

# 基于输入域分类的实时软件可靠性模型

宋治平<sup>1,2</sup>, 秦湘河<sup>2</sup>, 朱怡安<sup>1</sup>

(1. 西北工业大学, 陕西 西安 710043;

2. 西安卫星测控中心, 陕西 西安 710043)

**摘 要:** 航天测控软件是航天测控系统的重要组成部分, 作为任务的关键性实时软件, 其可靠性直接影响着整个测控系统的可靠性。长期以来, 测控系统软件相对缺乏系统的可靠性测试研究。文中针对航天工程地面测控软件的特点及可靠性测试要求, 综合考虑缺陷等级、输入域等因素, 按照数学算法建立了基于输入域分类的实时软件的可靠性混合模型, 并对该模型的实际应用进行了介绍。文中模型和测试算法对航天测控工程领域典型功能软件进行了可靠性测试, 实验结果表明了该方法的可行性、有效性和易用性。

**关键词:** 航天测控软件; 测试输入域; 缺陷等级; 可靠性模型

**中图分类号:** TP31

**文献标识码:** A

**文章编号:** 1673-629X(2013)07-0096-05

**doi:** 10.3969/j.issn.1673-629X.2013.07.024

## Reliability Model of Real-time Software Based on Input Domain Classification

SONG Zhi-ping<sup>1,2</sup>, QIN Xiang-he<sup>2</sup>, ZHU Yi-an<sup>1</sup>

(1. Northwestern Polytechnical University, Xi'an 710043, China;

2. Xi'an Satellite Control Center, Xi'an 710043, China)

**Abstract:** TT&C software is one of the important composing of the TT&C system. Its reliability makes a direct impact on the reliability of the TT&C system. For a long time, the measurement and control system software is relative lack of system reliability testing research. In this paper, aiming at the aerospace engineering ground measurement and control software system characteristics and reliability testing requirements, considering synthetically some factors such as input domains and defect levels, according to the mathematical algorithm, the reliability mix model of the real time software based on the sort of input domain is established. The applications of the established model are introduced. The typical functional software is tested in reliability with model and test algorithm for aerospace measurement control project areas, the experimental results show that the method is feasible, effective and easy to use.

**Key words:** TT&C; test input domain; defect level; reliability model

## 0 引言

航天测控系统是指对导弹、卫星、飞船等飞行器进行跟踪、测量和控制的技术系统, 其软件的稳定和可靠直接影响航天测控任务的成败。航天测控软件具有严格的实时性、可靠性和高精度要求。随着我国航天事业的飞速发展, 航天测控软件开发量不断增多, 软件的质量要求不断提高, 对软件测试工作的效率、质量提出了更高的要求。

在软件比重日益增加的今天, 研究系统的软件可

靠性对整个产品质量的提升具有很大的现实意义<sup>[1]</sup>。软件可靠性测试是为了验证待测软件是否达到用户的可靠性指标, 通过测试发现并纠正软件中的缺陷, 提高其可靠性水平, 并验证它是否达到了用户的可靠性要求。文献[2]介绍了软件可靠性测试的现状, 然后重点对现有可靠性理论进行了分析和总结, 指出软件可靠性模型的不一致性以及软件可靠性测试时间长、费用高、资源消耗大是操作软件可靠性测试没有真正开展的主要原因, 并在此基础上提出了软件可靠性测试

收稿日期: 2012-09-21

修回日期: 2013-01-06

网络出版时间: 2013-03-05

基金项目: 总装备部重大专项课题(2009SY2603002)

作者简介: 宋治平(1981-), 男, 工程师, 硕士, 主要研究方向为软件工程与软件评测; 秦湘河, 高级工程师, 硕士, 主要研究方向为软件工程与软件评测; 朱怡安, 教授, 博士生导师, 主要研究方向为高性能计算、云计算、普适计算。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20130305.0819.053.html>

研究的一些新思路。文献[3]进行了软件可靠性统计方法研究,运用输入空间和输入分布的概念,分析软件测试数据的特征,给出软件可靠性的点估计和置信下界的计算公式;文献[4,5]在传统的基于操作剖面的可靠性测试方法基础上提出了一种层次化网络体系结构的使用剖面的概念,研究了构造使用剖面的系统方法和形式化的建模方法,有效支持可靠性测试数据的自动生成;文献[6]提出了基于操作剖面的软件可靠性压力测试思想和操作剖面、压力测试点相互结合、互为补充的软件可靠性测试方法,并给出了软件可靠性测试新方法实施的技术途径;文献[7,8]阐述了软件可靠性测试的机理,对国内外安全关键软件的可靠性加速测试方法和基于灰盒的软件可靠性加速测试方法的研究成果和进展进行了分析、总结和论述,并在此基础上研究了软件可靠性测试存在加速的可能性,提出了实现加速的思想和方法。软件可靠性工程理论上取得了较大的突破,但在工程实践上开展的却不多<sup>[9]</sup>。

长期以来,测控系统软件测试注重基于结构分析的测试,重点关注软件功能和性能等要求,相对缺乏成体系的可靠性测试研究。文中针对航天工程地面测控软件系统的特点和可靠性测试要求,基于输入域和数据驱动特点对实时软件系统的测试按照数学算法进行可靠性建模,并结合具体工程实践活动给出其实际使用效果。

## 1 软件可靠性模型和建模的一般过程

软件可靠性模型用于在开发早期预计软件的可靠性,它主要关注于缺陷的去除情况。软件可靠性模型包括3个部分:假设、可靠性因子以及把可靠性与这些因子相关联的数学函数。通过软件可靠性模型,可以在项目的测试阶段评价开发的状态,监视软件的操作性能,控制设计的变更和新特性的引入。

### 1.1 软件可靠性模型

自20世纪70年代以来,人们对软件可靠性测试进行了大量研究,提出了各种软件可靠性模型。软件可靠性模型大体上可分为两类<sup>[10,11]</sup>:解析模型和数据驱动模型。

解析模型即为传统的软件可靠性增长模型,包括马尔可夫模型、非齐次泊松过程模型等。这类模型需要对软件内错误及失效过程的特性提出假设,并用某种随机过程进行描述。例如 Jelinski - Moranda 模型<sup>[12,13]</sup>假设软件失效过程是一个马尔可夫过程,并且所有软件内错误引起的软件失效率都相同。Geol - Okumoto 模型<sup>[12-13]</sup>则假设软件失效过程是一个非齐次泊松过程,并且对每次观察到的软件失效,其对应的软件内错误都可以准确地进行排除。各种解析模型提出

的各种假设都或多或少存在于软件实际失效过程不相符的现象。同时,在实际工程应用中,选择合适的模型进行描述比较困难。因此,这些都影响了解析模型的准确性和适用性。

数据驱动的软件可靠性建模方法一般情况下不需要对软件的失效过程做任何假设,而是通过观测到的软件失效数据,将其视为一个时间序列进行分析与建模,并能够预测软件将来的失效行为。数据驱动的软件可靠性模型包括基于人工神经网络的模型、基于时间序列分析的模型<sup>[14,15]</sup>和基于支持向量机的模型等。但也存在一些内在的缺点,人工神经网络模型存在数据过拟合以及学习算法的效率不高等问题<sup>[16]</sup>,支持向量机的模型存在核函数寻找难度大、计算复杂度高等问题<sup>[17]</sup>。

### 1.2 建模的一般过程

软件可靠性建模过程就是用数学模型模拟软件的可靠性特征,并给出可靠性特征的定量估计。建模过程由以下几部分组成:

(1)提出模型假设。由于目前人们对软件可靠性特征中的某些特征还无法确知,或某些特征本身具有不确定性。因此,需要基于某种假设进行软件可靠性建模;另外是为了便于进行数学处理。

(2)选择度量方式。根据需要,从直接的、间接的,甚至辅助的各种度量中,选择一种或多种度量来估计软件的可靠性。

(3)数学建模。将已经选择的可靠性度量,表示为软件的某些特性函数。

(4)参数估计。对某些通过模型无法直接获得的度量或参数,需要使用某种参数估计方法来确定它们的值。

(5)确定数据输入域。模型中未知参数需要通过搜集的故障数据来确定,而这些故障数据则需要通过软件运行进行收集。因此,需要确定数据输入域。

## 2 基于输入域分类的实时软件可靠性模型

航天测控软件系统设计一般采用基于重要测控事件的作业调度与数据驱动式相结合的方法,系统底层服务软件系统的处理结果为应用层姿态控制、轨道确定、卫星遥控注入等上层应用软件的处理输入,软件处理对实时性、容错性和精度要求较高,较少需要人工 GUI 交互操作。测控软件处理的测控数据,都是具有特定制约关系的一组数据诸元构成的复杂结构;另一方面其基于关键事件的调度运行模式决定了软件运行具有一定程度的规律性。对于航天测控软件系统的可靠性测试,需要对软件的运行和使用环境有较充分的了解,明确软件的所有运行状态,状态转换条件、各种

运行环境下输入边界,并对输入的概率分布有大致的估计。整个软件系统可以在真实和模拟环境下完整的运行。针对航天工程地面测控软件系统的特点及对其可靠性的测试要求,基于测试输入域和数据驱动特点对实时软件系统的测试按照数学算法进行建模,提出基于输入域分类的实时软件可靠性模型。

### 2.1 输入域选择方案设计

软件测试是在一定的环境条件下,通过一系列设计好的测试用例验证软件是否达到预先设计好的目标和要求。测试用例在设计时,既要包含正常情况下的输入测试,也要包含异常情况下的测试。通过提供用例输入,观察测试输出结果,将输入与输出进行比较,确定软件是否满足要求。软件可靠性测试多采用黑盒测试,其用例设计方法包括等价类划分法、边界值分析法、因果图法、判定表驱动法、正交试验设计法、猜错法等<sup>[18]</sup>。然而,测试用例在设计时,不可能穷举所有的输入,只能通过选取输入域中有代表的集合进行测试,例如边界值等。因此,测试用例的选择直接影响软件可靠性模型的可信性,完善的测试用例是可靠性模型的保证<sup>[19]</sup>。

测试数据输入域主要从软件需求规格说明、程序观察和额外的属性规约中得到。其中,额外的属性规约是指在需求规格说明中没有进行描述,但在测试中可能用到的那部分数据。为了便于进行测试,从输入域中选取输入点可按照下列方式进行:

首先,按照模块化程序设计思想,一个软件系统由多种功能组成,不同功能或模块的输入数据属性不同,软件功能模块运行具有相对独立性,可按照软件功能或模块结构将输入域分成若干个互不相交的子域,正确划分输入域是完成有效测试的关键。

其次,在划分好的各个子域中选取特征值点,特征值是指各子域中具有代表性的合理输入,构成集合  $S_1$ 。考察各子域特征值点的正确性,可增大该子域所有正常输入的正确概率。

第三,考虑采用边界值分析法,从各相邻子域边界值点集中选取  $N$  个测试点,如果有  $P$  个子域,则测试点个数为  $P \times N$  个,构成集合  $S_2$ 。测试点选择方法有多种,可采用随机选取或按照一定的概率分布选取,具体方法视实际测试需要而定。输入域选择过程,如图 1 所示。

### 2.2 软件可靠性度量模型

为推导可靠性模型,现作如下定义:

(1) 测试用例输入域:设输入域为  $K$ ,根据上述描述步骤可得  $K = S_1 \cup S_2$ 。按照软件功能模块,将  $K$  划分成  $L$  个互不相交的子域  $K_1, K_2, K_3, \dots, K_L$ ,即  $K = K_1 \cup K_2 \cup \dots \cup K_L$ ,且  $K_i \cap K_j = \emptyset (i \neq j, \text{且 } i, j = 1, 2, \dots,$

$L)$ 。将属于同一个子域的元素排成一列,对于每一个输入域  $K_i$ ,选取测试点个数为  $m_i (i = 1, 2, \dots, L)$ ,取  $M = \max\{m_i\}$ ,所有选取的测试点组成一个  $M$  行  $L$  列的矩阵。取该矩阵中各行向量为测试向量,那么输入域就包含  $M$  个  $L$  维的测试向量  $[V_1, V_2, \dots, V_n, \dots, V_M]^T$ ,且测试向量  $V_1, V_2, \dots, V_n, \dots, V_M$  相互独立,  $V_n = \{v_{n1}, v_{n2}, \dots, v_{ni}, \dots, v_{nL}\} (n = 1, 2, \dots, M; i = 1, 2, \dots, L)$ 。

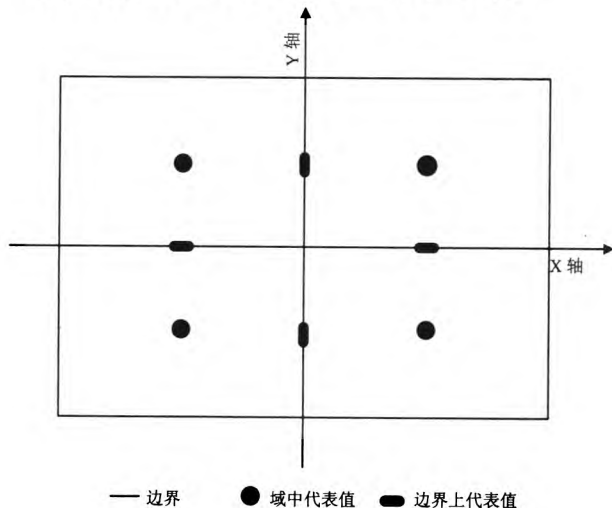


图 1 输入域选择示意图

(2) 可靠性测试函数:用可靠性测试函数  $C$  来表示测试结果,即测试输入  $v_{ni}$  与输出  $C(v_{ni})$  的关系。它满足  $C(v_{ni}) = 1$  或者  $C(v_{ni}) = 0$ ,取 1 表示输入  $v_{ni}$  引起了软件缺陷,取 0 表示输入不引起相应缺陷。

(3) 缺陷影响因子:用缺陷影响因子  $d$  表示不同缺陷对软件可靠性的影响程度。软件错误一般可分为 6 类:较小错误、一般错误、较严重错误、严重错误、非常严重错误和最严重错误。因此,将缺陷影响因子  $\alpha$  分为 6 个等级。 $\alpha(v_{ni})$  表示输入  $v_{ni}$  对应的缺陷影响因子,根据前面所述分类,  $d(v_{ni})$  表示为向量  $[\alpha_1, \alpha_2, \dots, \alpha_6]$ ,取值可以设定为  $[0.1, 0.25, 0.5, 0.7, 0.8, 1]$ ,向量中的每一个分量  $\alpha_i (i \text{ 取整数, } i \in [1, 6])$  的取值可以根据实际情况进行调整。 $\alpha_i$  值越大,表明这个缺陷对软件可靠性的影响越大。

(4) 缺陷发生概率:  $P(v_{ni})$  表示输入  $v_{ni} (v_{ni} \in K)$  发生缺陷的概率。

根据上述定义,得到一个通用的计算软件可靠性的公式:

$$\Gamma = \int_{v_{ni} \in K} P(v_{ni}) C(v_{ni}) \alpha(v_{ni}) dp \quad (1)$$

### 2.3 软件可靠性测试算法

测试向量表示方式为:  $V_n = \{v_{n1}, v_{n2}, \dots, v_{ni}, \dots, v_{nL}\} (n = 1, 2, \dots, M; i = 1, 2, \dots, L)$ ,其中的  $L$  个元素  $v_{ni}$  分别属于  $L$  个输入域。由于每个输入域  $K_i$  中选取的测试点个数不尽相同,按照测试用例的设计要求,对测试向量的取值按如下规则选取:

$$v_{ni} = \begin{cases} \text{当 } v_{ni} \text{ 有输入数据时, } v_{ni} \in R \\ \text{当 } v_{ni} \text{ 没有输入值时, } v_{ni} = \varnothing \end{cases}$$

由定义(2)可知,对于输入  $v_{ni}$  可能产生缺陷也可能不产生缺陷,那么对于测试向量  $V_n$  得到的可靠性测试函数用  $C_n$  表示:

$$C_n = [c_{1i}, c_{2i}, \cdots, c_{6i}, \cdots, c_{Li}] \tag{2}$$

$c_i = 1$  或  $0 (i \text{ 为整数且 } i \in [1, L])$ ,  $1$  表示产生缺陷,  $0$  表示没有缺陷。

由定义(4)可知,产生的缺陷被分为 6 个等级,将  $c_i$  扩展为 6 维列向量:

$$c_i = [c_{1i}, c_{2i}, c_{3i}, c_{4i}, c_{5i}, c_{6i}]^T \tag{3}$$

$c_{mi} = 1$  或  $0 (m \text{ 为整数且 } m \in [1, 6])$ , 对于每一个  $v_{ni}$ ,  $c_i$  最多只有一个元素为 1。

对于由多个输入数据所构成的输入向量  $V_n$  产生的缺陷等级的描述用一个  $6 \times L$  的二维矩阵表示。对应的  $C_n$  是一个 6 行  $L$  列的二维矩阵,行标  $m$  即为对应的缺陷等级,列标  $i$  即为相应的输入域:

$$C_n = \begin{bmatrix} c_{11} & \cdots & c_{1L} \\ \cdots & \cdots & \cdots \\ c_{61} & \cdots & c_{6L} \end{bmatrix} \tag{4}$$

如果矩阵列向量  $c_i$  的每个分量  $c_{mi}$  均为 0,表示测试向量  $V_n$  中的分量  $v_{ni}$  没有产生缺陷,这是因为测试输入  $v_{ni}$  符合设计要求或是该测试输入  $v_{ni} = \varnothing$ ;  $c_{mi} = 1$  表示测试向量  $V_n$  中的分量  $v_{ni}$  产生了缺陷,缺陷等级为  $m (m \in [1, 6])$ 。

对每一个测试向量  $V_n$  进行测试,均可得到相应  $C_n$ , 因为测试向量  $V_1, V_2, \cdots, V_M$  相互独立,可对  $C_n$  求和,则可得可靠性因子矩阵  $C$ :

$$C = \sum_{n=1}^M C_n \tag{5}$$

对每一个测试向量,采用均值法处理,并考虑缺陷影响因子,可靠度可表示为:

$$\Gamma_{V_i} = (\sum_{n=1}^L \alpha \times C_n) / L \tag{6}$$

其中,  $\alpha$  表示各个缺陷等级的影响因子,  $L$  表示输入子域的个数。  $\Gamma_{V_i}$  值越大,缺陷越严重,可靠性越低。可靠度也可表示为:

$$Q_{V_i} = 1 - \Gamma_{V_i} \tag{7}$$

这时  $Q_{V_i}$  与可靠性正相关。

因为测试向量  $V_1, V_2, \cdots, V_M$  相互独立,可采用均值法来处理  $M$  个测试向量的测试结果,对于输入数据按照等概率考虑,取输入产生缺陷概率  $P = 1$ ,则得到基于测试用例输入域的可靠度可表示为:

$$Q_{\text{系统测试}} = \frac{(1 - \frac{0.1}{13}) + (1 - \frac{0.7 + 0.25}{13}) + (1 - \frac{0.1}{13}) + (1 - \frac{0.25}{13}) + (1 - \frac{0.1}{13}) + (1 - \frac{0.5}{13}) + 6}{12} = 0.9871$$

$$Q = \frac{1}{M} \sum_{i=1}^M Q_{V_i} \tag{8}$$

3 应用实例

在某型号卫星任务参试的实时外测处理软件中,按照软件功能模块划分输入域,在各子域中提取特征值点,再从各子域相邻边界值中选取一系列的测试点集。将待测数据共划分为 13 个子域,因此  $L = 13$ ,得到如表 1 所示的测试用例的输入域  $K = K_1 \cup K_2 \cup \cdots \cup K_{13}$ ,其中各输入域中测试点的最大值为 12,即  $M = 12$ ,由此可以得到一个 12 行 13 列的测试矩阵。则可得该软件的测试向量为  $[V_1, V_2, \cdots, V_{12}]^T$ 。

表 1 软件测试输入域分布

输入子域序号	特征值点数目	边界间测试点	测试点总和	输入子域序号	特征值点数目	边界间测试点	测试点总和
1	5	4	9	8	4	3	7
2	4	6	10	9	5	3	8
3	3	4	7	10	4	4	8
4	6	6	12	11	3	5	8
5	5	4	9	12	6	4	10
6	2	3	5	13	7	3	10
7	6	6	12				

在软件功能及可靠性测试中,对软件的各种功能及可靠性进行 2 个月的详细测试,在测试过程中收集到的数据,测试向量统计和缺陷分布如表 2 所示。其测试结果统计如表 3 所示。

表 2 测试向量和缺陷分布

软件模块	1	2	3	4	5	6	7	8	9	10	11	12	13
发生缺陷次数	1	0	2	1	0	1	0	1	1	0	0	0	0
相应缺陷等级	1	0	2,4	1	0	2	0	1	3	0	0	0	0

表 3 软件测试结果统计

测试用例数	程序失效数	较小错误	一般错误	较严重错误	严重错误	非常严重重错误	最严重错误
115	7	3	2	1	1	0	0

取缺陷影响因子  $\alpha = (0.1, 0.25, 0.5, 0.7, 0.8, 1)$ , 相应的可靠性因子矩阵  $C$  表示为:

$$C = \begin{bmatrix} 1001000100000 \\ 0010010000000 \\ 0000000010000 \\ 0010000000000 \\ 0000000000000 \\ 0000000000000 \end{bmatrix}$$

利用公式计算得到被测试软件的可靠度为:

该软件产品经过两个月的实际测试后投入运行,在连续半年的工作中未出现任何软件失效现象,使相关可靠性研究在实际系统中进一步得到验证。

#### 4 结束语

软件测试的目的是为了提高软件的可靠性,软件可靠性评价的结果又可以为软件测试服务。通过对软件测试及软件可靠性评价技术的研究,选取了适用的可靠性统计模型,并通过该模型计算出了某软件模块的可靠性以及软件的可靠性。该研究成果不仅能够充分利用这些测试结果,满足在不同阶段对软件可靠性进行评价的需要,而且有利于衡量软件测试工作的充分性。该模型具有简单、实用、易于工程化等特点,但该模型的适用范围及模型的一致性等问题有待进一步研究。

#### 参考文献:

- [1] 张玲,袁娜,马永刚,等. 基于测试用例和时间域软件可靠性模型[J]. 计算机技术与发展,2009,19(11):167-170.
- [2] 刘志方,钟德明,曾福萍,等. 软件可靠性测试的理论分析[J]. 测控技术,2008,27(10):62-64.
- [3] 程维虎,杨振海. 软件可靠性模型和估计[J]. 数理统计与管理,2010,29(1):52-61.
- [4] 艾骏,陆民燕,阮铤. 面向软件可靠性测试数据生成的剖面构造技术[J]. 计算机工程,2006,32(22):7-9.
- [5] 艾骏,陆民燕,阮铤. 实时嵌入式软件可靠性测试数据自动生成方法[J]. 测控技术,2007,26(3):59-61.
- [6] 蔡建平. 软件可靠性测试方法新探[J]. 计算机工程与设计,2009,30(20):4658-4661.
- [7] 吴玉美,阮铤. 软件可靠性测试的加速机理研究[J]. 计算机应用,2006,26(6):1449-1451.
- [8] 吴玉美,陆民燕,阮铤. 软件可靠性加速测试方法研究[J]. 计算机工程与应用,2006,42(8):62-64.
- [9] 李秋英,李海峰,徐刚. 基于覆盖率信息的软件可靠性增长测试实践[J]. 计算机应用研究,2010,27(7):2594-2597.
- [10] Hu Q P, Dai Y S, Xie M, et al. Early software reliability prediction with extended ANN model[C]//Proceedings of the 30th Annual International Computer Software and Applications Conference. [s. l.]:IEEE Computer Society Press,2006:234-239.
- [11] Hu Q P, Xie M, Ng S H, et al. Robust recurrent neural network modeling for software fault detection and correction prediction[J]. Reliability Engineering and System Safety,2007,92(3):332-340.
- [12] Xie M. Software Reliability Modeling[M]. Singapore:World Scientific Publisher,1991.
- [13] Pham H. Software Reliability[M]. Singapore:Springer-Verlag,2000.
- [14] Butschy B, Albeanu G, Boros D N, et al. Improving software reliability forecasting[J]. Microelectronics and Reliability,1997,37(6):901-907.
- [15] Guo J, Liu H, Yang X, et al. A software reliability time series growth model with Kalman filter[J]. WSEAS Transactions on Computers,2006,5(1):1-7.
- [16] 杨波,黄洪钟,郭凤昌. 数据驱动的软件可靠性模型研究[J]. 中国科技论文在线,2007(10):768-774.
- [17] 楼俊钢,江建慧,帅春燕,等. 软件可靠性模型研究进展[J]. 计算机科学,2010,37(9):13-27.
- [18] 吴艳征,宋志强. 浅谈黑盒测试用例设计方法[J]. 科技信息(学术研究),2008(16):178-179.
- [19] 蒋天乐,徐国治. 软件缺陷及软件可靠性技术[J]. 计算机仿真,2004,21(2):141-144.

(上接第 95 页)

#### 参考文献:

- [1] Loh A. Exploring Generic Haskell[D]. Utrecht:Utrecht University,2004.
- [2] Frisch A, Garrigue J, Rémy D, et al. The OCaml system release 4.00[M]. [s. l.]:Institut National de Recherche en Informatique et en Automatique,2012.
- [3] Johan J, Andres L, Dave C. Dependency-style generic Haskell[J]. SIGPLAN Notices,2003,38(9):12-13.
- [4] Jansson P, Jeuring J T, Cabenda, et al. Testing properties of generic functions[R]. Utrecht:Utrecht University,2006.
- [5] Hinze R. Polymorphic programming with ease (extended abstract)[M]//Functional and Logic Programming. [s. l.]:[s. n.],1999.
- [6] 李阳. OCaml 语言中泛型编程工具的研究与实现[D]. 南京:解放军理工大学,2012.
- [7] de Rauglaudre D. Camlp5-Reference Manual[M]. [s. l.]:Institut National de Recherche en Informatique et Automatique,2008.
- [8] Holdermans S, Jeuring J, Loh A, et al. Generic views on data types[C]//Proc of International Conference on Mathematics of Program Construction. [s. l.]:[s. n.],2006.
- [9] 陈林. 泛型程序重构技术研究[D]. 南京:东南大学,2009.
- [10] 孙斌. 面向对象、泛型程序设计与类型约束检查[J]. 计算机学报,2004,27(11):13-16.
- [11] 陈林. 一种基于类型传播分析的泛型实例重构方法[J]. 软件学报,2009,20(10):2617-2627.
- [12] 陈林. 基于源代码静态分析的 C++0x 泛型概念抽取[J]. 计算机学报,2009,32(9):1792-1803.

# 基于输入域分类的实时软件可靠性模型


作者:

宋治平, 秦湘河, 朱怡安, SONG Zhi-ping, QIN Xiang-he, ZHU Yi-an

作者单位:

宋治平, SONG Zhi-ping(西北工业大学, 陕西西安710043; 西安卫星测控中心, 陕西西安710043), 秦湘河, QIN Xiang-he(西安卫星测控中心, 陕西西安, 710043), 朱怡安, ZHU Yi-an(西北工业大学, 陕西西安, 710043)

刊名:

计算机技术与发展 

英文刊名:

Computer Technology and Development

年, 卷(期):

2013, 23(7)

本文链接: [http://d.wanfangdata.com.cn/Periodical\\_wjz201307024.aspx](http://d.wanfangdata.com.cn/Periodical_wjz201307024.aspx)