

从面向对象到面向目标的需求分析

From Object-Oriented to Goal-Oriented Requirements Analysis

程 勇¹ Cai Zhiming² 袁兆山¹

(合肥工业大学计算机与信息学院 合肥230009)¹

(Knowledge Management Lab, University of Toronto, Toronto, Ontario, Canada)²

Abstract Software requirements are the basis of software design, test and maintenance. The trend that future software should be intelligent and distributed and the changing software environment challenge the Object-Oriented methodology that is a primary technology to build software system currently. Goal-Oriented requirements analysis is discussed in detail in this paper. Goals are more powerful logical mechanism to identify, organize and justify software requirements than objects. This methodology provides convincing supports to build high quality software system in the future.

Keywords Object, Goal, Requirements analysis, SRS

一、前言

软件系统的成功极大地依赖软件需求分析的质量。有人做过统计,在美国和西欧国家,软件项目失败的两个最主要的原因:一是缺乏用户参与和不完整的需求,二是低质量的需求规格说明书^[1]。如果能找到一些方法和技术能精确描述用户的需求,(自动地)生成完整、一致、可修改性好、可跟踪性强的软件需求规格说明书(SRS),则可大大地提高软件系统的成功率,缓解甚至解决软件危机。本着这一思想,许多学者把研究的重点投向软件工程的需求分析阶段,并在七十年代末形成了软件工程一个新的分支—需求工程^[1]。需求工程主要是研究软件系统的目标、功能、所加约束与软件行为、软件演化及软件簇之间的关系,它是一个多学科的以人为中心的过程。具体来说,需求工程就是利用社会学、认知心理学、人类学、语言学、民族学等学科提供的基础理论和实践技术来获取需求并加以建模。需求工程还表示一系列的工程决策,从确认问题,解决问题直到产生关于该问题的详细的需求说明。

一般来说,需求工程分为五个阶段,分别为:需求获取、需求分析、需求编写、需求验证、需求管理^[2]。值得注意的是:由于所要求解问题的复杂性,开发者和用户交流的障碍,需求易变性等特点,需求分析活动不仅仅限于软件开发的初期阶段,而是贯穿于软件开发的

整个生命周期。需求可分为功能需求和非功能需求。前者说明系统要“做什么”,它是从用户的角度说明软件系统必须具有的行为。后者则是在功能需求的基础上进一步刻画软件系统及其环境,它包括有关设计和环境的约束,通信规程。如:互操作性、可靠性、可重用性、健壮性、高效性、灵活性、可维护性、可测试性等。其中,功能需求是软件需求的核心,在需求开发过程中,需求分析员面对不同的用户,他们的需求观是不同的,可以分为业务层需求、用户层需求、功能层需求等三个层次^[2]。

随着软件系统规模越来越大,复杂程度越来越高。需求分析也走过了从无到有,从无需求分析,结构化设计方法,面向对象需求分析直到目前处于孕育阶段的面向目标、面向 Agent 的需求分析方法。

二、面向对象需求分析

面向对象方法是一种运用对象,类,继承,封装,聚合,消息传递,多态性等概念来构造系统的软件开发方法^[1]。面向对象思想自六十年代由犹他大学的博士 Alan Kay 提出以来,首先体现在软件工程的编码阶段,自 Smalltalk 以来,出现了 Eiffel, C++, Java 等为代表的一些优秀的面向对象编程语言。八十年代中期,面向对象思想开始渗透到软件工程的前期阶段,即在系统分析和设计阶段就运用面向对象思想,九十年代,出

程 勇 硕士研究生,研究方向为计算机网络与分布式软件,目前主要兴趣是,通信软件系统需求工程,计算机网络, CSCW, 移动计算, Cai Zhiming 加拿大多伦多大学计算机科学系博士后,研究方向为基于目标和情景的通信软件系统性能需求分析,袁兆山 教授,研究方向为软件工程,软件工程环境,

现了面向对象方法学大战。著名的方法有:Booch 1993,OOSE 和 OMT 2,UML 的出现,被认为是最重要的具有划时代意义的重大成果之一。UML 是世界著名的面向对象技术专家 Grady Booch,Ivar Jacobson 和 Jim Rumbaugh 在各自所提出方法的基础上,集众家之所长,反复修改而完成的。1997年11月17日,UML1.1版被OMG(Object Management Group)批准为标准。下面主要讨论UML中的需求分析和表示方法。

UML 是一种建模语言而不是一种方法,它不仅支持面向对象的分析与设计,还支持从需求分析开始的软件开发全过程^[1]。UML 定义了10种基本的模型图(用例图、类图、对象图、包图、状态图、顺序图、合作图、活动图、构件图、配置图)从多个角度(需求视图、行为视图、结构视图、实现视图)考察系统。UML 中的模型图是一个统一的整体,它们之间的关系如图1所示^[1]。UML 的需求视图由多幅用例图组成,用例的概念是 Jacobson 在1992年首先提出来的。它是对一个执行者(Actor)使用系统的一项功能时所进行的交互过程的一个文字描述序列^[1]。执行者(Actor)是在系统之外与系统进行交互的任何事物,如自然人,其他系统和设备等。用例并不是面向对象方法中的主要概念,但它能确切地描述用户需求中的功能需求,故受到人们的广泛重视。UML 中的用例图由一组被系统边界包围的用例,边界外圈的执行者,执行者与用例之间的通讯关系组成。

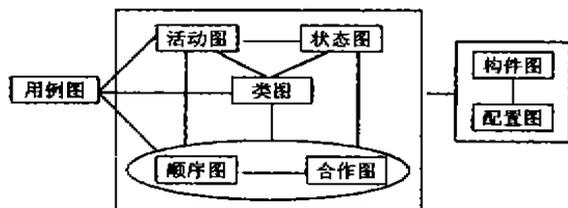


图1 UML 中模型图间的关系

在开发过程中,首先由用户方的高级管理人员和

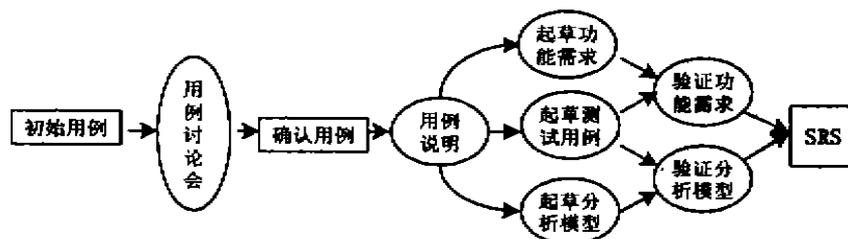


图2 面向对象需求获取过程

市场分析人员确定业务需求,业务需求常常与企业的目标(如使企业运作高效,提高企业在市场的竞争力等等)联系在一起,它定义了软件系统的视图和范围,并为以后的开发工作奠定了基础。业务需求过于抽象,并不反映各用户类(不一定都指人,还可以是其他应用程序和系统接口)关心的各种各样的功能和非功能需求。还要采用用例的方法来获取用户需求,首先要确定软件系统的用户类及其代表并划分他们需求的优先级。以一个高校信息系统为例,用户类可能有:学生,教师,管理人员、各种硬件组件和其他的系统等。

获取用例首先要确定系统的执行者,需求分析员可以让用户回答一些问题来识别执行者:如谁是系统的主要使用者?谁来维护,管理系统?系统要控制哪些硬件?系统要和哪些其他的系统进行交互?谁对系统产生的结果感兴趣等等。然后,确定业务过程,执行者(可以映射到一个和多个用户类代表)需要使用系统提供什么服务?各执行者对系统中的哪些数据感兴趣?执行者要处理哪些系统事件?需求分析员把这些交流的结果记录下来形成初始用例。初始用例还可以从其他途径获取,如实地考察用户工作情形,以特定的说明形式表达业务过程和日常行为,从这些说明中获取初始用例等。初始用例获取大致完成后,要进行用例获取讨论会,讨论会的目的是渐增式的挖掘用例,不断地加以评价和提炼,发现最重要的用例并确定优先级。

对每一个用例都必须仔细检查,审核并记录在SRS的用例描述部分。用例比起结构化方法的优点是:后者以功能为中心,前者以任务和用户为中心,用例明确地规定了用户执行的任务,不仅有助于需求分析者和开发者明确用户的业务和应用领域,而且还便于测试和跟踪,并驱动后期开发过程。

用例并不能向开发者提供确切的功能的细节。因此,用例还要细化为功能需求。功能需求是最具体的需求,开发者可以从中直接获取有关功能的详细细节,每个用例可导出多个功能需求。功能需求也必须要详尽地记录在SRS中。

最后书写的 SRS 要保持完整性、一致性、可修改性好、可跟踪性强的特点,并为后期的开发工作打下基础。

三、面向目标需求分析

过去二十年,Internet 已由科研人员的学术研究工具变为知识经济时代的关键基础设施(Infrastructure)。当前,基于 Internet 的电子商务迅猛发展,给传统的软件系统提出了新的挑战。传统的软件系统往往只是替代了人们的某些重复繁琐的工作。现在,人们发现,软件系统的引入甚至能导致整个组织(如企业)工作流程的改变,但是改变组织工作流程后,用户发现软件系统又不能满足他们的要求。因此,需求工程不仅要开发软件系统,而且还要对软件系统所处的环境进行建模。人们之所以要开发软件系统,主要是方便自己的工作,从更深意义上考虑,就是要更容易地达到某一目标。目标比对象更具稳定性,基于此,人们提出了基于目标的需求分析方法。

许多学者在需求分析的不同阶段都引入了目标概念。如 Dardenne 把目标作为一个中心概念来获取需求^[15],Anton 则用目标概念再工程遗产系统,从用例中获得目标再生成软件新环境下的需求说明书^[6]。Bubenko, Eric Yu, Jacobs 用目标把组织和组织环境同需求联系起来^[16~20], van Lamswerde 用目标来澄清需求和支持需求重用,Robinson, Dardenne, Chung 用目标概念处理需求冲突,Feather 用目标驱动设计过程^[21, 27, 31, 32], Lee 用目标的概念来跟踪需求底层的原理^[22], Chung 用目标的概念来管理变化^[26], Potts 和 Pohl 还把目标用于基于场景的需求方法中^[23~26], Chung 还把目标的概念用于处理非功能需求的框架(NFR 框架)^[16, 17]。

概括说来,目标在需求分析中有下列主要作用:

1. 方便非功能需求分析。非功能需求分析是需求工程研究中的重点和难点。目前大多数方法都是采用非形式化的方法来描述它们,如我国南京大学开发的软件需求定义语言 NDRDL 就是用自然语言来描述非功能需求,缺乏对非功能需求建模是 UML 的主要缺陷。面向目标的需求分析中,非功能需求用软目标来表示,软目标逐层分解为子目标,有利于消解冲突。Lawrence Chung 的 NFR 框架就是一个面向目标和过程的非功能需求框架^[12]。

2. 方便需求获取。在细化目标的过程中,常常会发现和揭露一些用户的需求,传统的需求分析技术仅把需求看成是过程和数据,并不能发现过程和数据底层的原因。

3. 便于把需求与组织和业务环境联系起来。系统和环境的关系要根据目标来表达,而不是其他活动和实体。

4. 方便冲突处理。目标提供了一种有用的方式来消解冲突,通过定义满足目标的不同标准就能导致不同的方式来处理冲突。

5. 驱动后续设计过程。目标是一个重要的机制来连接需要和设计阶段。

此外目标在支持需求复用,澄清需求,处理需求变更与需求验证,跟踪需求背后的决策依据等方面有广泛的应用。值得注意的是:面向目标的需求分析方法并不与面向对象的需求分析方法相矛盾。J. Mylopoulos 认为:面向目标和面向对象的需求分析方法可以看作是互补的,前者主要强调需求分析的早期阶段,而后者主要强调需求分析的后期阶段。面向目标的需求分析方法 KAOS 是一个极好的例子把这两种类型的分析方法结合起来^[9]。

KAOS (Knowledge Acquisition in autoMated Specification) 是由著名的需求工程专家 Axel van Lamswerde 领导的一个面向目标的需求工程项目。该项目的主要目的是:1)用目标,约束,对象,假设,事件,行为,主体等概念对功能和非功能需求进行形式化建模;2)支持需求框架的重用;3)采用机器学习技术;4)开发有关目标细化,操作目标,冲突检测和消解,主体分派的技术;5)开发基于目标,基于情景,基于视角策略的需求获取技术;6)开发相关的支持工具和开发环境。

KAOS 需求获取方法涉及到三个层次的建模:分别为元层次,域层次和实例层次。元层次主要由元概念如“Agent”,“Action”,“Relationship”,元关系如“Performs”,“Input”,“Link”,元属性如“Load of Agents”,“Cardinality of Link”,以及元约束所组成。元模型不仅决定了获取语言的结构,而且它还决定了需求数据库的结构。域层次是应用领域中的具体概念和所考虑问题的类型,域层次中的概念是元层次抽象的实例,域层次概念通过元关系的实例而相连。实例层次是域层次概念的实例。

由此可以看出:元、域和实例层次各自由元类型、类型、类型实例组成。KAOS 方法的需求模型是元模型的一个实例。后者可以表示为一个图。图中的每个结点表示一种抽象如目标、行为、主体、实例和事件。图中的每条边表示这些抽象间的语义链(Semantic Link)。结点和链上的属性约束它们的实例,即需求模型的元素。需求获取过程根据这些约束以某种方式遍历元模型来获得不同结点和链的实例,获取过程要受策略控制,策略指导应选择图中哪条路径。

KAOS 方法分为三个组成部分^[1]。(1)用于获取和构造需求模型的概念模型及其相关的获取语言;(2)用于详细说明需求模型的一系列策略;(3)一个在获取过程中提供指导的自动化辅助工具。

Anne I Anton 提出的 GBRAM(Goal-Based Requirements Analysis Method)也是一个面向目标的需求分析方法^[1]。在 GBRAM 方法中,Anne I. Anton 定义了目标、情景、目标障碍、目标分解、约束等一些基本概念。GBRAM 方法主要分为目标分析和目标演化两个过程。

1. 目标分析过程 在现实世界中,目标比过程、组织结构和系统更加稳定。目标可以从流程图或实体关系图的过程描述信息中提取,当对遗产系统进行再工程时,也可以从遗产系统 SRS 描述的用例中导出,还可以与用户面谈获取目标,用户比较倾向于根据操作和动作来表述需求,所以需求分析员注意寻找动词词语来提取目标。在定义目标的过程中,还要确定 agent 和约束。约束可以根据依赖关系(目标依赖和 agent 依赖)来确定,一旦目标、agent 和约束确定以后,就可以进行目标分析演化。

2. 目标演化过程 对初始目标集进行分析,剔除同义目标和联合冗余目标,对剩下的目标加以提炼和细化,剔除细化后的同义目标,并对那些涉及实现细节的目标进行处理,使其不涉及实现细节。总之,缩小初始目标集有三种主要方法:消除完全相同的目标,改进细化目标和同义目标,过程如图3所示。在这个过程中,还要考虑目标间的依赖关系和所加的约束。目标之间不可避免有冲突,要确定目标障碍,目标障碍就是妨碍目标达到的原因^[1],情景有助于揭示目标障碍。情景在这里指系统和其环境的行为描述,情景还常常揭示隐藏的目标和问题。最后,就是要实施的目标,把它连同 agent,情景,约束记录到目标计划书中,最后,采用文[13]中所介绍的方法可以把目标计划书转化为软件需求说明书,指导后续开发过程。相比 KAOS 方法,GBRAM 方法更适合于遗产系统再工程。

目前不同的基于目标需求分析方法中目标的作用和地位是不同的,为了建立一个统一的面向目标的需求分析方法,下面是一些将来有待研究解决的问题。

1. 在需求工程的各个阶段,哪些方法中目标的含义是相同的,哪些是有区别的。
2. 面向目标需求分析方法与传统需求分析方法的关系。
3. 在处理多个相关的目标概念时,哪种结合或集成方法是最合适的。
4. 各种分析方法适用的领域。
5. 目标概念向面向 agent 需求工程扩展的问题。

6. 面向目标的方法与基于情景方法的结合。
7. 目标需求语言定义。
8. 是否有工具支持。

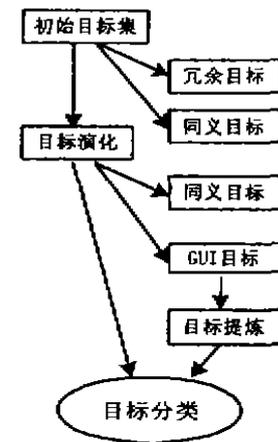


图3 GBRAM 方法

结论 目标在需求分析中起着重要的作用。有学者认为,需求的本质就是达到一个目标。这预示着需求工程本体论基础的扩展。长期以来,需求工程的本体限制于静态本体(处理实体的本体)和动态本体(处理活动的本体),面向目标分析方法的重大意义在于促使人们在需求工程中包含以目标为基础的意愿(intention)本体。面向目标需求工程中不仅对静态本体和动态本体建模,而且对意愿本体进行建模,为建造未来智能化、分布式的软件系统提供了有力的支持。

参考文献

- 1 邵维忠,杨芙清. 面向对象的系统分析. 北京:清华大学出版社,1998
- 2 Wiegers K E 著,陆丽娜等译. 软件需求. 北京:机械工业出版社,2000
- 3 刘超,张莉. 可视化面向对象建模技术—标准建模语言 UML 教程. 北京:北京航空航天大学出版社,1999
- 4 王云,刘又诚,周伯生. UML 可视化建模系统的模型一致性检查机制. 计算机研究与发展,2000,37(1)
- 5 蔡智明,刘宗田,袁兆山. 等. Web 上的组件化柔性 CASE 研究. 计算机科学,1999,26(3)
- 6 袁兆山,等. 软件体系结构描述语言 ADL 及其研究进展. 计算机科学,2000,27(1):36~39
- 7 袁兆山,等. 面向对象类库可复用性的统计分析. 计算机应用研究,1996,13(5):37~39
- 8 Anton A. goal-based requirements analysis. Available at: <http://www.csc.ncsu.edu/faculty/anton/publications.html>,2000
- 9 Axel van Lamsweerde. Goal-Driven Requirements Engineering: the KAOS Approach. Available at: <http://www.info.ucl.ac.be/research/projects/AVL/ReqEng.html>,2000

- 10 Yu E. Eric Yu's Home Page. Available at <http://www.cs.toronto.edu/~eric/>, 2000
- 11 Yu E, Mylopoulos J. Using Goals, Rules and Methods to Support Reasoning in Business Process Reengineering. In: Proc. 27th Hawaii Int. Conf. System Sciences, Maui, Hawaii, January Vol. IV, 1994. 234~243
- 12 Chung L. Requirements Engineering. Available at <http://www.utdallas.edu/chung/re.html>, 2000
- 13 Coleman D. et al. Object-Oriented Development: the Fusion Method. Prentice Hall, 1994
- 14 Anton A I. Goal Identification and Refinement in the Specification of Software-Based Information Systems. [Ph. D. thesis]. Department of Computer Science, Georgia Institute of Technology, June 1997
- 15 Dardenne A. et al. Goal-Directed Requirements Acquisition. Science of Computer Programming, 1993, 20: 3~50
- 16 Chung L, Nixon B, Yu E. Dealing with Change: An Approach Using Non-Functional Requirements. Requirement Engineering, Springer-Verlag, 1996, 1(4): 238~260
- 17 Chung L. et al. Non-Functional Requirements in Software Engineering. Kluwer Publishing
- 18 Bubenko J A. Information Modeling in the Context of System Development. Proc. IFIP, 1980. 395~411
- 19 Bubenko J A. Extending the Scope of Information Modeling. In: Proc. 4th Int. Workshop on the Deductive Approach to Information Systems and Databases, Lluet-Costa Brava, Catalonia, 1993. 73~98
- 20 Dubois E, Yu E, Petit M. From Early to Late Formal Requirements: a Process Control Case Study. In Proc. 9th Int. Workshop on Software Specification and Design, Ise-Shima, Japan, April 1998
- 21 Robinson W N. Negotiation Behavior During Requirements Specification. In Proc. 12th Int. Conf. on Software Eng., 1990. 268~276
- 22 Lee J. Extending the Potts and Bruns Model for Recording Design Rationale. In Proc. 13th Int. Conf. on Software Eng., Austin, Texas, 1991. 114~125
- 23 Pohl K. The Three Dimensions of Requirements Engineering. In Proc. CAISE'93, Paris. Springer-Verlag, Berlin, 1993. 275~292
- 24 Pohl K. Process-Centered Requirements Engineering. Wiley/Research Studies Press, New York, 1996
- 25 Pohl K, Haumer P. Modelling Contextual Information about Scenarios. In Proc. 3rd Int. Workshop on Requirements Engineering, Foundations of Software Quality REFSQ'97, Barcelona, Catalonia, Spain, 1997. 187~204
- 26 Potts C, Takahashi K, Anton A. Inquiry Based Requirements Analysis. IEEE Software, March 1994. 21~32

科学技术贵以奉献与共享 《计算机科学》夙愿作益友

欢迎阅读/订阅 2002 年《计算机科学》

《计算机科学》由国家科技部主管，西南信息中心主办，系“中国科技论文统计与分析用期刊”和“全国科技核心期刊”。主要报导国内外计算机科学与技术的发展动态，内容涉及程序理论、计算机软件、网络与信息、数据库、人工智能、人机界面、国际会议、应用等。

《计算机科学》杂志以其新颖、准确、及时为特色，突出动态性、综述性、学术性。报告特点是：“前沿学科”与“基础研究”相结合；“核心技术”与“支撑技术”相结合；“倡导”与“争鸣”相结合。广采百家之长，博览计算机世界之态势。重在突出文章的思想性（即哲理性的、范型性的、风格性的），令人有开拓思路之感；及时介绍新理论、新概念和新技术，尤其令高等院校学生有开阔视野之感。

今天，我国的信息化进程正在迅速推进，以人为本的信息技术与产品层见叠出，好酷个气氛，辉映之下，《计算机科学》又有了新发展。1999 年度其影响因子=0.675，总引用频次=396，此两项指标均列全国（计算机技术类）第二位。为了开拓信息资源、服务社会，2002 年《计算机科学》将改装为国际标准（大）16 开，版面扩大至 144 页，每期约 40 万字。版载内容将紧跟研究热点，更贴近读者。

2002 年每期定价 18.00 元，半年 108.00 元，全国各地邮局均可订阅，邮发代号 76-68。若错过订期者可直接寄现金到本社购买。

地址：重庆市渝中区胜利路 132 号《计算机科学》杂志社

邮编：400013 电话：63500828 E-mail: jsikx@swic.ac.cn

欢迎订阅《计算机科学》月刊，欢迎投稿，欢迎刊登广告