

软件需求工程——方法及工具评述

卢 梅

李明树

(中国科学院软件研究所计算机科学开放实验室 北京 100080)

摘 要 文中从需求工程的基本概念和研究内容出发,简单介绍了需求工程生命周期和需求规范等概念;比较全面地总结了现有的有代表性的需求工程开发方法和工具,对其中一些重要的方法及工具作了分类和评述,并指出了需求工程方法和工具开发与实际领域相脱离等不足之处;最后探讨了需求工程研究现状中存在的一些主要问题及一些相应的解决方案。

关键词 需求工程,需求分析,需求规格,需求分析方法,需求分析工具

中图法分类号 TP311.5

REVIEW OF METHODS AND TOOLS OF SOFTWARE REQUIREMENTS ENGINEERING

LU Mei and LI Ming-Shu

(Open Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100080)

Abstract Based on the basic concept and research contents of requirements engineering (RE), some concepts of the life cycle of RE and requirements specification are introduced in the present paper. The recent representative RE methods and tools are summarized. Some important tools and methods are classified and reviewed. Finally, some current issues of RE and corresponding solutions are also discussed in the paper.

Key words requirement engineering (RE), requirements analysis, requirement specification, requirement analysis methods, requirement analysis tools

1 引 言

需求工程是随着计算机的发展而发展的.在计算机发展的初期,软件规模不大,软件开发所关注的是代码编写,需求分析很少受到重视.后来软件开发引入了生命周期的概念,需求分析成为其第一阶段.随着软件系统规模的扩大,需求分析与定义在整个软件开发与维护过程中越来越重要,直接关系到软件的成功与否.人们逐渐认识到需求分析活动不再仅限于软件开发的最初阶段,它贯穿于系统开发的整个生命周期.80年代中期,形成了软件工程的子领域——需求工程(requirement engineering, RE).

进入 90 年代以来,需求工程成为研究的热点之一.从 1993 年起每两年举办一次需求工程国际研讨会(ISRE),自 1994 年起每两年举办一次需求工程国际会议(ICRE),在 1996 年 Springer-Verlag 发行了一新的刊物——《Requirements Engineering》.一些关于需求工程的工作小组也相继成立,如欧洲的 RENOIR (Requirements Engineering Network of International Cooperating Research Groups),并开始开展工作.

原稿收到日期:1999-03-03;修改稿收到日期:1999-06-06.本课题得到国家自然科学基金和“八六三”高技术研究发展计划基金资助.卢梅,女,1973 年 11 月生,硕士研究生,主要研究方向为需求工程.李明树,男,1966 年 5 月生,研究员,博士生导师,主要研究方向为智能软件工程、实时系统.

2 需求工程基本内容

需求工程是指应用已证实有效的技术、方法进行需求分析,确定客户需求,帮助分析人员理解问题并定义目标系统的所有外部特征的一门学科;它通过合适的工具和记号系统地描述待开发系统及其行为特征和相关约束,形成需求文档,并对用户不断变化的需求演进给予支持。RE可分为系统需求工程(如果是针对由软硬件共同组成的整个系统)和软件需求工程(如果仅是专门针对纯软件部分)。软件需求工程是一门分析并记录软件需求的学科,它把系统需求分解成一些主要的子系统和任务,把这些子系统或任务分配给软件,并通过一系列重复的分析、设计、比较研究、原型开发过程把这些系统需求转换成软件的需求描述和一些性能参数^[1]。本文以下如无特别说明,需求工程主要是指软件需求工程。

2.1 需求工程的阶段

需求工程是一个不断反复的需求定义、文档记录、需求演进的过程,并最终在验证的基础上冻结需求。80年代,Herb Krasner定义了需求工程的五阶段生命周期^[2]:需求定义和分析;需求决策;形成需求规格;需求实现与验证;需求演进管理。近来,Matthias Jarke和Klaus Pohl提出了三阶段周期的说法^[3]:获取、表示和验证。综合了几种观点,我们把需求工程的活动划分为以下5个独立的阶段:

- (1) 需求获取:通过与用户的交流,对现有系统的观察及对任务进行分析,从而开发、捕获和修订用户的需求;
- (2) 需求建模:为最终用户所看到的系统建立一个概念模型,作为对需求的抽象描述,并尽可能多的捕获现实世界的语义;
- (3) 形成需求规格:生成需求模型构件的精确的形式化的描述,作为用户和开发者之间的一个协约;
- (4) 需求验证:以需求规格说明为输入,通过符号执行、模拟或快速原型等途径,分析需求规格的正确性和可行性;
- (5) 需求管理:支持系统的需求演进,如需求变化和可跟踪性问题。

2.2 需求规格

IEEE为需求作如下定义^[4]:

- (1) 用户解决问题或达到系统目标所需要的条件;
- (2) 为满足一个协约、标准、规格或其他正式制定的文档,系统或系统构件所需要满足和具有的条件或能力;
- (3) 对上述条件的文档化的描述。

规格就是一个预期的或已存在的计算机系统的表示。它可以作为开发者和用户之间协议的基础,来产生预期的系统。规格定义系统所有必须具备的特性,同时留下很多特性不做限制。通常,我们要求规格比组成特定系统的实际的软件和硬件更简洁、更全面、更易于修改^[5]。

需求工程的主要结果是软件需求规格 (software requirement specification, SRS), SRS是对外部行为和系统环境 (软件、硬件、通信端口和人)接口的简洁完整的描述性文档。项目管理者用它来对项目进行计划和管理,在许多情况下,它也被作为是用户的使用手册或帮助用户理解系统的文档。它广泛地适用于对各类应用领域中的客户问题进行理解与描述,实现用户、分析员和设计人员之间的通信,为软件设计提供基础,并支持系统的需求验证和演进。

2.2.1 需求规格基本内容

SRS的基本内容包括行为需求和非行为需求。行为需求定义系统需要“做什么”,描述系统输入输出的映射及其关联信息,完整地刻画系统功能,是整个软件需求的核心。非行为需求定义系统的属性,描述和行为无关的目标系统特性,包括系统的性能、有效性、可靠性、安全性、易维护性、可见性和顺应性。

好的 SRS应具有如下特点^[6,7]:

正确性、无二义性、完整性、一致性、可验证性、可修改性、可跟踪性、易理解性以及有好的注解等。

2.2.2 需求规格基本描述方法

现有的需求规格描述方法有 3 种: 形式化方法、非形式化方法和半形式化方法。

形式化方法是具有严格数学基础的描述系统特征的方法。形式化方法具有准确、无二义性的特点, 有助于验证有效性和完整性。

非形式化方法使用未作任何限制的自然语言, 易于理解和使用。但它具有固有的二义性, 且难以保证其正确性、可维护性, 难以用计算机系统提供自动化的支持。

半形式化方法介于上述两者之间, 在宏观上对语言和语义有较精确的描述, 而在某些局部方面则允许使用非形式化的自然语言。

2.2.3 需求规格的技术支持

需求工程研究的核心是关于需求规格描述方法和技术的研究, 它致力于寻求以下几点支持: ① 需求规格的表示、获取机制; ② 需求规格文档制作及品质保证机制; ③ 需求规格的演示验证机制。需求规格技术支持工具主要有^[8]。

有限状态机 (FSM)、决策树和决策表、伪代码程序设计语言 (PDL)、状态图 (Statecharts)、需求工程验证系统 (REVS)、(实时) 结构化析 (SA/RT) 方法、Petri 网、需求描述语言 (SDL) 以及需求语言处理器 (RLP) 等。

3 需求工程活动

需求工程的活动可分为 3 个层次: 方法学——即整体的、全面的、指导性的方法; 方法——具体的、详细的实现途径和策略; 工具——指一步步形成的手工或自动的辅助过程。下面我们将讨论这 3 个层次的应用及发展。

3.1 需求工程方法学

需求工程方法学包括大家所熟悉的软件工程的生命周期模型, 如瀑布型、渐增型、快速原型及螺旋型。另外还有 Lano^[9]所提出的操作概念规格, 在需求产生前由开发者写成, 既能满足开发者希望需求规格精确的要求, 又能满足用户希望其易于理解的要求。Howes^[10]还提出用用户手册的方法来解决用户和开发者之间的通信问题。Sutcliffe^[11], Maiden^[12]等人提出从领域知识的角度出发在需求工程的环境中定义通用的领域语义模型和组合模型。Alford^[13]提出的“任务分割”的概念, 大大降低了需求分析的问题复杂度。Chou^[14]和 Eckert^[15]的文章讨论了面向对象的需求工程方法学的概念和模型。Drake^[16]提出用于确定系统需求边界的限定过程。Gotel^[17]在他的文章中对需求跟踪性问题进行了阐述。Rosca^[18], Krog site^[19], Jaffe^[20], Zave^[21], Robinson^[22], Basili^[23], Yu^[24], MATHALONE^[25]等人的文章也分别从不同方面对需求工程方法学进行了论述。

3.2 需求工程方法

3.2.1 需求分析方法分类

需求分析方法可大致分为 4 类, 如图 1 所示。

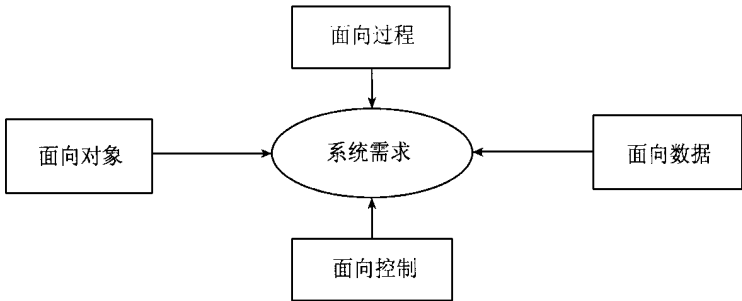


图 1 需求分析方法分类图

面向过程的分析方法的主要研究系统输入输出转化的方式, 对数据本身及控制方面并不很重视。传统的结构分析方法 SA^[26] (structure analysis), SADT^[27] (structure analysis and design technique) 就属于这一范畴; 另外还有可执行操作模型如 PAISley^[28]和 Descartes^[29]; 以及形式方法如 VDM^[30] (Vienna design

©1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. <http://www.cnki.net>

method)和 $Z^{[31]}$.

面向数据的方法强调以数据结构的形式描述和分析系统状态. JSD^[32]和关系实体 (ER)模型^[33]都是面向数据的.

面向控制的方法强调同步、死锁、互斥、并发以及进程激活和挂起. 数据流图就是典型的面向控制的方法. SADT^[27]是以面向控制的方法为辅的.

面向对象 OO的方法把分析建立在系统对象以及对象间交互的基础之上^[34],使得我们能以 3个最基本的方法框架——对象及其属性、分类结构和集合结构来定义和沟通需求. 面向对象的问题分析模型从 3个侧面进行描述,即对象模型 (对象的静态结构)、动态模型 (对象相互作用的顺序)和功能模型 (数据变换及功能依存关系). 需求工程的抽象原则、层次原则和分割原则同样适用于面向对象方法,即对象抽象与功能抽象原则是一样的,也是从高级到低级、从逻辑到物理、逐级细分. 每一级抽象都重复对象建模 (对象识别)→动态建模 (事件识别)→功能建模 (操作识别)的过程,直到每一个对象实例在物理 (程序编码)上全部实现.

面向对象的方法正在成为需求分析中的一个热点,并展现出良好的应用前景. 并产生了一些流派,如 Yourdan和 Coad的 OOA方法^[34]、Booch^[35]的方法、Jacobson的 OOSE^[36]、Rumbaugh的 OM T^[37]方法等.

3.2.2 需求工程代表性方法

需求工程方法和上述的生命周期及方法学模型是一致的. 新的需求工程的方法层出不穷,我们不可能覆盖所有的方法,在这里仅讨论一些成熟的、有代表性的、广泛应用在需求工程或软件工程领域中的一些重要的方法.

(1) CORE^[38]——Controlled Requirement Expression

Systems Designers公司制定了一整套标准和过程, CORE方法是对这套标准和过程很好的体现. 它建立在一些图表符号的基础上,这些图表符号借鉴了一些经典的规格及设计表示法思想,能够较少二义性、严格的表达系统需求. 用这些符号, CORE从不同的观察点来分析目标系统,并对此一一建模,最后形成一个组合模型. 这一特点使得 CORE适于系统动态特征的描述,而且能较好地满足 SRS完整性的要求.

(2) SADT^[27]——Structured Analysis and Design Technique

70年代由 SofTech公司的 Douglas Ross开发,用于分析、表示软件的需求及设计. SADT是需求定义开发方法中第一个基于图形的方法,每个 SADT模型由一系列图表组成,每个图表都有一个支持文本. 作为典型的结构化分析方法, SADT强调数据转换及功能分解的图形描述,表现在对从系统接收信息或传输信息给系统的外部实体的描述上. SADT还从需求定义过程中不同参与者的观察角度出发使用相互关联的多个模型来表示系统,它适用于许多建模技术.

(3) SREM^[39]——Software Requirement Engineering Methodology

SREM是 TRW于 1973~1978年开发的方法,最初用于具有严格性能需求的大型嵌入式系统. SREM描述一系列需求定义步骤,使用一定语言 (RSL)和工具 (REV S)来产生需求及需求规格,并进行语言翻译、一致性和完整性的自动检测、生成文档. SREM最突出的一点是使用了一个通用的需求库. 需求库信息的集中大大方便了用户对系统需求信息的增加、删除和修改. SREM后来还增加了对分布式并行系统的支持,即现在的 DCDS(distributed computer design system)系统.

(4) JAD方法^[40]——Joint Application Development

这一方法由 IBM于 1977年开发出来,在 80年代初首先投入使用. 这一技术的核心在于一高度结构化的工作研讨会,研讨组由行政人员、项目管理者、用户、系统专家、技术人员、JAD辅助人员及文档记录人员组成. 用户得到系统人员的帮助及经验丰富、客观的项目的辅助者的指导. 这个辅助者与项目管理者 and 用户会谈一起定义项目的范围和目标. JAD方法能够加速持不同观点的用户之间的协商,加深对软件需求的理解,并生成供用户参考的模型或原型. JAD成功之处在于它对群体需求获取的协调,同类方法还有 NCOS^[41]和 CRC^[42].

(5) Scenario方法^[43,44]

又称情景实例的分析方法,是近几年兴起的另一较有应用前景的需求工程开发方法. 情景实例分析是从用户所设想和期望的特定目标系统来理解、分析和描述系统. Scenario是由一些智能体 (Agent,包括外部用

户、外界激励和外部的功能实体)来初始化的.它包括事件和改变系统状态或触发新事件的特定激励,一个事件通常很短,作为系统内部或外部的输入或响应. Scenario方法包括 Scenario的获取、表示、验证、原型化等过程.当需求分析人员和用户均对最终的 Scenario满意时, Scenario分析过程完成.

(6) AMORE^[45]——The Advanced Multimedia Organizer for Requirements Elicitation

这一方法是卡内基·梅隆大学软件工程研究所开发的,提出用多媒体的方法进行需求捕获和建模. AMORE采用层次或网状结构,如分层的数据流图、控制流图、对象层次图、任务分解图等形式来组织大量的需求,并提供浏览和导引工具来促进需求的捕获.需求分析人员把 AMORE作为最接近其需求产生自然形式的需求存储平台,以获得最大限度的可跟踪性及促进对用户最初的需求意图的理解.而且填补了需求产生的最初形式和被一些需求规格方法和 CASE工具所采用的形式化表示法之间的距离.

上述几种方法中 CORE SADT和 AMORE注重于需求的建模及表示; JAD方法是群体讨论方法的代表; SREM严格的需求分析步骤能实现一定程度上的自动化支持; Scenario强调从用户的角度用事件及状态变化来描述需求; JAD方法和 DCDS思想的结合,对分布式多用户需求协同的研究提供了很好的思路. CORE和 SADT强调用图形的界面进行交流,比传统的字母和符号易于接受,能促进需求的快速引出和定义.但是,原始的需求信息在到这种形式化或半形式化的文字或图形符号的形式的转换过程中,其完整性和可跟踪性等可能会受到损失.而 AMORE使用最适合需求最初产生形式的媒体(如图像、声音)捕获需求,较好地弥补了这一不足.

CORE和 SADT体现了从多个不同的观察角度进行需求建模的思想, Freeman^[46]提出的多视角分辨方法也是遵循这一思想的,它还提供了对各种需求进行系统地比较和分析机制,较好地解决了需求冲突的问题.此外,还有许多代表方法从需求建模、需求定义和分析等角度对需求工程提供了支持.如 Heimdahl的文章中提出用 Mealy 机模型的 RSM(requirements state machine)对形式化的基于状态的需求规格进行完整性和一致性检查^[47]; Johoson提到的基于知识工程的需求重用和共享的方法^[48]; Ohnishi提出的可视化需求定义方法^[49]; Deno等人提出的基于“使用用例(use case)”的面向对象的需求分析方法^[50]; Potts提出的基于询问的需求分析方法^[51]; Sutcliffe文章中讨论了结合早期原型、基于脚本的分析及设计推理3个重要技术的经验学习^[52]; Faulk的 SCR方法^[53]等等.

3.3 需求工程工具

随着需求工程方法研究的不断成熟,计算机技术的迅速发展,需求工程工具的发展也产生了巨大飞跃.它们辅助需求的捕获、管理及需求文档的生成过程,并对需求工程的自动化提供支持.限于篇幅的关系,我们只能列举一些有代表性的工具.

3.3.1 基于操作方法需求工程工具

(1) GIST^[54]

GIST是一种非确定性规格说明语言.使用它可为一个有待原型化的系统产生一个形式化的、可执行的规格.用户利用自动辅助工具产生一个可执行的原型,进而改进需求规格. GIST结合了命令式、逻辑式和函数式的程序设计的概念.由 GIST生成的需求规格根据应用领域对象来描述系统活动.系统的特性是以表现为一个独立进程的“demons”形式来描述的.

(2) PAISLey^[28]

这是由 Pamela Zave在马里兰大学及后来在 AT&T 贝尔实验室开发的需求分析工具. PAISLey使用操作方法,是适用于嵌入式系统的需求规格说明语言.所谓操作是指最终生成的规格说明书能被运行或解释,最终产生的行为将模拟被创建系统所要求的行为. PAISLey享有某些函数型程序设计的特性,但 PAISLey的循环过程和交换函数是使它区别于纯函数型语言的重要之处,也使得它更适用于实时系统. PAISLey已被 Mesa System公司商业化了.

(3) STATEMATE^[55]

这一工具是由 I-logic公司在1980年于曼彻斯特开发,是为了获得一个被 Harel称为状态流图(statechart)的有限状态机的扩充.状态流图有助于需求分析人员对复杂实时系统行为无二义性的建模.使用 STATEMATE,需求分析能从功能、行为及结构3方面描述系统.这一工具最强大功能之一是它的仿

真能力,这3个视图任一方在屏幕上出现,通过一个描述系统功能和行为的可执行的模型,系统就能观察其在仿真的实时环境下的行为。

典型的操作方法的工具还有 JSD^[32],软件所在80年代末开发的 RSL/RSA等。上述工具的共同特点即它们最终结果是严格的形式化需求规格,对快速原型提供支持,需求能得到及时的验证和反馈。可执行规格和原型技术无疑为 RE提供了很好的实现途径。上述几种方法又各有侧重,如 PAISLey和 STATEM ATE主要适于嵌入式实时系统, GIST在 AI和数据库系统的应用比较广泛。

3.3.2 基于知识的需求工程工具

(1) RA^[56]—— Requirement Apprentice

RA是由 MIT研究人员开发的基于知识的系统,为需求的开发提供了一个智能助手。RA建立在特定需求领域的知识库基础上,帮助分析人员建立和修订需求规格。其核心是缩小非形式化和形式化需求规格的距离,实现前者到后者的转化。RA进程是对需求的产生一个机器可操纵的需求表示。它能回答询问和为需求分析人员、用户和系统设计者生成各种文档。这一需求分析方法的一个优点在于它能产生不同的信息表示,并根据最终用户、分析员、设计者的不同需要作相应改动。

从 AI的角度来看,RA所面临的主要问题是知识获取。RA是要从起初杂乱无章且欠精确的声明中导出一内部协调的需求声明。为此,RA要依靠一些技术,如相关定向推理、混合知识表示和通用构件的重用。

(2) TMMRP^[57]—— Technology to Manage Multiple Requirement Perspective

TMMRP是德国 USU公司基于元模型 (metamodel)对各种不同需求进行管理的工具。需求模块是对现实世界或理想世界的抽象表示,元模型则是对已存在或目标需求模块及其相互关系的抽象表示。概念库 (ConceptBase)在知识表示机制方面对这种多级抽象的管理提供了很好的支持。它是一个客户机/服务器式的元数据管理器,服务器存储、查询、更新元模型,客户端通常是一建模工具。概念库及先进的询问功能软件的支持,使得简单且可定制的元模型方法能在很短的时间内产生高质量的需求文档。

(3) QARCC^[58]—— Quality Attribute Risk and Conflict Consultant

QARCC是美国南加州大学开发的一个基于知识的需求检测工具,它用于系统生命周期的早期以检测潜在的冲突。QARCC在 WinWin^[59]系统 (USC的软件工程中心开发的的群件支持系统,在协商的获胜环境下确定软件和系统需求)环境下,识别理想的质量属性的获胜状态 (win condition),并使用一个知识库来标识软件体系结构和进程策略。

(4) PROMIS^[60]—— Prototyping MIS

PROMIS是中国科学院数学所陆汝钊教授等人设计的一个 MIS开发环境,其特点是以一个大容量的知识库来支持 MIS的开发,该知识库包括软件工程知识库和领域知识库。领域知识库存放着适用于具体领域的一个抽象概念模型, PROMIS根据用户输入的具体业务信息对模型作匹配、剪裁、扩充、修订等工作,并根据用户需求把此模型转化为可执行 MIS系统。

基于知识的工具还有 Boehm 等人的 WinWin系统和 Zeroual的 KBRAS^[61]系统等。上述几种方法的共同点即它们均把 AI技术应用于需求工程领域,具有一个知识库和一个推理机制,在此基础上进行需求分析、检测等活动。AI中知识表示和知识获取、定向推理等方法对于领域建模、问题理解和需求获取的研究是有重要意义的。这也是目前需求工程很有特色并且很有前景的一个研究方向,为我们提供了一个很好的视野和思路。我们目前开发的工具 UserTOOL在与 AI的结合进行了一些理论和实践上的探索,结合了博弈论 (Game Theory)的思想,利用 Game框架来解决多用户的需求冲突问题。

3.3.3 面向对象的需求工程工具

(1) UML^[62]—— Unified Modeling Language

UML是美国 Rational公司开发的一种用于描述、可视化和构架软件系统的建模语言。它统一了 Booch、Rumbaugh和 Jacobson的表示方法,并对其作了进一步的发展,最终统一为大众所接受的标准建模语言。UML的重要内容可以由下列5类图来定义:①用例图 (use case diagram),从用户角度描述系统功能,并指出各功能的操作者;②静态图 (static diagram),包括类图、对象图和包图;③行为图 (behavior diagram),描述系统的动态模型和组成对象间的交互关系,包括状态图和活动图;④交互图 (interactive diagram),描述对

象间的交互关系,包括顺序图和合作图;⑤实现图(implementation diagram),包括构件图和配置图.行为图和交互图是UML的动态建模机制,其余几类图是静态建模机制.

(2) ORASS^[63]——Object-Oriented Requirement System

这是南京大学计算机软件所对需求级软件自动化技术及系统的初步探索. ORASS由需求定义支撑子系统ORS和需求定义自动转化子系统FUNS组成. ORS支持面向对象的需求模型HORM的构造,形成软件需求定义. FUNS实现从图形化语言ORL描述的需求定义到面向对象规格语言OOZE(object-oriented Z environment)语言书写的形式功能规格的自动转换.

从上述工具中我们可以看出,用面向对象的思想进行需求分析,其根本要点在于,利用对象的概念模型建立一个针对于问题域的模型,用户和软件工程师通过该模型进行交流,在此模型基础上形成需求规格说明书. OO建模的好处还在于它不用在分析和设计之间划一道鸿沟,设计只需要在分析的基础上进一步根据实现系统的限制不断进行各种成分的扩充和细化.而且具有模型稳定性、可重用等等优点,这将大大降低软件维护和升级的成本.

UML可以说是代表了面向对象方法的软件开发技术的发展方向,具有巨大的市场前景,也具有重大的经济价值和国防价值.我们需求分析工具UserTOOL的开发,正在用Rational公司的可视化建模工具ROSE^[64],应用UML的思想和方法进行建模.面向对象的需求工具还有Bucci等人的TROL^[65], Tkach等人开发的VMT^[66],以及北大的JB工程^[67]等.

3.3.4 需求跟踪工具

(1) ARTS^[68]——The Automated Requirement Traceability System

ARTS是LMSC公司为软件工程师开发的需求跟踪性系统.它以“需求树”的形式链接用户定义的需求,并以这种分层结构提供对可跟踪性的支持.其首要解决的问题是选择一个数据库管理系统并加入可跟踪性构件. ART是一个数据库管理系统,在这个系统中需求是作为数据库中的记录被定义的,这些记录是由包含和需求对象有关的信息的域和属性组成.数据库系统能对需求进行选择、存储和操作. ART的开发方法是建立在快速原型和渐增模型二者结合的基础之上的.

(2) TOOR^[69]——Traceability of Object-Oriented Requirement

TOOR是牛津大学Francisco等人开发的需求跟踪工具. TOOR对面向对象的形式化规格说明语言FOOPS^[69](for functional and object-oriented programming system)的结合使得它适于面向对象的系统开发.它使用类FOOPS模型来声明需求并在项目演进的过程中自动生成新的需求链. TOOR还提供超媒体工具以更近似的反映分析人员的直观想法和活动.它由以下3部分组成:操作管理器:用于控制和执行所有对象并对操作进行建模;数据库管理器:用于控制对TOOR数据库的访问;通信管理器:用于控制和其它系统及通信用户间的交互.

需求管理,尤其是需求的可跟踪性问题一直是开发大型系统的主要问题之一.需求跟踪性问题是当前需求工程的研究核心问题之一,它把需求、设计和执行相互联系起来,能有效地验证需求规格,检测错误,管理需求演进,是保证系统成功的关键点之一.当前的需求跟踪工具还有Remap^[70]系统, RETH^[71]和Radix^[72]工具等.

3.3.5 其它

还有一些有代表性的工具,如最先为结构化分析提供计算机辅助手段的PSL/PSA^[73];支持SREM方法的工具REVS^[74]和RSL^[74];能有效描述需求接口和任务间相互关系的N2图^[75];突破传统的Von Neumann思想束缚的过程无关的第四代语言4GL^[76];Goldin等开发的协助需求引出的工具AbstFinder^[77];可视化需求分析语言VRDL^[78]等等.

尽管工具和技术近20年内有了质的飞跃,它们还是存在局限性的.目前的需求工具之间及系统间数据的传输是非常困难的,这些工具并不支持从任意级抽象的层次观察需求信息.大多工具都是为单用户设计的,对于目前较为普遍大型的协同开发环境的支持很少.我们目前开发的工具UserTOOL就希望能在多用户分布式协同方面有所突破,能推进需求工程在分布式环境的研究^[79].

目前需求工程的方法和工具的主要不足之处在于它们与实际应用领域尚有很大差距,未能充分重视影

响需求、开发及演进的目标系统开发环境.需求工具和方法的开发方向应致力与缩短与实际应用领域的差距.

4 需求工程目前的一些问题及其讨论

需求工程的研究到今天已相对成熟了,能作为一个独立的领域出现,有大量优秀的方法和工具对它提供支持.但在 RE的研究中,不可避免的存在着一些问题.

4.1 需求工程存在的几点问题

需求工程的发展不仅只受技术因素的影响,许多其它因素也会产生不容忽视的影响.我们注意到,在一些重要的社会因素的影响下,需求工程的研究在以下几个方面存在一些重要问题.

(1) 用户和开发者的协同

RE应该是一种协同、契约型作业,用户和开发部门一同达到的一个精确、无二义的协议声明.软件发展近十年来的发展趋势——系统小型化,软件生命周期缩短、通用构件及软件体系结构的重用——使大多软件开发忽视了这一要求.

(2) 对市场驱动的软件的支持

现在开发的大量软件的需求不是从用户角度出发,而是基于市场驱动的.RE活动通常是在对具体领域问题的观察得到基本解决方案形成后才进行的,其内容需涉及产品规划和市场分析.传统的 RE对这些问题的支持很少.

(3) 需求优先级分配问题

竞争使软件商要限制软件产品某些功能来加速开发进程以缩短投入市场的时间.某些非关键性需求的修改会简化目标系统的设计与实现,开发者应区分需求的优先级,在理想目标系统和需实现的系统功能之间作适当取舍.目前,在需求的优先级分配及需求集的选取机制两方面,RE进展缓慢.

(4) 需求不完备性问题的处理

80年代软件开发模型发生很大转变,原因之一是人们认识到一开始就对需求和实现作出所有正确的决定几乎是不可能的.问题的关键在于为需求完整性确定合适的边界条件,决定能接受的不完备性的种类和程度,留出一些需求有待开发阶段完善.

(5) 对设计产品的重用

开发者需能较快地描述问题及求解限制,因此需要一些评估选择策略和需求技术,使 RE能捕获和操纵设计级的产品(如通用构件).但目前具体支持很少.需求跟踪技术正迅速增涨的研究兴趣及活动,或许会对此提供一些支持.

(6) 需求分析方法和工具的支持

考虑到确定需求和确立系统的开发环境的广泛性,目前能对整个需求分析进程进行全面描述的方法和大型工具是很少的.开发者应能把需求分析过程划分成若干子问题,合理利用现有的通用工具,为具体的子需求提供自动化的支持.

4.2 当前需求工程开发方法的几点探索

上述一些问题推进了一些新的开发方法的研究,如面向环境的开发方法、通用构件的重用及面向用户的开发方法,它们为上述问题提供了一些解决途径,下面我们将对之作一些讨论.

(1) 面向问题及需求环境的方法

近来,软件界意识到从系统作用的环境这一角度去考虑需求的重要性.Jarkle和 Pohe把需求工程描述为在上下环境(context)中建立视图(vision)的过程^[3].Borgid等人提出概念建模作为需求工程的基础,正把研究转向这一方向^[80].

对此,Michael Jackson^[81]提出面向问题的方法(problem-oriented approach).他认为软件是对目标机器的描述,其开发涉及机器的结构设计,而需求则相当于目标,应该存在于机器之外,即问题环境.面向问题的方法把系统需求看作问题领域中不同现象之间的关系,并为每一类问题的提供一个固定的方法模板.这一思

想把领域专家和需求分析人员的知识一同作为需求工程的核心,改变了传统开发方法过于注重问题解决途径的结构和性质而忽略问题本身的不足。

(2) 面向获取的需求工程 PORE^[82] (procurement-oriented requirement engineering)

PORE(procurement-oriented requirement engineering)是基于构件思想的需求工程方法模型。软件构件或既有模式提供了一个“语言流通货币”,它有助于发掘并沟通存在于一些通用软件产品中的大量潜在的用户需求。其重要结果是需求规格的完整性要求不再是总是必须的了,即需求规格不必从一开始就完整地体现用户的所有需求。

PORE吸收了 3 方面的技术:即人机交互的任务建模技术、软件工程的功能建模技术和系统设计的体系结构建模技术,并从这 3 个层次实现产品选择,需求获取及产品建模。一旦产品模型建成,就可以定义产品—需求的顺应性,这就需要 AI 中基于相似性推理技术的支持。同时也为 AI 中用于类比推理法的计算机制的发展提供了新的方向。

(3) 用户主导的需求分析方法^[83]

用户主导的需求分析方法对传统需求工程乃至软件工程的根本问题——谁是系统需求和系统目标的最终定义者提出了挑战,认为只有最终用户的直接参与并发挥主导作用,才能真正解决问题空间与求解空间的一致性问题,消除应用领域与计算机领域的鸿沟,并自动适应系统需求的不断变化。Amam 等人的文章给出了一经验学习的结果,从理论上用户的参与在 RE 中的可行性和有效性作了定性的证明^[84]。

我们目前开发的应用软件系统开发的需求分析工具 UserTOOL,正在尝试研制一种有实用价值的面向某一行业领域的用户主导式应用软件开发辅助工具及原型系统,形成一套让用户成为应用系统的实际定义者的软件开发方法,建立面向领域的用户框架问题,继续完善面向用户的需求分析理论与方法,推动形成用户工程理论。

5 结束语

从需求工程的过程各阶段的作用来看,日后研究的重点还应放在需求分析、建模和可跟踪性问题的研究。当前软件开发中的一些热点技术,如面向对象技术、自动化工具、构件技术,以及传统的形式化技术、领域建模技术的发展,仍将继续为需求工程的研究提供有力的支持。RE 研究现状中另一明显的不足是理论解决方案通常是在对实际问题简化的基础上得到的,理论研究和实践脱节。要获取需求突破,改善需求工程的开发效率和质量,很重要一点是探索有效的解决途径,缩小理论和应用之间的正在增长的差距,使开发出的系统和模型切实满足应用领域的需要。

我们期待着在需求工程领域的研究在与丰富的计算机科学实践经验结合的过程中,提炼出更多成熟、完善的方法和工具,从而推动整个需求工程的进程。

参 考 文 献

- 1 Thayer R H, Dorfman M. Tutorial System and Software Requirements Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1990. 1~3
- 2 Siddiqi J, Shekaran M C. Requirements engineering: The emerging wisdom. IEEE Software, 1996, 23(3): 15~19
- 3 Jaeke M, Pohl K. Requirements engineering, 2001: (virtually) Managing a changing reality. Software Engineering, 1994, 20(11): 257~266
- 4 Macaulay L A. Requirements Engineering. London: Springer, 1996. 4~5
- 5 Pamela Z. A comparison of the major approaches to software specification and design. In: Thayer R H, Dorfman eds. Tutorial System and Software Requirement Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1990. 197~199
- 6 Balzer R, Goldman N. Principles of good software specification and their implications for specification language. In: Proc Spec Reliable Software Conf, 1979. 58~67
- 7 Liskov B, Zilles S. An introduction to formal specifications of data abstractions: Current trends. In: Yeh R T ed. Programming Methodology-Vol I: Software Specification and Design. Prentice-Hall, 1977. 1~32

- 8 Davis A M. A comparison of techniques for the specification of external system behavior. *Communications of the ACM*, 1988, 31(9): 1098~ 1115
- 9 Lano R J. A structured approach for operational concept formulation. In Thayer R H, Dorfman eds. *Tutorial System and Software Requirements Engineering*. Los Alamitos, CA: IEEE Computer Society Press, 1990. 48~ 57
- 10 Howes N R. On using the users manual as the requirements specification. In Thayer R H, Dorfman M eds. *Tutorial System and Software Requirement Engineering*. Los Alamitos, CA: IEEE Computer Society Press, 1990. 164~ 169
- 11 Sutcliffe A. The domain theory for requirement engineering. *IEEE Trans on Soft Eng*, 1998, 24(3): 174~ 195
- 12 Maiden N A M. Requirements critiquing using domain abstractions. In *Proc of 1st Int'l Conf on Requirements Engineering*. Los Alamitos, CA: IEEE Computer Society Press, 1994. 184~ 193
- 13 Alford M. Attacking requirements complexity using a separation of concerns. In *Proc of 1st Int'l Conf on Requirements Engineering*. Los Alamitos, CA: IEEE Computer Society Press, 1994. 2~ 5
- 14 Chou S, Chung C. An OOA model with system function specifications. In *Proc of 1st Int'l Conf on Requirements Engineering*. Los Alamitos, CA: IEEE Computer Society Press, 1994. 16~ 23
- 15 Eckert G. Types, classes and collection in object-oriented analysis. In *Proc of 1st Int'l Conf on Requirements Engineering*. Los Alamitos, CA: IEEE Computer Society Press, 1994. 32~ 39
- 16 Drake J M *et al.* System bounding issues for analysis. In *Proc of 1st Int'l Conf on Requirements Engineering*. Los Alamitos, CA: IEEE Computer Society Press, 1994. 24~ 31
- 17 Gotel O Z *et al.* An analysis of the requirements traceability problem. In *Proc of 1st Int'l Conf on Requirements Engineering*. Los Alamitos, CA: IEEE Computer Society Press, 1994. 94~ 101
- 18 Rosca D *et al.* A decision making methodology in support of the business rules lifecycle. <http://www.itd.nrl.navy.mil/conf/ISRE97/papers.html>. 1999
- 19 Krogstie J. Integrating the understanding of quality in requirements specification and conceptual modeling. *Software Engineering, Notes*, 1998, 23(1): 86~ 91
- 20 Jaffe M S *et al.* Software requirements analysis for real-time process-control systems. *IEEE Trans on Soft Eng*, 1991, 17(3): 241~ 258
- 21 Zave P *et al.* Requirements for telecommunications services: An attack on complexity. <http://www.itd.nrl.navy.mil/conf/ISRE97/papers.html>. 1999
- 22 Robinson W N *et al.* Supporting multi-perspective requirement engineering. In *Proc of 1st Int'l Conf on Requirements Engineering*. Los Alamitos, CA: IEEE Computer Society Press, 1994. 206~ 215
- 23 Basili V R *et al.* A knowledge-based approach to the analysis of loops. *IEEE Trans on Soft Eng*, 1996, 22(5): 339~ 360
- 24 Yu E K. Towards modeling and reasoning support for early-phase requirements engineering. <http://www.itd.nrl.navy.mil/conf/ISRE97/papers.html>. 1999
- 25 Mathalone S. A behaviorally-based methodology for modeling system specification. *Software Engineering, Notes*, 1997, 22(3): 39~ 42
- 26 Svoboda C P. Structured analysis. In Thayer R H, Dorfman eds. *Tutorial System and Software Requirements Engineering*. Los Alamitos, CA: IEEE Computer Society Press, 1990. 218~ 237
- 27 Ross D T. Structured analysis(SA): A language for communication ideas. *IEEE Trans on Soft Eng*, 1977, 1(1): 16~ 34
- 28 Zave P. An insider's evaluation of PAISLEY. *IEEE Trans on Soft Eng*, 1991, 17(3): 212~ 225
- 29 Urban J E. The descartes specification language. In Thayer R H, Dorfman eds. *Tutorial System and Software Requirements Engineering*. Los Alamitos, CA: IEEE Computer Society Press, 1990. 331~ 344
- 30 Bjoerner D. On the use of formal methods in software development. In *Proc of 9th Int'l Conf on Software Engineering*, 1989
- 31 Norries M Z (A formal specification method). A debrief report. In Thayer R H, Dorfman M eds. *Tutorial System and Software Requirements Engineering*. Los Alamitos, CA: IEEE Computer Society Press, 1990. 345~ 369
- 32 Cameron J R. A overview of JSD. *IEEE Trans on Soft Eng*, 1986, 12(2): 43~ 46
- 33 Chen P. Entity-relationship approach to data modeling. In Thayer R H, Dorfman M eds. *Tutorial System and Software Requirements Engineering*. Los Alamitos, CA: IEEE Computer Society Press, 1990. 238~ 243
- 34 Yourdon E, Coad P. *Object-Oriented Analysis*. 2nd Ed. New Jersey: Yourdon Press, 1991
- 35 Booch G *et al.* *Software Engineering with Ada*. 3rd Ed Edwood City, Calif: Benjamin/Cummings, 1994
- 36 Jacobson I. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Reading, MA: ACM Press, 1992
- 37 Rumbaugh J *et al.* *Object-Oriented Modeling and Design*. New Jersey: Yourdon Press, 1991
- 38 Mullery G P. CORE— A method for controlled requirement specification. In Thayer R H, Dorfman M eds. *Tutorial System and Software Requirements Engineering*. Los Alamitos, CA: IEEE Computer Society Press. 1990. 304~ 313
- 39 Alford M. SREM at the age of eight: the distributed computing design system. *IEEE Computer*, 1985, 18(4): 36~ 46
- 40 Macaulay L A. *Requirements Engineering*. London: Springer, 1996. 94~ 97

- 41 Harwell R M. National council on systems engineering (NCOSE)—— Requirements managements specification. In Proc of 1st Int'l Conf on Requirements Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1994. 60~ 60
- 42 Macaulay L A. Requirements Engineering. London: Springer, 1996. 102~ 110
- 43 Hsia P *et al.* Formal approach to Scenario analysis. IEEE Software, 1994, 21(3): 33~ 41
- 44 Zorman L. Requirements Envisaging by Utilizing Scenarios(REBUS). <http://www.cc.gatech.edu/computing/SW-Eng/re-thesis.html>. 1999
- 45 Wood D P *et al.* A multimedia approach to requirement capture and modeling. In Proc of 1st Int'l Conf on Requirements Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1994. 53~ 56
- 46 Frerman P A. Requirement validation viewpoint resolution. IEEE Trans on Soft Eng, 1991, 17(12): 1253~ 1269
- 47 Heimdhal M E. Completeness and consistency in hierarchical state-based requirements. IEEE Trans on Soft Eng, 1996, 22(6): 363~ 377
- 48 Johnson W L. Sharing and reusing of requirements knowledge. In Proc of 6th Annual Knowledge-based Soft Eng Conf. Los Alamitos, CA: IEEE Computer Society Press, 1991. 57~ 65
- 49 Ohnishi A. A visual software requirements definition method. In Proc of 1st Int'l Conf on Requirements Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1994. 194~ 201
- 50 Dano B. Producing object-oriented dynamic specifications: An approach based on the concept of 'Use Case'. <http://www.itd.nrl.navy.mil/conf/ISRE97/papers.html>. 1999
- 51 Potts C *et al.* Inquiry-based requirements analysis. IEEE Software, 1994, 21(3): 21~ 32
- 52 Sutcliffe A. A technique combination approach to requirements engineering. <http://www.itd.nrl.navy.mil/conf/ISRE97/papers.html>. 1999
- 53 Faulk S *et al.* Requirements specification and analysis with SCR. <http://www.itd.nrl.navy.mil/conf/ISRE97/tutorials.html>. 1999
- 54 Swartout W *et al.* GIST English generator. In Proc National Conf Arti Intell Pittsburgh, PA, 1982. 404~ 409
- 55 Harel D *et al.* STATEMATE: A working environment for the development of complex reactive systems. In Proc 10th Int'l Conf Software Eng, Los Alamitos, CA: IEEE Computer Society Press, 1988. 396~ 406
- 56 Reubenstein H B *et al.* The requirement apprentice: Automated assistance for requirements acquisition. Trans on Soft Eng, 1991, 17(3): 226~ 240
- 57 Nissen H B, Zemanek G V. Managing multiple requirements perspectives with metamodels. IEEE Software, 1996, 23(3): 37~ 48
- 58 Boehm B. In H. Identifying quality requirement conflicts. IEEE Software, 1996, 23(3): 25~ 34
- 59 Boehm B *et al.* Software requirements as negotiated win conditions. In Proc of 1st Int'l Conf on Requirements Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1994. 74~ 83
- 60 (陆汝钊, 金芝等). 基于领域知识的需求信息获取. 软件学报, 1996, 7(3): 137~ 144
(Lu Ruqian, Jin Zhi *et al.* An approach to acquiring requirement information based on domain knowledge. Journal of Software(in Chinese). 1996, 7(3): 137~ 144)
- 61 Zeroual K. KBRAS: A Knowledge-based requirements acquisition system. In Proc of 6th Annual Knowledge-based Soft Eng Conf. Los Alamitos, CA: IEEE Computer Society Press, 1991. 38~ 47
- 62 Rational Software Corp. Unified Modeling Language. <http://www.rational.com/uml>. 1999
- 63 张家重, 王志坚等. 对象式需求模型及机器支撑. 软件学报, 1998, 9(6): 414~ 418
(Zhang Jiazhong, Wang Zhijian *et al.* A hierarchical object-oriented software requirements model and its mechanical support. Journal of Software(in Chinese). 1998, 9(6): 414~ 418)
- 64 Rational Software About Rational ROSE. <http://www.rational.com/uml/Rose1.htm>. 1999
- 65 Bucci G *et al.* An object-oriented dual language for specifying reactive systems. In Proc of 1st Int'l Conf on Requirements Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1994. 6~ 15
- 66 Tkach D *et al.* Visual Modeling Techniques—— Object Technology Using Visual Programming. California: Corporate and Professional Publishing Group. 1996
- 67 杨芙清等. 面向对象的CASE环境青鸟II型系统的设计与实现. 中国科学, A辑, 1995, 25(5): 533~ 542
(Yang Fuqing *et al.* The design and implementation of an object-oriented CASE environment of Jade Bird 2 system. Science in China(in Chinese), Series A, 1995, 25(5): 533~ 542)
- 68 Flynn R F *et al.* The automated requirement traceability system(ARTS): An experience of eight years. In Thayer R H, Dorfman M eds. Tutorial: System and Software Requirements Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1990. 423~ 438
- 69 Pinheiro F C *et al.* An object-oriented tool for tracing requirements. IEEE Software. 1996, 23(3): 52~ 64
- 70 Ramesh B *et al.* Supporting systems development using knowledge captured during requirements engineering. IEEE Trans Soft Eng, 1992, 18(6): 498~ 510
- 71 Kaindl H. The missing link in requirements engineering. Software Engineering Notes, 1993, 18(2): 30~ 39

72 Yu W. Verifying software requirements—— A requirement tracing methodology and its software tool—— Radix. IEEE Selected Areas in Comm. 1994. 234~ 240

73 Sayani H. PSL/PSA at the age of fifteen. In: Thayer R H, Dorfman M eds. Tutorial System and Software Requirements Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1990. 423~ 438

74 Requirement Engineering Tools and Techniques. <http://www.jrcase.mq.edu.au/seweb/requirements/requirements.html>. 1999

75 Lano R J. N² Charts. In: Thayer R H, Dorfman M eds. Tutorial System and Software Requirements Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1990. 423~ 438

76 Cobb R H. In Praise of 4GLs. In: Thayer R H, Dorfman M eds. Tutorial System and Software Requirements Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1990. 423~ 438

77 Godlin L *et al.* AbstFinder, a prototype natural language text abstraction finder for use in requirements elicitation. Auto Soft Eng, 1997, 4(10): 375~ 412

78 Ohnishi A. A visual software requirements definition method. In: Proc of 1st Int'l Conf on Requirements Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1994. 194~ 201

79 Li M. UserTOOL: A user-driven domain-specific requirement analysis tool. ISCAS-LCS-98-04. 1998; also in: Proc of 3rd Joint Conf on Knowledge-Based SE. 1998. 64~ 67

80 Borgida A *et al.* Knowledge representation as the basis for requirements specification. Computer. 1985, 4: 82~ 91

81 Jackson M. Software Requirements and Specifications. Reading, MA: ACM Press, 1995

82 Maiden N A, Ncube C. Acquiring COTS software selection requirements. IEEE Software. 1998, 25(3): 37~ 47

83 Li M. User-oriented requirement analysis in automated MIS production. In: Proc of 12th Int'l Conf on CAD/CAM Robotics and Factories of the Future(CARS& FO F 96), 1996. 1~ 5

84 Aman K E *et al.* User participation in the requirements engineering process: An empirical study. Requirements Engineering, 1996, 1(1): 4~ 26