特定领域建模与代码生成的研究与实现①

Research and Implementation of DSM and Code Generation

杨兴涛[1]　苏桂平[2]　王瑞芳[3]　王小芳[4]

1 (中国科学院研究生院 工程教育学院 北京 100049)

2 (中国科学院研究生院 信息科学与工程学院 北京 100049)

3 (大连大学 信息工程学院数学系 辽宁 大连 116622)

4 (吉林大学 数学研究所 吉林 长春 130012)

摘　要： DSM 通过提高抽象程度实现软件生产力的提高。本文在深入研究 DSM 与元建模以及代码生成等技术后，提出一种特定领域建模与代码生成的方法，并建立起它的实现。本文归纳总结了创建特定领域元模型及其实现的方法与原则；介绍了使用 GMF 通过特定领域元模型快速定制生成特定领域建模工具的过程；提出基于系统架构与框架创建覆盖不同业务类型的样例代码的方法与原则；总结归纳了特定领域建模与代码生成的整体过程；提出根据特定元模型将样例代码制作成代码模板的方法与原则；最后介绍了通用代码生成引擎的组成部分及其作用以及代码生成的过程。本文提出的特定领域建模与代码生成的方法，可生成 100%的可直接运行代码，是提高软件开发效率与产品质量的有效途径。

关键词： 特定领域建模 元模型 系统框架 样例代码 代码模板 代码生成

1　引言

程序语言向着人类认识世界和解决问题的方式的方向发展，即提高抽象程度。抽象程度决定着软件生产力。高级程序语言并非是对低级程序语言的抛弃，而是兼容，低级程序语言作为高级程序语言的一个支撑层次而存在。所以单独的建模语言对提高软件生产力的帮助并不明显。DSM(Domain-Specific Modeling)强调生成 100%的代码，期望通过积累形成新的程序语言，从根本上提高软件生产力。

模型是为理解事物而对事物的一种抽象，通常由符号与符号的组织规则构成。建模工具是模型的图形化表达，增强了模型的易理解、易操作等特性。代码模板是将模型转变为技术实现的手段，是模型到技术实现的中间映射层。模型、建模工具、代码模板分别通过符号化、图形化、模板化三个过程实现了抽象程度的提高。

本文在深入研究 DSM 与元建模以及代码生成等技术后，提出一种特定领域建模与代码生成的方法，并建立起它的实现。本文归纳总结了创建特定领域元模型及其实现的方法与原则；介绍了使用 GMF(Graphical Modeling Framework)通过特定领域元模型快速定制生成特定领域建模工具的过程；提出基于系统架构与框架创建覆盖不同业务类型的样例代码的方法与原则；总结归纳了特定领域建模与代码生成的整体过程；提出根据特定元模型将样例代码制作成代码模板的方法与原则；最后介绍了通用代码生成引擎的组成部分及其作用以及代码生成的过程。本文提出的特定领域建模与代码生成的方法，可生成 100%的可直接运行代码，是提高软件开发效率与产品质量的有效途径。

2　创建特定领域元模型

元模型就是用来描述模型的模型，如使用 UML 类图描述系统的结果就是模型，对类图中各个元素(如

① 基金项目:国家 863 计划项目(2001AA141010)
收稿时间:2008-10-09

类、接口、关联等)的定义就是元模型[1]。通过创建特定领元模型就可以由 GMF 快速配置生成特定领域建模工具。

2.1 创建特定领域元模型的原则

通常由领域专家提取特定领域关键概念，创建特定领域元模型。同时，由于特定领域建模的另一个目标是生成 100%的代码，所以创建特定领域元模型也要考虑技术实现，即创建特定领域元模型的另一个来源是与技术实现的完整映射。

系统架构通常存在两个特殊层次，用户界面层与数据访问层，这两个层次是软件系统的特有概念。本文将用户界面层与数据访问层作为特定领域建模的共性层次，即为特定领域模型建立两个通用模型，用户界面模型与数据访问模型。

本文提出的创建特定领域元模型应遵循的原则，如下：

(1) 通过特定领域分析提取特定领域关键概念建立特定领域核心元模型，并建立核心元模型之间的关联关系。

(2) 建立特定领域核心元模型的多角度视图的元模型，如静态结构、动态行为、物理部署等视图的元模型。

(3) 提取用户界面关键概念建立用户界面元模型，如样式、布局、界面组件、交互数据等。

(4) 提取数据访问关键概念建立数据访问元模型，如对象关系映射、JDBC、数据库表结构、XML 等。

(5) 提取技术实现的关键概念建立辅助配置管理元模型，为代码生成提供配置管理辅助工具。

(6) 对同一领域关键概念使用统一名称进行标识。

上述 6 条原则从特定领域与技术实现两个方面保证了所建立的特定领域元模型的完整性。

2.2 使用 XML 模式描述特定领域元模型

DSM 并没有提供具体的技术实现，本文使用 EMF(Eclipse Modeling Framework)创建特定领域元模型。EMF 支持 3 种类型输入，本文使用 XML 模式描述特定领域元模型，并作为 EMF 的输入，生成特定领域元模型 Ecore。

使用 XML 模式描述特定领域元模型的方法，如下：

(1) 将关键概念定义为 XML 模式的 Element。

(2) 将关键概念的性质或属性定义为 Attribute。

(3) 为 Element 或 Attribute 定义简单或复杂类型。

(4) 将关键概念之间的关联关系定义为 Element，并嵌套添加子 Element，分别指向关联端的 Element。

(5) 将成对或成组出现关键概念定义为 group。

使用 XML 模式作为 EMF 输入的优势在于可通过 XML 模式对特定领域模型进行语法与一致性等模型验证工作。针对特定领域模型多角度视图，为防止不同视图存在相互冲突，必须进行一致性检查验证[4]。本文对系统功能、结构、行为等模型描述进行一致性检查，检查措施有 XML 模式约束、错误定位提示、自动检测修正等。一致性检查验证的基础是特定领域元模型创建原则的第 6 条。

3 特定领域建模

DSM 从大幅度提高软件生产力的角度提出特定领域建模，它提取特定领域关键概念建立特定领域建模工具，进行特定领域建模，结合特定领域代码生成器直接生成最终产品。它强调特定领域建模，模型可以只适用于本领域，甚至是本产品或项目。

3.1 使用特定领域元模型配置生成特定领域建模工具

目前，通常使用 GMF 实现创建特定领域建模工具。GMF 提供图形化定制环境，开发人员可在图形环境中快速轻松定制生成自己的特定领域建模工具。

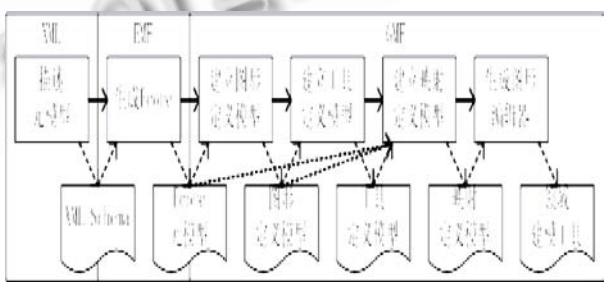

图 1 配置生成特定领域建模工具

如图 1 所示，本文使用 GMF 创建特定领域建模工具的具体过程如下：

(1) 输入 XML 模式，通过 EMF 生成元模型 Ecore。

(2) 建立图形定义模型，输入 Ecore，定制图形定义内容，生成.gmfgraph 文件。

(3) 建立工具定义模型，输入图形定义模型，定制工具定义内容，生成.gmftools 文件。

(4) 建立映射定义模型，输入 Ecore、图形定义模型、工具定义模型，定制 Ecore 与图形工具之间的映射，生成.gmfmap 文件。

(5) 通过映射定义模型，自动生成完整的特定领域建模工具。

由 GMF 建立的特定领域建模工具具有图形化可拖拽的操作界面，可以非常方便的进行特定领域建模。

3.2 以系统架构与框架为基础建立样例代码

特定领域元模型的创建必须考虑技术实现，但技术实现并不单纯服务于特定领域元模型，它同时还提供代码模板制作的基础，样例代码。

系统架构是系统的高层规划，提供系统划分与设计的方法，定义系统静态结构与动态交互等关系，在高层为系统搭建提供理论指导。目前常用的有 MVC(Model、View、Controller)三层架构。系统框架是系统架构指导下对系统的共性部分进行抽象提炼形成的实现基础，在此之上开发完整的业务系统。目前常用的有 SSH(Struts，Spring，Hibernate)框架。系统架构与框架是软件发展成熟的表现，它以复用为手段，使软件开发不必从零开始。

样例代码是面向业务以系统框架为基础建立的完整可运行的程序代码，它说明了以系统框架为基础如何开发业务系统的问题。要实现生成 100%的代码必须参照完整的样例代码制作代码模板，即对样例代码整体模板化。以 SSH 框架为例，需要将包括 Jsp、Config、Action、Application Context、Hibernate hbm、Po 等在内的所有程序资源制作成代码模板。这就要求样例代码的完整性及对不同业务类型的覆盖度。

本文提出创建完整的样例代码应遵循的原则，如下：

(1) 针对不同业务类型建立样例代码，如信息管理类、流程审批类、信息发布类、决策支持类、数据检索类等。

(2) 以(1)为基础，建立不同业务实体关联关系的样例代码，如 1 对 1 关联、1 对多关联、多对多关联等。

(3) 以(2)为基础，建立不同界面表现的样例代码。

3.3 特定领域建模与代码生成

特定领域建模是代码生成的基础，基于系统架构与框架建立完整的样例代码是制作代码模板的前提，特定领域建模与完整的代码模板是生成 100%的可直接运行代码的保证。

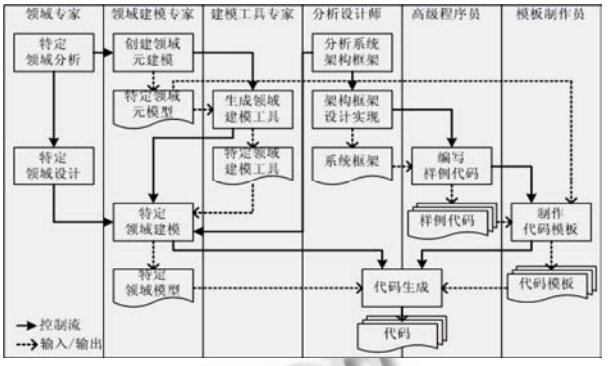

图 2　特定领域建模与代码生成

如图 2 所示，本文提出特定领域建模与代码生成的方法具体过程如下：

(1) 领域专家进行特定领域分析，提取领域关键概念。

(2) 领域建模专家使用 XML 模式描述领域元模型。

(3) 建模工具专家输入 XML 模式通过 EMF 产生特定领域元模型 Ecore，通过 GMF 定制生成特定领域建模工具。

(4) 系统分析设计师分析设计实现系统架构与框架。

(5) 高级程序员编写不同业务类型的完整样例代码。

(6) 领域建模专家进行特定领域设计，通过特定领域建模工具进行特定领域建模，并对模型进行检查模型验证。

(7) 模板制作员将样例代码制作成完整的代码模板。

(8) 开发人员结合特定领域模型与代码模板，使用通用代码生成引擎生成 100%的可直接运行代码。

与传统的建模与代码生成技术相比，本文提出的特定领域建模与代码生成的方法，从多个角度以多种手段保证了最终生成 100%的可直接运行代码。

4　代码模板制作和代码生成

代码生成通常使用代码模板通过代码生成器生成代码。代码模板的好处在于代码模板的易维护性与代码生成器的通用性，针对不同领域模型与系统框架，可快速删减或修改代码模板以响应环境的变化，而不需要修改代码生成器。

4.1 代码模板制作

代码模板是代码的雏形，是对代码的抽象表达，它固化代码的共性部分，标记特性部分，在代码生成时，通过替换将特性部分转化成实际需要的代码。代码模板的共性部分也称为静态内容，特性部分也称为动态内容。

代码模板的特性部分有两个来源：特定领域模型及系统架构与框架。如针对业务表的增删改查，共性部分是增删改查，特性部分是各个业务表。又如数据库连接，共性部分是连接数据库，特性部分是数据库的类型、名称、访问地址等。样例代码是系统架构与框架的具体体现。代码模板的特性部分需要由领域模型与样例代码共同保障。

本文提出代码模板制作的 2 点保障机制，如下：

(1) 建立完整的特定领域模型，涵盖代码模板体现的所有特定领域关键概念。

(2) 建立完善的样例代码，涵盖代码模板体现的所有平台技术相关概念。

本文使用基于签替换原理的模板引擎 Velocity 的 VTL(Velocity Template Language)制作代码模板。标签替换就是根据 VTL 标记的名称在特定领域模型中查找相应的值进行替换，形成具体代码。VTL 通过${}与#指令标记代码模板的特性部分。${}引用动态数据，如变量、对象等，#指令建立模板语言的控制结构，如选择、循环、引入文档等。

4.2 通用代码生成引擎与代码生成

通用代码生成引擎通常由读入器、验证器、生成器、编写器、配置工具 5 部分组成，共同完成代码生成。本文在模板引擎 Velocity 应用程序接口的基础上进行了功能扩展，建立起通用代码生成引擎的一个具体实现，如图 3 所示：

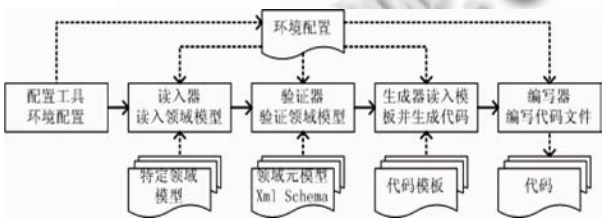

图 3 通用代码生成引擎与代码生成过程

各部分功能如下：

读入器：读入模型与代码模板并放入环境上下文中。

验证器：根据特定领域元模型检查验证特定领域模型。

生成器：动态加载 VTL 标记的特性内容，形成代码。

编写器：将代码写入到目标代码文件中。

配置工具：配置生成环境，如资源位置、编码格式等。

本文基于通用代码生成引擎的代码生成过程如下：

(1) 通过配置工具配置代码生成环境，加载类库，指定资源存放位置，设置输入/输出编码格式、日志等信息。

(2) 读入器根据环境配置中资源存放位置，读入特定领域模型与代码模板，并放置入环境上下文。

(3) 验证器根据环境配置中资源存放位置，读入特定领域元模型，对特定领域模型进行检查验证工作。

(4) 生成器基于标签替换原理将 VTL 标记的特性内容替换为特定领域模型中的相应的值，并形成代码。

(5) 编写器将代码写入到代码文件中，完成代码生成。

5 结束语

本文提出一种特定领域建模与代码生成的方法。通过建立特定领域元模型、配置生成特定领域建模工具、分析设计实现系统架构与框架、建立样例代码、制作完整的代码模板、建立通用代码生成引擎，最终实现了特定领域建模与生成 100%的可直接运行代码。实践表明，特定领域建模与可生成 100%的可直接运行代码的代码生成方法是提高软件生产力的有效手段。

有效的领域分析与设计方法是提取特定领域关键概念与创建完整的特定领域元模型的基础；与传统建模技术相比，基于 DSM 的软件开发关键在于能否快速有效的建立特定领域建模工具。这都将是本文以后研究的方向。

参考文献

1 刘辉,麻志毅,邵维忠.元建模技术研究进展.软件学报, 2008,19(6):1317－1327.

2 林慧苹,范玉顺,黄琛.产品设计中基于元模型的知识管理.计算机集成制造系统, 2007,13(4):663－667.

3 袁峰,李明树.基于MDA的TRISO-Model模型管理方法及应用.软件学报, 2007,18(7):1612－1625.

4 王云,刘又诚,周伯生.UML 可视化建模系统的模型一致性检查机制.计算机研究与发展, 2000,37(1).

5 熊鹏程,范玉顺,胡耀光.MDA 实施过程中的数据一致性研究.机械与电子, 2006,6:3－6.