

非规范软件需求管理

牟克典¹, 金 芝^{1,2}, 陆汝钊^{1,2}

(1. 中国科学院计算技术研究所, 北京 100080; 2. 中国科学院数学与系统科学研究院, 北京 100080)

摘 要: 对软件需求阶段的非规范信息进行有效管理有助于改进需求获取过程, 提高需求质量. 本文主要讨论了软件开发过程中常见的非规范需求并提出了以容忍不一致性的需求推理为核心的管理框架, 来有效管理模糊需求信息、冗余需求信息等与不一致信息密切相关的非规范需求信息.

关键词: 需求工程; 非规范需求; 不一致性

中图分类号: TP18 **文献标识码:** A **文章编号:** 0372-2112 (2004) 12A-247-04

Managing Non-Canonical Software Requirements

MU Ke-dian¹, JIN Zhi^{1,2}, LU Ru-qian^{1,2}

(1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China;

2. Academy of Mathematics and System Science, Chinese Academy of Sciences, Beijing 100080, China)

Abstract: It may be desirable to manage non-canonical software requirements to facilitate requirements development and improve the quality of requirements specifications during the requirements stage. To address this, the framework for managing frequent non-canonical requirements is provided. The technique for reasoning about requirements in the presence of inconsistency plays an important role in the framework. Moreover, it could be used to manage other non-canonical requirements associated with inconsistency, including vague requirements or redundant requirements.

Key words: requirements engineering; non-canonical software requirements; inconsistency

1 引言

需求工程的目的是获取完整、协调、明确的需求规格说明, 为软件工程的后继阶段, 如设计、编码、测试以及演化等提供良好的基础. 软件系统的需求信息对软件开发项目的成功具有重要影响, 有关研究表明绝大多数软件开发项目的失败都可归结到不合格的需求信息^[1]. 由不恰当的需求信息所导致的错误如果在软件测试甚至运行阶段才被检测到, 修复这些错误的代价是十分高昂的. 例如对同一错误而言, 在需求分析阶段的修复费用与在测试阶段的修复费用的比值在 JPL 介于 1:10 和 1:30 之间^[2], 而在 IBM 的 Rochester 实验室这个比值为 1:13^[3]. 从这个意义上讲, 在需求过程中完全根除或者尽可能减少需求错误的个数对降低软件测试甚至运行阶段的高昂维修费用具有关键作用^[4,5]. 获取优秀的软件需求规格说明是提高未来软件系统质量、控制开发成本、节约维修费用的有效手段.

尽管目前没有统一的形式化标准来区分“优秀”的需求描述和“不合格”的需求描述, 但是通常认为“优秀”的单个需求描述不仅是完整的、正确的、必要的、无歧义的, 而且是可行

的、可验证的以及被设置了优先级别的^[5,6,7]. 对于“优秀”的软件需求规格说明而言, 通常认为完整性、一致性、可跟踪性和可修改性是其应该具备的基本特征^[5]. 事实上, 对绝大多数软件开发项目来说, 获取完全准确的需求信息并不容易, 在需求抽取与分析过程中存在很多难以避免的困难和障碍^[8], 例如:

- (1) 在软件需求获取的初期阶段, 需求分析人员收集到的需求信息经常是零碎的、不系统的.
- (2) 客户经常对自己的需求仅有模糊的认识, 不知道如何明确表达或者适时作出决策.
- (3) 客户或者应用领域的专家都不习惯表达他们认为是常识的知识或信息, 使得提供给软件开发人员的需求信息可能是不完整的.
- (4) 在很多情况下, 软件开发人员对应用领域并不熟悉, 只能基于客户和领域专家提供的信息来抽取需求, 而客户和领域专家经常不知道如何以软件开发人员熟悉的术语和方式表达他们的需求信息, 他们之间的通讯鸿沟 (Communication Gap) 导致他们之间的相互误解.
- (5) 在多视点^[9]、多风险承担者参与的需求获取模式中,

收稿日期: 2004-09-02; 修回日期: 2004-11-22

基金项目: 国家自然科学基金重点基金 (No. 60233010); 重大基金 (No. 60496324); 国家重点基础研究发展规划项目 (No. 2002CB312004); 中国科学院知识创新工程重要方向项目; 中科院管理、决策与信息系统实验室资助

各风险承担者不同的技术背景、职责、利益立场等不可避免的导致需求获取中的分歧、矛盾以及冗余的信息等^[10,11]。

尽管有些当前的需求工程方法和技术可以解决或者缓解某些困难(例如基于本体的领域建模方法^[8]可以有效处理通讯鸿沟问题),但没有一种有效的方法或技术可以完全根除这些常见的困难和障碍。因此,在软件开发实践中我们经常不得不面对如下问题:

- (1) 模糊的需求信息。
- (2) 不完整的需求信息。
- (3) 冗余的需求信息。
- (4) 不一致的需求信息。
- (5) 有分歧的质量属性要求。
- (6) 需求优先级设置上的冲突^[12]等等。

在本文中,我们统称这些有问题的需求信息为“非规范需求”信息。这些非规范需求信息对软件工程的其它后继阶段和软件质量都有直接影响,例如:

- (1) 模糊的或者有分歧的需求信息增加了软件工程设计、编码、测试等阶段的不确定因素。
- (2) 不一致的需求信息既可能导致未来系统中的局部错误,也可能导致未来系统崩溃。
- (3) 需求优先级设置上的冲突可能影响整个项目开发计划的安排等等。

尽管当前的大多数软件开发技术都力图回避这些有问题的需求信息,但是为了避免遗漏潜在的需求,在软件需求规格说明中还是不得不记录这些信息并进行跟踪和修改。在这个意义上,需求工程获取完整、协调、规范的软件需求信息的目的并不是一蹴而就的,而需要经历若干不断修改和减少这些有问题的需求信息、不断精化需求规格说明的循环分析过程。

如果从软件的整个生命周期来看,非规范需求信息对软件开发过程的影响其实并不全是负面的。通常非规范信息是系统中由需求导致的错误的可能来源之一,对非规范需求信息进行有效管理,有助于在需求过程中尽可能减少需求错误、提高软件开发质量、改进软件开发过程。例如对不一致信息的有效管理不仅可以促进各分布式开发小组之间的沟通和交流,还可以防止对设计作出过早决策,确保各风险承担人的观点和立场都被考虑^[13,14]。

在这些非规范需求信息中,不一致信息的管理和利用已经被需求工程领域所广泛关注^[13~17]。其实不一致信息和上面所提到的其它非规范信息是紧密相连的,例如模糊的需求信息可能隐含着不一致信息,而一致信息的检测和分析过程中可能发现冗余的信息或者不完整的信息,也可能导致原来的确定信息变成需要重新确定的模糊信息。

在本文中,我们主要讨论软件开发中常见的非规范需求信息以及相应的管理策略和框架。本文的基本结构安排如下:第二节讨论软件开发中常见的非规范需求信息;第三节提出管理非规范需求信息的基本框架和相关策略;第四节对本文的工作与相关工作进行比较;最后对本文工作进行总结。

2 非规范需求

在引言部分我们其实已经使用了术语“非规范需求”来笼

统地表示我们所提到的一些通常被软件开发人员认为有问题的需求描述。在本节中我们对“非规范需求”分别给出一个广义的和狭义的定义。

定义 1 (广义的非规范需求(集)): 相对于一定的软件需求规格说明的撰写标准或规则,我们把不满足某些标准和规则的需求(集)称为广义的非规范需求(集)。

其实这个定义和软件开发过程中所采用的软件需求规格说明的撰写标准和规则相关,例如当我们采用 IEEE 推荐标准^[6]时,我们一般认为如果单个需求陈述不满足完整性、必要性等 7 项基本特性中的某些特性,我们就认为此需求是广义的非规范需求;而对于需求规格说明来说,如果不满足完整性或协调性,或者不可跟踪和修改,则认为是广义非规范的需求规格说明。值得一提的是判断需求陈述是否满足某些特性在很多情况下都需要借助于应用领域的领域知识。为了区别起见,我们称符合需求规格说明标准的需求为规范需求。

非规范需求的检测与修复是需求验证的主要任务,其中对不一致信息的管理是近来软件开发中特别关注的问题。在本文中我们以不一致信息的管理作为核心,主要讨论与不一致信息相关的常见的非规范需求,着重指冗余需求和模糊需求。冗余需求是需求描述集中不一致性的主要来源之一,而模糊需求是潜在不一致性的可能来源。因此我们在这里将非规范需求集的概念进行限制,定义狭义的非规范需求集。

定义 2 (狭义的非规范需求(集)): 需求集 S 是狭义的非规范需求集是指 S 中有模糊的需求描述,或者冗余的需求描述,或者不一致的需求描述。

下面我们讨论这三类非规范需求:

(1) **不一致需求**: 前面已经提到对于不一致信息的管理是近来需求工程领域比较关注的问题之一。不妨设 S 是基本的需求规格说明(或者部分规格说明), D 表示与特定应用场景相关的事实, E 是 S 对应的未来系统在应用场景 D 中的期望运行结果,则检测 S 在应用场景 D 中是否是不一致的是指检测是否 $\exists E$, 满足 $S \neq D$ 。

(2) **冗余需求**: 如果对于某个 $\emptyset \subset S_1$ 来说,若 $S \neq D \wedge E$ 且 $(S - S_1) \neq D \wedge E$, 或者 $S \neq D \wedge E$ 且 $(S - S_1) \neq D \wedge E$ 则我们可以认为 S_1 相对于应用场景 D 是可能的冗余需求集。

(3) **模糊需求**: 考虑模糊需求的基本出发点是尽可能避免需求信息的遗漏。尽管不一致信息有助于软件开发人员发现比较模糊的、不确定的需求信息^[14],但目前对不一致信息检测和分析的研究工作仅局限于考虑能明确表示的需求集,基本上忽略对模糊需求的考虑。其实模糊需求从本质上来讲是一种“可能”的需求,基于当前的信息很难确切的表达出来,并且随着需求的精化,模糊需求也逐渐向可明确表示的需求转化。特别要注意到当这些模糊需求转变为可明确表示的需求以后,很可能导致新的不一致信息。在一般情况下,对模糊需求信息可以通过如下两种策略进行处理:

(4) **TBD (To Be Determined) 策略**^[5]: 对于一些特别模糊或者分歧特别大的需求信息来说,很难在当前阶段作出相关的决定。处理这种情况最明智的策略就是暂时将这种信息标

记为 TBD,并建立可跟踪的 TBD 需求列表留待合适的开发阶段处理. TBD 需求可能隐含着潜在的不一致信息,反过来,对不一致信息的进一步分析可能导致某些涉及的需求转变为需要进一步确定的 TBD 需求.

(5) 缺省策略:此策略是软件开发中常用的表示变化性的策略,在通常情况下,对于可变化的属性来说,在软件开发过程中暂时推荐一个当前认为最可能的属性值作为缺省值.领域分析和建模是利用领域知识来缓解需求获取困难、增强需求复用的有效途径之一^[8].对于领域模型来说,一方面需要表示应用领域中类似系统之间的共同需求(共性),一方面又要具有一定的灵活性,能表示领域模型的变化性,以适应本领域不同的系统开发需要.在本文中我们采用缺省方式来表示领域模型中对应于变化性的需求,这样一方面表示这种需求没有最终确定,具有一定的不确定性,同时也表示基于当前的信息,我们可以暂时作出缺省决策.其实,即使没有领域模型,在软件开发过程中,为了避免遗漏需求信息,有经验的需求分析人员会以“一般情况下...”的方式向客户提出建议,如果客户不直接否决,则可以暂时以缺省策略处理这种建议.和 TBD 需求类似,我们可以建立可跟踪的缺省需求列表.如果缺省的需求信息和规范的需求信息相冲突,在很大程度上缺省的需求信息是被审查和修改的对象.

事实上,在需求精化过程中,对不一致信息的检测和分析经常伴随着规范需求集与模糊需求集之间如图 1 所示的转化关系.

对非规范需求进行有效管理,至少可以在以下几个方面改进软件开发过程、提高未来软件产品的质量、节约测试阶段的维修成本:

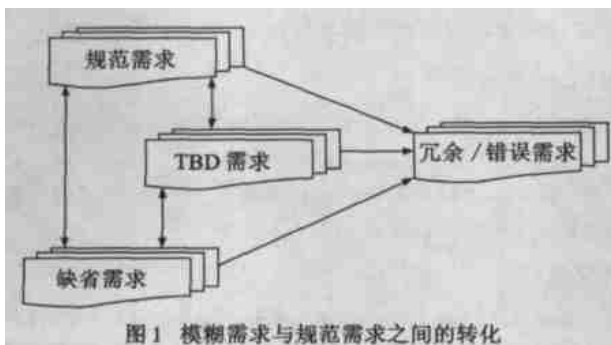


图1 模糊需求与规范需求之间的转化

- (1) 对模糊需求的考虑有利于避免遗漏潜在的需求信息.
- (2) 采用缺省策略可以利用当前不完全的信息作出比较灵活的决策.
- (3) 采用缺省策略有利于检测潜在的不一致信息;
- (4) 利用不一致信息来改进软件开发过程^[13,14].
- (5) 通过检测冗余需求来减少不一致信息.
- (6) 通过检测冗余需求来避免设计、编码、测试等阶段的开发资源浪费.

3 非规范需求管理框架

对非规范需求进行有效管理是利用非规范需求改进软件开发过程、提高软件产品质量、节约开发成本的关键.在本文

中我们提出以容忍不一致性的需求推理为核心的非规范需求管理框架,如图 2 所示.我们知道很多需求方面的问题都和特定的应用场景相关联^[17],因此我们提出的管理框架所关注的重点是基本的需求规格说明在特定的应用场景中的非规范性.

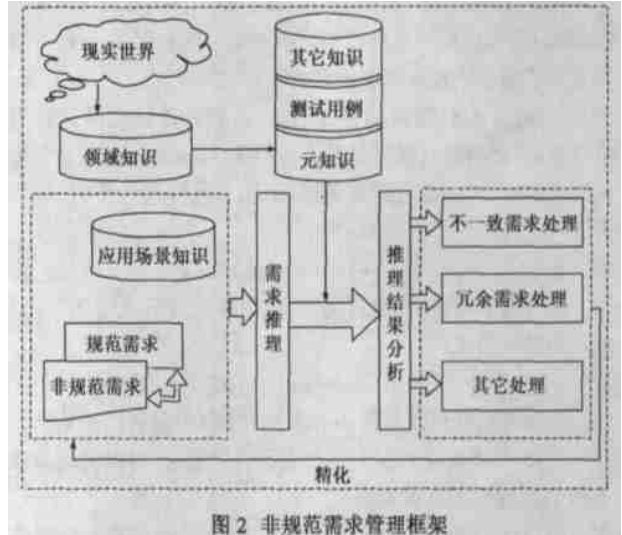


图2 非规范需求管理框架

在此管理框架中,我们首先在特定的应用场景中对需求进行推理,并且借助于与特定应用场景相关的知识(例如设计的需求测试用例^[18]等)对推理结果进行分析:

- (1) 如果推理结果中出现了不一致性,则我们可以采用 Nuseibeh 等提出的不一致性管理框架进行处理^[14].
- (2) 如果推理结果检测到冗余需求,则通过和客户的进一步协商来删除冗余信息.
- (3) 如果没有得到期望的推理结果,则可能遗漏了某些信息,相关的开发人员和客户需要进一步的协作来确定是否遗漏信息,以及遗漏了那些信息.

通过对推理结果的分析和处理,我们可以进一步对基本需求规格说明和相关辅助知识采取一定的行动.

- (4) 对基本的需求规格说明进行精化,包括规范需求和非规范需求的相互转化、补充新信息等.
- (5) 对用于辅助推理结果分析的特定场景相关知识进行修正和完善.

作为管理框架的核心,需求推理主要关注:

需求的形式化表示和推理工具:由于大部分需求描述的概念和术语都可以转化为经典逻辑,并且经典推理比较直观、自然,因此经典逻辑对需求形式化具有相当吸引力^[17].进一步考虑到非规范需求的特点,则适合需求推理的逻辑工具至少应该具有如下特点:

- (1) 超协调性:即要求在推理过程中容忍不一致性,弱化经典推理来禁止平凡推理,但经典推理的大部分良好特性得以保留.
- (2) 非单调性:主要出于对需求规格说明可修改性的考虑.
- (3) 可以跟踪和记录推理过程,为分析推理结果所涉及的需求信息提供基础.
- (4) 在语法上可以区分不同的非规范信息,例如不一致信

息、TBD 信息、缺省信息等,因此要求在目标语言中具有非规范信息的标识符号。

(5)在语义上可以对缺省和 TBD 问题赋以 $\{ True, false \}$ 以外的真值。

基于上述考虑,我们提出采用 APC (Annotated Predicate Calculus) 的改进框架来进行需求推理^[18],这是因为 APC 逻辑具有如下适合非规范需求推理的特征:

(1)通过 APC 逻辑的目标语言,可以在语法上形式化表示非规范需求,并且通过目标语言中的 Annotation 来区分非规范需求的种类。例如对需求描述“出租车允许进入小区”可以表示

$$taxi(x) \quad enter(x) : T,$$

而对需求描述“一般情况下,出租车允许进入小区”可以表示为如下公式:

$$taxi(x) \quad enter(x) : dT,$$

这样我们就可以直接从语法表示上区分缺省需求信息。

(2)APC 逻辑在语义上也扩充了真值集合,可以满足对缺省和 TBD 问题的真值要求。

(3)APC 逻辑可以实现容忍不一致性的推理和非单调推理,可以满足非规范需求推理所需的超协调性和非单调性。

(4)APC 逻辑的证明理论和经典推理具有很大的类似性。

(5)我们可以进一步对 APC 逻辑进行改造,通过对前面附加可唯一标识的标记来记录和追踪推理过程,例如我们可以对 $taxi(x) \quad enter(x) : dT$ 附加唯一的标志,通过 $\{ a \} :: taxi(x) \quad enter(x) : dT$ 来标识。

推理结果的分析 满足上述要求的需求推理在得到推理结果的同时,也记录了推理过程所涉及的需求信息,为推理结果的分析 and 基本需求规格说明的精细化提供了基础。事实上,对推理结果的进一步分析还需要一些辅助知识:

(1)与整个应用领域相关的元知识,例如一致性规则、需求的相对重要性等可以辅助确定不一致信息的来源。

(2)与特定应用场景相关的知识,例如针对特定应用场景设计的需求测试用例,其中的期望响应部分可以作为检测冗余需求的触发点^[18]。

(3)其他知识,例如项目决策者的原则、投资人的偏好等。

基于对推理结果的分析,我们可以对需求规格说明进行精细化,同时对辅助分析推理结果的相关知识进行更新,为精细化以后的需求推理提供基础。

4 比较与讨论

需求抽取与分析过程中不可避免的困难和障碍使得需求规格说明在很大程度上仅是反映项目需求的基准线,不得不包含一些非规范的需求信息。随时彻底根除需求规格说明中的各种非规范性往往是事与愿违的,在有些情况下甚至是不现实的。如果转换角度,把非规范需求信息看作检测或者识别各种潜在需求问题的触发点,则可以通过对非规范需求信息的管理和利用来改进需求过程。

不一致需求是软件开发中特别关注的非规范需求之一, Nuseibeh 等已经提出了关于软件开发中不一致性的管理框

架^[14]。事实上,其它各类非规范需求信息和不一致性具有密切联系,将其它非规范需求信息管理和不一致性的管理相割裂不仅可能忽略潜在的不一致性,而且可能增加了不一致性处理的困难。例如,如果我们不允许 TBD 问题,则当风险承担者对不一致信息处理上分歧很大,暂时无法达成统一意见时,可能需求过程会陷入僵局,无法继续进行。因此在本文中我们不仅关注不一致信息,同时也关注模糊信息等其它非规范需求信息。这是本文工作和不一致性管理框架^[14]最大的区别。

另一方面,不一致需求的管理仍然是非规范需求管理的关键问题。我们上面已经提到容忍不一致性的需求推理是非规范需求管理的核心,换言之,我们的框架其实就是在更广范围内的不一致性管理框架,而不一致性管理框架可以看作非规范需求管理框架的一部分。在这个管理框架中,我们可以形式化表示模糊需求,在检测不一致需求的同时也可以检测冗余需求。在这个意义上,本文工作也是对不一致信息管理的发展和改进。

5 结论

非规范需求信息尽管给软件工程的各个阶段带来了负面影响,但是如果对非规范需求进行有效管理,利用非规范信息作为改进需求抽取与分析过程的出发点,可以有效减少需求错误与缺陷,对提高软件产品质量、控制开发成本、降低维修成本方面具有重要意义。本文主要讨论了软件开发过程中常见的非规范需求,即不一致需求、模糊需求和冗余需求。事实上,模糊需求和冗余需求和不一致需求具有密切联系,基于这样的观点,我们最后提出了以容忍不一致性的需求推理为核心的非规范需求管理框架。在某种意义上,本文提出的非规范需求管理框架可以看作一种扩展的不一致性管理框架。

作者简介:

牟克典 男,1975 年生于甘肃会宁,博士,主要研究领域为需求工程,不确定推理。E-mail:mukedian@ict.ac.cn。

金 芝 女,1962 年生于安徽休宁,博士,研究员,主要研究领域为基于知识的软件工程,需求工程,人工智能。

参考文献:

- [1] Glass R. Software Runaways[M]. Harlow:Prentice Hall,1998,21.
- [2] J C Kelly, J S Sherif, J Hops. An analysis of defect densities found during software inspections[J]. J Syst Softw,1992,17(2):111-117.
- [3] S Kan. Metrics and Models in Software Quality Engineering[M]. Massachusetts:Addison-Wesley Publishing Company, Inc,1995.
- [4] P Grunbacher, M Halling, S Biffl. An empirical study on groupware support for software inspection meetings[A]. the Proceedings of 18th IEEE Automated Software Engineering[C]. Montreal Canada,2003. 4-11.
- [5] K E Wiegers. Software Requirements[M]. Microsoft Press,2003.
- [6] IEEE. 1998, IEEE Std 830 - 1998: IEEE Recommended Practice for Software Requirements Specifications[S]. Los Alamitos, CA: IEEE Computer Society Press,1998.
- [7] Davis Alan M. Software Requirements: Objects, Functions, and States[M]. Englewood Cliffs, NJ:PTR Prentice Hall,1993.

(下转到 221 页)

离散小波变换四种常用的融合算法为例进行了多类图像的融合质量评价,实验结果表明该方法得到的客观评价指标主观评价的结果是一致的,是一种实用的、有效的、通用的融合图像质量评价方法,可以为不同场合下选择不同的算法提供依据。

参考文献:

- [1] 李树涛,王耀南,张昌凡.多传感器图像融合的客观评价与分析[J].仪器仪表学报,2002,23(6):651-654.
- [2] 阳方林,郭红阳,杨风暴.像素级图像融合效果的评价方法研究[J].测试技术学报,2002,16(4):276-279.
- [3] Gemma Piella. New quality measures for image fusion[A]. The 7th International Conference on Information Fusion[C]. Stockholm, Sweden, June 28 to July 1, 2004. 542-546.
- [4] 李玲玲,周成平,丁明跃,张天序.基于易操纵金字塔的多传感器图像融合[J].计算机工程与应用,2003,22:24-26+41.
- [5] 姜庆娟,谭景信.像素级图像融合方法与选择[J].计算机工程与应用,2003,25:116-120.
- [6] 钟伟才,刘静,刘芳,焦李成.基于形态学 top-hat 算子的多传感器图像融合[J].电子学报,2003,31(9):1415-1417.
- [7] 洪靖云,张桂林,洪汉玉,杜峥.基于高频通道的图像融合算法[J].红外与激光工程,2003,32(5):509-512.
- [8] 毛士艺,赵巍.多传感器图像融合技术综述[J].北京航空航天大学学报,2002,28(5):512-518.
- [9] 夏明革,何友,唐小明,夏仕昌.多传感器图像融合综述[J].电光与控制,2002,9(4):1-7.
- [10] 吉书鹏,丁晓青.可见光与红外图像增强融合算法研究[J].红外与激光工程,2002,31(6):518-521+544.
- [11] 李玲玲,周成平,丁明跃,张天序.一种基于多尺度分解的多传感器图像融合方法[J].华中科技大学学报(自然科学版),2003,31(4):82-84.
- [12] 夏明革,何友,唐小明,关欣.像素级图像融合方法分类与比较[J].火力与指挥控制,2002,27(3):1-4.
- [13] Zhou Wang, Bovik A C. A universal image quality index[J]. IEEE Signal Processing Letters, 2002, 9(3):81-84.
- [14] Xydaes C, Petrovi V. Objective image fusion performance measure[J]. Electronic Letters, 2000, 36(4):308-309.

作者简介:

胡良梅 女,1974年4月出生于安徽合肥,合肥工业大学博士研究生,讲师,主要研究方向:信息融合、图像理解、智能信息处理. E-mail:liangmeihu@sohu.com.

高隽 男,1963年10月出生于安徽淮南,中国科学技术大学博士,合肥工业大学教授,博士生导师,主要研究方向:图像处理与分析、模式识别与人工智能、神经网络理论及应用、光电信息处理、智能信息处理.

(上接第 250 页)

- [8] Lu R, Jin Z. Domain Modeling-Based Software Engineering[M]. Kluwer Academic Publishers, 2001.
- [9] A Finkelstein J, Kramer, B Nuseibeh, L Finkelstein, M Gøedicke. Viewpoints: A framework for integrating multiple perspectives in system development [J]. International Journal of Software Engineering and Knowledge Engineering, 1992, 2(1), 31-58.
- [10] A Silva. Requirements, Domain and specifications: A viewpoint-based approach to requirements engineering[A]. Proceedings of ICSE02[C]. Orlando, Florida, USA, May, 2002. 94-104.
- [11] S Easterbrook, M Chechik. A framework for multivalued reasoning over inconsistent viewpoints[A]. In Proceedings of International Conference on Software Engineering (ICSE '01) [C]. Toronto, Canada, May 2001. 411-420.
- [12] 牟克典,金芝,陆汝钊.视点合成中重叠需求的不一致优先级处理[J].计算机学报,2004,27(10):1379-1387.
- Kedian Mu, Zhi Jin, Ruqian Lu. Handling inconsistent priorities of overlapped requirements in merging viewpoints[J]. Chinese Journal of Computers, 2004, 27(10):1379-1387.
- [13] B Nuseibeh, S Easterbrook, A Russo. Leveraging inconsistency in software development[J]. IEEE Computer, 2000, 33(4):24-29.
- [14] B Nuseibeh, S Easterbrook, A Russo. Making inconsistency respectable in software development[J]. Journal of Systems and Software, 2001, 58(2):171-180.
- [15] A Finkelstein, D Gabbay, A Hunter, J Kramer, B Nuseibeh. Inconsistency handling in multi-perspective specifications[J]. Transactions on Software Engineering, 1994, 20(8):569-578.
- [16] G Spanoudakis, A Zisman. Inconsistency management in software engineering: Survey and open research issues[A]. S K Chang, editor. Handbook of Software Engineering and Knowledge Engineering [C]. World Scientific Publishing Co., 2001. 329-380.
- [17] A Hunter, B Nuseibeh. Managing inconsistent specification [J]. ACM Transactions on Software Engineering and Methodology, 1998, 7(4):335-367.
- [18] Kedian Mu, Zhi Jin, Ruqian Lu. A logical approach to testing requirements[A]. to appear in Proceedings of IASTED conference on SE05 [C]. Zurich, Switzerland ACTA Press, 2005.