

文章编号:1001-9081(2009)09-2530-04

# 基于最小置信度和评价分析的软件质量模糊综合评价改进方案

杨宇科<sup>1</sup>, 李昌国<sup>2</sup>

(1. 四川师范大学 实验室与设备管理处, 成都 610066; 2. 四川师范大学 基础教学学院, 成都 610066)

(hoconan@qq.com)

**摘要:**阐述了软件工程中软件质量评价的重要性,以及作为评价方法的软件质量模糊综合评价的实现过程;指出现行软件质量模糊综合评价过程中的不足;提出了以最小置信度为隶属准则,以基于量分权重和满意度的评价分析为分析手段来指导软件质量评价的新思想;给出了基于最小置信度与评价分析相结合的软件质量模糊综合评价改进及实现方案。最后通过实例验证了该方案的合理性与有效性。

**关键词:**软件质量;最小置信度;评价分析;量分权重

**中图分类号:** TP311.5 **文献标志码:** A

## Improved scheme for fuzzy integrated evaluation of software quality based on minimum confidence and evaluation analysis

YANG Yu-ke<sup>1</sup>, LI Chang-guo<sup>2</sup>

(1. Department of Laboratory and Equipment Management, Sichuan Normal University, Chengdu Sichuan 610066, China;

2. College of Basic Teaching, Sichuan Normal University, Chengdu Sichuan 610066, China)

**Abstract:** The importance of software quality evaluation in software engineering and the realizing process of fuzzy integrated evaluation as the software quality evaluation method were elaborated. The insufficiency in current fuzzy integrated evaluation process was pointed out. A new idea to guide software quality evaluation was proposed, which used minimum confidence as the membership rule, and used evaluation analysis, based on the quantified weight and satisfaction, as the analysis approach. Based on the minimum confidence and the evaluation analysis, an improving and realizing scheme for the fuzzy integrated evaluation of software quality was given. Finally, the rationality and validity of this scheme were verified through real case.

**Key words:** software quality; minimum confidence; evaluation analysis; quantified weight

## 0 引言

随着信息技术的发展和计算机应用范围的不断扩大,计算机软件已广泛应用于航空、航天、工业控制、交通、银行、金融、医疗等领域,与人们的生产生活息息相关,同时对计算机软件的需求也愈来愈复杂,要求越来越高。但庞大的规模和复杂的功能往往导致软件在开发、服务提供和工程实施过程中出现诸多错综复杂的问题,使软件质量的保障越来越具有挑战性。

软件质量评价体系及方法<sup>[1]</sup>是保证软件质量的必要措施,其前瞻性和有效性对软件质量的监控和保障起着决定性作用。软件质量评价<sup>[2]</sup>是指根据软件质量评价标准,对软件开发过程中的软件质量进行全程和持续的测量,揭示软件质量的当前状态,估计软件质量的后续走势。软件质量评价与软件开发过程并行实施,贯穿于软件生存周期过程,可精确控制软件产品的质量,是为需方、开发方和评价方提供量化的软件质量管理的有力手段。显然,软件质量评价不但为软件质量的控制与保证提供可靠支持,也为软件的定型、验收、鉴定、评价和市场推广提供客观和科学的依据。因此,在软件工程领域,软件质量评价正越来越受到人们的广泛关注。

## 1 软件质量模糊综合评价及实现过程

由于影响软件质量的某些因素是模糊的,所以,对于软件质量的评价是具有模糊性,再加上人为的主观原因,人们对某些影响因素的褒贬程度不尽相同,很难直接用统计学的方法确定这些因素的具体判断值<sup>[3]</sup>,因此如何对模糊信息资料进行量化处理和综合评价就显得尤为重要。为此,利用模糊综合评价原理对工程软件质量进行评价有其科学性和实用价值。

### 1.1 软件质量模糊综合评价现行指标体系

软件质量模型是软件质量评价的基础,软件质量模型代表了人们对软件质量特性的认识程度和理解程度,也代表了软件质量评价研究的进展状况。从 20 世纪 70 年代至今常见的质量模型分别是 MaCall (1977), Boehm (1978), FUEPA (1987), ISO/IEC 9126: 1991 (1991), ISO/IEC9126: 2001 (2001)<sup>[4]</sup>。MaCall 模型的贡献在于提出了软件质量特性和软件度量之间的关系,然而并没有考虑软件的功能性。Boehm 模型与 MaCall 的模型类似,并提出了一个质量特性的层次结构,即所有的质量特性归宿与软件的总质量,另外 Boehm 模型增加了硬件领域的一些特性。FURPS 模型的不足之处是没有考虑软件产品的可移植性。ISO/IEC9126: 1991

收稿日期:2009-03-18;修回日期:2009-05-10。

基金项目:四川省科技厅科技支撑项目(2008GZ0211);四川省教育厅科研项目(07ZC058)。

作者简介:杨宇科(1979-),女,四川成都人,助理研究员,硕士,主要研究方向:软件工程;李昌国(1980-),男,四川资阳人,助理实验师,硕士,主要研究方向:图形图像、虚拟现实。

中提出的质量模型已经比较全面,包括6个软件质量特征,每个质量特征包括一些子特征,然后每个子特征又包括一些度量,但是该模型没有明确告诉我们如何对它们进行测量,2001年的ISO和IEC组织对ISO/IEC9126进行了修改,提出了ISO/IEC9126:2001标准,其中质量模型结构上与ISO/IEC9126:1991相同,只是增加了一些子特征。自从1991年

ISO/IEC9126标准发布以后,得到了广泛的应用和认可,国内外很多软件公司依据该标准对软件产品进行质量评价,并在应用的过程中积累了很多的评价经验。现行的软件质量模糊综合评价指标体系是以ISO/IEC9126标准定义的质量模型为基础<sup>[5]313,[6]812,[7]989,[8]652</sup>,建立的软件质量评价指标体系,如图1所示。



图1 软件质量模糊综合评价指标体系

该评价指标体系中,软件质量的评价通过软件质量功能性、可靠性、易用性、维护性、移植性和效率六个方面来衡量。

1) 软件质量功能性由准确性、适应性、互操作性、安全性和功能依从性五个指标评定;

2) 软件质量可靠性由容错性、成熟性、易恢复性和可靠依从性四个指标评定;

3) 软件质量易用性由易学性、易理解性、易操作性、吸引性和易用依从性五个指标评定;

4) 软件质量维护性由易分析性、易改变性、稳定性、易测试性和维护依从性五个指标评定;

5) 软件质量移植性由适应性、共存性、易安装性、易替换性和移植依从性五个指标评定;

6) 软件质量效率由时间特性、资源利用性和效率依从性三个指标评定。

## 1.2 软件质量模糊综合评价现行过程

对软件质量模糊综合的评价,实质是通过建立软件产品固有的质量指标评价体系,逐步细化其特性评价因子,实现对软件质量各层指标的评价<sup>[5]314,[6]813,[7]989,[8]653-654</sup>。根据上述软件质量模糊综合评价指标体系,将其划分为两级指标:第一级指标评价软件总体质量,包括功能性、可靠性、易用性、维护性、可移植性和效率6个特性指标;第二级指标评价软件质量的其他特性,如容错性、成熟性、易恢复性、可靠依从性等子特性。具体评价实现过程如下。

设因素集  $U = \{u_1, u_2, \dots, u_n\}$  表示被评价对象各因素组成的集合,它按某些属性分成  $s$  个子集  $U_i = \{u_{i1}, u_{i2}, \dots, u_{in_i}\}$ ,  $i = 1, 2, \dots, s$ , 满足条件:

$$\begin{cases} \sum_{i=1}^s n_i = n \\ \bigcup_{i=1}^s U_i = U \\ U_i \cap U_j = \emptyset; i \neq j \end{cases} \quad (1)$$

对每一子因素集  $U_i$ , 分别作出综合决策。设  $V = \{v_1, v_2, \dots, v_m\}$  为评价集,  $U_i$  中的各因素的权重分配为:

$$\begin{cases} A_i = (a_{i1}, a_{i2}, \dots, a_{in_i}) \\ \sum_{j=1}^{n_i} a_{ij} = 1 \end{cases} \quad (2)$$

对子集  $U_i$  的各个因素在评价集进行评价,得到  $V$  上的模糊集,从而建立从  $U$  到  $V$  的一个模糊映射  $f: U \rightarrow F(V)$ 。根据该模糊映射可以确定一个评价矩阵,即单因素矩阵  $R_i$ :

$$R_i = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1m} \\ r_{21} & r_{22} & \dots & r_{2m} \\ \vdots & \vdots & & \vdots \\ r_{n_i1} & r_{n_i2} & \dots & r_{n_im} \end{bmatrix} \quad (3)$$

则得二级的评价向量:

$$B_i = A_i \circ R_i = (b_{i1}, b_{i2}, \dots, b_{im}); i = 1, 2, \dots, s \quad (4)$$

将每个  $U_i$  视为一个因素,记  $\mu = (U_1, U_2, \dots, U_s)$ , 于是  $\mu$  又是一个因素集  $\mu$  的单因素决策矩阵为:

$$R = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_s \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1m} \\ b_{21} & b_{22} & \dots & b_{2m} \\ \vdots & \vdots & & \vdots \\ b_{s1} & b_{s2} & \dots & b_{sm} \end{bmatrix} \quad (5)$$

每个  $U_i$  作为  $U$  的一部分,反映了  $U$  的某种属性,可以按他们的重要性给出权重分配  $A = (a_1, a_2, \dots, a_s)$ , 于是得到一级的评价向量:

$$B = A \circ R = (b_1, b_2, \dots, b_m) \quad (6)$$

最后根据最大隶属度识别准则即得出软件质量相应的评价等级。

## 2 软件质量模糊综合评价存在的问题及改进

### 2.1 软件质量模糊综合评价存在的问题

#### 1) 最大隶属度识别准则问题。

按照上述软件质量模糊综合评价方法,假设最终得到一级评价向量为  $(b_1, b_2, \dots, b_m)$ , 且  $b_j = \max\{b_1, b_2, \dots, b_m\}$ , 其中  $j = 1, 2, \dots, m$ , 根据最大隶属度识别准则可得出软件质量评价等级为  $v_j$ 。但是,此时不难发现最大隶属度识别准则仅仅是选取一组向量中数值最大的一个因素,而并没有考虑到该组向量中的其他因素的权重大小,因此,采用最大隶属度识别准则得出的最终评价结果未必合理,如下两种情况所示。

① 若评价空间  $U$  的一个评价集  $\{v_1, v_2, v_3, v_4, v_5\} = \{\text{优秀}, \text{良好}, \text{较好}, \text{一般}, \text{差}\}$ , 软件质量的综合评价向量为  $(0, 0.4, 0.15, 0.3, 0.15)$ , 按最大隶属度识别准则,软件质量级别属于  $v_2$  (良好),但这显然不合理,属于  $v_2$  占40%,而60%则不属于  $v_2$ 。

② 若评价空间  $U$  的一个评价集为  $\{v_1, v_2, v_3, v_4, v_5\} = \{\text{优秀}, \text{良好}, \text{较好}, \text{一般}, \text{差}\}$ , 软件质量的综合评价向量为  $(0.226, 0.226, 0.226, 0.161, 0.161)$ , 按最大隶属度识别准则,此时软件质量级别属于什么呢?显然没有办法确定。假设对评价集作如下等级赋值:优秀(95)、良好(85)、较好(75)、

一般(65)、差(50),对上述综合评价向量应用归一化公式  $\alpha = \left( \sum_{i=1}^m b_i^2 a_i \right) / \left( \sum_{i=1}^m b_i^2 \right)$ , 即得  $\alpha = 78.05$ , 该值与75接近,于是得出质量评价结果为较好。使用归一化方法固然解决了上述出现的问题,但是,这种人为地为等级赋值难免不带有主观性,另外,假若使用归一化计算得出的分数位于所赋某两个数值的中间,即距离左右两个赋值的相对距离相同,那么,此时又很难确定质量等级了。

## 2) 软件质量评价结果缺乏评价分析。

根据软件质量评价体系的指标对软件质量进行评价,最终要分出等级。区分出等级是对软件质量的认可和对开发者的激励,同时也给进一步地开发和选用提供典型样板。但是,对于那些获奖等级不理想的软件质量开发者而言却是一种简单的否定。而且,这些评价往往是一种终结性评价,能得到修改和再评价的机会很少。另外,通过评价,等级好的软件质量优点在什么地方? 等级差的缺点在什么地方? 需要做怎样的修改才可以成为专家眼中的高等级的软件质量? 这些问题不搞清楚的话,软件质量的好坏与价值就难以判断和表现出来。

## 2.2 软件质量模糊综合评价的改进方案

鉴于上述软件质量模糊综合评价存在的问题,文中提出了以最小置信度为隶属准则,以基于量分权重和满意度的评价分析为分析手段来指导软件质量评价的新思想,它兼顾了所有因素权重的大小,从而更能客观有效地为软件质量做出合理的评价,同时,在软件质量评价等级的基础上,给出详细的评价分析,通过评价分析可以得出软件质量优缺点在什么地方,以及做怎样的修改才可以成为专家眼中的高等级的软件等。

### 2.2.1 最小置信度识别准则

**定义1** 最小置信度。指针对于最终评价向量而采用兼顾所有质量因素权重大小的识别准则中参考的最小阈值。

根据定义,设最小置信度为 $\lambda$ ,显然 $\lambda = 0.5$ 。令

$$k_0 = \min \left\{ k: \sum_{i=1}^k b_i > \lambda, k = 1, 2, \dots, m \right\} \quad (7)$$

其中 $b_i$ 为最终的评价向量,则软件质量等级属于第 $k_0$ 个评价等级 $v_{k_0}$ 。显然,相对于最大隶属度识别准则,最小置信度识别准则很好地兼顾了所有质量因素权重,让软件质量评价等级结果更加合理。按照最小置信度识别准则,上述情况①的质量等级属于 $v_3$ ,即较好,情况②的质量等级属于 $v_3$ ,即较好,从而很好地解决了这两种情况。

### 2.2.2 基于量分权重和满意度的评价分析

#### 1) 确定量分的因素权重。

**定义2** 量分的因素权重。是指采用因素集 $U$ 在权重评测尺度集 $E = (e_1, e_2, e_3, \dots, e_t)$ 中进行评测,得到 $E$ 上的模糊集 $R'$ ,然后使用公式 $W = R'E^T$ 计算求得的权重评价向量。

对于因素集 $U = \{u_1, u_2, \dots, u_n\}$ 按某些属性分成 $s$ 个子集 $U'_i = \{u_{i1}, u_{i2}, \dots, u_{in_i}\}, i = 1, 2, \dots, s$ ,满足式(1)的条件:

对每一子集 $U'_i$ 的各个因素在权重评测尺度集 $E = (e_1, e_2, e_3, \dots, e_t)$ 进行评价,得到 $U'_i$ 对应的单因素矩阵 $R'_i$ 。

$$R'_i = \begin{bmatrix} r'_{11} & r'_{12} & \cdots & r'_{1t} \\ r'_{21} & r'_{22} & \cdots & r'_{2t} \\ \vdots & \vdots & & \vdots \\ r'_{n_i1} & r'_{n_i2} & \cdots & r'_{n_it} \end{bmatrix} \quad (8)$$

则得二级量分的因素权重向量:

$$W_i = R'_i E^T = (w_{i1}, w_{i2}, \dots, w_{it}) \quad (9)$$

和二级的权重评测向量:

$$B'_i = A_i \circ R'_i = (b'_{i1}, b'_{i2}, \dots, b'_{it}); i = 1, 2, \dots, s \quad (10)$$

将每个 $U'_i$ 视为一个因素,记 $\mu' = (U'_1, U'_2, \dots, U'_s)$ ,于是 $\mu'$ 又是一个因素集 $\mu'$ 的单因素决策矩阵为:

$$R' = \begin{bmatrix} B'_1 \\ B'_2 \\ \vdots \\ B'_s \end{bmatrix} = \begin{bmatrix} b'_{11} & b'_{12} & \cdots & b'_{1t} \\ b'_{21} & b'_{22} & \cdots & b'_{2t} \\ \vdots & \vdots & & \vdots \\ b'_{s1} & b'_{s2} & \cdots & b'_{st} \end{bmatrix} \quad (11)$$

于是得到一级量分的因素权重向量:

$$W = R'E^T = (w_1, w_2, \dots, w_t) \quad (12)$$

#### 2) 因素满意度及量化。

因素满意度是一个相对的概念,是用户期望值与最终获得值之间的匹配程度。根据因素满意度的概念,评价集 $V = \{v_1, v_2, \dots, v_m\}$ 就是因素满意度测评尺度集,按照量分的因素权重求法同理可得,二级因素满意度向量为:

$$D_i = R''_i V^T = (d_{i1}, d_{i2}, \dots, d_{im}) \quad (13)$$

一级因素满意度向量为:

$$D = R''V^T = (d_1, d_2, \dots, d_m) \quad (14)$$

#### 3) 基于量分权重和满意度的评价分析。

根据评价结果和评价指标,检查质量因素是否达到了要求的质量标准,并给出对应质量因素分析。分析原因自顶向下逐层进行分析,找出可疑的地方加以改进。文中采用质量因素的量分权重及其满意度来表示它的质量特性,若量分权重一定的情况下,质量特性将随着满意度越高而越好。通过大量实际数据分析和拟合可以得到图2所示的质量因素评价分析坐标图<sup>[9]</sup>。

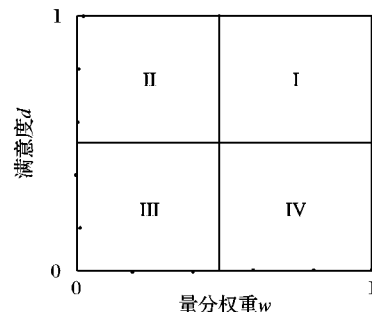


图2 基于量分权重和满意度的评价分析

显然,图2以(0.5,0.5)为坐标中点把量分权重和满意度构成的向量集坐标系划分成了四个象限,根据质量因素位于不同的象限可得出如下评价分析。

①I象限质量因素的相对量分权重高,反映是用户关心的主要质量因素;而满意度也高,反映用户对该质量因素认可度高,开发者对于该类质量因素适合采用稳定增强策略,使继续壮大。

②II象限质量因素的量分权重低,表明该质量因素显得不重要,满意度高说明超出用户的期望平均值,开发者对于该类质量因素适合采用稳定型策略,即保持现状。

③III象限质量因素量分权重低,满意度低,反映该质量因素为其次需要改善的因素,开发者要对此类质量因素加以重视,可以作为次于IV象限质量因素给予改进。

④IV象限质量因素量分权重高,而满意度低,是用户最关注的质量因素,而满意度差距大,开发者可以要优先考虑改善此类质量因素。

上述评价分析适用于单因素评价分析,对于综合第二、一级质量因素的评价分析,我们将采用如表1所示方法进行。

即对于质量因素集  $U'_i = \{u_{i1}, u_{i2}, \dots, u_{in_i}\}$ ,  $i = 1, 2, \dots, s$ , 其中  $U'_i$  是一级质量因素,  $u_{ij}$  是二级质量因素。若质量因素  $U'_i$  属于 I 或 II 象限, 将根据质量因素  $u_{ij}$  位于不同象限分为四种情况: a)  $u_{ij}$  属于 I 象限, 则此类质量因素应该采用稳定增强策略; b)  $u_{ij}$  属于 II 象限, 则此类质量因素应该采用保持现状策略; c)  $u_{ij}$  属于 III 象限, 则此类质量因素应该采用改进策略; d)  $u_{ij}$  属于 IV 象限, 则此类质量因素应该采用优先改进策略; 若质量因素  $U'_i$  属于 III 或 IV 象限, 又将根据质量因素  $u_{ij}$  位于不同象限分为两种情况: a)  $u_{ij}$  属于 I 或 II 象限, 则此类质量因素应该采用改进策略; b)  $u_{ij}$  属于 III 或 IV 象限, 则此类质量因素应该采用优先改进策略。

表1 一二级质量因素综合评价分析表

一级质量因素		二级质量因素			
		$u_{ij}(i = 1, 2, \cdots, s; j = 1, 2, \cdots, n_i)$			
		I	II	III	IV
$U'_i$ ( $i = 1, 2, \cdots, s$ )	I	①	②	③	④
	II	①	②	③	④
	III	③	③	④	④
	IV	③	③	④	④

采用的策略: ①稳定增强; ②保持现状; ③改进; ④优先改进。

### 3 实例

下面以已开发的实验室开放管理平台来简要说明基于最小置信度和评价分析的软件质量模糊综合评价改进方案的具体应用。限于篇幅原因, 以下仅给出计算结果。

#### 1) 等级评价。

按照软件质量模糊综合评价体系, 将  $U$  分为  $U_1, U_2, U_3, U_4, U_5, U_6$  其中  $U_1 = \{u_1, u_2, u_3, u_4, u_5\}$ ,  $U_2 = \{u_6, u_7, u_8, u_9\}$ ,  $U_3 = \{u_{10}, u_{11}, u_{12}, u_{13}, u_{14}\}$ ,  $U_4 = \{u_{15}, u_{16}, u_{17}, u_{18}, u_{19}\}$ ,  $U_5 = \{u_{20}, u_{21}, u_{22}, u_{23}, u_{24}\}$ ,  $U_6 = \{u_{25}, u_{26}, u_{27}\}$ 。

由式(2) ~ (6) 可得一级评价向量  $B = A \cdot R = (0.23, 0.26, 0.16, 0.21, 0.14)$ , 最后根据最小置信度准则(7), 可得  $k_0 = 3$ , 因此该实验室开放管理平台的质量评价等级为  $v_3$ , 即较好。

#### 2) 评价分析。

设置量分权重评测尺度为5级: 非常重要(0.9), 很重要(0.7), 重要(0.6), 一般(0.3), 不重要(0.1), 即  $E = (e_1, e_2, e_3, e_4, e_5)$ 。根据评价集的划分, 满意度测评尺度也分为5级, 分别是: 优秀(0.9), 良好(0.7), 较好(0.6), 一般(0.3), 差(0.1)。

由式(8) ~ (14) 可得: 第二级量分的质量因素权重向量为  $W_1 = (0.56, 0.61, 0.63, 0.49, 0.41)$ ,  $W_2 = (0.63, 0.44, 0.69, 0.55)$ ,  $W_3 = (0.66, 0.62, 0.61, 0.42, 0.63)$ ,  $W_4 =$

$(0.61, 0.63, 0.66, 0.40, 0.51)$ ,  $W_5 = (0.51, 0.54, 0.60, 0.67, 0.41)$ ,  $W_6 = (0.69, 0.47, 0.58)$ ;

第二级质量因素满意度向量为  $D_1 = (0.58, 0.63, 0.53, 0.61, 0.49)$ ,  $D_2 = (0.62, 0.53, 0.58, 0.65)$ ,  $D_3 = (0.65, 0.63, 0.61, 0.51, 0.60)$ ,  $D_4 = (0.60, 0.61, 0.61, 0.48, 0.61)$ ,  $D_5 = (0.41, 0.53, 0.64, 0.59, 0.51)$ ,  $D_6 = (0.58, 0.46, 0.61)$ ;

第一级量分的质量因素权重向量为  $W = (0.51, 0.63, 0.62, 0.55, 0.59, 0.70)$ ;

第一级质量因素满意度向量为  $D = (0.56, 0.58, 0.61, 0.59, 0.54, 0.55)$ 。

根据上述一二级质量因素综合评价分析表, 经过综合评价可得出: 需要保持现状的质量因素是:  $u_4$  (安全性)、 $u_7$  (成熟性)、 $u_{13}$  (吸引力)、 $u_{24}$  (移植依从性); 需要改进的质量因素是:  $u_5$  (功能依从性)、 $u_{18}$  (易测试性)、 $u_{26}$  (资源利用性); 需要优先改进的质量因素是:  $u_{20}$  (适应性); 此外剩下的质量因素都需要稳定增强。

### 4 结语

本文对现行的软件质量模糊综合评价方法进行改进, 实例结果表明, 改进后的基于最小置信度和评价分析的软件质量模糊综合评价方法可以兼顾所有质量因素的权重大小, 使软件质量评价结果更加合适, 以及通过对评价结果的分析, 更容易使软件质量的好坏与价值得以判断和表现出来。

#### 参考文献:

- [1] ISO/IEC 9126 软件工程规范[S]. 北京: 中国标准出版社, 2001.
- [2] 朱三元. 软件质量及其评价技术[M]. 北京: 清华大学出版社, 1990.
- [3] BEVAN N. Measuring usability as quality of use[J]. Journal of Software Quality, 1995, 4(2): 115-150.
- [4] WONG B, JEFFERY R. A framework for software quality evaluation [C]// PROFES 2002: Proceedings of the 4th International Conference on Product Focused Software Process Improvement. Berlin: Springer-Verlag, 2002: 103-108.
- [5] 杨扬. 计算机软件质量模糊综合评价方法[J]. 小型微型计算机系统, 2000, 21(3): 313-315.
- [6] 李良宝, 韩喜双. 软件质量的多级模糊综合评价[J]. 哈尔滨工业大学学报, 2003, 35(7): 812-819.
- [7] 周津慧, 王宗, 杨宗奎, 等. 基于模糊评价方法的软件质量评价研究[J]. 系统工程与电子技术, 2004, 26(7): 987-991.
- [8] 陆鑫, 廖建明. 基于模糊集理论的软件质量评估研究[J]. 电子科技大学学报, 2007, 36(3): 652-655.
- [9] 方海光. 我国教育软件价值评测研究[D]. 成都: 中国科学院成都计算机应用研究所, 2006.

(上接第2529页)

- [3] Synopsys, Inc. Reference verification methodology user guide version 8.6 [EB/OL]. [2009-02-20]. <http://www.synopsys.com/>.
- [4] GLASSER M. Advanced verification methodology cookbook version 2.0 [EB/OL]. (2006-07-24) [2009-02-10]. <http://www.mentor.com/>.
- [5] Cadence Design Systems, Inc. The unified verification methodology white paper [EB/OL]. (2005-02-01) [2009-01-05]. <http://www.cadence.com/>.
- [6] COLGAN J. Open verification methodology relieves inefficiencies [EB/OL]. (2007-09-07) [2009-01-10]. [\[design.com/Articles\]\(http://design.com/Articles\).](http://electronic-</a></li>
</ol>
</div>
<div data-bbox=)

- [7] TSAI T. Techniques for selective reuse of verification components in hierarchical verification of large designs [C]// SNUG: Synopsys Users Group. San Jose: [s. n.], 2008: 97-106.
- [8] GAISLER J, CATOVIC E, ISOMÄKI M. GRLIB IP Core user's manual version 1.0.19 [S]. Sweden: Gaisler Research, 2008.
- [9] STÖHR B, SIMMONS M, GEISHAUSER J. FlexBench: Reuse of verification IP to increase productivity [C]// Proceedings of the 2002 Design, Automation and Test in Europe Conference and Exhibition. Washington, DC: IEEE Computer Society, 2002: 35-41.