

具有层次结构的正交变化模型的研究

刘炜生, 杨贯中

(湖南大学 信息科学与工程学院, 长沙 410082)

摘要: 目前的正交变化模型存在对变化点和变体的定义不准确、无法描述复杂的层次结构关系以及无法自动地辨识领域中的共性和可变性的缺点。针对这些不足进行研究, 通过从时间和空间的角度去重新明确地定义变化点和变体之间的关系, 提出一种具有层次结构的正交变化模型。该模型具有良好的层次结构扩展能力, 能够应对复杂的变化性描述, 并且通过该模型可以明确地辨识出领域中具有共性和可变性的部分, 使自动计算成为可能。具有层次结构的正交变化模型为领域中相关计算提供了基础, 与以往的正交变化模型比较, 更接近变化性描述具有复杂性这一实际情况, 因此具有更实际的使用价值。

关键词: 软件产品线; 变化点; 变体; 可变性; 正交变化模型; 产品线工程; 领域工程

中图分类号: TP301 **文献标志码:** A **文章编号:** 1001-3695(2013)02-0476-04

doi:10.3969/j.issn.1001-3695.2013.02.043

Research on hierarchical orthogonal variability model

LIU Wei-sheng, YANG Guan-zhong

(College of Information Science & Engineering, Hunan University, Changsha 410082, China)

Abstract: There are some defects in the orthogonal variability model in recent research, including that the definition of the variation point and variant is not accurate, that it can't describe the relation of complex hierarchy of variation point and variant and have limit capability of identifying the commonality and variability in the domain automatically. As for these deficiencies, this paper proposed a hierarchical orthogonal variability model by explicitly defining and refining the relationship between variation point and variant from the perspective of the time and space. This model had good capability of extending its hierarchy, so it was able to react to the complexity in the description of variability. It could also identify the commonality and variability of the domain explicitly through this model, which made it possible for variability calculating automation. Hierarchical orthogonal variability model provided a basis for calculation in the domain. It is more close to the situation that the description of variability is complicated, and so it has more practical value, comparing to the original orthogonal variability model.

Key words: software product line; variation point; variant; variability; orthogonal variability model; product line engineering; domain engineering

领域工程是目前软件行业研究的热点之一, 面向领域的软件产品线工程已经成为继面向对象、面向组件编程之后, 实现软件复用的重要手段。面向领域的软件产品线工程, 其基础是根据领域中的各个产品实例, 找出领域中的共性和变化性, 而识别领域中各个产品实例的共性和可变性是领域分析阶段的主要产出之一, 它也是领域设计和领域实现等后续阶段的重要组成部分^[1]。因此保持可变性在领域工程各个阶段的一致性、可追溯性等十分重要。杨芙清等人^[2]提出以正交变化模型来描述领域中的可变性, 以领域架构作为共性、领域构件作为可变性的实现方法, 来设计和实现软件产品, 这在一定程度上解决了可变性的一致性和可追溯性等问题。

基于特征的领域分析(FODA)方法在一定程度上达到了对可变性进行描述的目的^[3], 但是将可变性直接融合于特征模型中会造成特征模型中的可变性与其他阶段的模型所描述的可变性信息冗余和可能出现的不一致, 虽然促进了一系列的领域设计方法的形成^[4], 但这些方法在体现可变性上并不完善, 包括: 太多的冗余信息^[5]; 描述的可变性在领域分析、领域设计和领域实现阶段难以保持一致性^[6]和可追溯性; 可变性

的维护性差。因此使用正交变化模型来描述领域中的可变性成为一种需要, 它不仅能够独立于其他模型来描述领域中的共性和可变性, 同时它能够与其他的模型“正交”产生新的模型, 如文献[3]中所提出的特征模型。

Pohl等人^[6]以变化点和变体的概念为基础提出了正交变化模型, 该模型描述了变化点、变体以及它们之间的各种关系, 包括变化依赖、工件依赖及约束依赖等。但该模型仍然存在一些比较明显的问题, 包括: 对于变化点和变体的定义不完整, 它们之间的关系描述缺乏或冗余; 该模型对于复杂的变化关系存在缺陷, 即无法描述复杂的可变性的层次结构。

1 正交变化模型

定义1 变化点是表示在时间或空间上可以发生变化的概念。这里的概念是软件产品线中包含了一系列功能、非功能且具有显著区别的属性的集合。

变化点用 $VP(p_1, p_2, \dots, p_n)$ 表示, 其中 p_i 表示变换点的某一项属性。例如“摄像头”这一变化点包括了“像素”“开启后到开始采集响应时间”“最大帧频”等属性。变化点包含了它

收稿日期: 2012-07-22; 修回日期: 2012-08-31

作者简介: 刘炜生(1987-), 男, 湖北恩施人, 硕士, 主要研究方向为软件工程(loplip@hnu.edu.cn); 杨贯中(1963-), 男, 湖南长沙人, 教授, 硕士, 主要研究方向为网络教育、计算机网络工程与分布式计算、多媒体通信、软件工程、领域工程。

所有变体的公共部分。

定义2 变体是某个时间点上继承了某一个变化点并从时间或者空间的角度去组成或者实现该变化点的概念。这里的概念是软件产品线中包含了一系列功能、非功能且具有显著区别的属性以及对应的值的集合。

变体可以用 $V(p_1 \text{ \$ values}, \dots, p_n \text{ \$ values})$ 表示。其中: p_i 表示某一项属性,如响应时间、最大帧频等;符号 \$ 表示属性与值(或者值列表)之间的关系,如响应时间小于 5 ms 中的“小于”,最大帧频等于 30 fps 中的“等于”以及风机输出压力范围从 4 cm 水柱到 20 cm 水柱中的“范围从...到...”均是代表属性与值之间的关系;values 表示属性能够取的值(或者值列表),如前述例子中的 30 fps、5 ms 等。

变体为零个或多个变化点的属性绑定确定的值的同时可以增加新的属性及其对应的值,因此它可以看做是特殊的变化点。例如“200 万像素,开启后到开始采集视频信息时间为 0.5 s,最大帧频 30 fps 的摄像头”属于变化点“摄像头”的低端实现的变体,而“1000 万像素,开启后到开始采集视频信息响应时间为 0.02 s,最大帧频 60 fps 的摄像头”属于同一变化点中的高端实现的变体。

1.1 结构关系

根据以上变化点和变体定义,定义如图1所示的仅包含结构关系的正交变化模型。

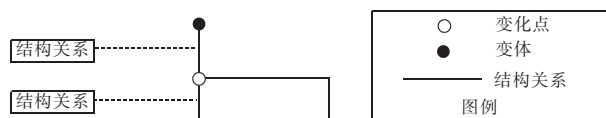


图1 正交变化模型结构关系图

结构关系由“——”表示,是一种上层对下层的关系,此处它是一种抽象的表示,实际上由 composed-of 和 implemented-by 两种关系组成。其中,composed-of 用“—◁”表示,表明该变体只是父节点变化点的一部分实现,是组成部分之一,两个及两个以上子节点才能共同包含父节点所有不可变及可变的属性和操作;implemented-by 用“——■”表示,表明该节点至少包含其父节点中所有不可变化的及可变化的属性和操作。由图1可知,这种上下层的结构关系是自包含的关系,而且通常情况下是一对多的关系,因此这种自包含的循环结构最终可以形成树型结构,弥补了文献[6]中正交变化模型无法形成多层次结构的缺点。由于领域中变化点和变体之间的关系特殊性,定义以下约束:

a) 最顶层节点一定是变化点(由空心圆圈表示),而且有且仅有一个,如“颜色”“电机”等。

b) 变化点可以包含一个或多个子节点,这些节点可以是变化点,也可以是变体,由具体的结构关系决定,如“颜色”可以包含“浅色”“深色”及“无色”,其中“无色”是一个变体,它们之间是一种 implemented-by 的关系,而“深色”和“浅色”则是子变化点,与“颜色”是 composed-of 的关系;它们也可能由其他的子变化点或变体组成,如“深色”还可能包含“黑色”“墨绿色”等。

c) 变体(由实心圆圈表示)继承自变化点,是一种特殊的变化点,因此它也可以包含在由变化点的自包含关系所形成的树型结构中。

d) 变体可以包含子节点,但此时作为父节点的变体与其所有的直接子节点(包括变化点子节点)只能是 composed-of

关系。

e) 变化点与变化点不能是实现关系。

f) 叶子节点在某一个特定的阶段可以是变化点,表明该节点需要在下一阶段的分析中进行精化,以达到最终满足所有叶子节点都是变体的形式。

由上述约束关系可以总结出如表1所示的变化点与变体的结构关系。

表1 节点关系表

序号	父节点	关系	子节点	图例	存在性
1	变化点	composed-of	变化点		存在
2	变化点	implemented-by	变化点	无	不存在
3	变化点	composed-of	变体		存在
4	变化点	implemented-by	变体		存在
5	变体	composed-of	变化点		存在
6	变体	implemented-by	变化点	无	不存在
7	变体	composed-of	变体		存在
8	变体	implemented-by	变体	无	不存在

值得注意的是变体作为父节点可以以 composed-of 的关系包含变化点子节点。例如 telephone(电话)和 Internet(互联网)是实现了 communication(通信方式)这个变化点的两个变体,针对 Internet(互联网)这个变体,它以 composed-of 关系包含一个 connection(连接方式)变化点,这个变化点可以是 TCP 变体,也可以是 UDP 变体,即最终通过 TCP 或者 UDP 的方式完成互联网连接的实现。

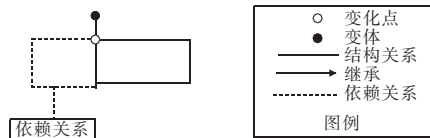
本文提到变体是一种特殊的变化点(详见定义2),不仅在结构上使得变体可以与变化点形成层次结构,而且在领域分析中也具有重要的意义。例如 telephone(电话)和 Internet(互联网)是实现了 communication(通信方式)这个变化点的两个变体,在需求分析阶段,telephone 节点是变体,因为此刻需求分析人员可能尚不明确具体 telephone 是否可变。到设计实现阶段,相关人员在确定要设计实现的 telephone 的具体网络规格时,需要确认是否需要支持多种网络规格(2G、3G 或者 4G 网络),因此可能产生 2G 制式手机、3G 制式手机和 4G 制式手机三个变体。这时,telephone 节点由变体变成了变化点,分别包含了三个 implemented-by 关系的三个实现变体,即变体可以向变化点转换。这种情况也适用于以 composed-of 包含子节点的变化点。

变化点也可能向变体转换。当变化点只包含一个实现变体(implemented-by 关系),且未来该变化点不再有其他子节点变体出现,那么该变化点能够被当前唯一的一个实现变体取代,变成变体。

1.2 依赖关系

依赖关系用“-----”表示,它代表一个节点对另外一个节点的依赖,它与结构关系一样,表示一种抽象的依赖,实际依赖关系分为排斥依赖(excludes)和需求依赖(requires)两种关系。其中排斥依赖表示两者不能同时存在,用“←—→”表示;需求依赖表示其中一方会使用到另外一方,用符号“——→”表示,箭头指向的一方表示被依赖的一方。虽然两种关系都可能是多对多的关系,但两者之间的关系只能是一对一的关系,即两者之间不能同时存在多个依赖关系。将上面的结构模型中

加入依赖,如图2所示。



根据节点类型的不同,可以将这种依赖关系分成三类:变化点与变化点的关系;变体与变体的关系;变化点和变体的关系。对于结构关系模型中的组成关系(composed-of),通过引入需求依赖和排斥依赖,可以表现FODA中复杂的结构关系,如必选、可选、多选一以及多选多关系^[3,5],这种隔离关注点的方式使得各种关系变得单一;通过单一关系的组合来表现复杂的关系,可以保证关系的原子准确性并且避免了关系的冗余表达。由于FODA中专注于特征之间的关系,因此在本模型描述中,只提取变化部分,不考虑特征,如表2所示。

表2 FODA特征关系与可变关系映射表

序号	FODA中的定义	本文中的表现方式
1	必选	或者 或者 或者 或者 文献[5]提出了一种基于关系的捕获共性和可变性的算法,该算法同样适用于本文中的正交变化模型,但文献[5]中使用refine关系所捕获的共性并不完整。在产品线工程中,定义共性为:该产品线所涉及的领域中所有已知的产品以及未来可能出现的产品中都必须拥有的特征。因此,某个变化点的某个完整的实现变体(具有implemented-by关系的子节点)可能是所有产品必选的,如所有的台式电脑都必须支持以太网的有线连接的方式连接互联网。基于这种考虑,共性可以抽象为某一个变化点或变体的所有后代节点的集合及与该变化点或者变体有直接或者间接requires依赖的节点集合的交集,后代节点不仅包含组成关系的节点,还应当包含实现关系的节点。

2 基于正交变化模型的变化性计算

定义3 领域变化模型是指某一阶段的模型与正交变化模型结合,描述该阶段中所有共性与可变性的模型,其中包含变化点、变体以及它们之间的相互关系。

例如特征模型与正交变化模型结合、设计模型与正交变化模型结合等。领域变化模型分为可变模型和通用性模型。

定义4 可变模型是描述某一个领域或者产品线工程中产品实例间具有差异部分的模型。

定义5 通用性模型是描述某一个领域或者产品线工程中所有产品实例公共部分的模型。

2.1 案例分析

按照前述共性的抽取方法,从领域变化模型的根节点计算可得该领域的共性部分,即通用性模型,该部分可用于形成领域参考架构^[8],本文不作重点阐述,可参看文献[5];提取共性之后的剩余部分即可变模型。下面以一个家庭集成系统(HIS)^[7]为例,比较基于正交变化模型描述方法在表现领域可变性时的优势,以及抽取出的可变模型的应用场景。由于本

文着重于层次结构对于描述变化性时的完备性比较,因此不作正交相关的分析和描述。

如图3所示,该模型由特征模型^[7]与正交变化模型结合产生,描述了HIS领域特征中的共性和可变性。

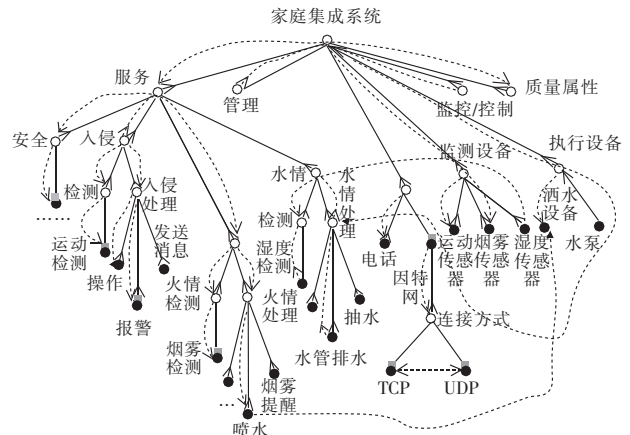


图3 家庭集成系统的领域变化模型

该模型只抽取了家庭集成系统的部分具有代表性的特征作为分析和比较的依据。

a) HIS节点作为顶层变化点,是该模型中唯一的根节点;在第二层中的服务、管理、通信、检测设备、执行设备等作为组成该变化点的子变化点,它们分别提供了该变化点的一个方面的实现。

b) 根据上面定义的模型,就“服务”变化点作单一结构分析,循环的结构关系使其可以作为其他变体或者变化点的父节点。如图3所示,服务包括安全、入侵、火情和水情,这四个变化点体现了服务的四个组成方面,即安全、入侵、火情和水情。在上面定义的模型中,结构关系实际包含了四种关系,即必选关系、可选关系、多选一关系和多选多关系。从特征图中可知,水情是可选关系,另外三种均是必选关系。正交变化模型的关系在HIS的特征图中由三种关系组成,包括组合关系(composed-of relationship)、泛化关系(generalization relationship)和实现关系(implemented-by relationship)。

c) “连接方式”变化点(家庭集成系统—通信—互联网—连接方式)包含两个变体,分别为TCP和UDP,这两个变体是多选一的关系,即在TCP和UDP两种协议中选一个来实现“连接方式”变化点。

d) 递归的,这四种服务的变体又包含了“检测”和“处理方式”两个变体以及下面的子变体,如喷水、烟雾提醒及发送消息等。

e) 在该特征图中,没有明显的包含特征之间的依赖关系,原文中在特征图的左下角,用文本表示了特征之间的依赖,描述如下:

- (a) Water requires Sprinkler.
- (b) Flood requires Moisture sensor.
- (c) Pumping requires Sump pump.
- (d) Message requires Communication.

在本文定义的模型中,依赖关系用虚线表示,其与结构关系相同,循环的结构特征使得这种依赖关系实际由三类关系组成,即变体与变体的依赖、变体与变化点的依赖、变化点与变化点的依赖。特征图中,Water requires Sprinkler体现的是变体与变体之间的依赖;Flood requires Moisture sensor体现的是变化点与变体之间的依赖。很明显“处理方式”变化点依赖于“处理设备”变化点,此为变化点与变化点之间的依赖。

2.2 通用性推导

根据前述共性定义:某一个变化点或变体的所有后代节点的集合及与该变化点或者变体有直接或者间接 requires 依赖的节点集合的交集,将上述 HIS 的变化模型中的共性模型抽取出来如图4所示,灰色区域中包含的即是 HIS 领域中的共性(包括 HIS 节点本身)。

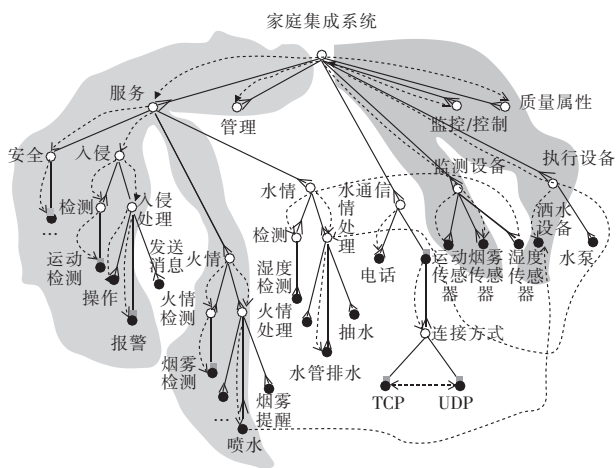


图4 家庭集成系统的共性抽取

将剩余部分去掉后,得到如图5所示的共性模型。该模型代表了 HIS 产品线中,所有单一产品实例必须包含的元素。

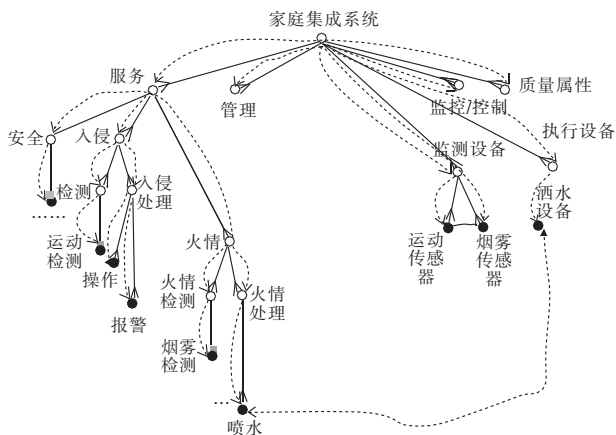


图5 家庭集成系统通用性模型

2.3 可变性计算

将通用性模型剔除后的剩余部分即某个领域的可变模型, HIS 的可变模型如图6所示。

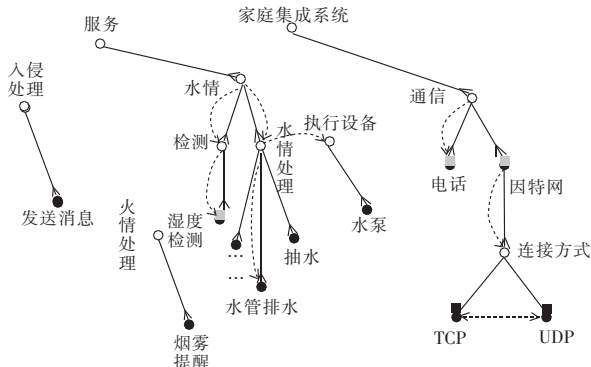


图6 家庭集成系统的可变模型

由图6可以看出,该模型并不具有单一根节点,除去节点之间直接纯粹的依赖关系不考虑,这个模型可以看成是由多个具有独立根节点的树组成的“森林”。考虑到正交变化模型的

本质是一个或者多个可变的“正交”,将每一个节点及其相关联的所有节点以及它们的关系组成称为一个“树包”,每一个“树包”体现了计算可变性的一个方面,零个或多个树包与共性模型组合,再进行正交变化组合,就可能得到一个该领域或者该产品线的产品实例。

假设树包 i 存在的变化的数目为 n ,用 $C_i(n)$ 表示,则整个领域或者产品线中可能存在的产品的实例数量 PPN (possible product number) 可以由如下公式计算所得:

$$PPN = \prod_{i=1}^n C_i(n)$$

由于树包的结构通常已经相当简化, $C_i(n)$ 中的 n 值很容易计算出来,对于只有两层的简单结构关系,计算结果如表3所示。即使存在一定的复杂性,因为该模型是一种递归的结构关系,因此也可以通过前述相同的方法进一步地“拆解”达到简化、计算的目的^[9]。公式中 i 和 $C_i(n)$ 中的 i 具有相同的含义,表示第 i 个“树包”, n 表示领域中的树包的总数。

表3 简单关系的实例数计算

序号	关系	可能存在的实例数
1	可选关系	2
2	多选一关系	n, n 代表互斥的节点数
3	多选多的关系(至少选一个)	$2n-1, n$ 代表多选关系节点数

3 结束语

具有层次结构的正交变化模型解决了目前的正交变化模型在定义变化点和变体时,无法清晰描述多层次变化点和变体之间关系的问题,这对于解决复杂的领域可变性问题具有十分重要的意义。在该模型中,对变化点和变体具有明确的定义,但是变化点和变体在不同的阶段可以相互转换,以达到弥补前一阶段分析不足的缺陷,这也符合随着精化的不断深入,越来越多的外部变化点转变为内部变化点的规律^[6]。该模型基于已有的共性自动推理的算法,提出了一种计算产品线变化性的方法,通过该方法可以更好地为应用工程提供可信的支持。目前基于该模型的变化性计算尚未考虑如“变量打包”等^[6]复杂情况,因此在对应用工程的支持上仍存在一定的限制。

参考文献:

- [1] MOON M, YEOM K, CHAE H S. An approach to developing domain requirements as a core asset based on commonality and variability analysis in a product line [J]. IEEE Trans on Software Engineering, 2005, 7(31): 551-569.
- [2] 杨笑清, 梅宏. 构件化软件设计与实现 [M]. 北京: 清华大学出版社, 2008.
- [3] KANG K C, COHEN S G, HESS J A, et al. Feature oriented domain analysis (FODA) feasibility study, CMU/SEI-90-TR-021 [R]. Pittsburgh: Software Engineering Institute, Carnegie Mellon University, 1990.
- [4] KANG K C, KIM S J, LEE J, et al. FORM: a feature-oriented reuse method with domain-specific reference architectures [J]. Annals of Software Engineering, 1998, 5: 143-168.
- [5] FEY D, FAJTA R, BOROS A. Feature modeling: a meta-model to enhance usability and usefulness [C] // Proc of the 2nd International Conference on Software Product Line. London: Springer-Verlag, 2002: 198-216.
- [6] POHL K, BOCKLE G, Van der LINDEN F. Software product line engineering: foundations, principles and techniques [M]. Berlin: Springer, 2005.
- [7] KANG K C, LEE J, DONOHUE P. Feature-oriented product line engineering [J]. IEEE Software, 2002, 19(4): 58-65.
- [8] 李林斐. 基于设计模式的构建可变性实现 [D]. 重庆: 重庆大学, 2003.
- [9] Van DEURSEN A, KLINT P. Domain-specific language design requires feature descriptions [J]. Journal of Computing and Information Technology, 2001, 10(1): 1-17.