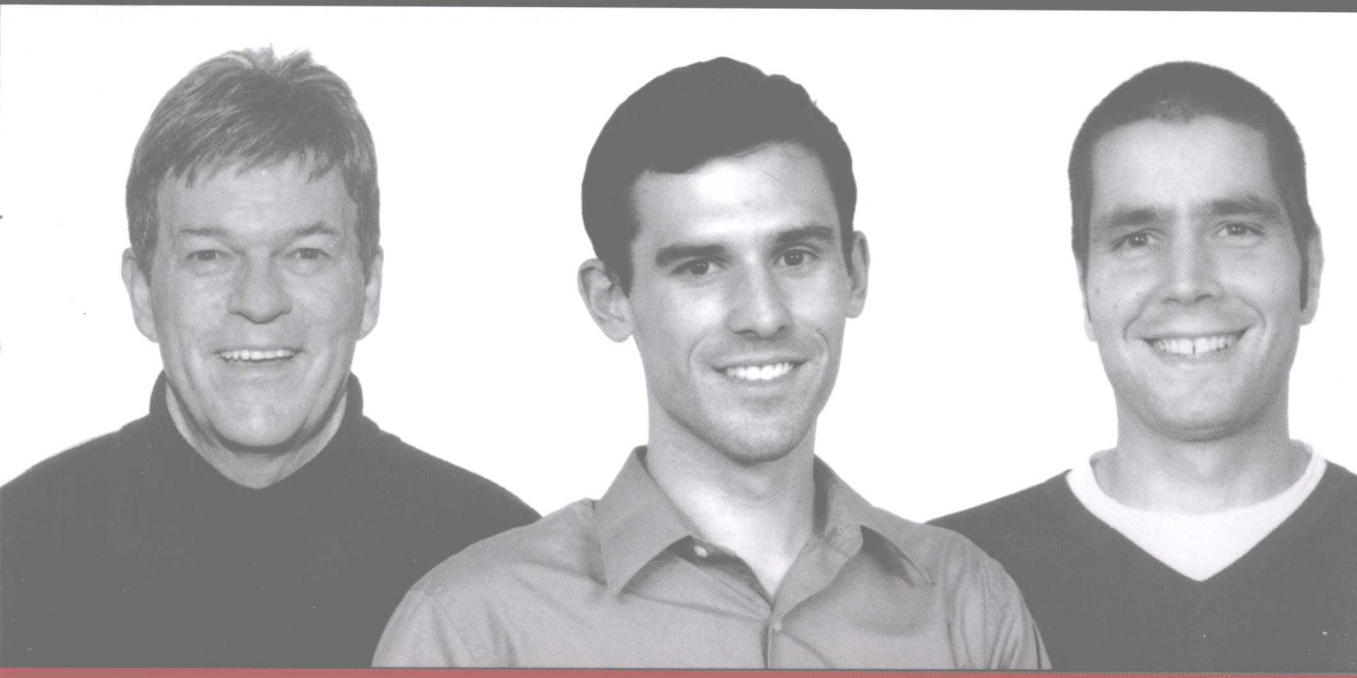
PROGRAMMER TO PROGRAMMER™



Professional Rich Internet Applications: AJAX and Beyond

Rich Internet Application高级编程：后AJAX时代

Dana Moore
(美) Raymond Budd 著
Edward Benson
张云 付勇 译

wrox™
WILEY
清华大学出版社

Rich Internet Application 高级编程：后 AJAX 时代

Dana Moore
(美) Raymond Budd 著
Edward Benson

张云 付勇 译

清华大学出版社

北 京

Dana Moore, Raymond Budd, Edward Benson
Professional Rich Internet Applications：AJAX and Beyond
EISBN：978-0-470-08280-5

图书在版编目(CIP)数据

Rich Internet Application 高级编程：后 AJAX 时代/(美)摩尔(Moore,D.)，(美)布德(Budd,R.)，(美)本森(Benson,E.)著；张云，付勇 译. —北京：清华大学出版社，2009.1

书名原文：Professional Rich Internet Applications：AJAX and Beyond

ISBN 978-7-302-18922-0

Ⅰ.R…　Ⅱ.①摩…　②布…　③本…　④张…　②付…　Ⅲ.主页制作—程序设计　Ⅳ.TP393.092

中国版本图书馆 CIP 数据核字(2008)第 179496 号

责任编辑：王　军　徐燕萍
装帧设计：孔祥丰
责任校对：成凤进
责任印制：孟凡玉
出版发行：清华大学出版社　　地　　址：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　邮　　编：100084
社　总　机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：清华大学印刷厂
装 订 者：三河市新茂装订有限公司
经　销：全国新华书店
开　本：185×260　印　张：31.75　字　数：773 千字
版　次：2009 年 1 月第 1 版　　印　次：2009 年 1 月第1 次印刷
印　数：1～4000
定　价：68.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：(010)62770177 转 3103　　产品编号：025084-01

前　言

前言简要介绍本书的主题——RIA(Rich Internet Application)体系结构及其演变。为了更好地服务读者，我们尽量混合 3 种元素来实现以下目标。

- 理解 RIA 与前一代 Web 应用程序之间的不同，以及这种不同的重要性，无论您是集体开发人员还是个人软件开发人员。
- 提供重要的支持代码示例，这些示例完整而又重要，但非常简单，用一到两页代码就可以表示。
- 描述并展示重要架构和 API。

0.1　本书读者对象

本书面向对现实问题的实际解决方案感兴趣的软件开发人员。本书阐述了使用激活技术来创建具有与桌面系统匹配特性的分布式系统的方法。不管您是擅长 Web 开发、服务器体系结构还是敏捷语言，都会发现本书的价值。

因为示例是用 Python、Java 和 Ruby on Rails 写的，如果您熟悉其中的某种语言那就最好。我们假设您有开发和部署 Web 应用程序的经验，能够阅读基本的 UML 图表。最后，您应该熟悉基本的数据库管理操作，虽然各章都清楚地概述了必要的数据库管理步骤。

0.2　本书的重点

本书的作者都是专业的公共软件开发人员，而一般的理解代码和 API 的最快速的方法是阅读实际代码示例，但这会导致一连串说教式的示例。虽然这种方法可以说明主题的重点，但最终却无法实现，因为这些简单示例不能构建任何脱离于所述上下文环境的实例。因此，本书重点介绍一个有趣而且功能完整的应用程序，将它作为贯穿本书的主题。该应用程序还有助于进一步掌握 RIA 设计、架构和实现方式。

同时，还可以构建顶点应用程序，我们称之为 ideaStax。ideaStax 是一种想法捕获工具，它让用户——不管是个人用户还是公共用户——能够记录、标记、搜索、组织和提出想法。想法是 ideaCards 不可分割的单元，ideaCards 是 ideaStax 的构建代码块。ideaStax 还有其他用途。首先，如果从头开始构建应用程序来反映 RIA 的概念和性能，它可以说明这些应用程序在性能和特征方面的变化。在某种程度上，可以将 ideaStax 作为桌面应用程序(例如，Microsoft 的 PowerPoint)的重新预想。

除了从 Wrox 出版商站点(www.wrox.com)获得 ideaStax 的代码之外，作者还开发了一个站点(ideaStax.com)，您可以在那里提出想法和代码，展示方法，以开源软件的最佳惯例参与改进应用程序及其性能。如果您提出的想法有价值，就会成为该项目的提交者。

0.3 本书的目标

从几十年前开始，开发人员就相信有些东西客观上是正确的，甚至不容置疑。我们将自己当成特定开发和部署“部落”的成员。不只是将自己当成开发人员，还要将自己当成“Java 开发人员”或“Microsoft .NET 开发人员”，并接受部落成员暗示的限制和技术约定。在 Java 中，这意味着“一种语言，多个平台”，而在 Microsoft Windows 32 API 中，则是“多种语言，一个平台”。无论如何，从传统来看，写富客户平台就意味着要求开发人员做出让步，并让终端用户管理安装和更新，公司 IT 部门也要参与进来。

作为软件开发人员，不可能没有使用过富客户 API 为 Windows、MAC OS/X 或 Linux 编写代码。因此我们都很熟悉富客户平台(RCP)模型。在过去 30 年中，我们在计算机上编写、使用、管理或部署的几乎每个应用程序都符合 RCP 的性能和限制。如果像 RCP 开发人员那样思考，就会认为操作系统、数据访问和存储、应用程序的用户体验紧密相连是理所当然，这可以说明某种应用程序(例如，字处理器)的响应度。

现在这种情况正在改变，而且在迅速改变。到了 2005 年末，AJAX(Asynchronous JavaScript and XML，异步 JavaScript 和 XML)中出现了一种“逼近”思想。AJAX 是使用一些常用成熟技术组合应用程序的概括性术语，但似乎不仅仅是其组成部分的总和。现在人们逐渐认识到，构造转变肯定会改变我们开发和部署应用程序方法的所有基础。Web 正在经历从新条款集合、静态表格和公告牌到自身位置不可知的计算平台的深刻变革。

同时，我们将探究基于浏览器的应用程序中的这些变更元素——从更大推动力和响应度到更短开发周期再到对社会网络更加依赖。短语“在平台层次以上编程”已经与 RIA 紧密相关。我们不会创造习语，但我们用多种语言写的多个工作示例说明这个概念对于 Web 2.0 应用程序(甚至更高版本应用程序)的重要性。

我们的目标是为集体和个人开发人员利用这种新兴方法做好准备。随着客户端软件的衰落，RIA 这个新事物逐步出现，这让开发人员和终端用户都能利用今天这个互连的环境。利用这种新方法的开发人员会成功，并且会处于其专业领域的前沿。

学习本书中的这些技术方法，理解 RIA 如何不同以及为何不同，这会让您从完全不同的视角观察这个世界。首先可以发现，应用程序的大多数“动作”实际上没有在用户的桌面上发生，而是在某个位置的服务器上发生；用户通过其浏览器看到的和体验到的东西都非常单纯——由浏览器的约束协调解决。

RCP 应用程序非常丰富、灵敏，在 Web 上通常不可能有这种响应度。一般来说，易于部署到 Web，这种简单性在您能够提供的体验与用户可以从桌面应用程序中获得的体验之间存在差异。现在，从 Web 开始，这种差异正在逐渐消失。新一代应用程序正在出现，其中富客户和 RIA 之间的界线已经很模糊，也许不久的将来会完全消失。

以 Google Suggest 为例，输入时可注意该提示术语更新的方法，这几乎是即时的。现在查看 Google Maps，放大，用光标抓住地图，稍微滚动一下。不等重载页面，所有事情几乎立即发生。Google Suggest 和 Google Maps 是有效使用 AJAX 的 Web 应用程序新方法的两个示例，分别表示 Web 上可能的基本转换。

考察下列模型，它们用来创建新的应用程序或重新设计现有应用程序。

- **桌面模型**——在这种模型中，富客户平台(RCP)工具仍然“拥有”操作数据库模型(例如，Word 文档、Excel 表格等)的能力。桌面范例通过结合操作系统、数据格式、文件系统组织方法来创建“有墙壁的花园”，它会约束用户。
- **Web 1.0 或“守旧派”Web 应用程序**——在这种模型中，用户填写表格，单击“提交”按钮，然后紧张地等待响应，希望没有破坏应用程序的完整性。多年以来开发人员使用各种即兴方法来处理这种层次的 Internet 应用程序中断，包括会话健壮和隐藏字段的 cookies，它们用表格形式的应用程序来协调多步骤面向事务的应用程序。
- **RIA 模型**——在这种新模型中，用户拿起手头上的设备，启动浏览器，访问其资源和应用程序。使用哪种类型的机器以及使用哪种浏览器并不重要——任何现代浏览器都可以。在 RIA 模型中，用户不需要为了尽责或高效而将东西从家里带到办公室再带回家里。通常以产业标准格式(HTML、XML)存储用户创建的数据资源。用户可以将它们保留在网络上的服务器中，或进行本地复制(例如，复制到主机硬盘或闪存)。

  新一代 RIA 的关键是，在维护应用程序状态和健壮性的过程方面，部分页面替换(partial page replacement)要使用应用程序服务器端的频繁程度。RIA 灵活性的关键是部分页面替换的思想，我们会反复强调这一点。在每个用户交互之后，用户不必等待服务器打开整个页面，基于浏览器的应用程序几乎不可能有助于提供更好的用户体验。开发人员体验变得更好而且包含的特别码更少，这并不是巧合。

因此，为了开发人员本身，为了用户，为了利用改革的潜在可能，要考虑使用 RIA 创建新应用程序的直接结果。RIA 模型直接并且立即威胁到像 Microsoft 这种主流用户的地位。目前，谁拥有 OS 和 RCP 工具，谁就控制着用户。接受指定平台和工具集(例如，MS Windows 和 MS Office)的用户越多，控制用户的层次就越高。有一种所谓的网络效果，它为潜在客户(依赖已经拥有该工具的客户数)创建一个值。RIA 模型也会产生网络效果，因此受到病毒的威胁，同时为开发人员创造了新的机会——为自己或为企业写代码。

0.4　架构和 JavaScript 库

过去几年出现了一些令人难以置信的架构和 JavaScript，这里讨论其中的一些。当前架构的一般观点认为，由服务器给定当前连接速度、页面组成和部分页面替换已经普遍。宽带的大幅度提升(在美国，就有 7000 多万宽带连接的计算机)使服务器交互达到了前所未有的高度。

因此，即使 XmlHttpRequest 和响应(关于这一点还将详细介绍)几年前在现代浏览器中就有了，但直到最近才看到像 Google Map 这样的应用程序，使部分页面替换成为规则而不是异常。后来，在简单的 Java、Python 和 Ruby 服务器中这种应用程序大量出现，在复杂、格式化的多层方案中则逐渐减少。本书讨论了一些服务器架构，包括 Google Widget Toolkit (Java)、TurboGears (Python)和 Ruby on Rails (Ruby)。

在客户方面也发生了极大的变化。JavaScript 对开发人员而言意味着痛苦的经历。怪诞的语言决不是开发人员所喜欢的，它原来缺少强大的统一开源库层。现在已有许多库，包括 MochiKit(其口号是“让 JavaScript 更少”)、Prototype(基于众多 Rails 应用程序选择的库)等。以后会讨论这两个库。

0.5 混合与外来

除了严格基于浏览器的 RIA 之外，我们还介绍了另外两个具体形式的以网络为中心的应用程序，这在 Mac OS/X 上的 DashboardWidgets 和 Windows 上的 Google Desktop Widgets 被首次举例说明，而 Yahoo Widgets 在这两个平台上都进行了举例说明。构件就像应用程序，每个都驻留在它自己的微浏览器环境中。而这样的应用程序通常与本地桌面应用程序有相同的特征，它们使设计、构造和建立 HTML、样式表和 JavaScript(Web 开发人员熟悉的元素)变得更加容易。本书选择讨论 Yahoo Widgets，因为它在多种操作系统上存在。

其次，更具有开创性的以网络为中心的应用程序范例是沉浸式 3D 环境(immersive 3D environment)，像 Second Life 这样的应用程序对此进行了例证。过去为了创建沉浸式环境进行了大量尝试。但这些尝试因为各种原因——网络、带宽问题或糟糕的用户体验——全部以失败告终。但现在，也许是第一次，沉浸式环境不但变得可能，而且还充满活力。Second Life 给开发人员提供了简单而又完整的脚本语言和对象构建环境。我们将讨论这种新兴现象，并提供“实际存在的”和连接到网络的对象的可用代码。

0.6 本书主要内容

本书探讨了许多架构、浏览器内置 API 和服务器代码示例。它也提供了各种实现方式语言，并用代码例证了这些实现方式。为了达到最好效果，必须选择是从不同架构和语言组成它，还是单个实现方式架构和(扩展)单一语言，并设计应用程序。经过多次试验和讨论之后，我们决定在特定架构(Ruby on Rails)中实现 ideaStax，使用 Ruby on Rails(通常简称为 Rails)，主要是因为其代码简洁，其综合性能以及产生有用和易于理解代码的能力强，即使开发者没有 Rails(甚至是 Ruby)经验。

这不是说，本书强调的其他实现方式架构(例如，TurboGears 或 Google Widget Toolkit (GWT))在某种程度上次于 Rails，实际上，如果您喜欢使用另一种语言(TurboGears 的 Python，或 GWT 的 Java)，那么这些都是较好的选项。我们必须做出选择，考虑到 Rails 团队的发展速度，这是合理选择，但决不是唯一选择。

虽然选择用 Rails 运行顶点应用程序，但我们不想偏好某种语言或架构——毕竟，本书讨论的是 RIA，而不是 Rails。我们决定采取折衷的方法，希望您能够喜欢：我们使用其他语言来证明关键概念的代码，例如，聚合(mashup)、客户端验证和自动完成；社会站点概念，例如标记。不管您的语言偏好是什么，都应该保持对它的兴趣。

许多针对开发人员的书籍必须在更大的覆盖宽度和更大的深度之间做出决定。由于工具和库的爆炸性增长，我们选择宽度。因此，我们给出了大量的 RIA 和 Web 2.0 观点，但没有详细讨论任何一种特定架构。然而我们从每种架构中提取了一些重要概念。因为这些架构的细节有联机文档(虽然联机文档的性质有不同变化)，所以本书尽量从开发人员视角讲解每种架构或给出一些“内幕”。

0.6.1　第 I 部分：RIA 简介

这一部分提供了开发 RIA 所需的背景。您会了解到 RIA 的特征及其运行方式，进一步理解 Web 应用程序的概念，并通过实现简单的 Hello World 示例 RIA 来体验它们。最后，介绍了一些通过探讨某些常见 JavaScript 调试方法和工具来提高生产率的技巧。

- 第 1 章探讨了 RIA 的一些特性，阐述了真正的 RIA 不同于传统 Web 应用程序的原因。探讨了在开发传统应用程序与 RIA 时，开发过程、行销和运用的基本思想上存在的不同。
- 第 2 章介绍了 RIA 的基础知识，本书的余下部分需要这些基础知识。进一步讨论了可扩展的标记语言(XML)和文件对象模型(DOM)。接着回顾了可扩展的超文本标记语言(XHTML)、层叠样式表(CSS)和 JavaScript。最后，用一个简单示例把所有复习的内容串连起来。
- 第 3 章详细讨论了 RIA 中常见的聚合现象。这一章介绍了聚合的定义，介绍了编写自己的聚合所需的步骤。还详细介绍了 XMLHttpObject 和 JavaScript Object Notation (JSON)。
- 第 4 章通过 Hello World 示例详细介绍了 RIA 开发方法。本章首次详细讨论了 AJAX 应用程序与传统 Web 应用程序之间的不同。然后简要列出了写简单的 Hello World Web 应用程序所需的步骤。同时还介绍了 Java 和 PHP 中实现和响应异步调用所需的步骤。最后用简单示例说明 Java 和 PHP 中实现和响应异步请求的方法。
- 第 5 章讨论了与调试 Web 应用程序相关的问题。这一章详细介绍了独立于语言的调试工具，以及 RIA 开发人员可用的方法。它将 Firefox 作为主要测试浏览器，描述了该工具可用的一些常见调试插件。还介绍了不同调试工具的用法，包括 Venkman 和 FireBug。最后，讨论了日志实用程序，简要概述了开发简单 JavaScript 日志工具所需的步骤。

0.6.2　第 II 部分：RIA 开发入门

第 II 部分扩充了第 I 部分介绍的方法，阐述了一些更高层次的主题和建立在它们之上的开发程序包。

- 第 6 章讨论了设计的模型-视图-控制器(MVC)范例，对于平台开发人员而言，这可能是需要知道的最重要的概念。MVC 模式表示一种设计和组织代码模块的方法，

以便在域信息和业务逻辑(模型)、用户输出可视化(视图)和用户输入管理(控制器)之间分离关注点。这种类型的体系结构对于 RIA 开发人员而言特别重要，因为 Web 基于的主要客户机/服务器体系结构让 RIA 成为 MVC 设计近乎完美的媒介。第 6 章介绍了本书强调的 3 种服务器端开发程序包：Java、Ruby on Rails 和 TurboGears，并用 MVC 的常见术语解释了这些架构。

- 第 7 章对各种 JavaScript 架构进行了比较，因此您可以大致了解哪种工具适用于哪种任务。在传统 Web 开发中，大多数编程工具在服务器上完成，客户端的任务主要是面向设计的。RIA 编程要求和服务器端一样注重客户端逻辑，因此选择正确的 JavaScript 库是一项重要任务。这种比较有助于理解每种架构的优点及不足。
- 第 8 章进入了“编译到 JavaScript 架构”的新时代，并概述了 Google 和 Ruby on Rails 小组使用的两种不同方法。这些解决方案用更易于编码、维护和调试的更高级的语言取代了 JavaScript 编程。在 RIA 页面被发送给客户之前的某一点，更高级的代码由机器翻译为 JavaScript。
- 第 9 章介绍了本书的顶点应用程序 ideaStax。ideaStax 是简单的 Web 应用程序，它让用户能够创建和维护包含丰富文本和图像的便签的堆栈。由于第 III 部分中各章阐述了各种 RIA 接口技术，因此使用 ideaStax 在简单和一致的上下文中设计示例。第 9 章概述了 ideaStax 概念的目标和性能，其中许多会在第 III 部分实现。

学完这部分之后，应该能够全面理解专业 RIA 开发的基本设计原则，牢固掌握各种工具和架构，以便实现您的设计目标。

0.6.3 第 III 部分：RIA 开发进级

这一部分介绍了现实中开发 RIA 所需的工具。同时介绍了 Web 应用程序中的常见问题，以及在开发 RIA 时如何解决或最小化这些问题的方法。这一部分还介绍了新的性能，例如拖放、用服务器数据(在现实 RIA 中不断使用这些数据)自动完成表单字段。整个部分着重介绍了几种强大的 JavaScript 库，例如 Dojo、Script.aculo.us 和 MochiKit。

这些章节包含各种示例，用来说明使用 3 种不同语言——Java、Ruby 和 Python——开发 RIA 的方法。不管您喜欢哪种语言，通过观察是对象还是函数出现在模型、视图或控制器中，都可以帮助您理解示例的意图。即使非常熟悉示例语言，也可以探讨其他备选语言的性能。如果知道模型、视图或控制器有哪些对象，将示例移植到其他语言就可以完成这项任务。完成之后，可登录本书的 Web 站点 www.wrox.com，里面包含用 3 种参考语言实现的示例。再回头查看第 6 章的 3 种备选实现方式及它们合并 MVC 的方法。

所有章节都定义了合并到顶点 Stax 应用程序的性能。在本书的 Web 站点上可以查看顶点的完整代码和工作版本。同时可参考第 9 章了解应用程序。

- 第 10 章探讨了与 Web 应用程序中表单字段确认相关的问题。探讨了表格输入数据问题以及用来验证数据的常见方法，包括使用 Dojo JavaScript 库的客户端确认，Java 中的服务器端确认和使用 AJAX 对 Java 中单个字段的服务器端确认。
- 第 11 章建立在前一章的基础之上，讨论了表格可用性方面的其他问题。探讨了基于 AJAX 的、用来加强用户表格交互的方法。它建立在第 10 章示例的基础之上，并合并了一些基于 AJAX 的改进。

- 第 12 章介绍了通过添加拖放性能来增强 RIA 互动性的方法。这一章定义了拖放的概念，使用 Script.aculo.us 库简要探讨了其运行原理，并运用 Script.aculo.us 和 Ruby on Rails 来说明如何开发具有拖放性能的 RIA 的方法。
- 第 13 章介绍了一些方法，在这些方法中微妙的 JavaScript 效果就可以提高 RIA 的可用性。介绍了 Script.aculo.us Effects 库，并展示了将这些效果绑定到 RIAPage 事件和动作上的几种方法，同时还提供了几种场景(在这些场景中，特定效果非常有用)中的代码示例，包括动态用户通知和全屏预览。
- 第 14 章详细探讨了标记和分级的概念。介绍了标记和分级的定义，说明了将这些性能合并到新 RIA 和其他 Web 应用程序的理由，以及它们越来越流行的原因。本章还包含用 Python 编写的示例。
- 第 15 章建立在第 14 章介绍的示例的基础上，探讨了作为较大 RIA 一个方面实现的标记和搜索。通过将性能添加到顶点应用程序 Stax，来说明将标记和搜索添加到更复杂的应用程序中的方法。使用 Ruby on Rails 编写的实现方式取代了第 14 章使用 Python 编写的实现方式。

0.6.4　第 IV 部分：高级主题

第 IV 部分介绍了 RIA 超出传统 Web 浏览器的扩展范围。由于 RIA 开发变成了一个越来越专业和需要能力的过程，因此对它们的研究慢慢扩展到计算的所有领域，从操作系统到应用程序再到游戏。通过使用 API，现在许多 RIA 已经配合桌面应用程序，因此用户会两边受益。将联机应用程序——例如 Yahoo 的 Flickr——无缝集成到 Apple 的 iPhoto，让用户能够在本地管理自己的相册，同时又能与全世界共享他们最喜欢的相册。这一部分讨论了这些结合的可能性，同时列举了说明如何使用它们的示例。

- 第 16 章简要介绍了应用程序编程接口(API)的重要性。成功的 RIA 不仅仅为用户提供完美的 Web 接口，还会为开发人员提供高效的编程接口。对 API 的访问是聚合(mashup)和桌面 RIA 得到扩展的可能的原因。这一章介绍了开发、部署和控制第三方访问到 RIA 的方法。
- 第 17 章介绍了桌面构造的概念。构件是小的应用程序，放置在用户的桌面上，但能够访问特定信息或功能。本章说明了如何使用 Yahoo! Widgets 来快速访问通过 API 提供的常用 RIA 功能。这种构件的可用性考虑到了与用户桌面的对应，这样完成简单任务时用户就不需要打开 Web 浏览器并登录。
- 第 18 章探讨了移出浏览器界线、移入沉浸式虚拟世界的早期阶段(在 Neal Stephenson 的小说 *Snow Crash* 的启发下)。沉浸式应用程序是全球终端用户填充和创建的丰富环境。第 18 章将您带入称为 Second Life 的世界，详细介绍如何允许该世界与 Web 交互的技术方法和示例。Second Life 给用户(“居民”)提供了成熟的编程语言来操纵他们周围的世界，它支持访问外部业务和在 3D 沉浸式环境中显示 Web 文档。实体经济在这种虚拟空间发展壮大，传统公司已经开始注意到了这一点。将 RIA 连接到虚拟世界(例如，Second Life)，会使产品处于尖端环境下，而这种环境在某一天会变得非常平常。

0.6.5 第V部分：附录

附录扩充了第 10 章的讨论，继续讨论 Dojo 确认功能和标记。

0.7 运行示例的条件

许多书专注于一种方法或一种语言，因而预先定义所需的运行软件或环境很容易。本书包含广泛的浏览器端 JavaScript 库、多个动态页面生成架构、多种服务器端实现语言。每一章都概述了特定支持和环境要求，可以查找标题“开始”，这样就会枚举特定章节的支持元素。作者的站点 RIABook.ideastax.com 也有到最新版本支持软件的链接，并提供作者和其他读者对支持代码的其他注释。当然 Wrox 站点(www.wrox.com)和 RIABook.ideastax.com 站点也会维护这些代码。

0.8 源代码

在完成本书的示例时，可以选择手动输入代码或者使用本书附带的源代码文件。本书用到的所有源代码可以从 www.tupwk.com.cn/downpage 下载，也可以从 www.wrox.com 下载。进入该站点后，只需找到本书的名称(使用 Search 框或者书名列表)，单击本书的详细页面上的 Download Code 链接，就可以得到本书所有的源代码。

注意：
由于很多书有相似的名称，所以用 ISBN 搜索更为容易。本书的 ISBN 是 978-0-470-08280-5。

下载了代码后，用您喜欢的压缩工具把它解压缩。此外也可以去 Wrox 的主下载页面 www.worx.com/dynamic/books/download.aspx 找到本书或其他 Wrox 出版的书的代码。

0.9 勘误表

尽管我们竭尽所能来确保在正文和代码中没有错误，但人无完人，错误难免会发生。如果您在 Wrox 出版的书中发现了错误(比如拼写错误或者代码错误)，我们将非常感谢您的反馈。发送勘误表将节省其他读者的时间，同时也会帮助我们提供更高质量的信息。

到 www.wrox.com 站点上，用 Search 框或者标题列表找到本书的名称，在详细页面上点击 Book Errata 链接就能找到本书的勘误表。在这个页面中可以看到所有被提交的本书的勘误表，它们是由 Wrox 的编辑发布的。在 www.worx.com/misc-pages/booklist.shtml 中有完整的书的列表，其中包括每本书的勘误表。

如果您在书的勘误表页面上没有看到您发现的错误，可以到 www.wrox.com/contact/techsupport.shtml 上填写勘误表或者将错误发送至 wkservice@vip.163.com。我们会检查这些信息，如果属实就把它添加到本书的勘误表页面上，并在本书随后的版本中更正错误。

0.10　p2p.wrox.com

如果想和作者或者其他人讨论，请加入在 http://p2p.worx.com 的 P2P 论坛。该论坛是基于 Web 的系统，您可以发布关于 Wrox 出版的书和相关技术的消息，与其他读者或技术人员交流。该论坛有预定功能，在您选择的感兴趣的主题有新帖子时，会邮件通知。Wrox 的作者、编辑、其他业界专家和像您一样的读者都会出现在这些论坛中。

在 http://p2p.wrox.com，您会找到很多不同的论坛，它们不但有助于您阅读本书，还有助于您开发自己的应用程序，加入论坛的步骤为：

(1) 到 http://p2p.wrox.com 上单击 Register 链接。

(2) 阅读使用说明，单击 Agree 按钮。

(3) 填写加入必需的信息和其他您愿意提供的信息，单击 Submit 按钮。

(4) 您将收到一封 email，描述如何验证您的账户和完成加入过程。

注意：

不加入 P2P 也可以阅读论坛里的消息。但是如果要发布自己的消息，就必须加入。

加入之后，就可以发布新的消息和回复其他用户发布的消息。可以随时在 Web 上阅读论坛里的消息。如果想让某个论坛的新消息以 E-mail 的方式发给您，可以单击论坛列表里论坛名字旁边的 Subscribe to this Forum 图标。

要了解如何使用 Wrox P2P 的更多信息，请阅读 P2P FAQs，其中回答了论坛软件如何使用的问题，以及许多与 P2P 和 Wrox 出版的书相关的问题。要阅读 FAQs，单击任何 P2P 页面里的 FAQ 链接即可。

目　录

第Ⅰ部分　RIA 简 介