

面向过程的需求建模

张国生

(云南大学 信息学院, 云南 昆明 650091)

摘要:提出目标、角色、需求任务建模方法,将软件系统高层目标层层分解、精化成操作化目标,分配给角色作为角色目标,在需求工程过程中,用具体的需求任务实现角色目标。定义了3个单一化规则对建模过程进行约束,可以避免操作化目标、角色和具体的需求任务之间的冲突。定义了操作化目标与角色目标之间的映射函数、角色目标与具体的需求任务之间的映射函数以及它们的复合函数,将操作化目标与具体的需求任务直接关联,使软件系统目标与需求工程过程紧密结合,可以更加直观、形象、准确地对系统需求建模,通过聚集需求任务实现系统高层目标。

关键词:目标;角色;需求任务;分解;精化;单一化规则;操作化目标

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2013)08-0005-04

doi:10.3969/j.issn.1673-629X.2013.08.0.02

Process-oriented Requirements Modeling

ZHANG Guo-sheng

(College of Information, Yunnan University, Kunming 650091, China)

Abstract: Goal, role and requirements task modeling methodology is presented. The software system high level goal is hierarchically decomposed and refined as operational goals. The operational goals are assigned to roles as their goals. The role goals are achieved by concrete requirements tasks in the requirements engineering processes. It defines three singleton rules which constraint modeling processes and avoid conflicts of the operational goals, the roles and the concrete requirements tasks. The function maps the operational goals to the role goals, the function maps the role goals to the concrete requirements tasks and their compound functions are directly related the operational goals to the concrete requirements tasks. The approach integrates software system goals and requirements engineering processes. The system requirements are modeled more intuitionistically, visually and exactly. The requirements tasks are aggregated to achieve the system high level goal.

Key words: goal; role; requirements task; decomposition; refinement; singleton rule; operational goal

0 引言

随着信息技术的不断发展,软件系统的规模不断扩大,软件所能实现的功能日益复杂,软件及其运行环境与外界的交互方式和交互场景变得越来越不可预测,需求工程是软件工程的一个分支,主要关注软件系统对现实世界的目标、功能、约束的体现。如何获取各涉众方的“真实”需求是需求工程非常重要的方面,面向目标的需求工程方法认为,需求阶段的主要任务是要确定软件系统需求相关者想要实现的各项目标,建立实现这些目标所需要的服务和约束条件的规格说明。

文献[1]认为需求工程不能仅仅局限在软件工程为需求阶段设定的目标上,更强调要用一种自然的方

式,去系统地寻找和发现正确的和潜在的用户需求,为这些需求建立适当的模型,并在对模型进行正确性分析和有效性验证的基础上,为软件开发的后续阶段提供合理、正确的软件规格说明。

文献[2]提出基于自动规约的需求获取方法KAOS,以目标为中心来获取需求规约,支持从由时序逻辑表示的高层目标生成候选系统体系结构设计。文献[3]提出了面向目标和过程分析的非功能需求框架,支持对候选系统设计方案进行评估和比较,比较的标准是各候选方案对非功能需求目标的贡献大小。文献[4]用目标驱动软件设计过程。文献[5]用面向过程的视图将组织功能分解为任务、过程、资源类型和资源,用形式语言表示它们之间的关系,任务被分解成更

收稿日期:2012-10-23

修回日期:2013-01-26

网络出版时间:2013-04-22

基金项目:云南省教育科学基金项目(2012C106);云南大学重点教改项目(WX070142)

作者简介:张国生(1968-),男,硕士,副教授,研究方向为软件工程及其形式化。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20130422.1722.034.html>

加具体的层次关系。

文献[6]认为,构建一个软件系统最困难的部分是精确地决定构建什么,因此,软件构建者为用户实现的最重要功能是产品需求的迭代抽取和精化。

文献[7]提出要得到尽可能准确、一致地反映客户需求的需求规格文档,整个需求过程的活动需要不断地反馈、增量、迭代、演化,需求过程的每一个活动都与敏捷方法密切相关,构成完整的需求过程生命周期,每一个需求过程活动也需要借助于敏捷方法不断地反馈、增量、迭代、演化,因此,每一个需求过程活动也构成一个子周期。

在开发软件系统时,引入工程化原理的良定的过程有助于及时交付高质量的产品,要开发高质量的软件需求规格说明,也需要可遵循的良定过程作为指导,用一个系统化和严谨有序的过程,来管理并控制软件需求规格说明的开发进程。文中提出将软件系统按照系统目标分解、精化产生目标分解树,将树叶目标节点即操作化目标分配给角色作为角色目标,用需求任务实现角色目标,将系统目标与需求工程过程紧密结合,从而获得准确、全面、可靠的系统需求。软件过程模型描述了软件过程要素(如:活动、资源、角色、过程产品等)以及这些要素之间的关系^[8]。

1 软件系统目标

目标定义为系统想要达到的状态或条件,是系统设计意图的一种说明和陈述,隐含地表达了期望系统所体现出的行为以及要满足的约束^[9]。目标可以分为功能性目标和非功能性目标,功能性目标又称硬目标,描述要实现的服务,是需求相关者期望发生的所有场景的集合,要求所表达的内容是清晰一致的。非功能性目标描述对服务质量的偏好,如良好的保密性、较高的安全性、较强的准确性、较好的易用性等,也可以描述对开发过程质量的期望,如良好的适应性等,非功能性需求用软目标表示,软目标逐步分解为子目标,直到获得可度量、可判断、可操作的目标。

1.1 目标分解树

在需求目标树结构中,将一个高层的、抽象的、粗略的目标分解为一组相对低层的、具体的和细化的子目标,系统目标被不断分解,直到树叶节点子目标为可操作化目标,从而生成系统目标分解树。子目标之间可以是“与”关系,也可以是“或”关系,当子目标之间是“与”关系时,所有子目标被满足,父目标才满足;当子目标之间是“或”关系时,只要子目标之一被满足,父目标即可被满足。

定义 1: 目标分解树 $\text{Goal_Tree} = (G_ID, G_Description, G_Type, G_Relation)$, 其中:

G_ID 表示目标标识;

$G_Description$ 表示目标描述,包括目标属性集和目标状态集;

G_Type 表示目标类型:硬目标、软目标和操作化目标, $G_Type = (HardGoal, SoftGoal, OperationGoal)$;

$G_Relation$ 表示目标之间的关系:“与”关系、“或”关系和无关系, $G_Relation = (And, Or, None)$ 。

软件系统目标分解树是一个递归定义,高层目标节点被分解成下一层子目标节点,子目标节点可以继续分解,直到树叶节点为具体的可操作化目标。

1.2 操作化目标

操作化目标是对硬目标和软目标进行分解和求精的结果,将出现在软件设计与实现中,操作化目标是目标分解树中靠近底层叶节点的目标^[10],用于表示满足高层目标的具体设计,提供目标系统中的操作、过程、数据表示、结构、约束等。操作表示输入和输出之间的关系,是输入状态到输出状态的转换,其数学含义是作用于对象集合之上的关系,通过前置条件、后置条件和触发条件来定义,其中,前置条件是操作执行的起始状态需要满足的最弱必要条件,后置条件是操作执行的终止状态需要满足的最强条件,触发条件是操作执行的起始状态需要满足的最弱充分条件。这些条件被区分为两大类:领域前置和后置条件、需求前置和后置条件,前一类用于描述操作所引发的领域中的基本状态迁移,后一类用于描述该操作为确保需求的满足要引发的额外状态迁移。

定义 2: 操作化目标 $\text{OperationGoal} = (OG_ID, Inputs, Outputs, PreCondition, PostCondition, TriggerCondition)$, 其中:

OG_ID 表示操作化目标标识;

$Inputs$ 表示操作化目标输入;

$Outputs$ 表示操作化目标输出;

$PreCondition$ 表示操作化目标前置条件;

$PostCondition$ 表示操作化目标后置条件;

$TriggerCondition$ 表示操作化目标触发条件。

1.3 目标精化

目标精化主要是通过对目标子树的根节点提问“为什么”来获取更高抽象层次的目标,诱导出关于环境和上下文的相关信息;通过对目标子树的叶节点提问“怎么做”来获取更低抽象层次的具体目标,诱导出关于操作方式的相关信息;通过对目标分解树中的操作化目标提问“怎样用其他方法做”来获取其他可能的候选方案——需求任务。

设一组目标 G_1, G_2, \dots, G_n 是目标 G 的完全精化,当且仅当以下条件满足:

a) $G_1, G_2, \dots, G_n \vdash G$ (必要性);

- b) $G_1, G_2, \dots, G_n \not\models \text{False}$ (一致性);
 c) $n > 1$ (非平凡性);
 d) $\forall i, 1 < i < n, G_1, \dots, G_{i-1}, G_{i+1}, \dots, G_n \not\models G$ (最小性)。

2 角色

角色是成员抽象行为的描述,在特定的组织环境下,对组织中的部分行为和责任进行抽象刻画,成员抽象成为角色,每个成员与一定的角色相关联^[11]。角色可以由一到多个成员扮演,每个成员也可以扮演一到多个角色,是实现系统目标的实体,角色具有与不同方面的任务相关的权限和责任。

定义3:角色 $\text{Role} = (\text{R_ID}, \text{R_Goal}, \text{R_Attributes}, \text{R_Relation}, \text{R_Plans}, \text{R_Actions}, \text{R_Permissions}, \text{R_Protocols})$, 其中:

R_ID 表示角色标识;

R_Goal 表示角色目标,是一个2元组 $(\text{R_Gs}, \text{R_Gc})$, R_Gs 是目标描述; R_Gc 是目标约束,它是一个资源约束;

R_Attributes 表示角色所具有的属性;

R_Relation 表示角色之间的关系,由关系约束和关联角色2元组 $(\text{Rel_Constraint}, \text{Rel_Role})$ 描述, $\text{Rel_Constraint} = \{\text{comm}, \text{coop}\}$, 分别表示角色之间的命令关系 comm 和协作关系 coop ;

R_Plans 表示角色规划;

R_Actions 表示角色行为;

R_Permissions 表示角色权限;

R_Protocols 表示角色协议。

3 需求任务及其功能

3.1 需求任务

软件系统需求工程过程由需求活动组成,每个需求活动又包含多个需求任务,每个需求任务实现一个具体的功能,所有需求任务组合起来完成系统总的目标,需求任务是需求工程过程中的基本功能单位^[12]。

定义4:需求任务定义为一个4元组 $\text{RT} = (\text{Input}, \text{Output}, \text{Pre}, \text{Post})$, 其中:

a) Input 和 Output 分别表示需求任务的输入和输出;

b) Pre 和 Post 为一阶谓词公式,分别表示执行需求任务 RT 所需的前置条件和后置条件。

3.2 需求任务功能

定义5:需求任务功能定义为一个4元组 $\text{RTF} = (D, R, \text{Pre}(\mathbf{X}), \text{Post}(\mathbf{X}, \mathbf{Y}))$, 其中

a) $\mathbf{X} = (x_1, x_2, \dots, x_m)$ 为输入向量, $\mathbf{Y} = (y_1, y_2, \dots, y_n)$ 为输出向量, \mathbf{X} 和 \mathbf{Y} 中的元素称为变量, $\{\mathbf{X}\} = \{x_1,$

$x_2, \dots, x_m\}$ 和 $\{\mathbf{Y}\} = \{y_1, y_2, \dots, y_n\}$ 分别表示输入变量集和输出变量集;

b) $D = D_1 \times \dots \times D_m$ 表示输入向量的定义域,且 $x_i \in D_i (1 \leq i \leq m)$; $R = R_1 \times \dots \times R_n$ 表示输出向量的值域,且 $y_j \in R_j (1 \leq j \leq n)$;

c) $\text{Pre}(\mathbf{X})$ 和 $\text{Post}(\mathbf{X}, \mathbf{Y})$ 为一阶谓词公式,分别表示需求任务的前置条件和后置条件;

d) 满足前置条件 $\text{Pre}(\mathbf{X})$ 的输入向量 \mathbf{X} 称为合法的输入;对于合法的输入 \mathbf{X} ,需求任务将产生满足后置条件 $\text{Post}(\mathbf{X}, \mathbf{Y})$ 的合法输出 \mathbf{Y} ;

e) $A(F) = (\text{Pre}(\mathbf{X}), \text{Post}(\mathbf{X}, \mathbf{Y}))$ 称为需求任务的功能 RTF , $A(F)$ 的执行表示一个满足前置条件 $\text{Pre}(\mathbf{X})$ 的输入向量 \mathbf{X} ,如果 $A(F)$ 执行终止,将产生满足后置条件 $\text{Post}(\mathbf{X}, \mathbf{Y})$ 的输出 \mathbf{Y} 。

4 GRT 建模

目标分解树将软件系统目标不断分解、精化为操作化目标,并将操作化目标分配给相应的角色作为角色目标,由角色在具体的需求任务中实现功能目标,将目标、角色、需求任务组合,共同实现目标、角色、任务 GRT 建模,可以更加直观、形象、准确地对软件系统需求建模,目标、角色、任务之间的关系如图1所示。

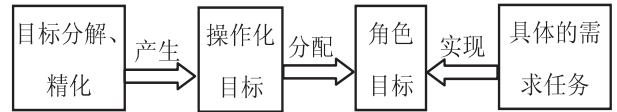


图1 GRT 建模示意

4.1 GRT 建模规则

在 GRT 建模中,充分融合了目标、角色和具体的需求任务,建模时需要考虑:如何将操作化目标分配给角色作为角色目标,如何用具体的需求任务实现操作化目标,以及将具体的需求任务指派给角色来承担等。从软件工程过程建模角度,定义了3个单一化规则对建模过程进行约束。

设 GOAL 、 ROLE 、 TASK 分别表示操作化目标集合、角色集合、具体的需求任务集合,二元谓词 $\text{ASSIGN}(G_i, R_j)$ 表示将操作化目标 G_i 分配给角色 R_j ;二元谓词 $\text{ACHIEVE}(T_i, G_j)$ 表示用具体的需求任务 T_i 实现目标 G_j ;二元谓词 $\text{UNDERTAKE}(T_i, R_j)$ 表示具体的需求任务 T_i 由唯一一个角色 R_j 承担。

规则1(目标单一化规则):任何一个操作化目标只能分配给一个角色,即

$\forall G_i, 1 < i < n, G_i \in \text{GOAL}, \exists R_1 \in \text{ROLE}, \text{ASSIGN}(G_i, R_1) \wedge \exists R_2 \in \text{ROLE}, \text{ASSIGN}(G_i, R_2) \rightarrow R_1 = R_2$

规则2(任务单一化规则):任何一个操作化目标一定存在唯一一个具体的需求任务实现,即

$\forall G_i, 1 < i < n, G_i \in \text{GOAL}, \exists T_1 \in \text{TASK}, \text{ACHIEVE}(T_1, G_i) \wedge \exists T_2 \in \text{TASK}, \text{ACHIEVE}(T_2, G_i) \rightarrow T_1 = T_2$

规则 3 (角色单一化规则): 任何一个具体的需求任务只能由唯一一个角色承担, 即

$\forall T_i, 1 < i < n, T_i \in \text{TASK}, \exists R_1 \in \text{ROLE}, \text{UNDERTAKE}(T_i, R_1) \wedge \exists R_2 \in \text{ROLE}, \text{UNDERTAKE}(T_i, R_2) \rightarrow R_1 = R_2$

规则 1 使得操作化目标和角色之间为一一对应关系, 规则 2 使得操作化目标和需求任务之间为一一对应关系, 规则 3 使得需求任务和角色之间为一一对应关系。3 个单一化规则可以避免操作化目标、角色和具体的需求任务之间的冲突。

4.2 GRT 映射函数

在建模过程中, 首先对软件系统高层目标进行分解生成目标分解树, 将目标分解树的树叶节点即操作化目标分配给相应的角色作为角色目标, 然后, 用具体的需求任务实现角色目标。函数 G_Assign 和 $G_Achieve$ 定义如下:

$G_Assign: \text{GOAL} \rightarrow \text{ROLE}, R_Goal$

$G_Achieve: \text{ROLE}, R_Goal \rightarrow \text{TASK}$

函数 $G_Assign(G_i)$ 返回与角色 R_i 相关的角色目标 R_i, R_Goal , 函数 $G_Achieve(R_i, R_Goal)$ 返回与角色 R_i 的目标相关的具体的需求任务 T_i , 而复合函数 $G_Achieve(G_Assign(G_i))$ 则返回操作化目标 G_i 所对应的具体的需求任务 T_i , 将操作化目标与具体的需求任务直接相关联, 使软件系统目标与需求工程过程紧密结合, 通过聚集需求过程任务实现软件系统高层目标。

5 结束语

面向过程的需求建模, 将软件系统高层目标层层分解、精化成操作化目标, 分配给角色作为角色目标, 然后, 用具体的需求任务实现角色目标。通过角色、需求任务、需求任务功能建立与操作化目标之间的映射关系, 定义了单一化规则和 GRT 映射函数, 将操作化

目标与具体的需求任务直接相关联, 使软件系统目标与需求工程过程紧密结合, 聚集需求过程任务实现软件系统高层目标, GRT 建模方法将目标、角色、需求任务组合, 可以更加直观、形象、准确地对软件系统需求建模。

参考文献:

- [1] 张国生. 需求演化过程建模[J]. 微电子学与计算机, 2012, 29(5): 54-57.
- [2] Dardenne A, Lamsweerde A V, Fickas S. Goal-directed Requirements Acquisition [J]. Science of Computer Programming, 1993, 20(1-2): 3-50.
- [3] Chung L, Nixon B A, Yu E, et al. Non-functional Requirements in Software Engineering[M]. [s. l.]: Kluwer Academic Publishers, 2000.
- [4] Feather M S, Fickas S. A Framework for Distributed System Designs[C]//Proc. of Knowledge-based Software Engineering Conference. Monterey: [s. n.], 1994: 6-13.
- [5] Popova V, Sharpanskykh A. A Formal Framework for Modeling and Analysis of Organizations[J]. IFIP International Federation for Information Processing, 2007, 244: 343-358.
- [6] Brooks F P. No Silver Bullet: Essence and Accidents of Software Engineering[J]. IEEE Computer, 1987, 20(4): 10-19.
- [7] 张国生. 敏捷需求过程建模[J]. 微电子学与计算机, 2012, 29(2): 27-30.
- [8] 赵欣培, 李明树, 王青, 等. 一种基于 Agent 的自适应软件过程模型[J]. 软件学报, 2004, 15(3): 348-359.
- [9] 金芝, 刘璘, 金英. 软件需求工程: 原理和方法[M]. 北京: 科学出版社, 2008.
- [10] Lamsweerde A V. Goal-oriented Requirements Engineering: A Guided Tour[C]//Proc. of Fifth IEEE International Symposium on Requirements Engineering. Toronto: [s. n.], 2001: 249-263.
- [11] Xu H, Zhang X. A Methodology for Role-based Modeling of Open Multi-agent Software Systems[C]//Proceedings of the 7th International Conference on Enterprise Information Systems (ICEIS 2005). Miami, Florida: [s. n.], 2005: 246-253.
- [12] 张国生. 基于层次着色 Petri 网的需求工程过程框架[J]. 计算机应用与软件, 2011, 28(8): 17-19.

(上接第 4 页)

- [5] 凌玲, 卢文, 胡于进. 不规则区域矩形件排样的一种改进算法[J]. 微型机与应用, 2011, 28(9): 112-115.
- [6] 张德富, 韩水华, 叶卫国. 求解矩形 Packing 问题的砌墙式启发式算法[J]. 计算机学报, 2008, 31(3): 509-515.
- [7] 刘嘉敏, 佟德刚, 黄有群. 临界多边形生成算法的改进[J]. 沈阳工业大学学报, 2005, 27(5): 567-570.
- [8] 黎自强, 滕弘飞. 一种新的凸多边形不干涉算法[J]. 计算机工程与应用, 2008, 44(1): 11-13.
- [9] Dowsland K A, Vaid S, Dowsland W B. An algorithm for poly-

gon placement using a bottom-left strategy [J]. European Journal of Operational Research, 2002, 141(2): 371-381.

- [10] Bennell J A, Dowsland K A, Dowsland W B. The irregular cutting-stock problem—a new procedure for deriving the no-fit polygon[J]. Computers & Operations Research, 2001, 28(3): 271-287.
- [11] Liu D, Teng H. An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles [J]. European Journal of Operational Research, 1999, 112(2): 413-420.

面向过程的需求建模

作者：[张国生](#)，[ZHANG Guo-sheng](#)
作者单位：[云南大学 信息学院, 云南 昆明, 650091](#)
刊名：[计算机技术与发展](#)

英文刊名：[Computer Technology and Development](#)

ISTIC

年，卷(期)：2013(8)

本文链接：http://d.wanfangdata.com.cn/Periodical_wjfz201308002.aspx