

eclipse



EMF: Eclipse Modeling Framework, Second Edition

Eclipse Modeling Framework 2.0

中文版 (第2版)

Dave Steinberg
(美) Frank Budinsky 著
Marcelo Paternostro
Ed Merks
战晓苏 邓中亮 傅效群 译

修订
更新版



清华大学出版社

Eclipse Modeling Framework 2.0 中文版 (第 2 版)

清华大学出版社

北 京

Authorized translation from the English language edition, entitled EMF: Eclipse Modeling Framework, Second Edition, 978-0-321-33188-5 by Dave Steinberg, Frank Budinsky, Marcelo Paternostro and Ed Merks, published by Pearson Education, Inc, publishing as Addison-Wesley, Copyright © 2009. All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and TSINGHUA UNIVERSITY PRESS Copyright © 2010.

北京市版权局著作权合同登记号 图字: 01-2009-2132

本书封面贴有 Pearson Education(培生教育出版集团)防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Eclipse Modeling Framework 2.0 中文版(第2版) / (美) 斯泰因贝格(Steinberg, D.)等著; 战晓苏 等译.

—北京: 清华大学出版社, 2010.6

书名原文: EMF: Eclipse Modeling Framework, Second Edition

ISBN 978-7-302-22632-1

I. E… II. ①斯… ②战… III. 软件工具—程序设计 IV. TP311.56

中国版本图书馆 CIP 数据核字(2010)第 082578 号

责任编辑: 王 军 韩宏志

装帧设计: 孔祥丰

责任校对: 成凤进

责任印制: 杨 艳

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: 北京国马印刷厂

经 销: 全国新华书店

开 本: 185×260 印 张: 33 字 数: 803 千字

版 次: 2010 年 6 月第 1 版 印 次: 2010 年 6 月第 1 次印刷

印 数: 1~3000

定 价: 69.00 元

产品编号: 027837-01

序言一

在不同的人的眼里，建模的含义有很大的不同，甚至在软件工程学科内也是如此。有些人会立即想到统一建模语言(Unified Modeling Language, UML)，有些人会想到模型驱动的体系结构(Model-Driven Architecture, MDA)，还有些人可能会追忆 CASE 工具时代。而随着使用频率的增加，熟悉 Eclipse 社区的人会想到 Eclipse 建模架构(Eclipse Modeling Framework, EMF)，该架构通过使用实用的建模和代码生成工具，为应用程序开发奠定了坚实的基础。

从 Eclipse 的工具项目开始，EMF 就凭借卓越的质量和无与伦比的社区支持赢得美誉，并导致一些在 Eclipse 上形成的互补建模项目快速出现。代码生成器、图形表示架构、模型转换、验证和搜索都在 EMF 的基础上构建，它们现在都包含在 Eclipse 建模项目中，这个顶级项目的成长和成功主要取决于其核心组件——EMF。

在许多方面，EMF 项目都是其他 Eclipse 项目的模型。从其提交人员孜孜不倦地回答项目新闻组中的问题，到 API、特性、性能和文档开发方面的专业性和开放性，EMF 都表现得空前绝后。围绕 EMF 产生的建模社区的多样性，使 EMF 成为了寻求协作的典范，包括个人贡献人员、商业供应商和学术机构。EMF 对于 Eclipse 极具价值的另一个证据是预期在下一个 Eclipse 平台 e4 中使用它。现在，e4 平台的开发人员已经计划利用 EMF 的功能来提供一致并且基于模型的基础和运行库；很明显，这就向模型驱动的软件开发迈进了一步。

由于如此多的技术都建立在 EMF 的基础之上，因此理解 EMF 的体系结构和功能对于成功使用该架构至关重要。多年以来，本书的第 1 版一直是许多开发人员使用 EMF 构建应用程序、扩展 EMF 自身功能的重要资源。随着第 2 版的问世，开发人员将可以了解到 EMF 的许多增强功能，并有效地加以利用。第 1 版中关于 API 的内容被新的章节取代，这些章节将介绍 EMF 持久化、客户端编程、改动记录、验证和富客户端平台(Rich Client Platform, RCP)开发。除更新了原始材料之外，新版本还讨论了 EMF 2.3 版和 2.4 版的最新功能，包括泛型、内容类型和 REST API。这是最受欢迎的第 2 版。

我希望您会像我一样发现本书第 2 版的价值所在。

Richard C. Gronback
Borland Software Corporation 首席科学家
2008 年 11 月

序言二

在许多方面，Eclipse 建模架构都是一种典型的开源项目。Eclipse 基金会(Eclipse Foundation)用许多方法来评估每个项目对社区所做贡献的价值。评估标准包括行业采用情况、健壮技术、作为可扩展平台的稳定性以及开放、透明的项目管理方式。在所有这些方面和其他方面，EMF 多年以来都是最佳的 Eclipse 之一。

在行业采用方面，我经常对 EMF 的极大成功感到非常惊讶。我在 Eclipse 基金会中的主要任务就是游说人们采用 Eclipse 技术。不论是刚刚成立的公司，还是企业或者经过确认的软件供应商，能力 EMF 都是它们使用的主要 Eclipse 技术之一。EMF 简化复杂软件应用程序和产品的开发的能力已经广为人知。很少有一种架构能够在如此众多的领域中极大地提高开发人员的生产率。

这种采用情况主要由一个非常简单的价值定位驱动：EMF 是一种优秀的技术。人们对 EMF 的体系结构、API 的完整性、灵活性和性能已经有了足够的重视。性能是将来在现实应用程序中使用的所有技术的关键。

作为一种平台，EMF 已经改变了建模工具行业。领先的模型工具供应商(例如 Borland 和 IBM)都将自己的产品建立在 EMF 的基础之上，并且依据如下方面制定战略决策：EMF 是它们未来的核心建模架构。在过去 4 年内，我见过的几乎每一种新的建模产品都建立在 EMF 的基础之上。

因此，EMF 在行业内被大量采用，但这只是故事的一半。人们也将 EMF 视为社区成功案例。EMF 团队在与社区交互方面始终做得相当出色。Ed Merks 是 EMF 项目负责人，也是本书的作者之一，他在 Eclipse 社区享有盛誉，因为他始终即时回答所有采用 EMF 的人员对 EMF 的质询。这种以身作则的做法让整个 EMF 团队变成了 Eclipse 中最注重社区活动的团队之一。

EMF 是庞大的 Eclipse 社区成功案例，我相信本书将进一步推动这种成功。

Mike Milinkovich
Eclipse 基金会执行总监
2008 年 11 月

前言

本书全面介绍 Eclipse 建模架构(Eclipse Modeling Framework, EMF), 并指导开发人员学习 EMF 开发人员指南。EMF 是一种强大的架构和代码生成工具, 可用于构建基于简单模型定义的 Java 应用程序。为了使建模对于主流 Java 编程人员实际有用, EMF 统一了 3 种重要技术: Java、XML 和 UML。可以使用 UML 建模工具或者 XML Schema 定义模型, 甚至在 Java 接口上指定简单的批注也能定义模型。在最后一种情况下, 开发人员只编写描述模型的抽象接口的子集, 系统将自动生成并合并剩余代码。

通过将建模概念与这些概念的简单 Java 表示形式关联起来, EMF 成功地在建模人员和 Java 程序员之间架起了一座桥梁。对于 Java 程序员来说, 本书是对建模的系统介绍, 同时又是对建模人员理论的强化, 这种理论就是: 只要有合适的工具, 可以自动完成许多编码工作。本书将说明 EMF 就是这样的工具, 并且说明如何使用 EMF 帮助完成您原先认为使用模型无法完成的许多任务。

EMF 提供一种运行库架构, 从而可以在 UI 中更方便地验证、持久化和编辑任何已建模的数据, 并且自动支持改动通知和记录。元数据可用于启用使用统一的反射 API 的任何数据的通用处理。由于 EMF 具有所有这些功能(甚至更多功能), 因此它在 Eclipse 中是数据共享以及工具 and 应用程序之间细粒度互操作性的基础。同样, Eclipse 本身也是组件和 UI 级别上的集成平台。许多组织目前都使用 Eclipse、EMF 以及 Eclipse 建模项目中不断出现的基于 EMF 的技术作为自己的商业产品和开源产品的基础。

本书假设读者熟悉面向对象编程的概念, 特别是熟悉 Java 编程语言。以前对建模技术(例如 UML)的研究虽然很有帮助, 但是现在不需要了解这方面的知识。第 I 部分(第 1 章~第 4 章)主要概述了 EMF 和建模中一些最重要的概念。这一部分介绍了使用 EMF 建模和构建应用程序所需的所有基本的 Java 编程技能。第 II 部分(第 5 章~第 9 章)全面介绍了 EMF 的元模型 Ecore, 然后详细介绍了 Ecore 和其他支持的模型定义形式(UML、带有批注的 Java 和 XML Schema)之间的映射关系。第 III 部分(第 10 章~第 13 章)详细分析了 EMF 的代码生成器模式和工具, 然后介绍了重要的 EMF 应用程序的端到端示例。第 IV 部分(第 14 章~第 21 章)详细介绍了 EMF 的运行库架构, 并且讨论重要的 EMF 编程技术。

本书的很多内容基于 EMF 2.2, 它是支持传统的 Java 1.4 语言的最新版本。在 2.3 版本中, EMF 采用了 Java 5.0 的主要语言功能, 因而与以前的 Java 运行库不兼容。在编写本书时, EMF 2.2 版本仍然相当普及, 因此它是学习 EMF 的良好基础。第 1 章到第 20 章中的代码都基于该版本, 但是由于 EMF 具有向后兼容性, 因此所有示例都可以在 2.4 版本(本书发行时 EMF 的最新版本)上运行, 而不需要执行任何改动。第 21 章专门讨论了 EMF 2.3 和 2.4 中的改动, 因此使用 EMF 2.4 作为其中示例的基础。

联机示例

本书的 Web 站点是 <http://www.informit.com/title/9780321331885>, 可以从该站点中下载本书中使用的所有示例模型和代码。该站点还提供了勘误表以及与本书相关的其他新闻。

使用这些示例要求具有 Eclipse 和 EMF。可以从 <http://www.eclipse.org/downloads/> 上下载某个 Eclipse 包(推荐 Eclipse Classic), 也可以从 <http://www.eclipse.org/modeling/emf/downloads/> 上下载多合一的 EMF SDK。

目 录

第 I 部分 EMF 概述

第 1 章 Eclipse	3
1.1 项目	3
1.1.1 Eclipse 项目	4
1.1.2 建模项目	4
1.1.3 工具项目	4
1.1.4 技术项目	4
1.1.5 其他项目	5
1.2 Eclipse 平台	5
1.2.1 插件体系结构	5
1.2.2 工作区资源	5
1.2.3 平台用户界面	6
1.2.4 富客户端平台	7
1.3 更多信息	7
第 2 章 EMF 介绍	9
2.1 统一 Java、XML 和 UML	9
2.2 建模与编程	12
2.3 定义模型	13
2.3.1 Ecore 模型(元模型)	13
2.3.2 创建和编辑模型	15
2.3.3 XMI 串行化	15
2.3.4 Java 批注	16
2.3.5 Ecore 概述	17
2.4 生成代码	18
2.4.1 生成的模型类	18
2.4.2 其他生成的内容	20
2.4.3 重新生成和合并	20
2.4.4 生成器模型	21
2.5 运行库架构	22
2.5.1 通知和适配器	22
2.5.2 对象持久化	23
2.5.3 反射 EObject API	26
2.5.4 动态 EMF	27

2.5.5 数据集成基础	28
2.6 EMF 和建模标准	29
2.6.1 统一建模语言(UML)	29
2.6.2 元对象工具	29
2.6.3 XML 元数据互换	29
2.6.4 模型驱动的体系结构	30
第 3 章 使用 EMF.Edit 编辑模型	31
3.1 显示和编辑 EMF 模型	31
3.1.1 Eclipse UI 基础知识	32
3.1.2 EMF.Edit 支持	33
3.2 项提供程序	34
3.2.1 内容和标签项提供程序	35
3.2.2 项属性源	36
3.2.3 命令工厂	37
3.2.4 更改通知	38
3.2.5 项提供程序实现类	39
3.3 命令架构	40
3.3.1 公共命令架构	40
3.3.2 EMF.Edit 命令	43
3.3.3 EditingDomain	45
3.4 生成 EMF.Edit 代码	47
3.4.1 编辑生成	47
3.4.2 编辑器生成	48
3.4.3 重新生成 EMF.Edit 插件	49
第 4 章 使用 EMF——简单概述	51
4.1 示例模型：入门级采购订单 模型	51
4.2 创建 EMF 模型和项目	52
4.2.1 从批注 Java 创建 EMF 模型	53
4.2.2 从 Rational Rose 类模型创建 EMF 项目	58
4.2.3 从 XML Schema 创建 EMF 项目	62

4.2.4 为 Ecore 模型创建生成器模型.....	65	6.7 Rational Rose 中的 Ecore 属性	100
4.2.5 其他格式.....	66	6.7.1 包属性.....	101
4.3 生成代码.....	67	6.7.2 分类器属性.....	101
4.4 运行应用程序.....	69	6.7.3 结构特性属性.....	101
4.5 继续开发.....	71	6.7.4 操作属性.....	102
第 II 部分 定义 EMF 模型		6.7.5 模型元素属性.....	102
第 5 章 Ecore 建模概念	75	第 7 章 Java 源代码	103
5.1 Ecore 模型用法	75	7.1 类的 Java 规范.....	104
5.2 Ecore 内核	76	7.1.1 属性.....	104
5.3 结构特性.....	77	7.1.2 引用.....	106
5.3.1 属性.....	79	7.1.3 属性和引用的简洁符号	108
5.3.2 引用.....	80	7.1.4 操作.....	109
5.4 行为特性.....	81	7.2 枚举类型的 Java 规范.....	111
5.5 分类器.....	81	7.3 用于包的 Java 规范.....	112
5.5.1 类.....	82	7.4 映射的 Java 规范.....	114
5.5.2 数据类型.....	84	7.4.1 包中映射条目类的显式定义	114
5.6 包和工厂.....	85	7.4.2 映射类型引用、操作和参数的定义	115
5.7 批注.....	86	7.5 批注的 Java 规范.....	117
5.8 模型数据类型.....	89	第 8 章 扩展的 Ecore 建模	119
5.9 Ecore 和用户模型	90	8.1 特性映射.....	119
第 6 章 UML	91	8.1.1 多重特性和跨特性订单.....	119
6.1 UML 包.....	91	8.1.2 FeatureMap 接口.....	122
6.2 分类器的 UML 规范.....	92	8.2 使用特性映射建模.....	123
6.2.1 类.....	92	8.2.1 UML.....	123
6.2.2 枚举类型.....	93	8.2.2 批注 Java.....	124
6.2.3 数据类型.....	94	8.2.3 XML Schema	125
6.3 属性的 UML 规范.....	94	第 9 章 XML Schema	127
6.3.1 单值属性.....	95	9.1 模式.....	128
6.3.2 多值属性.....	95	9.1.1 没有目标名称空间的模式	128
6.3.3 包含默认值的属性.....	95	9.1.2 包含目标名称空间的模式	128
6.4 引用的 UML 规范.....	96	9.1.3 全局元素或者属性声明	129
6.4.1 双向非包容引用.....	96	9.1.4 元素或者属性形式默认值	130
6.4.2 包容引用.....	96	9.1.5 EMF 扩展.....	130
6.4.3 映射引用.....	97	9.2 简单类型定义.....	131
6.5 操作的 UML 规范.....	98		
6.6 文档.....	99		

9.2.1 限制	131	9.7.2 属性通配符	167
9.2.2 包含枚举方面的限制	132	9.7.3 EMF 扩展	167
9.2.3 列表类型	134	9.8 批注	168
9.2.4 联合类型	134	9.8.1 文档	168
9.2.5 匿名类型	134	9.8.2 Appinfo	168
9.2.6 EMF 扩展	136	9.8.3 忽略的批注	169
9.3 复杂类型定义	137	9.8.4 非模式属性	169
9.3.1 扩展和限制	137	9.9 预定义的模式简单类型	170
9.3.2 简单内容	138	9.10 EMF 扩展	172
9.3.3 匿名类型	139		
9.3.4 抽象类型	139	第III部分 使用 EMF 生成器	
9.3.5 混合类型	139	第 10 章 EMF 生成器模式	177
9.3.6 EMF 扩展	141	10.1 模型类	177
9.3.7 操作	142	10.1.1 接口和实现类	178
9.4 属性声明	145	10.1.2 存取器方法	178
9.4.1 ID 属性	145	10.1.3 抽象类	180
9.4.2 ID 引用或 URI 属性	146	10.1.4 接口	180
9.4.3 必需的属性	147	10.2 属性	180
9.4.4 默认值	147	10.2.1 简单属性	180
9.4.5 限定属性	148	10.2.2 数据类型属性	181
9.4.6 全局属性	148	10.2.3 枚举类型属性	183
9.4.7 属性引用	149	10.2.4 多值属性	185
9.4.8 EMF 扩展	149	10.2.5 默认值	186
9.5 元素声明	152	10.2.6 Volatile 属性	187
9.5.1 AnyType 元素	153	10.2.7 Non-Changeable 属性	188
9.5.2 ID 元素	154	10.2.8 Unsettable 属性	188
9.5.3 ID 引用或 URI 属性	154	10.3 引用	190
9.5.4 Nillable 元素	156	10.3.1 单向引用	190
9.5.5 默认值	156	10.3.2 双向引用	191
9.5.6 限定元素	157	10.3.3 多重-多值引用	193
9.5.7 全局元素	157	10.3.4 Non-Proxy-Resolving 引用	194
9.5.8 元素引用	158	10.3.5 包容引用	195
9.5.9 替换组	158	10.3.6 Volatile 引用	196
9.5.10 EMF 扩展	160	10.3.7 Non-Changeable 引用	197
9.6 模型组	163	10.3.8 Unsettable 引用	197
9.6.1 重复模型组	163	10.3.9 映射引用	199
9.6.2 重复模型组引用	165	10.4 特性映射	200
9.7 通配符	166	10.5 操作	202
9.7.1 元素通配符	166		

10.6	类继承	203	12.4.3	XSD2GenModel	273
10.6.1	单一继承	203	12.4.4	Ecore2GenModel	273
10.6.2	多重继承	203	12.4.5	生成器	274
10.6.3	接口继承和实现	204	12.5	生成器 Ant 任务	275
10.7	反射方法	205	12.5.1	emf.Rose2Java	276
10.7.1	特性 ID	205	12.5.2	emf.XSD2Java	277
10.7.2	反射存取器	206	12.5.3	emf.Ecore2Java	277
10.7.3	反向握手方法	209	12.6	模板格式	278
10.7.4	特性 ID 转换方法	211	12.6.1	示例模板	278
10.8	工厂和包	212	12.6.2	模板的可扩展性	280
10.9	转换类和适配器工厂	216	第 13 章	示例——实现模型	
10.10	可选的生成器模式	219		和编辑器	283
10.10.1	性能优化	219	13.1	开始入手	283
10.10.2	隐藏 EMFisms	224	13.2	生成模型	285
10.11	定制生成的代码	227	13.3	实现易失特性	285
第 11 章	EMF.Edit 生成器模式	231	13.4	实现数据类型	288
11.1	项提供程序	232	13.5	运行 ExtendedPO2 编辑器	292
11.1.1	内容和标签提供程序	233	13.6	限制引用目标	293
11.1.2	项属性源	235	13.7	将模型划分为多个包	296
11.1.3	命令工厂	237	13.7.1	解析包的依赖关系	297
11.1.4	更改通知	238	13.7.2	回顾限制引用目标	298
11.1.5	对象创建	239	13.8	同时编辑多个资源	301
11.2	项提供程序适配器工厂	243	13.8.1	跨文档、非包容引用	301
11.3	编辑器	246	13.8.2	跨文档包容引用	306
11.4	操作栏参与者	249	第 IV 部分	使用 EMF 编程	
11.5	向导	251	第 14 章	探索元数据	313
11.6	插件	252	14.1	包	313
第 12 章	运行生成器	255	14.1.1	一般性访问包元数据	314
12.1	EMF 代码生成	255	14.1.2	定位包	316
12.2	生成器用户界面	259	14.2	反射	318
12.3	生成器模型属性	261	14.2.1	创建对象	318
12.3.1	模型对象属性	261	14.2.2	询问和修改对象	319
12.3.2	包属性	267	14.3	动态 EMF	324
12.3.3	类属性	269	14.4	扩展的元数据	327
12.3.4	特性属性	269	第 15 章	持久化	331
12.4	命令行生成器工具	270	15.1	持久化架构概述	331
12.4.1	无头调用	270	15.2	EMF 持久化 API	334
12.4.2	Rose2GenModel	271			

15.2.1	URI	334	16.4	复制对象	397
15.2.2	URICConverter	335	16.5	比较对象	401
15.2.3	资源	336	第 17 章	更改模型	403
15.2.4	Resource.Factory 和 Resource.Factory. Registry	340	17.1	描述更改	403
15.2.5	ResourceSet	343	17.1.1	应用更改描述	405
15.3	XML 资源	344	17.1.2	更改多值特性	406
15.3.1	默认的串行化格式	345	17.1.3	更改资源	408
15.3.2	反串行化	349	17.2	更改记录	409
15.3.3	选项	350	第 18 章	验证架构	413
15.3.4	动态 EMF	357	18.1	约束和不变式	413
15.3.5	扩展的元数据	359	18.2	对生成代码的影响	416
15.3.6	其他特性	362	18.3	调用验证	419
15.4	EMF 资源和资源工厂实现	364	18.4	EObject 基本约束	424
15.4.1	基础 XML	365	18.5	XML Schema 约束	424
15.4.2	通用 XML	365	第 19 章	EMF.Edit 编程	427
15.4.3	XMI	365	19.1	重写命令	427
15.4.4	Ecore	367	19.2	定制视图	432
15.4.5	EMOF	367	19.2.1	隐藏模型对象	432
15.4.6	生成的资源	368	19.2.2	使用列表和表查看器	437
15.5	性能考虑事项	368	19.2.3	添加非建模的中间视图 对象	443
15.5.1	推荐的 XML 资源选项	368	第 20 章	Eclipse IDE 之外的应用 领域	453
15.5.2	缓存固有 ID	370	20.1	富客户端平台	453
15.5.3	缓存资源 URI	370	20.1.1	RCP 和 EMF	454
15.6	活动对象的自定义存储	370	20.1.2	启动 RCP 应用程序	454
15.6.1	使用 EStore	372	20.1.3	RCP 应用程序中的生成 代码	458
15.6.2	EStore 和生成类	373	20.1.4	部署 RCP 应用程序	459
第 16 章	客户端编程工具箱	377	20.2	独立应用程序	459
16.1	树迭代器和开关	377	20.2.1	将 EMF 添加到类 路径中	461
16.2	适配器	381	20.2.2	注册资源工厂	463
16.2.1	对象调整	381	20.2.3	注册包	464
16.2.2	行为扩展	387	第 21 章	EMF 2.3 和 2.4	465
16.2.3	内容适配器	390	21.1	Java 5.0 支持	465
16.2.4	观察生成的类	391			
16.3	交叉引用器	393			
16.3.1	基础交叉引用器	393			
16.3.2	交叉引用适配器	395			

21.1.1	枚举	466	21.3.3	批注 Java 模型导入器	486
21.1.2	泛型	469	21.5.1	模型对象属性	489
21.2	EMF 持久化增强	477	21.5.2	包属性	491
21.2.1	资源删除	477	21.5.3	枚举属性	492
21.2.2	内容类型	478	附录 A	UML 符号	493
21.2.3	其他增强	481	附录 B	示例模型概述	497
21.3	其他新特性	484	附录 C	参考文献	511
21.3.1	Ecore 验证	484			
21.3.2	引用键	485			



第 I 部分

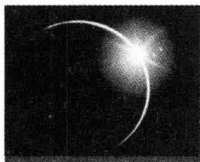
EMF 概述

第 1 章 Eclipse

第 2 章 EMF 介绍

第 3 章 使用 EMF.Edit 编辑模型

第 4 章 使用 EMF——简单概述



第 1 章

Eclipse

在开始介绍 EMF (Eclipse Modeling Framework, Eclipse 建模架构)之前,有必要简单介绍一下 Eclipse。如果您非常熟悉 Eclipse,就可以跳过本章而直接学习第 2 章,从 EMF 的角度来看,第 2 章才是“真正”的第 1 章。

Eclipse 是一个开源软件项目,其目的是提供高度集成的工具平台。Eclipse 的工作包括一个核心项目,该项目包含用于工具集成的通用架构,以及使用这种架构构建的 Java 开发环境。其他项目扩展核心架构,以支持特定类型的工具和开发环境。Eclipse 中的项目以 Java 形式实现,并可以在许多操作系统上运行,包括 Windows、Mac OSX 和 Linux。

Eclipse 将忠诚而又热心的开发人员吸引到一种能够自由交流技术和思想的环境中,希望他们能够创建最优秀的集成平台。Eclipse 生成的软件在 Eclipse Public License (EPL)下可用,EPL 包含许多法律术语,但从广义上讲,EPL 允许免费使用、修改和重新分发软件,并且可以与专利组件一起作为商业产品的一部分分发。作为免费软件许可,EPL 是得到免费软件基金会(Free Software Foundation, FSF)批准和承认的开源促进会(Open Source Initiative, OSI)。对 Eclipse 作出贡献的所有软件都必须得到 EPL 的许可。

Eclipse 的开发由 Eclipse 基金会(Eclipse Foundation)监管,它是一个独立的非盈利组织。该基金会的成员包括 100 多家公司,这些公司支持 Eclipse 并提供基于 Eclipse 的商业产品以及没有法人代表个人代码提交人员。Eclipse 基金会遵守一系列规章制度,其开发过程定义了各个参与人员的角色和责任,包括董事会、Eclipse 管理组织(Eclipse Management Organization)、项目管理委员会(Project Management Committee)、成员、Eclipse 用户和开发人员。

1.1 项目

Eclipse 的开发工作可分为许多顶级项目,包括 Eclipse 项目、建模项目、工具项目和技术项目。Eclipse 项目包括使用 Eclipse 进行开发所需的核心组件。这些组件基本上保持不变,并且可以作为称为 Eclipse 软件开发工具包(Software Development Kit, SDK)的单个

单元下载。其他项目的组件都有特定用途，它们通常是独立的，可以单独下载。随着实践的发展，会不断创建新的项目并给现有项目添加新的组件。

1.1.1 Eclipse 项目

Eclipse 项目支持平台或者架构的开发，以实现集成开发环境(IDE)和其他应用程序。Eclipse 架构使用 Java 实现，并可用作其他语言(例如 C++、XML 等)的开发工具。

Eclipse 项目本身可划分为 4 个主要子项目：Equinox、平台、Java 开发工具(Java Development Tools, JDT)和插件开发环境(Plug-in Development Environment, PDE)。一般来说，这 4 个子项目足以用来扩展架构和开发基于 Eclipse 的工具。

Equinox 和平台是 Eclipse 的核心组件，许多人直接将它们当作 Eclipse。Equinox 是 OSGi R4 核心架构规范的实现¹，这种规范提供作为所有 Eclipse 的基础的组件模型。平台定义支持工具集成所需的其他核心架构和服务，这些服务包括标准工作台用户界面和用于管理项目、文件和文件夹的机制等。第 1.2 节将更详细地介绍平台。

JDT 是使用 Eclipse 构建的功能完备的 Java 开发环境，其工具高度集成，代表 Eclipse 平台的能力所在。JDT 可用来开发 Eclipse 或其他目标平台的 Java 程序，JDT 甚至可用来开发 Eclipse 项目自身。

PDE 提供了视图和编辑器来为 Eclipse 创建插件。PDE 通过支持插件开发活动的非 Java 部分(例如注册插件扩展等)来以 JDT 为基础构建并扩展 JDT。

1.1.2 建模项目

Eclipse 建模项目是 Eclipse 中基于模型开发技术的改进和提升的重点。其核心是 EMF，这也是本书的主题，它为建模提供了基本架构。其他建模子项目建立在 EMF 核心之上，它们能够提供某些功能，例如模型转换、数据库集成和图形编辑器生成。其中还包括许多重要建模标准的实现，例如，UML2 项目提供 UML 2.x 元模型的基于 EMF 的实现。

1.1.3 工具项目

Eclipse 工具项目开发许多典型的、可扩展的基于 Eclipse 平台的开发工具。它包含大量子项目。有些项目为使用其他语言——例如 C/C++、COBOL 和 PHP——提供了一些工具；有些项目，例如图形编辑架构(Graphical Editing Framework, GEF)，能够支持更多种类的 Eclipse 工具。在建模项目形成之前，EMF 刚开始也是工具项目的一部分。

1.1.4 技术项目

Eclipse 技术项目使研究人员、学术人员和教育人员有机会参与 Eclipse 的不断演进的过程。这个项目是新工作或者实验工作的临时大本营，它可能得出自然的结论，也可能启动另一个项目来研究其成熟性。为了实现此目标，其他顶级项目也可能包括所谓的孵化器项目。

¹ 在 <http://www2.osgi.org/Specifications/HomePage> 上可以找到 OSGi 规范。