

对象式软件需求模型及其机器支撑*

张家重 王志坚 伊波 徐家福

(南京大学计算机软件研究所 南京 210093)

(南京大学计算机软件新技术国家重点实验室 南京 210093)

摘要 为了研究需求驱动软件自动化技术,研制对象式软件需求分析支撑系统的需要,文章提出了一个层次化对象式软件需求模型 NDHORM(Nanjing daxue hierarchical object-oriented requirements modelling),它主要包括对象关系模型、类关系模型和类字典3个组成部分.文章基于对对象式需求模型的简要讨论,详细介绍了 NDHORM 模型的组成、层次结构及对象精化,给出了 NDHORM 的构模过程,最后简要介绍了 NDHORM 模型的机器支撑系统.

关键词 软件自动化,面向对象,需求分析,需求模型,机器支撑.

中图法分类号 TP311

由于研究从图形化软件需求定义到形式功能规约的转换途径的需要,我们对面向对象技术、方法与语言进行了深入研究.^[1]近期提出了一个层次化对象式软件需求模型 NDHORM(Nanjing daxue hierarchical object-oriented requirements modelling),设计了一种图形化对象式软件需求定义语言 NDORL(Nanjing daxue object-oriented requirements definition language)^[2],研制了对象式软件需求分析支撑系统 NDORASS(Nanjing daxue object-oriented requirements analysis support system).^[3]这些工作是南大计算机软件所对需求驱动软件自动化技术及系统的初步探索.

NDORASS 系统支持用户借助 NDHORM 模型及方法进行面向对象的需求构模,对所构模型进行一致性、完整性检查,从而形成软件需求定义,并可将来需求定义自动转换成用面向对象规约语言 OOZE(object-oriented Z environment)书写的形式功能规约. NDORASS 系统由需求定义支撑子系统 NDORS(Nanjing daxue object-oriented requirements system)和需求定义自动转换子系统 NDFUNS(Nanjing daxue functional-specification system)组成. NDORS 用以支撑软件需求定义,特别是需求模型 NDHORM 的构作. NDFUNS 实现从 NDORL 语言描述的需求定义到 OOZE 语言书写的形式功能规约的自动转换.

本文首先简要讨论对象式需求模型的有关问题,然后着重介绍 NDHORM 模型的组成部分、模型的层次及对象精化过程,并给出了相应的构模过程,最后简要介绍了 NDHORM 模型的机器支撑系统 NDORS. NDORASS 系统的其他部分内容将另文介绍.

1 对象式需求模型

近年来,人们为克服传统的功能需求模型中存在的功能本身的易变性、功能分解结构的随意性以及功能结构与现实问题结构常难对应等问题,而采用面向对象 OO(object-oriented)方法,使得有关研究成为当前软件需求工程的热门课题,并展现出良好的应用前景.

一般认为,需求获取是需求分析的核心,在 OO 构架之下,构作需求模型的过程就是需求获取的过程.对象式需求模型是以 OO 方法,通过对对象及其相互关系来表达软件需求的一种模型.

(1) 基本模型

尽管不同的 OO 方法所采用的具体模型不尽相同,但都可归结为如下 5 个基本模型:①整体-部分模型:描述对象(类)是如何由简单的对象(类)构成的,亦称聚合模型.②分类模型:描述类之间的继承关系.③类-对象模型:描述属于

* 本文研究得到国家 863 高科技项目基金和国家杰出青年科学基金资助.作者张家重,1965 年生,博士,教授,主要研究领域为软件自动化.王志坚,1959 年生,博士,教授,主要研究领域为软件自动化,归纳推理.伊波,1958 年生,博士,副教授,主要研究领域为类比程序设计,软件自动化.徐家福,1925 年生,教授,博士生导师,主要研究领域为软件语言,软件自动化,新型程序设计.

本文通讯联系人:张家重,济南 250014,山东师范大学计算机系

本文 1995-05-18 收到原稿,1997-06-17 收到修改稿

每个类的对象所具有的行为。④对象交互模型:描述对象之间的交互方法。⑤状态模型:描述对象在其生存期内的行为。

(2) 代表性方法与存在的问题

一般认为,代表性方法有:①Coad等人于1991年提出的一种演进式面向对象需求方法。②Shlaer等人于1988年借助信息建模方法提出的一种面向对象方法。③Rumbaugh等人借鉴了Shlaer途径的设计思想,于1991年提出的一种对象建模技术OMT(object-oriented modelling technology)。④Firesmith于1993年提出的ADM-3[7]可以说是一种较完整的途径。⑤Seidewitz等人于1992年提出的一种以面向Ada语言而著称的OO方法。[8]

然而,正如文献[9]所述,现有OO方法或多或少都存在着一一些问题,如:①忽略行为建模:由于OO方法强调定义问题对象、类及其接口的静态模型,而对于动态模型只限于用一个状态机来刻画个体对象的行为,大多数OO方法对于系统的整体行为缺少必要的分析。②忽略功能建模:一般的OO方法往往较少注意系统或对象的功能,尽管它使人们放弃了功能分解方法,但它丝毫没有减轻分析目标系统要“做什么”的负担。系统中对象之间的消息传递并不能完整体现其功能和精确地描述其行为(即操作)的输入和输出关系。

此外,现有方法大都采用非形式方法刻画模型,缺少较严格的建模语言,并且很少注意模型的完整性和一致性问题,难以直接适合从“非形式”到“形式”自动转换的需要。由此,为便于实现这种转换,我们分析与借鉴了现有代表性方法的设计思想,提出了一个适于软件自动化技术与系统研究需要的需求模型,称为层次化对象式软件需求模型NDHORM,它以精确定义的需求定义NDORL语言作为其建模语言。

2 NDHORM 模型组成

对象式软件需求模型NDHORM主要包括对象关系模型、类关系模型和类字典三部分,以下分别加以介绍。

2.1 对象关系模型 ORM(object-relationship model)

对象关系模型是刻画问题领域对象的动态行为及其关系的模型。行为关系说明对象间的交互通信及消息传递情况,亦即反映对象的请求行为和接收行为。对象之间消息的一次传递称为一个事件。对于消息的发送对象而言,该事件称为请求事件;对于消息的接收对象而言,该事件称为接收事件。

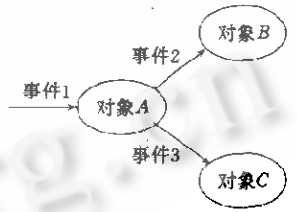


图1 对象关系示意图

该模型采用行为(或事件)制导的方法来识别对象,并借助由抽象到具体而逐步精化的手段,不断地对某些对象实施精化步,而得到一个层次化的对象关系模型。显然,以对象及其之间关系为主体的层次化对象分析与建模过程符合人们的思维方式,有利于对需求定义及其构造过程的复杂性进行有效控制。

采用图形化方式描述的对象关系模型叫作对象关系图,其主要成份量是对象和事件。对象由结点表示,并以相应的对象名字来标志,同时它还可以具有属性及行为说明等信息。事件由与对象结点相关联的一有向弧表示,并以一个事件名字来标志,同时它还可以具有输入与输出变元。图1所示为一层对象关系图的简单示意图。有关对象精化与层次模型将在本文第3节中详细介绍。

2.2 类关系模型 CRM(class-relationship model)

为便于刻画相似对象的共性以及方便组织和复用需求定义,我们引入类关系模型,它是刻画对象所属类、类成分(属性与行为)及类间关系的模型。类间关系包括聚合关系和子类关系两种。该模型要求目标系统中的每一对象均应属于某个类。类可以是虚拟的,即不存在其实例对象。类关系模型以对象关系模型为基本框架而构造,其聚合关系对应于对象关系模型的层次关系。类关系模型本身不层次。子类关系根据问题的实际需要而定义。

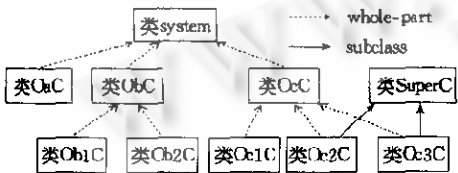


图2 类关系示意图

图2所示为一个类关系模型示意图。其中结点代表类,有向弧代表类间关系,亦即 whole-part 表示聚合关系,subclass 表示子类关系。类 SuperC 是作为类 Oc2C, Oc3C 的父类而引入的新类。

2.3 类字典 CD(class dictionary)

类的属性和行为除了可以在模型图中列出必要的名表外(图2中从略),其具体内容均须在类字典中精确刻画,且主要采用形式化方法。其中所允许的非形式描述仅作为概念解释,以便非专业人员理解,而形式化成分便于作为软件设计及进一步到功能规约转换的准确依据。特别地,关于行为的功能描述采用了前、后置断言的方法,这样可以方便地

描述对象状态空间的变化,从而起到了某些分析方法中的状态转换模型的作用。

事实上,类字典是对问题领域中相关专业知识的解释说明或详尽描述.有关细节可参阅 NDORI 的语言文本。

3 模型的层次

现有面向对象方法有的没有涉及模型的层次问题,而有的对层次结构刻画得相对薄弱,并缺乏较严格的定义,因而不利于作为大型软件系统的需求模型,不便于有效地控制构模过程的复杂性,也难以提供自动化的机器支撑.有鉴于此,我们在 NDHORM 的对象关系模型中引入了层次结构,并借助行为制导的对象分析方法来构作。

3.1 对象精化

一个软件系统可以建立若干不同层次的,且每层均须完整的、异层均须一致的对象关系模型.其层次关系是一个树型结构,可相对地定义为:对于第 i 层第 j 个模型 $ORM_{i,j}$,选择其中某些对象 $O_k(k=1,2,\dots)$,分别精化为由若干相应分量构成的不同子对象关系模型 $ORM_{i+1,k}(k=1,2,\dots)$,从而由 $ORM_{i+1,k}(k=1,2,\dots)$ 组成了一层对象关系模型 ORM_{i+1} ,则称 $ORM_{i+1,k}(k=1,2,\dots)$ 为同层的,即均属 ORM_{i+1} ;称 ORM_{i+1} 或 $ORM_{i+1,k}(k=1,2,\dots)$ 比 $ORM_{i,j}$ 低一层。

模型的层次结构依赖于需求构模过程中所采取的有限次对象精化.一个对象是否需要进一步精化取决于分析人员对该对象是否已经清楚地理解,并能够准确地表达出来.或者说,当发现某对象不能精确地描述其属性或行为时即行精化.一次对象精化可概括成下列 5 个操作步骤:

(1) 对象选择

在某对象关系模型 $ORM_{i,j}$ 中选择待精化的对象,设为 O_k .

(2) 对象识别

分析问题的本原结构,判定 $ORM_{i,j}$ 中对象 O_k 所具有的行为与哪些对象有关,由此识别出组成对象 O_k 的一些关键对象,而得到 O_k 的子模型 $ORM_{i+1,k}$ 的一部分对象.在这里,称 O_k 为精化对象或聚合对象,称 $ORM_{i+1,k}$ 中的对象为 O_k 成分对象。

(3) 行为精化可分为以下两步:

① 对于 O_k 的行为集 $B(O_k)$ 中的每个行为,判定它在 $ORM_{i+1,k}$ 中是否需要分解,作分别处理:(a)是,将其分解成若干子行为,并落实到 $ORM_{i+1,k}$ 中相应的对象上;(b)否,直接落实到 $ORM_{i+1,k}$ 中合适的对象上。

② 对于 $ORM_{i+1,k}$ 中的已有对象,首先分析其每个行为是否需要引用其他对象的行为,若被引用对象在 $ORM_{i+1,k}$ 中尚不存在,则识别出来,并加入到 $ORM_{i+1,k}$ 中.然后建立这些行为引用关系。

(4) 事件构作

在 $ORM_{i+1,k}$ 中的对象之间借助客户/服务(Client/Server)模型确定出所有的行为引用关系,即事件。

(5) 属性识别

对于 $ORM_{i+1,k}$ 中的每个对象,逐一分析其行为将对哪些数据施行操作,从而识别出该对象的一些属性及其有关约束,也可从输入/输出数据中识别出另外一些属性。

当然,步骤(1)~(5)虽然是顺序地给出,但并不意味着它们必须是严格顺序的.在具体的构模过程中可能需要交叉、重复地进行.另外,若新识别出的对象为已有对象,则说明它可以去复用,因此就不需再行分析或精化。

由上述过程可以看出,对象行为的分解类似于功能模型中的功能分解方法.通过这种行为精化为主线的对象构模方法,既有利于刻画目标系统的功能,又便于排除问题领域中的那些无关对象。

3.2 对象精化及模型层次示意

对象精化方法与过程以及对象关系模型的层次可通过图 3 的示意图来简单说明.第 0 层表示将目标系统视为一个单一对象 system,其外部行为有 $se1$,也就是它与外部环境对象之间发生的事件.根据外部行为识别出其在问题领域中所涉及的问题对象 Oa ,并以此进一步识别出与 Oa 具有行为引用关系的对象 Ob 和 Oc ,从而将对象 system 精化,得到第 1 层.其中外部行为 $se1$ 作为对象 Oa 的外部事件,而 $ie1,ie2$ 为内部事件.该层中对象 Oa 无需再行精化,称为原子对象; Ob 和 Oc 尚需精化,并得到第 2 层中相应的(子)对象关系模型.继续该精化过程,直到所有新识别出的对象均为原子对象为止。

如果根据图 3 所示的对象关系模型来构作类关系模型,那么图 2 即为其一个对应结果.可以看出,图 2 中的聚合关系与图 3 中的对象精化,即层次关系相对应.由于具有这种对应关系,在 NDORS 系统中由系统自动生成。

4 构模过程

NDHORM 的构模过程是一个行为制导的过程,即根据行为的抽象级进行对象层次分解.构模过程包括 3 个主要阶段.首先构造问题域的对象关系模型,然后根据对象关系模型构造系统的类关系模型,最后将类关系模型的详细信息描述写入类字典中。

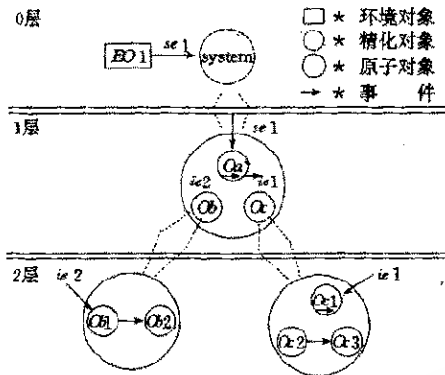


图3 对象关系模型的层次结构示意图

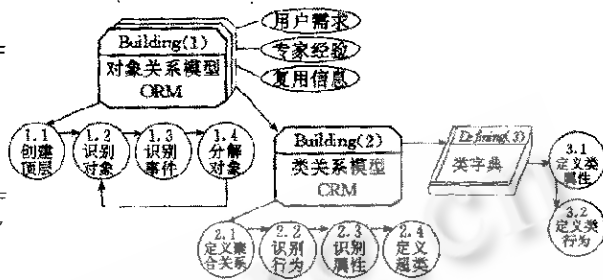


图4 需求构模过程

下面给出具体的构模步骤,这些步骤不是需要严格按顺序执行的,它们可以交叉、循环,有时甚至可以相互引用。

- (1) 建立顶层对象关系图,也就是创建系统对象 system,根据用户对系统的功能需要,识别出与 system 相关的外部事件,如果需要的话,也可以建立若干子系统对象 subsystem;
- (2) 按照行为制导的原则,识别同一抽象级上的内部对象;
- (3) 识别这些内部对象之间的行为作为内部行为;
- (4) 如果某个对象的抽象级别太高,就将它作为非原子对象,按照对象精制步骤精制该对象,得到新的一层对象关系图,同时将与此非原子对象相关的事件作为外部事件,转(2);
- (5) 建立基本的类关系模型,即根据对象关系图,定义类和聚合关系,建立类关系图。其中,对象关系图中的原子对象对应类关系图中的原子类,对象关系图中的非原子对象对应类关系图中的非原子类;
- (6) 对于一些相似的类,引入超类,并定义相关的子类关系;
- (7) 对于类关系图中的每一个类,形式地定义其属性,写入类字典中;
- (8) 对于类关系图中的每一个类,形式地定义其行为,写入类字典中。

上述步骤(1)~(4)是根据问题域中用户需求、分析专家的经验 and 复用信息等来建立对象关系模型;步骤(5)、(6)是构筑类关系模型,步骤(7)、(8)是定义类字典。该过程可直观地如图4所示。

5 机器支撑

作为 NDORASS 的一个组成部分,NDORS 子系统主要支撑 NDHORM 的模型构建。它由总控界面、模型编辑模块、层次管理模块、模型检查模块、模型库管理模块以及文档输出模块等功能模块组成,其系统结构如图5所示。

(1) 总控界面 该界面采用 HCI(human-computer interface)概念模型设计,它实施对各功能模块的控制与调度,并提供用户与系统的接口界面。除提供了多窗口设施外,其友善性还表现在色彩与图元及其操作属性的有机搭配、选单指导、错误报警等方面。

(2) 模型编辑 模型编辑模块实现对象关系模型、类关系模型和类字典的编辑构建与维护。系统提供了表示模型基本成分的若干图元,例如,对象图元、类图元、子类关系图元等,通过使用这些图元,用户可以比较容易地构造出 NDHORM 模型。此外,系统还提供了若干图形处理功能以最大程度地方便用户,例如,采用“橡皮条”和“吸附”技术的图元编辑功能、图元移动过程的动态显示功能等。

(3) 层次管理 对 ORM 层次结构进行管理,其功能包括支持对象精制过程,实现 ORM 的不同层次切换(Zoom in/out),提供层次结构视图,并可借助该视图任意切入某一层次等。

(4) 模型检查 模型检查模块基于一组 NDHORM 的正确性准则,而实现对 NDHORM 模型各部分及其之间的完整性和一致性的自动检查。这些准则主要包括局部性准则和全局性准则。前者对应于 NDORL 语言所定义的局部语法,主要是关于 ORM、CRM 及 CD 的内部合法性、一致性与完整性等的准则。后者则对应于 NDORL 语言所定义的全

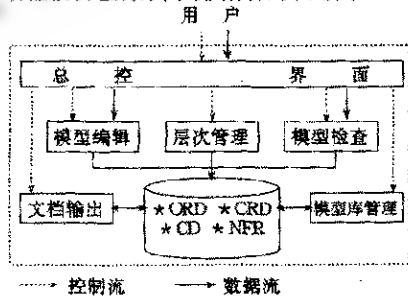


图5 NDORS子系统结构图

局语法,主要是关于 NDHORM 各成分之间的一致性等准则。因篇幅所限,本文不再给出其详细的形式构架。

(5) 模型库管理 模型库管理完成两方面的功能,即单个模型管理和库信息管理。前者实现一模型的图形化显示形式与内部表示形式的相互转换;后者则通过若干管理设施,如浏览、检索、版本更新等,对库中多个软件系统的不同需求模型进行管理。其中 NFR 为非功能性需求。

(6) 文档输出 实现需求模型各部分的完整显示或打印。该模块采用 PostScript 语言较好地解决了图形化模型的合理分割问题,使得所形成的文档规整美观。

6 结束语

NDHORM 模型的层次性及其行为制导的对象分析方法使得用户能够比较自然地识别对象,方便地控制问题的分析过程与模型的复杂性,这是 NDHORM 的主要特色所在。在 SUN Sparc 490 工作站上实现的机器支撑系统 NDORS 采用 HCI 概念模型设计了交互式、多窗口界面和图形化编辑工具,较好地支持 NDHORM 构模过程和模型检查。尽管我们在面向对象构架之下,对需求模型、需求语言及其机器支撑进行了实验性研究,并探索了由图形化需求定义到形式功能规约的自动转换技术,但对于需求级软件自动化而言,所做的工作毕竟只是初步尝试,进一步工作是通过更广泛的试用来完善与提高模型及系统的能力,并期望与已有功能级、设计级软件自动化系统进行集成,以尽可能地实现对软件开发全过程的自动化支持。

致谢 特别感谢课题组的吕建博士、朱鸿博士、金凌紫博士以及陶先平、丁俊华、孙圣强、杨大军和鲍一民等同志。

参考文献

- 1 徐家福等. 对象式程序设计语言. 南京: 南京大学出版社, 1992
(Xu Jia-fu et al. Object Oriented Programming Language. Nanjing: Nanjing University Press, 1992)
- 2 张家重, 吕建, 王志坚等. 一种图形化对象式需求定义语言的设计. 软件学报, 1996, 7(11): 647~655
(Zhang Jia-zhong, Lü Jian, Wang Zhi-jian et al. On the design of a graphical object-oriented requirements definition language. Journal of Software, 1996, 7(11): 647~655)
- 3 张家重. 对象式需求分析及其自动化技术的研究[博士论文]. 南京大学, 1997
(Zhang Jia-zhong. A study of object oriented requirements analysis its automation [Ph. D. thesis]. Nanjing University, 1997)
- 4 Coad P, Yourdon E. Object-oriented analysis (2nd). New Jersey: Yourdon Press, 1991
- 5 Shlaer S, Mellor S. Object-oriented systems analysis: modeling the world in data. New Jersey: Yourdon Press, 1988
- 6 Rumbaugh J et al. Object-oriented modeling and design. New Jersey: Yourdon Press, 1991
- 7 Firesmith D G. Object-oriented requirements analysis and logic design. New York: John Wiley & Sons Inc., 1993
- 8 Sedgewitz F, Stark M. Principles of object-oriented software development with ada. Rockville, Md: Millenium Systems Inc., 1992
- 9 Bailin S C. Object-oriented requirements analysis. In: Marciniak John J ed. Encyclopedia of Software Engineering. New York: John Wiley & Sons Inc., 1994. 740~756

A Hierarchical Object-Oriented Software Requirements Model and Its Mechanical Support

ZHANG Jia-zhong WANG Zhi-jian YI Bo XU Jia-fu

(Institute of Computer Software Nanjing University Nanjing 210093)

(State Key Laboratory for Novel Software Technology Nanjing University Nanjing 210093)

Abstract For sake of doing research on requirements-level software automation and developing an object-oriented requirements analysis support system, the authors have proposed a hierarchical object-oriented requirements model NDHORM (Nanjing daxue hierarchical object-oriented requirements modelling), which includes an object relationship model, a class relationship model and a class dictionary. First, the authors discuss object-oriented requirements analysis methods, then introduce the components, hierarchical structure of NDHORM, and its object refinement mechanism at length in this paper. Further, the authors chiefly address the modeling process of NDHORM. Finally, its mechanical support system is also reported.

Key words Software automation, object-oriented, requirements analysis, requirements model, mechanical support.