软 件 工 程 技 术 丛 书

HZ BOOKS

软件复用与构件技术系列

# 大规模基于构件的软件开发

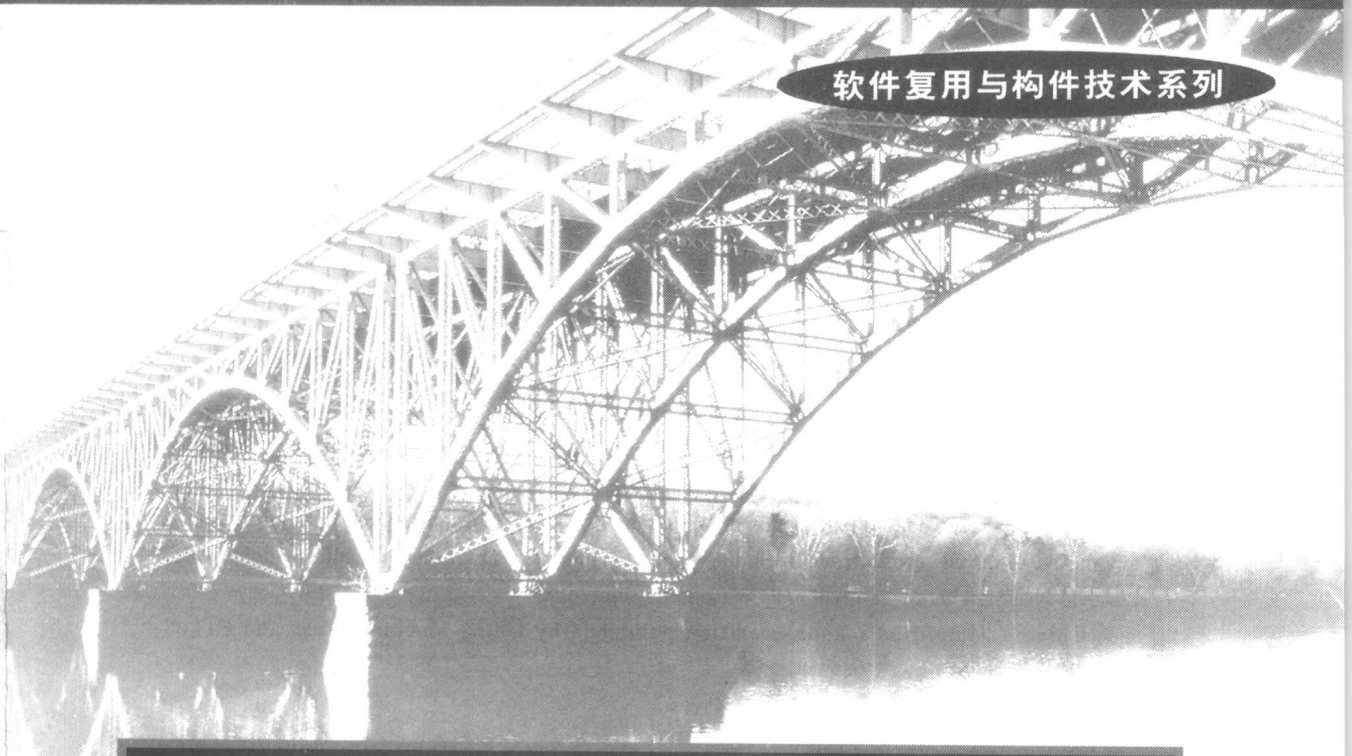
# Large-Scale, Component-Based Development

（美） Alan W. Brown 著　　赵文耘 张志 等译

机械工业出版社 China Machine Press
中信出版社 CITIC PUBLISHING HOUSE

软 件 工 程 技 术 丛 书

软件复用与构件技术系列



# 大规模基于构件的软件开发

# Large-Scale, Component-Based Development



（美） Alan W. Brown 著

赵文耘 张志 等译

机械工业出版社 China Machin

中信出版社 ISHING HOUSE

随着Internet时代的到来，计算机界面临着一系列的变化，这些变化使得传统的软件开发方法不能满足商业界对于软件的需求，软件业面临着越来越大的压力。

本书针对这个背景，提出了大规模基于构件的软件开发方法。其主要内容包括：应用程序开发所面临的挑战，基于构件的开发方法学相关技术和标准，面向构件建模的方法，以及对未来发展方向的展望。

本书作者对计算机界的历史、现状和未来发展趋势的见解十分深刻，把基于构件的开发方法的理论与实践很好地结合在一起。

本书适合IT相关管理与技术人员以及大学计算机及相关专业的本科生、研究生阅读。



**本书版权登记号：图字：01-2002-0526**

**图书在版编目（CIP）数据**

大规模基于构件的软件开发 /（美）布朗（Brown, A. W. ）著；赵文耘等译. -北京：机械工业出版社，2003.7

（软件工程技术丛书 软件复用与构件技术系列）

书名原文：Large-Scale, Component-Based Development

ISBN 7-111-11918-5

Ⅰ.大… Ⅱ.㊀布… ㊁赵… Ⅲ.软件开发 Ⅳ.TP311.52

中国版本图书馆CIP数据核字（2003）第025829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：张金梅

北京牛山世兴印刷厂印刷·新华书店北京发行所发行

2003年7月第1版第1次印刷

787mm x 1092mm 1/16 · 12.5印张

印数：0 001 -5 000册

定价：25.00元

# 序 言

商业杂志连篇累牍地告诉我们，Internet改变了一切。软件工作者长期以来帮助自己以外的所有行业实现了自动化，现在他们也应该在业务模式创新方面与时俱进。

但我们知道怎样去做吗？答案是并不十分清楚。如果我们知道怎样去做，那么软件就不会在商业圈子里得到现在这样的名声——总是延期、超过预算以及实现不了预期的功能；经理们也不必把精力放在解决缺少开发者这样的问题上；并且我们都可以早点下班回家。但我们还是有了一点解决问题的线索，并且可以从最近工具、技术和标准上的进展得到一点希望。

本书在清晰和全面地总结这些进展方面所做的工作非常出色。大多数软件书籍都属于以下两类：面向开发者的技术书籍或面向管理者的管理书籍。虽然分两类书籍并没有什么错，但本书的与众不同之处在于它把两类读者的兴趣集中在一起，同时研究三个内容：应用程序开发、Internet以及构件的兴起。

今天的应用程序开发面临着相当大的压力。在业务方面，我们要支持持续不断发生的变化，这种变化在企业界已经习以为常。在用户方面，在任何时间、任何地点、对任何人，一切都必须可以通过万维网（World Wide Web）得到——并且不必担负把关键业务过程的完整性和安全性暴露给整个世界的危险。在人力资源方面，我们必须忍受软件天才的缺乏，并且在竞争对手把我们最好的开发人员在一夜之间挖走的情况下仍然可以生存。好像这还不够，我们很少能用得到那些在软件工程入门课程上所教的内容。尽管每一个有自尊心的软件工作者的梦想是从零开始建立一个完美的系统，但我们实际所做的是对前辈留给我们的一些“遗产系统”进行一次次的包装。

Internet不仅仅意味着“天涯若比邻”的美妙图景，它也引起了今天的几乎所有企业级开发中的问题，要求每一个应用程序包括所有可能的构件——业务逻辑、一个或多个（通常是数个）图形用户界面、中间件、分发、数据库、安全性、可伸缩性、内置的演化计划，以及与本公司内外其他系统的接口。

基于构件的开发似乎对任何一种解决方案都是必需的。对象技术中深奥的软件工程原理与最初简单的Windows世界中的“用户控件”的结合产生的构件，为清晰、规范地重新封装遗产系统带来了希望，同时也为新开发的系统带来了希望：可以使用模块化、可插拔、可演化并且与世界的其它部分有着良好接口的元素。随着通信标准的不断提出，我们正走在通往应用程序开发新时代的路上。

在这条路上，由于我们仍然必须掌握多种技术、标准、元标准以及使标准之间互联的标准，实际数量如此之大以至于那个古老的笑话（“我们热爱标准，而且关于标准我们最欣赏的是有许多标准可供我们选择”）失去了讽刺意味，并且变成了构件和中间件世界中目前事实的客观描述。Alan Brown做出了非凡的工作来引导读者穿过技术和缩写词的丛林，从EAI到ERP，从EJB到RMI，从UML到XML，从RPC到MOM等等。在整个讨论中，他把技术以及技术在业务中的应用结合在一起，向开发者、管理者，以及两者兼任的人，说明了基于构件的Internet企业级应用程序开发。

——Bertrand Meyer
现代软件工程先驱
Eiffel语言之父

# 译者序

计算机技术对人类社会产生了巨大的影响，许多人的工作和生活已经离不开计算机了。而软件是计算机系统中的关键部分。在过去的10年中，随着Internet的出现，计算机领域面临着巨大的变化。Internet对计算机领域产生的影响是深刻的。商业界对于软件的需求也越来越热切了，他们希望软件系统能够更好地支持他们的业务。但面对这些变化和需求，传统的软件开发方法显得力不从心，因此就需要有一种新的更有效的软件开发方法。基于构件的开发方法由于其自身的特点越来越受到人们的重视，它对于解决软件开发所面临的挑战具有十分重要的意义。现在出现了很多讨论EJB、COM+、CORBA等构件技术的书籍，但对于基于构件的开发方法进行总体讨论的书籍还不多见，本书的出现可以使人们对于基于构件的开发方法有更为清晰和深刻的认识，而不仅仅局限于某种特定的技术。

本书介绍了在Internet时代的电子商务革命，分析了促使软件开发发生变化的业务驱动因素和新技术，指出了当今的软件开发所面临的挑战。针对这些挑战，本书提出了对于开发企业级解决方案非常关键的一些重要技术，这些技术是基于构件的开发方法的基础。作者介绍了基于构件的开发方法的基础理论，以及怎样应用构件技术。关于构件技术的应用，书中对基于构件的开发方法相关的技术、方法和标准做了介绍和比较，并且给出了实例。在最后一部分，作者还对未来的软件开发做了展望。

本书没有对各种技术的细节做过多的介绍，而是从整个计算机业发展的高度对软件开发进行了深刻的论述。本书介绍了基于构件的软件工程（CBSE）的原理，但更多的是结合软件开发的实践发展来展开论述的，这是由于CBSE本来就是软件复用理论与众多的实践相结合而产生的。因此，即使读者没有软件复用理论基础，在读完本书后，对于基于构件的开发方法也会获得清晰的认识。本书的作者对于软件开发的认识是十分深刻的，论述也是十分严谨的，相信读者必定能够从中得到不少的启发。本书不仅适合软件技术人员和学生阅读，对于想了解软件开发的现状和未来发展方向的决策者也会有很大帮助。

参加本书翻译工作的有：赵文耘、张志、李川、奚德、沈钺，张志负责统稿，赵文耘审阅全文。在翻译过程中，我们力求忠实于原文。但是由于译者的知识水平有限，以及时间仓促，难免有翻译不当之处，欢迎读者批评指正。

2002年10月于上海

# 前　言

软件工程正在进入一个新时代。Internet以及与它相关的技术正在改变着顾客、供应商和公司之间相互作用来实施业务活动、通信以及合作的方式。结果是创造了扩展现有业务的巨大机会，可以随时随地的把更多样的和更深层次的信息传送给那些需要这些信息的人，完全新型的商业模式兴起了，如果没有Internet时代所孕育的业务和技术进步，这些商业模式是不能想像的。就像美国商业部长William Daley所说的那样：

"技术正在改造着现在的经济并且正在改变着企业和顾客。这不仅仅是电子商务、也可能是电子邮件、或是电子交易、电子文件。它是经济机遇中的'e'。"

这种影响最近已经被由Booz-Allen&Hamilton的经济学家情报联合会（Economist Intelligence Unit，EIU）所做的研究证实[⊖]。他们调查了500多个高级主管，所问的问题是Internet怎样改变了他们的企业策略。结果显示超过90％的人相信Internet将在未来的三年内改变他们的企业策略或对其产生重要影响。而且，其中许多主管都认为有必要重构他们的业务，以利用其业务环境发生的基本变化。

然而，伴随着这些变化而来的是一系列威胁。许多单位对这些新技术必存畏惧，不知道怎样去利用它们，而且不知道这些技术怎样与现有的在技能和基础设施方面的投资相结合。它们需要的是一个理解Internet时代软件解决方案的概念性框架，以及技术如何驱动这场革命的现实观点。

基于构件的开发（Component-Based Development，CBD）和构件是满足这些需要的方法。我们看到，越来越多的组织开始采用构件作为"封装现有功能、获取第三方解决方案以及建造支持新业务过程的新服务的"方法。最新的分布式系统技术支持并且鼓励用构件的观点来进行应用集成和部署。而且，基于构件的开发提供了一种很适合于今天的以Internet为中心的软件解决方案的设计范型。本书考察了构件和基于构件的开发，以及它们在Internet时代提供企业级解决方案中所扮演的角色。

## 基于构件开发的起源和角色

从根本上说，基于构件的开发（CBD）是一种主要靠组合以前已经开发出来的软件来进行应用程序开发的技术。软件业中的很多人开始把CBD看作是一种令人激动的全新应用程序开发方法，它提供了减少软件开发时间、提高所发布的应用程序质量的希望。

复用以前开发的构件并不是什么新的想法。自从软件开发活动出现以来，就有了在减少创建新软件的工作量方面的努力，包括像宏语言这样小规模的努力，像过程资产库（process asset library）这样大规模的努力。虽然每一种努力都对软件的复用产生了影响，但没有一种达到了人

---

⊖ C.V.Callahan and B. A. Pasternack, *Corporae Strategy in the Internet Age*, Booz-Allen & Hamilton, June 1999.

们所期待的或要求的全部效果。

最近，计算机技术有很多重要的进展。这些进展促使软件业重新考虑怎样来开发软件，并且为计算机支持的软件制品的复用提供了新的机遇。这些进展直接影响到软件业中的每一个人。其中的三项进展特别值得注意：

第一，硬件技术的快速发展已经持续了不止一个年代，结果是计算机技术的性能价格比持续提高。各单位在计算能力上比几年前都已经有了很大提高，这具体体现在分布在单位各级的数目巨大的桌面计算机。

第二，对远程信息的分布式访问的开发代价更低，维护更方便，对用户更加友好，响应更加迅速。这是一系列在支持客户机/服务器构架的分布式基础设施的技术、高吞吐量网络以及分布式数据管理等方面的进展的结果。许多分布式基础设施技术现在都已经很普遍，支持下面一系列协议和标准，包括TCP/IP（transmission control protocol/internet protocol）、RPC（remote procedure call）、HTTP（hypertext transmission protocol）、CORBA（Common Object Request Broker Architecture）以及IIOP（Internet inter-ORB protocol）。

第三，在万维网、Internet以及intranet技术方面的令人兴奋的进展，改变了人们对信息访问和可用性的看法。这导致了许多新的工具、过程和技术的出现来支持新的思维和工作方式。最终用户对应用程序的期望与几年前相比，已经有了很大的不同。

在这些进展的基础之上，一种称为基于构件的开发的新的解决方法产生了。从CBD的最纯粹的本质上来说，它就是利用这些进展来为未来的应用程序提供一个基础设施，这个基础设施可以为单独开发的软件制品的连接提供更加方便的手段。结果，无论是从组织内部还是从世界上任何一个地方得到的软件制品的集成，以及减轻从许多不同地方获取的软件制品的演化、转换和集成的负担而使用可用的计算能力来实现智能协助，它都提供了更大的机会。

在遵循了CBD方法后，软件设计、实现、部署和演化的所有方面都会受到影响。结果，软件项目可以从一个以代码编写和错误修正为中心的过程转变为一个更为受控的组装过程，在这个过程里新代码的开发降到了最低程度，系统的升级变成了替换具有良好边界的系统功能单元的过程。这是许多不同方法和技术的目标，这些方法和技术在软件业中赢得了大量的关注，它们现在都归在了企业应用集成（Enterprise Application Integration, EAI）领域。

采用基于构件的方法的必要性和回报都是很明显的。然而，就像任何新的软件方法一样，现在在希望实现CBD的人员与实现CBD的工具、过程和技术之间存在着一条明显的鸿沟。为了成为一个效果良好的、可重复的、能为每一个应用领域开发大规模的和健壮的解决方案的过程，CBD有很多障碍需要去克服。

目前软件实践者所面临的最紧迫的任务是理解促使转向CBD的业务驱动的因素，获得构筑CBD的基础技术，以及获得为了理解怎样及何时在特定情形下应用CBD技术所需的洞察力。这些都是本书所要回答的问题。

## 本书的目标

本书提供了理解CBD以及把它成功地应用于企业级解决方案所需的相关内容。CBD是一种

新的软件开发方法，它将在未来显著地影响软件开发的实践。本书的目标有以下三个：

（1）对CBD的基本技术进行介绍。有很多技术都对这种方法做出了贡献。其中的每一种都要根据这种技术是怎样产生的、它的主要优势和弱点以及它可能的发展方向来进行研究。

（2）不仅仅是简单地枚举每一种技术的进展，本书还提供了每一种技术怎样对CBD的更大目标做出贡献的整体视图。为了更完整地理解每种技术的相关性和影响，读者可以把这种技术发展应用到需要的环境中去。

（3）虽然我们讨论了这些技术的学术背景，但本书实际上是非常实用的。在讨论这些技术的时候，都结合了它们对现在和未来的大多数软件工程实践的影响。

读完本书之后，读者将理解在软件工程领域的关键技术进展，了解这些内容有助于软件工程师在利用这些技术的时候将处于领先地位。

## 本书的读者

本书的主要读者是信息技术（Information Technology, IT）经理、软件工程实践者，以及对在组织中提高软件工程实践能力感兴趣的项目经理。正在学习高级软件工程课程的学生也可以获得对现代软件工程实践和技术的有价值的观点。本书给这些读者提供了理解范围如此之广的技术所需的背景知识。具备了这些知识，读者将能更好地对相关和感兴趣的技术进行细致的研究。

描述性的文字是针对那些想增长见闻的经理、分析员和类似程序员这样的人。本书不想用详细的业务案例来说明CBD，也不使用需要在你的工作站上录入的代码实例。本书中的内容相当宽泛，提供了丰富的各类读者都感兴趣的材料，并且提供了每一个领域相关的参考文献。

## 本书的结构

本书由四个部分组成。不同的读者可根据兴趣分别阅读。这四个部分如下：

（1）**第一部分**包括电子商务的背景信息，即在Internet时代的企业级解决方案的驱动力。这个部分是任何对现代软件工程发展趋势和技术有兴趣的人进行深入学习的基础。

（2）**第二部分**介绍了基于构件方法的关键要素和CBD在整个软件解决方案生命周期中的角色。这一部分回答了三个基本问题：什么是CBD，什么是构件，怎样定义构件的行为。

（3）**第三部分**描述了CBD的实践。更详细地考虑了各种CBD技术，提供了一个应用CBD技术的说明，并且考虑了这些思想将怎样影响未来的企业级解决方案。

（4）**第四部分**提供了一些关于Internet时代的企业级解决方案的展望和建议。它包括对CBD现在和未来的发展方向的看法，并提供了本书相关的参考资料。

## 致谢

如果没有许多人的辛苦工作、支持和洞察力，本书是不可能完成的。我非常高兴地感谢他

们的贡献。

本书的很多思想基于我现在的和以前的在Sterling Software（现成为Computer Associate International, Inc的子公司。网址为www.cai.com/sterling）的同事的工作。我从很多人的著作以及与他们的讨论中获益匪浅，这些人包括：Balbir Barn、Bill Barnett、John Cheesman、Doug Conley、John Daniels、John Dodd、David Helffrich、Mike Jones、David Marshall和Doug McCammish。除此之外，Bill Gibson、Keith Short、Desmand D' Souza和Alan Wills对本书的方向也做出了重要的早期贡献。特别是，John Cheesman和John Dodd的工作对本书的形式和内容有重要的影响。

很多人都对本书内容的改进提供了帮助。我特别要感谢Paul Allen、Scott Farris和Fred Long在我写作本书的时候对草稿进行了评审。我要感谢他们的友谊、洞察力和耐心。

最后，最衷心的感谢要送给Moria West-Brown。她不间断的支持和鼓励对这本书的完成至关重要。

——Alan W.Brown
alan@CBDEdge.com

# 目　录

# 第一部分

## 电子商务和正在改变的应用程序开发的角色

# 第1章 引 言

现在，我们已经看到很多关于计算机技术对生活影响的描述。随着计算机越来越普及，应用越来越广泛，计算机技术正在改变着我们的生活方式、经营方式以及沟通方式。

虽然这个背景是众所周知的，但有一个关键的方面却少有人注意：这就是它给那些负责开发、部署、维护和演化软件解决方案的组织带来的挑战，这些解决方案使用一些新技术。在本章中，我们将考察软件开发组织在现在的计算机技术开发环境下正在面对的挑战。这使我们去考虑一些主要的业务驱动力，它推动着那些依赖于计算机技术来完成活动的组织。这样，就可以提炼对软件开发组织的主要要求，以及用来开发、部署、维护和演化系统的技术。

## 1.1 动机

我们都经常使用软件系统。它们的可用性（和失败）影响到世界上很大一部分人。然而，经常被忽略的一点是：包括软件系统的开发和维护在内的软件业正在日益成为许多国家国民经济的战略因素。软件业对世界经济、安全和就业有着重要的和引人注目的影响。例如：

- 在1998年，据估计，全世界程序员开发工具的市场份额超过了240亿美元，并且继续以每年高于11%的速度在增长[1]。
- 计算机系统是许多安全关键系统（像飞行控制、医院病人监护和电站管理）的核心。这些系统失效的后果是非常严重的。例如，在1996年，由于欧洲阿里亚纳5型运载火箭的坠毁所造成的损失估计是5亿美元，而这次发射是没有办理过保险的，它坠毁的原因是阿里亚纳火箭的控制软件中有一个错误[2]。
- 截止1999年底，全世界差不多有2亿Internet用户。到2003年，估计将有超过5 亿的人在网上冲浪[3]。这为相关产品和服务提供了一个巨大的潜在市场，并且提供了一个重要的就业机会。

虽然软件的普遍性和重要性是不容怀疑的，而且软件失效导致的后果得到了很大的重视，但对那些从事计算机行业的人来说，最令人不安的可能是许多软件项目缺乏控制和可预见性。许多调查证明，软件开发组织在估计软件费用、确定软件开发和维护活动中使用的目标技术以及在整个软件生命周期中获取可见性和控制能力等活动中，都遇到了很多困难。

结果，在过去的10年里，软件开发逐渐发展成为一个工程学科，鼓励使用更加严格定义的基于良好数学原理的并且有强大的经验基础支持的软件开发方法。这已经被“软件经验工厂（software experience factory）”[4]和软件能力成熟度模型（CMM）[5]等ISO 9000[6]这样的软件认证工作以及一大批新的软件开发方法、工具和技术所支持。虽然这其中每一种

创新都提高了开发和维护软件的能力，但没有什么迹象表明，软件开发组织所面临的问题已经减少。

出现这些问题的一个主要原因是，软件系统的规模和复杂程度都在日益增长。在过去30年里，越来越多的领域都运用了计算机技术。这至少有3点原因：

第一，计算能力有了巨大的提高。计算机主存现在用百兆来计量，辅存用千兆来计量，个人计算机的中央处理器（CPU）的速度用数百MIPS来计量。而且，尽管有反面的预言，但是新技术的进展表明，计算能力的增长是不会立即结束的。摩尔定律（每18个月处理器的速度就会翻一番，而价格会减少一半）将继续起作用。

第二，计算能力的实际价格在不断下降。以很低的价格就可以得到功能强大的桌面计算机，使得计算机技术可以被大多数的业务组织和个人使用。加上计算机网络的发展（包括intranet和Internet），使各个组织可以利用它们所拥有的计算能力而不必关心自己在组织中的物理分布。

第三，信息技术作为许多业务功能的关键部分，已经被越来越多的人所接受。计算机技术不仅是一个重要的业务推动者，它现在经常被认为是一个组织的竞争优势的不可缺少的一部分，显著地影响着组织的经营方式、市场竞争方式以及产品和服务差异化的方式。

## 1.2 软件开发的挑战

计算机技术的能力、可用性和使用的快速增长，已经产生了许多重要的影响。对那些负责提供和维护软件系统的人们来说，最引人注目的影响是：

- 软件提供者没有按照要求的那样完成软件系统的需求（所谓的“软件危机”）。
- 计算机软件的复杂性持续增长，既体现在系统的规模上，也体现在为了满足大量的功能性和非功能性需求所带来的操作的复杂性上。
- 计算机技术变化的速度增加了具有较长生命周期项目的风险，技术的进步使其早期所做的技术决策显得过时了。
- 当业务需求变化时，不能轻易放弃在开发和部署计算机技术上所做的巨大投资——许多组织都有10年以前开发的关键系统在运行。

显然，当软件系统达到这样的规模和复杂程度以后，软件生产中的问题变得非常重要。在规定的预算之内按时完成这样的系统有很多的困难。要设计一个系统，使它可以在操作环境改变、用户需求改变和错误（在这样的系统中是不可免的）暴露出来的时候很容易地演化，这将是一个更艰巨的任务。这些是任何参与软件系统的开发、部署、维护和演化的组织所面临的挑战。

## 1.3 通向未来的关键：控制复杂性和快速适应变化

那些部署软件系统的组织，在努力改进业务活动方式的时候，会面临着许多方面的压力。

它们面临的最大挑战是怎样控制它们要部署的系统的复杂性，而同时又要快速适应变化。对成功的系统部署来说复杂性和变化的组合是最大的风险。

为了更进一步研究这个挑战，考察两个关键因素：复杂性和变化的来源是很有帮助的。

### 1.3.1　控制复杂性

为了成功，开发和部署的系统必须满足需求的变化。当这些需求在数量、多样性和难度等方面增加时，软件系统的复杂性自然就会增加。过去的几年里，在开发一定范围的复杂系统的同时，我们可以看到系统的复杂性有了很大的增加。复杂性的来源至少存在于4类需求中：

*1. 功能需求*

与以前相比，软件系统部署在更为广泛的应用领域中。基于计算机的解决方案越来越多地成为不同领域中技术创新方法的核心，这些领域包括航空、制造业、金融管理和娱乐业等。这其中的每一个领域都给软件开发人员带来了很多新的功能需求，推动能力极限的扩展。

而且，随着技术的演化，业务的所有新领域都开始利用这些技术。一个最好的例子就是电子商务对零售业的影响。Internet技术的进步，使得零售商品的提供者和用户之间全新的贸易模式成为可能。这使那些行业的许多系统的需求发生了重大改变。例如，订单管理系统必须关心多种非传统的订货、支付以及配送零售商品的方法，而它所面对的顾客是那些在Internet上进行在线定购、使用电子货币支付并且想要实时地跟踪发货情况、查询和修改订单的人们。

*2. 非功能需求*

一个软件系统完成已设计好的功能是必要的，但并不是充分的。它还必须精确地、可靠地、可预测地、安全地和快速地完成这些功能。系统要求的这些属性经常称为系统的非功能需求。由于软件系统操作的普遍性和重要性，非功能需求对所开发和部署的系统来说更为重要。

许多组织依赖计算机系统来进行它们的日常活动。对于这些组织而言，如果那些系统不可靠或不能用，那么将产生代价昂贵的后果。因此，一些事关重大的应用通常会明确诸如出错频率、发生问题时的启动次数、并发用户数、事务吞吐量等方面的严格要求。为了满足这些需求，通常会为所提供的解决方案增加很多额外的复杂性。采用的方法包括使用冗余硬件和软件、为异常处理进行容错设计、使用事务日志和回滚来支持热备份以及各种多级别的安全性方案。很明显，这些方法中的每一种都将为解决方案增加复杂程度。

*3. 技术需求*

近年来，开发和部署软件系统的技术有了很多改进。我们可以看一下这个新技术的一个方面，也就是部署应用程序的平台。

软件系统的一个方向是把分布式计算机网络作为它们运行的平台，低价个人电脑加速了这个需求。无论是由一个中央计算机来运行大多数来自远程桌面计算机的访问的普及，还是真正的由互联的计算机组成的端到端系统，为了使应用程序能够在这样的环境下运行，

都会增加应用程序的复杂性。实际上，随着向分布式计算的迁移的加速，这些需求变得更为紧迫，并且可能导致越来越复杂的解决方案。

举例而言，对于一个典型的应用程序来说，可以使用的目标执行平台具有相当的多样性。图1-1显示了这样一个配置。作为这个构架的一个结果，任何要在它上面执行的应用程序必须设计一些方法来处理不同部分之间的通信、跨越不同部分执行动作的同步以及减小任何一个部分失效时的影响。

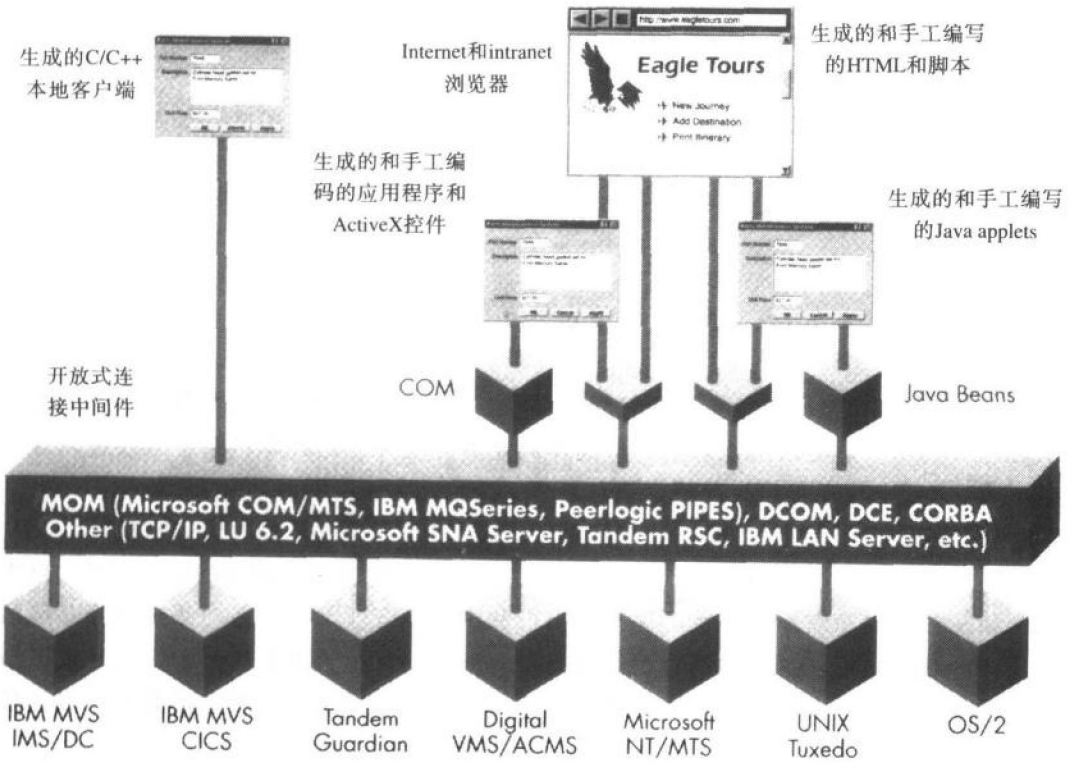



图1-1 一个应用程序的一组目标部署技术

**4. 组织需求**

开发和部署软件系统的组织自身也是跨越组织边界分布的，并且也越来越多地在地理上处于分布状态。对任何以这种方式分布的项目，软件开发过程必须应付和设法利用这种情况。回溯到像IBM/360操作系统这样的软件开发项目的最初时代，Fred Brooks就指出，开发小组的组织形式将对被开发的应用程序的结构和复杂程度产生重大影响[7]。在当前向分布式开发团队、远程办公和离岸开发转移的情况下，这种复杂性只会增加。

**5. 外部软件**

许多组织所遇到的另一个复杂性来源是，使用第三方产品和软件包作为应用程序的组成部分。在过去的几年中，部署第三方系统的组织在迅速增加，这往往是基于对维护费用减少而产生的长期经济效益的预期。然而，大多数组织也对它们的关键应用系统的一些方面