

高可信软件的防危性评估研究

杨仕平, 熊光泽, 桑楠, 吴新勇

(电子科技大学 计算机科学与工程学院, 四川 成都 610054)

摘要: 在分析安全关键软件防危性测评的必要性基础上, 提出了适合于评估关键软件防危性的评估指标, 给出了防危性评估指标与可靠性评估指标之间的关系。总结了4种传统测评方法评估高防危性需求软件的局限性。研究了基于重要性采样及压力测试技术测评高防危性软件的可行性, 并详细讨论了其具体实施过程, 其间结合核电安全关键控制系统进行了例证。最后就该领域内的相关工作及发展方向进行了展望。

关键词: 安全关键; 评估; 防危性; 可靠性; 重要性采样; 压力测试

Research on safety evaluation of high dependable software

YANG Shi-ping, XIONG Guang-ze, SANG Nan, WU Xin-yong

(School of Computer Science and Engineering, UEST of China, Chengdu 610054, China)

Abstract: On the basis of analyzing the necessity of safety testing and evaluation for safety critical software, the safety evaluation criteria fitted for testing and evaluating safety critical software are proposed and the relations between safety evaluation criteria and reliability evaluation criteria are presented. Followed this, the limitations of the four classical testing and evaluation approaches used to evaluate the software with high safety requirements are summarized. The feasibility of safety testing and evaluation based on the technology of importance sampling and stress testing is researched, and the concrete implement process about this approach is discussed in detail, during the course of doing this, the safety critical control system of the nuclear power plant is used in a practical example in order to exemplify the correctness of the proposed approach. Finally, the related work and future trends of the research in this field are listed.

Key words: safety critical; evaluation; safety; reliability; importance sampling; stress testing

1 引言

随着美国亚利安娜V型火箭发射失败、俄罗斯核潜艇事故、美国9·11事件的相继出现, 关键系统的可信性问题日益受到全球的普遍关注。关键系统可进一步细分为安全关键系统(Safety Critical Systems, SCS)与事务关键系统, 其中SCS是指系统功能一旦失效将危及人的生命以及环境可能遭到严重破坏的系统, 而事务关键系统则是指系统功能一旦失效将导致财产重大损失的系统^[1]。本文关注的重点在于SCS, 这类系统广泛存在于航空航天、国防、交通运输、核电能源和医疗卫生等诸多领域中, SCS的可信性(Dependability)^[2]则是指系统在任务开始时且可用性给定的情况下, 在规定的时间内和环境内能够使用且能完成规定功能的能力, 即系统“动则成功”的能力。可

信性通常包含可用性、可靠性、可维护性、防危性(Safety, 防止危险发生)、安全性等特征, SCS所关注的可信性属性主要在于可用性、可靠性、可维护性、防危性这4个特征, 其中又以可靠性与防危性为重中之重。据统计资料表明, 随着软件技术在SCS中的大量使用, 软件的设计缺陷逐渐成为SCS发生灾难性事故的主要根源, 因而SCS的可信性问题重点在于其中所使用软件的可靠性与防危性, 同时称SCS中用于关键控制的软件为安全关键软件, 本文将主要讨论安全关键软件的防危性评估。

2 安全关键软件的防危性评估指标

在研究安全关键软件的防危性评估指标之前, 我们先总结一下与软件可靠性相关的评估指标。目前与软件可靠性相关的主要评估指标有平均无故障时间MTTF、平

基金项目: 国防科技预研基金资助项目(2000J6.7.1. DZ0206)。收稿日期: 2002-12-25; 修订日期: 2003-04-02

作者简介: 杨仕平(1974-), 男, 博士研究生, 主要研究方向为实时操作系统的防危核机制与实现、安全关键系统; 熊光泽, 男, 博士生导师, 教授, 主要研究方向为实时计算机系统及应用; 桑楠, 男, 副教授, 主要研究方向为实时软件技术; 吴新勇, 男, 博士研究生, 主要研究方向为安全实时操作系统。

均故障间隔时间 $MTBF$ 等。然而这些可靠性指标都不能很好地定量评估安全关键软件的防危性,其原因在于可靠性与防危性所关注的内容不同,可靠性主要关注软件系统能否在给定的时间与环境中持续工作,而防危性则主要关注软件系统能否防止关键系统发生重大灾难性事故。通常安全关键软件可处于正常运行、安全失效、非安全失效3种状态。由前可知,核电SCS中用于紧急停堆的关键控制软件的运行频率是非常低的,一般可能一年或几年才运行一次,这样当核电SCS无紧急情况出现(如反应堆的温度超过规定值)时,即便是用于紧急停堆的关键控制软件失效也不会造成重大事故发生,此时称该关键软件的失效为安全失效,但是当核电SCS有紧急情况出现时,此时紧急停堆关键控制软件如发生失效则称该软件失效为非安全失效。由前面的分析可知,可靠性评估指标 $MTTF$ 与 $MTBF$ 都不能很好地来评估软件的防危性,其原因是安全关键软件允许失效,只要其失效的时机是合适的,失效后的恢复是及时的,一般也不会导致系统发生灾难性事故。由此可知,只要安全关键软件不发生非安全失效,该关键软件系统仍具有较高的防危性。为了能很好地评估关键软件的防危性,应使用非安全失效的平均时间 $MTTUF$ (Mean Time To Unsafe Failure)作为防危性评估指标之一,同样也可以使用非安全失效的平均间隔时间 $MTBUF$ (Mean Time Between Unsafe Failure)作为另一个防危性评估指标。防危性评估指标与可靠性评估指标之间具有一定的对应关系,设 C 为关键软件失效后的修复率,防危性评估指标与可靠性评估指标之间有下面二式成立:

$$MTTUF = \frac{MTTF}{1-C} \quad (1) \quad MTBUF = \frac{MTBF}{1-C} \quad (2)$$

由上面的式(1)及式(2)可知,防危性评估指标与可靠性评估指标之间具有一定的对应关系,通常关键软件的防危性评估包含可靠性评估,这种包含关系是合理的同时也是非常必要的,如核电紧急停堆关键控制软件一旦处于运行(紧急停堆)状态,其软件的可靠性便对能否成功停堆起着至关重要的决定作用。根据式(1)与式(2)所表示的对应关系,我们便可以借鉴已有的可靠性评估方法来评估关键软件的防危性。

3 高可靠软件的传统评估方法

根据可靠性需求的不同,Butler&Finelli^[6]把软件系统分为3类:①高可靠性需求的软件系统:失效率 $<10^{-7}$ 失效/小时;②中等可靠性需求的软件系统:失效率在 $10^{-3} \sim 10^{-7}$ 失效/小时之间;③低可靠性需求的软件系统:失效率 $>10^{-3}$ 失效/小时。现在有许多模型可用于软件可靠性评估,如密度模型、可靠性增长模型、基于组成结构的测量模型及稀少事件模型等都可用于评估软件的可靠性。尽管有这些模型都可用于评估软件的可靠性,但它们的评估能力是不相同的。

3.1 基于故障密度模型的可靠性评估

基于故障密度的可靠性评估模型其基本假设是随着软件编码缺陷的增加,软件的可靠性将降低。故障密度模型的预测能力依赖于应用的特征、开发环境、重用程度及其它因素。故障密度通常以每千行源码(KSLOC)所包含的编码缺陷来计算,为度量软件的可靠性,再把故障密度转化为软件失效率,然而这种转换需要假设软件运行时故障的发生概率,由于故障的缺陷本质及所处位置的不同,其发生故障的概率变化范围很大,因而基于故障密度的可靠性评估其准确性是较差的。

3.2 基于可靠性增长模型的可靠性评估

为量化软件的可靠性,早在20世纪80年代便使用可靠性增长模型(Reliability Grow Models)来定量地评估软件的可靠性。可靠性增长模型使用连续测试期间所观测到的软件失效频率的减小趋势推测软件未来的失效可能性。目前已有许多可靠性增长模型被提出用来评估软件的可靠性,如Schneidewind模型、指数模型、Musa/Okumoto对数泊松模型、Littlewood/Verrill模型等。尽管可靠性增长模型经历了多年的发展,但它仍有许多不足的地方。首先是它只能评估低可靠性需求的软件系统(失效率在 $10^{-1} \sim 10^{-3}$ 失效/小时之间),对于高可靠性需求的软件系统,经实践证明使用该可靠性增长模型来评估软件的可靠性是不可行的^[6]。如对于失效率为 $10^{-7} \sim 10^{-9}$ 失效/小时的软件系统,当只有一个版本被测试时,为发现软件中的一个缺陷,将需要 $10^8 \sim 10^{10}$ 小时(11415年~1141550年)来测试可靠性。增长模型的另一个缺陷是不能根据软件的组成结构来构建软件的可靠性评估模型。通常,可靠性增长模型把整个软件作为一个不可见其内部组成结构的黑箱,而评估时也是把整个软件作为一个整体,所有的这些假定都不符合安全关键软件系统的实际组成情况。现实中的关键软件系统一般都使用容错机制以提高其可靠性,而容错机制一般包括错误检测与处理、冗余管理、备份任务等。据统计资料表明,由于有容错机制的存在,实时容错系统中80%~95%的任务级失效都是可恢复的,因而整个软件系统的可靠性不能简单地通过组件(任务)级中所观测到的失效来量化。由前面的分析可知,现有的可靠性增长模型确实不能很好地用来评估高可靠性需求的关键软件系统。

3.3 基于结构化模型的可靠性评估

可靠性增长模型的主要缺陷之一是不能根据软件系统的组成结构来构建可靠性评估模型,而基于结构化模型的可靠性评估能有效改善可靠性增长模型的局限性。基于结构化模型的可靠性评估最初被用于评估高可用的、连续不断运行的系统,然而关键系统不完全等同于这些系统,表现在:①既包含连续运行剖面又包含间歇性运行剖面,如对关键参数的监测便是连续不断的,而紧急情况时报警则是间歇性的;②紧急稀少事件的重要性^[6],如

容错系统中，主任务向备份任务转换便是稀少事件，但如不能成功转换则将可能导致灾难性事故发生。在基于结构化模型进行软件的可靠性评估时，应用软件、操作系统、其它支撑软件及硬件组件都被看成系统中的等价组成元素，这些元素的运行时间、失效

率、关联失效率、恢复时间及恢复率通常在可靠性测试时被收集，一旦得到这些数据，便可使用一定的模型进行可靠性评估。使用该方法评估软件可靠性可分两步来实现，首先是系统可靠性模型的建立，其次是使用测试或运行数据来确定模型中参数的具体值。基于结构化模型的可靠性评估可使用可靠性块图模型、K/n模型及马尔可夫链3种模型，其中可靠性块图模型与K/n模型都属于组合模型，且假定各组件模块的失效是独立的，而马尔可夫链则属于随机模型，它可以表示各组件模块之间的相互作用及失效关联。使用基于结构化模型的可靠性评估方法具体实现时可分为4步进行：①数据收集；②数据分类；③统计分析；④可靠性建模。这4步的具体内容可参考文献[8]。

该方法已经经过多年的研究，目前可用于评估失效率 10^{-3} 至 10^{-5} 失效/小时之间的系统，但该方法的有效性取决于系统在多大程度上满足如下假设：①系统的开发过程必须是高质量的，同时也必须保证软件验证与确认、测试过程中发现的再生性缺陷被完全排除；②用户的需求已完全被设计人员、开发人员及测试人员很好地理解；③以前的运行数据及测试数据能在很大程度上代表软件实际的运行环境，同时也对软件进行了充分的测试。

3.4 基于压力测试的稀少事件评估模型

由前可知，安全关键软件的运行频率是非常低的，如核电紧急停堆关键软件可能一年甚至几年才运行一次，其原因是触发该关键软件投入运行的紧急事件（如反应堆的温度超过规定值）是非常稀少的。而对关键软件的可靠性要求通常是非常高的（一般在 10^{-7} 至 10^{-9} 失效/小时之间），要使这类软件具有较高的可靠性，测试是必需的，如果使用传统的统计测试方法则需要花费很长的测试时间，因而缺乏一定的实用性。压力测试主要通过故障注入等技术使某些很少运行的关键操作加快运行频率，使其在有限的时间内暴露出更多的缺陷。压力测试的主要问题是测试用例不能很好地代表软件的实际运行情况，通常所测得的失效率大于软件的实际失效率。假设能计算出这种放大率，则测得的失效率可还原回实际的失效率。以上本文讨论了4种可靠性评估方法，各种方法各有优缺点。为便于比较，使用表1作为总结。

4 基于重要性采样的高可靠软件评估

在第3节中讨论了4种可靠性评估方法，但它们都不

表1 4种传统可靠性评估方法的比较

技术	生命周期阶段	典型的措施	优点	缺陷	预测能力
故障密度模型	所有阶段	故障/KSLOC	参考数据可用	必须假设发生率	低
可靠性增长模型	测试	失效/运行小时	某些参考数据可用	需要观察多个失效	中等
结构化模型	测试与操作	每个码段每小时的失效次数	可建模软件的结构	很少的参考数据	中等/高
基于压力测试的稀少事件评估模型	操作	失效/操作年	可应用于高完整性系统	无参考数据	较高

能用于评估安全关键软件的防危性，因为安全关键软件系统所要求的软件失效率不但非常小（一般应小于 10^{-7} 失效/小时），而且这类软件运行的频率是非常低的。另外，这类软件的可靠性对整个系统的可靠性起着至关重要的作用，如果使用传统的生命测试方法对其进行可靠性测评，则需要大量的实验样本，以便对可靠性参数进行估计。为减小实验样本数，Kahn^[9]早在1953年便提出重要性采样理论，它的主要思想是给关键样本分配较高的重要性，从而加大关键样本对估计结果的影响力度。重要性采样作为一种统计方法，近年来被广泛应用于蒙特卡罗（Monte Carlo）仿真，以评估计算机系统的可信性。它不但能使对样本的估计保持较高的准确性，而且也能减少样本数^[9-10]。由前可知，压力测试能加速软件的运行频率，以使关键软件能在有效的时间内暴露出更多的缺陷，从而方便关键软件的防危性评估。而重要性采样则主要在于加大关键样本的重要性，当重要性采样与压力测试相结合时将能够用于评估高可靠、高防危的软件系统，图1为二者用于评估高可靠、高防危软件的流程图。整个评估流程包括运行剖面的选择、重要性采样、压力测试、参数估计及可靠性评估等过程。

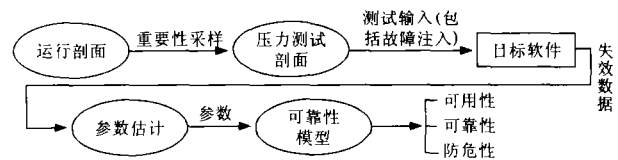


图1 基于重要性采样及压力测试的高可靠软件评估流程

4.1 运行剖面的选择

“剖面”是离散可选元素的集合，且每个组成元素都具有一定的发生概率^[9]，而软件的运行剖面是软件如何被使用的定量表示。安全关键软件的运行剖面可通过系统模式与关键操作来分类，即必须进行：①关键系统模式及发生概率的标识；②稀少运行的关键操作及其发生概率的标识。

为了标识关键系统模式，可把系统分为两大类：①连续运行的实时系统；②在线保护系统。这两类系统的运行剖面是大不相同的。对第1类系统，其输入是连续不断的（工作量可能波动），而对于第2类系统，其输入则是间歇性的（稀少事件）。第1类系统需要较高的可靠性，

可使用冗余措施容忍组件级的失效,操作系统、电话交换及空管系统都属此类。第2类通常需要系统能成功地响应紧急情况,因为一旦响应失败将可能会导致灾难性事故发生,核电保护系统便是此类系统的典型代表。与这两类系统相对应的关键模式为失效恢复模式和紧急处理模式。据最近的研究资料表明,美国 PSTN 网络的 70% 失效发生在恢复模式,这进一步确认了恢复模式的重要性。尽管这些关键模式也很少被运行,但稀少运行关键操作并不仅限于这些模式,还应包含冗余管理、异常处理、初始化与调整及数据边界条件或越界的处理。一旦关键模式与关键操作被标识后,紧接着要做的是估计这些操作发生的概率,这些操作的一部分(如初始化与校准)可由正常的工作量来确定,而另一部分(如冗余管理)则由故障到达率来确定。譬如,某过程控制程序每 100 小时初始化一次,同时假设初始化的时间为 5 分钟,则初始化操作发生的概率为 $5\text{min}/100\text{h}\approx 0.00083$ 。而对容错系统,假设要求进行主从设备之间转换的故障每 6 个月出现一次,且假设主从设备之间的转换需要 10 分钟,则冗余管理操作发生的概率为 $10\text{min}/6\text{months}\approx 0.000038$ 。尽管大多数时候是非常难于估计某关键操作发生的概率,但并不难于确定该关键操作发生概率的上界。设 P_c 为关键操作可能发生的概率, P_{\max} 为该关键操作发生概率的上界,则有 $P_c \leq P_{\max}$ 成立。

4.2 基于重要性采样及运行剖面的定量分析

运行剖面一般被分为关键集合(关键而很少执行的操作)及普通集合。假设 oc_1, oc_2, \dots, oc_n 为关键集合中的关键操作,这些关键操作发生的概率分别为 pc_1, pc_2, \dots, pc_n ; 而 or_1, or_2, \dots, or_m 为普通集合中的普通操作,这些普通操作发生的概率分别为 pr_1, pr_2, \dots, pr_m , 则有下式成立:

$$\sum_{i=1}^n pc_i + \sum_{j=1}^m pr_j = P_c + P_r = 1 \quad (3)$$

上式中的 $P_c = \sum pc_i$ 是关键集合中各操作发生的总概率,而 $P_r = \sum pr_j$ 是普通集合中各操作发生的总概率,且满足 $P_c \ll P_r$ 。

由压力测试的应用前提可知,在进行压力测试之前已对软件进行过大量的普通测试,由于关键集合中各操作发生的概率较小,因而已做的那些测试主要集中于普通集合上,而对于关键集合必须使用随机(或选择性)测试与故障注入等技术来加速关键操作的运行频率,以使其在有限的时间内暴露出更多的缺陷。在此假设某测试集合 O_s 中包含 k 个关键操作 $oc_{s1}, oc_{s2}, \dots, oc_{sk}$, 各关键操作的发生概率分别为 $pc_{s1}, pc_{s2}, \dots, pc_{sk}$, 设经过压力测试后各关键操作的发生概率分别增加到 $pc'_{s1}, pc'_{s2}, \dots, pc'_{sk}$, 同时假设 $\Lambda(O_s)$ 为放大比, 则有下式成立:

$$\Lambda(O_s) = \frac{pc'_{s1} + pc'_{s2} + \Lambda + pc'_{sk}}{pc_{s1} + pc_{s2} + \Lambda + pc_{sk}} \quad (4)$$

通常压力测试可分为两类:随机或选择性测试和故障或错误注入。对第1类压力测试应通过连续不断地对

关键集合采样来进行测试,而对第2类压力测试则可间歇进行,通常每次注入一个故障。对第1类压力测试,软件失效率的估计是非常关键的,设 λ'_c 为经过第1类压力测试后的软件失效估计,同时假设 λ_c 为关键集合在真实环境下的实际失效率,则有 $\lambda_c \Lambda(O_s) = \lambda'_c$ 成立。由于整个关键软件系统由关键集合与普通集合组成,设 λ_r 为普通集合的失效率,则整个系统的失效率 λ 为 $\lambda = \lambda_r + \lambda_c$, 由于有 $\lambda_r < \lambda_c$ 成立,则整个系统的失效率 λ 满足 $\lambda = \lambda_r + \lambda_c < 2\lambda_c$, 为保守起见,应使整个软件系统的失效率 λ 为 $2\lambda_c$, 由于有 $\lambda_c \Lambda(O_s) = \lambda'_c$, 所以有 $\lambda_{\max} = 2\lambda'_c / \Lambda(O_s)$ 成立。而对于第2类压力测试(故障/错误注入),重要的参数估计是故障恢复率 C 及防危系统响应紧急情况的成功率 p_e 。

(1) 基于故障注入的压力测试

故障注入是一种测试软件鲁棒性的有效手段,对容错软件特别有用,目前已有几种技术可用于注入故障或错误:①故障注入——修改程序的代码段;②错误注入——修改程序的数据段;③鲁棒性的基准测试。大多数情况下,注入所有可能的故障是不现实的,因为故障的类型是多种多样的,程序产生故障的方式也是非常多的,然而这些故障的表现形式却是有限的,几种典型的失效模式包括停止、悬挂、延迟输出和无效输出。使用故障注入进行压力测试时,首先标识出软件可能的失效模式,然后针对各失效模式设计出大量典型的故障,把故障注入到软件模块以诱导软件出现失效,一旦注入的故障被激活,便对软件模块进行监测,同时收集相关数据以便进行参数估计。

(2) 基于错误注入的压力测试

通常,导致软件失效的原因可分为3类:错误的输入数据、故障代码和前两者的组合(导致稀少事件发生)。错误注入主要是对第1类失效原因进行模拟,注入错误数据的一种方式是在更改系统的内部状态,这种方式可模仿硬件(如 CPU、内存及网络)原因导致的软件失效。注入错误数据的另一种方式是破坏输入数据,这种方式则主要模仿外部输入数据(如传感器数据)导致的软件失效。以上两种错误注入的实现方法与故障注入相同。

(3) 鲁棒性基准测试

鲁棒性基准测试是一种特殊的错误数据注入技术,它主要用于测试软件(操作系统或其它应用软件)怎样响应错误的输入,通常这些软件能容忍一般的错误输入或错误使用,进而避免系统崩溃。

5 实例研究

由图1可知,使用重要性采样及压力测试技术进行高可靠软件测评时,除了使用重要性采样外还应应对压力测试时的失效数据进行收集,然后再进行关键参数的估计,最后根据关键软件系统的组成结构建立可靠性评估模型,以便进行可靠性评估。由于篇幅有限,本文在此假

设已根据失效数据估计出关键参数,并根据失效参数的放大率 $\Lambda(O_s)$ 把估计值还原回实际的失效率。为了说明如何根据软件系统的组成结构建立模型进行可靠性评估,本文在此使用核电SCS作为实例进行研究。设核电SCS主要由发电控制系统及防危保护系统组成,则该SCS一般处于正常状态N,防危处理状态SP,安全失效状态SF,系统失效状态F,4个状态之间的转换如图2所示。

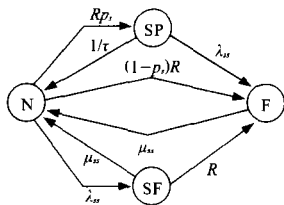


图2 核电安全关键系统的状态转换图

图2中的正常状态N表示安全保护系统与发电控制系统都处于正常状态;防危处理状态SP表示防危系统正在处理紧急情况;安全失效状态SF表示发电控制系统仍处于正常状态,但防危系统却不可用于响应紧急情况;系统失效状态F表示防危系统处理紧急情况失败后导致整个系统失效。而参数 p_s 表示防危系统处理紧急情况的成功率为 p_s ,参数 τ 表示紧急情况的平均处理时间,参数 R 表示紧急情况的到达率,参数 λ_m 表示防危系统的系统失效率,参数 μ_s 表示防危系统失效后的恢复率。

由图2可知,在正常状态N下,当紧急情况出现时,防危保护系统将以 p_s 大小的成功率来响应,同时系统从正常状态N转移到防危处理状态SP,其转移率为 Rp_s 。在防危处理期间,假设由于硬件或软件的原因而导致防危处理系统失效,则整个系统也将失效,系统将从防危处理状态SP转移到系统失效状态F,其转移率为 λ_m 。否则当防危处理系统成功处理紧急情况(平均紧急处理时间为 τ),则系统将从状态SP转移到正常状态N,其转移率为 $1/\tau$ 。当紧急情况在正常状态N出现而防危处理系统不能成功响应时,则系统将从状态N转移到失效状态F,其转移率为 $(1-p_s)R$ 。某些时候,在正常状态N,防危处理系统安全失效时,系统将从正常状态N转移到安全失效状态SF,其转移率为 λ_s 。如果在安全失效状态SF防危处理系统正在进行恢复,此时如果出现紧急情况,由于防危处理系统在该状态不可用,系统将从安全失效状态SF转移到系统失效状态F。参数 λ_m 是防危保护系统的失效率,多个同时发生的软硬件随机缺陷及单个相同缺陷都可能导致防危保护系统失效,然而,此参数的估计由防危保护子模型来实现。如图3所示,假设防危保护系统由主通道与备份通道两个通道组成。在正常状态N时,主通道与备份通道都可能失效,设通道的失效率为 λ ,假设主通道失效,系统将转移到备份通道,设其转移成功率为 C ,在如图3所示的模型中,系统将从状态N转移到状态1,其转移

率为 λC 。假设备份通道失效,系统也将从状态N转移到状态1,因为此时不需要从主通道转向从通道,故此时的转移率为 λ ,所以从状态N转移到状态1的(总)转移率为 $\lambda C + \lambda$,注意此失效率 $(\lambda C + \lambda)$ 表示单通道的失效率,并不表示多个通道的失效率。在状态1,假设另一个通道也失效,则整个防危保护系统也将失效,此时系统将转移到状态F,其转移率为 $1-C$,假设在状态1失效通道能成功恢复,设其恢复率为 μ ,系统将从状态1返回到正常状态N。防危保护系统失效后,将立即进行恢复,同时假设其恢复的成功率为 β_s ,系统将从状态F转移到状态N。

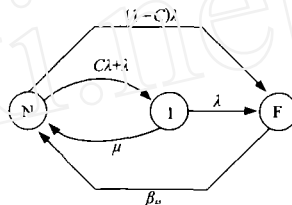


图3 防危保护系统的状态转换图

在上面所有参数中,防危系统处理紧急情况的成功率为 p_s 对整个系统的防危性能影响最大,设非安全失效的平均间隔时间MTBUF(Mean Time Between Unsafe Failure)即系统到达图2中状态F的平均时间为系统的防危性评估指标。图4为参数 p_s 对平均危险间隔时间的影响。

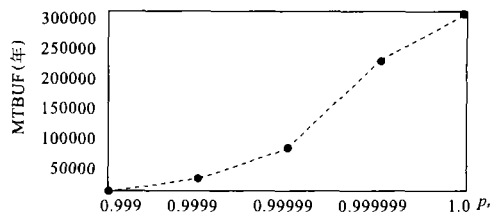


图4 防危系统处理紧急情况的成功率 p_s 对MTBUF的影响

由图4可知,为了提高整个系统的防危性,通过各种措施提高防危系统处理紧急情况的成功率为 p_s 是至关重要的。当然,图2中的参数 μ_s 也是非常重要的,它将由图3所示的防危保护子系统来确定。

6 结束语

随着计算机技术的广泛普及,在SCS的设计与运行过程中,嵌入其中的安全关键软件的防危性是一个需要着重考虑的关键因素。为此,许多发达国家均已制定了完善的、有利于提高安全关键软件防危性的第3方独立测试、评估和生产许可制度,甚至通过立法来管理安全关键软件的生产、验收和评估。对于安全关键软件,无论是在进行独立的第3方质量评估时还是在交付使用之前,都需要对软件进行防危性测评,以检验软件是否满足用户既定的防危性需求。高防危性需求的软件对可靠性的

(下转第169页)

步,如否则不满足唇同步。需要说明的是,图1中“极点态音频/视频知识库”由唇同步相关音频/视频规则组成,知识库通过事先训练生成。

7 结论

本文根据汉语发音特点将汉语对应口型划分为4类,进而区分为极点态和过渡态两种状态,并得出结论:汉语唇同步的验证也就是对极点态音频和极点态视频的同步验证。这一结论将汉语唇同步验证研究范围由对约400个无调音节的研究减少到对7个音素的研究,极大地缩小了研究规模。在此基础上提出基于极点态音频/视频知识库的唇同步识别与验证模型,并对模型中各部分进行分析,提出可以将基于运动对象识别的帧间差法与嘴唇形状、颜色和运动特征结合,实现嘴唇的精确定位,最后分析了唇同步验证过程。唇同步验证与识别技术在在线的闭环仿真、视频电话及视频会议等分布式多媒体系统中均可发挥了积极的作用。

参 考 文 献:

[1] Chen T,Rao R.Audio-visual interaction in multimedia communication[J].1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1997,(1):17-182.

- [2] Steinmetz R.Human perception of jitter and media synchronization[J].IEEE Journal on Selected Areas in Communications,1996,14(1):61-72.
- [3] Chen T,Graf H P,Wang K.Lip-synchronization using speech-assisted video processing[J].IEEE Signal Processing Letters, 1995,2(4):57-59.
- [4] Cosatto E,Potamianos G,Graf H P.Audio-visual unit selection for the synthesis of photo-realistic talking-heads [C]. 2000 IEEE International Conference on Multimedia and Expo,2000.619-622.
- [5] Melek Z, Akarun L. Automated lip synchronized speech driven facial animation[C].2000 IEEE International Conference on Multimedia and Expo,2000.623-626.
- [6] 胡光锐.语音处理与识别[M].上海:上海科学技术文献出版社,1994.
- [7] Yuille A L,Hallinan P W,Cohen D S.Feature extraction from faces using deformable templates[J].International Journal of Computer Vision,1992,8(2):99-111.
- [8] Gong S,McKenna S,Psarrou A.Dynamic vision:from images to face recognition[M].Imperial College Press,2000.
- [9] Wechsler H. Face recognition: from theory to applications [M].Berlin:Springer,1998.

(上接第165页)

要求也是非常高的,且防危性评估指标与可靠性评估指标之间存在一定的关系。传统的测评方法不能用于测评高可靠性需求的软件,本文所研究的基于重要性采样及压力测试的测评方法能很好地评估高防危性需求的软件。鉴于目前我国军方正在大力提倡对关键软件进行独立的第3方测试,这时由于不能很好地得知关键软件的内部组成结构,本文所研究的方法可能受到一定的限制,因而作为下一步的研究工作,我们将探索新的第3方关键软件测试方法。关于安全关键软件的测评还有许多工作要做,本文只是在该领域内做了一点有益的探讨,希望有更多的人能投身于此行列的研究。

参 考 文 献:

[1] 杨仕平,熊光泽,桑楠.安全关键系统高可信保障技术的研究[J].计算机科学,2003,(5).

[2] Laprie J C.Dependability: Basic concepts and terminology [M]. Vienna: Springer-Verlag, 1991.

[3] Barroca L,McDermid J. Formal methods: use and relevance for the development of safety-critical systems[J]. Computer Journal, 1992, 35(6): 579-599.

[4] Farr W H, Smith O. Statistical modeling and estimation func-

tions for software (SMERFS) [M]. Users Guide. NSWCCD TR84-371, Revision 3, 1993.

- [5] Herbert Hech, Patrick Crane. Rare conditions and their effect on software failures[J]. Proceedings of the 1994 Reliability and Maintainability Symposium, 1994,(1):334-337.
- [6] Butler R W,Finelli G B. The infeasibility of quantifying the reliability of life-critical real-time software[J]. IEEE Transactions on Software Engineering, 1993, 19(1):3-12.
- [7] Troubitsyna E. Reliability assessment through probabilistic refinement[J]. Nordic Journal of Computing, 1999.
- [8] Dong Tang, Myron Hecht, Xuegao An.MEADep and its application in dependability analysis for a nuclear power plant safety system [J]. IEEE Transaction on Nuclear Science, 1996, 25(3):1014-1021.
- [9] Kahn H, Warshall A W. Methods of reducing sample in monte carlo computations[J]. Journal of the Operations Research Society of America, 1953,(5):263-278.
- [10] Myron Hecht, Herbert Hecht. Use of importance sampling and related techniques to measure very high reliability software [J]. Aerospace Conference Proceedings, IEEE, 2000, (4):533-546.