

基于 QFD 和 TRIZ 的可信软件技术冲突解决方法

王晓暉, 熊伟*

浙江大学 管理学院, 浙江 杭州 310058

摘要:为解决软件开发中的技术冲突及提高软件的可信性,提出了基于质量功能展开(QFD)和发明性问题解决理论(TRIZ)的可信软件技术冲突解决方法。首先构建可信软件规划质量屋,在软件开发过程中引入用户视角;其次,重点分析质量屋的技术特性自相关矩阵中呈负相关的技术特性,尝试用 TRIZ 的发明创新原理予以解决,获得多个可行的创新性解决方案;在综合专家意见方面,考虑了专家语言表达的模糊性、不确定性和多粒度多语义的情况,用近年来最新发展的基于语言信息的决策理论来评估和选择最可行的解决方案;最后,通过在某航天嵌入式实时操作系统(ERTOS)软件开发过程中的实际应用说明了本文所提方法的适用性和有效性。

关键词:软件可信性;技术冲突;质量功能展开;发明性问题解决理论;多粒度;语言变量;方案评价

中图分类号:TP311.5 文献标识码:A

随着社会的进步和计算机技术的发展,软件已经渗透到社会生活、经济和国防建设的方方面面,人们也越来越依赖软件。相应地,人们对软件的可信性也提出了更高的要求,尤其是对于航天操作系统等安全关键软件^[1-3]。软件的可信性已成为软件工程的重要研究方向和国际上致力于解决的重大问题^[4-5]。

软件需求工程的发展,使人们认识到只有最终用户的直接参与并发挥主导作用,才能真正解决问题空间与求解空间的一致性,消除计算机技术人员和最终用户之间的鸿沟^[6]。而作为用户导向的产品设计方法,质量功能展开(Quality Function Deployment, QFD)是连接用户与技术人员的有效方法^[7-9]。QFD 是以用户需求为依据,并通过质量屋(House of Quality, HOQ)将用户需求转化为产品开发过程的一系列技术特性,在开发初期就对产品的质量和适用性实施全

方位保证的系统化方法。QFD 应用于商业软件开发已有不少成功的案例^[7, 10-11]。QFD 也可将软件的可信性需求逐步映射至软件开发的整个过程,保证可信性需求在软件的开发过程中得到正确而一致地实现,进而提高所开发软件的可信性,因而也同样适用于可信软件的开发^[12]。

在当前的软件开发中,一般应用的都是传统 QFD,对于 QFD 理论自身的一些缺陷考虑不足,这可能导致结果的失真^[13]。在传统 QFD 中,虽然 HOQ 的屋顶考虑了技术特性之间的自相关关系,但在具体应用中一般将这些信息作为一种定性的参考^[14],或只在确定技术特性重要度时予以考虑^[15],而传统 QFD 本身并未提供一套方法来解决设计中的技术冲突。技术特性自相关关系可分为 3 类^[15]:负相关、正相关和不相关。负相关是指当试图提高系统的某个技术特性时,引起系统另一个技术特性的恶化,它正好对应发明性问

收稿日期:2010-03-05; 退修日期:2010-04-21; 录用日期:2010-09-10; 网络出版时间:2010-11-11 09:09

网络出版地址:www.cnki.net/kcms/detail/11.1929.V.20101111.0909.002.html

DOI: CNKI:11-1929/V.20101111.0909.002

基金项目:国家自然科学基金(90718038);浙江省自然科学基金(Y7080086);浙江省社科规划课题(10CGGL20YBB)

* 通讯作者. Tel.: 0571-88206856 E-mail: zjuquality@yahoo.com.cn

题解决理论 (Theory of Inventive Problem Solving, TRIZ)^[16-17] 中技术冲突的定义。不难看出,呈负相关关系的技术特性,其本质就是产品开发中的技术冲突。因而可将 QFD 与 TRIZ 结合起来,借用 TRIZ 的创新原理来解决问题。当前 QFD 与 TRIZ 相结合的研究多局限于概念框架的提出或理论探讨,而缺少实际的案例佐证^[13, 18]。此外,在 TRIZ 的矛盾矩阵中每个空格都有 0~4 个创新原理,所以技术人员通常会获得多个可行的概念设计方案或灵感。层次分析法 (Analytic Hierarchy Process, AHP) 是最常用的多准则决策方法,可用于创新方案的选择^[19]。然而产品开发的早期阶段包含着大量的不确定性,决策者的评判往往是主观而模糊的^[9, 17]。为此,文献^[17]引入模糊 AHP 法来选择创新方案。然而模糊理论的应用又存在隶属函数的选择及解模糊化的问题,也有丢失决策信息的可能^[20-21],因而其结果在某些情况下可能是不准确的;另一方面,基于语言信息的决策理论因其能更简便有效地处理不确定信息而日益受到学者的关注^[21-24]。因此,本文拟用基于语言信息的决策理论,在考虑多个可信性准则的条件下来评估和选择最可行的软件技术冲突解决方案。

可信软件的开发过程中通常存在大量难以解决的技术冲突^[25],包括传统 QFD 在内的现有方法并不能很好地处理这一问题。因此,笔者提出了基于 QFD 和 TRIZ 的可信软件冲突解决方法:首先构建可信软件规划质量屋,并重点探讨质量屋的技术特性自相关矩阵中呈负相关的技术特性,尝试用 TRIZ 的发明创新原理予以解决,可获得多个可行的创新性解决方案;最后用基于语言信息的决策理论来选择最可行的解决方案。

1 语言信息决策理论预备知识

1.1 非平衡语言评估标度

由于客观事物的复杂性,以及人类思维的模糊性和不确定性,在许多实际的决策问题中,决策者难以用清晰数值给出评价,而往往直接用语言信息的形式来反映其自身的偏好^[21-24]。实际上,直接用语言变量来表征产品开发中主观而不确定

的信息既能合理地体现判断的模糊性,又能最充分地利用决策信息^[26]。在可信软件的开发过程,尤其是需求挖掘、系统架构和方案评价等阶段,存在大量的模糊和不确定性,也同样可用语言变量来进行决策和处理。

语言评估标度是用语言变量进行决策的基础。常见的两种标度的语言术语下标基本是对称而均匀分布的^[21-24],形式较为单一且缺乏合理的理论依据,已不能满足决策理论的发展和实际应用的需要。在现实生活中,如群体协商、项目评价和新产品投资开发等过程中,非平衡的语言信息经常出现^[27]。文献^[27]对平衡和非平衡语言评估标度进行对比研究发现,在相同的相似度阈值条件下,后者的群体一致性程度明显高于前者,且后者在达成群体一致方面所需的迭代次数也更少,因而也更实用。采用如下的以零为中心对称、且术语个数为奇数的非平衡语言评估标度^[27]:

$$S^{(k)} = \left\{ \begin{aligned} &S_a^{(k)} \\ &\alpha = 1 - k, \frac{2}{3}(2 - k), \frac{2}{4}(3 - k), \dots, \\ &0, \dots, \frac{2}{4}(k - 3), \frac{2}{3}(k - 2), k - 1 \end{aligned} \right\} \quad (1)$$

式中: $S^{(k)}$ 为离散语言标度集; $s_a^{(k)}$ 为语言术语; $s_{1-k}^{(k)}$ 和 $s_{k-1}^{(k)}$ 分别为决策者实际使用的语言术语的下限和上限; k 为正整数; 语言术语集的势为 $2k - 1$ 。

$S^{(k)}$ 满足下列条件:

- (1) 若 $\alpha > \beta$, 则 $s_a^{(k)} > s_b^{(k)}$ 。
- (2) 存在负算子 $\text{neg}(s_a^{(k)}) = s_{-a}^{(k)}$, 且 $\text{neg}(s_0^{(k)}) = s_0^{(k)}$ 。

为了便于计算和避免决策信息丢失,在 $S^{(k)}$ 的基础上定义一个拓展的连续性语言标度集 $\bar{S}^{(k)} = \{s_a^{(k)} \mid \alpha \in [-t, t]\}$, 其中 $t(t \geq k)$ 是一个充分大的自然数。若 $s_a^{(k)} \in S^{(k)}$, 则称 $s_a^{(k)}$ 为本原术语; 若 $s_a^{(k)} \in \bar{S}^{(k)}$, 且 $s_a^{(k)} \notin S^{(k)}$, 则称 $s_a^{(k)}$ 为虚拟术语。虚拟术语一般仅出现在计算中。

定义 1 对任意两个语言术语 $s_{a_1}^{(k)}, s_{a_2}^{(k)} \in \bar{S}^{(k)}$, $\lambda \in [0, 1]$, 其运算法则定义如下:

- (1) $s_{a_1}^{(k)} \oplus s_{a_2}^{(k)} = s_{a_2}^{(k)} \oplus s_{a_1}^{(k)} = s_{a_1 + a_2}^{(k)}$ 。
- (2) $\lambda s_{a_1}^{(k)} = s_{\lambda a_1}^{(k)}$ 。

定义 2 设映射 $\bar{S}^n \rightarrow \bar{S}$, 则定义语言加权平均 (Linguistic Weighted Averaging, LWA) 算子为

$$\text{LWA}(s_{a_1}, s_{a_2}, \dots, s_{a_n}) = \omega_1 s_{a_1} \oplus \omega_2 s_{a_2} \oplus \dots \oplus \omega_n s_{a_n} = s_{\bar{\alpha}} \quad (2)$$

式中: $\bar{\alpha} = \sum_{j=1}^n \omega_j \alpha_j$; $\mathbf{w} = [\omega_1 \quad \omega_2 \quad \dots \quad \omega_n]^T$ 为 s_{a_i} ($i = 1, 2, \dots, n$) 的权重向量, 且 $\omega_j \in [0, 1]$, $\sum_{j=1}^n \omega_j = 1$ [23], n 为平衡系数。

定义3 设映射 $\bar{S}^n \rightarrow \bar{S}$, $\mathbf{v} = [v_1 \quad v_2 \quad \dots \quad v_n]^T$ 是位置权重向量, 且 $v_j \in [0, 1]$, $\sum_{j=1}^n v_j = 1$, 则定义语言混合集成 (Linguistic Hybrid Aggregation, LHA) 算子为

$$\text{LHA}(s_{a_1}, s_{a_2}, \dots, s_{a_n}) = v_1 s_{\beta_1} \oplus v_2 s_{\beta_2} \oplus \dots \oplus v_n s_{\beta_n} = s_{\alpha_i} \quad (3)$$

式中: s_{β_j} 为语言加权数据组 s_{α_i} ($s_{\alpha_i} = n\omega_i s_{a_i}$, $i = 1, 2, \dots, n$) 中第 j 大的元素。显见, LHA 算子不仅能反映数据本身的重要性, 而且体现了数据所在位置的重要性, 因而更综合、更全面 [22]。

1.2 多粒度多语义信息的一致化

由于可信软件系统的复杂性, 对由 TRIZ 演绎得出的软件技术冲突解决方案的评价和选择往往需要多位专家的群体参与, 是典型的群决策问题。在这样的群决策中, 受限于经验和知识, 以及航天软件本身的复杂程度等主客观因素的影响, 各决策者所采用的语言粒度不尽相同, 因而有必要对由不同标度所表达的语言决策信息进行一致化。设 $\bar{S}^{(k_1)} = \{s_{\alpha}^{(k_1)} \mid \alpha \in [1 - k_1, k_1 - 1]\}$ 和 $\bar{S}^{(k_2)} = \{s_{\beta}^{(k_2)} \mid \beta \in [1 - k_2, k_2 - 1]\}$ 为任意两个给定的连续性非平衡语言标度集, 则定义它们间的转换函数 [27] 为

$$F: S^{(k_1)} \rightarrow S^{(k_2)} \quad (4)$$

$$\beta = F(\alpha) = \alpha \frac{k_2 - 1}{k_1 - 1} \quad (5)$$

$$F^{-1}: S^{(k_2)} \rightarrow S^{(k_1)} \quad (6)$$

$$\alpha = F^{-1}(\beta) = \beta \frac{k_1 - 1}{k_2 - 1} \quad (7)$$

利用式(4)~式(7), 可将决策者给出的多粒度语言信息一致化。例如, 要将标度集 $S^{(5)}$ 转换为 $S^{(4)}$, 根据式(4)~式(7), 可得 $\beta = F(\alpha) = \alpha \frac{4-1}{5-1} =$

$\frac{3}{4}\alpha$, 则有 $s_{\frac{3}{4}\alpha} \rightarrow s_{\frac{3}{4}\alpha}$, 其余可类似求得。

在进行一致化处理时, 通常选择被使用频率最高的术语集为基本语言术语集, 通过转换函数, 将其余语言变量以基本语言术语来表示。

2 可信软件技术冲突解决方法

作为用户驱动的产品设计方法, QFD 能使设计人员在产品设计过程中最大限度地满足用户需求。通过 QFD 的 4 阶段 HOQ 模型 [8], 可使产品的开发过程可视化, 并以结构化的方式向产品开发技术人员指明需要解决的问题。在可信软件的开发中, 就是指明软件开发过程中的技术冲突。

然而, 传统的 QFD 方法只能满足顾客需求, 却无法向设计人员提供解决产品设计问题的具体方法。同时, QFD 理论对技术冲突和技术瓶颈也往往无能为力 [13]。另一方面, 由 Altshuller 提出的 TRIZ 是从 250 多万份高水平专利中总结出的体现技术创新一般规律的理论体系 [16]。TRIZ 为产品设计人员提供了一套有效的问题解决工具, 给设计人员指出了方向性的解决思路。应用 TRIZ 能消减技术特性的负效应, 获得创新性的产品概念方案, 提升现有的设计层次。

因此, TRIZ 也是一种能完善和补充 QFD 理论的有利工具, 将二者相结合, 能为软件开发中技术冲突提供创新性解决方案。

为此, 本文提出基于 QFD 和 TRIZ 的可信软件技术冲突消解模型, 如图 1 所示。该模型的详细应用过程如下面 6 个步骤所示:

步骤 1 构建可信软件规划质量屋。

首先, 构建用户需求-技术特性质量屋, 即可信软件规划 HOQ。先将用户对于软件系统的可信性需求通过 HOQ 转化为软件开发过程中的技术特性, 继而重点探讨不同技术特性之间的自相关关系, 完成 HOQ 屋顶的技术特性自相关矩阵。

步骤 2 收集和分析技术特性自相关矩阵中的技术冲突。

在可信软件的开发设计过程中, 不同的设计参数之间往往存在“协同”或“冲突”的关系 [25]。因此有必要对那些可能被削弱的技术特性进行冲突分析, 以寻找可行的解决方案。通过 HOQ

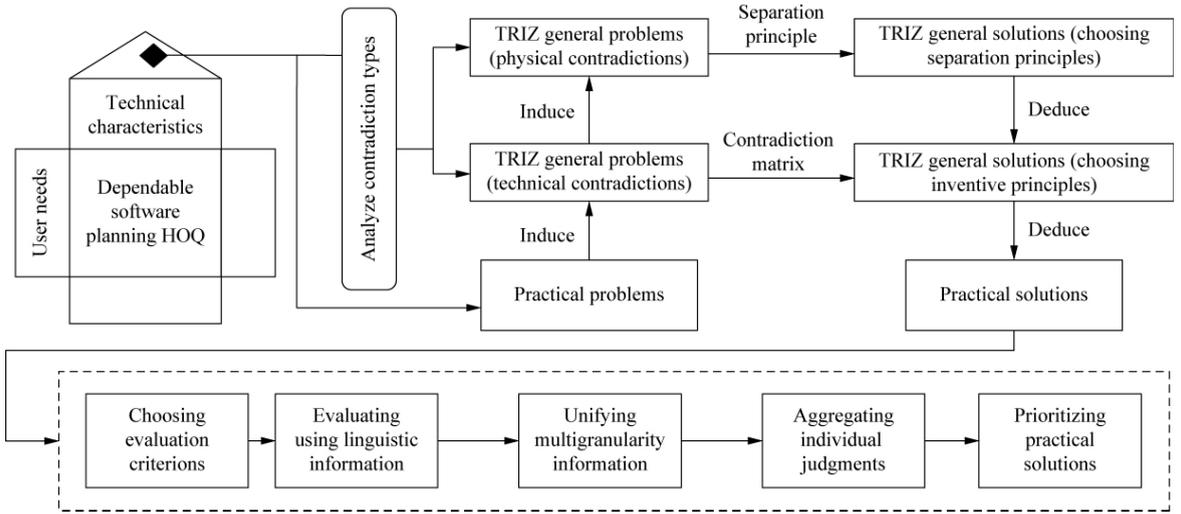


图 1 基于 QFD 和 TRIZ 的可信软件技术冲突消解模型

Fig. 1 Technical contradiction solving model for dependable software based on QFD and TRIZ

的自相关矩阵,可收集呈负相关关系的技术特性对以供分析。

步骤 3 构建 TRIZ 矛盾矩阵,获得相关的创新原理。

需注意的是,自相关矩阵中呈负相关关系的技术特性对,本身并不是以 TRIZ 的通用工程参数表达的,而是待解决的实际问题。因此,需要先将实际问题归结为 TRIZ 的标准问题,即先将负相关的技术特性抽象归纳为 TRIZ 的通用工程参数,继而以通用工程参数来表达软件开发中的技术矛盾。同时,要分析实际问题中的冲突类型,并将这些冲突进一步提炼为 TRIZ 定义的技术矛盾和物理矛盾^[16]。

对于技术矛盾的解决,可先构建局部的 TRIZ 矛盾矩阵,从中提炼出 TRIZ 推荐的发明原理;而对于物理矛盾,则要采用 4 类分离原理加以解决。需要注意的是,在一定情况下,物理矛盾经过分析后,可能转化为技术矛盾,而技术矛盾经过变换,也可能转化为物理矛盾。

步骤 4 依据创新原理演绎出技术冲突解决方案。

结合可信软件开发的专业知识,将 TRIZ 所推荐的创新原理演绎成实际的软件技术冲突解决方案。由于 TRIZ 的矛盾矩阵一般会推荐多个创新原理,通过技术人员的演绎,则可能获得多个可行的解决方案,假设方案集为 $P=[P_1, P_2, \dots, P_m]$ 。

步骤 5 根据软件可信性属性对解决方案进

行语言评价。

不失一般性,设有专家 $D_k (k=1, 2, \dots, t)$ 对上述解决方案进行评价,专家 D_k 的权重 $e = [e_1, e_2, \dots, e_t]$ 可由投票表决法^[28] 确定, $e_k \geq 0, \sum_{k=1}^t e_k = 1$ 。设可信性属性为 $A_j (j=1, 2, \dots, n)$, $d = [d_1, d_2, \dots, d_n]$ 为可信属性的权重向量, $d_j \geq 0, \sum_{j=1}^n d_j = 1$, d_j 可通过层次分析法 (Analytical Hierarchy Process, AHP)^[19] 求得。专家 D_k 在可信属性 A_j 下根据非平衡语言评估标度 S 对方案 $P_i (i=1, 2, \dots, m)$ 进行测度,得到属性评价值 $r_{ij}^{(k)} \in S$,从而构成初始的语言群决策矩阵 $R = [R_{ij}]_{m \times n}$,其中 $R_{ij} = \{r_{ij}^1, r_{ij}^2, \dots, r_{ij}^t\}$ 。考虑到各专家采用的语言信息粒度可能不同,须先用式(4)~式(7)进行一致化处理,将其转换到一个统一的语言标度集。

步骤 6 进行数据处理,得到解决方案的优先排序。

为得到群意见,可利用 LWA 算子(见式(2))将一致化后的语言群决策信息进行集成,则得到最终的群语言决策矩阵 $R^* = [r_{ij}^*]_{m \times n}$ 。继而用 LHA 算子(式(3))计算得到各方案 P_i 的综合评价价值 E_i 。最后根据综合评价价值 E_i 对各方案 P_i 进行排序,即可得到最佳的可信软件冲突解决方案。

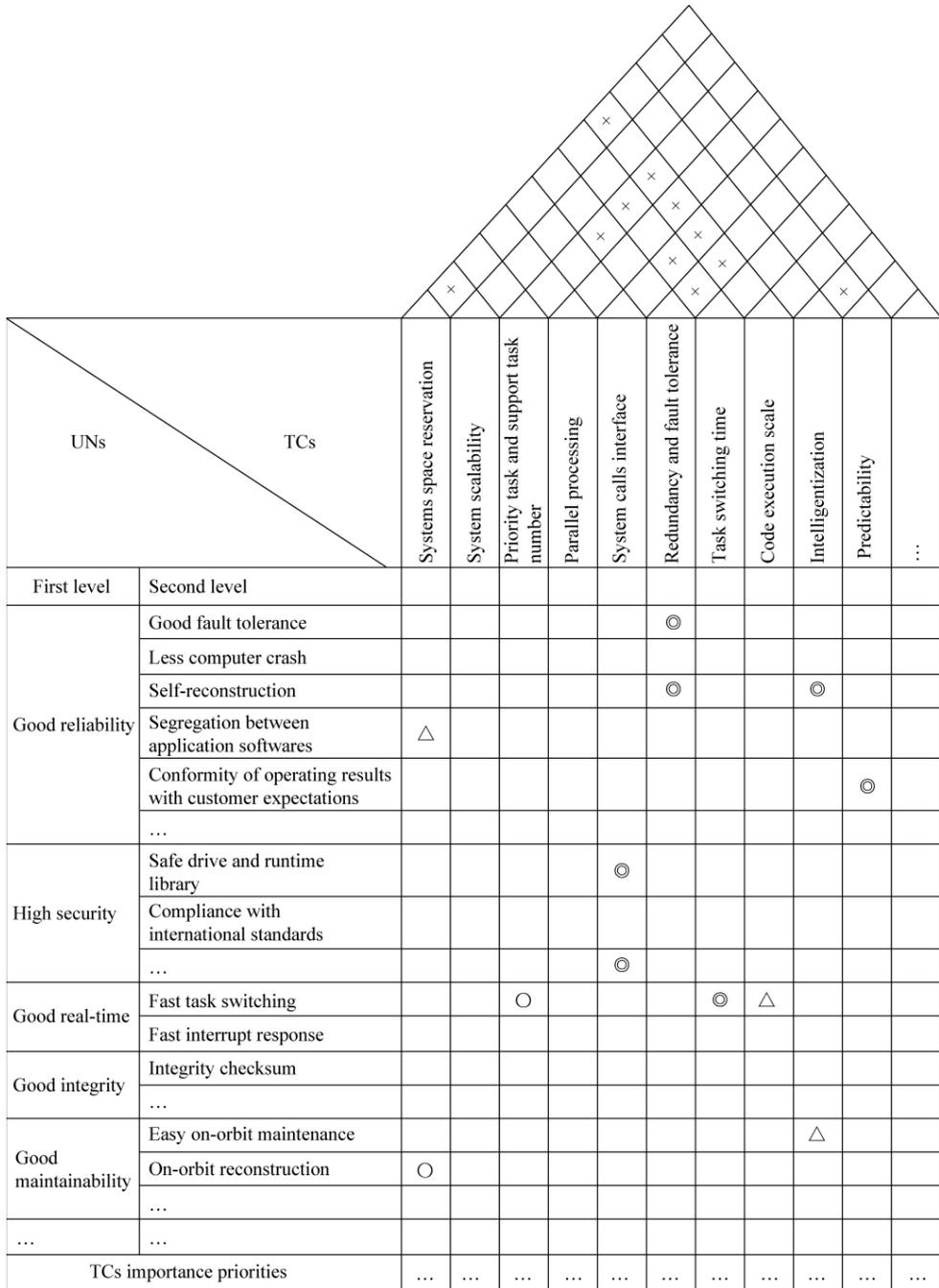
3 实例研究

(Embedded Real Time Operating Systems, ER-TOS)软件的开发工作,为提升所开发软件的质量和可信性,技术人员尝试用 QFD 和 TRIZ 来进行航天 ERTOS 软件开发。本文所提的基于 QFD 和 TRIZ 的可信软件冲突解决方法已成功地应用于该 ERTOS 软件系统的开发。限于篇幅,这里

仅给出其中的重要结果。

3.1 可信软件质量屋构建及技术冲突解决

QFD 团队在查阅可信软件相关文献及对软件用户进行访谈的基础上,确定了质量屋中的主要用户需求(User Needs, UNs),如图 2 所示。第



◎—Strong relationships; ○—Medium relationships; △—Weak relationships

图 2 航天可信软件规划质量屋(局部)

1 层次的可信性需求包括可靠性、安全性、实时性、完整性^[29]及可维护性等。图 2 中左侧两列即为层次化的可信软件需求。根据 QFD 团队中软件技术人员的专家经验和知识,确定了可信软件开发的技术特性(Technical Characteristics, TCs)。在此基础上,QFD 团队构建了可信软件规划质量屋(局部)。软件开发中不可避免地存在着技术冲突,QFD 的技术特性自相关矩阵能较好地体现这些技术冲突,见图 2 的屋顶自相关矩阵。

对上述 11 对技术冲突进行分析,要用 TRIZ 来解决技术冲突,首先需将具体的技术参数抽象化为 TRIZ 的通用工程参数。例如,在本案例中,要增加(改善)“支持任务数量及优先级数量”,就会使“任务切换时间”延长(恶化)。也即,“支持任务数量及优先级数量”为待改善的技术特性,可对应 TRIZ 的 26 号工程参数“物质或事物的数量”。而会恶化的技术特性“任务切换时间”则可抽象为“任务切换速度”,进而对应 9 号工程参数“速度”。对照 TRIZ 的矛盾矩阵,则可得到 28、29、34 和 35 共 4 条发明创新原理。类似地,对于其余的技术冲突对,也可得相应的发明创新原理。根据这些创新原理,QFD 团队的软件技术人员基于自身的经验和知识,提出了 5 个软件技术冲突解决方案(P_1, P_2, P_3, P_4, P_5),如表 1 所示。

表 1 创新原理及解决方案

Table 1 Inventive principles and solutions

Inventive principle	Practical solution
# 34 Rejecting and regenerating	P_1
# 28 System replacement	P_2
# 1 Segmentation	P_3
# 10 Prior action	P_4
# 11 Be prepared	P_5

其中各方案的具体含义为

P_1 :采用根据任务状态分组的方法,当任务状态为挂起或阻塞时,将其由就绪队列调整至相应状态的队列,以降低就绪队列的规模。

P_2 :利用位图结构代替线性表,组织任务就绪队列,使得就绪任务的搜索复杂度为 $O(1)$

操作。

P_3 :将两个相对独立的功能操作拆分为两个 CPU 来分别承担,提高系统的并行处理性能。

P_4 :利用预先准备的空间预先映射优化方法,实现用户与内核交互中数据传输的零拷贝。

P_5 :利用多版本比对方法,互相补偿,提高系统的可靠性及自适应性。

3.2 可信软件技术冲突解决方案的语言评价

不同的技术冲突解决方案对软件可信性的影响程度不同,其本身的技术可行性、成本和开发周期等因素也各不相同。因此,QFD 团队需要根据多种条件进行评价和决策,以确定解决方案实施的优先次序。

为保证所开发的航天 ERTOS 软件的可信性,以常见的软件可信性属性^[29]作为专家的评价准则,即 $A=[A_1=$ 可靠性 $A_2=$ 安全性 $A_3=$ 实时性 $A_4=$ 完整性 $A_5=$ 可维护性 $]$ 。

由于专家的偏好、对问题的理解程度以及知识背景等的差异,不同专家往往使用不同粒度的语言评估标度来表达决策意见。常见的非平衡语言评估标度有 $S^{(3)} = \{s_{-2}^{(3)} = \text{很差}, s_{-2/3}^{(3)} = \text{差}, s_0^{(3)} = \text{一般}, s_{2/3}^{(3)} = \text{好}, s_2^{(3)} = \text{很好}\}$; $S^{(4)} = \{s_{-4}^{(4)} = \text{很差}, s_{-4/3}^{(4)} = \text{差}, s_{-1/2}^{(4)} = \text{较差}, s_0^{(4)} = \text{一般}, s_{1/2}^{(4)} = \text{较好}, s_{4/3}^{(4)} = \text{好}, s_3^{(4)} = \text{很好}\}$; $S^{(5)} = \{s_{-4}^{(5)} = \text{极差}, s_{-2}^{(5)} = \text{很差}, s_{-1}^{(5)} = \text{差}, s_{-2/5}^{(5)} = \text{稍差}, s_0^{(5)} = \text{一般}, s_{2/5}^{(5)} = \text{稍好}, s_1^{(5)} = \text{好}, s_2^{(5)} = \text{很好}, s_4^{(5)} = \text{极好}\}$ 。这里,3 位专家 $D_k(k=1, 2, 3)$ (其权重向量为 $e=[0.50 \ 0.25 \ 0.25]$)利用上述语言评估标度,依 5 个可信性属性对由 TRIZ 演绎得到的 5 个方案进行评价。其中,专家 D_1 采用 $S^{(5)}$ 标度,而专家 D_2 和 D_3 则采用 $S^{(4)}$ 标度。3 位专家给出的语言评价信息构成初始的语言群决策矩阵 $R=[R_{ij}]_{5 \times 5}$,如表 2 所示。

针对上述多粒度的语言信息,首先用式(4)~式(7)进行一致化处理。考虑到有两位专家都采用 $S^{(4)}$ 标度,因此将 $S^{(4)}$ 作为统一标度集 S ,则只需将专家 D_1 的评价值进行转换: $s_2^{(5)} \rightarrow s_{3/2}^{(4)}, s_1^{(5)} \rightarrow s_{3/4}^{(4)}, s_{-2/5}^{(5)} \rightarrow s_{-3/10}^{(4)}, s_4^{(5)} \rightarrow s_3^{(4)}, s_0^{(5)} \rightarrow s_0^{(4)}$ 。

然后用 LWA 算子(见式(2))将一致化后的语言决策信息进行集成,可得最终的群语言决策矩阵 $R^*=[r_{ij}]_{5 \times 5}$ (见表 3)。

表 2 3 位专家给出的语言评价信息

Table 2 Linguistic evaluation information given by three experts

Practical solution	A_1	A_2	A_3	A_4	A_5
P_1	$s_0^{(5)}, s_0^{(4)}, s_0^{(4)}$	$s_0^{(5)}, s_{-1/2}^{(4)}, s_0^{(4)}$	$s_1^{(5)}, s_{-3}^{(4)}, s_{-1/2}^{(4)}$	$s_0^{(5)}, s_{4/3}^{(4)}, s_0^{(4)}$	$s_0^{(5)}, s_{-4/3}^{(4)}, s_{-1/2}^{(4)}$
P_2	$s_1^{(5)}, s_0^{(4)}, s_0^{(4)}$	$s_0^{(5)}, s_{-4/3}^{(4)}, s_0^{(4)}$	$s_2^{(5)}, s_{4/3}^{(4)}, s_{4/3}^{(4)}$	$s_0^{(5)}, s_{4/3}^{(4)}, s_{4/3}^{(4)}$	$s_0^{(5)}, s_3^{(4)}, s_3^{(4)}$
P_3	$s_1^{(5)}, s_0^{(4)}, s_0^{(4)}$	$s_0^{(5)}, s_0^{(4)}, s_0^{(4)}$	$s_0^{(5)}, s_{4/3}^{(4)}, s_{1/2}^{(4)}$	$s_0^{(5)}, s_{-1/2}^{(4)}, s_{4/3}^{(4)}$	$s_0^{(5)}, s_{-1/2}^{(4)}, s_{1/2}^{(4)}$
P_4	$s_1^{(5)}, s_{-4/3}^{(4)}, s_0^{(4)}$	$s_0^{(5)}, s_0^{(4)}, s_0^{(4)}$	$s_4^{(5)}, s_{4/3}^{(4)}, s_{4/3}^{(4)}$	$s_0^{(5)}, s_0^{(4)}, s_{4/3}^{(4)}$	$s_{-2/5}^{(5)}, s_{1/2}^{(4)}, s_0^{(4)}$
P_5	$s_2^{(5)}, s_3^{(4)}, s_3^{(4)}$	$s_2^{(5)}, s_{4/3}^{(4)}, s_3^{(4)}$	$s_{-2/5}^{(5)}, s_{-4/3}^{(4)}, s_0^{(4)}$	$s_1^{(5)}, s_{4/3}^{(4)}, s_0^{(4)}$	$s_{-2/5}^{(5)}, s_{-1/2}^{(4)}, s_{1/2}^{(4)}$

表 3 群语言决策矩阵

Table 3 Linguistic group decision matrix

Practical solution	A_1	A_2	A_3	A_4	A_5
P_1	$s_0^{(4)}$	$s_{-0.125}^{(4)}$	$s_{0.500}^{(4)}$	$s_{0.333}^{(4)}$	$s_{-0.458}^{(4)}$
P_2	$s_{0.375}^{(4)}$	$s_{-0.333}^{(4)}$	$s_{1.417}^{(4)}$	$s_{0.667}^{(4)}$	$s_{1.500}^{(4)}$
P_3	$s_{0.375}^{(4)}$	$s_0^{(4)}$	$s_{0.458}^{(4)}$	$s_{0.208}^{(4)}$	$s_0^{(4)}$
P_4	$s_{0.042}^{(4)}$	$s_0^{(4)}$	$s_{2.167}^{(4)}$	$s_{0.333}^{(4)}$	$s_{-0.025}^{(4)}$
P_5	$s_{2.250}^{(4)}$	$s_{1.833}^{(4)}$	$s_{-0.483}^{(4)}$	$s_{0.708}^{(4)}$	$s_{-0.150}^{(4)}$

由 AHP 法^[19]求得可信性属性 $A_j (j=1, 2, 3, 4, 5)$ 的属性权重向量 $d = [d_1 = 0.500 \quad d_2 = 0.166 \quad d_3 = 0.203 \quad d_4 = 0.063 \quad d_5 = 0.068]$ 。

以方案 P_1 为例,先计算语言加权数据: $s_{a_1} = 5 \times 0.500 \otimes s_0^{(4)} = s_0^{(4)}$, $s_{a_2} = 5 \times 0.166 \times s_{-0.125}^{(5)} = s_{-0.104}^{(5)}$, $s_{a_3} = 5 \times 0.203 \times s_{0.5}^{(4)} = s_{0.508}^{(4)}$, $s_{a_4} = 5 \times 0.063 \times s_{0.333}^{(4)} = s_{0.105}^{(4)}$, $s_{a_5} = 5 \times 0.068 \times s_{-0.458}^{(5)} = s_{-0.156}^{(5)}$ 。

对 $s_{a_i} (i=1, 2, 3, 4, 5)$ 从大到小重排,可得: $s_{\beta_1} = s_{a_3} = s_{0.508}^{(4)}$, $s_{\beta_2} = s_{a_4} = s_{0.105}^{(4)}$, $s_{\beta_3} = s_{a_1} = s_0^{(4)}$, $s_{\beta_4} = s_{a_2} = s_{-0.104}^{(5)}$, $s_{\beta_5} = s_{a_5} = s_{-0.156}^{(5)}$ 。

设 LHA 算子的位置权重向量为 $v = [0.15 \quad 0.20 \quad 0.30 \quad 0.20 \quad 0.15]^T$, 利用式(3)对方案 P_1 的评价值进行集结,得到 P_1 的群体综合评价值 $E_1 = s_{0.053}^{(4)}$ 。同理可得 $E_2 = s_{0.557}^{(5)}$, $E_3 = s_{0.254}^{(4)}$, $E_4 = s_{0.381}^{(4)}$, $E_5 = s_{1.131}^{(5)}$ 。

根据群体综合评价值 $E_i (i=1, 2, 3, 4, 5)$ 对可信软件技术冲突解决方案 P_i 进行排序,得

$$P_5 > P_2 > P_4 > P_3 > P_1$$

其中:“>”表示“优于”。可见,在可靠性、安全性、实时性、完整性和可维护性等可信性属性条件下,软件技术冲突解决方案 P_5 最佳。从各个可信性属性的权重也可看出,用户对航天 ERTOS 软件的可靠性、实时性和安全性要求更高,而这 3 个属性也是软件可信性最重要的特征,方案评价的结

果也反映了这一倾向。通过多版本比对的方法,实现互相补偿,可有效解决航天软件的“自主性”与“可预期性”之间的矛盾,既提高系统的可靠性,同时也增强系统的自适应性。

4 结论

所提出的基于 QFD 和 TRIZ 的可信软件技术冲突解决方法,可以系统化地识别技术冲突,分析技术矛盾,提出解决方案,评价和优选最可行的方案。在用 TRIZ 的创新原理解决技术冲突时,往往可演绎得到多个可行的备选方案,考虑到专家评价语言的模糊性、不确定性和多粒度多语义的情况,用近年来最新发展的基于语言信息的决策理论来评估和选择最可行的解决方案。以某航天嵌入式实时操作系统软件的开发实例验证了本文所提方法的可行性和有效性。本文提出的基于 QFD 和 TRIZ 的可信软件技术冲突解决方法为航天高可信软件的开发提供了新的思路,有助于技术人员识别和解决可信软件开发中的关键技术冲突,对保证安全关键应用领域中软件系统的可信性具有重要意义。

致谢

感谢浙江大学计算机科学与技术学院蔡铭副教授对本研究给予的大力支持、帮助和指导。

参考文献

[1] 陈火旺,王戟,董威.高可信软件工程技术[J].电子学报,2003,31(12A):1933-1938.
Chen Huowang, Wang Ji, Dong Wei. High confidence software engineering technologies [J]. Acta Electronica Sinica, 2003, 31(12A): 1933-1938. (in Chinese)

[2] 刘克,单志广,王戟,等.“可行软件基础研究”重大研究计划综述[J].中国科学基金,2008,22(3):145-151.

- Liu Ke, Shan Zhiguang, Wang Ji, et al. Overview on major research plan of trustworthy software [J]. Bulletin of National Natural Science Foundation of China, 2008, 22(3): 145-151. (in Chinese)
- [3] Boleat C, Colas G, Astrium E, et al. Overview of soft errors issues in aerospace systems[C]//11th IEEE International On-Line Testing Symposium. Toulouse: IEEE, 2005: 299-302.
- [4] Littlewood B, Stringini L. Software reliability and dependability: a roadmap [C]// Proceedings of the Conference on the Future of Software Engineering. Limerick: ACM, 2000: 175-188.
- [5] Avizienis A, Laprie J C, Randell B, et al. Basic concepts and taxonomy of dependable and secure computing [J]. IEEE Transactions on Dependable and Secure Computing, 2004, 1(1): 11-33.
- [6] Saiedian H, Dale R. Requirements engineering: making the connection between the software developer and customer [J]. Information and Software Technology, 2000, 42(6): 419-428.
- [7] Chan L K, Wu M L. Quality function deployment: a literature review [J]. European Journal of Operational Research, 2002, 143(3): 463-497.
- [8] Houser J R, Clousing D. The house of quality [J]. Harvard Business Review, 1988, 66(3): 63-73.
- [9] 马健, 唐晓青, 李军. 质量功能配置软件工具中的模糊处理[J]. 航空学报, 1997, 18(9): 698-702.
Ma Jian, Tang Xiaqing, Li Jun. Fuzzy process in quality function deployment system [J]. Acta Aeronautica et Astronautica Sinica, 1997, 18(9): 698-702. (in Chinese)
- [10] Haag S, Raja M K, Schkade L L. Quality function deployment usage in software development [J]. Communications of the ACM, 1996, 39(1): 41-49.
- [11] Herzwurm S, Schockert G. The leading edge in QFD for software and electronic business [J]. International Journal of Quality & Reliability Management, 2003, 20(1): 36-55.
- [12] 熊伟, 王晓墩. 基于质量功能展开的可信软件需求映射方法[J]. 浙江大学学报: 工学版, 2010, 44(5): 881-886.
Xiong Wei, Wang Xiaotun. Method for mapping software dependability requirements based on QFD [J]. Journal of Zhejiang University: Engineering Science, 2010, 44(5): 881-886. (in Chinese)
- [13] 石贵龙, 余元冠. 基于 QFD、TRIZ 和田口方法的问题解决理论研究[J]. 系统工程与电子技术, 2008, 30(5): 851-857.
Shi Guilong, She Yuanguan. Study on problem-solving theory based on QFD, TRIZ and Taguchi methods [J]. Systems Engineering and Electronics, 2008, 30(5): 851-857. (in Chinese)
- [14] Şen C G, Baraçlı H. Fuzzy quality function deployment based methodology for acquiring enterprise software selection requirements [J]. Expert Systems with Applications, 2010, 37(4): 3415-3426.
- [15] Wasserman G S. On how to prioritize design requirements during the QFD planning process [J]. IIE Transactions, 1993, 25(3): 59-65.
- [16] 檀润华. 发明问题解决理论[M]. 北京: 科学出版社, 2004.
Tan Runhua. Theory of inventive problem solving [M]. Beijing: Science Press, 2004. (in Chinese)
- [17] Li T S, Huang H H. Applying TRIZ and fuzzy AHP to develop innovative design for automated manufacturing systems [J]. Expert Systems with Applications, 2009, 36(4): 8302-8312.
- [18] 马怀宇, 孟明辰. 基于 TRIZ/QFD/FA 的产品概念设计过程模型[J]. 清华大学学报: 自然科学版, 2001, 41(11): 56-59.
Ma Huaiyu, Meng Mingchen. Model of the conceptual design process based on TRIZ /QFD /FA [J]. Journal of Tsinghua University: Science & Technology, 2001, 41(11): 56-59. (in Chinese)
- [19] Yang W, Wu Q M, Chen Y Q, et al. Research on inventive problem solving process model based on AHP / TRIZ [C] // International Technology and Innovation Conference (ITIC 2006). Hangzhou: IEEE, 2006: 2285-2290.
- [20] 李延来, 唐加福, 姚建明, 等. 质量屋构建的研究进展 [J]. 机械工程学报, 2009, 45(2): 280-293.
Li Yanlai, Tang Jiafu, Yao Jianming, et al. Progress of researches on building house of quality [J]. Chinese Journal of Mechanical Engineering, 2009, 45(2): 280-293. (in Chinese)
- [21] Wang S Y. Constructing the complete linguistic-based and gap-oriented quality function deployment [J]. Expert Systems with Applications, 2010, 37(2): 908-912.
- [22] Xu Z S. A note on linguistic hybrid arithmetic averaging operator in group decision making with linguistic information [J]. Group Decision and Negotiation, 2006, 15(6): 581-591.
- [23] Xu Z S. On generalized induced linguistic aggregation operators [J]. International Journal of General Systems, 2006, 35(1): 17-28.
- [24] Herrera F, Martínez L. A 2-tuple fuzzy linguistic representation model for computing with words [J]. IEEE Transactions on Fuzzy Systems, 2000, 8(6): 746-752.
- [25] Jayaswal B K, Patton P C. Design for trustworthy software [M]. New Jersey: Person Education Inc, 2006.
- [26] Zadeh L A. The concept of a linguistic variable and its application to approximate reasoning (Part I) [J]. Information Sciences, 1975, 88(3): 199-249.

- [27] Xu Z S. An interactive approach to multiple attribute group decision making with multigranular uncertain linguistic information [J]. *Group Decision and Negotiation*, 2009, 18(2): 119-145.
- [28] Bodily S E. A delegation process for combining individual utility functions [J]. *Management Science*, 1979, 25(10): 1035-1041.
- [29] 沈玉龙, 崔西宁, 马建峰, 等. 综合化航空电子系统可信软件技术[J]. *航空学报*, 2009, 30(5): 938-945.
Shen Yulong, Cui Xining, Ma Jianfeng, et al. Trust software technology in integrated avionics systems [J]. *Acta Aeronautica et Astronautica Sinica*, 2009, 30(5): 938-945. (in Chinese)

作者简介: 王晓敏(1984—) 男, 博士研究生。主要研究方向: 质量工程与管理, 产品研发管理。

E-mail: wangxt84@gmail.com

熊伟(1963—) 男, 博士, 教授, 博士生导师, 主要研究方向: 质量管理与质量工程、软件质量保证。

E-mail: zjuquality@yahoo.com.cn

Solution of Technical Contradiction for Dependable Software Based on QFD and TRIZ

WANG Xiaotun, XIONG Wei*

School of Management, Zhejiang University, Hangzhou 310058, China

Abstract: In order to improve software dependability, a method is proposed for solving technical contradictions in dependable software development based on quality function deployment (QFD) and the Russian theory of inventive problem solving (TRIZ). First, a dependable software planning house of quality (HOQ) is built to introduce the voice of customers into the software development process. Then, the technical characteristic pairs whose relationships are negative in the technical correlation matrix are found out and analyzed in detail. The inventive principles of TRIZ are applied to solve the technical contradictions, and several feasible solutions or inspirations could be acquired. Considering the vagueness and uncertainty of expert judgment, and the case of multi-granularity and multi-semantics in an expert's linguistic statement, the linguistic information decision-making theory developed in recent years is introduced to evaluate and select the best feasible alternative. Finally, a real-world case of astronautics of an embedded real time operating system (ERTOS) software development is provided to demonstrate the validity and applicability of the proposed method.

Key words: software dependability; technical contradiction; quality function deployment; TRIZ; multi-granularity; linguistic variable; alternative evaluation