

基于可拓聚类分析的软件质量测度算法研究^①

余 平¹, 胡 玲²

1. 重庆电子工程职业学院 软件学院, 重庆 401331; 2. 内江师范学院 教务处, 四川 内江 641100

摘要: 研究了一种基于可拓聚类分析的软件质量测度算法, 该算法首先建立了一种改进的软件质量测度评估体系, 在该体系的框架下进行了评估指标的测度级别划分, 进行了软件质量测度评估可拓聚类分析, 给出了评估指标的规范化处理模型及权重分配模型, 建立了不同软件质量测度级别的可拓经典域和可拓节域, 给出了软件质量测度分析对象关于不同评估指标和不同测度级别可拓域之间的可拓关联度计算模型, 并提出了基于可拓聚类分析的软件质量测度算法的实现模型. 基于案例进行了算法的验证分析, 表明了算法的有效性与可操作性.

关键词: 软件质量; 测度算法; 可拓聚类分析; 可拓理论

中图分类号: TP311

文献标志码: A

文章编号: 1000-5471(2018)11-0071-07

21 世纪是信息技术和计算机科学技术迅猛发展的时代, 集成化、智能化、系统化、大型化的软件设计将是计算机软件系统开发的热门, 由此也使得众多的研究学者对此展开了广泛的研究^[1-2]. 软件质量分析则是其中的一个研究热点, 软件质量的好坏不仅是软件系统开发以及运行的核心保障, 同时也是软件设计公司运营竞争力的重要支持条件^[3-4]. 由于现代软件系统开发一般具有较大的规模, 从上游的软件设计需求分析一直到后期软件系统的应用、推广和维护服务, 涉及到众多的信息处理, 并且往往包含有模糊信息处理. 目前, 涌现出较多的关于软件质量分析的研究成果^[5-9], 包括分析方法、分析模型、分析策略以及体系等方面, 对复杂系统软件开发起到较好的支持作用. 但应该注意的是, 由于研究视角不同, 使得软件质量分析的测度特性并不统一, 其评估分析以及应用分析都有一定的局限性. 此外, 已有模型在处理模糊信息方面往往没有考虑模糊区间信息, 影响了测度结果的可靠性. 可拓理论从定性角度和定量角度相融合的视角去形式化和模型化分析不确定性问题, 在工程上有着广泛的应用^[10-14], 为软件测度提供了一种新的解决途径.

1 改进的软件质量测度体系

1.1 测度指标选取遵循原则与方法

(1) 遵循原则

软件测度指标的选取遵循科学性原则、客观性原则、系统性原则、典型性原则、动态性原则、层次性原则、可度量性原则等, 以期科学、合理地选择测度指标.

(2) 遵循方法

软件测度指标的选取采用专家咨询法、统计分析法、问卷调查法以及文献查阅法相结合的方式, 以期增强测度指标选取和计算分析的可靠性和可信性.

1.2 测度指标体系

基于上述的原则和方法, 从软件技术与平台支持、软件系统性能、软件维护、升级与服务等方面建立

① 收稿日期: 2018-01-19

基金项目: 四川省教育厅自然科学重点科研项目(13ZA0001).

作者简介: 余 平(1969-), 女, 副教授, 主要从事软件技术研究.

层次性的测度指标体系，结构见表 1.

表 1 软件质量测度体系

1 级指标	2 级指标	指标描述
软件技术与平台支持 X_1	技术成熟度 x_{11}	定性
	技术标准化 x_{12}	定性
	技术先进性 x_{13}	定性
	平台兼容与互换性 x_{14}	定性
	平台可移植与扩展性 x_{15}	定性
软件系统性能 X_2	可重构能力 x_{21}	定量、定性
	功能性 x_{22}	定性
	可操作性 x_{23}	定性
	安全性 x_{24}	定性
	效率 x_{25}	定量、定性
	可靠性 x_{26}	定性
软件维护、升级与服务 X_3	可维护能力 x_{31}	定性
	可建模能力 x_{32}	定性
	可升级能力 x_{33}	定性
	设计与开发周期 x_{34}	定量、定性
	可测试能力 x_{35}	定性
	成本与效益比 x_{36}	定量、定性

1.3 软件质量测度级别划分

软件质量测度分析的目的就是要科学地、合理地 and 有效地鉴别出软件质量的好坏，因此测度级别的划分十分重要. 测度级别划分的粒度是其中关键的因素，粒度太大，则测度级别划分的层次不够清晰，无法有效鉴别；粒度太小，则测度等级划分的层次太细，同样也会使得测度结果受到影响. 为此，本文选择经典的测度层次“优、良、中、差”，将优标记为 C_1 ，良标记为 C_2 ，中标记为 C_3 ，差标记为 C_4 ，则测度级别集表述 C 为

$$C = \{C_1, C_2, C_3, C_4\}$$

(1)

2 软件质量测度算法

2.1 测度指标处理

一般而言，测度指标可以分为 3 种形式：经济型、成本型和趋近型. 为了讨论的全面性，本文对 3 种形式的指标进行标准化处理.

记第 i 个测度指标的量值为 $v(x)_i$ ，对应的指标节域量值为 $V(x)_i = [\min v(x)_i, \max v(x)_i]$ ，若测度指标为经济型指标，则该指标处理值 $u(x)_i$ 为

$$u(x)_i = \frac{v(x)_i - \min v(x)_i}{\max v(x)_i - \min v(x)_i}$$

(2)

若测度指标为成本型指标，则该指标处理值 $u(x)_i$ 为

$$u(x)_i = \frac{\max v(x)_i - v(x)_i}{\max v(x)_i - \min v(x)_i}$$

(3)

若测度指标为经济型指标，并且其最优值为 $v(x)_i^*$ ，则该指标处理值 $u(x)_i$ 为

$$u(x)_i = \begin{cases} \frac{v(x)_i - \min v(x)_i}{v(x)_i^* - \min v(x)_i} & v(x)_i < v(x)_i^* \\ 1 & v(x)_i = v(x)_i^* \\ \frac{\max v(x)_i - v(x)_i}{\max v(x)_i - v(x)_i^*} & v(x)_i > v(x)_i^* \end{cases}$$

(4)

2.2 测度权重分析

本文采用 AHP (Analytic Hierarchy Process) 法进行软件质量测度指标权重分析，具体步骤实现如下：

1) 利用 1—9 标度法进行测度指标的两两对比(标度值选取见表 2), 确定不同层次不同指标之间的相对重要度评判分值, 建立相应的评判矩阵 $\mathbf{A}$.

$$\mathbf{A} = (a_{ij})_{n \times n} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \tag{5}$$

表 2 标度值及其含义

标度值	含 义	标度值	含 义
1	两指标同等重要	9	前者比后者极为重要
3	前者比后者稍微重要	2, 4, 6, 8	处于上述评判状态之间
5	前者比后者明显重要	倒数	反比较
7	前者比后者十分重要		

2) 借助 Matlab 软件, 计算评判矩阵 $\mathbf{A}$ 的最大特征根 $\lambda_{\max}$ 以及相应的特征向量.

3) 基于最大特征根计算一致性指标值

$$CI = \frac{\lambda_{\max} - n}{n - 1} \tag{6}$$

4) 根据表 2 获取平均一致性指标值 RI , 并计算一致性比例 CR

$$CR = \frac{CI}{RI} \tag{7}$$

5) 若满足 $CR < 0.1$, 说明一致性满足要求, 否则需要调整评判矩阵使其满足, 进而采用和法原理、或者和积法原理、或者方根法原理获得测度指标权重.

$$\mathbf{W} = \{w_1, w_2, \cdots, w_i, \cdots, w_n\} \tag{8}$$

2.3 测度可拓经典域

基于可拓理论可知, 若第 i 个测度指标在第 j 个测度级别中的量值范围是 $V(x)_i^j = [\min v(x)_i^j, \max v(x)_i^j]$, 则第 j 个测度级别关于测度指标的可拓经典域模型 $\mathbf{R}_X^j$ 表示为

$$\mathbf{R}_X^j = \begin{bmatrix} N_X^j & x_1 & v(x)_1^j \\ & x_2 & v(x)_2^j \\ & \vdots & \vdots \\ & x_n & v(x)_n^j \end{bmatrix} = \begin{bmatrix} N_X^j & x_1 & [\min v(x)_1^j, \max v(x)_1^j] \\ & x_2 & [\min v(x)_2^j, \max v(x)_2^j] \\ & \vdots & \vdots \\ & x_n & [\min v(x)_n^j, \max v(x)_n^j] \end{bmatrix} \tag{9}$$

其中, N_X^j 为模型 $\mathbf{R}_X^j$ 的名称. x_i 为特征即测度指标, $v(x)_i^j$ 为测度指标量值范围.

特别地, 当测度指标规范化处理后, 模型 $\mathbf{R}_X^j$ 表示为

$$\mathbf{R}_X^j = \begin{bmatrix} N_X^j & x_1 & [\min u(x)_1^j, \max u(x)_1^j] \\ & x_2 & [\min u(x)_2^j, \max u(x)_2^j] \\ & \vdots & \vdots \\ & x_n & [\min u(x)_n^j, \max u(x)_n^j] \end{bmatrix} \tag{10}$$

2.4 测度可拓节域

同理, 第 j 个测度级别关于测度指标的可拓节域模型 $\mathbf{R}_X^{j*}$ 表示为

$$\mathbf{R}_X^{j*} = \begin{bmatrix} N_X^{j*} & x_1 & u(x)_1^{j*} \\ & x_2 & u(x)_2^{j*} \\ & \vdots & \vdots \\ & x_n & u(x)_n^{j*} \end{bmatrix} = \begin{bmatrix} N_X^{j*} & x_1 & [\min u(x)_1^{j*}, \max u(x)_1^{j*}] \\ & x_2 & [\min u(x)_2^{j*}, \max u(x)_2^{j*}] \\ & \vdots & \vdots \\ & x_n & [\min u(x)_n^{j*}, \max u(x)_n^{j*}] \end{bmatrix} \tag{11}$$

其中, N_X^{j*} 为模型 $\mathbf{R}_X^{j*}$ 的名称. x_i 为特征即测度指标, $u(x)_i^{j*}$ 为测度指标量值.

特别地, 存在下述关系

$$\begin{cases} u(x)_i^{j*} = [\min u(x)_i^{j*}, \max u(x)_i^{j*}] \\ \min u(x)_i^{j*} = \min_{1 \leq i \leq n} (\min v(x)_i^j) \\ \max u(x)_i^{j*} = \max_{1 \leq i \leq n} (\max v(x)_i^j) \end{cases} \tag{12}$$

法、统计分析法、问卷调查法等方式, 获得测度对象的初始数据并进行规范化处理, 则获得测度对象软件质量分析数据(表 3).

表 3 测度对象软件质量分析数据

1 级指标	权重	2 级指标	权重	测度数据
软件技术与平台支持 X_1	0.163	技术成熟度 x_{11}	0.265	0.70
		技术标准化 x_{12}	0.143	0.90
		技术先进性 x_{13}	0.472	0.60
		平台兼容与互换性 x_{14}	0.043	0.90
		平台可移植与扩展性 x_{15}	0.077	0.80
软件系统性能 X_2	0.540	可重构能力 x_{21}	0.069	0.80
		功能性 x_{22}	0.261	0.80
		可操作性 x_{23}	0.100	0.90
		安全性 x_{24}	0.243	0.70
		效率 x_{25}	0.158	0.70
		可靠性 x_{26}	0.169	0.80
软件维护、升级与服务 X_3	0.297	可维护能力 x_{31}	0.091	0.80
		可建模能力 x_{32}	0.056	0.50
		可升级能力 x_{33}	0.080	0.50
		设计与开发周期 x_{34}	0.106	0.90
		可测试能力 x_{35}	0.082	0.80
		成本与效益比 x_{36}	0.585	0.70

利用可拓距计算公式, 获得测度对象 2 级指标与经典域和节域之间的计算结果(表 4).

表 4 可拓距计算结果

2 级指标	经典域				节域
	C_1	C_2	C_3	C_4	
技术成熟度 x_{11}	0.20	0.10	-0.10	0.10	-0.30
技术标准化 x_{12}	0	0	0.10	0.30	-0.10
技术先进性 x_{13}	0.30	0.20	0	0	-0.40
平台兼容与互换性 x_{14}	0	0	0.10	0.30	-0.10
平台可移植与扩展性 x_{15}	0.10	0	0	0.20	-0.20
可重构能力 x_{21}	0.10	0	0	0.20	-0.20
功能性 x_{22}	0.10	0	0	0.20	-0.20
可操作性 x_{23}	0	0	0.10	0.30	-0.10
安全性 x_{24}	0.10	0.10	-0.10	0.10	-0.30
效率 x_{25}	0.10	0.10	-0.10	0.10	-0.30
可靠性 x_{26}	0.10	0	0	0.20	-0.20
可维护能力 x_{31}	0.10	0	0	0.20	-0.20
可建模能力 x_{32}	0.40	0.30	0.10	-0.10	-0.50
可升级能力 x_{33}	0.40	0.30	0.10	-0.10	-0.50
设计与开发周期 x_{34}	0	0	0.10	0.30	-0.10
可测试能力 x_{35}	0.10	0	0	0.20	-0.20
成本与效益比 x_{36}	0.10	0.10	-0.10	0.10	-0.30

同理, 利用可拓关联系数计算公式, 获得测度对象 2 级指标与经典域之间的计算结果, 如表 5.

表 5 可拓关联系数计算结果

2 级指标	经典域			
	C_1	C_2	C_3	C_4
技术成熟度 x_{11}	-0.40	-0.25	0.50	-0.25
技术标准化 x_{12}	0	0	-0.50	-0.75
技术先进性 x_{13}	-0.75	-0.333	0	0
平台兼容与互换性 x_{14}	0	0	-0.50	-0.75
平台可移植与扩展性 x_{15}	-0.333	0	0	-0.50
可重构能力 x_{21}	-0.333	0	0	-0.50
功能性 x_{22}	-0.333	0	0	-0.50
可操作性 x_{23}	0	0	-0.50	-0.75
安全性 x_{24}	-0.40	-0.25	0.50	-0.25
效率 x_{25}	-0.40	-0.25	0.50	-0.25
可靠性 x_{26}	-0.333	0	0	-0.50
可维护能力 x_{31}	-0.333	0	0	-0.50
可建模能力 x_{32}	-0.444	-0.375	-0.167	0.167
可升级能力 x_{33}	-0.444	-0.375	-0.167	0.167
设计与开发周期 x_{34}	0	0	-0.50	-0.75
可测试能力 x_{35}	-0.333	0	0	-0.50
成本与效益比 x_{36}	-0.40	-0.25	0.50	-0.25

利用可拓关联度计算公式，获得测度对象与经典域之间的计算结果(表 6).

表 6 可拓关联度计算结果

1 级指标	经典域			
	C_1	C_2	C_3	C_4
X_1	-0.079	-0.036	0.007	-0.040
X_2	-0.176	-0.054	0.081	-0.229
X_3	-0.105	-0.059	0.064	-0.086
可拓关联度	-0.360	-0.149	0.152	-0.355

利用优选模型，根据可拓关联度的大小可知，当前软件质量分析测度对象的测度等级应为中级. 从发展性的角度看，由于其质量未能达到优的级别，为了提升软件的质量，增强软件系统的应用性使其更加适合企业的应用需求，应该对软件系统进行有针对性的升级与改进. 基于可拓聚类分析的软件质量测度算法是一种改进的软件质量测度算法，通过对其评价指标体系进行扩展，给出一种更具有共性的评价指标体系，同时也考虑到评价指标体系的动态性，可以在算法的框架内对评价指标进行动态修改，使得算法具有较好的适应性和灵活性. 而且，本文通过引入可拓距进行软件质量测度分析，将能够更加清晰地表示出测度对象从量变到质变的程度，不仅能够有效地避免测度分析误差，也能够有力地提升测度分析的精确度.

4 结 语

为了更好地促进复杂系统软件质量的提升，本文针对软件质量测度过程中存在的问题进行了分析，研究了一种基于可拓聚类分析的软件质量测度算法. 该算法给出了软件质量分析的改进测度体系，建立了软件质量分析的可拓聚类分析模型，同时给出了软件质量分析的可拓聚类分析算法的实现流程. 通过案例验证对算法进行了详细的说明，验证了算法的可行性. 由于测度体系具有动态性，因此，随着软件质量分析测度对象的变化，软件质量分析测度体系也需要一定的适应性以及内容的深化，这也是后续需要研究的内容. 此外，根据验证实例分析过程也可以看出，本文给出的改进的软件质量测度算法，对于与软件测度分析相似的对象具有很好的适用性，采用本文给出的模型和算法，将为具有模糊区间信息的多属性测度分析提供良好的借鉴作用.

参考文献：

[1] 罗泽勇. 一种小型无人机的机载计算机软件系统设计 [J]. 东北师范大学学报(自然科学版), 2015, 47(1): 64—70.

[2] 夏伟,葛芳. 系统工程方法在计算机软件设计中的应用 [J]. 九江学院学报(自然科学版), 2016, 33(4): 53—55.

[3] 沈国华,黄志球,谢冰,等. 软件可信评估研究综述:标准、模型与工具 [J]. 软件学报, 2016, 27(4): 955—968.

[4] 倪德强. 回归分析在软件质量控制中的应用 [J]. 价值工程, 2015, 34(18): 174—176.

[5] 王金星,汪海涛,姜瑛,等. 基于三角模糊数层次分析法的软件质量评价模型研究 [J]. 计算机与数字工程, 2017, 45(9): 1693—1697.

[6] 郑鹏. 基于 LM—BP 神经网络的软件质量综合评价研究 [J]. 山东理工大学学报(自然科学版), 2016, 30(3): 74—78.

[7] 翁秀木. 一个通用的软件质量评估指标体系 [J]. 软件, 2015, 36(3): 59—63.

[8] 杨阳,汪海涛,姜瑛,等. 基于模糊层次分析法的软件质量评价模型的研究 [J]. 计算机与数字工程, 2017, 45(4): 620—623.

[9] 马宏艳,刘晓娜,杨芳萍. 一种软件质量评估的可拓集对分析法 [J]. 宁夏大学学报(自然科学版), 2017, 38(1): 64—68.

[10] 赵燕伟,周建强,洪欢欢,等. 可拓设计理论方法综述与展望 [J]. 计算机集成制造系统, 2015, 21(5): 1157—1167.

[11] ZHANG Xiao, YUE Jun-ju. Measurement Model and Its Application of Enterprise Innovation Capability Based on Matter Element Extension Theory [J]. Procedia Engineering, 2017, 174: 275—280.

[12] 蔡文,杨春燕. 可拓学的基础理论与方法体系 [J]. 科学通报, 2013, 58(13): 1190—1199.

[13] 翟红生,司志高. 基于可拓分类分析模型的产品快速配置设计 [J]. 西南大学学报(自然科学版), 2014, 36(5): 201—208.

[14] ZHOU Ying-ying, SHI Ji-lei, WU Lei-lei. Application of Extension Theory in Emotion Management [J]. Procedia Computer Science, 2017, 122: 502—509.

A Software Quality Measurement Algorithm Based on Extension Cluster Analysis

YU Ping¹, HU Ling²

1. Department of Software, Chongqing College of Electronic Engineering, Chongqing 401331, China;
2. Education Administrator Office, Neijiang Normal University, Neijiang Sichuan 641100, China

Abstract: A software quality measure algorithm based on extension clustering analysis is studied. Firstly, an improved software quality measure and evaluation system is established. Under this framework, the evaluation level is divided. Then the extension cluster analysis is carried out, and the standardized processing model and the weight distribution model of the evaluation index are given. The extension of the classical domain and joint domain of different levels of software quality measurement are established. The extension correlation degree calculation model between software quality measure analysis object and extension domain of different measure level with different evaluation index is given, and a realization model of software quality measurement algorithm based on extension clustering analysis is put forward. Finally, the verification analysis of the algorithm are carried out based on the case, which shows the effectiveness and maneuverability of the algorithm.

Key words: software quality; measurement algorithm; extension cluster analysis; extension theory

责任编辑 夏娟