

UML 及面向对象的分析与设计的研究*

屈喜龙

(西南交通大学 CAD 工程中心, 四川 成都 610031)

摘要: 详细阐述了 UML 本身的特点和一些容易混淆的概念。对支持 UML 建模的各种工具和 UML 中的各种框图, 也进行了详细的介绍。根据成、德、绵区域网络化制造系统的分析和设计经验, 清晰地说明如何用 UML 来完成一个系统分析和设计, 深入地研究了建模过程所需要用到的框图和各框图之间的关系, 对面向对象的分析和设计有很强的指导性。

关键词: UML; 面向对象; 分析与设计

中图法分类号: TP391

文献标识码: A

文章编号: 1001-3695(2005)09-0074-03

Research for UML and Object-Oriented Analysis and Design

QU Xi-long

(CAD Engineering Center, Southwest Jiaotong University, Chengdu Sichuan 610031, China)

Abstract: The traits and some confused notion of UML is expatiated, All of the tools and frames for UML modeling are introduced. Finally, how to complete the analysis and design for a system is presented clearly based on the analysis and design experience of Chengdu-Deyang-Mianyang Region's networked manufacturing system, and all of the frames and the relation between the frames used in process of modeling are studied deeply, which has great guidance for Object-Oriented analysis and design.

Key words: UML; Object-Oriented; Analysis and Design

自从 1968 年北大西洋公约组织的计算机科学家在联邦德国召开的国际会议上首次提出“软件工程”这个词以来, 软件工程的技术方法、工具、管理等各方面都得到了发展^[1]。而面向对象方法学, 对软件工程的发展无疑是最大的功臣。面向对象方法学提出后, 出现了各种各样的建模的符号体系, 这样, 极大地妨碍了软件设计人员、开发人员和用户之间的彼此交流。因此有必要在分析、比较不同的建模语言以及总结面向对象技术应用实践的基础上, 博采众长, 建立一个标准的、统一的建模语言。UML 的诞生, 结束了符号方面的“方法大战”。然而, UML 本身, UML 中的各种框图以及支持 UML 的各种建模工具让人眼花缭乱, 很容易混扰。本文就 UML 本身、UML 中的各种图, 及各种建模工具进行详细的分析。

1 UML(Unified Modeling Language)

UML 这个概念的本身就足以让人困扰。UML 是一种标准的图形化建模语言, 它是面向对象分析与设计的一种标准表示。UML 不是一种可视化的程序设计语言, 而是一种可视化的建模语言; UML 不是工具也不是知识库的规格说明, 而是一种语言规格说明, 是一种表示的标准; UML 不是过程和方法, 但是允许任何一种过程和方法使用它。

作为一种建模语言, UML 定义了建立面向对象系统模型所需要的概念并给出其可视化的表示法, 但是它并不介绍如何进行系统建模。所以它只是一种面向对象建模语言, 而不是一

种面向对象建模方法。UML 用于建立系统的分析模型和设计模型, 而不是用于编程, 所以它是一种建模语言而不是一种编程语言。用 UML 建立的系统模型不是可执行的程序, 但是可以通过适当的软件工具, 把模型的一部分语义转换为可执行程序是可行的。还有, UML 不是一种形式化语言。它只是部分地采用了形式化语言的定义方式, 但并不是严格的形式化。因此, 用它描述的系统模型不能由计算机作编译执行或解释执行之类的自动处理。UML 不仅包括可供用户使用, 直接用于应用系统建模的概念和可视化表示法, 还包括一些定义这种语言本身所需要的概念。UML 是一个可扩展的建模语言。除了由它定义的建模元素以外, 它还提供了一些扩展机制。从 UML 的早期版本开始, 便受到了计算机产业界的重视, OMG 的采纳和大公司的支持把它推上了实际上的工业标准的地位, 使它拥有越来越多的用户。它被广泛地用于应用领域和多种类型的系统建模, 如管理信息系统、通信与控制系统、嵌入式实时系统、分布式系统、系统软件等。近几年还被运用于软件再工程、质量管理、过程管理、配置管理等方面。而且它的应用不仅仅限于计算机软件, 还可用于非软件系统, 如硬件设计、业务处理流程、企业或事业单位的结构与行为建模。

2 UML 建模工具

UML 定义良好、易于表达、功能强大, 不仅支持面向对象的分析与设计, 而且支持从需求分析开始的软件开发全过程。但如何使这种可视化图形建模技术发挥最大的作用, 其辅助工具也是很重要的。自从 UML 产生以后, 建模工具也层出不穷, 如 Rational Rose, Prosa, Visio, Together, Visual UML, Object Domain, Magic Draw UML 等。下面详细分析和介绍几种最常用

的建模工具:

(1) Rational Rose。虽然开发这个工具的 Rational 公司已经被 IBM 收购了,但是,不可否认,Rational Rose 曾经是 UML 建模工具中最常用的一种。它给软件开发者提供了一整套的可视化建模工具,包括系统建模、模型集成、软件系统测试、软件文档的生成、从模型生成代码的前向工程、从代码生成模型的逆向工程、软件开发的项目管理、团队开发管理等。

(2) Rational XDE。它是在 Rational 公司被 IBM 收购以后才开发出来的。XDE 可以让用户随时看到设计和编码,而无须在设计和编码之间来回切换。它把设计和开发环境组合起来,形成一个综合性工具。Rose XDE Developer 包括了 Eclipse IDE,既可以在微软 Visual Studio .NET 中运行,也可以在 IBM WebSphere Studio 中运行。

(3) Visio。它是微软的产品,而且价格比 Rational Rose 要便宜得多,它加入了包含组件管理的对象管理器,在一个图中创建的对象通过拖放操作可以轻松地移植到另一个图中,而且允许用户用图表示任何事情,从架构布局到办公计划、指向路标和工程计划等。它能在市场占领一席之地的另一个关键原因是因为它是 Microsoft Office 的一个组件,尽管它并不经常与 Microsoft Office 搭配。但是它的界面、控件和功能与 Word, Excel 中的标准一样。

(4) Together^[2]。Borland 公司的 Together 的 .NET 版本中把 UML 建模设计和 Microsoft .NET 框架下的代码结合在一起,降低 UML 的复杂程度。Together for Visual Studio 版本是第一个 .NET 技术方面的高级建模环境,它最大的特点就是价格低,Together 已经支持 Eclipse (Java),JBuidler 和 CBuilder,现在,又本地化地支持了 Visual Studio .NET。该工具使用的是 UML 1.3和 1.4,随着 UML 2.0 的即将批准,Borland 承诺将在 2004 年底之前支持 UML 2.0。Together Edition for Microsoft Visual Studio .NET 2.0 要求首先安装 Visual Studio .NET 2003,并要求操作系统为 Windows 2000 (SP2 或更高)、XP Professional 或 Windows Server 2003。

上面四种工具各有优缺点,它们也各自在市场上占有一席之地。用户根据自己的需求来选择满足自己要求的工具。

3 UML 中的各种框图

UML 中的各种图让人眼花缭乱。文献[3]指出,开发一个系统模型需要九种图:类图、对象图、用况图、顺序图、协作图、状态图、活动图、构件图和部署图。其中类图和对象图一同称作静态结构图,顺序图和协作图被称作两种不同的交互图,而构件图和部署图则同称作实现图。文献[4]指出,UML 中的图分为五类:用例图、静态图、行为图、交互图、实现图。静态图包括类图和对象图,行为图包括状态图和活动图,交互图包括时序图和协作图。实现图包括组件图和配置图。在文献[5]中,把图分两类:结构图和行为图。其中,又把结构图分为四类:类图、对象图、组件图和部署图。而行为图又分为:用例图、活动图、序列图、协作图和状态图。文献[6]指出 UML 框图分为八类:用例图、活动图、协作图、顺序图、类图、层次图、组件图和部署图。下面结合成、德、绵区域网络化制造系统的实践经验,来说明采用 UML 如何完成一个系统的分析和设计,这个过程需要哪些框图,以及这些框图之间的关系。

4 面向对象的分析与设计所需的各种框图及其关系

4.1 需求(用例图)分析阶段

长期以来,在面向对象开发和传统的软件开发中,是根据典型的应用场景来了解需求的,但是,这种场景是非正式的,虽然经常使用,但是难以为之建立正式的文档。用例方法引起了面向对象领域的极大关注,并认为它是第二代面向对象技术的标志。用例模型的建立是系统开发者和用户反复讨论的结果。表明了开发者和用户对需求规格定义达成的共识。首先,它描述了待开发系统的功能需求;其次,它将系统看成“黑盒子”,从外部参与者的角度来理解系统;另外,它驱动了需求分析之后各阶段的开发工作,不仅在开发过程中保证了系统所有功能的实现,而且被用于验证和检测所开发的系统是否满足系统需求,从而影响到开发工作中的各个阶段和 UML 的各个模型。在 UML 中,采用用例图来表示需求分析是很合适的。所以,用例图是使用 UML 语言来设计一个系统的起点,是有关系统细节的最高形式。它是一种在系统完成后,能使管理机构、用户和其他非开发人员了解其功能的极好方法。用例并不详细介绍系统如何运作。

4.2 分析阶段

系统开发的分析阶段强调对需求、概念和操作的理解。两种框图足以胜任各个分析阶段的所有任务。

(1) 概念模型。关键问题也许是很复杂的,分而治之是处理这种复杂问题的一个通用策略,所谓面向对象这个概念的精髓之处在于能将问题分解成多个单独的概念或者对象,即分解成我们能够觉察到的事物。在面向对象的分析中,分解的尺度以概念为基础。而概念模型是问题域中概念的描述。识别出一组概念或对象是面向对象分析的核心,是为以后的设计和实现阶段获得更好的结果而必要的一项活动。在 UML 中,一个概念模型用一组静态结构图来表达,在静态结构图中,不需要定义操作和方法。使用概念模型的好处是它强调了领域中的概念,而不是软件实体。概念模型可以展示出一组概念、概念之间的关联以及概念的属性。概念模型除了能够将问题空间分为可以理解的小单元(概念)以外,建立一个概念模型还能够辅助我们澄清领域中的术语或词汇。概念模型可以被视为一个传达信息的载体(向感兴趣的小组(如开发人员)传达信息的载体),它说明了有哪些重要的术语,以及这些术语之间的关系。

(2) 顺序图。用况图向我们提示出参与者和我们最终要构造的软件系统之间是如何进行交互的。在交互过程中,一个参与者向一个系统发出一系列事件,并请求系统作出响应操作。一个好的做法是将参与者请求系统要执行的操作进行隔离考察。在进行逻辑设计之前,软件应用系统将如何工作,必须对系统行为进行调查,并且要将系统行为定义为一个“黑盒子”。系统行为描述了系统做什么,而不解释系统怎么做,系统顺序图是对系统行为所做的描述的一部分。所以建立系统的顺序图通常被认为是系统调查过程中的一部分,这个调查过程主要是查明要建立的系统是一个什么样的系统。因此,系统顺序图也是分析模型的不可缺少部分。UML 包含了系统顺序图的表示法,用来展示参与者向系统发起的事件。顺序图的建立要依赖已经建立的用况图。它展示了用况的特定事件发生过程中与系统直接发生交互的外部参与者、系统以及参与者所

发起的系统事件等。在系统图中,从上到下代表时间的流逝,事件的顺序要符合用况中所描述的事件的发生顺序。

4.3 设计阶段

分析阶段完成以后,就可以进入设计阶段。这个阶段将开发一个基于面向对象的系统逻辑解决方案。这个解决方案的核心是要建立交互图,用它来展示为了满足系统需求,各个对象之间是如何进行相互通信的,而通信的手段是消息传递。在交互图建立之后,接着要建立的是设计类图,它对要实现的软件类和接口的定义进行总结。

(1) 协作图。交互图能够展示出类模型中实例之间的消息交互。UML 中定义了两种交互图,即协作图和顺序图。两者都描述完成一个协作的一组对象之间所发生的交互情况,都是基于相同的基本信息,但强调不同的方面。顺序图展示按时间顺序排列的交互,它所展示的交互,是在一个协作中对象之间为产生所要求的操作或结果而交换的一组消息。顺序图不能展示对象之间的关系。协作图表现的就是参与交互的对象之间的关系,而不是表示时间顺序。但是,为什么在设计阶段只采用协作图呢?因为协作图有异常出色的表达能力,能够表达出更多的相关背景信息,如对象间的可见性的种类。协作图还可轻松地表达条件逻辑和并发,而且协作图相对顺序图来说,空间要节省得多。协作图的建立主要是依赖于分析阶段建立的概念模型,根据概念模型,设计者可以有选择地定义模型中的概念所对应的类,协作图就是展示这些类对象参与交互的过程。

(2) 类图。它说明了软件类的规格说明和应用程序接口。类图所能表达的典型信息包括:类、关联和属性、接口及其操作和常量、方法、属性类型信息、导航还有类或接口之间的依赖关系。与分析阶段中的概念模型不同的是,一个设计类图显示出了软件实体的定义而不是真实世界中的概念。建立类图要依赖定义好的协作图和概念模型,根据协作图,设计者能够识别

出参与解决方案的软件类和类中的方法,根据概念模型,设计者能够在定义的类中添加细节^[7]。

5 结束语

UML 在取得巨大成功的同时,还存在许多问题,用户和教师抱怨它内容庞大、难学难教而且太过复杂;学者认为它缺少一个精练的核心和定义良好的外围,有些语义定义得不够精确而且带有二义性;建模实践者认为它缺少支持自己领域建模要求的机制;工具开发商则因为规范本身的不确定性而产生理解上的偏差,它们对 UML 的自行诠释有可能误导用户。UML 的关键问题是过于庞大和复杂,以及在语言体系结构、语义等方面存在理论缺陷。本文对 UML 本身及与它相关的一些概念进行了详细的分析和比较,说明了如何使用 UML 建模以完成一个系统的分析和设计,对 UML 中的各种框图进行深入的研究,对面向对象的分析与设计有很大的指导意义。

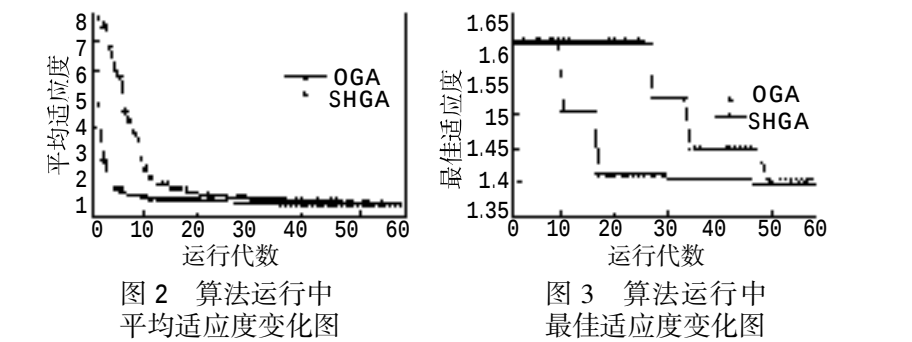
参考文献:

[1] 张海潘. 软件工程导论[M]. 北京:清华大学出版社,1998.1-5.
[2] Alun Williams[EB/OL]. http: //www. umlchina. com/News/Content/143. htm, 2004-05-26.
[3] 邵维忠,杨芙清. 面向对象的系统设计[M]. 北京:清华大学出版社,2003.160-174.
[4] 冀振燕. UML 系统分析设计与应用案例[M]. 北京:人民邮电出版社,2003.3-9.
[5] Jason T Roff. UML a Beginner's Guide[M]. 张瑜. 北京:清华大学出版社,2003.9-13.
[6] Wendy Boggs, Michael Boggs. Mastering Rational XDE[M]. 邱仲潘. 北京:电子工业出版社,2003.1-7.
[7] 拉尔曼. UML 和模式应用:面向对象分析与设计[M]. 姚淑珍. 北京:机械工业出版社,2002.4-95.

作者简介:

屈喜龙(1978-),男,湖南新邵人,博士研究生,研究方向为网络化制造的研究。

(上接第 73 页)



从运行结果上看(表 1),本文的 SHGA 算法不仅最优值 $f(x) = 1.397\ 121$ 优于文献[4,5]中的优化结果,而且本文的最优解严格满足约束条件 $C_1(x) = 0$,而文献[4,5]中却均未满足。这说明本文的基于模拟退火的混合遗传算法,在求解效果上具有一定的优势,能显著地提高 GA 的优化性能。同时本文还对其常用的带复杂约束的测试函数进行了验证,均取得较好效果。

表 1 算例运行结果及比较

方法	$f(x)$	x_1	x_2	$C_1(x)$	$C_2(x)$	出处
SHGA	1.397121	0.821380	0.910690	0.000000	0.339310	本文
OGA	1.401377	0.820234	0.902380	0.015474	0.353906	本文
混沌优化	1.398697	0.820742	0.910695	0.000109	0.002918	文献[4]
GA	1.4339	0.8080	0.88544	0.037	0.052	文献[5]

3 结束语

本文对遗传算法的改进方法进行了研究,将模拟退火的良

好局部搜索能力与遗传算法擅长的全局搜索性相互取长补短,提出将一种有效的模拟退火算法作为遗传算法的一个独立操作算子,置于遗传算法进化过程中,融合为模拟退火混合遗传算法(SHGA),并提出了动态交叉率和变异率的思想。设计了不带模拟退火算子的 OSA 算法,以便验证模拟退火在 SHGA 算法中的作用。数值优化结果表明,与其他优化算法相比,SHGA 算法可以显著提高遗传算法的优化性能和运行效率,对解决非线性约束优化问题有较大的应用价值。

参考文献:

[1] 周丽,孙树栋. 遗传算法原理及应用[M]. 北京:国防工业出版社,2001.
[2] 欧阳森,宋政湘,王建华,等. 一种快速收敛的遗传算法[J]. 计算机应用研究,2003,20(9):50-52.
[3] 杨若黎,顾基发. 一种高效的模拟退火全局优化算法[J]. 系统工程理论与实践,1997,17(5):29-35.
[4] 骆晨钟,邵惠鹤. 用混沌搜索求解非线性约束优化问题[J]. 系统工程理论与实践,2000,20(8):54-58.
[5] Homaifar A, et al. Constrained Optimization via Genetic Algorithms [J]. Simulation, 1994, 62(4):242-254.

作者简介:

周丽(1977-),女,湖北武汉人,讲师,硕士,主要从事系统优化方法与预测决策技术研究;黄素珍(1973-),女,浙江温州人,讲师,硕士,主要从事计算机算法研究。