

基于 UML 的软件建模工具的研制

麻志毅, 蒋严冰, 李劲宇, 戴耀飞

(北京大学计算机科学与技术系, 北京 100871)

摘 要: 本文讨论了基于 UML 的软件建模工具的设计和实现技术, 介绍了面向对象的软件建模工具所应支持的概念与模型, 对工具的系统功能结构与设计进行了详细地阐述, 并给出了一套具有通用性的具体实现方法。

关键词: 面向对象; 建模工具; 模型; UML

中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2002) 12A-2049-03

Development of Software Modeling Tools Based on UML

MA Zhi-yi, JANG Yan-bing, LI Jin-yu, DAI Yao-fei

(Department of Computer Science and Technology, Peking University, Beijing 100871, China)

Abstract: Development of object-oriented software modeling tools based on UML is discussed in this paper. The paper introduces concepts and models which object-oriented software modeling tools should support, expatiates the tools' system function structure and designs, and presents a universality method for implementing object-oriented software modeling tools based on UML.

Key words: object-oriented; modeling tool; model; UML

1 引言

正是由于 UML(Unified Modeling Language)^[1]和 OML(OPEN Modeling language)^[2]等标准的推出, 结束了面向对象技术领域中的“方法学大战”。这样也使得面向对象的软件建模工具有据可依, 不再是象以前那样, 由于所采用的概念与表示法互不相同, 导致了所建立的模型在交流上的困难, 也就更难谈得上工具间的信息普遍互换。

UML 不但在学术界, 在软件产业界也得到了广泛地支持。国内外据此研制出了一些基于 UML 的软件建模工具, 并发挥了较好的作用。由于 UML 本身还存在有一定的问题^[3], 而且面向对象的软件建模工具的研制也较为复杂, 这些工具都各具千秋, 也都有一定的缺陷^[4]。结合我们多年来研制面向对象的软件建模工具的基础, 加之在制定国家面向对象软件系统建模规范的过程中的所得, 并吸取其它软件建模工具的成功经验, 现提出一套设计与实现基于 UML 的软件建模工具的技术, 据此我们实现了基于 UML 的 JBOO 版本(JBOO3.0)。

本文首先介绍面向对象的软件建模工具所应支持的概念与模型, 然后详细地阐述这样的工具的系统功能结构与设计, 最后给出了一套具有通用性的具体实现方法。

2 面向对象的软件建模工具支持的概念与模型

根据目前的发展来看, UML 为软件产业界普遍认同。面向对象的软件建模工具应该对其予以支持。特别是国家面向

对象的软件建模规范对 UML 中的概念进行了划分, 面向对象的软件建模工具至少应该支持其中的核心概念。此类工具所采用的概念的表示法也都要遵循上述标准。

基于 UML 的软件建模工具应从五个角度、两个方面对系统建模, 所产生的模型的集合形成了系统模型。这五个角度为用况视图、进程视图、部署视图、设计视图和实现视图, 两个方面是指系统的静态结构方面和动态行为方面。每种视图由作用不同的图构成, 请见表 1。

表 1 工具支持的视图及其构成

| | 设计视图 | 进程视图 | 实现视图 | 部署视图 | 用况视图 |
|------|-------------|-------------|-------------|------|------|
| 静态方面 | 类图、对象图 | 同左 | 构件图 | 部署图 | 用况图 |
| 动态方面 | 交互图、状态图、活动图 | 同左, 注重进程、线程 | 交互图、状态图、活动图 | 同左 | 同左 |

上表中的每张图都描述了系统的一个局部模型。

3 系统的功能结构与设计

面向对象的软件建模工具是一种对软件系统的模型进行可视化、详述、构造和文档化的工具, 它是由一系列的子工具组成的。对需求的描述和对系统的分析与设计都要有相应的子工具支持, 其余的子工具为辅助工具, 对前述的子工具提供支持。这些辅助工具实现存储、导航、协作建模、文档生成、逆向、语法与语义检查、代码生成和模型管理等功能, 请见图 1。

用需求描述工具建立用况图, 用相应的静态结构建模工具建立类图、对象图、构件图和部署图, 用相应的动态行为建

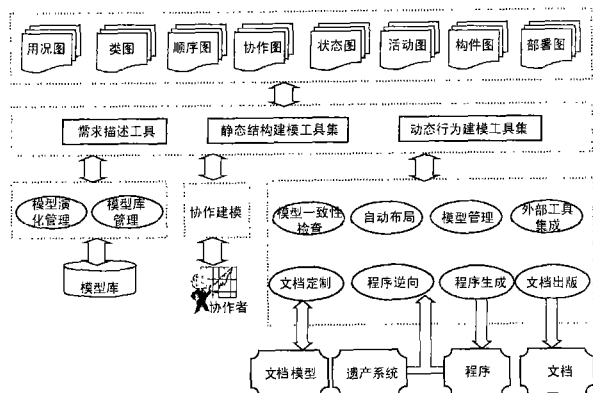


图1 系统功能结构图

模工具建立顺序图、协作图、状态图和活动图。

协作建模部分支持多方协作建模。用于协作建模的工具可能来自不同的厂家,这样首先要解决在不同种类的建模工具之间的数据交换问题。OMG(Object Management Group)根据UML元模型和XMI(XML Metadata Interchange)规范^[5],给出了UML的XMI DTD标准,工具据此存储模型,用以进行工具间的数据交换。具体的方法为,用XML DTD描述UML元模型,用XML文档描述UML模型数据。这样,用一种工具建立的模型可转换为相应的遵循UML和XMI的工具建立的模型,反之亦然,图2说明了JBOO模型与其他遵循UML和XMI规范工具所建立的模型之间的转换。

由于协作建模的各个结果都是按XMI规范存储的,对它们进行合并,利用的是XMI中的增量技术。增量文件包含了当前的设计结果相对基准模型的变化信息,这些信息可以利用XMI规范中提供的XMI.add、XMI.replace和XMI.delete来准确地表达。

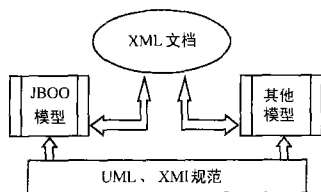


图2 JBOO模型与其他遵循UML和XMI规范工具所建立的模型之间的转换

为了复用,引入了模型演化管理和模型库管理。

(1)模型演化管理 若要跟踪模型的变化情况,可设立此功能。即为模型设立若干演化点,作为里程碑。为了反映模型间存在的差异,在模型间可进行比较。

(2)模型库管理 为了复用,可把模型入库,或从模型库中把模型提取出来。关于模型库及其管理,请看图3。结合图1可以看出,通过协同建模部分导入的模型,可以加入到模型库

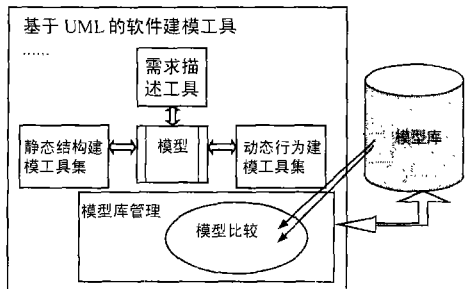


图3 模型库及其管理

中;在模型库的模型,也可以通过协同建模部分向外部导出。

模型的正确性和一致性检查工具对所建立的模型可进行实时的检查,也可进行全局性的检查,对全局性的检查要生成错误状况报告。在检查时,不但要检查同一图中的元素的正确性和一致性,还要检查相同种类的图中的相关元素间的一致性,及不同种类的图中的相关元素间的一致性。以UML元模型为基础,构造语法检查表,检查用户绘制的模型图中的建模元素之间的搭配的正确性;定义一定的语义检查规则,检查用户定义的衍型和约束与元模型的一致性。

为了使所建立的模型中的元素的分布格局更加合理,自动布局工具按照一定算法,经过计算,可对模型图进行自动布局。特别是对逆向过来的模型来说,由于其上没有图形元素的位置信息,该功能尤为重要;否则各种图形元素叠加在一起,很难区分。

模型管理器组织软件建模过程中所产生的各种模型,保证项目以一种有意义的方式进行组织。具体地讲,应该对从用况视图、进程视图、部署视图、设计视图和实现视图方面产生的系统模型进行组织,对于从每个视图产生的模型都放在一个包中。进一步地,用户可以在每个视图包中,定义自己的包,形成包的树状结构。每个包中即可以有图,也可以有包。

程序逆向工具从已存在的软件代码中,逆向产生UML模型。这样的软件代码可以是遗产系统的程序源代码,也可以是从当前模型生成的程序

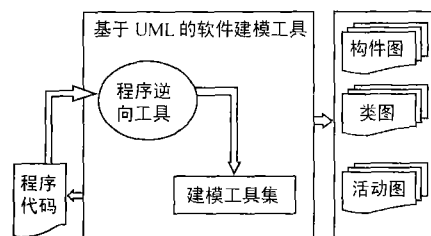


图4 程序逆向工具的作用

源代码。特别是利用逆向工具可保持代码与模型的一致性。图4说明了程序逆向工具的作用。

代码生成工具按选定的语言模板,可生成相应种类语言程序的部分源代码。此外,工具还应该提供语言模板描述语言,用户可用其定义自己的语言模板,请看图5。

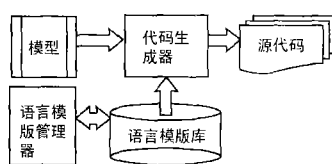


图5 代码生成

使用文档出版工具,按选定的模板,可生成相应种类(如类图和顺序图等)和格式(如HTML或文本)的文档。工具还应该提供文档模板描述语言,用户可用其定义自己的文档模板,请看图6。对于所建模型,若图形较大,可按多个页面输出,再进行拼接,形成文档;也可以按比例进行缩放,以满足在规定的页数内,进行输出。

外部工具集成用于把其它的外部工具集成到建模工具中,特别是与象Microsoft IE这样的internet浏览器集成,可提取构件库中的构件^[6],用于建模。

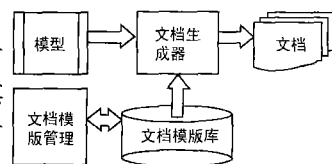


图6 文档出版

4 系统的实现

系统的实现分为工具环境层、图形层、模型层、通用图形编辑框架层、UML 元模型层和基本的设计模式及公共类层这几部分,其间的关系请看图 7. 为了便于对系统的维护和扩展,这几部分之间是松耦合的.

图中的虚线箭头表示的是依赖关系. 实现建模工具时,要用到很多设计模式和公共类,把它们组织在一起,形成基本的设计模式及公共类层,作为实现建模工具的基础.

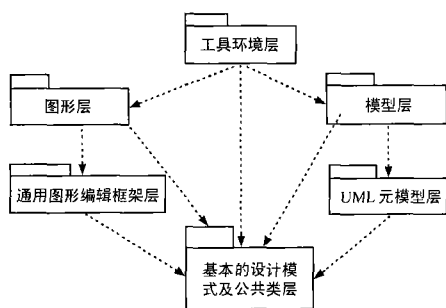


图 7 建模工具的系统实现框架

UML 元模型层的实现以 OMG 发布的 UML 中的物理元模型为基础,可能对其做很小的调整. 例如,在其中略去工具所不支持的概念部分,并加入了支持撤消/重做部分. 该层处理 UML 模型,包括创建 UML 模型元素,以及对 UML 模型元素进行添加、修改和删除等操作,并提供操作 UML 模型的所有接口. UML 元模型层是对模型层中的概念的抽象,不涉及模型的信息.

从工具所支持的各种模型图中,抽取出共性,得到通用图形编辑框架. 绘图框架实现了与建模语义无关的绘图操作等功能,具有可扩充型和通用性. 在图形层中绘制各种图元素时要使用它.

图形层要完成对所有模型图中的各建模元素的绘制. 其实现是通过扩展通用图形编辑框架进行的,即实现通用图形编辑框架中的抽象类,对虚函数进行重载.

模型层是对 UML 元模型层的包装,其中的大部分类继承了 UML 元模型层中的叶子节点类. 与图形层相配合,实现了模型数据和表示的分离.

工具环境层围绕着图形层和模型层,完成对模型和图形的整体操作. 该部分包含了界面上的菜单、图表等功能,实现了代码生成、文档生成、自动布局、模型检查、项目和存储等功能. 图 8 描述了工具环境层的实现框架.

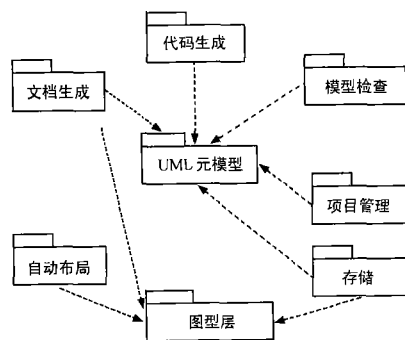


图 8 工具环境层的实现框架

图中的虚线箭头表示的是依赖关系.

图中的存储包括两类存储模型的数据文件:模型语义数据文件保存当前模型的语义信息,模型表示数据文件保存图形信息. 对这两种数据的存储在逻辑上是分开的.

5 结束语

我们结合原有的工作基础,基于 UML1.4 和 XMI 1.2,在本文中提出了一套研制面向对象的软件建模工具的方法.

据此,我们开发出了青鸟面向对象软件建模工具 JBOO3.0. 与同类工具相比(包括 JBOO 以前的版本),利用这套思路开发的 JBOO3.0,概念体系更加清晰、整洁和完整,而且还具有文档模板定制和编程语言定制等功能,并且在支持复用与协作开发方面也具有一定的特色.

从开发工具的角度来讲,按此方法建造的工具,在设计与实现上,结构更为合理,易于维护和扩展;对后续工具(适应 UML 标准的变化)或相关工具的开发者来讲,也易于复用已有的设计方案和代码.

参考文献:

- [1] Object Manage Group. UML documentation version 1.4 [DB/OL]. <http://www.rational/uml/index.jsp>.
- [2] UK: Addison-Wesley, 1997. The OPEN Process Specification [S].
- [3] Cris Kobryn. The Road to UML 2.0: Fast track or detour [J]. Software Development, April 2001.
- [4] 麻志毅. 面向对象的软件建模工具 [A]. 2001 全国软件技术研讨会论文集 [C]. 大连: 大连出版社, 2001. 8.
- [5] Object Manage Group. XMI documentation version 1.2 [DB/OL]. <http://www.omg.org>.
- [6] 张文娟. 支持变换性的软件构件模型及其相关技术研究 [D]. 北京: 北京大学研究生院, 2001.

作者简介:



麻志毅 1963 年生于内蒙古自治区赤峰市, 博士, 北京大学计算机系副教授, 主要研究方向为软件工程与软件工程环境、面向对象技术等, 发表学术论文 40 余篇.



蒋严冰 1975 年生于山东省泰安市, 北京大学计算机系博士研究生, 研究方向为面向对象建模语言以及形式化方法.