

试论软件开发中的需求工程

吴会松 (华北石油勘察研究院计算机室 062552)

摘要:随着软件开发规模的扩大,以及人们对软件产品质量要求的提高,用工程化方法解决需求问题已是当务之急。鉴于此,本文对需求工程的概念及内容提出了自己的看法,试图从工程的角度探讨需求分析在当前的主要问题及可行途径,并对现有的可用成果作了分析。

关键词:软件开发 需求工程 软件需求 语言 模型

1. 需求工程的概念

在软件开发中,需求是个变化的内容,在软件开发的各个阶段都有可能产生需求的变化,由此而引发了多种各具特色的软件开发方法(这些已超越了本文的范围)。本文所提之需求工程,只涉及软件开发中系统设计之前的阶段。

什么是需求工程?目前还没有统一而严格的定义。本文对需求工程的定义是:在软件开发的系统设计之前,采用人们能够认可而有效的原理、方法、工具及记录形式,准确、系统地描述待开发软件的必备功能、运作特征及有关约束的工程化方法。一般而言,它的内容有这么几部分:收集并分析用户需求、预测未来的变化(包括用户需求的变化及计算机技术的变化)、根据分析及预测对用户的需求作规格化的描述。

2. 需求工程的各个阶段

根据当前软件开发的特点,可以把需求工程分成几个相对独立(中间会有反复)的阶段:

(1)基本分析阶段。开发人员采用各种方法(现在主要是采用个别交谈加集体讨论的方法),以求能得到对用户需求的基本理解,包括基本知识和约束、限制以及对未来变化的预测。对该阶段的分析、记录,就形成了最原始的非形式化的用户需求描述。

(2)需求确定阶段。开发人员在有关需求理论的指导下,系统地最原始的非形式化的用户需求描述中整理出用户的客观需求及双方对未来变化的预测,尽量排除其中的非形式化特征(如二义性、不一致性等),并用规定的规格及形式记录成文(即需求规格说明书)。

(3)需求验证阶段。开发人员以需求规格说明书为基本输入,用快速原型、模拟、符号执行等手段向用户展现需求规格说明书所描述的系统的实际面貌及行为特

征,以获取用户对需求的修改及认定,以便达成双方最终认可的需求规格说明书(这也是甲、乙双方的“技术协议”)。

以上各个阶段往往要反复进行,最终目标是使开发人员做出的需求规格说明书能让“用户满意”。显然需求工程的中心是“用户需求”,其最终的结果是需求规格说明书。

为了满足软件开发的实际需要,需求规格说明书至少应包括两方面内容:功能需求和非功能需求。所谓功能需求,是指对待开发软件的功能描述(包括输入/输出、处理方式、处理所依据的方法理论等等);非功能需求,则是指一些内外环境的限制,如开发环境、开发过程、安全性、操作方式及所涉及的国家政策、法令等。

3. 需求工程面临的主要问题及其研究的核心

人们都认为需求分析在软件开发中极为重要,但到目前为止,开发人员在需求分析阶段所得到的(理论、方法及工具的)支持却最少。从工程的角度看,需求工程目前面临的主要问题有:

- (1)用户的需求往往包含着潜在的矛盾;
- (2)用户需求受时效的限制,而且会产生无规律的变化;
- (3)对需求的描述形式及获取需求方式的非形式化;
- (4)日常概念中的固有矛盾(如“有翅膀的是鸟”与“昆虫有翅膀不是鸟”等);
- (5)软件应用领域的复杂多变。

显然,根据需求工程的内容及其面临的问题,其研究的核心应该是需求规格说明技术。需求工程应解决以下几个根本问题:

①寻找能适应各种应用领域问题的理解与描述方法,以及产生需求规格说明的机制。

②需求规格说明的收集、表达、制作、维护及质量保证的方法。

③演示、验证需求规格说明的有效手段。

4. 需求规格说明方法

(1)人们日常分析问题的基本手段

人们在分析、理解复杂问题时,主要采用三种基本手段:分解法、抽象法和投影法。

分解法是将一个复杂问题分解成若干个子问题,并根据子问题复杂程度的不同,继续对子问题进行分解,直到最后得到的每一部分都足够简单。

抽象法则是先获取对问题的总体抽象了解(即高层认识),然后逐步细化。这是一种由一般到特殊的渐进过程。

投影法是将问题投射到不同的层面上,使人们能从不同的角度来理解、认识问题的各个方面,并借以互相验证。这种方法容易发现局部知识中的缺陷,对查明用户中不同群体需求的矛盾非常有利。

(2)需求规格说明方法必备的基本要求

①实用的需求规格说明方法必须具有上述三种基本手段,从而使软件开发人员能够充分理解用户的需求,各类人员间也能充分交流思想。

②应该支持增量式需求的确定及扩展。

③形成的需求规格说明应易于维护。

④能支持使用用户的专业术语,尽量少用软件术语。

(3)可用的需求规格说明方法

可用的需求规格说明方法有三大类:非形式化方法、形式化方法、基于知识表示的综合方法。这些方法可在不同的领域满足上述的部分要求,但能满足所有上述要求的实用方法还有待于进一步探寻。

①非形式化方法。用非形式化的步骤及手段,来获取用户的需求、描述需求规格说明。我们见得最多的是结构化分析方法(SA)及各种面向对象分析方法(OOA)。这些非形式化方法通常都具有分解、抽象、投影等基本手段,容易入门,也容易使用,而且支持使用用户的专业术语(SA要差一些)。但非形式化固有的特点又使得最终的需求规格说明的正确性及可维护性难以保证。显然,如果条件具备,人们会选择其他的方法。

②形式化方法。这是一种以数学为基础、用严格的数学方法描述目标系统特征的方法。目前能见到的只有两类:面向模型的方法和面向特征的方法。面向模型的方法是用数学结构(一系列的函数、关系、集合…)来直

接构造、定义目标系统模型及行为的方法。面向特征的方法则是通过给出系统应满足的公理集来描述系统的特征,间接定义目标系统的行为(有公理规格说明与代数规格说明之分)。形式化方法往往需要形式化的需求规格说明语言。形式化方法具有严格定义的分解、抽象、投影等机制,能够满足上述所有的基本要求。其缺点是:数学定义的方法一般开发人员不易掌握,文档及规模过于庞大。这些都限制了它的应用范围。

③基于知识表示的综合方法。这其实是人工智能在软件工程中的一种应用。它结合了形式化方法和非形式化方法的长处,用知识表示技术作形式化基础,用非形式化的框架给出需求规格说明的指导原则。这种方法需要一些智能化的辅助系统来检测、提示当前的状态。一般而言,这种方法能够满足上述所有的基本要求。它的前景取决于辅助系统的智能化程度。与形式化方法一样,该方法也存在着不易掌握和文档及规模过大的问题。

5. 需求规格说明语言

(1)对需求规格说明语言的基本要求

①应具有表达抽象、分割的机制及相关的指导原则。

②应具有足够的表达能力来描述实际问题的本质及不可避免的变化、例外等特征。

③语言本身应该易学、易用,产生的文档应该易读、易懂。

④应具有检测已有需求一致性的标准。

(2)目前能用的需求规格说明语言

目前能用的需求规格说明语言可分成四类:自然语言、结构化形为描述语言、形式化语言和半形式化语言。

目前用得最多的是自然语言。但对于大规模的需求工程来说,如果只用自然语言来描述复杂的问题及环境,后果是不会理想的。其原因不是语法/语义的复杂,而是自然语言中难以避免的二义性。它不能提供一致性的标准,很难保证每个人对所描述的内容都产生相同、一致的理解。这种非形式化的特征会使得开发人员(也包括用户)间难以产生精确的共识,很可能会因此而产生危险的后果。

结构化形式描述语言在一些方法(如PSL/PSA文本、一系列面向对象的模型及DFD等)中很有市场,它引入了结构图来表示需求,但也只能是部分的消除二义性,并没有本质上的变化。

形式化语言可以从数学理论出发给出需求的精确定义,似乎是一种理想的描述语言,但现实中还存在着大量

的多变性以及难以预料的例外情况,再加上形式化语言固有的不易学习性,这使得目前的形式化语言系统的应用效率大打折扣。显然,目前急需能针对各种情况的、高效的形式化语言解释器,以简化形式化语言的学习和应用。

半形式化语言可以看成是对以上语言的一种折中。它是在形式化框架中加入一些允许作非形式化描述的手段,提高学、用的效率。是一类前景看好的描述语言。

在实际应用中,一般都是混合着使用上述几种方法。我们经常可以看到,在一篇需求说明中,针对不同的加工,既有自然语言的说明,又有SA的描述。这种混合往往会给工作带来方便。

6. 需求工程的质量保证

我们可以从三个方面来确定这个问题。

(1) 需求规格说明的语法质量

应保证语法的正确性,保证描述模型所用的语句满足需求规格说明语言的语法规则,而且应在其定义的子集之内。在这方面应提供几种保证语法要求的机制:预防失误(预防在模型中加入错误语句)、检查错误(即在模型的制作中或制作后查出错误)、修改语法错误。前两者可以在需求规格说明语言中加以处理(几乎所有的计算机语言都或多或少地提供了这类机制),一般能做到半自动化。第三种却很难自动化。

(2) 需求规格说明的有效性和完备性(即语义质量)

有效性是指所描述模型的语句的正确程度及所用语句与问题的相关程度;完备性是指制作的模型包含领域中正确及相关语句的程度。显然这两项指标越高越好。但在实际中,达到完全的有效性和完备性是不可能的。最有实用价值的是“在一定可信度下的有效性及完备性”,即“可信有效性”和“可信完备性”。一个有用的可信有效性,表示所描述的模型多少有些“无用的废话”,但去除它们所得到的收益却抵不了因此而投入的开销。这反映了一种对“投入产出”所做的权衡。但这也只是个模糊的概念,因为目前还没见到度量有效性的手段。

(3) 需求规格说明的可理解性

实践说明,对需求模型的“完全”理解也只能是一种遥远的理想(现在很多问题都源于理解失误),因此也需要引入“可信度”。要特别注意这里所说的理解,是人脑对需求的理解。可以利用各种方法增强可理解性,如采用可视化模型、各种动画(及不可动的)图形和表格等。

7. 对现有工作的分类评注

需求工程的研究方向有两条路:形式化途径和非形式化途径。形式化途径在文献[1]中已有精辟的论述,这里只探讨非形式化的途径,主要有两类:结构化分析(SA)方法和面向对象的分析(OOA)方法。

SA方法是一种自顶向下逐步求精的功能分解方法。分析人员先作出高层的数据流图,在此基础上反复求精,得到更详细的加工处理及输入/输出,从而得到了一层新的数据流图,然后再以此为基础不断求精,又能得到更详细的新层数据流图。反复进行这个过程,直到对系统分解的足够详细为止。这种方法容易掌握,目前用的相当多。其缺点是对用户的专业术语支持的很不理想,与软件设计的距离过大,这些都直接影响了人员间的思想交流和软件开发。

OOA方法支持对现实领域的模拟,能够比较方便、准确地建立目标系统的模型,是目前在理论上叫得最响的方法。这种方法有三大类:

(1) 基于数据库的设计

这是一种在ER模型的基础上忽略实体上的函数过程的方法。优点是问题域中的对象容易定义,而且稳定,后面的数据库设计也比较直接了当。缺点是为理解问题所必须模型化的不少对象不能与数据库实体相对应。

(2) 基于面向对象的设计(OOD)

这种方法不去刻意区分需求对象和设计对象(其实这两种对象只是详细程度不同)。这种方法的优点是从需求到设计可以直接转换,而且对象容易定义。缺点是因为需求分析与设计所选择对象的标准不同,抽象、分割、投影的操作也不相同,使软件人员不易适应。

(3) 基于结构化的分析方法

这种方法力图使功能分解与实体分解合为一体,将数据流图的图表模型转换成以现实实体为基础的面向对象模型。目前还没见到该方法的应用报导。

参考文献

- [1] 钱家骅等,需求工程的形式化途径,《计算机科学》,1992,(19)6,44--49.
- [2] 吴会松,面向对象开发方法的基本要求及评价标准,《宁夏大学学报》(自然科学版),1996,17(增刊),79--83.
- [3] 吴会松,过程描述语言的定义及应用,《华北石油设计》,1995.2,58--60

(来稿时间:1996年12月)