

基于 MDA 的行政审批业务网办系统^①

何燕萍 范冰冰 (华南师范大学 计算机学院 广东 广州 510631)

摘要: 针对行政审批业务网办系统早期开发存在的效率低下、优化改造困难等问题,提出一种基于模型驱动架构的开发方法,并以该系统某业务处室预受理子系统为开发实例,重点讨论了系统的分析和设计过程。实践证明,该方法克服了传统精细手工开发存在的缺陷,保证了各业务处室预受理子系统的一致性,是一种非常适合网办系统开发的新方法。

关键词: 模型驱动架构; Web 应用系统; UML; 开发方法

Online Administration Examination and Approval System Development Based on MDA

HE Yan-Ping, FAN Bing-Bing(School of Computer, South China Normal University, Guangzhou 510631, China)

Abstract: In order to find a solution to the low development efficiency and difficult optimization of the the online examination and approval system,this paper presents a development approach which is based on Model Driven Architecture. A government department subsystem is given as a case study to focus the analysis and design of this approach. Practice has proved that this method overcomes the shortcomings brought on by the traditional development approach and promises the consistency of all the subsystems. So it is an especially suitable method for the system.

Keywords: MDA; web application system; UML; development approach

1 概述

某市劳动与社保行政审批业务网上办理系统(以下简称网办系统)是基于 B/S 架构的大型电子政务系统,通过 Web 方式实现开放互联网上的业务服务窗口,企业或个人均可在线填报业务申请、上传业务材料。现阶段业务主要涉及该市社会保障局医保中心,基金中心等多个业务处室,各业务处室网上预受理子系统构成网办系统。在系统早期基于传统的精细手工开发过程中,出现不少问题:各子系统业务办理主流程一致,只存在少量业务细节上的差异,但只能用手工作按处室分别进行重复开发,重用性低,代码的质量依赖于开发人员的水平。同时,网办系统的开发是一个长期的过程,系统对新技术适应困难,多次需求变更使得最初系统的分析和设计模型变得过时。因此,需要一种工厂化生产的方法来开发此 Web 应用系统,在保证系统质量的前提下,最大程度缩短系统开发周期,提高开发效率。

模型驱动架构(Model Driven Architecture,MDA)为上述问题提供了一种新的解决方法。MDA 通过下面两种方式改变了软件开发的重点:(1)模型比代码更有价值。(2)模型将更精确,模型中不会允许随意分隔分析和设计^[1]。MDA 定义了计算独立模型(Computation Independent Model, CIM)、平台独立模型(Platform Independent Model, PIM)和平台相关模型(Platform Specific Model, PSM)。开发人员将精力集中在网办系统主业务流程 PIM 上,不必维护 PSM,根据 PIM 运用映射规则自动生成主业务流程 PSM。模型建立后,各处室预受理系统的主业务流程代码可由代码生成工具自动生成,确保各子系统主业务流程代码的一致性,避免了由于开发人员开发水平导致的各子系统质量参差问题。当业务流程需求变化时,这些变更将在 PIM 中反映出来并在 PIM 中测试,然后运用映射规则生成新的 PSM。

^① 收稿时间:2010-03-11;收到修改稿时间:2010-04-18

通过这种方法, PSM 永远与 PIM 保持同步。同时, 由于 PIM 是跨平台的, 可适应不同平台和新的技术, 这样, PIM 不会因为技术的变化而过时, 使得支持新技术或者不同技术的任务变得容易起来。因此, 使用 MDA 方法, 提高了网办系统一致性、重用性、可维护性, 实现对新技术的快速适应, 降低系统开发成本和缩短开发周期, 延长了系统的生命周期。

最好的技术不是最新的技术, 而是最合适的技术, MDA 正是最适合网办系统开发的技术。本文将 MDA 的思想与网办系统的特点相结合, 提出一种基于 MDA 的网办系统开发方法。

2 基于MDA的网办系统开发框架

2.1 网办系统体系结构

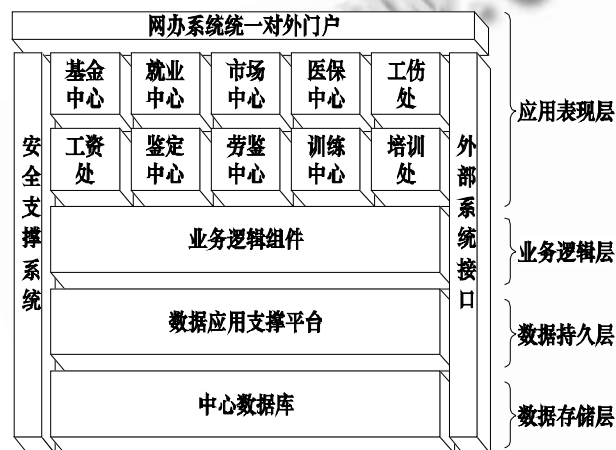


图 1 网办系统体系结构图

如图 1 所示, 网办系统体系结构分为四层: 应用表现层、业务逻辑层、数据持久层和数据存储层。分层设计有利于“解耦”代码, 易于维护。下面对各层进行介绍:

应用表现层: 为用户提供人机交互操作的界面, 负责把业务逻辑层传递的信息展现给用户以及把用户操作的数据传递给业务逻辑层处理。

业务逻辑层: 负责网上业务申请、审批等业务流程、业务规则的处理, 并且把处理结果传递给应用表现层或者数据持久层。

数据持久层: 直接和数据库交互, 负责向数据库提取或保存数据, 并不参与业务逻辑运算和界面展示, 从而从架构上隔离了业务逻辑层、应用表现层对数据

库直接操作的弊端。

数据存储层: 负责网办系统业务中心数据的存储。

2.2 基于 MDA 的网办系统开发框架

针对网办系统体系结构, 本文提出了一个基于 MDA 的网办系统开发框架(图 2)。CIM 对应软件生命周期中的业务模型, 抽象层次最高, 着眼于网办系统需求描述。PIM 对应分析模型, 创建 PIM 的过程中立于任何实现平台, 着眼于网办系统整体架构。一个 PIM 可以转换成多个 PSM。PSM 对应设计模型, 包含了所有 PIM 中表示的功能, 并添加了针对具体实现平台的设计思想和实现技术。

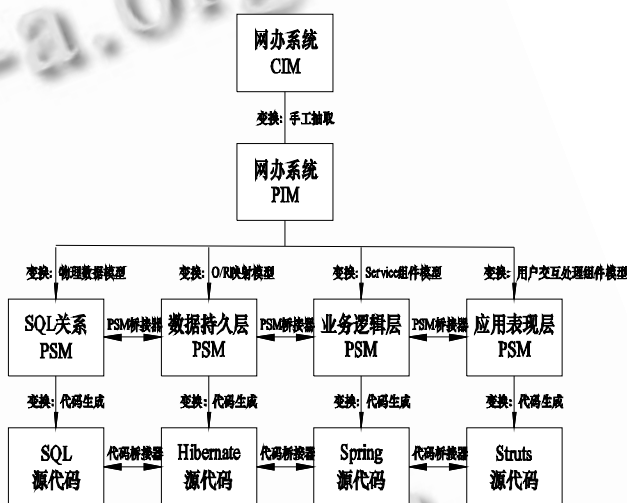


图 2 基于 MDA 的网办系统开发框架

2.2.1 建立计算无关模型(CIM)

在实际工程中, 由于缺少领域模型, CIM 并没有形成通用的标准, 直接通过工具完成 CIM 到 PIM 的转换实现技术并不成熟。为使开发人员能够通过 CIM 与用户进行充分有效的沟通和交流, 网办系统以各业务处室的用户需求文档作为 CIM, 并使用用户易于理解的文字和图形进行描述。

2.2.2 建立平台无关模型(PIM)

PIM 主要包括静态模型和动态模型。其中, 静态模型以静态类图(类图及关联)为主, 动态模型包括顺序图、活动图、协作图等描述系统动态行为特征的模型。PIM 必须具备由第三方自动化工具生成 PSM 所需要的足够信息量, 因此 PIM 的建立至关重要。PIM 层的抽象程度越高, 其工具性越强, 但对使用者的要求也就更高^[2]。为了能够更加精确地描述网办系统, 需要不

断地根据用户需求等各种因素对 PIM 进行精化。UML 在 MDA 中可被用于描述各种模型，在网办系统开发过程中，使用 UML profile 来描述 PIM，实现简单，使用者上手容易。

2.2.3 模型转换

(1) CIM->PIM: 为保证对用户需求的准确理解和完整表达，CIM 到 PIM 的转换直接以人工抽取的方式完成。

(2) PIM->PSM: PIM 到 PSM 的转换是 MDA 开发的关键，转换过程由工具在一定转换规则的指导下进行。现阶段，越来越多转换工具支持基于 OMG 制定的 QVT(Query View Transformation)规范书写的转换规则。QVT 并没有给出具体模型转换到代码的规则，但给出如何来编写这些规则的标准。由于 QVT 规范定义的转换具有语义定义良好，表达能力强等特点，本文使用 QVT 中用户友好的 Relations 语言刻画转换规则。Relations 语言能够支持复杂的对象模式匹配和对模板生成，并且在模型转换中，隐式生成模型元素的转换记录，便于开发人员追踪[3]。

网办系统有多个 PSM，因此还需要生成 PSM 之间的通信桥接器，以实现跨平台互操作性。以下对网办系统 PIM 到各层 PSM 转换进行概述：

① PIM 到 SQL 关系 PSM 的转换：

将 PIM 中的实体类与联系(Relationship)标记为数据库实体类与关系，直接得到并不依赖于具体的计算机操作系统或者某个具体的 DBMS 的概念数据模型(CDM)。此时，根据网办系统使用的 DBMS，通过一组无歧义的转换规则进一步转换为物理数据模型(PDM)。

② PIM 到数据持久层 PSM 的转换：

网办系统需要与社保局内部系统交互，经过友好协商，内部系统基本表结构对网办系统公开，网办系统持久层框架使用 Hibernate。通过 Hibernate 的 O/R 映射机制，可建立 SQL 关系 PSM 中的数据表和持久层 PSM 中实体类间的映射。

③ PIM 到业务逻辑层 PSM 的转换：

网办系统业务逻辑层使用 Spring 技术。该层涉及多种业务逻辑处理，需要根据 PIM 中的实体类生成对应的 Service 组件模型。目前 PIM 到业务逻辑层 PSM 的 MDA 转换工具能力并不强，例如，AndroMDA 目前已提供 Spring 的转换包 AndroMDA-Spring，但

AndroMDA 仅支持 Spring 框架的部分 IoC 功能。因此，本文 PIM 到业务逻辑层 PSM 的转换借鉴了文献[4]中的方法。

④ PIM 到应用表现层 PSM 的转换：

Sybase 公司的 PowerDesigner，Interactive Objects 公司的 ArcStyler、开放源码的 AndroMDA 等工具均支持完成 PIM 中实体类到页面表单、用户交互处理组件模型的映射，从而生成对应的 PSM，本文使用 ArcStyler 完成该转换。

有时候由 MDA 工具自动生成的 PSM 并不能令人满意，需要根据网办系统所使用的平台特性对 PSM 加以修改并精化，以更好地指导代码生成器生成高质量的代码。与此同时，开发人员对 PSM 的改变能够反映到 PIM 中去。

(3) PSM->Code: 该转换是由 MDA 工具根据 PSM 模型直接生成代码的过程，在此不再详述。

3 实例分析

以基于 MDA 实现的网办系统某处室预受理子系统为例说明系统设计的过程。其余业务处室的预受理子系统主业务流程与该子系统相同。

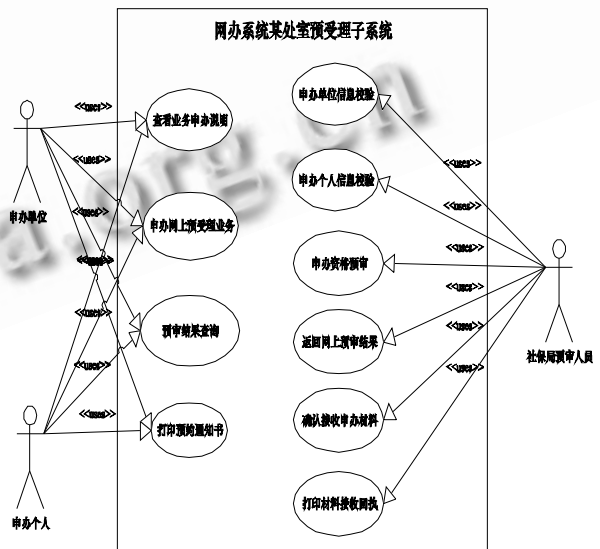


图 3 预受理子系统 CIM

步骤 1：建立预受理子系统 CIM

通过以座谈会和各种问卷调查的方式，经过与社保局该处室的业务领导多次交流，归纳出该处室的需求分析文档，并建立该处室预受理子系统的用例模型，

如图 3 所示。

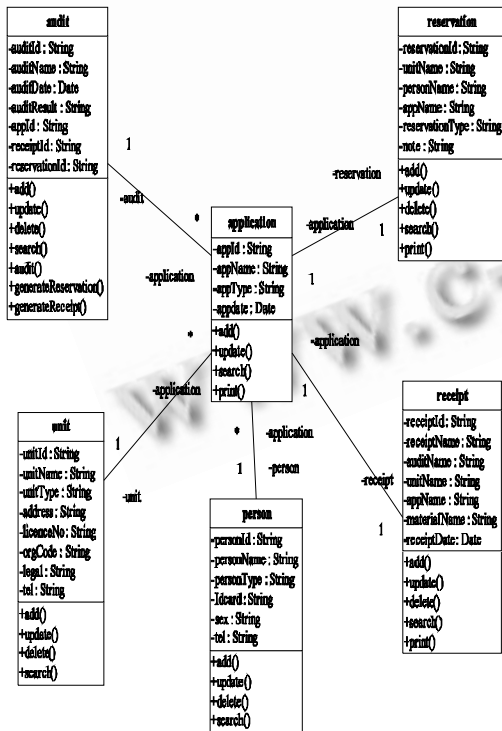
步骤 2: 抽取预受理子系统 PIM

根据需求文档和用例模型抽取该子系统 PIM, 建立 PIM 静态模型, 如图 4(a)所示, 该模型中含有六个类: person(申办个人类), unit(申办单位类), application(申办业务类), audit(预受理信息类), reservation(业务预约类), receipt(收件回执类), 每个类均有一个该类的唯一标识 ID。图 4(b)显示了该子系统 PIM 动态模型。

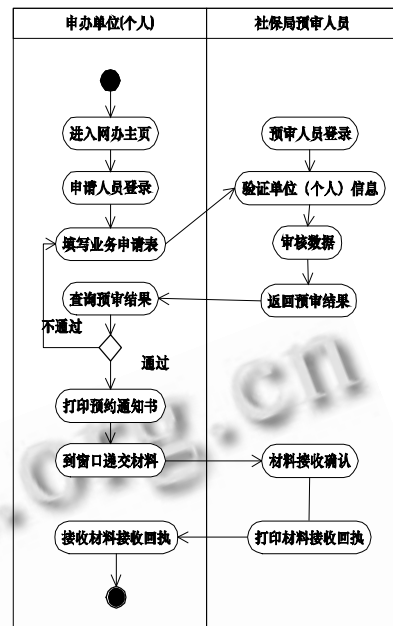
步骤 3: 自动生成预受理子系统 PSM

通过 QVT 规范来定义映射规则, 利用 MDA 工具自动实现 PIM 到 PSM 的转换。例如, PIM 静态类模型变换规则定义如下:

- (1) 对 PIM 中的每个类, PSM 中均要有一个与之同名的类相对应。
- (2) 对 PIM 类中的每个公有属性, PSM 中均要进行如下操作:
 - ① 生成一个和该属性同名的私有属性。
 - ② 为该私有属性生成对应的 get、set 公有操作, 即分别在属性名前加上“get”“set”前缀, 生成以属性类型为返回类型的操作方法。



(a) 预受理子系统静态类图



(b) 预受理子系统活动图
图 4 预受理子系统 PIM

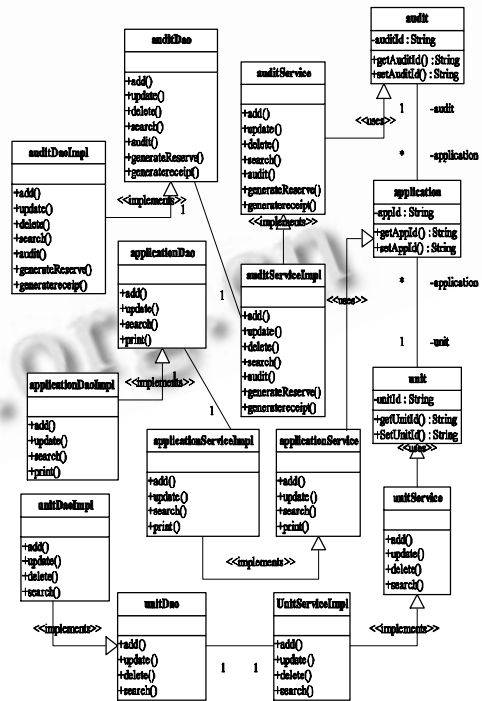


图 5 预受理子系统 PIM 类转换后的 PSM 类

另外, QVT 规范可用于逆向工程, 利用上述规则的逆规则可实现 PSM 到 PIM 变换, 有助于将网办系统中早期基于传统精细手工开发方法的部分处室预受理子系统纳入 MDA 开发框架。

(下转第 131 页)

(上接第151页)

图5显示了根据预受理子系统 PIM 类变换后的部分 PSM 类,一些无关或者重复的模型细节被省略。

步骤4:自动生成预受理子系统 Java 代码

此过程是 MDA 工具自动生成的过程,不再详述。

4 总结

在网办系统早期基于传统手工精细开发的过程中,由于缺乏合理的开发框架,随着设计的进展,分析投资的成果不断减少,软件熵在开发过程中不断增加。通过在网办系统开发中使用基于 MDA 的开发方法,结合 Java Web 中成熟稳定的框架技术,开发人员可充分复用已有的实现模型,从而将精力集中在系统模型的分析与设计,提高了系统开发效率并保证各业务处室预受理子系统主业务流程的一致性。同时,通过 QVT 规则的逆规则实现 PSM 到 PIM 变换,有助于将系统早期使用传统精细手工方法开发的预受理子系统纳入 MDA 开发框架。本方法已成功应用于网办

系统部分新业务处室预受理子系统的开发中,效果良好,将在网办系统后续业务处室子系统开发中推广使用。同时,由于现阶段 MDA 工具仍有一定的局限性,代码无法达到 100%自动化生成,部分转换仍需开发人员手工完成,但是,基于 MDA 的开发方法符合历史潮流,相信在不久的将来,网办系统工厂化生产会成为现实。

参考文献

- 1 Chris Raistrick, Paul Francis. Model Driven Architecture with Executable UML.赵建华,张天等译.北京:机械工业出版社,2006:10—50.
- 2 徐晓钟.模型驱动架构(MDA)相关技术研究 with 实现.计算机科学,2006,33(12):278—279.
- 3 MG.Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification. <http://www.omg.org/spec/QVT/1.0/PDF/>, 2008-04-03/ 2009-12-20.
- 4 姚易.面向Spring框架的MDA模型转换方法研究[硕士学位论文].吉林:东北师范大学,2009.