文章编号 1004-924X(2010)02-0485-06

基于体系结构的软件可靠性评估

魏 颖[1,2,3],张 波[2,3],李 丽[2,3],沈湘衡[2],陈 媛[2,3],张格非[4]
(1. 北华大学 计算机学院,吉林 吉林 132000;
2. 中国科学院 长春光学精密机械与物理研究所,吉林 长春 130033;3. 中国科学院 研究生院,北京 100390;
4. 中国科学院 研究生院 信息安全国家重点实验室,北京 100049)

摘要:对软件可靠性评估的重要工具之一——基于体系结构的可靠性模型进行了实验验证。为了准确地分析软件模块间的调用关系并估算模块的可靠度,首先,依据评估方式的不同将基于体系结构的软件可靠性模型划分为合成型与分级型两类;然后,对软件体系结构的确定与软件模块的划分进行了分析,并阐述了模块可靠度和模块间转移概率的估算方法与步骤;最后,对某地面目标模拟源主控系统软件进行了实例分析。结果表明,实验系统的确定可靠度值为 0.938,而合成型、分级型模型估计值分别为 0.972 和 0.969,由此验证了两种模型的可应用性,为工程应用中实施基于体系结构的软件可靠性评估提供了参考。

关 键 词:软件可靠性;基于体系结构的模型;软件可靠性评估
中图分类号:TP311.5 **文献标识码**:A

Architecture-based software reliability evaluation

WEI Ying[1,2,3],ZHANG Bo[2,3],LI Li[2,3],SHEN Xiang-heng[2],CHEN Yuan[2,3],ZHANG Ge-fei[4]

(1. *Computer Institute of Beihua University*, *Jilin* 132013, *China*; 2. *Changchun Institute of Optics*, *Fine Mechanics and Physics*, *Chinese Academy of Sciences*, *Changchun* 130033, *China*;
3. *Graduate University of Chinese Academy of Sciences*, *Beijing* 100049, *China*;
4. *State Key Laboratory of Information Security*, *Chinese Academy of Science*, *Beijing* 100049, *China*)

Abstract: Architecture-based reliability models for the software reliability evaluation are tested and verified in this paper. To analyze the failure behavior and the relationship between software modules precisely, the architecture-based models are classified as a composite type and a hierarchy type according to the different evaluation methods. Then, how to determine the software architecture and to divide the software models is disccussed and the estimation methods and processes for software reliability and transferring probalilities between the softwares are given. A control software for a ground target is analyzed and verified, results indicate that the assessing value of composite and hierarchical models are 0.972 and 0.969, respectively, which is closed to the actual reliability of 0.938. The method can provide a references for the implementation of architecture-based software reliability evaluation.

Key words: software reliability; architecture-based model; software reliability evaluation

收稿日期:2008-08-26;**修订日期**:2008-11-10.
基金项目:国家 863 高技术研究发展计划资助项目(No. 2007AA703112)

1 引　言

软件可靠性是软件质量的重要属性之一。近年来，软件可靠性的评估模型层出不穷，它们大致可分为黑盒模型与白盒模型两类。黑盒模型是过去一些年中被广泛使用的预测软件可靠性的方法，该模型将软件看作一个整体，在分析软件可靠性时只考虑软件与外部环境的交互，而不考虑软件本身的体系结构。这一类模型大多只依据失效数据进行可靠性预测，并实施于软件生命周期的后期，忽略了软件测试数据与模块的可靠性。然而，随着面向对象技术的成熟与软件的规模化和组件化发展，软件可靠性的预测与评估也不得不考虑模块本身的可靠性和软件内部的体系结构。白盒模型又称结构模型，该模型将软件结构的动态信息与模块的失效行为相结合来进行软件的可靠性评估。与黑盒模型相比，这类基于结构的可靠性模型的优点是：定量分析软件与模块间的依赖程度，确定软件中的关键模块和接口，从而通过准确估算的模块可靠度以及模块间的连接关系评估软件系统的可靠性。由于软件体系结构在软件生命周期的初期就可以确定，因此这类模型可以贯穿整个研制周期进行软件可靠性的预测及评估[1]。第一个结构模型是 Littlewood 在 1979 年 IEEE 可靠性分会上提出的，发展到现在已经有几十种之多，并且部分模型在国外的软件可靠性工程领域已经得到了验证与应用[2-3]。

为了进一步验证基于体系结构的可靠性模型的特点，本文根据软件评估方式的不同将可靠性模型划分为合成型和分级型两类，基于两个代表型模型重点分析了基于体系结构的软件可靠性的评估过程和步骤，并给出了实验验证结果。

2 基于体系结构的可靠性模型

2.1 软件结构的 Markov 特性

通过对软件模块的划分，系统的可靠性特征需要借助模块的可靠性来表示。一般来说运行中的模块存在正常与失效两种状态，由于出现失效的随机性，因此可以将模块的运行看作一个状态随时间变化的随机过程。若后续模块的运行只取决于当前运行的模块而与之前运行的模块无关，即具有无后效性，并且若模块间的转移时间间隔为离散的，转移概率符合离散分布，则该模块运行符合离散时间 Markov 过程(DTMC)，这是大部分基于体系结构的可靠性模型假设的前提条件。一般来说，离散时间 Markov 模型由以下几部分构成：

(1) 包含 n 个状态的有限集合。

(2) 非负的 $n\times n$ 随机矩阵 $\boldsymbol{T}=(P_{ij})$，P_{ij} 表示系统从状态 s_i 转移到状态 s_j 的转移概率，其中 $P_{ij}\geqslant 0$，$\sum_{j=1}^{n}P_{ij}=1$，$1\leqslant i,j\leqslant n$。

(3) 向量 $\boldsymbol{\pi}=(\pi_1,\cdots,\pi_n)$，$\pi_i$ 表示状态 s_i 是系统初始状态的概率，$i=1,\cdots,n$。

2.2 软件可靠性模型

依据不同的实施方法，基于体系结构的软件可靠性模型被分为合成法和分级法两类。合成法是将软件体系结构与失效行为结合在软件可靠性模型中，进行软件可靠性预测。分级法是指首先建立软件结构模型，然后将软件失效行为添加到结构模型中，从而实现软件可靠性的评测。在目前存在的众多的可靠性结构模型中，我们选择出两个代表模型加以介绍。

2.2.1 合成法可靠性模型

Cheung 模型是最早考虑到将模块的可靠度应用到软件可靠性预测的模型之一[4]。该模型假设软件流程图有一个入口节点与一个出口节点。软件模块间的调用关系具有 DTMC 特性并表示为转换概率矩阵 $\boldsymbol{P}=[p_{ij}]$，其中 p_{ij} 表示程序从模块 i 转移到模块 j 的概率。模块间的失效行为是相互独立的。R_i 表示模块 i 的可靠度，即模块 i 成功运行的概率，也就是指模块 i 成功运行后的输出值是正确的并转移到下一模块的概率。

该模型在软件结构中增加两个吸收状态 C 和 F，分别表示程序运行的成功状态和失败状态。模块成功过渡的状态转移矩阵定义为：$\boldsymbol{M}[i,j]=R_ip_{ij}$，表示状态 i 运行结果正确并成功转移到状态 j 的概率。在这个模型中，R_n 表示结束状态 s_n 到达状态的概率。任意状态 s_i 若失效，则到达 F 状态的概率是 $1-R_i$。

系统可靠性表示从初始状态 s_1 成功运行到结束状态 s_n 的所有可能性。即计算公式为 $R=Q(1,n)R_n$。其中：

$$Q(1,n)=\sum_{k=0}^{\infty}M^n(1,n)=(-1)^{n+1}\frac{|(\boldsymbol{I}-\boldsymbol{M})_{n,1}|}{|\boldsymbol{I}-\boldsymbol{M}|}.$$

$\boldsymbol{I}$ 是 k 行 k 列单位阵；$|(\boldsymbol{I}-\boldsymbol{M})_{n,1}|$ 是矩阵 $(\boldsymbol{I}-\boldsymbol{M})$ 除去第 n 行与第 1 列的简化矩阵的行列式。因此系统可靠性为：

$$R=(-1)^{n+1}\frac{|(\boldsymbol{I}-\boldsymbol{M})_{n,1}|}{|\boldsymbol{I}-\boldsymbol{M}|}R_n, \qquad (1)$$

该模型可以被用来评估不同体系结构的软件可靠性，例如：批处理/通道，调用/返回，并行/管道过滤和容错等结构。另外，该模型已经集成在净室可靠性管理系统中，该系统专门用来规划并验证基于模块的软件系统可靠性。

2.2.2　分级法可靠性模型

Gokhale 等人提出了一种分级法可靠性模型，并已经在应用中的到了验证[2]。模型假设软件结构符合 DTMC 特性。该模型提出模块的可靠度由与时间相关的模块失效率 $\lambda_i(t)$ 求出。若每个模块的失效行为满足改进的非齐次泊松分布过程。模块 i 的可靠度 R_i 是由失效率 $\lambda_i(t)$ 与软件每次运行时模块累计运行时间 V_it_i 求出，即：

$$R_i=\mathrm{e}^{-\int_0^{V_it_i}\lambda_i(t)\mathrm{d}t}.$$

其中 V_i 表示模块 i 的访问次数。t_i 表示模块 i 每次运行的平均时间。V_i 由以下的线性公式给出：

$$V_i=q_i+\sum_{j=1}^{n}V_jp_{ji}.$$

q_i 表示模块 i 是初始模块的概率。p_{ji} 是模块 j 到模块 i 的转移概率。由于该模型是分级法构造形式。软件结构模型是提前确定的，该模型假设模块间的调用是顺序执行的，这与硬件可靠性分析的串行结构相似。软件可靠性被定义为：

$$R=\prod_{i=1}^{n}R_i^{V_i}. \qquad (2)$$

3　软件可靠性评估

3.1　软件可靠性评估的步骤

基于体系结构的软件可靠性模型评估的一般步骤为：

(1) 确定软件结构，分解并划分系统功能模块；

(2) 定义模块的失效行为，估计模块的可靠度及模块间的状态转移概率；

(3) 结合软件可靠性模型与软件的体系结构对系统可靠性进行评估。

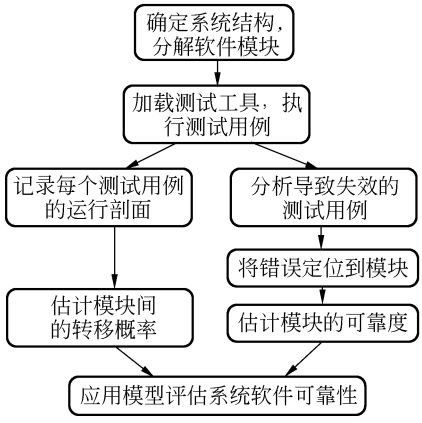


图 1　软件可靠性模型评估的步骤

Fig. 1　Estimating steps of software reliability model

为了获得稳定而有效的测试数据，通常在软件开发周期的系统测试阶段进行可靠性评估。图 1 细化了软件可靠性评估的具体步骤，由于基于体系结构的可靠性分析是需要动态执行软件的，因此在确定软件结构并分解系统模块之后需要执行测试用例，左分支通过测试工具的记录与计算，估计出模块间的控制转移概率；右分支表示分析导致系统失效的测试用例的执行情况，将错误定位到具体失效模块，累计模块的失效数，从而估计模块的可靠度。最后，应用可靠性分析模型，预测软件可靠性。在估计转移概率和模块可靠度过程中，测试工具必不可少，但是只有测试工具还不够，还需要建立模块与可靠性操作剖面之间的关系数据库，用来记录和统计软件测试过程中模块的转移调用次数。同时，要借助软件变更日志、软件测试日志等文件才能将软件错误与失效相对应，并最终将失效定位到相关模块[5]。

3.2　软件结构的确定与软件模块划分规则

软件结构反映了软件各模块运行的相互关系。软件结构可以在软件研制过程中的早期即软件需求分析以及软件设计时期被确定。软件可靠性测试人员可以通过查看文档或与软件设计师的沟通来掌握软件采取的体系结构风格[6]。

软件模块在具体应用中的划分是需要权衡的。模块划分得过细过多而产生的庞大状态空间，会导致软件可靠性模型参数很难确定从而增加了评估困难。另外，模块划分的过粗过少也会

造成由于忽略模块的分布差异而使失效数据丢失，从而影响可靠性评估的准确性。因此，软件模块的分解与划分依赖于模块数量、软件复杂度和模块使用信息的权衡。

3.3 模块可靠度

基于体系结构的软件可靠性评估很重要的一个步骤就是确定软件模块的可靠性。模块的可靠度依赖模块代码的特性和模块本身的运行方式。准确的估计模块可靠度的因素有三方面：模块代码是否可用；模块被测试的方法；模块本身是重用的还是新研制的。在基于体系结构的可靠性模型中通常假设模块失效是独立的，并且模块失效会直接导致系统失效。模块 i 的可靠度表示为模块成功执行规定功能的概率即：R_i。从直观角度来看，模块的可靠度可以通过模块的失效率求出，即：$R_i=1-f_i/n_i$，其中 f_i 是模块 i 运行的失效次数，n_i 为总的运行次数。这两个参数都可以参照测试工具记录的测试日志而计算出来。另外，模块的失效密度与模块的测试覆盖率有着紧密的联系，测试覆盖率很好的反映了模块的访问情况。因此，我们可以用基于时间的测试覆盖率 $c(t)$ 来估测模块的失效密度 $\lambda(t)$，测试覆盖率 $c(t)$ 被定义为：到时刻 t 为止，被激活的潜伏故障点占软件中全部故障点的比例。其中潜伏故障点是指程序中的某种结构或功能单元，与软件测试中使用的测试准则相关。该方法的通用框架形式如下：

$$\lambda(t)=ac'(t)\ , \tag{3}$$

其中：a 表示模块的初始错误数，$c(t)$ 是相继执行软件测试用例后得出的模块的测试覆盖率。$c'(t)$ 是 $c(t)$ 的一阶导数。在给定失效密度与模块的总运行时间 γ_i 的前提下，模块 i 的近似可靠度为：

$$R_i \approx \mathrm{e}^{-\int_0^{\gamma_i}\lambda_i(\theta)\mathrm{d}\theta}, \tag{4}$$

依据公式(3)(4)最终可得模块可靠度：

$$R_i = \mathrm{e}^{-\int_0^{\gamma_i}a_i c'_j(\theta)\mathrm{d}\theta} = \mathrm{e}^{-a_i c_i(\gamma_i)}\ .$$

3.4 模块间的转移概率

在基于体系结构的分析方法中，模块间转移概率是重要的因素。转移概率具体就是指软件模块间控制转换发生的可能性，在具有良好设计的软件系统中，模块间的控制转换是有限的。在软件设计阶段，也就是在软件开发和集成之前，转移概率可以通过分析程序结构和使用已知的操作剖面而得到。在代码集成测试阶段，通过分析软件测试产生的测试数据，进一步验证并确定预测值。而在实际应用过程中，可以通过运行全部的测试用例后估算各模块间的访问次数求出转移概率即：$p_{ij}=n_{ij}/n_i$ 其中 n_{ij} 表示模块 i 到模块 j 的访问次数，$n_i=\sum_j n_{ij}$ 表示模块 i 可达的所有模块的访问次数。

3.5 实例验证

某地面目标模拟源主控系统软件的功能是完成主控计算机对热像仪与温度控制箱的远程控制，并将热像仪的图像显示数据与温度测试数据返回给主控计算机，并存入数据库。软件的编程环境是 VC++，采用的是结构化模块设计。本文选取该软件的红外图像数据采集与处理这一功能模块进行可靠性分析。该功能模块的控制流程如图 2 所示：

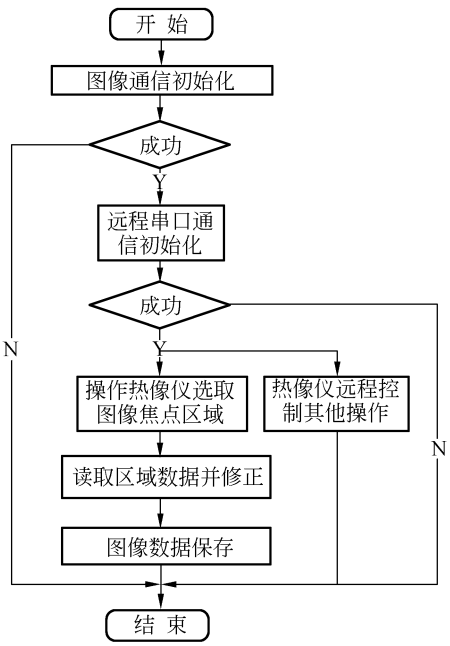


图 2　图像采集与处理功能结构控制流程图

Fig. 2　Control flowchart of picture collection and processing function module

该功能有 6 子模块构成，每个子模块能否被执行，直接由其前驱模块能否被成功执行所决定，因此具有无后效性。模块的控制转移过程可以转化为 Markov 过程链如图 3 所示。

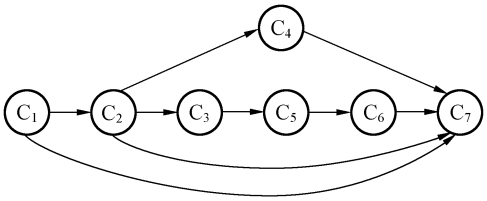


图 3　模块转移的 Markov 过程链

Fig. 3　Markov process chain of module transition

该功能模块在软件研制阶段的后期进行了功能测试，对测试过程建立了跟踪日志，跟踪日志中所记录的信息有：第 i 个测试用例，各子模块的访问次数，是否发生失效，发生失效的系统时间 t_i，导致失效的错误原因。对每个测试用例运行时所经过的子模块访问路径、子模块的运行次数以及引起模块失效的错误数进行了统计。其中各子模块的访问次数的测量方法是利用软件测试中的插桩技术，在每个模块的出口设置若干个转移计数变量，例如：模块 A 转移到模块 B 的计数变量为 sum_atob；模块 B 转移到模块 C 的计数变量为 sum_btoc。统计结果为：总的测试用例个数 $n=97$，发生错误总数 $f=6$。其中通信初始化模块错误数为 2，串口通信初始化模块错误数为 2，热像仪选取焦点区域操作模块错误数为 1，图像数据保存模块错误数为 1。因此，通过转换公式 $R_i=1-f_i/n_i$ 得出每个子模块的可靠度如表 1 所示。

表 1　子模块可靠度

Tab. 1　Reliability of sub-modules

R_1	R_2	R_3	R_4	R_5	R_6	R_7
0.981	0.981	0.998	1.000	1.000	0.990	1.000

通过跟踪日志对运行子模块时的访问次数的记录，利用公式 $p_{ij}=n_{ij}/n_i$，求出子模块间的状态转移概率如表 2 所示。

表 2　子模块间的转移概率

Tab. 2　Transition probability between sub-modules

$P_{12}=0.98$	$p_{17}=0.02$	
$p_{23}=0.78$	$p_{24}=0.20$	$p_{27}=0.02$
$p_{35}=1$		
$p_{47}=1$		
$p_{56}=1$		
$p_{67}=1$		

依据公式 $R=1-f/n$ 计算系统的确实可靠度，求得该功能模块的确实可靠度为 0.938。

应用 2.2 节中的合成模型和分级模型分别进行估算。结果比较如表 3 所示。

表 3　可靠度估算结果值

Tab. 3　Estimation results of the reliability

确实可靠度	0.938
合成模型估计值	0.972
分级模型估计值	0.969

4　结　论

本文从基于体系结构的软件可靠性模型入手，依据不同的实施方法，将可靠性模型划分为合成型和分级型，并分别列举了两个代表性的模型。重点分析了基于体系结构的软件可靠性评估的过程与步骤，并对评估过程中的三个要素的估算与确定方法进行了研究。最终经过实例验证得出：当软件结构符合 Markov 特性时，利用结构模型分析的可靠性预测结果与实际可靠度相似。相似的原因来源于验证实例的规模较小，软件测试结果充分而准确，因此模块可靠度和模块间转移概率的估计值相对精确。而对大型软件进行可靠性评估时，会面临模块结构划分比较复杂，引起失效的错误不唯一等问题，这都会使模型评估的结果出现偏差，这些问题都需要在工程应用中做进一步探讨。

参考文献：

[1]　KATERINA G P, TRIVEDI K S. Architecture-based approaches to software reliability prediction [J]. *Computers and Mathematics with Applications*, 2003, 46: 1023-1036.

[2] GOKHALE S S, WONG W E, HORGAN J R, *et al*.. *An analytical approach to architecture-based software performance and reliability prediction*[J]. *Performance Evaluation*, 2004, 58: 391-412.

[3] KATERINA G P, HAMILL M, PERUGUPALLI R. Large empirical case study of architecture-based software reliability[C]. *Proceedings of the 16th IEEE International Symposium on Software Reliability Engineering*, 2005: 43-53.

[4] LYU M R. *Handbook of Software Reliability Engineering*[M]. IEEE Computer Society Press and McGraw-Hill Book Company, 1996.

[5] WANG W L, PAN D, CHEN M H. Architecture-based software reliability modeling[J]. *The Journal of Systems and Software*, 2006, 79: 132-146.

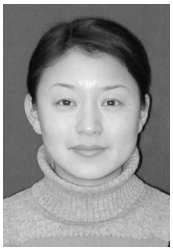
[6] GOSEVA-POPSTOJANOVA K, MATHUR A P, TRIVEDI K S. Comparison of architecture-based software reliability models[C]. *Proceedings of the International Symposium on Software Reliability Engineering*, 2001: 22-31.

[7] 李洁. 计算机控制系统的可靠性分析[J]. 光学 精密工程, 2000, 8(6): 584-587.
LI J. Analysis of computer control system's reliability[J]. *Opt. Precision Eng.*, 2000, 8(6): 584-587. (in Chinese)

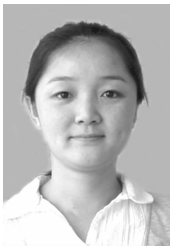
[8] CHIU K C, HUANG Y S, LEE T Z. A study of software reliability growth from the perspective of learning effects[J]. *Reliability Engineering & System Safety*, 2008, 93(10): 1410-1421.

作者简介：

魏　颖(1979－)，女，吉林人，博士研究生，研究方向为软件可靠性评估技术与方法等。E-mail：weiyingjl@beihua.edu.cn

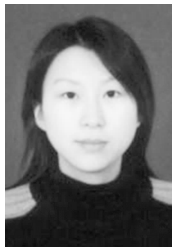


张　波(1974－)，女，吉林人，助理研究员，研究方向为嵌入式软件测试等。E-mail：zhanghui_net@tom.com



李　丽(1982－)，女，内蒙古呼伦贝尔人，博士研究生，研究方向为软件测试用例生成等。E-mail：xf1590@sina.com

沈湘衡(1952－)，男，湖南衡阳人，研究员，研究方向为电子学检测与系统质量评估技术。E-mail：shenxiangheng@yahoo.com.cn



陈　媛(1981－)，女，吉林长春人，博士研究生，研究方向为基于数据挖掘的软件测试等。E-mail：chris_chen226@yahoo.com.cn

张格非(1985－)，男，吉林长春人，硕士研究生，研究方向为面向对象的软件度量和软件质量控制等。E-mail：entrerilll@sohu.com