基于AOP的MDA模型转换

陈 成,李 行
(广东工业大学 计算机学院,广东 广州 510006)

摘 要:模型驱动架构的软件开发方法能够直接生成可用的软件产品,但在开发过程中的建模与模型转换难以实现核心关注点与横切关注点的分离,从而导致模型的纠结和重复。基于模型层的面向方面思想(AOP)可以先将核心功能与横切功能分开独立建模,然后再通过编织技术将它们集成起来,从而有效地解决了这个问题。探讨了通过扩展UML语言使之能表现方面的特性的方法,以及如何针对MDA的不同层次来使用相对应的方面来表达。为模型转换的具体实现提供了可行的方法。

关键词:MDA; AOP;面向方面的UML
中图分类号:TP311 **文献标识码**:A **文章编号**:1673-629X(2008)07-0087-03

AOP-Based MDA Model Transformation

CHEN Cheng, LI Xing
(Faculty of Computer Science, Guangdong University of Technology, Guangzhou 510006, China)

Abstract: Model-driven architecture development methodology can be used to directly generate software products. But in the development process modeling and model transformation is difficult to separate the core concern and cross-cutting concerns. It will cause model entangled and duplication. Based on the aspect-oriented thinking (AOP) in model layer provide an effective approach to solve this problem. It can separately model the core functions and cross-cutting functions, then weave them together. Discusses the method of extended UML language to perform the concept of aspect, and how to address the different levels of MDA correspond to the use of the aspect to express. It provides a workable solution to realize the transformation of MDA model.

Key words: MDA; AOP; aspect-orientation UML

0 引 言

使用传统的软件开发方式,一般在设计之前就要确定好实现系统的编程语言和支持平台,这会导致程序开发人员不仅要考虑用户的业务需求,还要考虑用实现它的特定技术以及相关平台的限制。在实际的项目中,花在具体实现上的时间可能要远远超过理解需求和设计的时间,从而降低了开发效率;其次面对特定平台开发的程序很难移植到其他平台,而代码层面上是很难实现操作系统、数据库系统、相关配置等方面的平台无关性。

模型驱动架构(Model Driven Architecture)方式的出现使解决这些问题成为可能。使用MDA只需要对业务逻辑进行高层次的抽象建模,不需要考虑应用平台和实现细节,具体的代码将由这些模型自动生成。MDA中的模型将不再是设计的辅助工具,而是系统开发过程中的产品。

介绍了MDA的层次结构、模型转换以及面向方面的建模,然后将AOP的思想[1,2]运用于MDA的开发过程中(PIM到PIM和PSM到PSM的转换),提出了在模型层面上实现分离横切关注点的开发方法。

1 MDA概述

1.1 MDA简介

MDA是由OMG发布的一种基于UML,MOF,XMI和CWM等一系列标准的框架,这些标准分别解决了MDA的模型建立、模型扩展、模型交换、模型转换这几个方面的问题。MDA将软件系统的模型分为平台无关模型(PIM, Platform Independent Platform)和平台相关模型(PSM, Platform Specific Model)。PIM注重的是应用系统的业务逻辑,它是抽象的并且与具体实现无关的。标准的模型转换技术将PIM针对特定的平台生成与之对应的PSM,最后基于PSM转换生

收稿日期:2007-10-15
基金项目:国家自然科学基金(60474072,60174050);广东省自然科学基金(04009465,010059)
作者简介:陈 成(1983-),男,硕士研究生,研究方向为软件设计与实时系统;导师:张立臣,教授,研究方向为分布式与实时系统。

成具体的代码,例如特定针对.NET 或 J2EE 平台的标准转换[3]。

1.2 MDA 模型分层

MDA 的模型包括 PIM 和 PSM 上下两层,而 PIM 或者 PSM 又都可以分成若干层,每一层都由上一层产生。最上面的一层 PIM 实际上就是核心业务逻辑的分析模型,是完全与实现无关的特定对象结构和行为的抽象。当它向下转换成一个或多个 PIM 时,则需要加入规范,相关领域的信息以及通用的功能,但这一层的 PIM 仍然是与平台无关的。由上至下的顺序是从高度的架构模型到低层的设计模型,再到最底层的实现构造。上一层的模型加上新的模型和可选的参数决定了每一步的转换行为[4,5]。

图 1 显示了分别包括两层的 PIM 和 PSM 结构,最上面一层是帐户管理的核心业务逻辑,它结合权限控制转换成第二层提供了新功能的 PIM,这层的 PIM 通过模型转换产生了包括帐户管理和权限控制的适合 J2EE 平台的 PSM,最后由这层 PSM 生成 Java 源代码。如果要生成.NET 相关的产品,只需对最底层的 PIM 应用针对.NET 平台的转换即可。

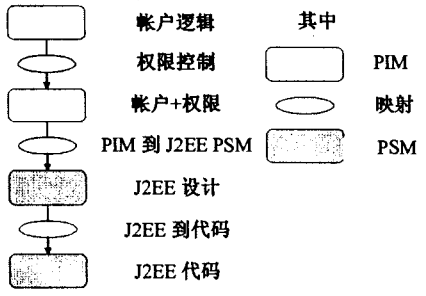


图 1 MDA 的模型分层

1.3 MDA 模型转换技术

模型转换的本质是读取源模型,按照转换规则将其转换为目标模型,模型转换目前有以下几种实现技术:

(1) 手动转换方法。由程序员使用操作模型的 API 进行转换,它是通过编程来实现的。

(2) 基于模板的代码生成技术。它是将模板作为转化规则,将源模型的每一个元素都映射为一个代码段。

(3) 基于关系代数的模型转换。将上下层模型中对应的元素映射表达为一个"关系对",将"模型转换"表达为一个二元关系或者一组二元关系。利用关系代数中的"关系"来表达模型转换的定义,"性质"来表达模型转换中的约束。

(4) 基于元模型间映射的模型转换。它利用元模型之间的映射进行模型转换。

2 AOP 介绍

2.1 AOP 是 OOP 的补充

Aspect oriented Programming (AOP),是一种弥补 OOP 限制的分离核心关注点和横切关注点的编程思想。它包括连接点(Join Point),切入点(Pointcut),通知(Advice),方面(Aspect),编织器(Weaver)。它将水平散布在各个对象中的,与领域问题不直接相关的逻辑,经分离后封装所得到的结果作为一个"方面",当到达由切入点表达式指定的连接点时,由 Advice 指定要执行的代码,然后由编织器将方面中的代码织入到设定的连接点(join points)。

面向对象编程(OOP)之所以能够广泛的应用,是因为它有效地把核心业务逻辑分解成松耦合的模块和类,然后再按所需要的功能结合到一起。然而像日志记录和错误处理等很多通用的服务是无法很好地封装在单个模块中的,它往往横切多个不相关的对象。如果在这些对象中加上实现通用服务的行为,会造成代码混乱、重复,降低开发效率。

AOP 可以清晰地分离关注点,像帐户逻辑可以作为功能组件,按 OOP 的思想进行开发。而像事务处理、安全检查、权限控制这样的通用服务可以作为方面单独编写,在合适的连接点将其织入到核心功能组件中。

2.2 面向方面的建模

面向方面的建模是一个比较新的领域,目前还没形成统一的建模的标准。UML 是建模语言的事实标准,然而它主要是应用于面向对象的建模,没有表示方面概念的图形,也无法表达方面与核心类的关系,更不能体现编织的概念。为了使 UML 能够进行面向方面的建模,通常需要对 UML 进行扩展。

UML 的扩展机制包括构造型(Stereotype)、标记值(tagged value)和约束(constraint),UML 允许继承原有的构造块来生成新的构造块,这些新的构造块就是构造型,它可以有与原构造块不同的语义,不同的约束,不同的标记值。如可以定义《aspect》为新的构造型。标记值可以为模型元素添加新的特征。约束扩展了建模语言的语义,可以增加新的规则或修改语义。

按照文献[3]实现的 UML 扩展方式,图 2 是对帐户的日志功能的方面建模。它描述了客户类调用帐户类的存款和取款方法时由 log 类写日志这一横切功能。模型由四种元素组成:aspect 类,普通类,相关的连接点,bind 关系。aspect 类定义了 logCall 和 methodCall 两个连接点以及 advice(确定在 methodCall 绑定的方法执行后调用 logCall 绑定的方法)。而连接点 logCall 和 methodCall 分别定义了它们与普通类(log,Cus-

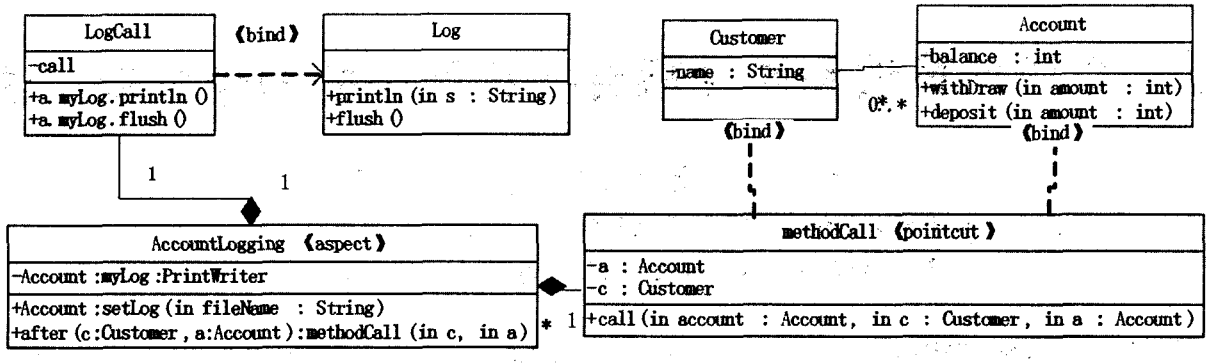


图2　对日志功能的方面建模

tomer,Account)相关的调用方法,在图中它们与普通类是一种binding关系。通过实现这样的扩展就可以很方便地对各种通用服务进行面向方面的建模。

3　面向方面的MDA

3.1　AOP在MDA中的作用

MDA的开发流程实际上就是图1中模型自上而下转换的过程。由于这个系统要提供像权限控制、事务管理等通用服务,所以表达核心业务逻辑的最上层的PIM需要若干次的转换才能得到符合所有需求的最底层PIM。但是一旦通用服务的需求发生了变动,模型的维护将变得非常困难。这与使用传统的OOP开发方法时在代码级别出现的代码"纠缠"和代码重复非常相似,只是层次不同而已。产生的原因在于没有在模型级别实现核心与方面功能的独立设计。将AOP思想应用于模型层面,则可以把MDA的模型转换(PIM到PIM,PSM到PSM)看作是将不同方面的模型和实现核心业务逻辑的模型组合起来生成有某种特定功能的模型。由AOP处理水平方向的各种问题领域,将其实现为面向方面的模型,由面向对象的模型处理核心业务逻辑,再将它们在适当的点编织起来产生其他的PIM或者是PSM。从而通过MDA的模型转换实现功能的集成。

开发的各个层面上都会涉及到方面,例如权限控制和事务管理存在于需求分析层(平台无关层),而记录日志和语言相关的错误处理则只出现在代码层(平台相关层)。因此需要根据通用服务所在的层次建立相应的面向方面的模型。

3.2　应用举例

图1的例子中,最上面的PIM是帐户管理的业务逻辑,第二层是增加了权限控制功能的PIM。如果用面向方面的思想重新考虑,则应该将权限控制作为平台无关的方面建模,再将它与业务逻辑PIM编织起来。

图3表达了如何在MDA中使用AOP,为了更完备的说明起见,对原例子增加了一些新的通用服务需求:事务管理、日志和Java错误处理。权限控制与事务管理分别被转换为J2EE相关的PSM,再与转化了的帐户逻辑的J2EE相关应用设计PSM"编织"成拥有核心业务功能、权限控制、事务管理功能的J2EE相关设计PSM。而日志设计独立地转换为日志相关的J2EE PSM,并没有在这一层与包含核心功能的PSM集成。最后经由PSM到代码的转化,包含业务逻辑实现的代码才与日志以及Java出错处理的代码编织起来,形成了实现全部功能的系统产品。

日志设计模型之所以没有在PIM转换成PSM后立即与应用设计模型组合到一起,是因为它还需要在PSM这一层进行进一步的定制和平台相关信息。所以应该在设计中根据方面所偏向的层次决定是否建模成PIM或者PSM,以及决定什么时候与包含核心业务逻辑的模块编织到一起。

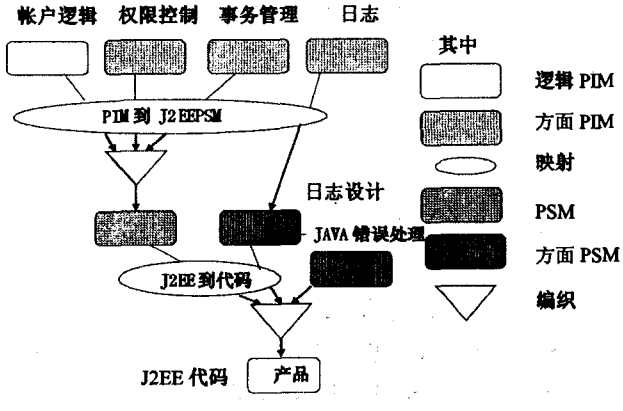


图3　MDA使用AOP的例子

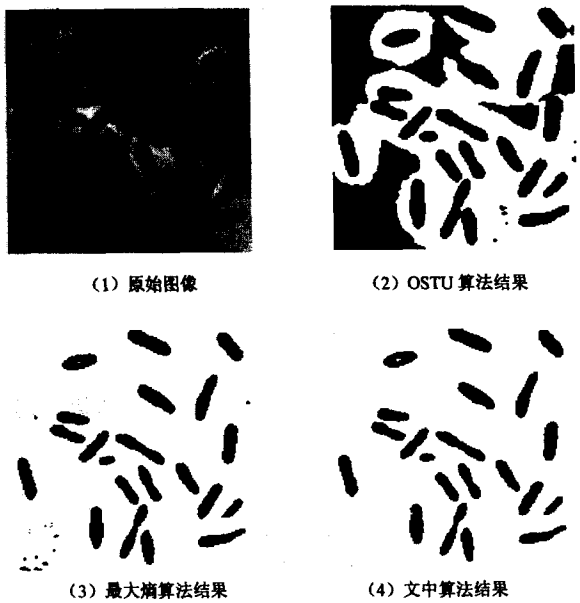
4　结束语

MDA通过模型转换自动生成软件产品,降低了应用功能设计和技术平台之间紧耦合的关系,但是由于横切功能的需求,在它的模型创建和转换阶段仍然会不可避免地出现模型纠结和重复。为了解决这个问题,将AOP思想提高到模型层面,通过扩展UML实

(下转第92页)

代,算法即终止,求得最大熵值为 8.3425。图 1(1)为原始图像,图 1(2)为 OSTU 算法分割的结果,图 1(3)为最大熵算法分割的结果,图 1(4)为文中算法分割的结果。从图中可以看出,OSTU 算法的结果最差,最大熵算法虽然把所有目标分割出来,但是也把一些背景分割成了目标(左下角处的小黑块),文中算法效果最好,不仅把所有目标分割出来,并且不存在误分割。

另外,从算法的执行时间上看,在主频为 2.4G,内存为 256M 的 PC 机上运行本算法,综合多次实验结果来看,一般迭代到第三次就可以得到最佳分割图像,时间开销约为 4～6s。相比人工实验确定实验参数的时间开销要有一个数量级的优势。



(1) 原始图像 (2) OSTU 算法结果

(3) 最大熵算法结果 (4) 文中算法结果

图 1 bcteria 图像及其实验结果

3 结束语

脉冲耦合神经网络是当前智能领域研究的热门课题,它广泛应用于图像处理的各个方面,如降噪、分割、边缘检测、目标识别、特征提取等。PCNN 模型参数的选择对实验结果有着极其重要的影响,但是针对不同图像这些参数的选择只能通过人工逐次实验来获得,这对其推广应用很不利。文中提出了一种基于 PSO 算法的脉冲耦合神经网络自动系统的实现方案,以改进的最大熵函数作为适应度函数,利用 PSO 算法的全局搜索能力,较好解决了 PCNN 需人工设置参数的问题,同时分割出来的图像质量效果较好,为后续的图像识别的精度提供了保证。

实验仿真结果说明文中的研究思想具有较好的应用前景,是对 PCNN 理论的进一步完善。显然,其对于 PCNN 的理论研究和实际应用具有很重要的现实意义和借鉴意义。同时,也应看到由于对视觉机制认识的不足,PCNN 模型离真正的视觉模型还有一定差距。因此,需要对 PCNN 模型进一步改进,使其具有更强的视觉仿生能力,以便在图像处理、计算机视觉研究中取得更大的成功。

参考文献:

[1] 吴启迪,汪 镭.智能微粒群研究及应用[M].南京:江苏教育出版社,2005.

[2] Johnson J L,Padgett M L. PCNN model and application[J]. IEEE Trans on Neural Network,1999,10(3):480－498.

[3] Ranganath H,Kuntimad G,Johnson J L. A neural network for image understanding[M]//In:Fiesler E,Beale R. handbook of Neural Computation. Oxford,UK:Oxford Univ. Press,1997.

[4] 马义德,戴若兰,李 廉.一种基于脉冲耦合神经网络和图像熵的自动图像分割方法[J].通信学报,2002,23(1):46－51.

[5] 马义德,齐春亮. 基于遗传算法的脉冲耦合神经网络自动系统的研究[J].系统仿真学报,2006,18(3):722－725.

(上接第 89 页)

现面向方面的建模,将核心业务功能与通用功能分离开来,使它们能够独立建模,然后通过由织入技术实现的模型转换将它们连接起来。把 AOP 的思想应用于 MDA 的建模与模型转换中为 MDA 开发方法提供了新的实现途径。

参考文献:

[1] 曹东刚,梅 宏.面向 Aspect 的程序设计———种新的编程范型[J]. 计算机科学,2003,30(9):5－10.

[2] 董云卫,郝克刚.一种面向方面的软件体系结构[J].微机发展,2004,14(6):52－56.

[3] Kandé M M,Kienzle J,Strohmeier A. From AOP to UML: Towards an Aspect－Oriented Architectural Modeling Approach[R]. Berne, Swiss:Swiss Federal Institute of Technology,2004.

[4] Zhao Jianjun,Xu Baowen. Measuring Aspect Cohesion[C]// In Proc. Fundamental Approaches to Software Engineering (FASE'2004). Barcelona,Spain:Springer－Verlag,2004:29－31.

[5] Zhao Jianjun. Slicing Aspect－Oriented Software[C]// In Proc. 10th IEEE International Workshop on Program Comprehension (IWPC'2002). Paris,France:[s.n.],2002:251－260.