一种面向特征的领域模型及其建模过程*

张 伟+, 梅 宏

(北京大学 信息科学技术学院 软件研究所,北京 100871)

A Feature-Oriented Domain Model and Its Modeling Process

ZHANG Wei+, MEI Hong

(Institute of Software, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)

+ Corresponding author: Phn: 86-10-62757801, Fax: 86-10-62751792, E-mail: zhangw@cs.pku.edu.cn

http://www.sei.pku.edu.cn/belljointlab/

Received 2002-12-31; Accepted 2003-03-05

Zhang W, Mei H. A feature-oriented domain model and its modeling process. *Journal of Software*, 2003,14(8): 1345~1356.

http://www.jos.org.cn/1000-9825/14/1345.htm

Abstract: The feature model has been widely adopted as a domain requirements capturing model by most of the current domain engineering methods. But these methods lack a well formatted framework for the feature models which they use. This has led to the redundancy and confusion in feature model representation between different domain engineering methods, and has made domain analysts difficult to built feature models effectively in practice. In this paper, a uniform framework of feature model is presented from the aspects of basic structure, variability representation and constraint mechanism, and variability binding time. A concrete form of this abstract framework is also given based on the different type of features existing in requirements (service, use case, function, and behavior characteristic), and the relationship among them. Then, combining with a real software domain, the modeling process of the feature model is discussed systematically. This approach will be beneficial to successful domain modeling practices.

Key words: feature model; domain analysis; domain engineering; software reuse

摘 要: 特征模型作为捕获领域需求的重要模型已被现阶段的主流领域工程方法所接受,但这些方法缺乏对特征模型组织框架的细致研究和说明,在一定程度上导致了特征模型在表现形式上的冗余性和混乱性,也使得领域分析人员在实践中很难有效地进行领域建模活动.从特征模型的基本组织结构、变化性的表现方式和限制机制、变化性

* Supported by the Key Project of the National Natural Science Foundation of China under Grant No.60233010 (国家自然科学基金重点项目); the National Grand Fundamental Research 973 Program of China under Grant No.2002CB312003 (国家重点基础研究发展规划(973)); the National Science Foundation for Distinguished Young Scholars of China under Grant No.60125206 (国家杰出青年科学基金); the Major Project of Science and Technology Research of Ministry of Education of China under Grant No.MAJOR0214 (国家教育部重大项目); the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No.20010001001 (国家教育部博士点基金)

第一作者简介: 张伟(1978－),男,江苏徐州人,博士生,主要研究领域为领域工程,软件构件技术,软件复用.

的绑定时间等方面对特征模型的组织框架及剪裁机制进行了统一、抽象的描述.在考察服务、用例(use case)、功能、行为特点等不同类型的特征及其相互关系的基础上,给出了一种特征模型的具体形式,并结合具体的领域,对其建模过程进行了详细论述.此项研究对于领域建模活动的成功实施具有一定的指导作用.

关键词: 特征模型;领域分析;领域工程;软件复用

中图法分类号: TP311 **文献标识码:** A

面对日益复杂的软件系统,软件复用被认为是解决"软件危机"、提高软件开发效率和质量、实现软件产业工业化生产方式的重要途径.面向对象技术、软件构件技术、软件体系结构、领域工程、软件再工程等相关技术的发展则为软件复用提供了基本的技术支持,并推动了其在研究和实践中的迅速发展.

软件复用的研究和实践表明,特定领域的软件复用活动相对容易取得成功.这里的领域不是指特定的行业领域,而是指一组具有相似和相近软件需求的应用系统所覆盖的功能区域[1].一个特定的行业领域可能会覆盖若干个功能领域,同一个功能领域也可能贯穿于若干个行业领域中.领域的内聚性(领域知识逻辑上的紧密相关性)和稳定性(在一定时间内,领域知识不会发生剧烈的变化)为软件复用活动提供了可供复用的软件资产和潜在的经济利益,使得特定领域的软件复用相对容易获得成功.

借鉴传统产业的工业化生产方式,软件复用活动包含两个重要子活动:可复用软件资产的生产(development for reuse)和基于可复用软件资产的应用系统开发(development with reuse)[2].领域工程是可复用软件资产生产的主要技术手段,它包含领域分析、领域设计和领域实现 3 个阶段.领域分析在对领域中若干典型系统的需求进行分析的基础上,考虑预期的需求变化、技术发展及客观限制等因素,确定合适的领域范围,识别领域中的共性和变化性,获取一组具有足够复用性的领域需求,并对其进行抽象,形成领域模型.领域设计和实现则以领域模型为基础,识别、开发和组织领域中的构架/构件等可复用资产.在开发同一领域内的新系统时,可以根据领域模型,确定新应用的需求规约,进而选用合适的系统构架,并以此为基础选择构件进行组装,最终形成新的应用系统.这样,新系统的开发不再是从零开始,而是建立在对分析、设计、实现等阶段的软件资产大量复用的基础上.

领域分析是系统地获取领域需求的阶段,这一阶段的输出产品是领域模型.与针对单个应用系统的需求规约模型不同,领域模型是针对领域的需求规约模型.除具有一般需求规约模型的作用以外,在软件复用活动中,领域模型还具有如下两个重要作用:

① 为领域内新系统的开发提供可复用的软件需求规约.

新系统的开发者通过对领域模型进行适当的剪裁和扩展,形成适合该系统的需求规约.为了适应对领域需求的复用要求,领域模型不仅要记录领域内的系统具有的共性功能和质量属性,还要记录这些属性可能具有的变化性,其组织结构还必须具有良好的可剪裁性和可扩展性*.

② 指导领域设计阶段和实现阶段可复用软件资产的生产.

为了提高构架/构件等软件资产的可复用性,一个重要的原则就是保持共性和变化性的分离.领域模型必须提供相应的机制来支持对变化性需求的隔离、封装和抽象.我们也希望领域模型对领域需求的良好组织能够在一定程度上减小需求和设计之间存在的鸿沟**.

领域模型在软件复用过程中的两个重要作用,对它的组织框架和具体形式提出了相应的要求.同时,面向领域的特点也使得一般的需求建模方法不再适用于领域模型的建模过程.因此,采用何种形式的领域模型,如何实施对领域的建模活动,成为领域工程研究和实践中的两个重要问题.

文献[3]首次提出"把特征(feature)作为系统需求规约的组织方式".Davis 认为,特征是从用户角度对系统的感知,用特征对系统需求规约进行模块化组织是一种非常自然的手段.这种面向特征的需求规约组织方式在

* 可剪裁性和可扩展性也是领域模型适应领域发展的客观要求.领域知识具有相对的稳定性,但并不排斥领域的逐渐变化和发展.可剪裁性和可扩展性使得领域模型具有了支持演化的基本能力,从而能够在一定程度上保护对领域工程活动的投资.

** 如何建立需求和构件之间的映射和追踪关系、如何从领域需求模型平滑过渡到领域构件模型,这是当前研究中的难题.其中一个重要的前提就是需求空间如何组织,即领域模型具有什么样的组织框架.

FODA[4]方法中被引入到领域工程的研究和实践中.FODA 使用特征和特征之间的关系,即特征模型,作为领域模型的重要组成部分.在现阶段的领域工程研究活动中,包括 FORM[5],FeatuRSEB[6],PLA[7]等,都采用了面向特征的领域需求组织方式.各种研究方法都把特征作为领域需求空间的一阶实体.以特征模型为中心,由多种相关模型(用例模型[6~8]、需求对象模型[7])共同构成领域模型逐渐成为各种方法共同采用的手段.同时,各种方法也对特征建模活动给出了相应的原则和策略.

但是,对于特征模型的组织框架及其建模过程,当前的研究还存在一些问题.特征模型在 FODA 中首次被引入领域模型中.随后各种方法都对特征模型的结构做了一些扩充、剪裁和修改.FeatuRSEB 方法引入了变化点(variation point)和变量(variant)的概念,表达一组同一维度的变化特征.同时,它还使用 OR,XOR 表示一组同一维度变化特征的多选一和多选多的剪裁特点.FORM 方法为了保持领域特征模型的简洁性仅使用了一种类型的绑定时间.一些方法出于组织特征的需要,在特征之间引入了一些新的关系.但各种方法都没有对特征模型的组织框架进行深入的研究,对特征模型中各种建模元素的语义也没有做出明确的说明.这些情况导致了特征模型在表现形式上的冗余性和混乱性.从目前看来,特征模型还缺少一个统一的组织框架.对于特征模型的建模过程,一方面,缺少对特征模型组织框架的明确定义,另一方面,缺少对特征获取过程的详细可操作的指南,使得领域分析人员在实践中很难有效地进行领域建模活动.

本文在现有领域分析方法所使用的特征模型的基础上,从基本组织结构、变化性的表现方式和限制机制、变化性的绑定时间等若干方面对特征模型的组织框架和剪裁机制进行了统一、抽象的描述.同时,在需求工程中需求的 3 个层次的基础上,考虑有效捕捉领域共性和变化性的因素,引入了需求的行为特点层,然后以此为出发点提出了一种领域特征模型的具体形式,并给出了相应的建模过程.

本文第 1 节主要介绍与特征相关的一些研究方向,对特征给出了一个多视角的定义,并分析了特征模型和传统需求规约模型的优缺点及其相互关系.第 2 节主要论述面向特征的领域建模方法,包括领域特征模型的组织框架、领域特征模型的具体形式、领域特征建模过程等几个子部分.第 3 节通过一个实例检验了本文提出的面向特征的领域建模方法的可行性.最后是对全文的总结.

1 特征和领域特征模型

1.1 关于特征

在进行进一步的论述之前,有必要对术语"特征"的含义加以说明.目前,术语特征常出现在领域工程、特征交互问题、特征驱动的软件开发、软件再工程等若干研究方向和实践中.

特征交互问题(feature interaction problem)大量出现在电信系统中.电信中的特征是指由电信网络提供给电话用户或系统管理者的功能单元[9].电信系统对外界提供的服务由众多的特征通过相互之间的交互提供给用户.特征交互问题是指特征之间存在的非预期的、对系统可能具有危害的交互现象.可以说,以特征为基本单元的用户需求组织方式在电信系统中得到了极好的体现.

特征工程(feature engineering)[10]认为特征是贯穿软件生命周期和跨越问题空间与解空间的一阶实体,并希望特征的引入能够减少用户和软件开发者之间存在的期望差异.其研究内容主要关注特征对软件生命周期的各个阶段的作用和影响.C.R.Turner 把特征定义为软件需求规约中一组相互紧密联系的需求构成的模块.虽然特征工程目前还没有考虑软件复用的因素,但其以特征作为需求空间基本模块的观点和领域工程是一致的.同时,研究特征对软件生命周期各个阶段的作用和影响对领域工程的设计和实现阶段也具有重要的借鉴作用.

FDD(feature-driven development)[11]是一种特征驱动的软件迭代开发过程.FDD 把特征作为每次迭代开发的基本增量单元和里程碑,通过特征优先级控制、特征开发时间控制等措施,达到减少软件项目实施时间的目的.基于此目的,FDD 把特征定义为具有客户价值并可在两星期内实现的软件功能.可以看出,FDD 的特征表现了一种考虑项目管理因素的需求分割方式.同时,系统特征的获取、特征优先级的确定使得 FDD 中的某些活动具有了领域分析的雏形.

文献[12]提出了一种基于特征的软件再工程方法.该方法把遗产系统中具有的可复用特征(能够用在同一

领域的其他系统中)作为软件演化的基本单元,通过收集分散在遗产系统中与某个特征相关的代码并将其封装为一个构件,达到问题空间和解空间的良好映射,从而有利于软件系统的不断演化.该方法对特征的理解和领域工程是基本一致的.对于可复用性的要求体现了该方法面向领域的特点.

在领域工程的研究中,各种方法对特征的定义并不完全相同.较早出现的 FODA 认为,特征是软件系统中用户可见的、显著的或具有特色的方面、品质、特点等[4].FeatuRSEB 则把特征的视角扩大到系统的用户和客户,认为特征是一个用户或客户可感知的系统特点[6].本文在综合现有各种研究和定义的基础上,对特征给出了一个多视角的定义:

从需求规约的组织结构角度来看,特征提供了一种对需求的分割和组织方式,即以特征作为需求空间内的一阶实体,系统具有的特征及其相互关系构成了系统的需求空间;从需求的内涵来看,一个特征体现了系统具有的某种能力或特点,反映了需求获取的参与者对系统的某种要求或理解;从需求的类型上看,一个特征可能是一种功能性的需求,或是对系统质量属性的要求,或者是外部环境对系统的某种约束条件.

在领域工程的实施过程中,出于捕捉领域设计及实现阶段共性和变化性的目的,有些方法(如 FODA)把特征的概念隐含地扩展到了这两个阶段.本文重点关注于领域分析阶段,仍然把特征作为一种需求的概念来对待.

1.2 特征模型和传统需求规约模型的比较

领域特征模型是面向特征的领域需求规约模型,通过记录领域具有的一组相对稳定的特征以及特征之间的关系反映整个领域的软件需求.这组特征分属于两种类型:共性特征和变化性特征.共性特征存在于领域内的每个成员系统中,变化性特征只存在于领域内的某些系统成员中.共性特征是领域的主要复用源泉,而变化性特征的范围可控性决定了领域中应用系统的生产成本[13].领域的范围决定了共性和变化性之间的平衡关系.

传统的软件需求规约一般采用结构化的自然语言描述需求,并辅以图形化模型增强对需求的理解.对需求的组织方式通常采用序列号、层次化编码、层次化文本标签等方法[14].这些方法基本上是一种对需求的简单罗列,也没有显式地捕捉需求之间存在的结构关系和语义关系.

特征模型本质上也是一种需求的组织方式.与传统软件需求规约的组织结构相比,特征模型具有以下优点:

(a) 可复用性:由于引入了对变化性的表现机制,特征模型可以描述一类具有共性需求并同时表现出一定变化性的应用系统.基于此,一旦关于某个领域的特征模型被建立,则该领域内成员系统的需求可以通过对特征模型的剪裁和扩展得到,从而达到对需求的复用.

(b) 结构良好:特征表示了一个相对独立和紧凑的需求单元.特征模型通过识别特征之间存在的语义关系,把各个单独的特征组织成一个有机的整体.这样,特征模型形成了对需求空间的一种良好分割和组织.而在传统的需求规约模型中,一个特征可能体现为若干段分散在需求规约中的文字,特征之间存在的关系也没有被明确地说明.

(c) 易于交流:如同每一个设计模式都对应一个名称一样[15],每个特征也被赋予了一个体现其含义的简短名称.所有的特征名称构成了领域内一个公共的术语空间,促进了软件开发参与者之间的相互交流.

(d) 易于图形化建模:特征模型捕获了领域具有的特征以及特征之间存在的关系.这种实体-关系的模式使得特征模型易于用图形化的方式表现出来.而且通过相应的工具支持,能够提高特征建模过程的效率.

但是,建立特征模型所需的投资是相对昂贵的.一些特征反映了领域的用户在长期的实践中形成的某种切实可行的或高效的工作手段和方法,是领域知识的一种沉淀.系统地获取特征需要进行针对整个领域的分析活动.在领域工程中,对领域分析的早期投资被分摊到后期的多次复用活动中.而在针对单个应用系统的需求获取活动中,进行领域分析所需的投资可能是项目预算所不能忍受的.

应该看到,领域特征模型并不是对传统需求规约模型的完全替代.在以领域工程的成果为基础的应用系统开发活动中,领域特征模型为该系统的需求提供了一个复用的框架.即便如此,软件工程师仍然可能要对领域特征模型剪裁后的结果作出一定的扩展以满足该系统的特殊需求.在这种扩展过程中,传统需求规约模型和特征模型形成了一种互补的关系.

2 面向特征的领域建模方法(FODM)

领域需求模型在软件复用中所具备的两个重要作用对它的组织框架和建模过程提出了新的要求.本节主要介绍面向特征的领域需求模型框架,并在此基础上给出了本文采用的特征模型的具体形式及建模过程.这种方法称为面向特征的领域建模方法(FODM).

2.1 面向特征的领域模型框架

下面从领域特征模型的基本组织结构、变化性的表现机制、变化性的绑定时间、变化性的限制机制、质量特征的组织方式等几个方面来论述领域特征模型的组织框架.

(a) 基本组织结构

一个特征描述了一个相对独立的需求实体.特征模型在关注单个特征的同时,也注重发现特征之间存在的语义关系.特征模型的基本组织结构提供了一种把各个独立的特征组织在一起的方式.

现有领域工程方法中的特征模型基本都采用了层次式的方式来组织特征.各层特征之间以整体-部分关系(whole-part association,简称 WPA)联系在一起.在面向对象方法中,WPA 被用来描述对象之间的整体-部分关系,能够在 OOA 和 OOD 阶段表现对象之间存在的丰富语义[16].在特征模型中,WPA 主要用来表现两种类型的语义:① 整体特征对部分特征的控制和协调作用;② 整体特征和部分特征逻辑上的紧密结合性.如图 1 所示的图元操纵服务及其子特征的 WPA 关系就反映了整体特征对部分特征的控制和协调作用.如在一次典型的操作中,用户先后发出图元选择和图元移动的请求,图元操纵服务首先调用图元选择,然后记录被用户选中的图元,接着以被选中的图元为参数调用图元移动.这样,图元操纵服务通过控制图元选择和图元移动的执行,并在两者之间传递信息(一种协调行为),从而完成用户的请求.图 1 中图元移动及其子特征之间的 WPA 关系则反映了整体特征和部分特征逻辑上的紧密结合性.即移动模式和移动约束是图元移动执行过程中体现出来的行为特点,两者不能脱离图元移动而单独发生作用;图元移动的执行(在一定条件下***)也必然要体现它的子特征.由 WPA 形成的整体部分结构使得特征模型具有较好的剪裁性和扩展性.部分特征可以方便地从整体特征中删除,而新的部分特征也可以方便地加入到整体特征中.

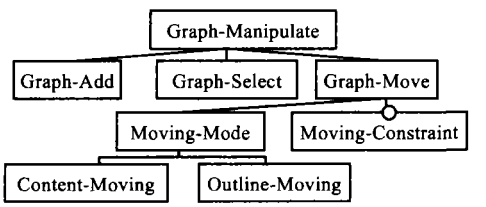

Fig.1 DFM fragement of graph editor
图 1 图元编辑器领域特征模型片断

一般认为,在系统开发过程的各个阶段,使用一致的概念能够减少这些阶段之间存在的鸿沟.而 WPA 也是 OOA 和 OOD 阶段存在的一个概念.因此,我们希望在领域分析阶段引入 WPA 能够对领域设计和实现阶段产生一定的指导作用.

(b) 变化性的表现机制

特征模型提供了两种表示领域变化性的机制:① 特征的可选性;② 特征自身的变化性.特征的可选性依附于 WPA 上,表现为部分特征相对于整体特征的可选性****.图 1 中移动约束相对于图元移动就体现出了这种特征的可选性.特征自身的变化性是指一个特征由于封装了不同的细节而体现出的具有不同行为特点的特征.本文采用维度(dimension)和值(value)的概念来描述特征及其具有的变化性[17].我们把一个自身具有变化性的特征称为一个维度,把它的封装了不同细节的变化性特征称为该维度上的一个值.图 1 中移动模式就是一个维度特征,这个维度上具有整体移动和虚框移动两个值.

维度-值机制对变化性的隔离、抽象和封装提供了很好的支持.维度是其所有值的一个抽象.这种抽象隔离了变化性对其他部分的影响.每一个值则封装了不同的变化性.同时,当新的变化性产生时可以把它作为一个值

*** 该示例中,移动约束相对图元移动是可选的.在选定的情况下,图元移动的执行就会体现出受约束性.这两者在逻辑上是紧密结合的.而移动模式和图元移动则完全是不可分割的.本句中的"一定条件"是指整体特征的可选子特征被选中的条件下.

**** 存在一种特殊情况:一个最高层的特征是可选的,而它并不属于某个整体特征.

添加到维度中,从而达到了一定的扩展性.

维度-值机制没有对同一维度上不同值之间的关系做出强制的规定.而FeatuRSEB方法中的变化点-变量机制则要求同一变化点上的不同变量之间是互斥(mutex)的.这是两者之间的主要不同点.维度-值机制和变化性限制机制(见下面(d))的组合也能够表现与变化点-变量机制同样的语义.

(c) 变化性的绑定时间

领域具有的变化性总要在领域工程阶段或软件系统生命周期的某个阶段被确定下来,称为变化性的绑定时间.典型的绑定时间有复用时(reuse-time)、编译时(compile-time)、装载时(load-time)、运行时(run-time)等[4,6].绑定时间的引入使得特征模型不仅能够描述领域中成员系统之间的差异(如通过剪裁复用时绑定的特征),而且能够描述一个软件系统本身具有的变化性(如通过选择一个运行时绑定的特征).

绑定时间是对系统灵活性的一种要求,并且会影响系统体系结构的设计.绑定时间越靠后,系统体现的灵活性也越高,同时也要求相应的体系结构来支持这种变化性.盲目地追求不必要的灵活性是不可取的,特别是当这种灵活性会对领域工程的投资产生影响时.在实际的领域工程活动中,应该根据客观条件确定此次领域工程支持的绑定时间集合,并对变化性特征的绑定时间作出恰当的选择.

(d) 变化性的限制机制

对变化性的限制体现了领域变化性在绑定/取消绑定时需要满足的约束条件.从绑定的角度来看,特征在其生命周期内存在两种状态:绑定状态和未绑定状态.因此,我们可以把"特征是否处于绑定状态"看成一个二值逻辑命题.为书写方便,本文用特征名称指代该命题的命题变元,则对特征 a,真值函数 T 的取值为

$$T(a)=\begin{cases}\text{TRUE}, & a\text{处于绑定状态}\\ \text{FALSE}, & a\text{处于未绑定状态}\end{cases}.$$

从逻辑命题的角度,可以把特征之间存在的语义约束关系表现为由特征名称通过逻辑运算符构成的命题逻辑公式.这种方式是对FODA[4]方法中特征之间的Require和Mutex关系的扩展.Require和Mutex关系可以用逻辑运算符来表示.如对特征 a,b:

$$a \text{ Require } b \Rightarrow a\rightarrow b;$$

$$a \text{ Mutex } b \Rightarrow a\rightarrow\neg b \Leftrightarrow b\rightarrow\neg a.$$

同时,逻辑运算符比Require和Mutex具有更简洁、强大的表示能力.Require和Mutex是单个特征之间关系的二元符号,逻辑符号→和→¬则是关于命题公式的二元符号,可以表达诸如 $a\vee b\rightarrow c, a\wedge b\rightarrow c$ 等诸多复杂约束关系.这些关系有的必须用多个Require和Mutex关系式才能表达,有的则是其无法表达的.

多选一(SingleSelection)和多选多(MultipleSelection)是特征之间经常出现的两种多元约束关系,特别是同一维度的不同值之间.为了能够简洁地在特征模型中表现出这两种关系,本文对其做出如下形式化的指代:

$$SingleSelection(f_1,f_2,\ldots,f_n)\Leftrightarrow\exists 1\le i\le n(\wedge_{1\le j\le n,j\ne i}\neg f_j)\wedge f_i,$$

$$MultipleSelection(f_1,f_2,\ldots,f_n)\Leftrightarrow\vee_{i=1,\ldots,n}f_i.$$

通过上述方式,一个领域特征模型内特征之间的约束关系表现为一组逻辑公式.但这同时也带来了另外一个问题:如何保证这一组逻辑公式体现的约束关系是合理的.即这组公式之间是否存在冲突使得对领域特征模型的任何绑定结果都不会满足这一组公式;是否一个变化性的特征永远不可能被绑定;是否一个变化性特征永远不可能被取消绑定.下面我们对这一问题进行阐述.

考察特征在其生命期内其绑定状态的变化情况,可以发现,领域特征模型内的特征分属于两个集合.一个集合内的特征在其生命周期内始终处于绑定状态,另一个集合内的特征绑定状态则是待定的.用 $FSet1$ 和 $FSet2$ 分别表示这两个集合,对特征 f,若 $f\in FSet1$,则 f 指代的命题变元的值恒为TRUE;若 $f\in FSet2$,则 f 仍然指代一个值未定的命题变元.

记一个领域特征模型内体现特征间约束关系的一组公式为 $\{R_1,R_2,\ldots,R_n\}$,则这组公式必须满足下面3个性质,才能保证其体现的约束关系是合理的:

性质1. 一致性(consistency).

存在对 $FSet2$ 内所有命题变元的一种解释,使得 $T(R_1\wedge R_2\wedge\ldots\wedge R_n)$=TRUE.

性质 2. 无冗余性(nonredundancy).

$\forall f \in FSet2$,令 $T(f)$=TRUE,存在对 $FSet2$–f 内命题变元的一种解释,使得

$$T(R_1 \wedge R_2 \wedge \ldots \wedge R_n)=\text{TRUE}.$$

性质 3. 必要性(necessity).

$\forall f \in FSet2$,令 $T(f)$=FALSE,存在对 $FSet2$–f 内命题变元的一种解释,使得

$$T(R_1 \wedge R_2 \wedge \ldots \wedge R_n)=\text{TRUE}.$$

对以上 3 个性质的解释如下:首先,如果不满足一致性,则对领域特征模型的任何剪裁结果也不会满足该组公式,这是由于该组公式本身就是不可满足的.无冗余性要求任何一个待绑定特征都有被绑定的可能,否则该特征在领域特征模型内是冗余的,从而可以将其删除(导致出现这种情况的更可能的原因是约束关系出现了错误,或是集合 $FSet1$ 内包含了一个待绑定的特征,正确的处理方式是从这两个方面找原因,而不应只是简单地将其删除).必要性则要求任何一个待绑定特征都有不被绑定的可能,否则该特征应该属于集合 $FSet1$,而不是 $FSet2$.

这样,通过对体现特征间约束关系的这组公式进行性质检查,可以确保特征建模得到的结果是合理的.要注意,满足这 3 个性质仅仅保证了合理性,对正确性的检查还需要其他手段,如对比领域内若干样本系统,检查是否可以通过对领域特征模型的剪裁得到这些系统的特征模型.只有在合理性和正确性均得到保证的前提下,领域建模的后继活动才能顺利进行.

(e) 对质量特征的组织

质量特征也是领域需求的必要组成部分.对质量特征的不同要求会对体系结构的选择产生重要影响.针对质量特征对领域影响范围的不同,我们把质量特征分为全局质量特征和局部质量特征两种类型.全局质量特征影响整个领域的设计和实现,并决定选择何种类型的软件体系结构.而局部质量特征影响领域内某个较小的范围,并对局部体系结构的选择产生作用.

2.2 领域特征模型的剪裁机制

领域特征模型的一个重要作用是为领域内新系统的开发提供可复用的软件需求规约.通过对领域特征模型的剪裁和扩展得到单个应用系统的需求规约,是实现需求复用的一种自然方式.本文论述的特征模型框架对这种剪裁和扩充活动提供了基础的支持(见第 2.1 节).本节主要结合特征的绑定时间、特征之间的约束关系这两个紧密联系的方面,论述对领域特征模型进行剪裁及对剪裁结果的合理性进行检查的过程.

一个特征被触发执行的前提条件是其已被绑定并处于运行时.从这个角度来看,本文统称运行时以前的各个绑定时间段为配置阶段(在这些绑定时间内特征是不可能被触发执行的).下面我们首先考察在配置阶段内特征模型剪裁活动的基本特点.

在配置阶段内的任何一个绑定时间内,对特征模型的剪裁存在两种不同的活动:

(1) 绑定一部分变化性特征(这些特征的绑定时间属性必须等于当前时间).

(2) 删除一部分变化性特征.这些被删除的特征包含两个部分,一部分是绑定时间属性等于当前时间但未被绑定的那些特征,另一部分则是绑定时间在当前时间之后,但出于某些原因被提前删除的特征.通过这两种活动,在配置阶段内的任意一个绑定时间结束时,领域特征模型内的特征分别属于 3 个不同的集合 $FSet1$,$FSet2$,$FSet3$,分别代表已被绑定的特征集合、待绑定的特征集合、已被删除的特征集合.$FSet1$,$FSet2$ 内特征的性质如第 2.1 节所述.$FSet3$ 内特征指代的命题变元的值此后恒为 FALSE.

第 2.1 节所述的 3 个性质此时可以作为检查配置阶段内某个绑定时间之后特征模型合理性的手段.一致性保证了此次绑定时间内的决策是无矛盾的;一个违反无冗余性的待绑定特征可以被划分到集合 $FSet3$ 中;一个违反必要性的待绑定特征则可以被划分到集合 $FSet1$ 中.

在运行时,对集合 $FSet2$ 内特征的配置与配置阶段有显著的不同.这些特征可以在运行时被反复绑定和取消,且当其处于绑定状态时随时可能被触发执行.这种动态性使得无冗余性在运行时刻失去了意义.例如,设特征 a,b 的绑定时间为运行时,且存在约束关系 $a \rightarrow b$.这个约束关系背后的含义可能是指特征 a 对特征 b 运行时产生的某些数据感兴趣.这样,即使当 b 在经过若干次运行后被取消绑定,但由于其产生的数据仍然可能是存在的,

故 a 的绑定还是有意义的.类似的例子可以在第 3.2 节看到,如图元编辑器内的特征图元组合撤销/重做和图元组合就符合上述情况.也就是说,在运行时,特征当前配置情况合理性的检查需要结合特征之间的动态关系才能进行.另外一个不同的点是,对 *FSet2* 内特征的配置可能不需要用户的直接参与,而是依靠内置的程序逻辑自动进行.

2.3 领域特征模型的具体形式

需求工程把软件需求分为 3 个不同的层次:业务需求、用户需求、功能需求[14].业务需求描述了组织结构或客户对软件系统高层次的目标要求.用户需求描述了用户和系统的交互过程.功能需求描述了为实现特定的业务需求,软件系统必须具备的功能.

这 3 个层次都是特征可能存在的地方.业务需求体现了软件系统具有的业务能力,这些能力是对系统所属领域的鲜明反映.用户需求中记录的交互过程可能会体现该领域内普遍接受的业务流程或体现该系统具有特色的交互序列.功能需求中记录的功能则是构成系统的基本元素,是实现业务需求和用户需求的载体.

此外,在一个功能的执行过程中表现出的共性行为特点,以及同一功能在领域不同成员系统中表现出的特色行为特点也是体现领域特征的重要方面.在传统的针对单个应用系统的软件需求规约中,功能的行为特点是通过对功能的详细文字描述体现出来的.这种文字描述不能够承担捕获功能的共性和变化性的任务.因此,应该把这种行为特点作为特征显式地记录在领域特征模型中,从而更好地捕获和表现领域的共性和变化性.

本文把业务需求、用户需求、功能需求中所具有的特征分别称为服务(service)、用例(use case)、功能(function).这样,一个软件系统具有的各种特征可以表示为如图 2 所示的结构.服务、功能、行为特点通过 WPA 组织成层次结构.每一个服务涉及到的用例组织成一个用例集合.在系统的边界上,用户通过用例与系统发生交互,从而完成特定的业务需求.

基于以上分析,本文使用如图 3 所示的特征模型的具体形式.该特征模型除了记录系统具有的服务、功能、行为特点、用例等特征以外,还显示地记录了系统具有的质量特征以及特征之间存在的约束关系.服务、功能、行为特点 3 种特征通过 WPA 形成层次式结构.用例部分的特征通过特征间的依赖关系与服务、功能或行为特点特征建立联系.质量部分中的局部质量特征还记录了其可能影响到的服务、功能或行为特点.

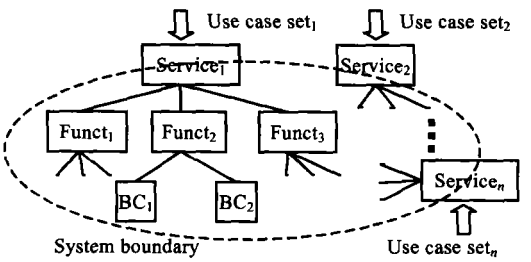

Fig.2 Functional features and their relation in a system

图 2 系统功能性特征的静态关系

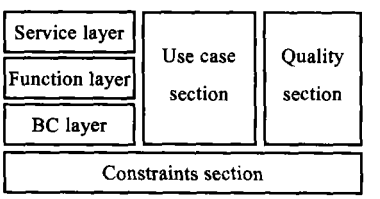

Fig.3 A concrete form of the DFM

图 3 特征模型的具体形式

2.4 特征建模过程

在上节所述的特征模型具体形式的基础上,本节给出对其进行建模的具体过程.考虑领域的服务、功能、行为特点等特征,以及与其相关的用例、质量需求特征,并注意分析领域的共性和变化性,同时把领域术语作为特征的重要来源[5],构成了如图 4 所示的特征模型建模过程.下面介绍此过程中每种活动的具体内容.

服务分析活动的主要工作是识别领域具有的服务特征.这要求领域分析人员通过与领域相关的组织机构的管理者、客户、市场销售人员及领域专家的交流,明确该领域具有的业务需求,为每一种典型的业务能力定义领域内统一的名称和说明,并将其作为特征模型服务层内的特征.在此活动中,可能会并发进行领域术语分析、共性变化性分析、交互过程分析、质量需求分析等活动.即通过对领域术语的分析,发现其中存在的服务;分析服务层特征的共性和变化性,识别服务之间存在的约束关系;对每一个服务分析与其相关的用例以及用例

的共性和变化性;分析服务具有的质量属性.

功能分析活动的主要工作是识别服务具有的功能特征.通过考察服务体现的业务能力以及与服务相关的用例,分析系统为完成特定的服务必须具有的功能,为每个功能定义领域内一致的名称和说明,并将其作为功能层内的特征,建立与服务层特征的整体部分关系.同样,在此项活动内仍然要并发进行领域术语分析、共性变化性分析等活动.当进行这些活动时,如果发现服务层特征的任何错误或不当,可以回溯到服务分析活动进行相应的修改和调整.

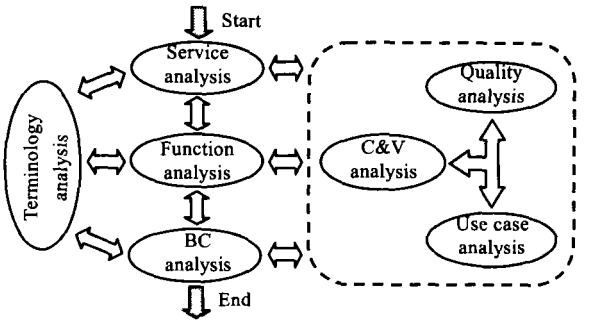

Fig.4 Feature modeling process

图 4 特征建模过程

行为特点分析活动的主要工作是识别功能具有的行为特点.具体包括:分析功能执行的前期行为特征,如功能执行的前置条件、前期准备工作等;分析功能主体行为的特点,发现其具有的显著特点和可能的变化性;分析功能的后期行为特点,如功能执行的后置条件、善后处理工作、功能执行完毕后的控制权转移等.为每一种行为特点建立领域内一致的名称和说明,并将其放入特征模型的行为特点层,建立与功能层特征的整体部分关系.如同服务分析和功能分析一样,在行为特点分析活动中也存在一系列的并发活动.在进行这些活动时,仍然可以回溯到前面的活动以便对可能的错误和不当做出修改和调整.

领域术语分析活动的主要工作是发现和领域术语相关的特征.领域术语反映了领域相关人员对领域的理解,是领域知识的一种沉淀.在一个成熟的领域内,人们可以通过领域术语方便地表达自己的思想,并互相交流.术语名称可以作为特征的候选名称.同时,领域术语分析也是对领域术语标准化的一个过程,特别是在一个不太成熟或存在同义、近义术语的领域内[5].

共性变化性分析活动的主要工作是识别领域的共性和变化性、分析特征之间可能存在的约束关系.要注意的是,特征之间的约束关系不只存在于同层的特征之间,不同层次、不同部分的特征之间也可能存在.特别是存在上层特征之间的约束关系,由于下层分析活动对上层特征的不断细化,可能会转化成跨层的约束关系或完全转化为下层特征之间的约束关系.每一个用例和其涉及的服务、功能、行为特点等特征之间也存在依赖关系.

交互过程分析活动的主要任务是发现在用户和系统的交互中存在的特征.在服务分析活动中,通过识别与服务相关的业务处理流程,提取出反映领域共性或具有特色的用例.随着功能分析、行为特点分析等活动的进行,交互过程分析会对已经识别的用例进行不断的细化并发现相关的变化性.通过交互过程分析形成的用例部分不是对软件系统用例模型的替代.特征模型用例部分更关注领域中具有共性的或具有特色的交互过程.在新系统对领域特征模型的定制过程中,可以对这些用例特征进行扩展和补充从而形成针对该系统的用例模型.

质量需求分析活动主要识别对领域可能的质量属性要求.伴随着服务、功能、行为特点等特征的识别过程,质量需求分析活动逐个考察这些特征可能具有的质量属性,以及对这些特征可能产生影响的质量属性.

3 实例研究

本节主要介绍面向特征的领域建模方法(FODM)在图元编辑器领域中的应用.图元编辑器是我们在开发领域分析支持工具的过程中标识出的一个功能领域.它是一个较简单和易于理解的领域,不需要太专业的知识背景.同时,它也是一个很成熟的领域,存在大量的以图元编辑器为基础的应用系统.因此,本文选择图元编辑器领域来示例 FODM 方法的应用.

3.1 图元编辑器领域

图元编辑器主要应用在计算机辅助建模(CAM)、计算机辅助设计(CAD)、用户界面可视化编程、通用演示系统等众多领域中.从用户角度来看,图元编辑器是一种允许用户直接在计算机屏幕上操纵具有特定语义或表示特定物理对象的图形符号(图元)的软件系统.例如,在支持 UML 的建模工具中,用矩形图元表示一个对象或

接口,用带有修饰符的线形图元表示对象或接口之间存在的关系.又如在电路设计领域中,一些特殊形状的图元可能表示特定的电子元件,而线形图元则代表电子元件之间的电路连接.用户可以随时操纵这些图元进行建模或设计活动.从功能角度来看,图元编辑器构成了这些领域的一个子领域,专门负责为用户提供直观、简便的操纵和显示机制.图元编辑器具有的直观性、虚拟性、易修改性等特点可以极大地提高其所在领域的建模或设计工作的效率.虽然在不同的专业领域中,图元表示的语义以及图元之间存在的约束关系有很大的不同,图元编辑器的具体行为也会体现出显著的差异性,但是,在对这些特定领域的变化性进行抽象、隔离和封装的基础上,我们可以得到一个具有很强共性的功能区域,即图元编辑器领域.

已经有机构和组织专门对图元编辑器的设计和实现进行了相关的研究工作,也已经存在了很多成熟的商用产品.本文中图元编辑器领域特征模型主要建立在对已有的研究 Unidraw[18],Hotdraw[19]以及两个开放源码软件项目 JHotDraw[20]和 GEF[21]的需求进行分析的基础上.

3.2 特征建模

首先,我们从分析图元编辑器对用户提供的主要服务和功能入手.

图元操纵服务是图元编辑器提供的核心服务.图元操纵服务通过控制和协调若干系统功能的执行,完成用户的图元操纵请求.考察用户对图元的典型操作可以发现,图元操纵服务进一步包含图元添加、图元选择、图元删除、图元移动、图元组合、层叠次序调整等系统功能.通过共性变化性分析,我们把图元组合、层叠次序调整作为两个可选的系统功能.通过交互过程分析,我们提取出对图元选择和图元移动存在的几种典型的操作方式.通过质量需求分析,我们发现对图元移动功能,可能存在响应速度、移动过程中视图的刷新机制、视图的闪烁等若干方面的非功能性需求.

视图缩放是图元编辑器提供的一种可选的服务.由于显示设备物理尺寸的限制或对局部细节的要求,用户可能需要缩小或放大视图以便能够观察到视图的整体布局或局部细节.

图元编辑器提供的另外一种可选服务是撤销/重做服务.撤销/重做服务通过记录系统状态的变化以及变化的可逆信息,为用户提供方便的系统状态回退和重演服务.对撤销/重做服务进行进一步的细化,可以发现它需要包含其他服务所涉及的系统功能对应的撤销/重做功能.通过共性变化性分析,可以发现对于可选的系统功能及其对应的撤销/重做功能存在较强的约束限制.例如,对图元组合、撤销/重做服务、图元组合撤销/重做,这三者之间存在这样的约束限制:(图元组合∧撤销/重作服务)↔图元组合撤销/重做.

在服务分析和功能分析的基础上,我们进一步对每个系统功能的行为特点进行了分析.对于图元选择,选择模式是它的一个显著的行为特点,且存在两种典型的选择模式:增量选择和非增量选择.通过共性变化性分析,我们发现,增量选择模式和图元选择操作方式之一的导航键方式之间还存在互斥的约束关系.同样,可以发现图元移动具有的移动模式、移动约束等行为特点及其变化性.

图 5 是本文采用 FODM 方法建造的图元编辑器领域特征模型.

需要指出的是,虽然特征模型存在服务、功能、行为特点这 3 个主要的层次,但特征建模的过程往往不是一个严格的顺序过程.可能在没有识别出本层存在的所有特征的情况下,就会针对某个特征进入到下一层的分析活动中去.因此,实际的建模过程是在这 3 个层次对应的分析活动之间不断迭代、反复进行的,在每一层次的分析活动中都会伴随着术语分析、变化性分析、交互过程分析、质量需求分析等并发活动.

4 结束语

本文介绍了一种面向特征的领域模型及其建模方法(FODM).在深入分析现有领域分析方法中使用的特征模型的基础上,对领域特征模型的组织框架从基本组织方式、变化性的表现方式、变化性的绑定时间、变化性的限制机制等方面进行了统一抽象的描述,特别是使用整体部分关系(WPA)作为组织特征的基本方式、采用部分相对整体的可选性以及维度和值的机制表现特征的变化性、使用由逻辑命题及逻辑运算符构成的公式记录特征间存在的约束关系.在需求工程中需求的 3 个层次的基础上,使用服务层、功能层、行为特点层作为特征模型的主体部分,用例部分和质量需求部分作为必要的补充部分,并对特征建模过程中涉及的各种活动及其相

互关系进行了详细讨论.

本文提出的特征模型框架能够兼容现有的几种重要的领域工程方法(FODA[4],FORM[5],FeatuRSEB[6],PLA[7])所使用的特征模型,且具有较好的结构性.本文提出的以命题逻辑为基础的变化性限制机制增强了传统Require-Mutex机制的表现能力,且对领域级及应用系统级特征模型的正确性检验均提供了有力的支持.本文提出的特征模型的具体形式建立在需求工程的研究成果之上,易于被传统需求分析人员所接受.但FODM方法还存在以下一些弱点:对领域范围的确定(domain scooping)缺乏基本的支持;对领域模型中其他模型成分及不同模型间的追踪性问题考虑不足;缺乏对特征及特征间动态交互关系的进一步分析.

进一步的工作包括在研究和实践中对FODM方法的不断完善、如何根据领域特征模型进行领域构件模型的设计、如何在设计阶段保持对领域变化性的隔离和封装等方面.对FODM方法支撑工具的开发也在进行中.

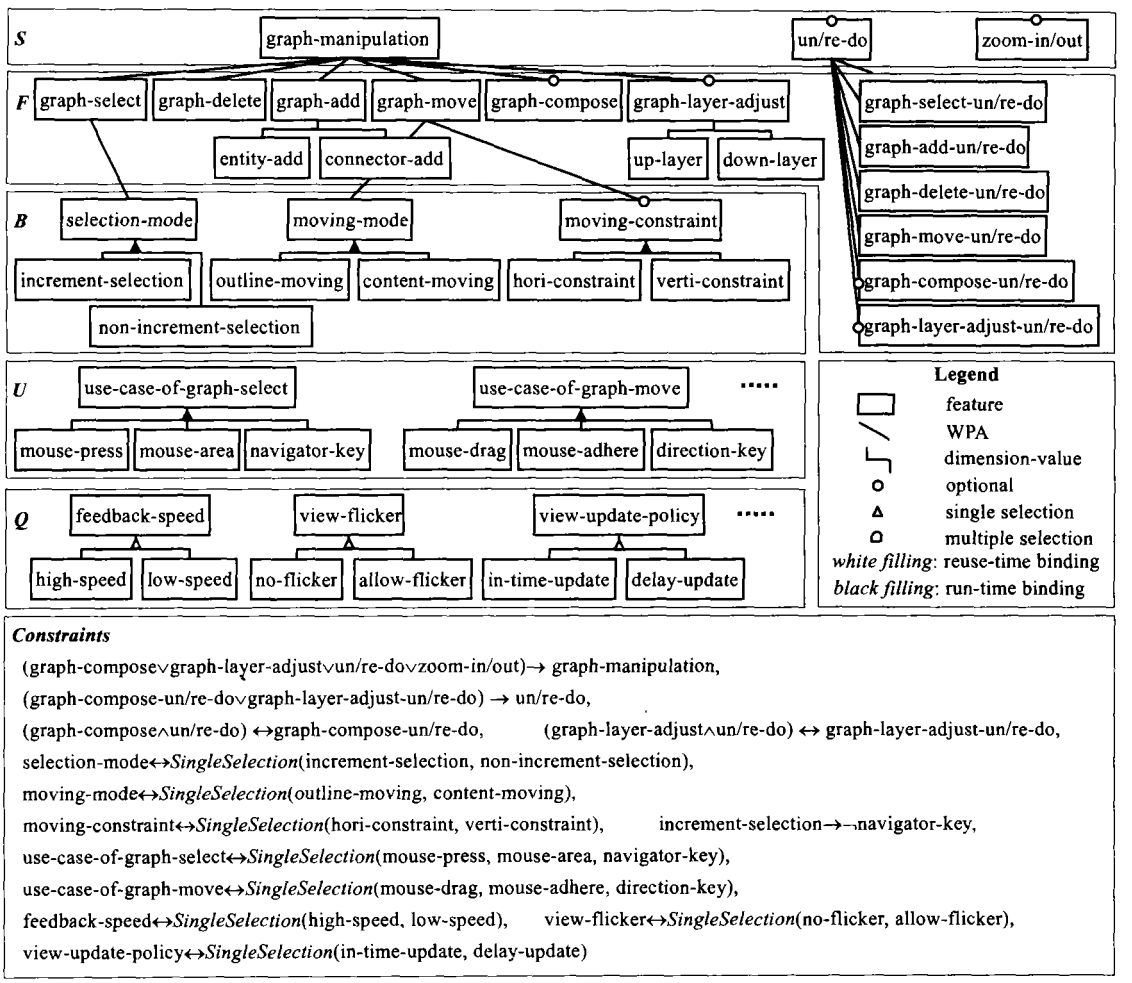

Fig.5 Domain feature model of Graph Editor

图5 图元编辑器领域特征模型

References:

[1] Li KQ, Chen ZL, Mei H, Yang FQ. An introduction to domain engineering. Computer Science, 1999,26(5):21~25 (in Chinese with English abstract).

[2] Karlsson EA. Software Reuse: A Holistic Approach. Chichester: John Wiley and Sons Ltd., 1995. x~xii.

[3] Davis AM. The design of a family of application-oriented requirements languages. Computer, 1982,15(5):21~28.

[4] Kang KC, Cohen SG, Hess JA, Novak WE, Peterson AS. Feature-Oriented domain analysis (FODA) feasibility study. Technical Report, CMU/SEI-90-TR-21. Pittsburgh: Carnegie Mellon University, Software Engineering Institute, 1990. 1~52.

[5] Kang KC, Kim S, Lee J, Kim K, Shin E, Huh M. FORM: A feature-oriented reuse method with domain-specific reference architectures. Annals of Software Engineering, 1998,5:143~168.

[6] Griss ML, Favaro J, d'Alessandro M. Integrating feature modeling with the RSEB. In: Devanbu P, Poulin J, eds. Proceedings of the 15th International Conference on Software Reuse. Victoria: IEEE Computer Society, 1998. 76~85.

[7] Chastek G, Donohoe P, Kang KC, Thiel S. Product line analysis: a practical introduction. Technical Report, CMU/SEI-2001-TR-001, Pittsburgh: Carnegie Mellon University, Software Engineering Institute, 2001. 1~42.

[8] Jacobson I, Christeron M, Jonsson P, Overgaard G. Object-Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley, 1992. 123~159.

[9] Keck DO, Kuehn PJ. The feature and service interaction problem in telecommunications software systems: a survey. IEEE Transactions on Software Engineering, 1998,24(10):779~796.

[10] Turner CR, Fuggetta A, Lavazza L, Wolf AL. A conceptual basis for feature engineering. Journal of Systems and Software, 1999,49(1):3~15.

[11] Nebulon Company. FDD Overview Presentation. http://www.nebulon.com/fdd/index.html.

[12] Mehta A, Heineman GT. Evolving legacy system features into fine-grained components. In: Proceedings of the 24th International Conference on Software Engineering. ACM, 2002. 417~427.

[13] Coplien J, Hoffman D, Weiss D. Commonality and variability in software engineering. IEEE Software, 1998,15(6):37~45.

[14] Wiegers KE. Software Requirements. 2nd ed., Buffalo: Microsoft Press, 1999. 3~22.

[15] Gamma E, Helm R, Johnson R, Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995. 1~29.

[16] Civello F. Roles for composite objects in object-oriented analysis and design. ACM SIGPLAN Notices, 1993,28(10):376~393.

[17] Geyer L. Feature modeling using design spaces. In: Knauber P, Pohl K, eds. Proceedings of the 1st German Workshop on Software Product Lines. Kaiserslautern: Fraunhofer IESE, 2000. 35~39.

[18] Vlissides JM, Linton MA. Unidraw: A framework for building domain-specific graphical editors. ACM Transactions on Information Systems, 1990,8(3):237~268.

[19] Johnson RE. Documenting frameworks using patterns. ACM SIGPLAN Notices, 1992,27(10):63~76.

[20] JHotDraw as Open-Source Project. http://www.jhotdraw.org.

[21] Java Graph Editing Framework. http://gef.tigris.org.

附中文参考文献:

[1] 李克勤,陈兆良,梅宏,杨芙清.领域工程概述.计算机科学,1999,26(5):21~25.