

文章编号: 1000-5471(2005)03-0426-04

一种基于 UML 和敏捷建模的 JEMM 方法研究

王毅嘉^{1,2}, 张为群^{1,2}

1. 西南师范大学 计算机与信息科学学院, 重庆 400715; 2. 重庆市智能软件与软件工程重点实验室, 重庆 400715

摘要: 将敏捷建模(agile modeling AM)与 UML 两者结合起来, 对开发中用 UML 进行敏捷建模的方法进行研究, 提出了基于 UML 和敏捷建模的 JEMM(just enough modeling method)方法, 也就是恰好够用的建模方法。

关键词: 统一建模语言; 敏捷建模; 模式驱动开发

中图分类号: TP311.52

文献标识码: A

现代的软件开发采用面向对象的方法进行建模, 常用的工具就是统一建模语言(UML), 它能够对面向对象系统进行可视化、详述、构造和文档化。而敏捷建模是一个基于实践的有效的建模方法, 是一种有效的, 轻量的软件开发方法, 它包括一系列以特定原则和价值为导向的可被软件专业人员应用到日常工作中的实际做法, 可应用于基于软件的系统。

本文将敏捷建模与 UML 两者结合起来, 提出了基于 UML 和敏捷建模的 JEMM(just enough modeling method)方法, 也就是恰好够用的建模方法。

1 基于 UML 的建模

在软件开发中建模其实是对现实的一种简化, 可以帮助我们对整个待开发的系统进行可视化, 并且详细说明系统的结构或行为。而且用 UML 描述的模型可以利用很多现有的工具映射成编程语言代码^[1]。这也对建模提出了更高的要求, 我们尝试在敏捷建模和 UML 的结合方面做一些研究, 用 UML 进行敏捷建模时, 需要遵从下面这些原则^[2]:

(1) 比传统模型够用, 敏捷模型是足够精确的, 我们用数学的方法找到主要的需求, 通过 UML 建立基本的类, 勾勒出待开发软件系统的主要特征。也就是用敏捷的方法抓住系统的主要功能, 建立一个足够用的模型。

(2) 所建的模型要符合创建者的目的, 也就是说我们在建模时要考虑这个模型的可验证性, 确定它符合需求, 从而进一步的进行代码生成。

(3) 适应需求的变化。在编码的过程中, 当需求变化时代码也要跟着变化, 因此我们必须在编码前, 也就是建模时引入敏捷方法^[3], 也就是我们归纳出的 JEMM 建模方法生成的代码具有适应需求变化的特征。

2 JEMM 方法

首先, 找到系统的主要功能, 在需求分析的基础上找到影响系统的主要因素进行建模。步骤如下:

A. 设影响系统功能分析的影响因素为 (X_1, X_2, \dots, X_n) , 整个系统的功能因素 Y , 对每个给定的 X

收稿日期: 2004-12-17

基金项目: 重庆市自然科学基金资助项目(CSTC, 2004BB0146); 重庆市信息产业发展基金资助项目(200401021)。

作者简介: 王毅嘉(1977-), 女, 四川成都人, 硕士研究生, 主要从事软件建模、软件测试的研究。

通讯作者: 张为群, 教授。

求 Y 的数学期望, 记为 $E(Y|X)$, 则对每个 X 有不同的期望值;

B. 设 $y = E(Y|X) = f(x)$, 则称 $f(x)$ 是 Y 对 x 的回归, $y = f(x)$ 是 Y 关于 X 的回归方程. 根据上面的定义, 我们要找到 $f(x)$, 因此我们可以先对 Y 和 $f(x)$ 做一些简化: Y 是一个正态随机变量, 服从正态分布, 因此有方差 $D(Y) = \sigma^2$;

C. 由于 $f(x)$ 与 x 呈线性分布, 设 $Y = a + b * x$, 并定义相关系数 r , 它是反映随机变量 x 与 Y 呈线性关系程度的一个度量指标.

$$r = \frac{\text{cov}(X, Y)}{\sqrt{DX \cdot DY}}$$

r 的取值范围是 $|r| \leq 1$, 当 $|r|$ 接近于 1 时, x 与 Y 呈密切线性相关, 当 $|r|$ 接近于 0 时, x 与 Y 非线性相关;

D. 通过 UML 绘制类图, 列出相应的属性和操作, 再用形式化语言 RSL 对这些类形式化, 在这些形式化的代码中找到类的主要操作. 根据 UML 的特点, 可以对系统建立静态模型和动态模型(图 1);

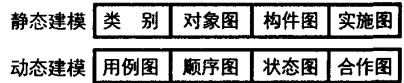


图 1 开发中的图

Fig. 1 Diagram During Development

E. 然后再根据前面找到的类和类的主要操作, 继续用 UML 对主要类敏捷建模, 生成一个恰好够用的类图.

F. 对这些类图编码, 我们按照敏捷建模的思想来设计代码, 通过设计代码实现当需求变化时设计不变的敏捷思想^[4].

3 JEMM 的实例

我们按照上面的步骤, 对文献[5]的一个供工业使用的天气监控系统进行详细分析. 首先, 把需求分析得出的所有的影响因素列出.

A. 设这些气象数据为 X_i , 则求得数学期望 $y = E(Y|X)$;

B. 根据公式计算出方差 $D(X)$, $D(Y)$, 并得到相应的相关系数 r , 可分为下面 3 类:

强相关量: $r \geq 0.7$

中度相关量: 当 $0.4 \leq r < 0.7$ 时, 称 X_i 与 Y_i 中度相关

轻度相关量: 其他量.

C. 因此我们可以找到那些跟 Y 密切线性相关的变量, 就可确定影响天气的主要因素, 并对类形式化. 在这些形式化语言 RSL 中找到主要的属性和操作.

图 2 即为我们找到的主要类的一个, 我们列出了它的属性和操作, 下面我们参照文献[6]对它形式化.

SYSTEM_OF_COORDONATES =

```

Class
  Type
    Position::
      X_coord; Real
      Y_coord; Real
  Value
    origin; Position = mk_Position(0.0, 0.0)
distance: Position >< Position
  ->Real
distance(p1, p2) ==
((x_coord(p2)-x_coord(p1)) ↑ 2.0 +
(y_coord(p2)-y_coord(p1)) ↑ 2.0) ↑ 0.5
end
    
```

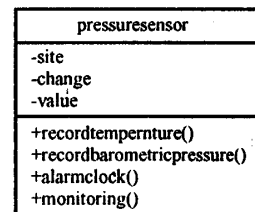


图 2 压力传感器的类图

Fig. 2 Class Diagram of Pressure Sensor

这个例子对软件建模中压力在屏幕上的显示进行了形式化. 我们看到在 RSL 语言中, 对压力的位置进
万方数据

行了定义, 用 x, y 的坐标来表示某个气压在屏幕上的位置, 并给出了位置的计算公式. 因此帮助我们找到了主要的属性, 也即压力大小所在的位置就可计算出其它属性, 帮助实时了解压力情况, 因此是一个主要的操作和属性.

D. 根据上面找到的类, 参照文献[7]的方法绘制一个恰好够用的类图(图 3).

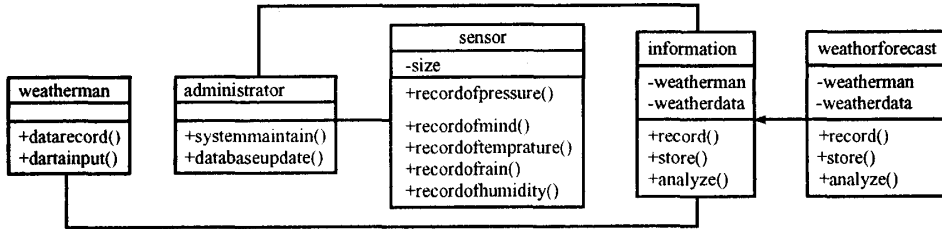


图 3 气象系统的一个足够的类图

Fig. 3 Enough Class Diagram of Weather Report

我们从图中可以看出, 整个系统中有了足够的压力、温度、风向等的传感器, 可以实时显示天气的情况并进行预测. 天气预报员对这些数据的输入和记录进行操作, 我们可以看出对于这个软件系统来说, 这个类图是足够的, 满足我们定义的 JEMM 方法.

E. 根据敏捷建模的另外一个特点: 响应不断变化的需求. 通过设计一些抽象类来避免需求变化引起的设计变化, 我们把它用在记录气象信息的纸带读入信息的小程序编写上, 下面我们根据文献[8]列出敏捷的代码和非敏捷的代码.

i. 非敏捷的代码

```

bool ptFlag = false;
bool punchFlag = false;
//remember to reset these flags
Void Copy( )
{
    Int c;
    While ((c = (ptflag ? Rdpt( ) : RdKbd())) != EOF)
        punchFlag ? WrtPunch(c) : WrtPrt (c); }
    
```

ii. 敏捷的代码

```

Class Reader
{
    Public:
        Virtual int read( ) = 0;
};
Class KeyboardReader : public Reader
{
    Public:
        Virtual int read( ) {return RdKbd( ); }
}
KeyboardReader GdefaultReader;
Void Copy(reader & reader = GdefaultReader)
{
    Int c;
    While ((c = reader. read()) != EOF)
    
```

```
WrtPrt(c); }
```

4 结 论

由此我们可以看出, 通过 UML 进行建模, 简化了分析数学模型的过程, 使整个开发过程可视化了, 并且更直观. 通过敏捷设计的代码也更能适应需求变化的要求, 通过构建抽象类免于在输入设备变化的情况下导致的代码设计变化, 通过采用形式化的方法找到了足够的类的属性和操作. 因此敏捷方法和 UML 的结合是对软件开发的一个新的尝试和较好的实际开发方法, 也即 JEMM 是对软件开发建模和设计一个有益的尝试.

参考文献:

- [1] Grady Booch, James Rumbaugh, Ivar Jacobson. The Unified Modeling Language User Guide [M]. Boston: Addison-Wesley, 2001.
- [2] Scott W. Ambler. Agile Driven Development is good enough [J]. IEEE Software, 2003, 71: 740 - 759.
- [3] Scott W. Ambler. Agile Modeling Effective Practices for Modeling and Documentation [EB/OL]. WWW. agilemodeling.com, April, 2003.
- [4] Robert C. Martin. Agile Software Development: Principles, Patterns, and Practices [M]. Beijing: China Machine Press, 2004.
- [5] 吴 翊, 吴孟达, 成礼智. 数学建模的理论与实践 [M]. 长沙: 国防科技大学出版社, 1999.
- [6] 罗 蜜, 张为群. 结合形式化方法的 UML 系统开发 [J]. 西南师范大学学报(自然科学版), 2003, 28(2): 203 - 208.
- [7] 王瑞金, 段会川, MartinGogolla. 统一建模语言 UML 及其建模实例 [J]. 计算机应用研究, 2002, 08: 80 - 84.
- [8] 沈备军, 陈 诚, 居德华. 敏捷软件过程的研究[J]. 计算机研究与发展, 2002, 39(11): 1456.

A JEMM Approach based on UML and Agile Modeling

WANG Yi-jia^{1,2}, ZHANG Wei-qun^{1,2}

1. School of Computer and Information Science, Southwest China Normal University, Chongqing 400715, China;

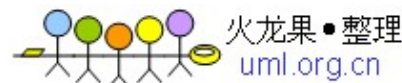
2. Intelligent Software and Software Engineering Research Institution, Chongqing 400715, China

Abstract: In software development today, it is usually found that inconsistency between requirement and functionality, so setting up an elegant model is very urgent for meeting the needs of users. But if to spend too much time on modeling, there must be less time on the rest work, so agile modeling appears for setting up an enough and functional model. Undoubtedly, UML is the best tool among all the tools of modeling. The authors combine these two together, and wish to find a good metric for software developing with UML and AM (agile modeling).

Key words: unified modeling language; agile modeling; model-driven development

责任编辑 章吉康

一种基于UML和敏捷建模的JEMM方法研究



作者: 王毅嘉, 张为群, WANG Yi-jia, ZHANG Wei-Qun

作者单位: 西南师范大学, 计算机与信息科学学院, 重庆, 400715; 重庆市智能软件与软件工程重点实验室, 重庆, 400715

刊名: 西南师范大学学报(自然科学版) **ISTIC PKU**

英文刊名: JOURNAL OF SOUTHWEST CHINA NORMAL UNIVERSITY(NATURAL SCIENCE EDITION)

年, 卷(期): 2005, 30(3)

被引用次数: 5次

参考文献(8条)

1. GRADY Booch;James Rumbaugh;Ivar Jacobson The Unified Modeling Language User Guide 2001
2. 沈备军;陈诚;居德华 敏捷软件过程的研究[期刊论文]-计算机研究与发展 2002(11)
3. 王瑞金;段会川 MartinGogolla, 统一建模语言UML及其建模实例[期刊论文]-计算机应用研究 2002(8)
4. 罗蜜;张为群 结合形式化方法的UML系统开发[期刊论文]-西南师范大学学报(自然科学版) 2003(02)
5. 吴翊;吴孟达;成礼智 数学建模的理论与实践 1999
6. Robert C Martin, Agile Software Development:Principles, Patterns, and Practices 2004
7. Scott W Ambler, Agile Modeling Effective Practices for Modeling and Documentation 2003
8. Scott W Ambler, Agile Driven Development is good enough 2003

引证文献(5条)

1. 张逸 在Scrum中实施敏捷建模[期刊论文]-重庆文理学院学报(自然科学版) 2009(5)
2. 李旭芳 基于迭代开发和UML的AMDD方法研究[期刊论文]-上海工程技术大学学报 2009(3)
3. 李旭芳 敏捷模型驱动开发中UML的应用[期刊论文]-计算机与现代化 2009(12)
4. 曹弘毅;杨君慧 基于Agile方法的软件设计实现[期刊论文]-网络安全技术与应用 2009(8)
5. JIANG Yu 基于UML的敏捷建模方法研究[期刊论文]-计算机工程与设计 2008(15)

本文链接: http://d.g.wanfangdata.com.cn/Periodical_xnsfdxxb200503014.aspx