

# 多视点中不一致性的处理：一种基于认知逻辑的方法

江 敏 毋国庆 刘小丽

(武汉大学计算机学院 武汉 430072)

**摘 要** 衡量一个软件系统是否成功,一个基本的要素就是看该软件系统是否满足了用户的要求。软件系统需求工程(RE)就是一个通过标识涉众及其需求并将其文档化以便于进行分析的过程。RE是一个跨学科,以人为中心的过程。多视点需求工程是需求工程的一个分支。多视点需求工程就是希望复杂系统中的不同参与者分别从自己的角度出发对预期系统进行描述,从而形成更完备的需求规约。由于多视点方法的这种特性,导致多个涉众有可能对同一问题进行描述,从而形成重叠的需求。这些重叠的需求就是涉众之间的公共知识,对公共知识的不同解释是导致需求规约中不一致问题的根源。不一致性是多视点需求工程中一个必须解决的问题。本文提出了基于问题域的多视点需求建模框架,并利用认知逻辑对该框架进行解释和推理,希望可以用形式化的方法来发现和解决存在于不同视点间的不一致性。

**关键词** 多视点,公共知识,不一致,认知逻辑

## Inconsistency Handling of Viewpoints: An Epistemic Logic-Based Approach

JIANG Min WU Guo-Qing LIU Xiao-Li

(School of Computer, Wuhan University, Wuhan 430072)

**Abstract** One prima factor of measuring achievement of a software system is whether it satisfies the needs for consumers. Software Requirement Engineering (RE) is processes that identifies stakeholders and document those requirements, which can convenient for analyzing. RE also is an inter-discipline and stakeholder-center process. Viewpoints requirement engineering is a branch of RE. Viewpoints approach hopes that stakeholders in a complex system should describe it from their own perspectives and then a more complete requirement specification can be generated. Just because of this characteristic, several stakeholders maybe describe a same problem. These overlapping requirements are the common knowledge among stakeholders. The different interpretations of that common knowledge are source of inconsistency. This paper puts forward requirement modeling framework based on problem-domain and viewpoints. We interpret and reason it with epistemic logic in order to achieve the following goals-to discover and solve the inconsistency which exists in different viewpoints formally.

**Keywords** Viewpoints, Common knowledge, Inconsistency, Epistemic logic

## 1 前言

构建复杂的模型往往牵涉到许多的参与者。这些参与者对希望描述的模型有着不同的观点和视角。由于参与者所承担和扮演的责任与角色的不同,导致他们的观点和视角都是部分和不完备的。我们称这些参与者及其观点的组合为多视点,多视点的研究主要在于视角之间的关系、视角与参与者之间的关系和参与者之间的关系。

多视点是群体工作的一般特征,而开发复杂的软件系统是一种典型的群体工作。多视点的思想在软件工程,尤其是需求工程中得到广泛运用。多视点需求工程方法试图以一种更加全面的方式理解涉众对软件系统的期望,所有与目标系统相关的人员,依据其职责、经验与技能,使用他们所熟悉的表达方式相对独立地对系统的特性进行陈述。

多视点的结构本质上是松散的,同时各视点间又有联系。由于多视点的这种特性,不可避免地导致某些需求会产生交织与散乱,这些交织与散乱的需求会在需求获取和分析阶段导致不一致和冲突。

运用多视点方法来解决一个大型项目中的实际问题是往往需要考虑如下三个方面的问题:

- 选择一个合适的多视点模型
- 标示出视点
- 管理多视点的信息

本文在研究了已有的多视点需求工程方法后,在它们的基础上提出了基于问题域的多视点建模框架(Problem Domain-based Viewpoints Modeling Framework),并使用认知逻辑对其进行解释,通过对这些认识逻辑公式的推理,可以找到各视点间存在的公共知识。进而提供一种解决多视点需求中的不一致性问题的方法。

## 2 相关研究

如何处理不一致性是多视点需求工程中的难点,至今还有许多研究人员提出了多种方法试图解决这个问题。这些研究工作包括:

A. Finkelstein<sup>[1]</sup>提出利用经典和基于 Action 的时序逻辑进行处理; Zave 和 Jackson<sup>[2]</sup>提出使用经典逻辑来表示部分的规约。规约的组合看作是这些逻辑断言的合取,当且仅当这些合取公式被满足时,这一些部分的规约才被认为是一致的; Balzer<sup>[3]</sup>提出通过放松约束和容忍不一致来处理某些特殊的不一致。

江 敏 博士研究生,研究方向为形式化方法、需求方程;毋国庆 教授,博导,研究方向为需求工程、知识工程和形式化方法。刘小丽 博士研究生,研究方向为需求工程。

Wallis<sup>[4]</sup>, Jackson<sup>[5]</sup>, Boiten 和 Derrick<sup>[6]</sup> 提出使用 Z 来解决这个问题; Leduc<sup>[7]</sup>, Khendek<sup>[8]</sup>, Ichikaw<sup>[9]</sup> 和 Steen 以及 H. Bowman<sup>[10]</sup> 使用 LOTOS 来进行研究。

S. Easterbrook 和 M. chechik<sup>[11]</sup> 使用多值逻辑来描述每一个视点, 并提出了  $\chi$ bel 框架和实现了  $\chi$ check 模型检查器来对其进行推理和效验。M. Sabetzadeh 和 S. Easterbrook<sup>[12]</sup> 提出了使用范畴论的方法来表示和分析基于图的多视点中的不一致性, 这项工作的主要贡献在于提出了一种抽象机制来合并不一致和不完备的视点, 并定义了句法不一致的概念。

不同的多视点模型具有不同的侧重点, 也就是说在不同的情况下应该使用不同的多视点模型。如果你关注于需求求精和抽象的需求定义, 那么可以使用 Leite 或 Kotonya 和 Sommerville 提出的方法; 如果你关注于细节的设计和冲突的解决那么由 Finkelstein、Nuseibeh 和 Easterbrook 等人提出并开发的方法可能更合适。CORE 中使用的方法更适合于传统的开发方法。

不一致性往往与认知特性有关, 需求规约中反映的就是涉众的知识、信念和意图。例如, 当系统的参与者对存在于他们中间的公共知识有着不同的理解和解释时, 不一致和不完备的问题就会发生。也就是说如何发现存在于涉众之间的公共知识应当是处理以上问题的关键。但是这些在以上的多视点模型中均不能够很好地表示出来, 它们在处理这个问题上显得力不从心。

### 3 基于问题域的多视点建模框架

Zave<sup>[2]</sup> 给出了一个关于需求工程非常好的定义: “需求工程是软件工程的一个分枝, 它关注于现实世界的目标、功能以及与软件系统相关的约束。同时它也关注于用来精确规约软件行为的上述相关因素的关系和演变。”

这个定义强调了“现实世界的目标”——软件系统开发的目的, 并说明了在软件需求中关注的三个对象——功能需求、非功能需求和目标是一个整体。

不同的人员在大型软件开发过程中扮演着不同的角色, 他们看待问题的层次是有差别的。例如领域工程师、架构分析师和高层管理人员主要考虑宏观上的问题; 描述系统的结构、性能和成本等特征。编程人员、终端用户则更多地考虑微观上的问题。他们会根据实际情况组合在一起形成若干个团队, 团队内部不同层次的人员之间相互影响并交换知识, 共同形成对目标系统的期望——最终的需求规约。每一个参与者对系统的需求是否合理, 不仅取决于其自身, 还与他所处的团队以及团队中其他人员的需求有关。

考虑以上两方面情况, 我们要提出的多视点建模框架 (PDVMF) 应能体现以下两方面特性:

1) 多视点模型应是二维的。也就是说软件系统首先应该是由若干问题子域组成, 并且一个问题子域中可能存在着若干视点, 它反映不同类型人员的关注点有所不同。

2) 必须显式地表明目标、功能需求和非功能需求之间的联系。将它们看作一个有机整体, 使用户的需求更加结构化。

视点在需求规约中代表了涉众对目标系统的期望。在需求获取的过程中判断目标系统是否能够满足用户的要求, 就是看用户提出的目标是否被实现。目标的实现由功能性需求作为支撑, 非功能性需求对功能性需求进行约束。当与某一目标有关的功能性需求和非功能性需求均得到满足时, 该目标被认为已实现了, 即  $FUN, Nfun = Goal$ 。同时, 通过对某一

目标进行追踪, 应当可以推导出与之相关的功能性需求和非功能性需求, 即  $Goal \vdash Fun, Nonfun$ , 我们用图 1 来表示三类实体之间的关系。

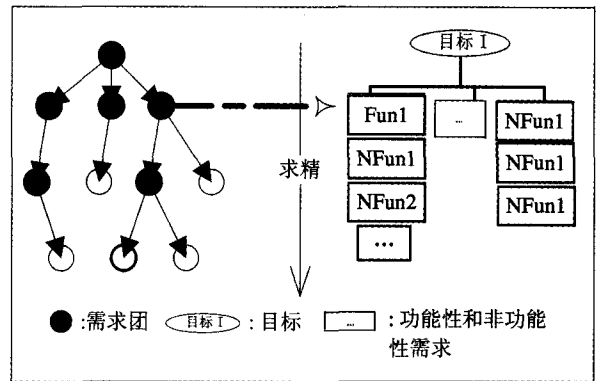


图 1 具有求精关系的需求团

我们将这三类实体的组合称为一个需求团, 它是 PDVMF 的核心。PDVMF 直接来源于以前的多视点研究工作<sup>[13]</sup>, 它使用模板来收集和表达涉众的需求。但与其它的方法的区别在于, 它要求涉众必须显式地声明需求块。也就是说无论涉众处在哪个问题域, 扮演何种角色, 必须使用他习惯的表达方式来陈述目标系统应达到的目标以及与该目标相关的功能和非功能性需求。

问题域由领域专家及系统分析人员进行划分。在确定问题子域之间的关联关系、各问题子域涉及哪些人员以及各人员之间的协作关系之后, 涉众可以根据自己所关注的问题子域, 使用视点模板来描述其需求。模型示意图如图 2 示:

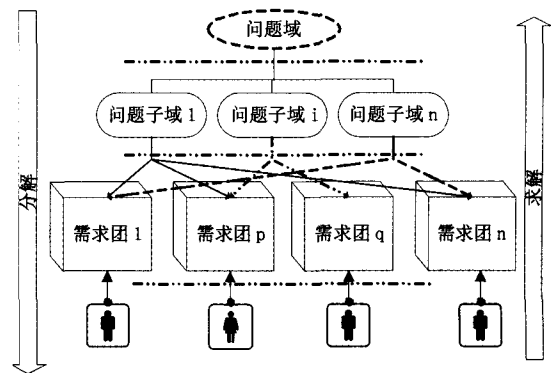


图 2 基于问题域的多视点建模框架

### 4 基于认知逻辑的不一致性推理

#### 4.1 认知逻辑简介

认知逻辑是研究知识和信念的逻辑。“知道”是认知逻辑研究的一个重要对象, 认知逻辑就是对“知道”的含义进行形式化的研究。它诞生于 20 世纪 60 年代, 在计算机科学、人工智能、博弈论、决策论、经济学和军事学诸多领域需要的推动下, 形成了若干既具有重大理论意义, 又具有实际应用价值的分支学科。

在这个世界上, 认识的主体不只是一个单一的人, 也不是有着统一认识的整个人类, 而是一群有着不同知识的个体。在多主体系统中, 主体之间的互知是推理的重要依据。互知是指一个认知主体不仅知道某一事件, 并且能够认识到自己的这种状态(自知), 同时还知道与其相关的主体对这一事件

的看法。而这一推理能力在需求工程,尤其是多视点需求工程中是非常重要的。

Halpern 和 Moses<sup>[14]</sup>对认知逻辑进行了很好的分析,他们指出,Kripke 的可能世界模型是探索认知逻辑的有效工具。

Kripke 模型是一个  $m+2$  元组  $M = \langle W, R_1, R_2, \dots, R_m, V \rangle$ ,其中  $W$  是全体可能世界集, $V$  是命题集  $\Phi$  在  $W$  中的每个可能世界  $W_i$  上的真值指派。对每个  $i, R_i$  是  $W$  上的二元关系,它的含义是:若  $\alpha R_i \beta$  为真,则从可能世界  $\alpha$  中的个体  $a$  的观点看, $\beta$  是一个可到达的现实世界。

$K$  为一个模态算子, $K_i p$  表示:主体  $a_i$  知道事实  $p$ ,而它的含义是  $p$  在所有  $a_i$  可达的世界中为真。反之,若  $p$  在至少一个可达世界中为假,这称为  $a_i$  不知道  $p$ ,记为  $\neg K_i p$ 。若在所有  $a_i$  可达世界中为假,则称  $a_i$  知道  $\neg p$ ,记为  $K_i \neg p$ 。从技术的角度考虑,我们采用称为  $S_3$  的系统。在一个有  $m$  个主体的系统中,它由如下公理和规则组成:

- (A1) 全体命题重言式
- (A2)  $(K_i \varphi \wedge K_i (\varphi \rightarrow \psi)) \vdash K_i \psi, i=1, \dots, m$
- (A3)  $K_i \varphi \vdash \varphi, i=1, \dots, m$
- (A4)  $K_i \varphi \vdash K_i K_i \varphi$  (正内省定律)
- (A5)  $\neg K_i \varphi \vdash K_i \neg K_i \varphi$  (负内省定律)
- (R1)  $\frac{\varphi \wedge \varphi \rightarrow \psi}{\psi}$  (三段论)
- (R2)  $\frac{\varphi}{K_i \varphi} (i=1..m)$  (必然律)

为了简要说明认知逻辑,我们举一个例子说明。

令  $M$  为一个 Kripke 模型,其中  $W = \{s_1, s_2, s_3\}$ ,主体集  $A = \{Alice, Bob\}$ ,命题集  $\Phi = \{p\}$  以及可达关系  $R_{Alice}$  (实线)和  $R_{Bob}$  (虚线)。如图 3 所示。

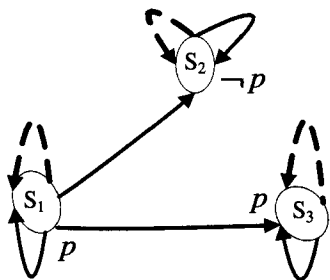


图 3 模型 M 的 Kripke 结构

我们令  $p = \text{“完成模块 A 需要 30 人/月”}$ ,由此可以推导出:

- (1)  $(M, s_1) \models p$
- (2)  $(M, s_1) \models \neg K_{Alice} p \wedge \neg K_{Alice} \neg p$

这说明在  $s_1$  中 Alice 不知道  $p$  是否成立,因为在  $s_1, s_3$  中 Alice 知道“完成模块 A 需要 30 人/月”为真,但是在  $s_2$  中得到了相反的结论,因此 Alice 在  $s_1$  这个状态中不能判断该命题的真性。

#### 4.2 PDVMF 的认知逻辑解释

我们将 PDVMF 看作一个双层的 Kripke 模型,问题域层看成一个 Kripke 模型,它由多个问题子域及其相互之间的关系组成。同时某一问题子域对应的视点层也是一个 Kripke 模型。每一个问题子域或视点相当于一个可能世界。涉众提出的需求看成是一组认知逻辑命题集合,命题是否为真,依赖于其可达的可能世界中的真值。

视点层用 3 元组  $M = \langle VD_{pd_i}, \pi_v, \mathcal{R}_v \rangle$  表示,其中  $VD_{pd_i} = \{vd_{pd_i}^1, vd_{pd_i}^2, \dots, vd_{pd_i}^n\}$  表示在问题子域  $pd_i$  中的各个视点。

$\pi_v: VD \rightarrow (P \rightarrow \{true, false\})$  表示在每个视点中原子命题的赋值函数, $\mathcal{R}_v \subseteq VD \times VD$ ,称为可达性关系,表示同一问题域中各涉众之间的依赖关系。

在这个结构上定义了两个模态词  $K_i \varphi$  和  $M_i \varphi$ ,其中  $K_i \varphi$  读作“视点代理  $i$  知道  $\varphi$ ”。 $M_i \varphi$  表示  $\neg K_i \neg \varphi$ ,它表示“视点代理  $i$  不知道  $\varphi$  是否为真”。为了更方便的对我们的模型进行推理,我们在不失直觉的情况下添加规则如下:

- (R3)  $Goal \vdash Sub_1 Goal \wedge Sub_2 Goal \wedge \dots \wedge Sub_n Goal$ 。  
当目标与子目标之间是 AND 关系时;
- (R4)  $Goal \vdash Sub_1 Goal \vee Sub_2 Goal \vee \dots \vee Sub_n Goal$ 。  
当目标与子目标之间是 OR 关系时;
- (R5)  $Goal_x \vdash (Fun_x^1 \wedge \dots \wedge Fun_x^i \wedge NF_x^1 \dots \wedge NF_x^i) \vee (\dots) \vee (Fun_x^e \wedge \dots \wedge Fun_x^n \wedge NF_x^e \dots \wedge NF_x^n)$ 。表示由某一目标可以推导出与其相关的功能性和非功能性需求。

这样视点模板中的需求团可以转换为认知公式表示。同样在问题域层也是一个 kripke 模型  $S = \langle PD, \pi_D, \mathcal{R}_D \rangle$ :

- $PD = \{pd_1, pd_2, \dots, pd_n\}$ ; 表示问题子域集。
- $\pi_D: PD \rightarrow (P \rightarrow \{true, false\})$  是被某一问题域内所有涉众认为是合理的命题的赋值函数。
- $\mathcal{R}_D \subseteq PD \times PD$ ; 表示问题子域之间的依赖关系。

我们可以将已经在视点层验证过的需求再次转换为认知逻辑命题,在问题域层进行推理,从而发现这些需求在更高的一个层次上是否存在分歧,以形成最终的需求规约。

#### 4.3 公共知识

所谓公共知识就是多个主体都知道的一些事实,而重叠的需求就应当是存在于涉众之间的公共知识。在多视点模型中,不一致性往往来自于重叠的需求。当涉众对于这些公共知识存在着不同的理解时,不一致性就会发生。因此发现公共知识应该是有效管理不一致性的手段。

为了实现对公共知识的推理,我们引入了两个新算子  $E^{pd_i}$  和  $CK^{pd_i}$ 。 $E^{pd_i} \varphi$  表示“在问题域  $pd_i$  中每个视点代理均知道  $\varphi$ ”及  $CK^{pd_i} \varphi$  表示“ $\varphi$  是问题  $pd_i$  中的公共知识”,它们可以表示为:

$$E^{pd_i} \varphi = K_{vd_i^1}^{\varphi} \wedge K_{vd_i^2}^{\varphi} \wedge \dots \wedge K_{vd_i^n}^{\varphi}$$

为了说明新引入的两个算子的语义,我们假设:有视点层模型  $M = \langle S, \pi, \mathcal{R} \rangle$ ,令  $s, t \in S$ 。

- (1) 我们用  $s \xrightarrow{R_i} t$  表示从  $s$  由  $R_i$  一步可达  $t$ ;
  - (2) 用  $s \rightarrow t$  表示  $(s, t) \in \mathcal{R}$
  - (3) 一步可达关系的自反传递闭包关系记为  $s \xrightarrow{*} t$ ;
- 那么我们规定

$$(M, s) \models E^{pd_i} \varphi \Leftrightarrow \text{对所有满足 } s \rightarrow t \text{ 的 } t, (M, t) \models \varphi \text{ 成立;}$$

$$(M, s) \models CK^{pd_i} \varphi \Leftrightarrow \text{对所有满足 } s \xrightarrow{*} t \text{ 的 } t, (M, t) \models \varphi \text{ 成立。}$$

注意,在问题域内主体不明确知道问题域中所包含的主体的确切情况时, $E^{pd_i} \varphi$  和  $CK^{pd_i} \varphi$  是不相同的。

利用这两个新引入的算子,我们可以对用认知逻辑表达的用户需求进行推理,从而判断出其所在的问题域内所有涉众是否知道该需求或者该需求是否为该问题域中的一个公共知识。

#### 4.4 不一致性的处理

在需求建模过程中不一致性可能表现为以下两种类型：

1) 涉众对某一需求有着截然不同的看法, 我们称之为绝对不一致性。例如: 在视点 A 中认为“日志”是必须的, 而与其相关的视点 B 则认为不需要此功能;

2) 涉众对某一需求有着近似的看法, 但在细节上有所差异。如视点 A 认为“必须有日志功能, 并且日志的记录过程必须在 1 秒钟内完成”。视点 B 也认为“必须有日志功能, 但可以不必在 1 秒钟内完成”。我们称之为相对不一致性。

我们通过对认知逻辑的研究发现绝对不一致性, 可以通过效验如下公式:

$$\neg K_i p \wedge \neg K_i \neg p$$

的可满足性。该公式的含义是: 视点  $i$  知道在某个相关的视点中  $p$  为假, 同时在另一个视点中  $p$  又为真。如果该公式是可满足的, 则表明绝对不一致性存在。

例如在图 3 所示的 kripke 模型中。我们可以推出  $(M, s_1) \models \neg K_{Alice} p \wedge \neg K_{Alice} \neg p$ 。因为在  $s_1, s_3$  中 Alice 知道  $p$  为真, 但是在  $s_2$  中得到了相反的结论, 因此 Alice 知道  $s_1, s_3$  和  $s_2$  对  $p$  的看法是不一致的。通过这种方法我们可以将不一致性的问题转化为求解一个逻辑公式的 SAT 问题, 从而提供了一个自动化发现不一致需求的手段。

由于相对不一致性的特性, 导致它难以完全用形式化的方法发现。我们解决这一类问题的思路是: 相对不一致性应存在于重叠需求中, 利用 CK 算子找出所有的重叠需求(也就是涉众之间的公共知识), 然后涉众一起对这些重叠需求进行讨论, 从而发现他们对这些需求的理解是否存在差异, 如果差异的确存在, 那么说明该需求就是不一致的, 这样就可以利用协商和放宽约束等方法来解决。利用这种方法可以使涉众的讨论更加具有目的性, 从而提高需求获取的效率。

### 5 示例

下面我们利用一个简化了的在线图书馆系统的例子来说明 PDVDM 模型及如何利用认知逻辑来发现需求中的不一致性。

为了研究的简单, 我们只考虑在某一个问题域中的涉众, 且对其需求进行了简化。假设在问题域  $D$  中涉及到读者 A, 图书管理员 B 和图书供应商 C。其相互之间的关联关系如图 4 所示:

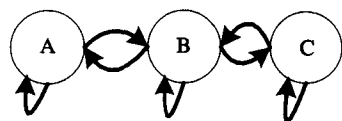


图 4 问题域 D 中涉众之间的关联关系

他们三人的视点模板中陈述的需求团分别如图 5~图 7。A 的一个需求团用自然语言描述是:

要完成目标  $G_1$ , 必须要满足  $fun_1$  和  $fun_2$  两个功能性需求, 而它们分别受到  $Nfun_1, Nfun_2$  的约束。这里  $G_1$  的含义是: “图书借阅”;  $fun_1$  表示“书目查询与图书传递”;  $fun_2$  表示: “日志记录”;  $Nfun_1$  “能在一分钟内完成”;  $Nfun_2$  “保证操作的可靠性”。它们都是认知逻辑命题。与其类似  $G_2$  为“对读者信息的操作”;  $G_3$  为“对图书目录的操作”;  $G_4$  为“可以借阅所有图书”。 $fun_3$ : “查询读者信息”;  $fun_4$  是“修改读者信息”;  $fun_5$  表示“添加新的书目”;  $fun_6$ : “修改已存在的书目”。

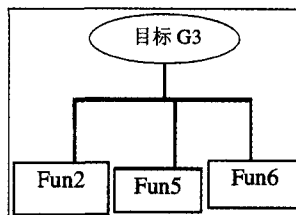


图 5 C 的需求团

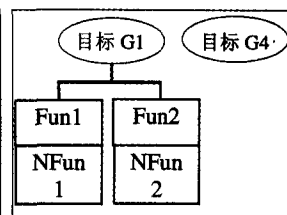


图 6A 的需求团

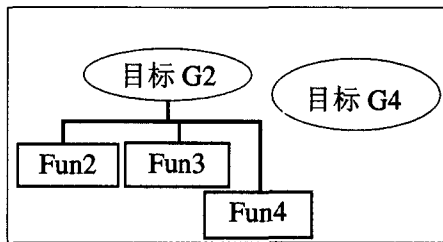


图 7 B 的需求团

我们可以有如下推导:

$$\begin{aligned} (M, B) &\models \neg G_4 \\ (M, B) &\models \neg K_B G_4 \quad (A \text{ 认为 } G_4 \text{ 为真}) \\ (M, A) &\models G_4 \\ (M, B) &\models \neg K_B \neg G_4 \quad (\text{依据 A 与 B 的可达关系}) \\ (M, B) &\models \neg K_B G_4 \wedge \neg K_B \neg G_4 \end{aligned}$$

这就是说, 从 B 的视点来看,  $G_4$  是一个绝对不一致的需求。

我们还可以进行如下推导:

$$\begin{aligned} (M, A) &\models G_1 \\ (M, A) &\vdash G_1 \quad (\text{完备性}) \\ G_1 &\vdash fun_2 \wedge Nfun_2 \quad (R5) \\ (M, A) &\vdash fun_2 \\ (M, B) &\vdash fun_2 \\ (M, A) &\models K_A fun_2 \\ (M, A) &\models K_B fun_2 \\ (M, A) &\models K_A fun_2 \wedge K_B fun_2 \end{aligned}$$

这就是说 A 知道她自己和 B 都认为需要  $fun_2$  这个功能性需求, 因此  $fun_2$  就有可能是一个相对不一致的需求。这需要相关的涉众进行讨论, 如果讨论的结果是他们发现各自对  $fun_2$  的理解的确存在着差异, 那么就可以选择某种策略来解决。

另外, 我们利用 CK 算子发现  $fun_2$  同时也是一个公共知识, 因此 C 也需要加入到讨论中来, 以确定他对该需求是否存在不同的理解。从而为发现和处理相对不一致的需求提供了一个有效的手段。

总结 需求工程可以被理解为知识表示, 知识获取和分析知识的过程。某一涉众所陈述的对预期系统的需求是否有

可能得到满足,不但与其领域相关,而且与该领域内其它涉众对其观点是否认同相关。

现有的多视点需求工程方法中关于不一致性处理的方法还很少有对认知属性进行研究的。但是我们相信,在需求工程中不一致性的问题是由于涉众对某一问题存在着不同的理解所导致的,也就是说这是一个与认知相关的问题。所以我们提出了基于问题域的多视点建模框架,并使用认知逻辑对该框架进行解释,从而提供了一种形式化的方法来发现和处理不一致性问题,希望这种方法可以提高需求规约说明书的质量,为开发出满足涉众需求的软件系统奠定基础。

本文的主要贡献在于提出了一个基于问题域的多视点建模框架,并使用认知逻辑对该框架进行解释。这不但符合软件开发活动中涉众处于不同领域的实际,同时也提供了一个形式化的方法来帮助解决多视点需求工程方法中必须要解决的 inconsistency 问题。

在我们现有的工作中,还有许多问题需要在以后的工作中进一步去解决。譬如:知识应该是一个动态的过程。涉众在不同的时间对预期系统有着不同的看法。也就是说,不一致性的问题应该是随着时间的改变而改变,它具有动态性,而不能仅仅从静态的视角观察它。但是在我们这个框架和解释下还不能表示知识是与时间相关这一特性的。如果不能解决这一问题,就难以真正处理变化的需求以及做到需求的可追踪性。这是我们以后需要进一步研究的课题。

## 参考文献

- 1 Finkelstein A, Gabbay D, Hunter A, et al. Inconsistency handling in multiperspective specifications. *IEEE Trans. on Software Engineering*, 1994, 20(8): 569~578
- 2 Zave P. Classification of Research Efforts in Requirements Engineering. *ACM Computing Surveys*, 1997, 29(4): 315~321
- 3 Balzer R. Tolerating inconsistency. In: *Proceedings of the Fifth International Software Process Workshop (ISPW '89)*, Kennebunkport, Maine, USA. IEEE Computer Society 1989. 41~42

- 4 Ainsworth M, Cruickshank A H, Groves L J, Wallis P J L. Formal specification via viewpoints. In: *Proceedings of the 13th New Zealand Computer Conference New Zealand Computer Society*, Auckland, New Zealand, 1993. 218~237
- 5 Jackson D. Structuring Z specifications with views. *ACM Transactions on Software Engineering and Methodology*, 1995, 4: 365~389
- 6 Boiten E, Derrick J, Bowman H, Steen M. Consistency and refinement for partial specification in Z. In: *Proceedings of the Third International Symposium of Formal Methods Europe (FME'96)*; Industrial Benefit of Formal Methods, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1996, 1051: 287~306
- 7 Leduc G. On the role of implementation relations in the design of distributed systems using LOTOS. Ph. D. Thesis, University of Liege, Liege, Belgium, 1991
- 8 Khendek F, von Bochmann G. Merging specification behaviors. Technical Report 856, Departement d'informatique et de recherche operationnelle, Universite de Montreal, 1993
- 9 Ichikawa H, Yamanaka K, Kato J. Incremental specification in LOTOS. In: *Proceedings of the IFIP WG.6.1 Tenth International Symposium on Protocol Specification, Testing and Verification X North-Holland Ottawa, Canada*, 1990. 183~196
- 10 Bowman H, Derrick J, Linington P, Steen M W A. FDTs for ODP Computer Standards and Interfaces, 1995, 17: 457~479
- 11 Easterbrook S, Chechik M. A Framework for Multi-Valued Reasoning over Inconsistent Viewpoints. In: *Proceedings of the 23rd International Conference on Software Engineering (ICSE'01)* (Toronto, Ontario, Canada May 12-19, IEEE Computer Society, 2001. 411~420
- 12 Sabetzadeh M, Easterbrook S M. Analysis of Inconsistency in Graph-Based Viewpoints: A Category-Theoretic Approach. In: *Proceedings of the 18th IEEE Int. Conf. on Automated Software Engineering (ASE 2003)* (Montreal, Canada, Oct. 6-10), IEEE Computer Society, 2003. 12~21
- 13 Nuseibeh B, Kramer J, Hunter A. A framework for expressing the relationships between multiple views in requirements specification. *IEEE Trans. on Software Engineering*, 1994, 20(10): 760~773
- 14 Fagin R, Halpern J Y, Moses Y, Vardi M Y. Reasoning about Knowledge. The MIT Press, Cambridge, MA, 1995
- 15 Nuseibeh B, Finkelstein A, Kramer J. Viewpoints: Meaningful relationships are difficult! In: *Proceedings of the 27th Int'l Conf. on Software Engineering*. Oregon; IEEE Computer Press, 2003. 676~683

(上接第 195 页)

坐标代表特征(基因)在该样本上的取值(基因表达值),被分至同一组的特征曲线在同一个图上,从图中可以看出,被分至同一组的特征均具有较强的相关性,因此可以通过保留每组代表特征并去除组内其余特征的方式消除特征冗余。图 2 给出采用 FSIGGA 算法对 Colon Tumor 数据集进行特征选择的遗传算法迭代运行结果,图中横坐标代表遗传算法的迭代次数,纵坐标代表每一代种群得到的最优结果(即最低的感知器分类错误率)。从图 2(a)中可以看出,在遗传算法的迭代过程中,2 维 Colon Tumor 数据集的线性分类错误率在持续下降,最后收敛到一个较低的错误率 8.06%。而 3 维的 Colon Tumor 数据集则具有更好的可分性,可收敛到一个更低的分类错误率 3.23%。图 3 给出了 Colon Tumor 数据在 FSIGGA 算法所选出的优化的 2 维(图 3(a))及 3 维(图 3(b))特征子空间中的分布散点图,分别用“\*”及“▽”代表两类样本,从图中可看出样本集在对应的特征子空间中具有较好的线性可分性,其中 3 维特征子空间中的线性可分性要优于 2 维特征子空间。图 4 给出了不同特征维数下的分类错误率,从图中可以看出,随着维数的增加,针对 FSIGGA 算法所选出的近似最优特征子空间的感知器分类错误率在下降,且维数达到 7 以后,分类错误率降至 0。

**结论** 本文主要针对高维数据的特征选择问题,融合 filter 及 wrapper 特征选择模型,提出了一种基于信息增益及遗传算法的特征选择算法。实验证明,该算法能较为有效地找出具有较好的可分离性的特征子集,从而实现降维并提高分

类精度。

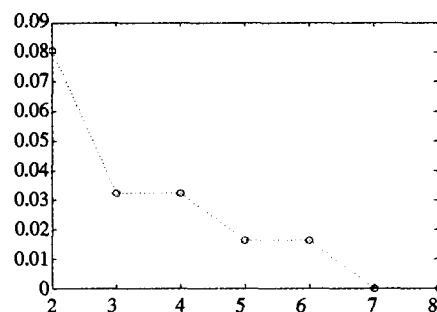


图 4 不同特征维数下的分类错误率

## 参考文献

- 1 John G H, Kohavi R, Pflieger K. Irrelevant Features and the Subset Selection Problem. In: *Proc. of the Eleventh Intl. Conf. on Machine Learning*, 1994. 121~129
- 2 Kohavi R, John G H. Wrappers for feature subset selection. *Artificial Intelligence*, 1997, 97(1-2): 273~324
- 3 Liu Huan, Yu Lei. Toward Integrating Feature Selection Algorithms for Classification and Clustering. *IEEE Transactions on Knowledge and Data Engineering*, 2005, 17(5): 491~502
- 4 Yang J, Honavar V. Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems*, 1998, 13(2): 44~49
- 5 YU Lei, Liu Huan. Efficient Feature Selection via Analysis of Relevance and Redundancy. *Journal of Machine Learning Research*, 2004(5): 1205~1224
- 6 Mitra P, Murthy C A, Pal S K. Unsupervised Feature Selection Using Feature Similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002, 24(3): 301~312