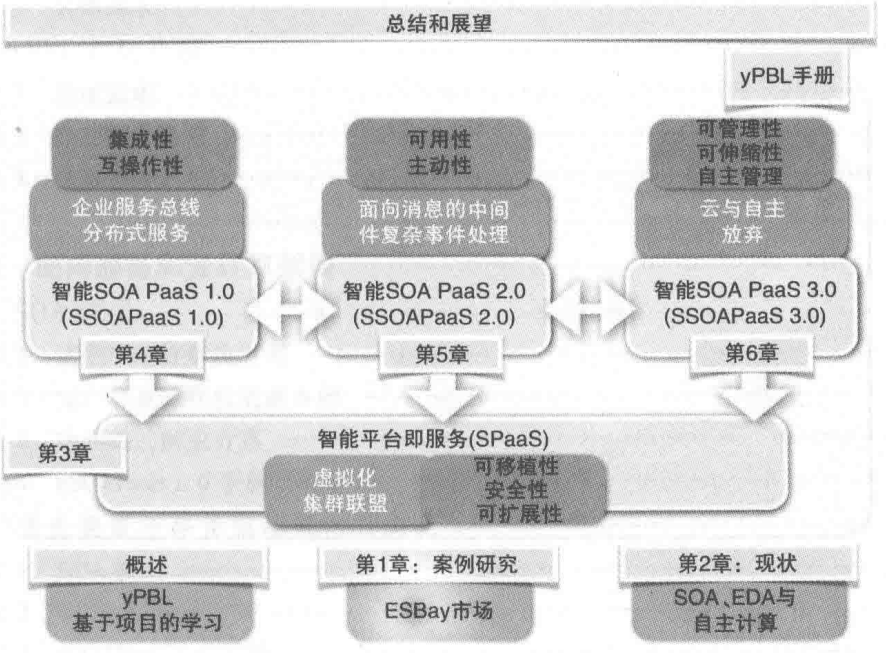
第4章基于这本手册，展示了通过SSOAPaaS 1.0平台，SOA范式和技术如何满足互操作性、可扩展性和集成性等非功能性需求。

第5章旨在说明面向消息的中间件（MOM）概念以及如何使用消息传递系统［即Java消息服务（JMS）］来提供异步通信信道（即点对点或发布/订阅模式）。此外，有关复杂事件处理（CEP）的概念也有所介绍。本章介绍了在SSOAPaaS2.0平台下，面向消息和事件驱动的范例和技术如何能够满足可用性和主动性等非功能性需求。

第6章展示了最后一本手册，它专注于如何在全球化和广泛联系的企业中架构现代SOA平台。这一手册向人们展示了非功能性需求，例如可管理性和可伸缩性如何在SSOAPaaS 3.0平台中实现。

最后，总结和展望部分（第7章）总结了通信中间件和SOA平台所扮演的角色在满足集成性、互操作性和大型网络系统的性能需求方面的演变。当前和未来的挑战和观点将被描述出来，这些挑战包括在云计算架构方面未来智能和自主SOA平台的设计和开发。此外，智能的自我配置指南、自主开发和优化策略也将会被展示出来。这些策略指导着下一代作为一种服务解决方案的平台开发。

图Ⅰ.1给出了整体结构和各种将要讨论的话题。



图Ⅰ.1 本书结构

为了展示如何在现代企业应用程序的开发中有效地应用SOA概念，本书提出了一个基于项目的学习方法，基于案例研究并且引入了分布式系统设计与开发

的基础方法。本书将介绍 SOA 和 EDA 以及云计算和自主计算模式如何在现代分布式架构中发挥基础性的作用。

本书将解释说明，通过使用一个实用的基于手册的方法，所有这些技术和方法的融合如何使智能 SOA 平台的搭建能够满足大量的非功能性需求。这些非功能性需求包括可集成性、互操作性、可用性、主动性、可管理性、可扩展性、可移植性、可伸缩性和安全性。

Ernesto Exposito 和 Codé Diop

目　录

第0章 概 述

在这个概述里，将会介绍面向服务的体系结构（SOA）范式应用于分布式系统的动机。此外，基于项目的名叫 yPBL 的学习方法以及案例研究，将会在接下来的内容里进行介绍。

0.1 分布式系统的演变

第一代网络应用程序开发的复杂性在原始传输控制协议/互联网协议（TCP/IP）网络模型［EXP 12］的环境中是相对重要的。第一代网络应用程序主要是由客户端/服务器或分布式 P2P 应用程序代表（见图 0.1）。这些应用程序的例子有文件传输、网络浏览、音频/视频点播、交互式音频和视频会议等。

这些应用程序的一部分静态实现直接使用 TCP 提供的完全可靠和完全有序的服务或使用用户数据报协议（UDP）提供的非可靠和非有序的服务，基于互联网协议（IP）提供的基础尽最大努力传输服务。最近的应用程序具有更复杂的服务质量（QoS）需求（如延迟、抖动、带宽、可靠性和顺序），并且已经能够利用更专业的传输服务［如数据报拥塞控制协议（DDCP），流控制传输协议（SCTP）和多路径传输控制协议（MPTCP）］或网络层服务［如集成服务（IntServ）、差分服务（DiffServ）和多协议标记交换（MPLS）技术］。此外，一个重要的应用程序和会话层协议已经发展成了方便、常见的网络功能，例如会话控制、多媒体数据传输和显示。

因此，与分布式系统开发相关的复杂性一直在快速增长，主要是由类似这些演变、传输、网络和更高层的多样性所造成的。今天，直接在传输层应用程序编程接口（API）工作的开发人员需要一个高水平的专家来提供如何在 IP 网络提供的服务上将这些传输服务与底层服务相结合的技术。

这里通过一个示例来说明这种复杂性：基于网络的移动应用程序的开发。如果开发人员需要直接访问传输层 API，不仅是在启动应用程序时，而且在不同网络环境情况下，都需要包括在应用程序逻辑中，选择使用适当的传输协议。它不仅意味着开发人员应该考虑静态情况下 Web 应用程序可以通过 WiFi 或蜂窝网络连接，还应该实现适当的逻辑以应对交接时从一个到另一个网络或终端断开连接和重新连接的情景。

为了应对这种复杂性，人们已经付出重要的努力来促进网络功能的实现，从

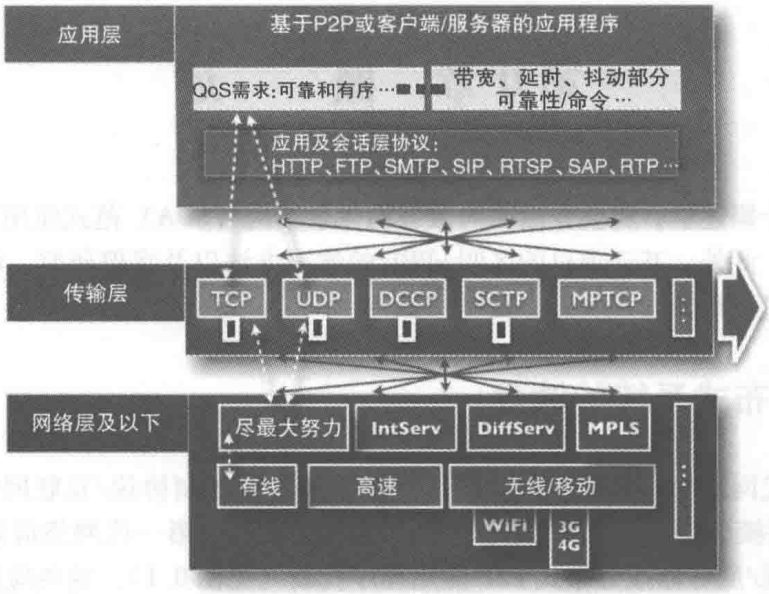


图0.1 第一代网络应用程序

而允许分布式应用程序的开发人员可以专注于应用程序逻辑设计。这意味着，例如，开发人员应该不需要应对在应用程序的分布式组件之间的数据传输过程中，通信系统的选择和配置或如何处理错误、延迟或其他意外事件。

在第二代网络应用程序的发展中，基于一种新层，即传输层和应用程序层之间的通信中间件（见图0.2），这些努力有了结果。

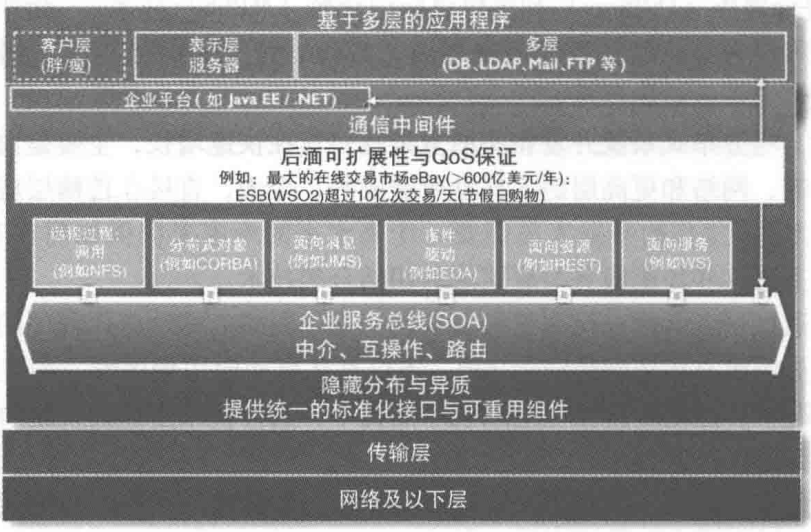


图0.2 第二代网络应用程序

通信中间件层可以被定义为一个适应层，它旨在隐藏所有与应用程序的分布相关的来自开发人员的应用程序组件的复杂性。换句话说，网络应用程序的开发人员（即多层应用程序，包括客户端、服务器和后端分布式企业组件）应该能够与任何分布式组件进行交互，就好像它们是位于相同的执行环境中一样。此外，通信中间件解决方案也可以通过提供统一的标准接口、实现组件的语言和环境的抽象（如 Java EE、.NET、C 和 C++）来隐藏异质性的组件。解决方案包括远程过程调用、分配对象、面向消息、面向资源等。面向服务的解决方案可以促进网络应用程序的实现能够由完全无关的传输层和网络层来提供服务。

SOA 范式促进了基于基本服务的分布式系统的设计，还可以实现复杂的组合业务流程。

最初的在 SOA 架构上的集成工作基于企业应用程序集成（EAI）的解决方案。EAI 旨在提供企业应用程序之间集成的服务。这个解决方案是基于服务器作为中心，并提供一个通用的 API 的应用程序集成来实现的。服务器还实现了所需的专有链接与非标准的第三方解决方案。EAI 实现了一个被定义为“轮辐式”（hub - and - spoke）的架构模型。EAI 的主要缺点是高度依赖中央枢纽的性能（即造成瓶颈和全球系统故障的潜在的可伸缩性问题的风险）。

为了应对 EAI 的缺点，这里提出了企业服务总线（ESB）的模式。ESB 集成模式遵循 SOA 方法并提出了一个框架，旨在通过实现用于计算机和网络架构中的著名总线通信拓扑让消费者和提供者之间同步或异步通信。ESB 作为一个中介来促进服务的提供和消费［CHA04］。与集中 EAI 解决方案相比，ESB 的使用增加了可用性、可靠性、性能、可伸缩性和便于维护（如包括性能更好或者更适应服务）和发展（如包括新服务或修改编排逻辑）的功能。ESB 为了应对可扩展性和容错性对 EAI 的限制可以集群或联合。同样，ESB 是基于一个共同的规范化消息传递方法，从而能够应对 EAI 的互操作性的局限性。服务消费者和提供者之间的中介是通过 ESB 提供的消息路由功能来实现的。

ESB 通过提供中介服务运行后端层来促进服务的提供者和使用者之间的协作。该中介服务通过提供足够的互操作性、传输和路由的消息功能，简化了一般静态和 N 个通信提供商与 M 个消费者之间复杂的点到点的联系。同样，在传输和网络层的底层复杂性，以及在通信中间件层特定于协议的格式和数据模型，都通过由 ESB 提供的一个统一的标准 Web 服务的 API 和数据模型被隐藏了。

这种方法一直被小型、中型和大型分布式应用程序所使用。大型分布式应用程序在通信中间件层操作的一个例子是著名的 eBay C2C 系统。eBay 能够处理由成千上万的分布式组件参与的每天数十亿个交易。eBay IT 基础架构是基于 ESB 中间件来设计的，它充当着各种中间件解决方案和分布式组件之间的调解人的角色。

本书主要是为那些有兴趣在 IT 基础架构领域获取知识和技能的人准备的，

他们可以应用IT架构领域的知识，在企业内部和企业间的水平上应用于这种分布式系统。

特别的，本书将介绍通信中间件层的演进，通信中间件层作为一种适应层，隐藏了复杂的分布式应用程序组件［HO 03］，发挥了重要的作用。书中将介绍中间件层的演变，特别是我们将集中精力在已经深刻影响了这类系统设计的主要范式上：SOA［KRA04、MAC06 ERL09］和事件驱动架构（EDA）［VAN06，TAY09］范式。

本书的目标是，根据实际用例以及如何在开发现代化的企业应用程序时有效地应用SOA概念来进行演示。此外，将展示一个重要的由云和自主计算范例所代表的SOA平台的演变。将说明如何把所有这些技术来进行融合，以使得智能SOA平台的建设能够满足大量的非功能性需求，包括可集成性、互操作性、可用性、主动性、可管理性、可扩展性、可移植性、可扩展性和安全性。

为了避免逐条给出理论的定义和概念，本书决定提出一个经过探索的实用方法，命名为基于项目的学习。这种方法是基于概念的合理化和能力分析的基础上实际项目的设计和开发。下面的内容将介绍这种学习方法。

0.2 yPBL：基于项目的学习方法

软件工程（SE）领域涉及复杂的软件开发过程，要求从分析师、设计师、开发人员和架构师在不同领域包括项目管理、沟通、设计开发方面具有高水平的知识和技能。此外，大型软件的设计、开发方法和模式的多样性以及新的软件技术的加速发展要求从事这一领域的人员要持续获取新的知识。

这在分布式系统的设计和开发中是特别重要的，尤其在从互联网开始，包括集中的客户端/服务器模型和多层、点对点、覆盖或云计算的分布式模型等网络部署模型不断改变的情况下。软件工程师和建模师在设计和开发分布式系统时需要面对一个常见的环境演变。这种演变包括通信系统的观点（即新协议在运输和服务、网络层和通信中间件层）和应用程序的观点（即企业应用程序需要的集成性、互操作性、可伸缩性、多租户等）。在这个演变的复杂背景下，是不容易获得所需的IT基础架构的设计和开发所需的知识和技能的。

在软件工程过程的领域中，提出了一些方法以有效地支持开发团队的成员来进行设计和开发软件产品。统一过程（UP）方法是在软件工程领域众所周知的一个方法，它提供了一个基于增量和迭代的序列阶段［JAC99］的有效的过程。统一过程阶段包括针对规范和需求的分析，软件解决方案的设计和实现，软件产品的测试、集成和部署。在增量迭代阶段来计划和执行，每个增量可以在过程中添加新客户需求。同样，错误检测和纠正以及需求变更请求可以被添加在每个迭

代中。按照约定在软件管理计划中，稳定的或实验的软件产品可以在迭代结束时被释放。UP 方法在新的特定的环境中已经被专业化了，这些方法有理性统一过程（RUP）［KRO 03］、企业统一过程［AMB05］或敏捷统一过程［AMB02］。

另一个有趣的专业术语是两个追踪统一过程（2TUP），它提出了面对不断变化的现实需求和软件工程技术，代表着在软件开发领域一个不变的现实［ROQ 04］。2TUP 的过程由于其图形表示也称为“y”过程，提出了一种分化的两个统一过程跟踪：第一个（左）跟踪表示相关的功能性和非功能性需求的软件产品；第二个（右）跟踪表示技术解决方案（如技术、环境和平台）。这种关注点分离有助于软件工程师专注于发现和指定需要满足的需求（左跟踪），同时允许他们探索和选择技术，可用于构建软件解决方案（右跟踪）。一旦功能性、非功能性和技术需求已确定和指定，功能和技术跟踪可以合并以生产软件设计规范。从这一点上，软件产品可以被开发、测试、集成和部署。这一系列的并行和串行化活动将应用增量和迭代过程中提出的方法来执行。这个有趣的方法的好处已经在很多工业土地研究软件项目的应用中被证明了。

为了有效地面对挑战，获取所需的知识和复杂的 SOA 领域的预期能力，这里决定遵循软件工程的方法，这个方法由一个项目或案例研究推动，将提供最基本的需求，推动读者整个学习的过程。

问题式学习（PBL）方法已经成功地应用于不同领域，这种方法的好处已经被广泛证明［SAV 95，BIG 03，SAV 06］。这种学习方法要求学习者理性地、积极地参与进去，由一组初始的问题或项目提供基础指导的整个学习过程［BLU 91，BAR 98，THO 00，HME 04］。

本书将遵循一个名为 yPBL 的基于项目的学习的方法，这非常适合 SE 领域［EXP 13］。yPBL 方法代表了一个专业化的“y”软件工程过程，提出了一种在其设计和开发过程中分离的功能性和非功能性需求的软件产品和技术（参见图 0.3）。

yPBL 方法提出了专业化的软件工程过程如下：

- 基础：在初始阶段，需要对所选案例进行研究分析，以确定系统开发的业务案例（功能性和非功能性需求），并获得初步潜在技术的概述（技术要求）和使用的项目管理活动（流程需求）。在这个阶段，功能、非功能、技术和流程需求都必须确认和验证。

- 细化：在细化阶段需要进行需求分析和潜在解决方案的初步设计（平台独立和面向模式设计）、探索和评价技术的使用、全球项目管理活动和软件产品发布执行的时间安排。在这个阶段，可以计划一些迭代以研究和发展现有技术的基本证明，可以应用这些技术来满足项目需求。这项工作是在基于被命名为基础手册的学习工具上积极协作开发的，使用了 yPBL 方法来审查和评估（一本手册为一个技术、一个方法要求）。

- 建设：在构建阶段，平台的系统架构设计（包括所选择的特定技术）以及功能的实现、测试和产品的发布。在这个阶段，一些迭代可以计划释放增量版本的产品（延续精化阶段的迭代计划）。遵循 yPBL 方法，在前阶段基础手册被用于设计、开发、测试和发布各种软件产品。

- 过渡：维护和系统的进化。这个阶段是产品发布时为了解决异常或应对新需求或约束而产生的。应用 yPBL 方法时，新的手册可以在过渡阶段阐述和生产新产品。

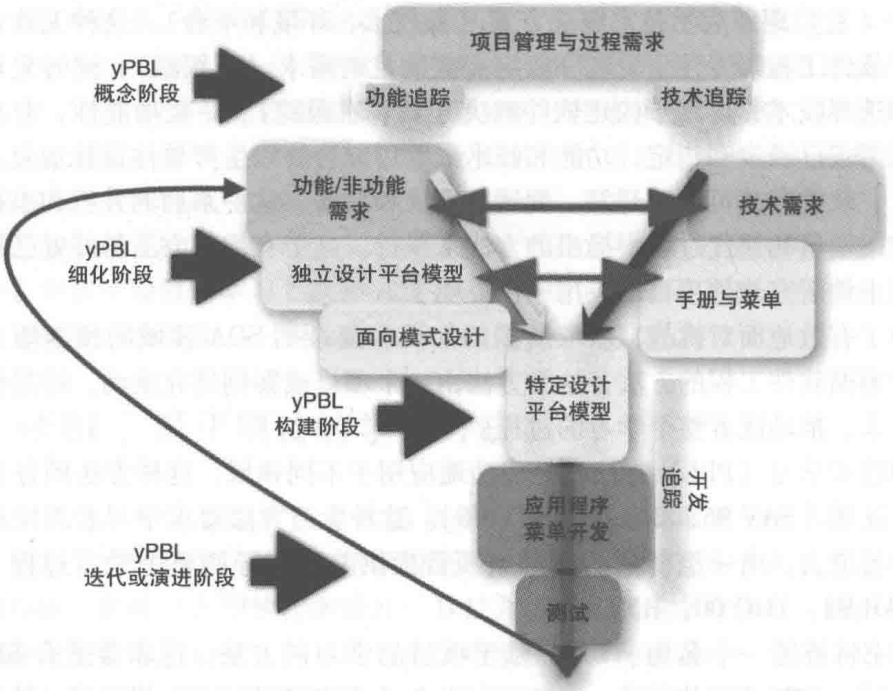


图 0.3　yPBL 方法示意图（包括阶段和实体）

0.3　yPBL 需求驱动矩阵

yPBL 方法提出了以下步骤，需要开始在初始阶段建立可以总结所有被覆盖项目需求的一个矩阵，这个矩阵将用于以下阶段中来推动所有流程活动：

- 用事件驱动的方法来表达功能需求：系统会采用黑盒方法来分析。主要的系统用例将会被确定。

- 在黑盒分析过程中，非功能性需求的收集：预期的质量属性和约束将在系统用例分析中被捕获。

- 识别技术的解决方案：对于功能性和非功能性需求，潜在的技术解决方案需要被确认。