Developing Enterprise Java Applications with J2EE and UML

用 J2EE 和 UML 开发 Java 企业级应用程序

Khawar Zaman Ahmed
Cary E. Umrysh 著
康博 译

Foreword by Grady Booch

清华大学出版社
http://www.tup.tsinghua.edu.cn

用 J2EE 和 UML 开发 Java 企业级应用程序

Khawar Zaman Ahmed
Cary E. Umrysh 著
康 博 译

清华大学出版社

(京) 新登字 158 号
北京市版权局著作权合同登记号：01-2002-3040

内 容 简 介

用 J2EE 开发企业级应用软件是当前一个非常热门的话题，而 UML 是用于建立面向对象系统模型的标准标记法。本书通过一个完整的实例，系统介绍了用 J2EE 开发企业级软件工程时，将 UML 建模技术应用到软件开发过程各个阶段的方法。本书首先介绍了 J2EE 的基本概念和主要技术，以及 UML 中的各种设计视图和基本原理，然后以软件工程的开发流程为主线，系统分析了使用 UML 进行分析、设计，并在使用 J2EE 技术时，结合应用了 UML 的方法和技巧。

本书从最基础的知识着手，非常适合于初学 UML 和 J2EE 的读者；本书后面几章还深入介绍了用 UML 为 J2EE 主要技术建模的内容，对于 J2EE 程序开发人员和软件工程项目管理人员也有很大的参考价值。

图书在版编目(CIP)数据

用 J2EE 和 UML 开发 Java 企业级应用程序/艾默德等著；康博译.—北京：清华大学出版社，2002

书名原文：Developing Enterprise Java Applications with J2EE and UML

ISBN 7-302-05595-5

Ⅰ.用… Ⅱ.①艾…②康… Ⅲ.JAVA 语言—程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2002)第 041123 号

出 版 者：清华大学出版社(北京清华大学学研大厦，邮编 100084)
http://www.tup.tsinghua.edu.cn

责任编辑：夏兆彦

印 刷 者：北京市昌平环球印刷厂

发 行 者：新华书店总店北京发行所

开　　本：787×1092 1/16　**印张**：14　**字数**：358 千字

版　　次：2002 年 7 月第 1 版　2002 年 7 月第 1 次印刷

书　　号：ISBN 7-302-05595-5/TP・3104

印　　数：0001～5000

定　　价：28.00 元

前　言

软件工程的发展史实际上就是抽象的发展史。随着软件越来越复杂，相应地我们也提高了编程语言和方法的抽象级别。因此，我们经历了从 C 到 Java、从结构化方法到面向对象的设计、从类到设计模型又到体系结构框架这一系列改变。

J2EE，即 Java 2 平台企业版就是这样一种框架。它是一种用来部署复杂系统的综合平台。通过提供一组机制（JSP、Enterprise JavaBeans、servlet）和服务（JDBC、JNDI、JMS 和少数 RMI 等）来提高抽象级别，这样开发小组就可以把精力集中在其核心业务逻辑上，而不必关注基础结构的构建工作。

然而，在 J2EE 所提供的技术和业务的实际需要之间还是存在着巨大的语义差距。要跨越这个差距，就要对 J2EE 和具体应用领域系统的合理体系结构有一个基本而又清楚的理解。而统一建模语言（Unified Modeling Language，UML）可以帮助我们做到这一点，因为 UML 在本质上是软件的蓝图语言。当系统变得复杂时，我们有必要将系统的主要元素可视化、具体化、结构化，并将之归档管理，而这些正是 UML 的拿手好戏。

Khawar 和 Cary 在本书中的讲解将帮助您消除这些语义差距。他们讨论了 J2EE 中所有的要点，这使您对它有个初步了解，还谈到怎样才能充分使用 J2EE 的机制和服务。本书还将引导您用 UML 为 J2EE 构建的系统进行建模，这样您就可以更好地思考并与开发小组交流构建优质软件的分析和设计意图。

本书的作者精通 J2EE 和 UML，文中的例子可以帮助您掌握它们。作者构建产品系统的经验全都表露在字里行间，特别是在综合案例分析中。

构建企业级系统有一定的难度，本书将给予您大力的支持。

——Grady Booch
Rational 软件公司

序　言

开发复杂的软件，要求的不仅仅是匆忙地编写代码行。作为业界工程的设计者或开发人员，您必须理解并能够保证开发软件过程中的各个步骤都成功，这些步骤包括体系结构、分析和设计技术、开发过程、可视化建模及基本的技术。

本书从 J2EE 开发角度考虑，将所有这些不同的元素组合在一起，为读者提供了一个完整的方案。本书将着重讨论下面这些重要问题：

- 统一建模语言(UML)，它与 J2EE 开发的关系
- 如何将 Java 和 UML 相互联系
- 软件体系结构中的关键概念是什么
- 软件开发过程如何才能适应 J2EE 软件开发
- 分析和设计方法如何才能帮您实现良好的 J2EE 应用设计
- J2EE 的关键技术是什么，如何将它们组合在一起
- 如何在 J2EE 开发中使用 UML

本书并没有重新设计新的方法，而是将已有的工作组合在一起，例如 Jim Conallen 的 Web 建模配置文件和用于 UML/EJB 映射技术规范的 Sun Java 技术规范第 26 条。

为了巩固所讲内容，本书通过提供一个 J2EE 应用程序开发实例来学习如何使用 Rational 统一过程(Rational Unified Process，RUP)和 UML，并提供了实际的实现。本书还提供几条改良建议，帮助您继续探讨 UML 和 J2EE 技术。

本书适用对象

本书适用于所有对 UML 和如何在 J2EE 开发中使用 UML 等知识感兴趣的读者。J2EE 应用程序开发人员可以通过本书学习将 UML 应用到 J2EE 应用程序开发的知识。专门从事 UML 开发的人员也可以通过学习在 UML 环境中使用 J2EE 而受益。想学习 UML 和 J2EE 的软件专业人员通过学习会提高自己的编程效率。

学习本书后，您将能够：

- 有效地利用 UML 来开发 J2EE 应用程序
- 从理论上掌握关键的 J2EE 技术
- 知道什么时候使用 Model 1 和 Model 2 体系结构，并确定在什么情况下使用值对象和会话 Bean 链这样的模型比较合适。
- 理解软件体系结构概念，例如分解、分层、组件、架构、模型和层等
- 在 J2EE 工程中使用案例分析、分析对象发现、以及将设计转换为 J2EE 工程的分析

- 理解软件开发过程的概念，以及当前流行的软件开发过程的基本知识
- 学习如何在 J2EE 工程中开始使用 RUP

本书所讨论的 Java 语言范围只限于将主要的 Java 概念映射到 UML 中。因此，我们假定读者比较熟悉 Java 语言(如果了解 C++或类似的语言，从例子中就足以理解这些基本概念)。有 UML、J2EE 或企业级应用开发方面的知识或经验并不是学习本书的先决条件，但总是有所帮助的。如果您是第一次接触 UML 和 J2EE，完整地按顺序阅读本书，将可以从中获得很多知识。

如果您已掌握了 UML，只是想学习一下 J2EE(或如何将 UML 应用到 J2EE 中)，就可以跳过前面几章，直接学习第 9~16 章。

反过来，如果您掌握了 J2EE，主要想学习 UML，您可以将精力集中在第 1~8 章，然后浏览一下本书的剩余部分就行了。

如果使用好的建模工具，并应用可视化建模来解决您的实际问题，那您将会获得最好的结果。

各章小结

第 1 章：对企业级软件开发和相关技术进行高度概括

第 2 章：讨论 J2EE 的一些基本知识，概述形成 J2EE 的基本技术和 API

第 3 章：概述 UML 及 UML 基本知识

第 4 章：概述 Java 语言在 UML 中的映射，并讨论一些基本的 UML 结构

第 5 章：介绍软件开发过程的概念，并概述本书所采用的方法

第 6 章：体系结构是优秀软件的一个重要方面，本章介绍了软件体系结构的概念并概述软件体系结构中的一些概念

第 7 章：表明该如何应用 UML 用例来更好地理解客户的需求。不管开发出来的软件有多优秀，如果它没有满足客户的需求，那它就是失败之作

第 8 章：深入分析客户需求，并创建案例分析的最初设计方案。本章讨论如何将所汇集的客户需求在软件中实施

第 9 章：为剩余章节中所讨论的 J2EE 技术打下基础

第 10 章：概述 Java servlet 技术，讨论如何用 UML 为它们建模，然后在案例分析中显示一个 UML 和 servlet 的典型应用程序。Java servlet 适用于面向请求响应的 Web 范例

第 11 章：讨论什么时候使用 JSP，以及如何在示例工程中使用 JSP。JavaServer Page(JSP)将 servlet 的功能与 HTML 页面的灵活性组合在一起

第 12 章：讨论如何在中间层使用会话 Bean，以及如何才能最好地建模并使用它们。会话 Bean 是 J2EE 所提供的 3 种 Enterprise Bean 之一。本章还谈到了在案例分析环境中会话 Bean 的应用

第 13 章：着重讲解实体 Bean 概念，它的优势和问题所在，以及如何用 UML 来对它进行有效建模。实体 Bean 提供了一种方便的方式来体现已存储数据

第 14 章：讨论消息驱动 Bean，以及如何用 UML 来对它们进行建模。消息驱动 Bean 是 J2EE Enterprise JavaBean 技术规范中的新增内容

第 15 章：讨论如何用 UML 帮助装配并部署分布式应用程序

第 16 章：讨论本书所使用案例的具体内容，包括共同需求、限制等

术语表：一些专门术语和它们的含义

目　　录

第 1 章　企业级软件概述 …… 1
1.1　什么是企业级软件 …… 1
1.2　企业级软件的演变 …… 3
1.3　企业级软件和基于组件的软件 …… 4
1.4　小结 …… 5
第 2 章　J2EE 简介 …… 6
2.1　什么是 Java 2 企业版平台 …… 6
2.2　J2EE 简史 …… 7
2.3　使用 J2EE 的原因 …… 8
2.4　J2EE 简介 …… 9
2.4.1　技术 …… 10
2.4.2　API …… 13
2.4.3　其他 J2EE 技术和 API …… 14
2.5　小结 …… 15
第 3 章　UML 简介 …… 16
3.1　UML 概述 …… 16
3.2　结合使用 J2EE 和 UML 的原因 …… 18
3.3　利用 UML 对 J2EE 建模遇到的难题 …… 19
3.4　UML 中的扩展机制 …… 20
3.4.1　模板 …… 20
3.4.2　附加值 …… 20
3.4.3　约束 …… 21
3.5　J2EE UML 建模的方法 …… 21
3.6　小结 …… 22
第 4 章　UML 和 Java …… 23
4.1　表示结构 …… 23
4.1.1　类 …… 23
4.1.2　变量 …… 24
4.1.3　方法 …… 25
4.1.4　对象 …… 25
4.1.5　接口 …… 26
4.1.6　包 …… 26

4.2 表示关系 …… 27
4.2.1 继承 …… 27
4.2.2 实现 …… 27
4.2.3 相关性 …… 28
4.2.4 关联 …… 29
4.2.5 聚合 …… 32
4.2.6 合成 …… 33
4.2.7 自反关系 …… 33
4.3 小结 …… 34
第 5 章 动作概况 …… 35
5.1 什么是软件开发过程 …… 35
5.2 开发软件的常用方法 …… 35
5.2.1 即兴开发方法 …… 35
5.2.2 瀑布法 …… 36
5.2.3 迭代法 …… 36
5.2.4 Rational 统一过程 …… 37
5.2.5 ICONIX 法 …… 39
5.2.6 OPEN 法 …… 40
5.2.7 Extreme Programming/功能驱动开发 …… 40
5.3 本书所用的方法 …… 41
5.4 主要的工作 …… 42
5.4.1 第 6 章：体系结构 …… 42
5.4.2 第 7 章：分析客户需求 …… 42
5.4.3 第 8 章：总体设计 …… 42
5.4.4 第 10—15 章：具体设计 …… 42
5.4.5 第 16 章：案例分析 …… 42
5.5 小结 …… 42
第 6 章 体系结构 …… 44
6.1 软件体系结构的概念 …… 44
6.2 使用体系结构的原因 …… 45
6.3 企业级应用体系结构中的主要概念 …… 46
6.3.1 分解 …… 46
6.3.2 组件 …… 47
6.3.3 框架 …… 48
6.3.4 模型 …… 49
6.3.5 分层(layering) …… 51
6.3.6 层(tier) …… 53

6.4 开发软件体系结构的方法 …… 53
6.4.1 J2EE 视图体系结构 …… 54
6.4.2 4+1 视图模型体系结构 …… 54
6.4.3 Hofmeister 等：4 种视图体系结构 …… 54
6.5 综合应用 …… 55
6.6 小结 …… 55
第 7 章 分析客户需求 …… 56
7.1 为什么要进行软件分析和设计 …… 56
7.2 问题分析 …… 57
7.3 用例建模 …… 57
7.4 标识参与者 …… 58
7.5 用例查找 …… 59
7.6 用例图 …… 60
7.7 用例关系 …… 61
7.7.1 包含 …… 61
7.7.2 扩展 …… 61
7.8 顺序图 …… 62
7.9 活动图 …… 63
7.10 小结 …… 65
第 8 章 总体设计 …… 66
8.1 用例分析 …… 66
8.2 用例实现 …… 66
8.3 精化用例描述 …… 67
8.4 顺序图 …… 68
8.4.1 边界对象 …… 68
8.4.2 实体对象 …… 69
8.4.3 控制对象 …… 70
8.5 协作图 …… 71
8.6 类图 …… 72
8.7 聚合分析类 …… 74
8.8 打包 …… 75
8.9 小结 …… 76
第 9 章 J2EE 技术概览 …… 77
9.1 J2EE 概况 …… 77
9.2 Servlets …… 78
9.3 Java 服务器页面 …… 78
9.4 企业级 JavaBeans(EJB) …… 78

9.5 会话 Bean ····· 79
9.6 实体 Bean ····· 79
9.7 消息驱动 Bean ····· 79
9.8 组装和部署 ····· 79
9.9 案例分析 ····· 80
9.10 小结 ····· 80

第 10 章 Servlet ····· 81
10.1 Servlet 简介 ····· 81
10.1.1 一般用途 ····· 82
10.1.2 最利于处理小型任务 ····· 82
10.1.3 J2EE 版本 ····· 83
10.2 Servlet 生命周期 ····· 83
10.2.1 生命周期方法 ····· 83
10.2.2 便利的方法 ····· 84
10.2.3 需要的方法和标记值 ····· 84
10.3 处理请求 ····· 85
10.4 响应的产生 ····· 86
10.5 HTTP 请求处理程序 ····· 87
10.5.1 高级处理程序方法 ····· 87
10.5.2 HTTP 请求快速指南 ····· 87
10.6 RequestDispatcher 接口 ····· 88
10.7 在 UML 中为 Servlet 建模 ····· 89
10.8 为 Servlet 的其他方面建模 ····· 90
10.8.1 Servlet 转移 ····· 90
10.8.2 Servlet 包含 ····· 91
10.8.3 ServletContext ····· 92
10.8.4 Servlet 会话管理 ····· 92
10.9 Servlet 部署和 Web 归档文件 ····· 93
10.10 在企业级应用中标识 Servlet ····· 93
10.11 小结 ····· 96

第 11 章 Java 服务器页面(JSP) ····· 97
11.1 JSP 简介 ····· 97
11.1.1 JSP 的典型用法 ····· 98
11.1.2 模型 1 和模型 2 的结构 ····· 98
11.1.3 JSP 与 Servlet ····· 99
11.2 JSP 剖析 ····· 99
11.2.1 模板数据 ····· 99

11.2.2 JSP 元素 100
11.2.3 JSP 可隐式访问的对象 102
11.3 标记库 102
11.3.1 标记处理程序类 102
11.3.2 标记库描述符 103
11.4 JSP 和 UML 104
11.4.1 为客户端关系建模 105
11.4.2 为服务器端关系建模 105
11.5 企业级应用中的 JSP 107
11.6 小结 109
第 12 章 会话 Bean 111
12.1 企业级 JavaBean 简介 111
12.2 EJB 视图和 UML 112
12.2.1 在 UML 中表示企业级 JavaBean 113
12.2.2 客户视图 114
12.2.3 内部视图 115
12.3 会话 Bean 115
12.3.1 用途广泛的 Bean 115
12.3.2 J2EE 版本 116
12.4 会话 Bean 和通话状态管理 116
12.5 实例钝化 118
12.6 事务处理 119
12.6.1 事务分类 120
12.6.2 托管 Bean 式事务 121
12.6.3 托管容器式事务 121
12.6.4 SessionSynchronization 接口 121
12.6.5 无状态会话 Bean 的限制 122
12.6.6 事务属性 122
12.6.7 为事务建模 123
12.7 会话 Bean 技术 123
12.7.1 Home 接口 124
12.7.2 Remote 接口 124
12.7.3 实现类 125
12.8 为接口行为建模 126
12.9 会话 Bean 生命周期 128
12.10 会话 Bean 常用方案 128
12.11 为会话 Bean 关系建模 130
12.11.1 会话 Bean 和简单 Java 类 130

12.11.2 会话 Bean 和 JavaBeans …… 131
12.11.3 会话 Bean 和 Servlets …… 131
12.11.4 会话 Bean 和 Java 服务器页面(JSP) …… 132
12.11.5 会话-会话关系 …… 132
12.11.6 会话 Bean 的继承 …… 133
12.12 管理性能 …… 133
12.13 本地客户端 …… 134
12.14 在企业级应用中标识会话 Bean …… 135
12.15 小结 …… 136
第 13 章 实体 Bean …… 138
13.1 实体 Bean 简介 …… 138
13.1.1 粗粒度(Coarse-Grained)的业务对象 …… 139
13.1.2 实体 Bean 应用越来越普及 …… 139
13.1.3 J2EE 版本 …… 140
13.2 实体 Bean 视图和 UML …… 140
13.2.1 客户端视图 …… 140
13.2.2 内部视图 …… 141
13.3 持久性 …… 142
13.4 抽象持久性 …… 143
13.4.1 抽象持久性模式 …… 144
13.4.2 EJB 查询语言(EJB QL) …… 145
13.4.3 持久性管理器 …… 145
13.5 托管容器式关系 …… 146
13.5.1 多样性 …… 146
13.5.2 方向性 …… 147
13.5.3 在 J2EE 1.2 中复制托管容器式关系 …… 147
13.5.4 本地关系 …… 147
13.6 实体 Bean 技术 …… 147
13.6.1 Home(本地)接口 …… 148
13.6.2 Remote(远程)接口 …… 149
13.6.3 主键类 …… 149
13.6.4 实现类 …… 150
13.6.5 永久域 …… 151
13.7 实体 Bean 生命周期 …… 152
13.8 实体 Bean 常用脚本 …… 154
13.9 为实体 Bean 关系建模 …… 154
13.9.1 实体 Bean 与其他 Java 类 …… 154
13.9.2 实体 Bean 与 JavaBeans …… 155

13.9.3 值对象方法 156
13.9.4 实体 Bean、Servlet 和 JSP 157
13.9.5 实体 Bean 和会话 Bean 158
13.9.6 实体 Bean 与实体 Bean 关系 158
13.10 在企业级应用中标识实体 Bean 158
13.11 小结 161

第 14 章 消息驱动(Message-Driven) Bean 162
14.1 介绍消息驱动 Bean 162
14.1.1 Java 消息服务 163
14.1.2 EJB 中的 JMS 和消息驱动 Bean 163
14.1.3 使用通信和消息驱动 Bean 的原因 163
14.1.4 使用消息驱动 Bean 的时间 163
14.1.5 J2EE 版本 164
14.2 消息驱动 Bean 视图和 UML 164
14.2.1 客户视图 164
14.2.2 为消息驱动 Bean 使用 UML 的好处 165
14.2.3 对消息建模 165
14.2.4 建模目标 166
14.3 消息驱动 Bean 技术 167
14.3.1 事务处理 167
14.3.2 实现类 167
14.4 消息驱动 Bean 的生存周期 168
14.5 使用消息驱动 Bean 的条件 169
14.6 为消息驱动 Bean 关系建模 169
14.6.1 为消息驱动 Bean 与其他类间的关系建模 169
14.6.2 消息驱动 Bean 和其他 J2EE 技术 169
14.7 在企业级应用中使用消息驱动 Bean 170
14.8 小结 171

第 15 章 装配和部署 172
15.1 对组件建模 172
15.2 J2EE 技术建模组件 172
15.2.1 表示 Web 组件 173
15.2.2 表示 EJB 174
15.2.3 企业级应用的组件建模 175
15.3 部署建模 176
15.4 使用跟踪能力 177
15.5 企业级 Java 应用程序中的装配和部署 178

15.6 小结 …… 181
第 16 章 案例分析 …… 182
16.1 案例分析背景 …… 182
16.2 问题陈述 …… 183
16.3 使用在线银行业务例子的原因和要求 …… 183
16.4 HomeDirect 要求 …… 184
16.4.1 查询服务 …… 184
16.4.2 转账支付服务 …… 184
16.4.3 交易服务 …… 185
16.4.4 管理服务 …… 185
16.5 起始阶段 …… 186
16.5.1 最初重现 …… 186
16.5.2 重现计划 …… 186
16.5.3 HomeDirect 参与者 …… 187
16.5.4 HomeDirect 用例 …… 187
16.5.5 用例图 …… 189
16.5.6 交互作用图 …… 190
16.6 加工阶段 …… 190
16.6.1 加工重现#1 …… 191
16.6.2 具体的序列图 …… 191
16.6.3 类图 …… 194
16.6.4 打包图 …… 195
16.6.5 组件相关性图 …… 195
16.6.6 加工重现#2 …… 196
16.6.7 加工重现#3 …… 197
16.6.8 在这个重现中实现的用例 …… 197
16.7 剩余阶段 …… 199
16.8 小结 …… 200
术语表 …… 201

第1章　企业级软件概述

- 什么是企业级软件
- 企业级软件的演变
- 企业级软件和基于组件的软件
- 小结

如果您曾听说过 B2B 和 B2C 这样的术语，那说明您对企业级软件已有了一定程度的了解。B2B 和 B2C 正是企业级软件中最流行的两种表现形式。

在第 1 章中，我们将深入探讨企业级软件，以及伴随着企业级软件的一些机遇和挑战。

1.1　什么是企业级软件

“企业”是指一个为了实现某个共同目的而在一起工作的人或实体的组织。这些组织的形式和大小并不限定，有大有小、是盈利性的或非盈利性的、可能是政府机关也可能是非政府机构。

不过，当某些人使用术语“企业”时，极有可能指的就是大型盈利性组织，如 Intel、通用汽车、Wal-Mart、美洲银行或 eBay 等等。

企业通常都有一些共同的需求，例如信息共享与处理、资产管理与跟踪、资源规划、客户及用户管理、商业机密保护等。而术语“企业级软件”就是能够支持企业共同需求的这类软件的总称。

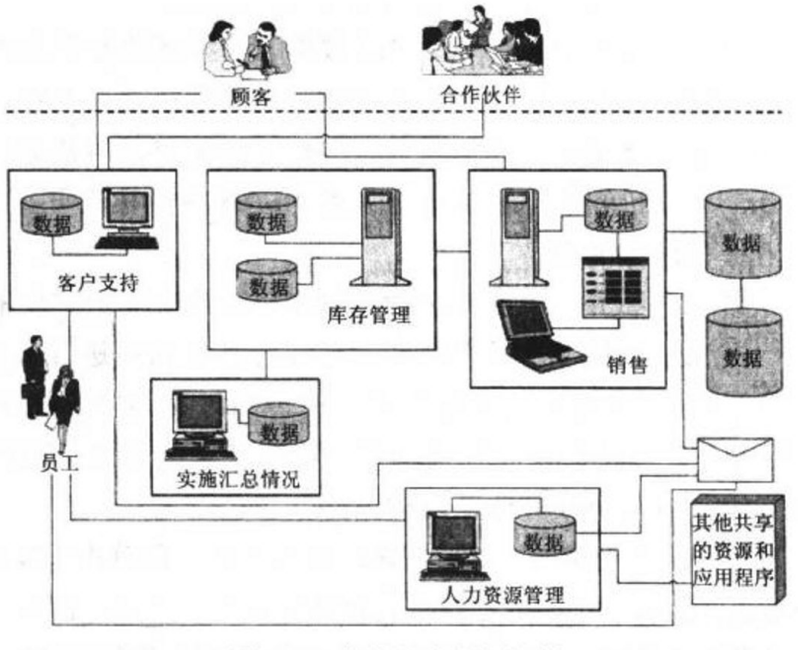

图 1-1　企业和企业级软件

图 1-1 显示的是一个企业级软件配置图，它本质上是不同系统的集合。软件根据组织中不同功能的部门(例如销售、人力资源等)组织起来。此外还有一个防火墙，可以保护企业内部数据，防止未被授权的用户访问。有些软件系统，例如用于销售和存货管理的软件之间会交互作用，不过绝大多数软件都是独立的。

今天的企业级软件可以由多个不同的部分组成，但企业已逐渐认识到，只要合乎企业利益，很有必要将各个分散的系统进行良好的集成，以尽可能相互支持。B2B 和 B2C 就是这种集成和相互支持的优秀应用。

企业希望对集成后的企业级软件的具体应用如下：

- 通过集成企业的客户支持和本身的产品知识，企业可以利用 Web 为它的客户提供更新更好的服务。
- 将企业售货机联网，企业可以获得更多的在线客户。
- 将销售管理系统和存货系统相链接，企业可以设计特定的低成本的 Web 销售渠道，这样可以进入未曾涉足的市场领域。
- 如果给企业员工所使用的服务提供一个前端，例如内部办公用品订货系统，将它与会计系统连接在一起，企业就可以降低总体开支并提高员工的工作效率。
- 在线使用企业 HR 系统，可以让员工根据他们自己的健康状况和 401(K)条款进行更多的选择，这样可以降低企业整体的管理费用。
- 使企业的人力资源密集型操作自动化，并使它可用于任何时间任何地点，在降低整体运营费用的同时，企业还可以给它的客户提供更好的服务。

开发企业级软件所面临的难题

成功的企业总是不断扩大规模、雇用更多的员工、拥有更多的客户和更多的 Web 站点访问量、有更高的销售额和收入、更多的分部等。为了适应这种增长，企业级软件也必须相应地升级，以适应规模更大的企业及更复杂的运作过程。

企业在其成长过程中会受到各种制约，比较常见的一种制约是计算机硬件跟不上企业处理需求的增长。另一种常见制约是企业无法将许多人安置在同一个实际位置甚至地理位置。因此，关于分布的难题就摆到了我们面前。多台计算机可以解决数据处理的需要，但却会引起软件的分布问题。新的建筑物或地理位置可以解决新增工作场所的需要，但却很难为不同地点的企业提供同等的服务。

将原来独立的系统连接起来，以提高企业的效率，这是一个最主要的挑战。原来的系统一般都是为特定目的而设计的，并没有特别考虑要与其他系统进行集成。例如，人力资源管理部门与财务部门并没有太多的交互作用，销售部门与客户支持部门也没有什么联系。这种不统一的软件开发方法经常会导致客户只购买某些优秀的关键产品以满足特定的需求，这样通常就很难集成整个软件体系结构。

与之相关的难题是，企业级软件需要处理多厂商的环境。部分由于发展的需要，部分出于要求所限，企业级软件会从多个厂商处获得具有类似作用的产品。例如，虽然 HR 应用程序可能基于 Oracle 8i 数据库构成，但客户支持应用程序却可能依赖于 Microsoft SQL Server。