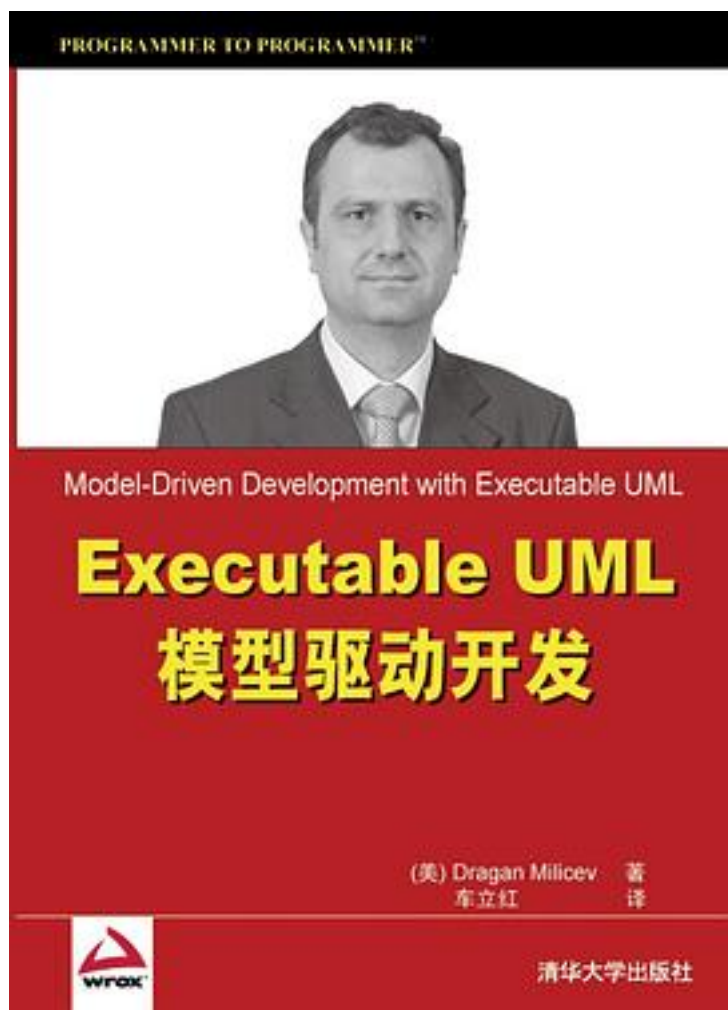


# Executable UML模型驱动开发



[Executable UML模型驱动开发\\_下载链接1](#)

著者:[美]Dragan Milicev

出版者:清华大学出版社

出版时间:2011-10

装帧:平装

isbn:9787302256311

本书讲述了某种软件系统的开发技术，并讲解了一种通过提高抽象层次和降低偶发复杂

性来提高开发效率的方法。

模型驱动开发是一种提高抽象层次的方法，已成功应用了十多年。其基本前提是使用模型而非(只)使用代码来指定软件。模型通常为非线性的形式，这与本质上为线性的代码正好相反。

非线性意味着相比每个元素最多只能有两个相邻元素的简单序列而言，模型的元素可以采用更自由的方式关联。因此，模型通常使用可视化表示法(如图表)而非纯文本来呈现。

作者介绍:

Dragan

Milicev博士是贝尔格莱德大学电子工程学院计算机科学系的副教授。他是Serbian Object Laboratories d.o.o. (SOL, [www.sol.rs](http://www.sol.rs))公司的创始人兼CTO，这是一家软件开发公司，致力于使用模型驱动的技术构建软件开发工具，并构建自定义应用程序和系统。在构建复杂的软件系统方面，Dragan

Milicev拥有25年的丰富经验，曾在20多个学术和国际行业项目中担任首席软件架构师、项目经理或顾问。值得一提的是他曾担任大部分SOL项目及其产品的首席软件架构师和项目经理，这些SOL产品包括：SOloist，一个用于信息系统的快速应用程序模型驱动开发架构；SOL UML Visual Debugger，世界上首批UML可视化调试程序之一，针对UML建模工具Poseidon而设计；SOL Java Visual

Debugger，用于Eclipse的插件，支持使用UML对象图表对测试对象结构建模。他曾在大部分知名的科学和专业的报刊、杂志上发表论文，为模型驱动的开发和UML的理论和实践作出了巨大贡献。Dragan

Milicev以前曾在塞尔维亚出版了3本有关C++、面向对象编程和UML的书籍。您可以通过dmilicev@etf.rs联系他。

目录: 第 I 部分 概述

第1章 信息系统建模 3

1.1 信息系统的定义 3

1.2 模型和建模范式、语言及工具 4

1.2.1 建模 4

1.2.2 建模语言 5

1.2.3 建模工具 8

1.2.4 建模范式 9

1.3 过程和方法 10

第2章 传统的IS开发方法 11

2.1 传统建模范式的特征 11

2.2 可用性方面 12

2.3 开发方面 14

2.3.1 范围中断 15

2.3.2 语义中断 15

2.3.3 开发阶段中断 16

2.3.4 中断的含义 16

2.3.5 用户界面开发问题 17

第3章 对象范式 19

3.1 面向对象建模 19

3.2 统一建模语言 20

3.2.1 UML的特征 21

3.2.2 UML的配置	23
3.3 传统的OO开发方法	23
3.4 所期望的面向对象信息系统的特征	26
3.4.1 可用性方面	26
3.4.2 开发方面	27
3.5 本书其余部分的内容	28
第II部分 OOIS UML概述	
第4章 入门	33
4.1 OOIS UML的主要特性	33
4.2 OOIS UML的组织	35
第5章 基本的语言概念	38
5.1 类和属性	38
5.1.1 需求	38
5.1.2 概念	38
5.1.3 交互表现形式	43
5.1.4 FAQ	44
5.2 关联	48
5.2.1 需求	48
5.2.2 概念	48
5.2.3 交互表现形式	51
5.2.4 FAQ	52
5.3 泛化/特化关系	53
5.3.1 需求	54
5.3.2 概念	54
5.3.3 交互表现形式	57
5.3.4 FAQ	58
5.4 操作	59
5.4.1 需求	59
5.4.2 概念	59
5.4.3 交互表现形式	64
5.4.4 FAQ	64
5.5 多态性	65
5.5.1 需求	65
5.5.2 概念	65
5.5.3 交互表现形式	66
5.5.4 FAQ	67
5.6 一致性规则	67
5.6.1 需求	67
5.6.2 概念	67
5.6.3 交互表现形式	71
5.6.4 FAQ	72
第6章 交互和查询	75
6.1 自定义表现	75
6.1.1 需求	75
6.1.2 概念	76
6.1.3 交互表现形式	80
6.1.4 FAQ	81
6.2 自定义行为	83
6.2.1 需求	83
6.2.2 概念	84
6.2.3 交互表现形式	94
6.2.4 FAQ	95
6.3 查询	97

- 6.3.1 需求 97
- 6.3.2 概念 97
- 6.3.3 交互表现形式 103
- 6.3.4 FAQ 104
- 第III部分 概念
- 第7章 一般概念 109
  - 7.1 OOIS UML的二分法 109
    - 7.1.1 特化/实现和分类器/实例二分法 109
    - 7.1.2 建模和执行 109
    - 7.1.3 编译和解释 110
    - 7.1.4 基本概念和派生概念 111
    - 7.1.5 形式化概念和非形式化概念 112
    - 7.1.6 结构和行为 112
    - 7.1.7 核心部分和扩展部分 112
    - 7.1.8 模型元素和图表 113
  - 7.2 一般的语言概念 116
    - 7.2.1 元素和注释 116
    - 7.2.2 包 117
    - 7.2.3 名称空间和可见性 119
    - 7.2.4 依赖 129
    - 7.2.5 多重性元素 132
- 第8章 类和数据类型 140
  - 8.1 类和数据类型的共有特征 140
    - 8.1.1 类和数据类型的概念 140
    - 8.1.2 作为分类器的类和数据类型 141
  - 8.2 类和数据类型的不同特征 143
    - 8.2.1 标识 143
    - 8.2.2 特性 149
    - 8.2.3 复制语义 152
    - 8.2.4 生存期 153
  - 8.3 实例的创建和销毁 154
    - 8.3.1 动作 154
    - 8.3.2 构造函数 159
    - 8.3.3 创建型对象结构 162
    - 8.3.4 析构函数 182
    - 8.3.5 对象的传播销毁 183
  - 8.4 数据类型 187
    - 8.4.1 原始数据类型 187
    - 8.4.2 枚举 188
    - 8.4.3 内置和用户定义的数据类型 189
- 第9章 属性 192
  - 9.1 作为结构特性的属性 192
    - 9.1.1 作为多重性类型元素的属性 193
    - 9.1.2 静态属性 194
    - 9.1.3 只读属性 196
    - 9.1.4 冻结属性 197
    - 9.1.5 派生属性 198
    - 9.1.6 属性的重新定义 201
  - 9.2 对属性执行的动作 203

- 9.2.1 读取属性动作 204
- 9.2.2 写入属性动作 208
- 9.2.3 符号null 213
- 9.2.4 冻结和解冻属性 215
- 9.2.5 属性迭代 215
- 9.2.6 通过反射访问槽 216
- 9.2.7 用其他详细级语言实现 217
- 第10章 关联 219
  - 10.1 二元关联 219
    - 10.1.1 二元关联和链接 219
    - 10.1.2 关联端和特性 221
    - 10.1.3 二元关联和关联端的语义 224
    - 10.1.4 关联端的特殊特征 230
    - 10.1.5 对二元关联执行的动作 250
  - 10.2 N元关联 259
    - 10.2.1 N元关联的概念 259
    - 10.2.2 N元关联和关联端的语义 260
    - 10.2.3 N元关联端的多重性 262
    - 10.2.4 N元关联端的特定规则 264
    - 10.2.5 对N元关联执行的动作 265
    - 10.2.6 概念建模问题 265
  - 10.3 关联类 268
    - 10.3.1 关联类的概念 268
    - 10.3.2 关联类的唯一性 270
    - 10.3.3 对关联类执行的动作 271
    - 10.3.4 概念建模问题 274
- 第11章 约束 276
  - 11.1 作为模型元素或对象的约束 276
    - 11.1.1 在标准UML中作为模型元素的约束 276
    - 11.1.2 在OOIS UML中作为模型元素的约束 279
    - 11.1.3 在OOIS UML中作为对象的约束 286
  - 11.2 对象约束语言 288
    - 11.2.1 与UML模型的关系 288
    - 11.2.2 操作符和表达式 290
    - 11.2.3 元组 293
    - 11.2.4 集合 293
    - 11.2.5 OCL的OOIS UML非标准语言 302
- 第12章 查询 305
  - 12.1 作为模型元素或对象的查询 305
    - 12.1.1 OOIS UML中查询的语义 305
    - 12.1.2 OOIS UML中作为模型元素的查询 306

- 12.1.3 OOIS UML中作为对象的查询 308
- 12.2 对象查询语言 310
  - 12.2.1 OQL查询的语义 311
  - 12.2.2 FROM子句中的导航 313
  - 12.2.3 WHERE子句中的选择 316
  - 12.2.4 SELECT子句中的投影 317
  - 12.2.5 排序和分组 319
  - 12.2.6 并集 320
  - 12.2.7 参数化和嵌套 320
  - 12.2.8 内联OQL查询 322
- 12.3 模式对象结构 323
  - 12.3.1 模式对象结构规范 323
  - 12.3.2 通过示范创建模式对象结构 328
- 第13章 操作和方法 330
  - 13.1 操作 330
    - 13.1.1 作为行为特性的操作 330
    - 13.1.2 操作的参数 332
    - 13.1.3 操作调用 335
    - 13.1.4 前置条件和后置条件 344
  - 13.2 方法 346
    - 13.2.1 作为操作的实现的方法 346
    - 13.2.2 OOIS UML原始详细级语言 348
  - 13.3 异常和异常处理 365
    - 13.3.1 异常和异常处理的概念 366
    - 13.3.2 异常类型 367
    - 13.3.3 抛出并捕获异常 368
    - 13.3.4 声明由操作抛出的异常 370
  - 13.4 并发和容错 371
    - 13.4.1 OOIS UML中的并发模型 371
    - 13.4.2 并发控制 373
    - 13.4.3 容错和事务 375
- 第14章 状态机 380
  - 14.1 状态机介绍 380
    - 14.1.1 动机 380
    - 14.1.2 状态机、状态和转换 383
    - 14.1.3 监护和效果 385
    - 14.1.4 语义 388
  - 14.2 高级概念 390
    - 14.2.1 复合状态和历史 390
    - 14.2.2 伪状态和最终状态 393
    - 14.2.3 进入和退出行为 396
    - 14.2.4 语义 397
    - 14.2.5 进入和退出点 403

- 14.2.6 子状态机 404
- 14.2.7 设计考虑事项 406
- 第15章 协作和交互 407
  - 15.1 协作和交互 407
    - 15.1.1 动机 407
    - 15.1.2 协作 410
    - 15.1.3 交互 412
    - 15.1.4 交互的语义 414
    - 15.1.5 消息 418
    - 15.1.6 片段 420
    - 15.1.7 交互引用 426
- 第16章 命令、表示和体系结构 429
  - 16.1 命令 429
    - 16.1.1 类命令 429
    - 16.1.2 内置命令 433
  - 16.2 表示 441
    - 16.2.1 表示层体系结构 441
    - 16.2.2 GUI样式配置 446
    - 16.2.3 GUI组件和小部件 449
    - 16.2.4 GUI组件库 451
  - 16.3 应用程序体系结构 458
- 第IV部分 方法
- 第17章 关于方法 465
  - 17.1 活动和制品 465
    - 17.1.1 开发活动和制品 465
    - 17.1.2 UML分析和设计模型 467
  - 17.2 需求工程 469
    - 17.2.1 需求工程的活动和制品 469
    - 17.2.2 需求规范文档 471
- 第18章 概念建模 476
  - 18.1 概念建模过程 476
  - 18.2 标识概念和关系 478
    - 18.2.1 标识并指定类和属性 478
    - 18.2.2 标识泛化/特化关系 482
    - 18.2.3 标识关联 485
    - 18.2.4 对类型-实例关系建模 487
- 第19章 功能需求建模 493
  - 19.1 执行者和用例 493
    - 19.1.1 执行者 493
    - 19.1.2 用例 494
    - 19.1.3 用例间的关系 497
    - 19.1.4 指定用例 503
  - 19.2 管理用例 504
    - 19.2.1 业务流程和用例 505
    - 19.2.2 发现并编程实现用例 508
    - 19.2.3 规划迭代 514
- 第V部分 补充内容
- 第20章 信息系统的特征 519
  - 20.1 特定于域的特征 519
    - 20.1.1 复杂性 519
    - 20.1.2 概念化 519

- 20.1.3 大规模的动态实例化 521
- 20.1.4 功能 521
- 20.1.5 演变 521
- 20.2 与使用性相关的特征 522
  - 20.2.1 交互性 522
  - 20.2.2 适当性 522
  - 20.2.3 时间性 523
  - 20.2.4 可用性和位置的  
独立性 523
  - 20.2.5 安全性 523
  - 20.2.6 操作的方便性 523
  - 20.2.7 源信息与派生信息的  
折中 524
- 20.3 与部署相关的特征 525
  - 20.3.1 数据的多样性和数量 525
  - 20.3.2 可扩展性 525
  - 20.3.3 持久性 526
  - 20.3.4 并发控制 527
  - 20.3.5 分布 530
  - 20.3.6 容错 535
  - 20.3.7 可移植性 536
- 第21章 软件开发的过程和原理 538
  - 21.1 项目管理过程模型 538
  - 21.2 目标和原理 540
    - 21.2.1 目标 540
    - 21.2.2 原理 542
- 第22章 关系范式 545
  - 22.1 介绍 545
  - 22.2 基本概念 546
  - 22.3 数学基础 550
  - 22.4 对结构执行的动作 551
  - 22.5 高级概念 554
    - 22.5.1 视图 554
    - 22.5.2 引用完整性 554
    - 22.5.3 触发器和存储的  
过程 555
    - 22.5.4 索引 556
    - 22.5.5 范式化 556
  - 22.6 SQL 557
    - 22.6.1 SELECT语句 558
    - 22.6.2 数据修改语句 566
  - 22.7 DBMS支持 567
  - 22.8 开发工具支持 568
- 第23章 结构化分析 571
  - 23.1 实体-关系建模 571
    - 23.1.1 基本概念 571
    - 23.1.2 高级概念 572
    - 23.1.3 对结构执行的动作 574
    - 23.1.4 到关系模型的映射 575
  - 23.2 数据流建模 577
- 第24章 对象范式介绍 579
  - 24.1 基本的面向对象概念  
和原理 579
    - 24.1.1 抽象和抽象数据类型 579



24.1.2 封装 581  
24.1.3 继承和子类型化 582  
24.1.4 多态性 583  
24.1.5 面向对象分解 584  
24.2 面向对象编程 585  
24.2.1 抽象数据类型和实例 585  
24.2.2 封装 586  
24.2.3 对象引用 586  
24.2.4 继承 587  
24.2.5 动作语言 587  
参考文献 589  
• • • • • ([收起](#))

[Executable UML模型驱动开发\\_下载链接1](#)

## 标签

uml

计算机

编程

开发

executable

## 评论

-----  
[Executable UML模型驱动开发\\_下载链接1](#)

## 书评

-----

[Executable UML模型驱动开发\\_下载链接1](#)