

需求工程综述

田 忠 钱乐秋

(复旦大学计算机科学系 上海 200433)

摘 要 需求工程是系统开发的初始阶段,其目的是得到关于客户问题的准确、清晰、一致的描述以及相应的求解限制。尽管人们已经认识到需求工程的质量影响到整个系统开发的成败,但开发人员在需求工程阶段所得到的支持却最少。本文从需求工程的基本概念出发讨论了需求工程的主要问题以及现有研究工作,并试图从领域、语言、模型及解释四个方面来讨论关于需求工程质量及其研究成果。

关键词 需求工程,综述。

A SURVEY OF REQUIREMENTS ENGINEERING

TIAN ZHONG · QIAN LEQIU

(Department of Computer Science, Fudan University, Shanghai 200433)

Abstract Requirements engineering is the initial phase of system development. It is our aim to formulate an accurate, concise and consistent description of client's problem and corresponding constraints on the solution. Although we generally accept that the quality of requirements engineering has great effects on the success or failure of a project, the system developers usually find themselves with little support in this phase. This paper gives the fundamental concepts, activities, problems, and current research efforts of requirements engineering. Also, we discuss the quality and research results of requirements engineering through the four aspects of conceptual modeling: language, domain, model and interpretation.

Keywords Requirements engineering, survey.

一、需求工程的基本内容

需求工程是指应用已证实有效的原理、方法,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。需求工程覆盖了体系结构设计之前的各项开发活动,主

本文 1995 年 3 月 5 日收到。田忠,副研究员,主要从事面向对象技术、需求工程及数字化图书馆技术等研究。钱乐秋,教授,主要从事软件工程的教学和研究。

要包括分析客户要求、对未来系统的各项功能性及非功能性需求进行规格说明,并针对不同的对象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)。在系统开发中,需求工程往往与体系结构设计交替进行,直到分解的子问题可以单纯地由软件或硬件系统解决。软件需求工程则是对应用于纯软件系统开发生命期中系统设计之前的第一阶段^[4]。因此,需求工程的目标相当简单明了:确定客户需求,定义设想中系统的所有外部特征。本文将只讨论软件需求工程,并简称为需求工程。

1.1 需求工程阶段与需求规格说明

软件产品的生产可以看成是一个映射。将客户最初的非形式想法映射成最终解决此问题的软件产品,需求工程则是这一巨大飞跃中起决定性作用的第一步。需求工程可划分成若干具有独立目标的子阶段:

- 问题分析 分析员采用集体讨论、个别交谈等方法,力求获得关于客户领域的基本理解,即关于问题的基本知识和求解限制,最终帮助客户确定软件系统的边界。分析员将问题分析阶段的成果用适当的机制记录下来,形成客户关于其问题的初始需求陈述;

- 需求获取 分析员在某种(或几种)需求获取方法的指导下,系统地从事非形式需求陈述中提取出客户的实际需求,以某种需求描述机制记录下来,形成客户问题的需求规格说明,并尽可能地消除其非形式的特征(如:二义性、不一致性、矛盾性);

- 需求验证 分析员以需求规格说明为输入,通过符号执行、模拟或快速原型等手段向客户展示现有需求规格说明所刻画系统的外部行为及相应的特征。

由此可见,需求工程的各项活动以需求为中心,紧紧围绕着需求规格说明展开:问题分析奠定了获取和解释需求模型的知识背景;需求验证确保所得到的需求规格说明反映了客户的真实意图。需求规格说明应包含两方面内容:一是功能需求,即待开发系统的输入/输出及相互依赖关系的描述;二是非功能需求,主要包括一些内外部环境的限制,如国家、行业的政策、法令、法规以及经济限制、开发过程要求、运行环境要求、性能要求、安全性要求、用户操作要求等。

尽管人们开始认识到需求工程在系统开发中至关重要,尽管人们开始认识到需求规格说明是整个系统开发的基石,但无论从技术上还是从支持环境或工具上看,目前需求工程所获得的支持都最少^[11]。需求工程研究是一项综合运用多学科成果的活动,其主要困难有:

- ① 应用领域/客户问题的多样性;
- ② 客户要求的时效性及多变性;
- ③ 客户要求的内在矛盾及多用户要求的潜在冲突;
- ④ 自然概念中存在的本质矛盾(如:“鸟会飞”与“鸵鸟不会飞”的矛盾);
- ⑤ 客户需求的描述形式及获取过程的非形式特征。

因此,需求工程研究的核心是关于需求规格说明技术的研究,应致力于寻求:

- 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,从而提取客户需求规格说明的机制;
- 需求规格说明的表示、获取、维护、文档制作及品质保证机制;
- 需求规格说明的演示验证机制。

1.2 需求规格说明的品质保证体系

所谓品质保证即是给出一系列质量评判标准与改进手段。多数有关需求分析的文献大多罗列一系列属性来描述什么是良好的需求规格说明,这些属性往往相互重叠和矛盾,并且目标和性质相互混杂。文献 [2] 从语言学出发给出了一个考察需求规格说明质量的框架。

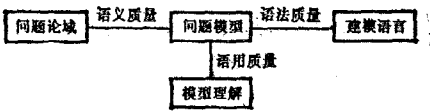


图1 需求规格说明品质保证框架

需求规格说明是分析员根据他对客户需求陈述的理解而构造的关于客户问题的一种模型。这一模型作为系统开发各方达成的共识是对系统进行设计、实现、测试、验收的基本依据。需求规格说明往往由分析员用某种语言(正文、图表等)写成需求规格说明书的形式。因此,需求规格说明实际上涉及到如图1的四个方面和三个关系:

四 个 方 面	三 个 关 系
<ul style="list-style-type: none">• 需求规格说明书(需求模型 M)• 构造需求规格说明的语言(建模语言 L)• 所解决的问题(问题领域 D)• 开发小组对于需求规格说明的理解(模型理解 C)	<ul style="list-style-type: none">• 构造 M 时 L 的应用情况(语法质量)• M 在 D 中的实际意义(语义质量)• 开发小组各成员对 M 的理解 C 的相互映证情况(语用质量)

这里,开发小组的成员包括提出该问题的客户代表、领域专家、参与解决此问题的分析员、设计师、测试者以及其他非技术人员(如项目经理、会计审计人员等)组成。从图中可以看出,需求规格说明的质量实际上有语法、语义、语用三个方面的内容,而对其产生影响的主要因素可归纳为三类:

建模语言与应用领域的适配性 即该语言是否适用于问题所处的领域,是否能充分描述此领域。

建模语言与开发小组的适配性 即开发小组成员是否能充分学习、理解并使用此语言。

开发小组与应用领域的适配性 即开发小组成员是否熟悉(或者是否能掌握)应用领域。

事实上,我们不仅可以从语法、语义、语用三个角度来考察需求规格说明的质量,还可以从图1的四个方面、三个关系来考察现有的各种需求工程技术。

二、需求工程基本途径

需求规格说明是需求工程的核心:无论是问题理解还是需求获取或是需求验证,所针对的都是需求规格说明,因而需求工程研究的关键是对需求规格说明的研究。问题理解主要是领域建模的工作,是一个识别和组织关于某类问题(问题领域)知识以支持对这类问题的描述和解答的过程^[3]。本文将主要讨论需求工程中需求规格说明的方法及相应的语言。

2.1 需求规格说明方法的选择

需求获取方法有两方面的含义:一是指导开发小组获得客户需求的方法框架;二是支持

控制此项活动进展的过程控制机制。前者是进行需求获取的一系列工作步骤；后者则是研究在受控条件下获得前者的工作数据以及各种记号之间的语义交叉点，属于需求工程研究中的一项中长期工作^[4]。目前有关需求规格说明方法的研究大都集中在前者。由于缺乏必要的数据库，关于后者的工作很少。

分割策略、抽象策略、投影策略是人类对复杂问题进行分析、理解的三种基本策略。分割策略要求将一复杂问题分解成若干个子问题，并依次逐渐分解下去，直到每一部分都是足够简单为止，反映出问题中各种概念间的整体部分关系。抽象策略则要求首先获得关于问题总体的高层认识，然后逐步细化这一认识，以得到更加具体层次上关于问题的认识，反映出对问题的理解由一般到特殊的渐进过程。投影策略要求将问题映射到各种不同的层面上，以便开发人员从较为单一的角度来考察问题和理解需求规格说明并相互印证，从而发现领域专家知识中的缺陷，显现客户中不同群体要求的矛盾及需求规格说明中所使用语句的不同理解。因此，任何实际的需求规格说明方法(Requirements Specification Method, RSM)都必须向使用者提供应用上述分割、抽象、投影策略的手段；必须有助于开发各方人员的思想交流；必须鼓励使用领域术语而非软件术语；允许并提示需求规格说明中多种意向选择的存在；必须支持增量式需求开发，所产生的需求规格说明必须易于维护。

现有的需求规格说明方法大致可分三类：形式方法、非形式方法、基于知识表示的综合方法。它们从各自立场或多或少地满足上述要求，但目前尚未有满足上述所有要求的主流方法。

形式化 RSM 是具有严格数学基础的描述系统特征的方法，有助于澄清客户的非形式需求陈述。形式方法大致可分两类：一是面向模型的，即应用一系列数学结构(如元组、关系、函数、集合等)构造系统的模型来直接定义系统的行为；二是面向特征的，即通过给出系统必须满足的公理集来描述系统的特征，从而间接地定义系统的行为。形式化 RSM 往往与形式化需求规格说明语言紧密结合在一起，前者典型的方法如 VDM、Z、Petri 网、CSP 等；后者又可分为公理规格说明(如 OBJ、Larch 等)和代数规格说明(如 Clear、Act、Raisc 等)。形式方法通常有严格定义的分割、抽象、投影机制，其数学定义有助于澄清认识，规格说明的构造往往是增量式的。但其数学定义不是所有开发人员都能轻易掌握的，与一般应用领域有相当的距离，并且其文档不可避免地过份庞大，再加上规模扩大问题，目前其应用相当有限。

非形式 RSM 往往以某种方法学或方法框架的形式出现，通常非形式地描述一系列进行需求规格说明的步骤和原则，并定义相应的记号。典型的非形式 RSM 如结构化分析(SA)方法和目前文献中不断涌现的各种面向对象分析(OOA)方法。SA 方法源于数据处理应用，是一种单纯的由顶向下的功能分解技术。OOA 方法大多通过对象(类)、状态、交互行为来刻画问题及问题的解，强调对现实世界中的对象及对象类的定义和求精。“对象”在需求分析中一般被定义为与问题领域有关的具有清晰边界的现实实体，是属性和行为的封装整体，其属性和行为在理解问题中至关重要。非形式 RSM 通常也定义有分割、抽象、投影机制来分析和组织问题及其描述，易于入门，鼓励使用领域术语。但其非形式性使得人们难以保证其正确性、可维护性。

基于知识表示的综合技术 是人工智能技术应用于软件工程中的典型范例，结合了形式技术及非形式需求规格说明方法的长处。知识表示技术为需求规格说明奠定了形式基础，

而非形式的方法框架给出了需求规格说明的指导原则。这种技术往往有某种辅助系统来检测当前的需求状态,提示下一步的工作。这类技术往往是隐式地定义投影、分割、抽象机制,鼓励使用领域术语(即共享的领域知识),并由此来帮助开发人员交流思想。其辅助系统一般能检测和提示多种意向的存在,并支持对需求的增量式开发。因而,其前景依赖于辅助系统的智能化程度和方法框架给出的各项指导原则的有效性。与形式化 RSM 类似,其实际应用中存在 scaling-up 的问题。

2.2 需求规格说明语言的选择

在需求工程中,需求规格说明语言(RSL)无论是显式或隐式定义的,都是表达和处理客户需求的最重要的介质。它不仅要表达问题本身,还要描述问题所处的环境。因而必须具有充分的表达能力来描述现实概念的不可避免的变化、时间、例外等特征;必须提供表达抽象、分割的机制及相应的指导原则;必须提供检测所获得的需求描述的一致性标准;必须是易学、易用,其文档必须是易读、易懂的。同 RSM 一样,现有 RSL 或多或少具有以上特点,但还没有哪种 RSL 占据主导地位。它们大致可分为四类:自然语言、结构化行为描述语言、形式语言、半形式语言。

自然语言(或受限自然语言)作为 RSL 一直是人们追求的目标。研究表明,应用自然语言表达需求的本质困难不在于其语法/语义上的复杂性,而是其固有的二义性。自然语言 RSL 虽可满足理想 RSL 的其它要求,但难以提供一致性标准,也难以确保每个人对给出的规格说明具有相同的理解。结构化行为描述(如 DFD、各种面向对象模型以及 PSL/PSA 文本等)用结构图表或正文表示需求,部分地消除了自然语言的二义性,但没有本质的改变。形式语言从数学理论出发给出问题的精确定义,但现实问题的多变性、日常概念不可避免的例外情况、时间概念的处理都给形式系统提出了严峻的挑战。研究人员也在努力开发各种解释器以简化形式语言的学习、应用和理解。

半形式语言是基于研究现状的一种折衷。在形式框架中添加某些机制,允许进行非形式的描述,如 RA 的模板^[6]允许在框架知识表示中使用 canned-text(结构自然语言)填充槽来例示框架。

当前实际使用的需求规格说明书大都使用自然语言书写,其长度(由几页到几千页不等^[4])与问题的复杂度没有必然的联系。使用自然语言书写需求规格说明书是相当危险的,其非形式特征将阻碍开发小组关于待开发系统各项细节达成共识。

2.3 需求规格说明的质量保证目标与手段

正如节 1.2 所指出的,我们从三个方面来确定质量保证目标及相应的手段。

需求规格说明语法质量目标是模型的语法正确性,即确保所构造模型中的所有语句均符合 RSL 的语法规则。模型中的语句在句法上完整,且不得使用 RSL 中不曾定义的符号。为此,有三种保障语法质量的手段:预防错误、检测错误和更正错误。前两者可以通过 RSL 的形式预防来完成,后者则较难自动化。预防错误即是要拒绝在模型中加入错误语句;检测错误则是在模型构造中或构造后发现错误;更正错误是用正确的语句去替换错误的语句。

需求规格说明语义质量目标可归纳为需求模型的有效性和完备性。有效性要求需求模型中的所有语句都是正确的且均与问题有关;完备性要求需求模型包含领域中关于问题的所有正确、相关语句。需求模型与领域越相似,该模型的语义质量越高。但对任何实际问题,人们不可能达到彻底的有效性和完备性。因而较为实际的目标是在一定可信度下的有

效性和完备性。可信有效性是指需求模型中存在某些无效的语句,但剔除它们的好处尚不及所花费的开销;可信完备性是指需求模型中还缺少某些关于领域的与问题有关的正确语句,但试图补充它们的好处尚不及所花费的代价。有效性反映了人们为达到某种目标在开销与成效之间作的权衡,遗憾的是我们还没有任何度量可信度的有效机制。多数提高模型质量的方法都依赖于人们对模型具体内容的理解,实际上是语用手段,一致性检测往往可自动完成,不必真正理解所给出的模型。

需求规格说明语用质量影响到人们对表达同一意义的多种表达方式的选择,其目标是理解性。理解性不仅要使得需求模型能够被理解,而且要确保开发小组所有成员确实理解该模型^[3]。与语义质量目标一样,我们也需在理解性中引入可信度。因为对一大型模型而言,我们不能期望所有人员都能理解模型的所有方面,他们只需理解与己有关的部分。任何有助于理解需求模型的手段都可以纳入达成此语用目标的有益途径。如各种逐项阅读以求理解的模型检查;以图表代替文字的模型可视化;利用动画表现系统动态特征的模型动画显示;根据统计数据预测模型所刻画软件系统行为特征的系统模拟;基于(人工或机器辅助)解释的模型查阅手段;基于过滤(甚至包括语言翻译)的阅读范围控制等。

三、需求工程现有研究成果

3.1 需求工程所需支持

在需求工程中,人们需要三个方面的支持:知识背景、技术方法、辅助工具。知识背景提供了进行需求工程的知识环境和基本语汇,是开发小组进行思想交流的基础。主要包括关于领域的专家知识;关于问题本身的特定知识;关于需求(执行过程、操作风格、指导方法、评价标准等)的知识。

知识处理技术 包括各类知识的表示与操纵、组织与推导。知识处理能力是影响需求工程成效的重要因素;

领域建模技术 是获得领域知识的关键技术,建立起开发小组的工作环境,其主要问题是领域元模型的设计、例化和验证;

问题获取技术 目前是采用集体讨论加上个别交谈的方法来获得关于问题的描述,但这两种非形式技术完全依赖于主持者的素质和交谈对象的合作意愿。需求规格说明中的许多困难(不断变化的要求、主观承诺等)很大程度上源于对问题的不切实际的描述;

需求获取技术 从问题描述中提炼出准确、清晰、一致的需求描述是系统开发由非形式到形式的转化中最关键的一步。需求获取技术不仅要支持开发小组获得客户的真实需求,并且要提供简炼、易懂的机制描述客户需求;

需求验证技术 是需求质量保证体系的重要组成部份,主要通过符号执行、快速原型等手段来保证需求规格说明的语用质量。针对需求规格说明的三种质量,需求工程环境应具有能保障语法正确性的模型编辑器;模型一致性检查器(包括模型的内部一致性及与环境的一致性);模型视图过滤器;文档生成/追踪器;模型解释器(包括文字释义、图表/动画示意、符号执行等);快速原型(功能原型、用户界面原型、性能原型)生成器。

3.2 现有工作

有关需求工程的研究遵循两条途径:形式途径(包括完全形式化方法的研究及基于知识

表示的各种综合技术的研究)和非形式途径。前者的典型工作我们在文献[6]中已作了讨论,本节我们将讨论非形式需求工程的研究。这方面的工作主要是解决需求工程的指导原则、方法框架和操作启示。其典型的工作有两大类:结构化分析方法和面向对象分析方法。

结构化分析(SA)方法是一种单纯的由顶向下逐步求精的功能分解方法。分析员首先用上下文图表(称为数据流图 DFD)表示系统的所有输入/输出,然后反复地对系统求精,每次求精都表示成一更详细的 DFD,从而建立关于系统的一个 DFD 层次。为保存 DFD 中的这些信息,使用数据字典来存取相关的定义、结构及目的。SA 方法是目前实际应用较为广泛的需求工程技术(著名的有 SADT、SREM、PSA/PSL)。它具有较好的分割、抽象能力,为开发小组找到了一种中间语言,易于软件人员所掌握。但它离应用领域尚有一定的距离,难以直接应用领域术语,与软件设计也有一段不小的距离,因而为开发小组的思想交流带来了一定的困难。

面向对象分析方法是八十年代以来的热门话题,强调系统是对现实的模拟,通过对象、状态、服务及对象间交互来同时刻画系统的静态、动态特征。正如[4]所指出的目前文献中大量的 OOA 方法可归纳为四类:

(1) 源于面向对象设计(OOD)^[7,8] 由于问题的理解本身就是模型化现实世界,因而使用面向对象技术是自然的。由 OOD 来的需求技术通常不区分需求对象和设计对象,唯一的差别是其详细程度。其优点是领域中对象较之功能更易于定义和稳定,且由需求到设计的转换直接了当;其缺点是需求分析与设计中选择对象的标准不同,同时分析与设计的分割、抽象及投影策略也有不同的出发点,这种视点转移难以控制和适应。

(2) 源于数据库设计^[9] 这类需求工程技术源于 ER 模型,往往忽略或轻视作用于实体上的函数或过程。其优点是问题领域中的对象比其功能更易于定义和稳定,此后的数据库设计相对直接了当;其缺点则在于理解问题所必须模型化的许多对象并不能对应于数据库实体。

(3) 源于需求分析技术^[4,10] 一些面向对象需求工程技术源于标准的需求分析或系统分析技术,通过分析现实实体来理解问题,强调所创立的对象完全属于问题领域。其优点是问题领域中的对象较之其功能更易于定义和稳定,只有与现实问题有关的对象被模型化,对象内封装了其属性和服务;其缺点是由分析到设计的模型转换可能不会轻而易举。

(4) 源于结构化分析方法 这类方法多数以“Object-Oriented Structured...”的标题出现,试图融合功能分解与实体分解,将 DFD 模型转换成基于现实实体的面向对象模型。

四、小 结

需求工程是软件开发中最为重要的阶段之一,但目前所获得的支持却最少。现有的研究大多集中在需求验证和可执行的需求规格说明语言方面。关于领域模型、问题理解、需求获取等的研究还刚刚起步。AI 中关于知识表示、知识获取与知识处理的研究对需求工程的研究将起重大作用。需求工程研究的进展不仅能提高需求规格说明质量,而且将从整体上极大地改善软件开发的效率和质量。

(下转第 49 页)

七、结 束 语

本文提出了一种针对特定说话人的汉语连呼数字识别算法。对发音材料各异的参数特征采用分级处理的方法,在降低计算复杂性的同时亦保证了较高的识别率。

参 考 文 献

- [1] S. K. Soong and E. H. Rosenberg, On the Use of Instantaneous and Transitional Spectral Information in Speaker Recognition, IEEE Assp-36, pp. 871~879, Jun., 1988.
 - [2] L. R. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of IEEE, Vol. 77, No. 2, Feb., 1989.
 - [3] J. F. W., C. H. W., Integrating Neural Nets and One-Stage DP for Speaker Independent Continuous Mandarin Digit Recognition IcAssp, pp.69~73, 1991.
 - [4] Myers C. S., Rabiner. L. R., A level Building Dynamic Time Warping Algorithm for Connected Word Recognition IEEE, Assp-29, pp. 284~297, 1981.
 - [5] 史锡尧、杨庆慈,现代汉语,北京师范大学出版社, pp.31~54, 1984.
 - [6] 叶明、李志钧,“基于动态时轴弯曲的汉语连呼数字识别算法”,《南京航空学院学报》, Vol. 24, No. 6, 1992, 12.
-

(上接第 22 页)

参 考 文 献

- [1] Alan A. M., “Software Requirements: Analysis and Specification”, Prentice-Hall, Englewood Cliffs, NJ, 1990.
- [2] Lindland O. I., Sindre G. and Solyberg A., “Understanding Quality in Conceptual Modeling”, IEEE Software, 11(2): 42~49, March, 1994.
- [3] Arango G and Prieto-Diaz R., “Introduction and Overview: Domain Analysis Concepts and Research Directions”, in Prieto-Diaz (ed.) “Domain Analysis and Software System Modeling”, 9~33, IEEE CS Press, 1991.
- [4] Davis A. M., “Software Requirements: Objects, Functions and States”, Prentice-Hall, Englewood Cliffs, NJ, 1993.
- [5] Reubenstein R. E., “Automated Acquisition of Evolving Informal Descriptions”, MIT AI Lab, AI-TR-1205, 1990.
- [6] 钱家骅等,“需求工程的形式化途径”,《计算机科学》,19(6): 44~49, 1992, 6.
- [7] Rumbaugh J. et al., “Object-Oriented Modeling and Design”, Prentice-Hall, Englewood Cliffs, NJ, 1991.
- [8] Booch G., “Object-Oriented Design with Application”, Benjamin/Cummings Publishing Company, 1994.
- [9] Shlaer S., Mellor S. J., “Object Oriented Systems Analysis: Modeling the World in Data”, Yourdon Press, Englewood Cliffs, NJ, 1988.
- [10] Rubin K. S., Goldberg A., “Object Behavior Analysis”, Communication of the ACM, 35(9): 48~62, Sept. 1992.
- [11] Hsia P., Davis A. M., Kung D., “Status Report: Requirements Engineering”, IEEE Software, 10(6): 75~79, Nov., 1993.