Broadview
WWW.BROADVIEW.COM.CN

· 李 · 维 · 作 · 品 ·

国内第一本介绍ECO的图书

ECO是Borland/CodeGear基于模型驱动架构的，第一个在.NET平台上实现出来的、强大的模型驱动架构框架。ECO允许开发人员使用模型驱动开发的软件工程方法，结合BDS的Together进行各种.NET应用系统的开发，让开发人员真正体验到结合ECO和BDS所具有的不可思议的、高效的开发能力。阅读本书，您可以：

- 学习最尖端的模型驱动架构软件工程
- 提高软件开发的生产力和品质
- 了解Delphi在开发模式方面的第二次巨大进步

Delphi MDA/DDA程序设计
—— 使用ECO

李维 著

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

Delphi MDA/DDA 程序设计

——使用 ECO

李 维 著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

ECO 是 Borland/CodeGear 基于模型驱动架构的，第一个在.NET 平台上实现出来的、强大的模型驱动架构框架。ECO 允许开发人员使用模型驱动开发的软件工程方法，结合 BDS 的 Together 进行各种.NET 应用系统的开发，让开发人员真正体验到结合 ECO 和 BDS 所具有的不可思议的、高效的开发能力。阅读本书，您可以学习最尖端的模型驱动架构软件工程，提高软件开发的生产力和品质，了解 Delphi 在开发模式方面的第二次巨大进步。

本书还将带领您全面掌握 MDA/DDA 软件工程及其开发技术，并且深入学习如何用 ECO 框架来开发各种.NET 应用系统。阅读完本书之后，您不仅会了解 MDA/DDA，而且也可以成为 ECO 专家，还能使用 MDA/DDA 和 ECO 框架，以您从未有过的开发方式来开发强大的.NET 应用系统。

图书在版编目（CIP）数据

Delphi MDA/ DDA 程序设计：使用 ECO / 李维著. —北京：电子工业出版社，2007.3
ISBN 978-7-121-03812-9
I. D… II.李… III.软件工具－程序设计 IV.TP311.56

中国版本图书馆 CIP 数据核字（2007）第 010840 号

策划编辑：周 筠
责任编辑：陈元玉
技术编辑：罗小平
印 刷：北京市天竺颖华印刷厂
装 订：三河市金马印装有限公司
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：28 字数：500 千字
印 次：2007 年 3 月第 1 次印刷
印 数：3 000 册 定价：49.80 元（含光盘 1 张）

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077。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。

导读

本书配有随书光盘，光盘中除了包含所有的范例应用程序之外，在 Recording 目录下笔者还使用 Camtasia 制作了数个录像，建议读者在阅读本书的同时观看这些录像，来学习如何使用 ECO 开发应用程序。表 0-1 整理了这些录像的说明。

表 0-1

录像文件	说　明
MakeECOPackage	展示如何使用 BDS 2006 开发 ECO 包
DevelopECOWebApp	展示如何使用 BDS 2006 开发 ECO ASP.NET 应用程序
DevelopECOWinFormApp	展示如何使用 BDS 2006 开发 ECO WinForm 应用程序
LearningOCL	展示如何使用 ECO 调试器学习 OCL

读者可以在阅读本书之后观看这些录像并顺便使用 BDS 2006 来学习开发 ECO 应用程序，或是在阅读本书之前先观看这些录像以便有基本的印象，然后在阅读本书第 1 章之后再回过头来观看一次这些录像，这样一来，相信读者很快便可以熟悉如何使用 ECO 了。

要观看这些录像，读者只须使用相关录像的 HTML 文件即可。

序

对Delphi来说，使用模型驱动开发并不是最近才发展出来的技术，在Delphi 7的Architect版本中，Borland便把Win32下模型驱动架构（Model Driven Architecture，MDA）/设计驱动架构（Design Driven Architecture，DDA）的产品Bold包装在其中。Bold对于许多Delphi的开发人员来说并不陌生，而且被视为非常强大的开发工具。

近年来，随着软件工程的逐渐普及，愈来愈多的开发人员了解到，在现今的开发竞争中，必须使用良好而且合适的软件工程来帮助提高生产力和软件质量。对Delphi开发人员来说，RAD似乎是最熟悉、生产力也高的软件开发方式，但是，随着UML、测试驱动开发、业务逻辑模型等概念、技术和相关工具慢慢进入开发人员日常的工作之中，许多Delphi开发人员也开始试着使用其他的软件工程并且搭配RAD来进行Delphi应用系统的开发工作。

在Delphi进入.NET的开发环境之后，Bold团队也决定开发一个.NET下的模型驱动/设计驱动开发框架，其目的在于帮助Delphi开发人员从更高的抽象层次来开发.NET应用程序，这个在.NET平台实现的MDA/DDA框架就称为ECO（Enterprise Core Objects）。ECO I出现于C# Builder 1.0时代，ECO II于BDS 2005推出，而目前BDS 2006中的ECO则是第3个版本。

笔者非常喜欢ECO，因为它不但功能强大，而且非常符合开发习惯。当使用ECO开发应用程序时，开发人员可以专注于设计要解决的问题模型。一旦这个模型定义、开发完毕之后，整个应用程序就几乎已经完成了大半的工作，因为ECO在应用程序运行时会执行开发人员设计的模型来解决客户的问题。因此，使用ECO就等于使用模型来开发软件，开发人员可以避免花费很多的开发时间于琐碎的细节之中，而能集中精力于设计完善的业务逻辑。ECO提供了服务框架、OR Mapping、可视化模型设计接口、自动程序代码产生器来提高开发人员的生产力。

由于ECO的功能非常丰富，且许多Delphi开发人员并不熟悉MDA/DDA的理念和开发方式，所以，尽管ECO是BDS最强大的竞争产品，却少有BDS的开发人员了解如何

使用 ECO。在国外，随着 ECO 被愈来愈多的 Delphi/C#开发人员用来开发应用系统，因此也有愈来愈多的 ECO 文章、白皮书和书籍出现，然而，这些有关 ECO 的资料不是英文、德文，就是每篇文章都只讨论某一 ECO 的功能，致使国内许多想全面学习 ECO 的开发人员无法一窥 ECO 的全貌而造成了严重的学习障碍。

笔者在学习 ECO 的过程中也是备尝艰辛，除了个人时间不多之外，资料的收集更是重大的挑战。在笔者逐渐掌握 ECO 技术之后本想只在 Blog 中写一些介绍性的文章，但无奈许多网友阅读了这些 ECO 文章之后便强烈建议笔者写一本完整的 ECO 书籍，好让更多的人了解并学习 ECO。在得到了博文视点周筠（Yeka）女士的支持之后，决定尽力完成一本 ECO 的中文技术书籍，但是没有想到这一写就是一年半多的时间，从 ECO II 写到 ECO III，最后终于在咬紧牙关努力冲刺之下于 2006 年底完成了本书，对于许多殷殷期盼本书的读者来说，时间拖了如此之久，笔者真的感到很抱歉。

本书的重点是希望帮助读者了解什么是 MDA/DDA 以及如何使用 ECO 开发应用程序，因此，讨论的内容从介绍如何使用 ECO 开发 ASP.NET/WinForms 应用程序开始，到介绍 MDA/DDA 的规范，ECO 的服务架框，OCL 语言的概念和使用，再到 OR Mapping，以及如何调整 ECO 应用程序的执行效率。希望读者在阅读完本书之后能够顺利地使用 ECO 开发各种应用系统。

虽然笔者已尽力确保本书的质量，但由于所知有限，书中难免出现错误，期望读者不吝赐教。最后谢谢博文视点强大的编辑/技术团队对于本书的支持，以及周筠女士的决心，谢谢他们敢于出版讨论的技术过于先进的书籍，希望在本书出版之后笔者不会被列为拒绝往来的客户才好☺。

李 维

2006 年 12 月于台北，新店

目录

第1章

开发第一个ECO应用程序

如果撰写程序的方法只有一种，那么程序设计就不会再那么有趣了。如果读者和笔者一样是从 DOS 时代便开始撰写程序的，那么就会和笔者一样观察到开发应用程序的方法层出不穷，从所有的程序代码完全由开发人员撰写，而系统只提供基本的中断（Interrupt）服务，到有了 API 可以调用，再到众多框架（Framework）的出现，逐渐改变了开发人员开发软件的方式和习惯，不过这些改变主要都聚焦在程序代码之上。

然而，随着高生产力和高质量软件的要求标准不断地提高，慢慢地，一些提倡软件工程的开发方法也逐渐出现在集成开发环境中，并且被愈来愈多的开发人员所接受而且使用之。在这些提倡的软件工程中，有的着重程序代码和测试的结合，有的着重开发设计阶段的完整性，有的则强调开发流程的重要性。模型驱动架构/设计驱动架构（Model Driven Architecture/Design Driven Architecture，简称为 MDA/DDA）便是这些许许多多的软件工程之一。MDA/DDA 包含了许多重要的理念、标准和技术，它的基本理念是软件开发应该着重解决的问题的架构设计，在软件的架构设计完毕之后，MDA/DDA 的实现方案应该会提供工具把开发人员设计的软件架构转换为特定平台、特定程序语言和特定的技术架构的结果，之后开发人员就可以根据转换之后的结果再进行后续开发。

本书稍后会详细介绍 MDA/DDA 包含的标准和技术，了解了 MDA/DDA 基本的概念之后，将为您介绍什么是 ECO。

1.1 认识 ECO

ECO 是 Delphi 根据 MDA/DDA 发展出来的技术，ECO 的全名是 Enterprise Core

Objects，从 ECO 的名称中就可以推知其目的是用来开发企业级的软件。ECO 是一个完整的框架，它是由许多高端的技术所结合而成的。

- 可视化设计界面，通过 Together 技术让开发人员设计应用程序的类架构（静态模型）以及状态机（动态模型）。
- 运行时框架，这个运行时框架称为 EcoSpace。EcoSpace 有几个重要的功能，首先，EcoSpace 能够在应用程序运行时执行开发人员设计的静态模型和动态模型。其次，EcoSpace 在运行时提供一组服务框架，允许开发人员通过 Delphi/C#程序代码于运行时存取各种运行时服务。最后，ECO 应用程序在运行时根据静态模型和动态模型建立的各种对象也都由 EcoSpace 所管理，EcoSpace 提供了 OR Mapping（Object Relational Mapping）的功能、对象事务管理、对象缓存等高端的功能，在稍后的章节中读者将会逐渐了解到 EcoSpace 的角色和功能。
- OCL。OCL 是 Object Constraint Language 的缩写，它提供了开发人员在静态模型中撰写业务逻辑，或在 ECO 应用程序运行时通过 OCL 来查询、处理对象。在稍后的章节中将会详细说明 OCL。
- Action 语言。Action 语言是从 OCL 扩展出来的语言，为什么有了 OCL 还需要 Action 语言？这是因为 OCL 是一个没有任何副作用（side effect）的形式语言（Formal Language），这个意思是说 OCL 不会改变对象的状态，而是使用在类图中定义对象的限制条件或是在类图中和程序代码中执行查询的工作。但是在 ECO 的状态机中却需要能够改变状态，所以 ECO 扩充了 OCL 成为 Action 语言，以便让 Action 语言，具有修改或改变对象状态的能力。简单地说，Action 语言是 OCL 的超集（Superset）。
- ECO 组件组，ECO 也提供了一组.NET 组件，这些组件可以让开发人员用来运用 ECO 提供的各种功能，例如使用 ECO 组件连接到 EcoSpace 以处理对象模型，使用 ECO 组件执行 OCL 查询，使用 ECO 组件把对象模型和后台的数据库连接在一起以利用 ECO 的 OR Mapping 等功能。而最重要的是这些 ECO 组件也提供和.NET 可视化控件（Controls）连接在一起的能力，以便让 EcoSpace 中的对象能够自动地显示在.NET 的图形用户界面组件中，稍后会介绍如何使用这些 ECO 组件。

其实在 ECO 之前，Borland 便已经在 Delphi 7 的企业架构师版中提供了 ECO 的前身：Bold。Bold 是 Win32 环境下实现 MDA/DDA 架构的产品，Borland 在 Delphi 6 之后并购了 Bold，并且在 Delphi 7 的架构师版中提供了第一个由 Delphi R&D 团队接手开发的 Bold。在 Delphi 决定推出.NET 的版本后，Bold R&D 团队也决定推出一个全新的 MDA/DDA 实

现架构，这就是 ECO。

ECO 是由 Bold R&D 团队使用 C#开发的在.NET 环境中实现 MDA/DDA 的产品，ECO 1.0 版在 Borland 的 C# Builder 1.0 版中推出，接着在 Delphi 2005 中推出了 ECO 2.0 版，并且在 BDS 2006 中又推出了 ECO 3.0 版。

ECO 3.0 提供了从 ECO 1.0 以来重大的进步，无论是在质量上或功能上都足以用来开发大型、复杂的应用程序，而且一旦开发人员掌握了 ECO 的开发概念和技术，也能够大幅地提高软件开发生产力。例如 Borland 就做过调查，结果显示：使用 ECO 开发软件的用户开发生产力的效率提高了 5 到 10 倍。

ECO 3.0 的重要改善包括了不但可以开发 WinForm 的应用程序，也可以开发 ASP.NET 应用程序，还可以通过.NET Remoting 开发.NET 环境下复杂的多层应用程序，让 ECO 能够帮助开发人员开发各种不同的应用架构。

此外，ECO 3.0 也提供了各种对象池（Object Pool）的机制，大幅提高了 ECO 在 Web、多层以及数据库架构方面的执行效率，让开发人员不必担心复杂的多线程和连接池/对象池等技术的问题。

本书内容即在于描述如何使用 ECO 3.0 开发各种应用程序，搭配 BDS 2006 强大的开发环境，读者将能够充分掌握 ECO 的威力并且体验使用 MDA/DDA 软件工程的好处。

1.2 ECO 的开发架构说明

在我们介绍如何使用 ECO 开发应用程序之前，让各位对 ECO 能够开发各种系统架构有初步的了解是非常重要的，因为掌握了这些架构以及基础原理之后，稍后在介绍如何开发 ECO 应用程序时，读者更容易掌握整个的开发流程。

让我们先从最简单的 ECO 架构来说明吧，虽然这种架构最简单，但是却非常适合用来说明 ECO 应用程序架构中最重要的基本概念。

1.2.1 使用 WinForm 之单机/主从架构的 ECO 应用程序

ECO 最基本的架构就是开发单机/主从架构的应用程序，在这种架构中，客户端就是.NET 的 WinForm 应用程序。ECO 的 WinForm 应用程序在运行时会建立一个称为 EcoSpace 的运行框架环境，EcoSpace 在运行时提供了各种 ECO 的服务，更重要的是 EcoSpace 运行框架环境会包含开发人员设计的业务逻辑模型，并且根据这个业务逻辑模型来建立对象实例，

根据业务逻辑模型来执行业务规则。简单地说，EcoSpace 就在“执行”设计人员于设计时期设计的模型架构。因此，请读者观察图 1-1 显示的使用 WinForm 之单机/主从架构的 ECO 应用程序架构，就可以知道在这种架构中，客户端的 WinForm 应用程序通过 EcoSpace 来执行开发人员设计的业务逻辑模型，例如建立模型中的对象，执行模型中定义的业务规则等，并且通过 EcoSpace 来存取 ECO 框架提供的服务，例如查询对象、存取对象、储存修改的对象，最后并通过 ECO 框架把修改过的对象自动借助 ECO 提供的 ORMapping 的机制储存到后台的数据库中。而客户端的 WinForm 应用程序从头到尾只须使用对象来完成工作，无须面对复杂的数据存取以及对象和数据之间转换的工作。

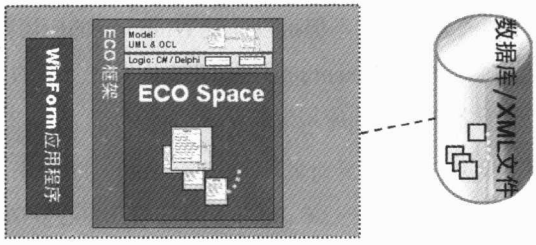

图 1-1 ECO WinForm 应用程序架构

对于单机和主从架构的 ECO 应用程序而言，这个架构的开发是最简单的，也非常适合初学 ECO 的开发人员学习开发架构，但是如果以目前热门的程序来说，开发 ASP.NET 的 Web 应用程序似乎更受欢迎，因此我们马上介绍 ECO 在 Web 中的开发架构。

1.2.2 使用 ASP.NET 之 Web 架构的 ECO 应用程序

ECO 提供了强大的 ASP.NET 开发能力，除了能够使用所有 ASP.NET 原本就提供的各种组件和服务之外，也能够使用所有 ECO 的功能和服务（见图 1-2）。在这种 ECO ASP.NET 架构中，ASP.NET 应用程序同样也可以通过 EcoSpace 执行开发人员在设计时期设计的业务逻辑模型。然而，由于 ASP.NET 应用程序会有许多客户端同时存取使用，因此，ASP.NET 应用程序需要和后台的数据库连接，并需要使用 ECO 持久化映像同步器（PMapper Synchronizer）来同步控制每一个客户端对于对象的修改，以保证后台数据库之中数据的一致性。