

# 基于 MetaEdit+ 的元建模<sup>①</sup>

王海林, 李建芬

(山西财经大学 信息管理学院, 太原 030006)

**摘要:** 统一建模语言 UML 的广泛使用大大提升了模型在软件开发中的作用, 以模型为核心的软件开发方法逐渐被人们所接受。但由于 UML 是作为一种通用建模语言而设计的, 它既庞大而又复杂, 所以不容易为领域专家所学习和掌握, 常常使软件设计人员陷入繁重的建模工作而不能脱身。使用面向特定领域的建模语言 DSL(Domain Specific Language)可以很好地解决这一问题, 但手工设计 DSL 成本很高, 因此利用工具生成 DSL 成为唯一的选择。元建模(meta-modeling)就是一种通过工具建立领域元模型进而生成 DSL 的技术, 以 LED 千足虫设计为例详细阐述了基于元建模工具 MetaEdit+ 的元建模方法。

**关键词:** UML; 元建模; MetaEdit+; LED 千足虫设计

## Meta-Modeling Based on MetaEdit+

WANG Hai-Lin, LI Jian-Fen

(College of Information Management, Shanxi University of Finance and Economics, Taiyuan 030006, China)

**Abstract:** The wide use of UML(Unified Modeling Language) dramatically promotes model status in software development. Model based software development approaches have gradually been accepted. UML, however, is large and complex because it is designed as a generic modeling language. So it is not easy for domain experts to learn. Software designers often fall into the heavy modeling work and can not get out of it. Using DSL(domain specific language) can solve this problem well. Generating DSL by using tools becomes the only choice because the manual design cost of DSL is very high. Meta-modeling is a technique that generates DSL through building domain meta-mode. In this paper, we take LED millipede design as an example to introduce in detail the MetaEdit+ based meta-modeling approach.

**Keywords:** UML; meta-modeling; MetaEdit+; LED millipede

## 1 引言

目前软件开发的主流方法是面向对象的开发方法<sup>[1]</sup>, 统一建模语言 UML(Unified Modeling Language)作为一种用于面向对象软件开发的标准建模语言因此而得到了广泛地应用<sup>[2]</sup>。UML 是一种通用的、可视化的建模语言<sup>[3]</sup>, 它的广泛应用大大提升了模型在软件开发中的作用, 以模型为核心的软件开发思想已逐渐为人们所接受。但由于 UML 是作为一种通用建模语言而设计的, 它既庞大而又复杂, 所以不容易学习和掌握, 尤其是对领域专家, 这就使得建模工作只

能由软件设计人员和领域专家合作来完成, 而实际建模的工作量往往很大, 再加上软件设计人员又对领域知识不熟悉, 需要花时间学习, 致使软件设计人员耗费了大量的时间和精力, 造成不必要的资源浪费。模型驱动开发方法 MDD(Model Driven Development)<sup>[4]</sup>就是一种以模型为核心的开发方法, 它与基于 UML 的面向对象的开发方法的一个重要区别在于对应用领域的重视, 它不要求建模工具“大而全”, 只要求其“少而专”。也就是说, 它不要求像 UML 这样的通用建模语言提供很多很全的功能, 只要求提供一种面向

① 基金项目: 山西省省筹资金资助出国留学人员项目(晋留学管办发 2009-4 号)

收稿时间: 2010-10-14; 收到修改稿时间: 2010-11-16

特定领域的建模语言 DSL(Domain Specific Language)能快速有效地解决实际问题即可。由于 DSL 是面向特定领域的,其提供的功能仅限于本领域且为领域专家所熟知,所以简单易学,整个建模工作可由领域专家独自完成。

如何提供 DSL 成为建模的关键。手工为不同的领域开发不同的 DSL 代价很高,显然行不通。借助工具软件为不同的领域设计不同的 DSL 可以大大降低开发成本,是一种可行的方法。通常我们把开发 DSL 的工具软件称为元建模工具(meta-modeling tools),为设计 DSL 所建的模型称为元模型(meta-model),而建立 DSL 元模型并提供或生成支持 DSL 的建模工具的过程称为元建模(meta-modeling)。

## 2 MOF的层次结构

元模型是模型的模型,它是对模型在更高层次上的抽象,这种抽象可以继续下去,元模型的模型称为元元模型(meta-meta-model),理论上讲,这种抽象还可以继续进行下去,但抽象到元元模型这一级就已经能满足实际应用的需要了。为规范元建模过程,国际对象管理组织 OMG(Object Management Group)制定了元对象机制 MOF(Meta-Object Facility)<sup>[5]</sup>,其结构如表 1 所示:

表 1 MOF 的层次结构

层次	说明	元素举例
M3	元元模型层	元类,元属性
M2	元模型层	类,属性
M1	模型层	职工,年龄
M0	实例(或对象)层	张华,35

在表 1 中,M0 层元素是 M1 层元素的实例,M1 层元素是 M2 层元素的实例,M2 层元素是 M3 层元素的实例。UML 是建模语言,UML 所建模型处于 M1 层,而 UML 元模型处于 M2 层。同理,DSL 所建模型处于 M1 层,而 DSL 元模型处于 M2 层。MOF 位于 M3 层。本文所述的元建模是针对处于 M2 层的 DSL 元模型而言的。

## 3 MetaEdit+

目前元建模工具有很多,比较有名的有 MetaEdit+、GME、DOME、EMF、GMF 等<sup>[6]</sup>,本文

介绍基于 MetaEdit+的元建模方法。

MetaEdit+是一款出现较早、使用最广的商用元建模工具,第一个商用版本 MetaEdit+1.1 发布于 1993 年,目前最新版本为 MetaEdit+4.5。和 UML 一样,MetaEdit+是一个基于图形的建模工具,与 UML 不同的是它主要用于元建模。实际上 MetaEdit+是一个建模工具集,它提供了一组建模工具,如 diagram editors, matrix editor, table editor, browsers, report and code generation, method tools, API & XML connectivity, object repository 等<sup>[7]</sup>。使用 MetaEdit+可以大大缩短软件开发周期,软件开发效率提高 5-10 倍<sup>[8]</sup>,因为它提供了代码生成功能,整个建模过程可不必编写代码<sup>[9]</sup>。

实际上,MetaEdit+是一种既可以建立元模型又可以建立模型的工具软件。首先,领域专家通过对 MetaEdit+元模型的实例化得到 DSL 元模型,用于刻画 DSL;然后,将建立的元模型用于配置 MetaEdit+,MetaEdit+就成为支持该元模型所刻画的 DSL 的建模工具了。

## 4 基于MetaEdit+的元建模

本文以一个有趣的 LED 千足虫的外观设计和电路设计<sup>[10]</sup>为例,详细介绍如何通过 MetaEdit+ 4.5 进行 DSL 元建模。

### 4.1 MetaEdit+元元模型

MetaEdit+元元模型称为 GOPRR, GOPRR 是 Graph、Object、Property、Role、Relationship 的缩写。图 1 至图 4 给出了 MetaEdit+元元模型。

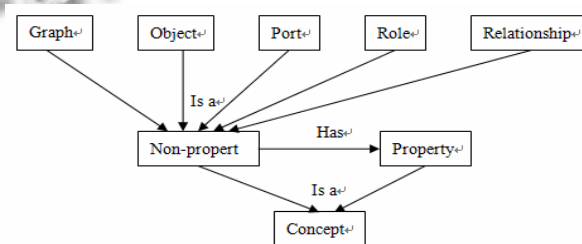


图 1 GOPRR 概念

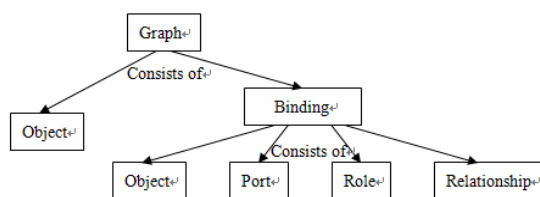


图 2 图的组成

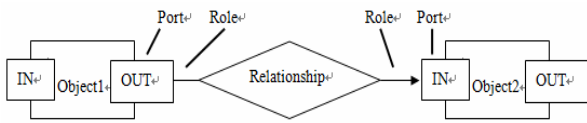


图3 图的绑定 Binding

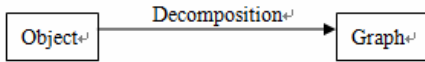


图4 对象引用图

### 4.2 建立 DSL 元模型

通过对 MetaEdit+ 元模型的实例化就可得到 DSL 元模型，具体步骤如下：

#### 1. 设计领域概念

创建领域概念，将其映射到 MetaEdit 元模型中。如电路中使用的电阻器、电容器、电动机、三极管都是领域概念，可映射到元模型中的 Object，作为 Object 的实例。电阻器的型号、电阻值可映射为 Property 的实例，电路中元器件的连接关系可映射为 Relationship 的实例。在 MetaEdit+4.5 中创建 Object 实例的步骤很简单，即用鼠标点击主窗口的 Metamodel|Object Tool 菜单或直接点击工具栏上的 Object Tool 按钮，打开 Object Tool 对话框就可创建 Object 实例了。下面给出千足虫外观设计（俯视图）和电路设计的部分 Object 实例。

#### 1) 外观设计（俯视图）中部分 Object 实例及说明：

Eye: 眼睛

SegmentHorizontal: 一节水平方向上的躯干

SegmentTilted: 一节向上倾斜的躯干

MicroMotor: 微型电动机，连接躯干，控制躯干转弯

PagerMotorUp 和 PagerMotorDown: 寻呼机电机，作为上下足

#### 2) 电路设计的部分 Object 实例及说明：

Sensor: 传感器

Switch: 开关

Schmitt: 斯密特触发器

MotorCircuit: 电动机及驱动电路

Resistor: 电阻器

CapacitorA 和 CapacitorB: 电容器

DividedA 和 DividedB: 分压器

2. 设计表示 Object 实例所用的图形符号。在 MetaEdit+4.5 中使用 Symbol Editor 来编辑图形符号。Eye, MotorCircuit, Schmitt 对应的图形符号如图 5 所示。

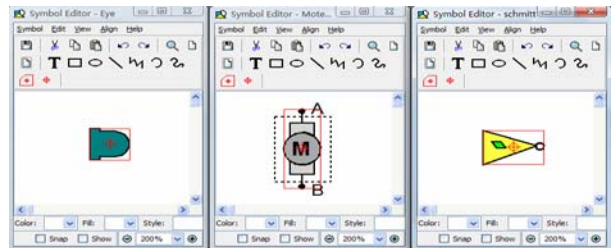


图5 表示元模型对象的图形符号

3. 创建联系实例 Link、角色实例 Line 和图实例 LED Millipede Circuit, 并在 LED Millipede Circuit 中将 Link、Line 与 Object 实例绑定。在 MetaEdit+4.5 中 LED 千足虫电路设计图的绑定设置情况如图 6 所示。

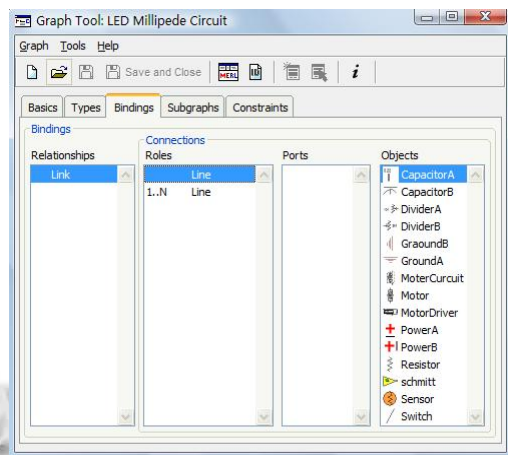


图6 LED 千足虫电路设计图的绑定设置

4. 保存设计结果。在 MetaEdit+4.5 中为保存前几个步骤所做的设计，用鼠标点击 Repository|commit 菜单或直接点击工具栏上的 commit 按钮，DSL 建模工具已经生成，至此 LED 千足虫的元建模工作就结束了。

### 5 利用DSL工具设计模型

为了让读者看一下通过元建模所生成的 DSL 建模工具的设计效果，下面图 7 和图 8 给出了 DSL 建模工具的设计界面，并附有 LED 千足虫外观设计俯视图（见图 7）和部分 LED 千足虫电路设计图（见图 8）。

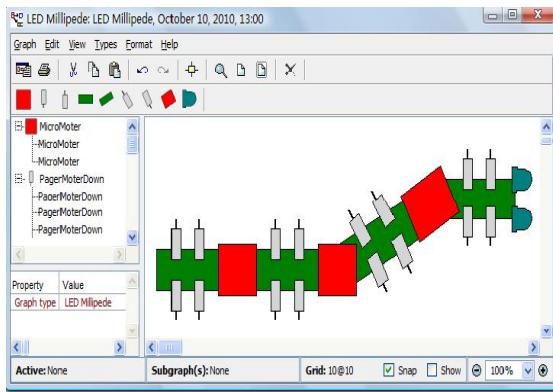


图7 LED千足虫外观设计俯视图

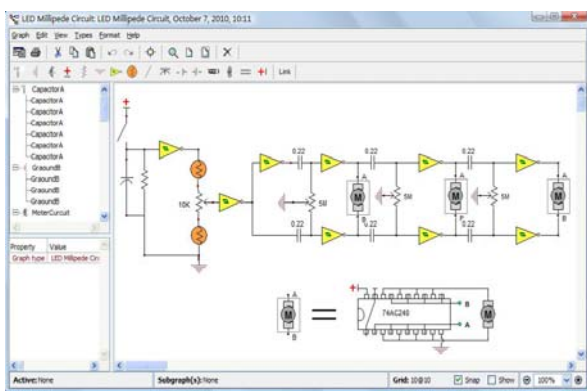


图8 部分LED千足虫电路设计图

## 6 结语

本文以LED千足虫外观设计和LED千足虫电路设计为例,详细介绍了基于元建模工具MetaEdit+的元建模过程。由于LED千足虫外观设计比较简单,所以在其元模型中仅配置了GOPRR的Graph和Object。LED千足虫电路设计元模型虽然配置了GOPRR的Graph、Object、Property、Role和Relationship,但为了简化设计,没有配置GOPRR的Port。

尽管MetaEdit+是一款不错的商品化软件,但在设计的过程中也发现了它的一些不足之处,如没有提供表示Object实例图形符号的旋转功能,同一对象因表示符号不同在MetaEdit+中只能定义为不同的实例,从而加大了设计的工作量。如在LED千足虫外观设计中寻呼机电机被定义为PagerMotorUp、PagerMotor

Down、PagerMotorUpTitled和PagerMotorDownTitled四个实例。再比如Object、Port等实例一旦创建就无法删除,需要在退出MetaEdit+时不保存所做的修改才能删除,这虽然不影响元建模,但也给使用带来很多不便。希望MetaEdit+在未来的新版本中能弥补这些不足,这将大大提高MetaEdit+的元建模效率。

## 参考文献

- 1 颜炯,王戟,陈火旺.基于UML的软件Markov链使用模型构造研究.软件学报,2005,16(8):1386-1394.
- 2 方红萍,陈和平.信息系统UML建模研究.计算机工程与设计,2006,27(19):3613-3615,3655.
- 3 谢正良,赵建华,李宣东,等.一种基于J2EE平台的MDA模型转换技术.计算机应用研究,2005(3):51-54.
- 4 Soley R, Frankel DS, Parodi J. The MDA Journal: Model Driven Architecture Straight from the Masters. New York: Meghan Kiffer Press, 2004:135-146.
- 5 Wikipedia. Meta-Object Facility. [2010-10-5]. [http://en.wikipedia.org/wiki/Meta-Object\\_Facility](http://en.wikipedia.org/wiki/Meta-Object_Facility)
- 6 刘辉,麻志毅,郝维忠.元建模技术研究进展.软件学报,2008,19(6):1317-1327.
- 7 Tolvanen JP, Pohjonen R, Kelly S. Advanced tooling for domain-specific modeling: MetaEdit+. Proc. of the 7th OOPSLA Workshop on Domain-Specific Modeling. University of Jyväskylä, 2007.
- 8 Kelly S. Comparison of Eclipse EMF/GEF and MetaEdit+ for DSM. 19th Annual ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications, Workshop on Best Practices for Model Driven Software Development. New York: ACM, 2004.
- 9 Tolvanen JP. MetaEdit+: MetaEdit+: integrated modeling and metamodeling environment for domain-specific languages. Companion to the 21st ACM SIGPLAN Symposium on Object-oriented Programming Systems, Languages, and Applications. New York: ACM, 2003:690-691.
- 10 Neupane K. The Millipede Project. [2010-10-1]. <http://www.prokn.co.cc/2010/10/millipede-project.html>