

# SRS 及其质量模糊度量方法的研究<sup>\*</sup>

甘早斌 陈正勇 陈传波 裴先登

(华中科技大学计算机科学与技术学院 武汉430074)

## Study of Fuzzy Evaluation for SRS and its Quality

GAN Zao-Bin CHEN Zheng-Yong CHEN Chuan-Bo PEI Xian-Deng

(College of Computer Sci. & Tech., Huazhong University of Sci. & Tech., Wuhan 430074, PR China)

**Abstract** Being the first stage of the software lifecycle, software requirements analysis plays a pivotal role in the entire software development process. The success and failure key to the entire software development projects is the quality of software requirements specifications (SRS). By means of fuzzy set theory, this paper firstly analyzes SRS and its characteristics, and then proposes the quota system for evaluating the quality of software requirements specifications. This paper also discusses the approaches of relating data acquirement and fuzzy evaluating, and the quantitative analysis for evaluation results. Finally, the expectation of the future work is given.

**Keywords** Software requirements specification, Requirements analysis, Fuzzy evaluation, Quantitative analysis

## 1. 问题的提出

1995年Standish通过对8000多个软件开发项目的调查发现,导致项目失败的最主要的两个原因是不完整的软件需求规格说明(13.1%)和缺乏用户参与(12.4%)<sup>[1]</sup>。Boehm研究发现要改正在产品付诸应用后所发现的一个需求方面的缺陷比在需求阶段改正这个错误要多付出68倍的成本<sup>[2]</sup>。近来很多研究表明这种错误导致成本放大因子可以高达200倍<sup>[3]</sup>。由此可以看出,软件需求分析作为软件开发过程的第一个阶段,是整个软件项目成功的核心所在,它为其它许多技术和管理活动奠定了基础。高质量的SRS,既可以减少软件开发中的错误,保证项目能满足用户需求,还可以减少修改错误的费用,从而大大缩短软件开发时间,提高软件开发效率,降低软件开发成本。

为了提高软件需求的质量,国内外许多学者在需求分析方面进行了大量的研究,提出了许多需求形式化描述语言和需求分析方法。而对于软件需求规格说明的质量评审,没有一个量化的衡量标准,目前大多采用定性分析法,很难得出一个较客观的量化评价。基于此,本文借助于模糊数学的基本理论<sup>[4]</sup>,在分析SRS及其质量特性的基础上,提出了软件需求的质量度量指标体系,并讨论了有关数据的获取以及模糊综合评价方法,同时对评价结果进行了定量分析。最后对下一步工作进行了展望。

## 2. SRS 及其质量特性

SRS也称功能规格说明、需求协议或系统规格说明,它精确地阐述了一个软件系统必须提供的功能和性能以及它所考虑的限制条件<sup>[5]</sup>。SRS不仅是系统测试和用户文档的基础,也是所有子系统规划、设计、编码的基础。SRS常采用结构化的自然描述语言、图形化分析模型、形式化逻辑语言等方法来编写。形式化逻辑语言具有很强的严密性和精确性,但可读性差,用户难以理解和接受;结构化的自然语言可读性强,易于

理解,通常具有模糊性和二义性;而图形化模型需要把文本表示和图形表示结合起来才能绘制出系统需求的完整描述,可增强用户和系统设计人员对系统需求的理解,现逐渐被大多数项目所采用。

### 2.1 SRS 的内容

每个软件开发组织针对不同的项目其SRS的内容和编写的格式要求可能不同,但至少包括以下内容:

- 1) 文档说明。主要包括SRS文档内容列表以及相应的修正历史记录,该记录包括对SRS所做的修改、修改日期、修改人员和修改原因。
- 2) 约定。描述设计和实现上的一些约定或限制,如开发环境和运行环境的约定、开发规范或开发标准的约定。
- 3) 系统功能需求。详细列出系统的功能需求,也就是必须提交给用户的软件功能。
- 4) 系统非功能需求。主要包括系统的性能需求、安全性需求、软件质量需求等。
- 5) 外部接口需求。主要描述用户界面需求以及目标系统与外部系统的接口(包括硬件接口和软件接口)。
- 6) 待确定需求问题。详细列出SRS中待确定问题的列表,便于跟踪调查。

### 2.2 SRS 的编写原则

许多需求文档的质量可以通过使用有效的技术编写风格和使用用户术语得以改进。因此,在编写高质量SRS时应遵循以下原则:

- 1) 每一需求都应具有相应需求标识号,保证需求的可跟踪性和可修改性;
- 2) 应符合中文的语法要求,保持句子和段落的简短;
- 3) 避免使用模棱两可的、主观的术语,减少不确定性和二义性;
- 4) 避免使用比较性的词汇,如最大化、最小化、最佳等等,含糊的语句表达将引起需求的不可验证性;
- 5) 正确使用各种可视化强调标志,如黑体、下划线、斜体

<sup>\*</sup> 本文得到国家高性能计算基金项目资助,项目编号为:99302。甘早斌 博士研究生,主要研究方向为软件工程,基于Internet的应用研究。

等;

6)少用技术专业术语,多使用用户易于理解的术语。

### 2.3 SRS 的质量特性

一个高质量 SRS 必须具备以下质量特性:

**正确性:**一个需求集是正确的,当且仅当其中每条需求都代表了要构造系统所要完成的功能;

**完备性:**需求集是完备的当且仅当它描述了用户关心的所有有意义的需求,包括与功能、性能、设计约束、属性或外部接口有关的需求;明确定义了软件对所有可能输入的各种响应,对所有的页、图、表以及所有名词和度量单元的定义提供完整的引用和标记;

**无二义性:**对所陈述的每一项需求都只有一种解释;

**一致性:**不存在矛盾的子集,如不出现术语使用冲突,性质冲突,或时间约束冲突;

**可行性:**每一项需求都必须是在已知系统和环境的权能和限制范围内可以实施;

**划分优先级:**根据每项需求的重要性和稳定性给每项需求分配一个实施优先级,以指明它在特定产品中所占的分量;

**可验证性:**每项需求能通过设计测试用例或其他验证方法的验证;

**可修改性:**对其中任一需求的修改很容易地、完整地、一致地进行,同时保持已存在的需求集的结构和风格;

**可跟踪性:**指每项需求的出处都清楚了,便于以后引用。

**可理解性:**指用户和开发人员都能完全理解单个需求和需求集包含的全部功能。

## 3. 软件需求质量评价方法

为了准确度量软件需求规格说明(SRS)的质量特性,必须建立一种度量方法。结合我们在项目管理过程中进行 SRS 质量评审的经验,综合文[5,6]给出的 SRS 质量度量指标,我们构造了一个 SRS 质量模糊综合评价指标体系(如表1),表中定义了 SRS 质量度量十大指标。

表1 软件需求规格说明质量模糊评价指标体系

序号	定性指标	模糊评价			
		优(u <sub>1</sub> )	良(u <sub>2</sub> )	中(u <sub>3</sub> )	差(u <sub>4</sub> )
1	正确	r <sub>11</sub>	r <sub>12</sub>	r <sub>13</sub>	r <sub>14</sub>
2	完备	...			
3	无二义				
4	一致				
5	可行				
6	优先级				
7	可验证				
8	可修改				
9	可追踪				
10	可理解				

该指标体系是从十个质量特性方面对 SRS 质量作出综合评价。其评价方法如下:

#### 1) 定性评价指标模糊化

SRS 质量度量的模糊评价集合和 SRS 质量特性指标的模糊评价集合定义为:

$$U = \{u_1, u_2, u_3, u_4\} = \{\text{优, 良, 中, 差}\}$$

SRS 质量度量值用  $\mu(u_i)$  表示,  $\mu(u_i)$  表示 SRS 质量分别隶属于优、良、中、差的程度。其中  $\mu(u_i) \in [0, 1]$  ( $i=1, 2, 3, 4$ )。

4)。

同样,对于 SRS 的十大质量定性指标的度量值则用  $\mu_i(u_j)$  表示,  $\mu_i(u_j)$  表示 SRS 的十大性能指标分别隶属于优、良、中、差的程度。其中  $\mu_i(u_j) \in [0, 1]$  ( $i=1, 2, \dots, 10; j=1, 2, 3, 4$ )。

#### 2) 计算各项指标的隶属度

这一步是 SRS 质量评价的关键。评审小组可以首先拟定一份审查清单,由评审小组成员分别按照审查清单对每一项需求进行严格审查,并分别对 SRS 中每项需求从十大性能指标方面进行评判(优、良、中、差),然后采用统计法分别计算出各项指标的  $\mu_i(u_j)$  ( $i=1, 2, \dots, 10; j=1, 2, 3, 4$ ),并构造 SRS 质量评判矩阵 R:

$$R = \begin{pmatrix} R_1 \\ R_2 \\ \dots \\ R_9 \\ R_{10} \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ \dots & \dots & \dots & \dots \\ r_{101} & r_{102} & r_{103} & r_{104} \end{pmatrix}$$

其中  $r_{ij} = \mu_i(u_j)$  表示第 i 个指标隶属于  $u_j$  的程度,  $i=1, 2, \dots, 10; j=1, 2, 3, 4$ 。

#### 3) 合成运算

由于十大质量定性指标在确定 SRS 质量时的权重是不相同的。为精确起见,我们采用专家经验法对各项指标赋予相应的权重,设权重矩阵为  $P = (p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8 p_9 p_{10})$ ,将指标权重乘以 SRS 质量评判矩阵得到 SRS 质量评判结果为:

$$C = P \cdot R = (c_1 c_2 c_3 c_4)$$

$$\begin{pmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ \dots & \dots & \dots & \dots \\ r_{101} & r_{102} & r_{103} & r_{104} \end{pmatrix} = (c_1 c_2 c_3 c_4)$$

其中:  $\sum_{i=1}^{10} p_i = 1; c_1, c_2, c_3, c_4$  分别表示 SRS 质量隶属于优、良、中、差的程度,即为软件需求规格说明质量的模糊评价。

## 4. 评价结果的量化计算及分析

为了更明确地表示 SRS 质量的好坏,可以进一步将隶属于优、良、中、差的程度量化。设 S 为  $U = \{\text{优, 良, 中, 差}\}$  对应的分数集,按四分制对优、良、中、差分别评分为 4、3、2、1,用矩阵可表示为:

$$S = (4 \ 3 \ 2 \ 1)^T$$

则  $S_1 = R_1 \cdot S; S_2 = R_2 \cdot S; S_3 = R_3 \cdot S; S_4 = R_4 \cdot S; S_5 = R_5 \cdot S; S_6 = R_6 \cdot S; S_7 = R_7 \cdot S; S_8 = R_8 \cdot S; S_9 = R_9 \cdot S; S_{10} = R_{10} \cdot S$  分别表示 SRS 的正确性、完备性、无二义性、一致性、可行性、优先级、可验证性、可修改性、可追踪性、可理解性等十个方面的得分;而  $C \cdot S$  则表示该 SRS 质量的综合得分。

软件需求的质量综合评分可以直接用来指导软件开发项目的下一步工作。评分越高表明软件需求的质量越高;评分越低表明软件需求的质量越差。根据评分结果,对照评价指标,检查某个质量特性是否达到了要求的质量标准。如果某个质量特性不符合规定的标准,就应当分析这个质量特性,找出达不到标准的原因,并加以修改或完善。

**结论** 度量的根本目的是为管理者提供决策支持信息,是管理的基础。对软件项目管理、软件过程管理和软件产品管理来说软件需求度量是必不可少的,软件过程控制和软件过程改善充分依赖于软件需求的质量。本文提出的模糊综合评价方法能客观地评价软件需求规格说明的质量,指出优

# Boosting 家族 Boost-by-majority 系列代表算法

涂承胜<sup>1</sup> 刁力力<sup>2</sup> 鲁明羽<sup>2,3</sup> 陆玉昌<sup>2</sup>

(重庆三峡学院计算机科学系 重庆万州404000)<sup>1</sup>

(清华大学计算机科学技术系 智能技术与系统国家重点实验室 北京100084)<sup>2</sup>

(烟台大学计算机学院 烟台264005)<sup>3</sup>

## The Typical Algorithm of Boost-by-Majority Series in Boosting Family

TU Cheng-Sheng<sup>1</sup> DIAO Li-Li<sup>2</sup> LU Ming-Yu<sup>2,3</sup> LU Yu-Chang<sup>2</sup>

(Dept. of computer science ChongQing three gorges college ChongQing WanZhou 404000)<sup>1</sup>

(Computer Science and Technology Dept., TsingHua University The State Key Laboratory of Intelligent Technology and System Beijing China, 100084)<sup>2</sup>

(Computer College of Yan Tai University Yantai 264005)<sup>3</sup>

**Abstract** Boosting is one of the most representational ensemble prediction methods. It can be divided into two series: Boost-by-majority and Adaboost. This paper briefly introduces the research status of Boosting and one of its serials-Boost-by-majority, analyzes the typical algorithms of Boost-by-majority.

**Keywords** Data mining, Machine learning, Combining prediction, Algorithms

## 1 引言

Boosting 由 Freund 和 Schapire 于1990年提出<sup>[4]</sup>,是提高预测学习系统预测能力的有效工具,也是组合学习中最具代表性的方法,其代表算法可分为 Boost-by-majority 和 AdaBoost 两个系列。Boosting 操纵训练例子以产生多个假设,从而建立通过投票结合的预测器集合。Boosting 在训练例子上维护一套概率分布。Boost-by-majority 通过在每一回迭代中重取样(resampling)生成不同的训练集,AdaBoost 在每个例子上调整这种概率分布。具体的学习算法用于产生成员分类器,成员分类器在训练例子上的错误率被计算出来并以此在训练例子上调整概率分布。权重改变的作用是在被误分的例子上放置更多的权重,在分类正确的例子上减少其权重。通过单个分类器的加权投票建立最终分类器,每个分类器按其在训练集上的精度而加权<sup>[6]</sup>。Boosting 方法一般用于提高不稳定的学习器的性能。

Kearns 和 Valiant 首先提出:在 Valiant 的 PAC 模型<sup>[1]</sup>

中,一个“弱”学习算法是否能被“提升”为一个具有任意精度的“强”学习算法<sup>[2,3]</sup>?1989年 Schapire 提出了第一个可证明的多项式时间 Boosting 算法<sup>[4]</sup>,对上述问题作了肯定回答。之后, Freund 设计了一个更高效的通过重采样或过滤运作的 Boost-by-majority 算法<sup>[5]</sup>,在一定程度上该算法是优化的,实践上却有一些缺陷。野点(Outliers)是训练样本中被标错的样本。AdaBoost 对野点的识别能力强,因为这些样本通过学习通常会得到较高的权值。但野点数目过多时,过分强调“困难”实例将有损 AdaBoost 的性能。为此,1998年 Friedmand 等提出了被称之为“Gentle AdaBoost”的 AdaBoost 变种算法<sup>[10]</sup>,它较少地强调野点。1999年 Freund 介绍了 Boost-by-majority 算法的一个自适应扩展版本 BrownBoost,该算法采用了不强调那些太“困难”以至于无法正确分类的野点<sup>[11]</sup>的方法。

本文集中介绍了 Boost-by-majority 系列的典型算法。限于篇幅,AdaBoost 系列典型算法另文介绍<sup>[12]</sup>。Boost-by-majority 的着眼点在于从大量数据集中取出少量数据作为学习的训练样本,以解决分类学习问题。

涂承胜 讲师,清华大学访问学者。刁力力 博士研究生。鲁明羽 博士研究生。陆玉昌 教授。

缺点,从而为软件需求分析人员改进和提高软件需求规格说明质量提供指导。

这种模糊度量方法可以推广至整个软件项目管理质量度量,将软件需求质量、进度、资源和费用、软件质量等作为软件项目管理质量属性,进而采用多级评价对软件项目管理质量进行综合量化评价。

下一步工作是以此模糊综合评价方法为基础,结合软件需求规格说明的自动生成、自动验证,开发一套软件需求分析综合管理系统,实现软件需求规格说明文档的自动生成、自动验证以及质量模糊评价等功能,进一步提高软件需求质量评价的自动化程度以及评价的准确性和可信度,从而提高需求分析的质量以及需求分析的自动化。

## 参考文献

- 1 The Standish Group. The Scope of Software Development Project Failures. Dennis, MA: The Standish Group, 1995
- 2 Boehm B W. Software Engineering Economics. Englewood Cliffs, NJ: Prentice-Hall, 1981
- 3 [美]Wieggers K E 著, 陆丽娜等译. 软件需求. 北京:机械工业出版社, 2000
- 4 李凡. 模糊信息处理系统. 北京:北京大学出版社, 1998
- 5 IEEE Std 830-1998. IEEE Recommended Practice for Software Requirements Specifications. Los Alamitos, CA: IEEE Computer Society Press, 1998
- 6 Leffingwell D, Widrig D. Managing Software Requirements: A Unified Approach. Addison Wesley Longman, Inc.
- 7 Davis A M. Software Requirements: Objects, Functions, and States. Englewood Cliffs, NJ: Prentice-Hall, 1993