

<<WF本质论>>

图书基本信息

书名：<<WF本质论>>

13位ISBN编号：9787111219637

10位ISBN编号：7111219635

出版时间：2007

出版时间：机械工业出版社

作者：Dharma Shukla,Bob Schmidt

页数：322

译者：周健

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<WF本质论>>

内容概要

《WF本质论》详细讲解Windows Workflow Foundation (WF) 的运作原理，并在解析原理的过程中给出WF的设计初衷。

WF使用了一种开创性的方法来编写和执行程序。

《WF本质论》两位作者从WF项目立项开始就参与这个项目，负责规划、设计、开发了其中的大部分技术。

在《WF本质论》中，作者高屋建瓴地探讨了WF中的关键概念和整体架构，不仅涉及如何使用WF，还涉及为什么这样使用WF，揭开了WF技术的神秘面纱。

通过简单而极具说明性的例子，《WF本质论》演示了如何利用WF的可扩展性编程来构造具体领域的程序。

WF编程模型关键部分的核心概念和观点。

活动的执行模型，包括活动自动机的细节、书签机制、调度策略以及WF运行时的线程模型。

高级执行概念，包括活动执行上下文、事务、持久化点、钝化、错误处理、取消操作、补偿和异步。

在应用程序中宿主WF运行时。

活动组件模型，包括验证的细节、WF程序的编译、序列化以及可视化。

数据绑定、XAML、依赖属性以及WF程序元数据。

声明性条件及规则、活动设计器以及如何驻留设计器。

自定义控制流模式，从简单的顺序和迭代控制流到复杂的图和状态机控制流等各种层次。

运行期间对WF程序实例的动态编辑。

<<WF本质论>>

作者简介

作者：(美)舒克拉(Dharma Shukla) 施密德 译者：周健Dharma Shukla 微软架构师，工作目标为下一代编程模型。

他是WF项目组的一名支柱成员，在定义WF体系结构的过程中扮演了重要的角色。

Bob Schmidt 微软高级程序经理，他的工作目标也是下一代编程模型。

自2003年开始，他的主要精力放在了WF的设计上。

他曾获得斯坦福大学计算机科学硕士学位。

<<WF本质论>>

书籍目录

| | | | | | |
|---------------|------------------------|--------------------------------|-----------------------|--------------------|------------------|
| 第1章 剖析WF | 11.1 线程进程灵活性 | 41.1.1 书签 | 61.1.2 可恢复语句组件 | 91.2 复合语句组件 | 111.3 控制流 |
| | 151.3.1 复合语句组件 | 171.3.2 控制流的健壮性 | 191.3.3 现实中的控制流 | 201.4 程序的声明 | 21 |
| 本章小结 | 23 | 第2章 WF程序 | 24 | 2.1 WF编程模型 | 24 |
| | 2.1.1 活动 | 24 | 2.1.2 复合活动 | 28 | 2.1.3 WF程序 |
| WF运行时 | 33 | 2.3 本章小结 | 38 | 第3章 活动的执行 | 40 |
| | 3.1 调度 | 41 | 3.2 活动自动机 | 42 | 3.2.1 活动的执行 |
| 和结果 | 43 | 3.2.2 活动执行上下文 | 46 | 3.2.3 活动的服务 | 47 |
| | 3.3 回顾书签 | 49 | 3.3.1 WF程序的执行 | 51 | 3.3.2 程序队列 |
| | 51 | 3.3.3 计时器 | 57 | 3.4 活动的初始化和终结化 | 60 |
| 3.5 组件活动的执行 | 64 | 3.6 WF线程 | 74 | 3.7 本章小结 | 79 |
| 第4章 活动执行的进阶话题 | 80 | 4.1 活动执行上下文 | 80 | 4.1.1 活动执行上下文管理器 | 82 |
| | 4.1.2 代理控制流 | 84 | 4.1.3 交错式迭代 | 93 | 4.1.4 活动执行上下文的结束 |
| | 96 | 4.1.5 AEC和WF程序的钝化 | 96 | 4.2 取消 | 98 |
| | 4.2.1 取消状态 | 98 | 4.2.2 复合活动的取消 | 104 | 4.2.3 提早完成 |
| | 106 | 4.2.4 取消处理器 | 107 | 4.3 错误处理 | 109 |
| | 4.3.1 异常状态 | 109 | 4.3.2 复合活动的错误处理 | 112 | 4.3.3 错误传播 |
| | 113 | 4.3.4 错误处理器 | 113 | 4.3.5 处理错误 | 114 |
| | 4.3.6 为错误处理建模 | 114 | 4.3.7 ThrowActivity活动 | 119 | 4.4 补偿 |
| | 120 | 4.4.1 补偿状态 | 120 | 4.4.2 补偿处理器 | 122 |
| | 4.4.3 默认补偿 | 123 | 4.4.4 自定义补偿 | 125 | 4.5 本章小结 |
| | 128 | 第5章 宿主应用程序 | 129 | 5.1 WF运行时 | 129 |
| | 5.1.1 服务 | 131 | 5.3 创建WF程序实例 | 134 | 5.4 运行WF程序实例 |
| | 144 | 5.5 WF程序实例的钝化 | 150 | 5.5.1 运行期间活动序列化 | 156 |
| | 5.5.2 基于代理的序列化 | 156 | 5.6 WF程序实例的挂起 | 162 | 5.7 WF程序实例的中止 |
| | 164 | 5.8 放弃WF程序实例 | 165 | 5.9 WF程序实例的完成 | 166 |
| | 5.10 WF程序实例的生命周期 | 166 | 5.11 本章小结 | 175 | 第6章 事务 |
| | 176 | 6.1 TransactionScopeActivity活动 | 177 | 6.2 保存点 | 181 |
| | 6.3 事务 | 183 | 6.4 数据的事务化传输 | 187 | 6.5 本章小结 |
| | 188 | 第7章 进阶话题 | 189 | 7.1 依赖属性 | 189 |
| | 7.1.1 活动数据 | 190 | 7.1.2 活动数据绑定 | 195 | 7.1.3 附着属性 |
| | 198 | 7.2 使用XAML定义活动类型 | 200 | 7.3 活动组件模型 | 200 |
| | 7.4 验证 | 211 | 7.4.1 活动验证器 | 217 | 7.4.2 复合活动的验证 |
| | 217 | 7.4.3 验证选项 | 219 | 7.5 编译 | 220 |
| | 7.5.1 编译器参数 | 221 | 7.5.2 编译器输入 | 222 | 7.5.3 验证和编译 |
| | 223 | 7.5.4 活动代码生成 | 224 | 7.6 设计器序列化 | 229 |
| | 7.6.1 代码序列化 | 231 | 7.6.2 XAML序列化 | 233 | 7.6.3 集合序列化 |
| | 234 | 7.7 本章小结 | 236 | 第8章 杂项 | 237 |
| | 8.1.2 码型条件 | 239 | 8.1.3 声明性条件 | 241 | 8.1.4 规则 |
| | 243 | 8.1.5 规则集的执行 | 247 | 8.2 动态编辑运行中的WF程序实例 | 248 |
| | 8.3 跟踪服务 | 255 | 8.4 设计器 | 261 | 8.4.1 设计器基类 |
| | 265 | 8.4.2 附着属性 | 269 | 8.4.3 设计器动作 | 275 |
| | 8.4.4 设计器图形 | 275 | 8.4.5 设计器布局管理 | 277 | 8.4.6 设计器主题 |
| | 279 | 8.4.7 工具箱条目 | 282 | 8.5 驻留设计器 | 284 |
| | 8.5.1 回顾WorkflowView控件 | 284 | 8.5.2 动态解析活动设计器 | 288 | 8.6 本章小结 |
| | 290 | 附录A 活动自动机 | 291 | 附录B 控制流模式 | 292 |

<<WF本质论>>

章节摘录

前言前 言 Windows Workflow Foundation (WF) 是一个通用的编程框架, 它可用于创建需要对外部实体的信号作出响应的交互式程序。

交互式程序的基本特征是它会在执行期间暂停某一长短未知的时段, 以等待输入。

当然, 交互式程序并不是什么新现象。

自从计算机网络出现以来, 交互式程序就广泛应用于两台计算机之间的通信。

这些技术还适用于运行在单台计算机上的数据交换。

这种技术的变体涵盖了从套接字到Web服务等各种技术, 这些技术在程序通信领域已应用了多年。

尽管在这几年中, 编程模型的互操作性、可伸缩性以及易编程性等方面有了长足的发展, 但开发人员在程序交互方面的投入依然匮乏。

比如, 时下流行的Web编程模型, 这类模型通常为开发人员强加控制流模型, 且非常生硬。

因此, WF希望改变这一切。

WF编程模型的核心概念就是活动—WF程序中的程序语句。

活动的执行具有固有的可恢复性, 同时以片段式的方式执行, 它根据活动与外部实体的交互来暂停和恢复执行。

当你在深夜合卷睡觉前, 一般都会在书本当前阅读的位置放置一个书签, 对WF中的活动来说也是这样, 当活动等待外部输入时, 它会在当前执行的位置做一个书签, 当收到外部输入打算恢复执行时, 则从书签的位置继续执行。

WF是一个框架, 而不是一组由语法定义的编程构造。

WF中活动的概念是可扩展的, 这使得WF程序中的表达式以及控制流相对于C#或Visual Basic这样的语言要丰富得多。

你可以使用领域专有的活动以及特定的控制流结构来表示WF程序, 因此, 它能够通过捕捉复杂的人机交互来高度拟真特定的场景。

WF运行时是一个元运行时, 它建立在通用语言运行时 (CLR) 之上, 并管理着WF程序的执行。

在一个分布式的环境中, WF程序可以在不需要做任何额外工作的前提下, 相当自然地暂停或恢复, 也可以执行任意长一段时间。

在WF程序空闲时, 它既不会执行失败, 也不会过度消耗系统资源。

在WF的上下文中, CLR的工作就是管理那些在内存中代表WF程序的对象。

而WF运行时的工作就是管理WF程序的完整生命周期, WF程序的生命周期是可以跨越CLR线程、应用程序域、操作系统进程甚至机器的。

总之, WF提供了一个编写和执行交互式程序的编程模型。

WF程序是由被称为活动的领域相关程序语句构建的。

这允许领域专家可以使用该领域的概念来描述。

关于本书 我们撰写这本书有一个简单的理由, 那就是我们都相信在WF的核心有一个令人瞩目的概念。

在当今主流平台上, 这些概念还是第一次出现。

WF的面向活动编程方法采用了与时下流行的编程范式完全不同的原则, 因此, 学习WF最好的方式就是首先关注WF编程模型赖以建立的基础概念。

仅仅熟悉System.Workflow命名空间下的一系列类型, 对于WF开发者来说是远远不够的。

总而言之, 这本书不会介绍WF的三个程序集中350个以上的类。

WF有意地回避了WF的外围设施, 而将注意力集中于WF的精髓—核心编程模式和运行时特性集。

大量经验告诉我们, 从基本原理着手来学习一个框架是成为资深WF开发者的最可靠路径。

撰写这本书是对我们信念的考验, 也是对交互程序所做假设以及这类程序开发方式的检验。

如果这本书以及WF中的新概念能帮助读者更好地构建应用, 那么我们就成功了。

本书的组织如下。

第1章介绍支撑WF编程模式的一些关键概念: 书签、延续、线程与进程灵活性、钝化、可恢复程序语

<<WF本质论>>

句以及可恢复程序的运行时。

对这些概念的讨论都游离于WF之外，以最简洁的形式出现，以求使讨论最明确化。

第2章将第1章中介绍的概念映射到了WF编程模型中，因此，第2章可以说是接下来几章的桥段。我们从这里开始开发活动，将这些活动组合为简单程序，从而运行这些WF程序。

第3、4章讨论活动执行的一些细节方面，包括书签机制、错误处理、取消操作以及补偿逻辑。而描述活动生命周期的活动自动机则是这两章一致的主题。

第5章探讨构建WF运行时宿主程序的方法，同时展现WF的可扩展性。

第6章讨论WF程序执行中事务的临界区功能。

第7章讲解许多与活动和WF程序创作相关的进阶话题，其中包括验证和编译。

第8章介绍WF核心概念的若干外围特性。

附录A是活动自动机的一个参考。

附录B为我们展现若干复合活动的代码，这些复合活动比正文中所开发的复合活动复杂得多。

这些例子强调了WF编程模型的那些扩展点，同时演示了复合活动构建复杂控制流模型的能力。

在本书中，我们完全把重点放在Windows Workflow Foundation上。

我们假定读者已经掌握C# 2.0和CLR的基本知识。

如果读者还不了解这些，可以参考《The C# Programming Language, Second Edition》，Anders Hejlsberg等（Addison-Wesley，ISBN：0321334434）和《Essential .NET, Volume I: The Common Language Runtime》Don Box和Chris Sells（Addison-Wesley，ISBN：0201734117），以上列出的书都是这一领域的权威资料。

本书中的示例都十分简单——我们有意识地降低场景对示例的干扰，将注意力集中在概念本身。

在你掌握了窍门之后，我们期望你能尝试将本书中的概念和技术应用到你所从事的行业中。

致谢 WF项目开始于三年前。

当时我们这个小型工程师团队已经强烈地预感到：改变陈旧的程序构造方式，开创一种崭新的程序开发方法已经势在必行。

我们的首次尝试发生在与Microsoft Office团队合作的时候，当时，我们思忖着该项崭新的技术是否能成为他们计划中的工作流特性的基础结构。

我们的工作进展十分顺利，WF成为了Microsoft Office 2007中工作流特性的基础，同时更让我们引以为荣的是，WF的第一个公开版本是作为.NET Framework 3.0的一部分发布的（众所周知，.NET Framework 3.0是和Windows Vista一起发布的）。

许多人为WF的设计出谋划策，没有他们的意见和建议，WF就没有今天的公开发行版。

在这里，我们不可能对每个为WF作出贡献的人们致谢，但是我们还是向他们所做的努力致敬。

我们的审校也做了相当多的工作，他们告诉我们什么该着重强调，哪里不能着墨过多，他们还帮我们指出了各章节的错误和遗漏。

他们是：Angel Azcarraga、Don Box、Krzysztof Cwalina、Joe Duffy、Omri Gazitt、Ian Griffiths、Mark Michaelis、Dennis Pilarinos、Jeffrey Richter、Andrei Romanenko、Akash Sagar、Chris Sells、Clemens Szyperski、Nate Talbert、Eric Zinda，正是有了以上各位的反馈意见，才能使本书变得更好。

Ian提供了极具针对性和洞察力的意见；Aditya Bhandarkar在设计期间一节中提供了额外的帮助；Don不仅提供建设性的意见，还慷慨地捉笔为此书撰写序言。

对我们来说，在宝贵的业余时间撰写这样一本书确实是一个挑战。

我们要感谢Addison-Wesley公司的Karen Gettman和Curt Johnson，所有耐心指导我们的出版界专家们，以及所有将我们的文字最终变为书本并呈现在您面前的人们。

如果本书还有什么纰漏的话，那一定是我们的缘故了。

我是幸运的，因为我从WF诞生那一刻起就参与其中。

WF是我迄今为止在微软做过的最有趣的项目。

对我来说，撰写本书是构建WF技术的一种延续。

现在这本书完成了，我也可以好好休息一下了。

多年来，Don Box一直是我的灵感源泉，他一直鼓励我撰写这本书，对此我十分感激。

<<WF本质论>>

Don帮助我构思本书的内容，他废寝忘食的精神以及和蔼可亲的态度，我将铭记在心。

再次深表感谢，Don！

Bob Schmidt和我共同完成了本书的撰写工作。

和Bob一起工作是一段十分愉快的经历。

我们先是一起构建WF技术，随后共同完成本书。

谢谢你，Bob，祝旅途愉快。

还要感谢那些一线的编程人员，感谢你们将不可能变为可能。

我一直十分感激Abhay Parasnis，他创建了一套独特的环境，不仅为孕育技术革新提供了温床，并且奇迹般地在短时间内让实现新技术成为可能。

良好的开端是成功的一半，在WinOE期间的工作给我留下了美妙的回忆。

感激我的父母为我不计回报地付出，还有Pushpa坚定的支持。

我还要感谢我的妻子Bina，她总与我形影不离，还有我的女儿Anya，使一切都变得这么美好。

Dharma Shukla 2006年7月作为首个研发经理加入WF团队，我确信这将是研究改变程序开发方法的契机。

能够为这种新技术的开发贡献自己的一份力量，我深感荣幸，与此同时，我也十分荣幸地参与了撰写这本关于WF的书。

Dharma Shukla诚邀我与他合著这部书籍，我深表感激，他的洞察力、热忱以及对真理的追求，使得我们彼此的讨论总是热情洋溢，而他的宽阔眼界与丰富的想象力，为这部书梳理出清晰的脉络。

我要对Don Box表示感谢，这期间，我们从他那里得到了很大的支持和指导，Don Box有着敏锐的目光，设计的初始思路中着重于基础的理念正是来源于他，这个理念使得我们能够凝聚力量。

Abhay Parasnis不仅在WF项目中是绝对的主力，同时还起到了模范带头作用，他的领导能力在无形中激励着我们。

Abhay Parasnis理应得到褒奖，这无可厚非。

由衷地感谢我的妻子Elaine，这么多年来她一直为我默默地付出，她是我坚强的后盾。

我的两个儿子，Thomas和Andrew，作为父亲，因为著书而频繁地奔波于各项会议而无暇照顾他们的时候，他们总是在耐心等待而无无理取闹，他们甚至知道什么时候才能进入我的办公室以提供给我些许乐趣与灵感—我向你们保证：这个美好的夏天将充斥着冰淇淋和你们喜爱的壁球运动，一同期待吧。

还有我的父母，你们让我领悟到，全身心投入的团队能创造多么伟大的奇迹，你们对我的帮助是无价的，感谢你们。

译者序 译者序 当看到这本书的英文名称时，你是否会联想到另一本重量级的书《Essential COM》。

诚然，这两本书除了名称类似外，确实还有些渊源，为本书作序的正是《Essential COM》的作者，Don Box。

既然本书能冠以Essential的名号，那必定有过人之处。

首先，据我所知，这是市面上第一本深入剖析WF（Workflow Foundation）的书（本书的中译本更是关于WF的第一本中文资料）。

本书详细解释了WF的底层运作原理，并在解析原理的过程中，交代出了WF的设计初衷。

从这个角度看，可以说这不是一本写给WF初学者的书籍。

它不能帮助读者学习如何使用WF设计器，也没有介绍出现在工具条中的每个活动，事实上，本书通篇没有介绍任何一个WF自带的活动。

但通读本书后，读者就能完全了解WF架构者的设计目的，从而最大限度地发掘WF的潜力。

可以这么说，WF是架构在CLR之上的另一个CLR，它完全改变了传统的程序开发方式。

其次，本书的两位作者见证了孵化WF的全过程。

Dharma Shukla是WF的架构师，而Bob Schmidt则是WF团队的项目经理。

关于WF，此二人应该是最有发言权的。

也正是因为这个原因，才使得本书具备相当的深度。

<<WF本质论>>

另一个不得不提的人就是Don Box，他的深刻洞察力以及把握全局的能力，为本书提供了强大支持。本书以一个最简化的Hello World程序开篇，随后渐进式地改造该程序，最后该程序具有所有WF中的核心特性，可谓是麻雀虽小五脏俱全。

在第1、2章中，作者从无到有地构建了一个最简化的WF程序。

第3、4两章则正式切入WF，介绍了作为WF核心概念的活动的整个生命周期。

随后几章则从WF的其他几个侧面入手，介绍了WF运行时的几个核心特性。

当然，本书不是一本参考手册，所以不能也没有必要包罗万象。

早在大学期间，我就开始关注 workflow 技术了，但那时对业界来说，workflow 技术始终是阳春白雪，直到WF的出现。

WF第一个beta版发布后，我急急地试用了一番，由此我预感到这将是一种使 workflow 走向普及的技术。去年本书的原版面世，我又迫不及待地浏览了一番，这让我对WF有了新的认识：WF不仅仅是一个嵌入式 workflow 引擎，它更是一种新颖的编程方式。

它打破了传统的以编程语句为原子单元的编程方式，取而代之以活动作为编程的基本元素。

这让程序的逻辑控制流等元结构得以无限扩展。

因此，当陈冀康编辑邀我翻译本书的时候，我甚感荣幸。

本书的翻译少不了周围朋友的支持。

感谢陈冀康编辑对我的信任。

与陈编辑的合作十分愉快，这段经历将使我终身难忘。

我还要特别感谢我的挚友沈群超，他的文字功底和语言组织能力使得原本生硬的语句变得生动。

他在新项目启动的强大工作压力下，完成了全书的审阅工作，没有他，这本书就不可能这么快和读者见面。

感谢我的导师兼上司黄纪纲。

在得知我正在进行本书的翻译工作后，他一直在给予我鼓励和支持，协助我校对错误。

甚至于在工作安排上也照顾到我的翻译进度，有这样老板的公司怎么能不成功。

我还得感谢我的女朋友。

她虽然在工作上帮不上什么忙，但每天晚上回家后看到她的笑脸对我来说又何尝不是一种最大的安慰呢。

在这里，对所有给予过我帮助而我又没能提到的朋友们一并致谢。

由于译者水平有限，加之时间仓促，译文中有错误或不当之处，敬请读者不吝指正。

我的邮箱是 zhoujian.ww@gmail.com。

周健 2007年7月 序言序 言 2006年7月，我写下了这篇序，当时我已经预感到这将是一个美好的开始。

起初，程序员们就已经发现，将表达其意图的程序用机器可读的方法表达具有举足轻重的作用。

也正是这个念头，导致了助记码的广泛使用。

更为重要的是，不仅开发者已经逐渐意识到，相关领域的专家也已经发表了专业意见——目前尚未存在任何运行时框架或语言设计器能完全胜任数据模式的定义工作——这种观念促使了行业相关语言的诞生。

Smalltalk、Java虚拟机（JVM）以及通用语言运行时（CLR）等系统已经证实：诸如反射、序列化、产生式编程等类型定义的机器可读标识法是有其实际价值的。

但是，类型的基本表示法（诸如字段、方法、类）很大程度上是一个封闭的世界，它不允许用户定义诸如控制流、并发、逻辑结构或者领域相关的概念（比如贴现政策或者四分音符），除非在将一个操作不透明地贯穿于多个方法体。

现在，人们开始反问自己：“既然类型定义能用于数据，那为什么其他设计期结构不能呢？”

幸运的是，有一个人早在2003年就提出了这个问题，他就是我的好友，曾就职于Biztalk Server团队的Dharma Shukla。

当Biztalk Server团队的成员开始将XLANG orchestration engine封装为一个可嵌入的平台组件时，他们已经可以把语言结构从XLANG移植到另一个XML语法的语言上了（事实上，这项提案确实已经提上了

<<WF本质论>>

日程)。

对于构建Windows Orchestration Engine (WinOE) 来说，这是迄今最为符合设计目的的方法。

幸运的是，Dharma先生早已极具预见性地洞察到：对所有的程序定义“一种实模式”并非可行之计，相反，他认为应该将它“元化”，用户可以为他们所工作的领域定义自己的操作码，于是WF项目朝着构建一个可扩展运行时的方法进行。

这个重要的决策使你能使用XML语法来定义和组合这些操作码，因此你有了一个系统，这个系统可以让你亲自决定用于描述某个领域中的应用程序的词汇表和文句结构。

在本书中，Dharma Shukla和Bob Schmidt向我们展示了Windows Workflow Foundation (WF) 是一个极好的元运行时的例子，它使程序员将程序的编写与执行置于其管理之下。

开发者为程序定义模式，并提供一个该模式的解释器来使其能被部署和运行。

这个简单的概念会有大量衍生体。

是的，我说过，好戏即将上演，我们坚信，并执着期待。

Don Box 2006年7月，华盛顿州 Yarrow Point

<<WF本质论>>

媒体关注与评论

如果我要编写一本工作流的图书，那么，本书就是我想写的。

——Jeffrey Richter (<http://Wintellect.com>) Dharma Shukla和Bob Schmidt用一种易于理解而又权威的方法介绍了 workflow 技术，真的值得一读。

——Clemens Szyperski 微软公司软件架构师 Dharma 和 Bob 使得任何已经精通 C# 和 .NET 框架的开发者以及那些想要登堂入室的开发者都可以学习和掌握这门技术，他们做的非常不错。

可别错过了这本书。

——Joe Duffy 微软公司 CLR 组项目经理

<<WF本质论>>

编辑推荐

第一本WF技术图书，带你了解.NET 3.0的核心技术之一。

[精彩试读一] "好戏即将上演....."-Don Box, 微软公司软件架构师 Windows Workflow Foundation(WF)使用了一种开创性的方法来编写和执行程序。

WF程序是由一些可恢复的语句块（称为活动）组成的，封装了专有领域的逻辑和控制流的模式，这些模式都可以映射到真实世界的处理流程。

《WF本质论》两位作者从WF项目开始立项就参与这个项目，负责规划、设计、开发了其中的大部分技术。

在《WF本质论》中，作者高屋建瓴地探讨了WF中的关键概念和整体架构，不仅涉及如何使用WF，还涉及为什么这样使用WF，揭开了WF技术的神秘面纱。

很多WF开发者都想知道如何通过编写活动来高效使用和扩展框架，《WF本质论》提供权威性的解释和答案。

通过简单而极具说明性的例子，《WF本质论》演示了如何利用WF的可扩展性编程来构造具体领域的程序。

<<WF本质论>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>