

# 本体在 UML 建模中的应用研究

王 谦 赵少锋 臧凤奎 李 明

(兰州理工大学计算机与通信学院, 兰州 730050)

**摘 要** 本体作为一种能够有效表达概念层次结构和语义的模型,被广泛地应用到计算机科学的众多领域。UML作为一种通用的建模语言,已成为面向对象建模领域公认的工业标准。在介绍本体和统一建模语言基本概念的基础上,比较分析了面向对象建模与本体建模的异同,提出用本体技术来弥补 UML 规范中存在的缺陷,使得 UML 能够更加规范地表达建模信息。

**关键词** 本体 统一建模语言 面向对象 描述逻辑  
中图法分类号 TP311.11; 文献标志码 A

UML是在多种面向对象建模方法联合的基础上形成的建模语言,于1997年11月被对象管理组织(OMG)采纳为建模语言规范。它的出现在面向对象领域得到广泛的关注与应用。UML可以对任何具有静态结构和动态行为的系统进行建模,它适用于以面向对象技术来描述任何类型的系统,而且适用于系统开发的不同阶段。UML在取得显著成绩的同时,也存在着许多缺点。正如Cris Kobryn在文献[1]中指出的:“UML 1.x系列修订版从来没有离开过批评。”比如,与四层元模型体系结构有关的问题、形式化方面的问题、类图和对象图并存问题、协作图问题,还有无道理的复杂性等。本体是描述概念及概念之间关系的概念模型,通过概念之间的关系来描述概念的语义。本体作为一种有效的知识表示方法,可用于不同领域内的人之间的交流和知识共享。UML更侧重于在建模过程中开发人员与用户之间的交流,本体更侧重于领域知识的表示和共享,而UML在表达方面还存在不足之处,以至于开发人员与用户之间交流时可能会产生歧义<sup>[1]</sup>,如果把本体用于UML建模,则能弥补这种不足,从而促进建模的顺利进行,提高开发效率。

许多学者作了大量研究工作,比如,钟凌燕

等<sup>[2]</sup>在利用IFVO进行企业建模的过程中,先利用UML建立本体论模型,再将其映射成IFVO的本体论标记语言OML。王翀等<sup>[3]</sup>采用UML的Profile扩展技术作为消除OWL复杂符号体系与本体建模易操作性需求之间的矛盾的关键技术,制定了UML建模元素与OWL Lite语法元素之间的映射关系,建立了OWL Lite的UML Profile元模型。曾宪文等<sup>[4]</sup>详细对比分析了OWL特性元素和UML特性元素及本体开发,讨论了OWL本体通过UML类图向面向对象语言转换的方法。郁书好等<sup>[5]</sup>结合建模实例分析UML和OWL的基本概念和建模原语在本体建模中的开发原理,并对两者在本体建模过程的方法进行分析评价。上述研究方法的特点:主要是比较分析了UML与本体元素以及它们的建模方法,建立了UML建模元素与本体的OWL语法元素之间映射关系,从而促进现有的本体开发过程,提高开发效率。这为我们如何将本体应用于UML建模提供了借鉴和参考。

## 1 本体和统一建模语言(UML)的概述

### 1.1 本体

古希腊哲学以探讨万物本原为开端,提出了本体这个概念,以研究这个世界上什么是存在,以及如何存在的。随着计算机科学的发展,Neches等人

2007年10月15日收到甘肃省教育厅科研项目(0603-10)资助

第一作者简介:王谦(1977-),男,汉族,硕士,研究方向:信息处理。

把本体的概念引入 AI领域,并将本体定义为“给出构成相关领域词汇的基本术语和关系,以及利用这些术语和关系构成的规定这些词汇外延的规则的定义”<sup>[6]</sup>,以研究在计算机中什么是存在,以及如何存在的。在 1993年,Gruber给出了本体的一个最为流行的定义<sup>[7]</sup>,即“本体是概念模型的明确的规范说明”。后来,Borst在此基础上,作了少许修正,把本体定义为<sup>[8]</sup>:“本体是共享概念模型的形式化规范说明”。德国卡尔斯鲁厄大学的 Studer等对上述两个定义进行了深入的研究,认为本体是共享概念模型的明确的形式化规范说明。这包含 4层含义<sup>[9]</sup>:概念模型 (conceptualization)、明确 (explicit)、形式化 (formal)和共享 (share)。

由本体的定义可知,本体的本质是概念模型,表达的是概念及概念之间的关系。在文献 [10]中,Perez等认为本体可以按分类法来组织,他归纳出表示本体的 5个基本建模元语 (Modeling Primitive)。这些元语分别为:类 (classes)、关系 (relations)、函数 (functions)、公理 (axioms)和实例 (instances)。通常也把 classes写成 concepts。

元语就是表示语言的语言。基于建模元语,有多种本体表示语言。现有的本体表示语言可以分为两类<sup>[11]</sup>,第一类是基于一阶谓词逻辑的,如框架逻辑 (Frame Logic)、描述逻辑 (Description Logic)<sup>[12]</sup>等。第二类是基于 XML标准的,这些语言包括:RDF(S)、SHOE、XOL、OML、OIL、DAML+OIL。

构造一个本体有各种本体构造方法和构造工具。目前比较常用的构造方法有:骨架法 (又称 Enterprise法)、TOVE法、Methontology方法、SENSUS的描述法、DEF-5方法以及七步法。目前比较常用的本体开发工具,如 Stanford大学 Medical Informatics小组开发的 Protégé2000,由 DARPA支持的 DAML项目组开发的 UBOT,而同属于 DARPA支持的 DAML项目组也开发了 DUET,此外,还有 Ontolingua、Ontosauns、OntoEdit、WebOnto等。

## 1.2 统一建模语言

作为一种建模语言,UML的定义包括 UML语义和 UML表示法两个部分。

### (1)UML语义

UML语义用于描述基于 UML的精确元模型定义。元模型为 UML的所有元素在语法和语义上提供了简单、一致、通用的定义性说明,使开发者能在语义上取得一致,消除了因人而异的最佳表达方法所造成的影响。此外 UML还支持对元模型的扩展定义。

### (2)UML表示法

UML表示法是用来定义 UML符号的,它为开发者或开发工具使用这些图形符号和文本语法为系统建模提供了标准。这些图形符号和文字所表达的是应用级的模型,在语义上它是 UML元模型的实例。

UML的重要内容可以由下列五类图来定义:

用况图 (Use case diagram),从用户角度描述系统功能,并指出各功能的操作者;静态图 (Static diagram),包括类图、对象图;行为图 (Behavior diagram),描述系统的动态模型和组成对象间的交互关系,包括状态图和活动图;交互图 (Interactive diagram),描述对象间的交互关系,包括顺序图与合作图;实现图 (Implementation diagram)。包括构件图、部署图。

除了上述所定义的五类图形建模元素之外,UML还提供了一些扩展机制。用户可以用这些机制定义 UML基本元素的特殊变种,从而获得适合自己需要的扩展。

UML的扩展机制包括约束、标记值和衍型<sup>[13]</sup>。

#### (1)约束

约束是用文字表达式表示的语义限制。每个表达式有一种隐含的解释语言,这种语言可以是正式的数学符号;或是一种基于计算机的约束语言,如 OCL;或是一种编程语言;或是伪代码或非正式的自然语言。约束可以表示不能用 UML表示法来表示的约束和关系。

#### (2)标记值

标记值是一对字符串 (一个标记字符串和一个值字符串)存储着有关元素的一些信息。标记值可以与任何独立元素相关,包括模型元素和表达元

素。标记是建模者想要记录的一些特性的名字,而值是给定元素的特性的值。例如,标记可以是 author 而值是对元素负责的人的名字,如 Charles Babage。

### (3) 衍型

衍型是在一个已定义的模型元素的基础上构造的一种新的模型元素。衍型的信息内容和形式与已存在的基本模型元素相同,但是含义和使用不同。例如,商业建模领域的建模者希望将商业对象和商业过程作为特殊的建模元素区别开来,这些元素的使用在特定的开发过程中是不同的。

## 2 面向对象建模与本体建模

UML是用于面向对象(O-O, Object Oriented)建模的语言,人们常常把O-O模型和本体模型看作不同的模型,因此用不同的语言来刻画它们。其实,这两者之间有不同点,也有相同点<sup>[14]</sup>。其不同点在于:

(1) 两者的建模目的不同,O-O模型是抽象的,用于消除和简化不必要的概念与关系,只遴选必要的知识来解决某个特定的问题。本体模型在于知识表达,往往需要囊括一个领域中所有的知识,强调知识的完备性。由于知识总在发展变化,因此本体建模是个不断完善的过程,这要求本体的开发工具能支持本体的持续更新和相互引用。

(2) O-O模型用于在软件系统开发时,创建精确、具体的实例。本体模型关注的是概念层的问题,虽然能说明某个事实,但是却不具体指导软件开发中元素的建立。本体的“实例”可能仅仅是一种观点,并且关于某个“实例”的多种观点可能共存于该本体之中,虽然多种观点相互并不矛盾。

它们的相同点在于:两者都是表达现实世界中可以用来处理的概念;两者都建立在类概念和关系之上,比如子类和聚合关系。无论是本体还是面向对象思想,都是为了得到在某个领域上的可重用的模型。

同一个领域的O-O模型和本体模型的相似性

是UML有效开发本体的基础,但O-O模型和本体模型的差异导致UML在一些方面不适于进行本体建模,比如本体中的属性是第一级的建模元素,而在UML中属性和关联都不是第一级的。为此,就必须扩展UML。

## 3 将本体用于UML建模

UML从早期的版本开始便受到计算机产业界的重视。OMG(Object Management Group)的采纳和大公司的支持把它推上了实际上的工业标准,使它拥有越来越多的用户。它已被用于多种类型的系统建模,如信息管理系统、通信与控制系统、嵌入式实时系统、分布式系统和系统软件等。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。而且它的应用并不仅限于计算机软件,还可以用于非软件系统,例如硬件设计、业务处理流程、企业或事业单位结构与行为建模。UML的出现也在学术界引起了高度重视,例如每年一次的UML研讨会以及其他一些场合发表的许多关于UML的研究文章。大量与建模问题有关的学术著作和论文也越来越多地使用UML的概念和表示法。UML之所以受到如此众多的学者和用户的重视,主要由于以下几个原因<sup>[15]</sup>。

第一个原因是统一行动产生的积极影响。

第二个原因是UML具有比以往各种O-O方法更丰富的表达能力。

第三个原因是UML在语法和语义的定义方面所作的贡献。

第四个原因是UML在标准化进程中所做的大量工作。

(1) UML所能起到的积极作用<sup>[15]</sup>

UML的统一作用首先在建模概念和表示法方面体现出明显的效果。UML的作用不限于面向对象建模,例如,用况图、活动图和状态图可分别用于需求建模、过程建模以及状态建模。这些图所涉及的概念实际上超出了面向对象的范畴。

(2) UML不能起到的作用<sup>[15]</sup>

正如 M. Fowler指出的, UML是一种建模语言而不是一种方法。所以,一个明显的事实是 UML不能起到一种建模方法的作用。它不能取代任何一种已经存在,或者可能新出现的建模方法;更不能成为一种统一的建模方法。UML是独立于方法的。就是说,不同建模方法都可以使用 UML概念和表示法。这就是 UML在各种方法中所能起到的作用,仅此而已。另外, UML也不能作为一种形式化需求描述语言使用。尽管 UML自身的定义部分地采用了形式化技术,但并不是严格、完全的形式化。它是以“半形式化”的方式定义的。

### (3)UML的缺点与问题<sup>[15]</sup>

在 UML World 2000研讨会上人们就指出了它的许多缺点。在 2000年 10月由美国 Software Development期刊组织的研讨会上提出的缺点包括:庞大的内容、无道理的复杂性以及缺少良好定义外围。经常被提到的其他问题还包括不精确性、工具实现的不完备性和缺少对体系结构建模的支持。来自工业界的主要批评是,它过于庞大和复杂,用户很难全面、熟练地掌握它,大多数用户实际上只使用它很少一部分概念。有许多概念使用感到困惑,含义不清,而且很少被使用。来自学术界的批评主要针对它在理论上的缺陷和错误,包括语言体系结构、语法、语义等方面的问题。

除上述提到的缺点之外,还有其他缺点。比如,在建模者之间以及建模者与用户之间沟通的过程中, UML语言的象征意义在一个图中应当被表述得足够准确和详细,以便对于不同的阅读者来说都提供了充足的信息。然而,一方面 UML的规范中没有提供一个标准来衡量“怎样的 UML图是描述充分的”;另一方面, UML作为一个语言,也无法直接在某个硬件平台中被语法检错和调试。所以在工程中使用 UML图,应该有相应的文字来描述它。为此, UML也提供了几种扩展机制,允许建模者在不用改变基本建模语言的情况下做一些通用的扩展。这些扩展机制已经被设计好,以便于在不理解全部语义的情况下就可以存储和使用。但是,扩展违反了 UML的标准形式,并且使用它们会导致相互

影响。在使用扩展机制之前,建模者应该仔细权衡它的好处和代价,特别是当现有机制能够合理工作时。典型地,扩展用于特定的应用域或编程环境,可是它们导致了 UML方言的出现,包括所有方言的优点和缺点<sup>[13]</sup>。

基于前面对面向对象的建模与本体建模的分析,在此简单地比较一下本体与 UML中的关系:

本体中概念之间的关系包括: association关系(用来描述概念之间的对等关系)、is-a关系(用于指出事物间抽象概念的类属关系)、instance-of关系(实例关系是典型的概念和个体之间的二元关系)、has-a关系(概念之间的整体和部分的的关系)。

UML定义的关系主要有 6种: Dependency关系(A的变化会影响元素 B,则 B依赖 A)、Generalization关系(通常所说的继承关系,特殊个体 is kind of 一般个体)、Association关系(元素间的结构化关系,是一种弱关系,被关联的元素间通常可以被独立地考虑)、Realize关系(元素 A定义一个约定,元素 B实现这个约定,则 B和 A的关系是 Realize)、Aggregation关系(表示部分和整体的关系,整体 has-a 部分)和 Composition关系(组合是聚合关系的变种,表示元素间更强的组合关系。如果是组合关系,如果整体被破坏则个体一定会被破坏,而聚合的个体则可能是被多个整体所共享的,不一定会随着某个整体的破坏而被破坏)。

从它们二者的关系看,确实存在着相似之处,因此,我们可以考虑将本体用于 UML建模,借助于本体来表示 UML中的概念及其关系,以弥补 UML的缺陷,从而使 UML图描述得更充分准确。

下面描述了将本体用于 UML的建模过程:

首先,根据问题域进行需求分析和系统分析,提取出概念及概念之间的联系,然后用本体对其进行准确、充分地表示。

面向对象建模专家利用 OOA和 OOD的方法,如 Booch方法、Coad/Yourdon方法、Jacobson(OOSE)方法、Rumbaugh(OMT)方法,对某个领域进行分析,得到面向对象的模型。

领域专家利用本体建模的方法,如基本顶级本

体、本体驱动的建模规则、可用属性类型,形式化的本体属性或关系等等,对某个领域的经验与认识加以本体化的改造,得到符合要求的本体模型。

然后,根据这两个模型进行取长补短,加以集成,得到一个本体化的面向对象模型。最后,再根据这个模型进行软件设计。或者,也可以先由面向对象建模专家建好模型,再根据建模人员之间以及建模人员与用户之间的交流所产生的不一致的概念和看法进行本体改造,如此反复,直到模型变得清晰、准确为止。

UML作为一种建模语言,还无法对其进行语法检错和调试。对此,我们可以借助于描述逻辑来解决。描述逻辑是基于对象的形式化知识表示的方法,它是一阶谓词逻辑的可判定子集<sup>[16]</sup>。一阶谓词逻辑具有很强的表达能力,但它致命的缺点是它的不可判定性。虽然描述逻辑没有一阶谓词逻辑的表达能力强,但是它能够提供可判定的推理服务,它能保证推理算法总能终止,并返回正确的结果。因此,我们先用本体描述 UML 中的概念,然后,再把概念转化为形式化语言,最后,再通过描述逻辑来推理判断,从而对 UML 进行语法检错和调试。

#### 4 结束语

UML是系统建模的描述语言,其图形化的建模更加直观,但是,这种直观图形表示法往往会引起表达不够充分明确,影响了建模的效率。而且,UML的逻辑基础薄弱,存在自动推理方面的不足,难以对其进行语法检错。本体作为描述概念及概念之间关系的概念模型,可以更加充分地描述领域知识。本文首先用本体的方法来表示 UML 中的概念及其关系,以使 UML 能更充分准确地描述问题;接着提出描述逻辑来解决 UML 中的语法检错和调试这个问题。但是,在 UML 中涉及到了状态及其转换,状态在描述逻辑无法表示,需要扩展描述逻辑。

#### 参 考 文 献

- 1 Cris K. UML2001: a standardization odyssey. *Communication of the ACM*, 1999; (42): 10
- 2 钟凌燕,高 济. 一种基于 UML 的本体论建模方法. *计算机工程*, 2003; 29(2): 38-39
- 3 王 翀,何克清,刘 进. 基于 OWL 元模型的本体建模研究. *武汉大学学报(理学版)*, 2004; 50(5): 581-585
- 4 曾宪文,陈向东,杨明福. 基于 UML 的本体建模研究. *计算机应用与软件*, 2006; 23(7): 42-43
- 5 郁书好,苏守宝,刘仁金. UML 和 OWL 在本体建模中的比较研究. *计算机技术与发展*, 2007; 17(1): 155-157
- 6 Neches R, Fikes R E, Gubner T R, et al. Enabling technology for knowledge sharing. *AIM magazine*, 1991; 12(3): 36-56
- 7 Gubner T R. A Translation approach to portable ontology specifications. *Knowledge Acquisition*, 1993; 5: 199-220
- 8 Borst W N. Construction of engineering ontologies for knowledge sharing and reuse. PhD thesis. University of Twente. Enschede, 1997
- 9 Studer R, Benjamins V R, Fensel D. Knowledge engineering principles and methods. *Data and Knowledge Engineering*, 1998; 25(1-2): 161-197
- 10 Perez A G, Benjamins V R. Overview of knowledge sharing and reuse components: ontologies and problem-solving methods. In: Stockholm V R, Benjamins B, Chandrasekaran A, eds. *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5)*, 1999; 1-15
- 11 Gómez-Pérez A, Corcho O. Ontology Specification Language for the Semantic Web. *IEEE Intelligent Systems*, 2002; 154-60
- 12 Baader F, Calvanese D, McGuinness D L, et al. *The description logic handbook (Theory, implementation, and applications)*. UK: Cambridge University Press, 2003
- 13 Rumbaugh J, Jacobson I, Booch G. *The unified modeling language reference manual*. Reading, Massachusetts: Addison-Wesley Longman, Inc., 1999
- 14 Knublauch H, Rose T. Round-trip engineering of ontologies for knowledge-based systems. In: *Proceeding of the 12th International Conference on Software Engineering and Knowledge Engineering*, 2000; 239-247
- 15 邵维忠,杨芙清. *面向对象的系统设计(第2版)*.北京:清华大学出版社,2007
- 16 周 鑫,张建军. 基于描述逻辑的语义 Web 本体研究. *科学技术与工程*, 2007; 7(3): 329-330

## Research on Applying Ontology to UML Modeling

WANG Qian ZHAO Shao-feng ZANG Feng-kui LIMing

(College of Computer and Communication Lanzhou University of Technology Lanzhou 730050, P. R. China)

[Abstract] Ontology has been applied as a kind of model in computer science which can express the hierarchy of concepts and semantics in effect. UML has been the industry standard formalism in the field of object-oriented modeling. UML and ontology are introduced in general. There are similarities and differences between object-oriented modeling and ontology modeling and it is analyzed how to apply ontology to UML modeling and solve disadvantages of UML so that UML can denote the model information adequately.

[Key words] ontology UML object-oriented description logic

(上接第 391 页)

### 参 考 文 献

- 1 谢 涛,陈火旺,康立山.多目标优化的演化算法.计算机学报. 2003; 26(8): 997-1003
- 2 Pareto V. Cours D'. Economie politique volume I and II. F. Rouge Lausanne 1896
- 3 Fonseca C. Fleming P. An overview of evolutionary algorithms in multi-objective optimization. Evolutionary Computation. 1995, 3(1): 1-16
- 4 Zitzler E. Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation. 1999; 3(4): 257-271
- 5 Zitzler E. Laumanns M. Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization<sup>1</sup>. EURO-GEN 2001—evolutionary methods for design, optimization and control with applications to industrial problems. Athens, Greece 2001
- 6 Srinivas N. Deb K. Multiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation. 1994; 2(3): 221-248
- 7 Deb K. Pratap A. Agarwal S. et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation. 2002; 6(2): 182-197

## Clonal Selection Algorithm for Multi-objective Optimization

LI Heng-jie HAO Xiao-hong\*, ZHANG Lei

(The School of Electrical Engineering and Information Engineering Lanzhou University of Technology Lanzhou 730050, P. R. China)

[Abstract] The evolutionary multi-objective optimization techniques are used to find the non-dominated set of solutions and distribute them uniformly in the Pareto optimal front. After comparing and analyzing the developing history of evolutionary multi-objective evolutionary algorithms, a multi-objective clonal selection algorithm is proposed based on clonal selection principle of immune system. Only some Pareto optimal solutions are selected for further evolutionary operation, and a simple mechanism is used to maintain a good spread of Pareto optimal solutions in the algorithm. It is shown by experimental results that the method can reach the Pareto optimal front very well, retain the diversity of the solutions better, and use much less time.

[Key words] multi-objective evolutionary algorithm clonal selection algorithm multi-objective optimization Pareto optimal solution