

文章编号: 1000-5013(2009)04-0478-03

利用 MDA 方法进行应用系统开发

周武斌, 余金山

(华侨大学 计算机科学与技术学院, 福建 泉州 362021)

摘要: 分析遵循模型驱动架构(MDA)规范的 AndroMDA 原理,以及相关应用程序的结构.以一个简化的小区车辆管理系统开发为例,说明利用 MDA 思想进行软件开发的全过程.即首先采用 MagicDraw 建立系统的平台无关模型,然后,对工程的配置进行修改和调整.在模型设计好且工程配置完成后,在 AndroMDA 中执行 maven install 命令,验证系统模型并生成应用程序代码.

关键词: 模型驱动架构;平台无关模型;平台相关模型;统一建模语言;实体

中图分类号: TP 311.11

文献标识码: A

1 AndroMDA 原理与应用程序的结构

模型驱动架构(MDA)是对象管理组织(Object Management Group,OMG)为了实现业务逻辑的分析设计与实现技术之间的低耦合关系,提出的新的软件开发方法^[1-3].它定义了基于模型的开发过程及其将模型自动映射到实现的方法.MDA 的关键之处在于确定了软件开发过程是由对软件系统的建模行为驱动的,其主要思想是将业务规约与实现技术分离,为应用系统建立平台无关模型(PIM),以便更好地适应业务和技术的变化.通过建立针对实现平台的映射,PIM 可以自动生成特定技术的平台相关模型模型(PSM)^[4].通过建立针对实现语言的映射,PSM 可以自动生成代码,从而实现最终的系统.

AndroMDA^[5]开源遵循 MDA 规范的代码生成框架,它可以从 UML 建模工具中读入 XML 格式的 UML 模型,并将其转化为特定的应用架构的生成类和可部署的组件.相对于开发者自己编写的代码来说,AndroMDA 为开发者生成大部分的代码,而自动生成的代码应用了最好的开发实践,代码的结构更为优化,性能更加优越.此外,AndroMDA 能使设计师抽出更多的时间,专注于高质量的模型设计,而且维护起来也更加方便,只需修改模型,然后重新自动生成代码即可.

AndroMDA 生成的应用程序采用了现代企业应用程序流行的 3 层体系结构,包括表现层、业务层和数据层,其中数据层又可以细分为数据访问层和数据存储层.

(1) 表现层.包含了应用程序中需要和用户交互的组件,如 Web 页面、Rich Client 表单\用户交互处理组件等.当前,AndroMDA 提供 Struts 和 JSF 技术来构建 Web 基础的表现层.

(2) 业务层.它封装了应用业务功能的核心.业务组件只提供服务接口,而隐藏了复杂的业务逻辑.业务层主要由 AndroMDA 生成的服务组成,使用 Spring 框架来配置服务.AndroMDA 也能生成 Web Services 服务或 EJB,还能为 JBPM 引擎生成业务过程和工作流.

(3) 数据访问层.提供了一种简单的访问和操作数据的应用程序接口.这一层抽象底层数据访问技术,因而允许业务层重点关注业务逻辑.AndroMDA 以流行的对象关系映射(Object Relational Mapping,ORM)工具 Hibernate 为基础生成数据访问层.

(4) 数据存储层.企业应用将数据保存在一个或多个数据存储结构中.数据库和文件系统是两种常用的数据存储结构.由于 AndroMDA 使用 Hibernate 来生成对数据的访问,用户可以使用 Hibernate 支持的任何数据库,常用的如 Mysql,SQLServer,Oracle,DB2 等.

收稿日期: 2008-05-23

通信作者: 余金山(1952-),男,教授,主要从事软件工程的研究.E-mail: yjs@hqu.edu.cn.

基金项目: 福建省自然科学基金资助项目(A0810013)

2 应用系统开发

2.1 车辆管理系统的描述

以一个简化的小区车辆管理系统开发为例, 来说明此方法的可行性. 该系统主要具备以下 3 点功能. (1) 用户管理. 添加用户、删除单个用户、删除多个用户. (2) 车辆管理. 添加车辆、删除单个或多个车辆. (3) 用户和车辆具有一对多的关系. 用户通过 Web 页面发起业务请求, Web 页面调用 JavaBean 接口, 将相关的参数传递给 JavaBean; JavaBean 再通过 Socket 通讯发送消息包到 CM 平台, 实现对 CM 平台的功能调用; 最后返回处理结果, 记录操作日志.

2.2 系统模型构建

MDA 倡导用模型驱动应用程序的开发, 从系统模型生成应用程序. AndroMDA 把实现 MDA 机制的规范和转换封装了起来, 用户需要在类似 UML 的环境下建立 PIM 模型的用例图、类图、活动图等, 即可按照 AndroMDA 的步骤生成可执行的应用程序. 目前, AndroMDA 主要支持 ArgoUML, MagicDraw 和 RSM 这 3 种建模工具. 这里, 采用 MagicDraw 建立系统的 PIM 模型.

(1) 实体模型描述. 设计实体模型实际上就是设计持久对象, 也称为数据建模, 相当于传统的软件开发过程中的数据库设计. 在“车辆管理系统”的 PIM 中定义了两个实体, Car 和 Person. 每个实体都有 <<Entity>>构造型的标记, 即说明 AndroMDA 是一个 Hibernate 实体.

模型在转换时会使用 Hibernate 转换包, 意味着不用担心应用程序持久性的处理, 因为它是自动生成的. 每个实体只有一个主键, 用 <<Identifier>>构造型来表示, 它将生成对应数据库的主键. 每个实体属性都有相应的类型, 而值得注意的是, 这些类型都是平台无关的. 定义一个具有 <<Criteria>>构造型的 CarSearchCriteria 类, 使用该构造型的类是用来储存复杂查询的参数, 主要用于复杂的多条件查询中.

(2) 业务过程模型描述. 业务过程模型描述了软件系统参与的业务的流程. 这里只给出了 Manage cars 的业务过程模型, 如图 1 所示. 由于篇幅原因, 没有列出 Manage person 的业务过程模型.

图 1 的业务过程描述了管理车辆的实现过程, 用活动图来表示. 车辆的信息显示在管理员 Web 页面上, 管理员可以选择修改车辆信息, 增加新的车辆, 删除旧的车辆. 如果用户选择添加新的车辆, 则会出现新的页面, 要求管理员输入新的车辆信息. 确认后, 通过 Socket 发送到车辆管理系统平台, 存储新的车辆信息, 最后把操作记录在日志文件.

图 1 中的第 1 个状态 getAllCarsInfo() 是对控制器类方法的引用, 而后“过渡”上的参数叫做页变量, 其作用是把参数传递给接下来产生的 JSP 页面. 此处, 把 Collection 形式的车辆传递给下一动作节点, 并将所有车辆显示在应用程序的用户界面上.

第 2 个动作节点中的 <<FrontEndView>>构造型是告诉 AndroMDA: 这个节点与最终用户相关, 应当

添加到用户界面. 在第 2 个动作节点之后, 用户可以选择带参数的不同“过渡”分支, 如表示删除车辆 remove(id: datatype::String) 或者删除多个车辆 removeCarList(id: datatype::String). 这些参数叫做事件参数. 如果使用了事件参数, AndroMDA 就可以知道服务器端需要向客户端请求这些参数. 用户要在 Web 界面上选择这些参数的内容(如动作节点 show listed cars 后面的事件参数), 或者输入相应的参数值. “过渡”上的标记值控制着激发相应事件参数的方式. 如{@ AndroMDA. presentation. Web. action. tablelink= cars}, 表示用户若在删除车辆的表上选择了 cars 这列, 就会激发 remove(id: datatype::String) 事件参数.

必须为每个活动图创建(并分配)一个控制器类. 控制器类是一个普通的类, 它唯一的任务就是把

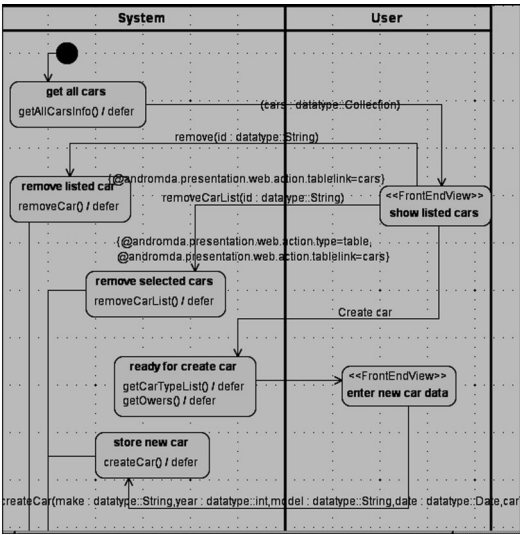


图 1 车辆管理业务模型

Fig. 1 Business model of vehicle management

来自用户界面的调用转发到业务逻辑层. Controller 里面的方法用来在活动图中调用, 换句话说, Controller 是用来在 Struts 的 Action 中调用. 将 Action 和 Controller 分离出来的目的, 主要是让开发者专注于业务逻辑的开发, 从而避开 Action 配置的技术细节. CarService 类加上 «Service» 构造型的标记, 意味着它是业务逻辑层的一部分, 业务逻辑用实体向其他层和同一层提供服务.

2.3 工程配置和运行

设计了应用程序模型之后, 需要对工程的配置进行修改和调整来满足各自的需求. 工程中的 andromda.xml 文件是工程生成工具的核心配置文件, 它由多个 namespace 组成, 每个 namespace 下面包含了每个组件的配置信息. 在模型设计好并且工程配置完成以后, 在 AndroMDA 中执行 maven install 命令, 它将首先验证系统模型. 只有当模型完全通过验证后, 才会生成应用程序代码.

3 结束语

MDA 是一种新的软件开发方法, 通过模型的抽象, 快速地实现业务需求, 可以将应用程序的开发极大的简化. 目前, MDA 还是一种不成熟的方法, 如 100% 代码生成、逆向工程等, 但随着技术的发展和工具的成熟, MDA 方法将会得到进一步的推广和运用.

参考文献:

[1] KLEPPE A, WARMER J, BAST W. MDA explained: The model driven architecture: Practice and promise[M]. New Jersey: Addison Wesley Professional, 2003.
[2] FRANKEL D S. Model driven architecture applying MDA to enterprise computing[M]. New York: Wiley, 2003
[3] STEPHEN J, SCOTT K, UHL A, et al. MDA distilled: Principles of model driven architecture[M]. New Jersey: Addison Wesley Professional, 2004.
[4] 侯金奎, 方建成, 张玉艳. 一种支持 MDA 的 PIM 建模方法[J]. 计算机工程, 2007, 33(8) : 71-73.
[5] 常浩浩, 覃 征. 基于 EMF 和 OCL 的 MDA 软件工程方法研究[J]. 计算机科学, 2007, 34(1) : 268-271.

Application System Development Based on MDA

ZHOU Wu-bin, YU Jin-shan

(College of Computer Science and Technology, Huaqiao University, Quanzhou 362021, China)

Abstract: The principle of AndroMDA, a model driven architecture(MDA) standard, and the structure of applications concerned are analyzed. A simplified community vehicle management system is used to show the whole process of application software development based on it. In the process, a platform independent model is first created using MagicDraw, then the project configuration file is modified an adjusted, after that, the model is verified and the program generated by execution command “ maven install” in AndroMDA.

Keywords: model driven architecture; platform independent models; platform specific model; unified modeling language; entity

(责任编辑: 鲁 斌 英文审校: 吴逢铁)