

邓子云 著



SOA 普及网



**電子工業出版社**  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

# SOA 实践者说:

分布式环境下的系统集成

---

邓子云 著

電子工業出版社

Publishing House of Electronics Industry

北京•BEIJING

## 内 容 简 介

本书用朴实的语言和不长的篇幅,讲述了 SOA 的精华内容和实用的知识,带有很强的实战性,并有完整的项目实战。全书内容共分为 8 章,涵盖了 Axis2、Tuscany Java SCA、OSWorkflow、Apache ODE 等开发工具,讲解了 Web Service、RMI、Spring 集成、WS-BPEL、 workflow 技术、基于 SOA 的系统集成平台等技术内容,附书光盘中还有所有章节及项目案例的完整源代码。

本书可作为计算机或相近专业高年级本科生、研究生的 SOA 课程教材或教材参考书,也可作为软件研发人员的开发手册、培训教材,适合有一定 Java 开发基础的读者阅读。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有,侵权必究。

## 图书在版编目(CIP)数据

SOA 实践者说:分布式环境下的系统集成 / 邓子云著. —北京:电子工业出版社, 2010.12  
ISBN 978-7-121-11954-5

I. ①S… II. ①邓… III. ①互联网络—网络服务器 IV. ①TP368.5

中国版本图书馆 CIP 数据核字(2010)第 194222 号

责任编辑:孙学瑛

印 刷:北京中新伟业印刷有限公司

装 订:北京中新伟业印刷有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×980 1/16 印张:21.25 字数:362 千字

印 次:2010 年 12 月第 1 次印刷

印 数:4000 册 定价:49.00 元(含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

# 前言

互联网给我们的 IT 技术和应用带来了太多、太快、太新的知识名词和领域。近年来，SOA 被学术界、工程界、媒体炒得沸沸扬扬。然而似乎浪潮还没完，紧接着云计算、物联网这些新的名词又接踵而至，这些新的名词似乎又都和 SOA 有着千丝万缕的关系。是的，SOA 不是一种纯粹的技术，而是一种面向服务的架构方法，更侧重于思想和方法的运用。

## 本书特点

---

目前，市面上有关 SOA 的图书已有一些，大多让人看了似乎理解了概念却又无从下手，浅显易懂、适合国人口味、又实战性强的可谓凤毛麟角。基于这种情况，我和我所在的团队在从事科研和项目开发的基础上，想开发出一本上手较快、实战性强的有关 SOA 的图书。因此全书：

- 没有复杂的理论、公式和计算，却有不少动手实践的内容；
- 没有一味的学术性解说，却带有完整的项目实战；
- 用朴实的语言和不长的篇幅，讲述了 SOA 精华的内容和实用的知识。

真诚希望本书能够对广大 IT 界的朋友们有所帮助。

我和我所在的团队近年来承担了不少有关 SOA 的项目，如湖南省“十一五”

重点科技计划项目“基于 SOA-BPM 组合架构的智能敏捷的第三方物流管理信息技术研究”，“制造业物流数据交换平台的研究与实现”，“一种物流信息系统集成中间件 LESB 的研究与实现”等纵向课题；也有不少利用了 SOA 架构的横向课题，如湖南省物流公共信息平台有限公司的重大产学研专项湖南省物流公共信息平台（<http://www.hnwlw.net>）。在多年的研究与开发过程中，形成了一支研发团队，也积累了一定的成果。

## 本书内容

---

全书内容共分为 8 章。第 1 章说明了 SOA 的核心概念，并讲解了如何准备好开发工具；第 2 章通过实例体验了 Tuscany 研发的设计思想，并逐步实现这些思想；第 3 章介绍了 Web Service 模型、角色与操作这些基本的概念，以及如何利用 Axis2 和 Tuscany Java SCA 来开发 Web Service。

第 4 章要理解 SCA 节点、SCA 域、资源包等基本概念，并学会开发分布式应用；第 5 章则要领略 Tuscany Java SCA 与 Spring 是如何实现集成的；第 6 章要掌握 WS-BPEL 的基本概念，以及如何来应用 WS-BPEL 定义业务流程，学会运用 Apache ODE 引擎来执行 WS-BPEL 流程。第 7 章学习了工作流技术的基本概念，并详细介绍了一种开源的工作流软件 OSWorkflow。

第 8 章则详细介绍了基于 SOA 的系统集成平台的设计与实现，剖析了平台实现的关键技术、报文数据交换的思想，以及配置管理系统的实现，接收和转发报文的实现，任务队列调度的方法，并在光盘中给出了这个平台的全部源代码。

在本书的编写过程中，我得到了我所在团队成员们的大力帮助。文振华教授、王如龙教授、黄友森教授进行了悉心指导；潘果、杨立雄、杨晓峰、罗铁清、唐爱国、孙美清、西米莎、王莹、伍江杭、杨焱、张益林、黄婧等帮忙整理了部分资料或参与了项目的实战开发；还有电子工业出版社的孙学瑛老师周到的出版服务工作，在此一并表示感谢。

本书可作为计算机或相近专业高年级本科生、研究生的 SOA 课程教材或教材参考书，也可作为软件研发人员的开发手册、培训教材，适合有一定 Java 开发基础的读者阅读。

由于时间仓促，加之作者水平有限，存在错误和不足在所难免，还请广大读者原谅。如有问题和思想需要交流，欢迎您和我们联系，我的联系邮箱是 [dengziyun@126.com](mailto:dengziyun@126.com)。也热忱欢迎有志之士前来加盟我们的研发团队。

邓子云

2010 年 8 月于星城长沙

# 目 录

|                                  |           |
|----------------------------------|-----------|
| <b>第 1 章 SOA 研发前的准备 .....</b>    | <b>1</b>  |
| 1.1 SOA 的核心概念 .....              | 1         |
| 1.1.1 什么是 SOA .....              | 1         |
| 1.1.2 SOA 中间件软件 .....            | 2         |
| 1.1.3 SOA 相关技术标准 .....           | 2         |
| 1.1.4 SCA 与 JBI .....            | 3         |
| 1.2 开发工具 .....                   | 4         |
| 1.2.1 Eclipse 的安装与配置 .....       | 4         |
| 1.2.2 Tuscany 的安装与配置 .....       | 6         |
| 1.2.3 其他工具 .....                 | 8         |
| 1.3 SCA 规范与术语约定 .....            | 9         |
| 1.3.1 规范内容 .....                 | 9         |
| 1.3.2 基本概念 .....                 | 9         |
| 1.3.3 图形描述 .....                 | 10        |
| 1.4 小结 .....                     | 13        |
| <b>第 2 章 Tuscany 研发初体验 .....</b> | <b>14</b> |
| 2.1 Tuscany Java SCA 体系结构 .....  | 14        |

|                                        |           |
|----------------------------------------|-----------|
| 2.2 第一个 SCA 应用：计算器 .....               | 15        |
| 2.2.1 设计思想 .....                       | 16        |
| 2.2.2 编写 POJO 类代码 .....                | 16        |
| 2.2.3 编写配置文件 .....                     | 22        |
| 2.2.4 编写客户端程序 .....                    | 26        |
| 2.3 绑定为 RMI 服务与引用 RMI 服务 .....         | 28        |
| 2.3.1 将计算器服务绑定为 RMI .....              | 28        |
| 2.3.2 通过绑定引用计算器 RMI .....              | 32        |
| 2.4 用脚本语言实现计算器服务 .....                 | 37        |
| 2.4.1 设计思想 .....                       | 37        |
| 2.4.2 编写脚本程序与配置文件 .....                | 39        |
| 2.5 计算器服务与 Web 应用 .....                | 41        |
| 2.5.1 设计上考虑的问题 .....                   | 41        |
| 2.5.2 修改 web.xml 文件 .....              | 43        |
| 2.5.3 编写 sca-contribution.xml 文件 ..... | 44        |
| 2.5.4 编写 JSP 页面 .....                  | 45        |
| 2.6 小结 .....                           | 46        |
| <b>第 3 章 Web Service 开发实战 .....</b>    | <b>48</b> |
| 3.1 Web Service 有关的基本概念 .....          | 48        |
| 3.1.1 Web Service 模型 .....             | 48        |
| 3.1.2 Web Service 模型中的角色 .....         | 49        |
| 3.1.3 Web Service 模型中的操作 .....         | 50        |
| 3.2 基于 Axis2 开发 Web Service .....      | 50        |
| 3.2.1 Web Service 开发方法 .....           | 51        |
| 3.2.2 打包和部署 Web Service .....          | 53        |
| 3.2.3 编写客户端程序 .....                    | 53        |
| 3.2.4 调试 Web Service .....             | 55        |
| 3.3 在 Tuscany 中开发 Web Service .....    | 56        |
| 3.3.1 设计思想 .....                       | 56        |
| 3.3.2 开发 POJO 类和编写配置文件 .....           | 57        |
| 3.3.3 编写启动服务端和客户端程序 .....              | 58        |



|                                        |           |
|----------------------------------------|-----------|
| 3.4 小结.....                            | 59        |
| <b>第4章 分布式应用 .....</b>                 | <b>60</b> |
| 4.1 基本概念.....                          | 60        |
| 4.1.1 SCA 节点.....                      | 60        |
| 4.1.2 SCA 域.....                       | 63        |
| 4.2 域的 Web 方式管理 .....                  | 63        |
| 4.2.1 资源包、组合构件与节点.....                 | 64        |
| 4.2.2 从文件系统中配置域.....                   | 65        |
| 4.2.3 连接非 SCA 服务.....                  | 66        |
| 4.2.4 从分布式域中启动节点.....                  | 66        |
| 4.3 分布式计算器应用.....                      | 67        |
| 4.3.1 设计思想.....                        | 67        |
| 4.3.2 代码编写.....                        | 67        |
| 4.3.3 配置域资源包.....                      | 71        |
| 4.3.4 配置组合构件.....                      | 74        |
| 4.3.5 配置节点.....                        | 75        |
| 4.3.6 调试程序.....                        | 81        |
| 4.4 小结.....                            | 85        |
| <b>第5章 SCA 与 Spring 集成.....</b>        | <b>87</b> |
| 5.1 SCA Spring 构件实现规范.....             | 87        |
| 5.1.1 一个示例情况的分析.....                   | 88        |
| 5.1.2 在 Spring 配置中直接使用 SCA 引用 .....    | 89        |
| 5.1.3 在 Spring 配置中明确声明与 SCA 之间的关联..... | 90        |
| 5.2 用 Spring 实现构件.....                 | 93        |
| 5.2.1 使用多个 Spring 配置文件 .....           | 94        |
| 5.2.2 如何感知 SCA 会用到 Bean 对象.....        | 95        |
| 5.2.3 SCA 构件与 Spring 构件互用示例 .....      | 96        |
| 5.3 运用 AOP 方式装配.....                   | 109       |
| 5.3.1 Spring AOP 基础 .....              | 110       |
| 5.3.2 基于 Schema 配置切面 .....             | 112       |

|                                             |            |
|---------------------------------------------|------------|
| 5.3.3 进一步改进工程 .....                         | 119        |
| 5.4 小结 .....                                | 121        |
| <b>第6章 WS-BPEL2.0 与 Apache ODE 基础 .....</b> | <b>123</b> |
| 6.1 基本概念 .....                              | 123        |
| 6.1.1 WS-BPEL .....                         | 124        |
| 6.1.2 抽象和可执行 .....                          | 125        |
| 6.1.3 描述约定 .....                            | 125        |
| 6.1.4 WS-BPEL 与 WSDL .....                  | 126        |
| 6.2 定义业务流程 .....                            | 126        |
| 6.2.1 <partner>和<partnerLinks> .....        | 127        |
| 6.2.2 <variables> .....                     | 127        |
| 6.2.3 <correlation sets> .....              | 128        |
| 6.2.4 <faultHandlers> .....                 | 129        |
| 6.2.5 <activities> .....                    | 130        |
| 6.2.6 基本活动 .....                            | 130        |
| 6.2.7 结构化活动 .....                           | 133        |
| 6.3 Apache ODE .....                        | 138        |
| 6.3.1 安装 Apache ODE .....                   | 139        |
| 6.3.2 HelloWorld 示例 .....                   | 141        |
| 6.3.3 在 Java 程序中调用 .....                    | 144        |
| 6.3.4 在 ODE 中发布处理 .....                     | 147        |
| 6.4 小结 .....                                | 149        |
| <b>第7章 OSWorkflow 基础与应用 .....</b>           | <b>150</b> |
| 7.1 基本概念 .....                              | 150        |
| 7.1.1 工作流的定义 .....                          | 150        |
| 7.1.2 工作流技术发展趋势 .....                       | 151        |
| 7.1.3 工作流参考模型 .....                         | 152        |
| 7.1.4 工作流管理系统体系结构 .....                     | 154        |
| 7.1.5 工作流定义规范 .....                         | 155        |
| 7.1.6 工作流引擎 .....                           | 157        |

|                                   |            |
|-----------------------------------|------------|
| 7.2 OSWorkflow 基础 .....           | 158        |
| 7.2.1 工作流描述 .....                 | 158        |
| 7.2.2 OSWorkflow 的核心概念 .....      | 160        |
| 7.2.3 下载和安装 OSWorkflow .....      | 167        |
| 7.2.4 GUI 设计器 .....               | 167        |
| 7.2.5 OSWorkflow API .....        | 171        |
| 7.3 使用 OSWorkflow .....           | 172        |
| 7.3.1 创建数据库 .....                 | 172        |
| 7.3.2 部署 Web 应用 .....             | 175        |
| 7.3.3 修改配置文件 .....                | 176        |
| 7.3.4 分析 Web 应用源代码 .....          | 180        |
| 7.4 订单处理工作流实例 .....               | 196        |
| 7.4.1 制作工作流图形 .....               | 197        |
| 7.4.2 部署工作流 .....                 | 202        |
| 7.4.3 使用工作流 .....                 | 202        |
| 7.5 小结 .....                      | 208        |
| <b>第 8 章 基于 SOA 的系统集成平台 .....</b> | <b>210</b> |
| 8.1 平台设计思想 .....                  | 210        |
| 8.1.1 需要解决的主要问题 .....             | 210        |
| 8.1.2 解决问题的关键技术 .....             | 211        |
| 8.1.3 接入情形的处理 .....               | 213        |
| 8.1.4 数据交换 XML 报文处理 .....         | 215        |
| 8.2 平台的配置管理实现 .....               | 220        |
| 8.2.1 平台首页设计 .....                | 221        |
| 8.2.2 接入配置管理 .....                | 222        |
| 8.2.3 生成描述文件 .....                | 234        |
| 8.2.4 接入情况监控 .....                | 266        |
| 8.2.5 转发配置管理 .....                | 268        |
| 8.2.6 转发目标系统配置管理 .....            | 268        |
| 8.2.7 报文与数据库的对应关系配置 .....         | 274        |
| 8.2.8 转发报文监控 .....                | 287        |

|                       |            |
|-----------------------|------------|
| 8.3 平台的报文数据交换实现 ..... | 292        |
| 8.3.1 接收报文处理 .....    | 293        |
| 8.3.2 任务队列的调度 .....   | 297        |
| 8.3.3 接收数据队列 .....    | 298        |
| 8.3.4 转发数据队列 .....    | 307        |
| 8.4 小结 .....          | 320        |
| <b>参考文献 .....</b>     | <b>322</b> |

# 第1章 SOA研发前的准备

在进入 SOA 研发前要准备什么呢？重点准备两个方面：一是掌握 SOA 的核心概念，二是选择和准备好开发工具。

本章就先一起来着力解决这两个方面的问题，再在后续章节中步入实战学习和演练。

## 1.1 SOA的核心概念

---

近年来，SOA 炒得很火，不光是杂志、网络等媒体，软件厂商、高校科研都在致力这方面的研究并做了大量的投入，一些软件厂商纷纷声称已经生产了支持 SOA 或以 SOA 为基础平台的软件，有的高校还将 SOA 作为研究生的课程来开设。那么 SOA 到底是什么？

### 1.1.1 什么是SOA

SOA (Service Oriented Architecture, 面向服务的架构) 是一种面向服务的架构方法。SOA 的核心概念是重用和互操作，它将企业的 IT 资源整合成可操作的、基于标准的服务，使其能被重新组合和应用。与传统的紧耦合 IT 架构相比，SOA 的松耦合架构更能适应业务的变化。在 SOA 中，可以用一个服务替换另一个服务而无须关心其底层的实现技术，唯一要考虑的就是服务接口，而它采用了通用的服务和 XML 标准，SOA 带来的另一个好处是可以充分利用现有的 IT 资源，包括遗留应用和数据库，新系统可以通过将遗留应用和数据纳入 SOA，而不是替换它们，这种架构最终将使企业的 IT 架构能够更快速、更有效地适应

业务需求的变化。

服务并非仅指 Web 服务 (Web Service)，应该这么理解，Web 服务只是服务中的最为典型和常用的一种，其他很多的构件封装形式也可以称为服务。一个构件向外界暴露接口以供访问，这个构件就称为一个服务。构件的种类就有很多了，比如 EJB、JMS、JavaBean、COM/DCOM、CORBA 等。

这么说来，SOA 并不是具体的某一种技术，而是通过服务来综合各种各样的技术，在网络中实现系统的松散耦合。可见，SOA 重点要解决的问题就是实现软件系统之间的集成。如果要更具体和形象一些的话，SOA 这个概念的载体就是基于服务实现系统集成的系统集成中间件软件了，因此 SOA 的研发工作就需要掌握基于这种中间件软件的开发方法与技术。

### 1.1.2 SOA中间件软件

国外著名软件厂商 IBM、BEA、微软、Sun、Oracle 等在成套新产品研发上投入了巨资，已研发出一些 SOA 中间件软件，如 Microsoft Biztalk Server、WebMethods Business Integrator、IBM SeeBeyond、TIBCO 和 Vignette 等。在我国，自 2004 年起，就有研究者和企业关注其技术及发展趋势。近年来，国内软件厂商已经从概念到实现，开发出 SOA 平台产品，并开始推广应用。如我国普元软件是重要的 SOA 厂商之一，并是 SOA 国际构件标准组织和电子商务标准主要制定者 OASIS 的核心成员。

不过商业软件总是比较封闭，源代码并不公开。开源软件则是开放式且免费的，有一些也在逐步走向成熟和应用，比如 Apache 基金会支持的 Tuscany、ServiceMix 等。本书将重点介绍基于体现 SOA 架构方法的 Tuscany 的研发方法。

### 1.1.3 SOA相关技术标准

SOA 相关的技术标准实在太多了，让人目不暇接。如：XML Schema、XML Query 和 XSLT 2.0，XOP (XML-binary Optimized Packaging，XML 二进制优化封装协议)、SOAP (Simple Object Access Protocol，简单对象访问协议)。

Web 服务标准集已经初具规模,内容涵盖传输层、消息机制、编程模型、服务发现和描述、可靠性、事务处理、安全和管理等方面。如 WSDL 用于 Web 服务的语义描述,WS-Policy 用来描述 Web 服务的能力和策略等;WS-Security、SAML 等用来描述 Web 服务相关安全性要求等。

OASIS 已经制定了 SOA 的参考模型 SOA-RM1.0 规范,它提供了一个整体的抽象框架,用来理解 SOA 先进技术理念的抽象框架,但其中描述的 SOA 参考架构能够为企业的 SOA 整体计划中提供一个具有全局性的整体框架加以指导,不过这与现实工程的应用还有一段距离。

#### 【提示】

标准和规范基本相似,但略有不同,规范是标准的建议文件。标准常由业界公认的标准化组织制定和发布,而规范多为厂商或非标准化组织发布,这里不对它们进行区分,统一称为标准。

### 1.1.4 SCA与JBI

目前 SOA 标准之争中主要就是 SCA 和 JBI 两个体系。SCA(Service Component Architecture, 服务构件架构)提供了一种编程模型,可以支持基于 SOA 的应用程序实现。SCA 支持实现服务构件的各种技术,绑定访问构件的各种协议。构件可以用不同的编程语言来实现。

SCA 体系主要是 SCA 和 SDO 两个规范。SCA 规范包括了装配模型和客户端模型两部分,前者约定了如何将异种构件装配并发布成服务;后者则约定如何在异种语言环境中调用服务,从而解决服务从服务端到客户端的跨语言、跨环境的问题。SDO(Service Data Objects, 服务数据对象)统一和简化了应用程序处理数据的方式,有了 SDO 后,应用编程人员就可以用一致的方法操作异构数据源,包括关系型数据库、XML 数据源、Web Services 和企业信息系统。

JBI 体系主要是 JBI 和 JDO 两个规范。JBI(Java Business Integration, Java 业务集成)定义了一个标准的体系结构允许第三方的构件插入到标准的基础设施上。从高层次上看,JBI 定义了可以从可插入构件构建集成系统的体系结构,这一结构中构件的交互使用一种经过中介的消息交换机制,而这一消息交换模

式是基于 WSDL 2.0 或 WSDL 1.1 的。JDO (Java Database Objects, Java 数据对象) 定义了持久保存类与 JDO 运行时环境之间的关系。

SCA/SDO 定义了与具体技术无关的服务构件组装模型及服务间访问的数据结构表示方式, 由于技术无关性及众多厂商的参与, 他们得到了众多厂商的支持, 也有开源的实现, 如 Tuscany。而 JBI/JDO, 它们都是基于 Java 的技术, JBI 更像服务总线的 Java 标准定义, 偏向于定义服务间的通信和组装模式, 而 JDO 是基于 Java 的数据对象表示, 因此它们使用的范围受到限制, 当前支持的主流厂家也不多, 但是开源的实现相对还是比较多的, 如 ServiceMix。

鉴于此, 本书后续内容的研发重点也就在 SCA 体系了, 不过其内容远不止 SCA 和 SDO 这么简单, 因为 SOA 中间件软件体现的是集成理念、SCA 定义的是与具体技术无关的服务构件, 所以它必然要涉及许多相关的开发语言、协议、构件装配方法方面的内容。

目标确定了, 接下来一起来搭建起开发环境, 熟悉开发工具吧。

## 1.2 开发工具

---

开发工具在选择上可主要考虑两个: 一是 Java 集成开发工具, 首选当然是 Eclipse 了; 另一个是 SOA 中间件软件, 这里选择主流的 Tuscany。

### 1.2.1 Eclipse 的安装与配置

Eclipse 是一款免费、开源的集成开发工具, 特别适用于 Java 程序的开发。本书的所有实例都将基于 Eclipse 来开发、实现。编写 Java 程序, Eclipse 不是必需的, 读者也可以使用任意文本编辑工具, 但使用 Eclipse 这样的集成开发工具将大大提高开发的效率。

通过网址 “<http://www.eclipse.org>” 可进入 Eclipse 的官方网站, 如图 1-1 所示。

单击 “Download Eclipse” 按钮进入下载主页面。其实也可以直接输入



“http://www.eclipse.org/downloads” 进入 Eclipse 的下载主页面。

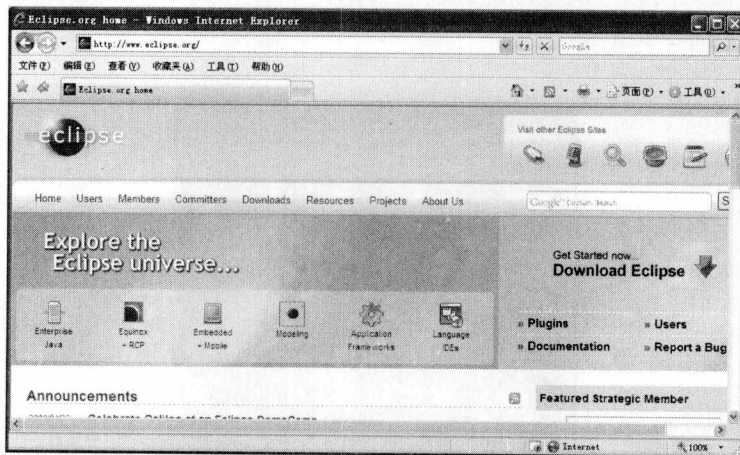


图 1-1 Eclipse 的官方网站

Eclipse 开发工具有很多种。“Eclipse IDE for Java Developers”专用于 Java 应用程序的开发,带有 Java IDE、CVS 客户端、XML 编辑器等工具,但对 Java Web 应用系统的开发支持不够,需要另外加装插件。“Eclipse IDE for Java EE Developers”适合进行 Java 企业级应用系统的开发,特别是 Java Web 应用系统的开发,但此工具软件需要 JDK1.5 或更高版本的 JDK 作为支持。

这里下载“Eclipse IDE for Java EE Developers”。解压缩后,在目录中有一个文件 eclipse.exe,双击它,即会弹出如图 1-2 所示的对话框。

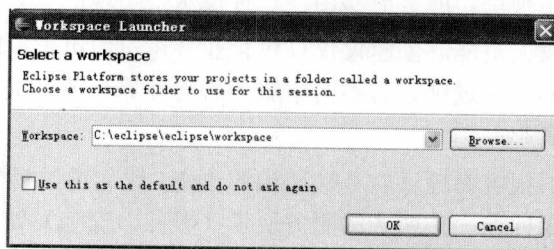


图 1-2 设置 Eclipse 的工作区

这个对话框用于设置 Eclipse 的工作区工作目录,如果选中“Use this as the