一种改进的基于架构的软件可靠性模型

陈俊文,谷建华,张 凡,董云卫
CHEN Jun-wen,GU Jian-hua,ZHANG Fan,DONG Yun-wei

西北工业大学 计算机学院,西安 710072
School of Computer Science,Northwestern Polytechnical University,Xi'an 710072,China
E-mail:cchenjw@yahoo.com.cn

CHEN Jun-wen,GU Jian-hua,ZHANG Fan,et al.Improved architecture-based software reliability model.Computer Engineering and Applications,2009,45(33):60-63.

Abstract: Architecture-based software reliability analysis typically takes the reliability of component as an invariable property of component itself,while disregards the fact that it is changed when the component context that it interacts with is changed due to a different operational profile of the system.This paper aims at architecture-based software reliability model which is one type of approach modeling component-based software,introduces an input profile matrix of transition destination component to build a relationship between operational profile and the component reliability,and provides an improved composite algorithm to solve the model.The application analysis of a case study indicates that this model is able to fully capture the impact of different operational profiles on the system reliability in aspects of both model parameters,transition probability and component reliability.This ability will benefit reliability analysis of higher accuracy at the stage of system design.

Key words: component-based software;software reliability;architecture-based model;operational profile

摘 要:基于架构的软件可靠性分析往往把构件的可靠性当作自身固有不变的属性,忽略了在不同的输入剖面下,因构件所处的交互环境不同造成的实际可靠性的变化。改进了一种基于架构的可靠性模型,引入转移目的构件剖面矩阵来建立系统操作剖面和构件可靠性的联系,并给出了改进后的可靠性合成算法。实例分析表明,该模型可以全面捕捉到在不同系统操作剖面下,因构件之间转移概率和构件可靠度这两个参数的变化对整体可靠性产生的影响,提高了系统设计阶段可靠性分析的精确性。

关键词:构件软件;软件可靠性;基于架构的模型;操作剖面
DOI:10.3778/j.issn.1002-8331.2009.33.019 **文章编号:**1002-8331(2009)33-0060-04 **文献标识码:**A **中图分类号:**TP311

随着大型软件系统中构件化技术的成熟和普及,基于构件的软件可靠性模型的研究逐渐成为可靠性研究的一个重要方向。现有的模型着重于通过挖掘基于构件的软件系统的架构特征进行数学建模,而对失效行为的建模和模型的应用方面的研究并不多。基于构件的软件系统包含基于剖面和基于架构两类可靠性模型,着重分析基于架构的模型,并具体针对其中基于路径的模型进行改进,以便于软件开发早期的可靠性分析。

1 软件可靠性模型

传统意义上有黑盒和白盒两种方式对软件可靠性进行度量。过去的很多研究集中在黑盒方式的可靠性模型上,即将软件看成一个单一的整体,而不考虑软件的内部结构对于整体可靠性的影响,主要应用在测试阶段的可靠性增长过程。而白盒方式的模型主要是从软件结构的角度考察系统的失效行为,在获知系统内部各个构件的可靠性基础上,通过分析系统运行时内部构件之间的动态交互过程,采用组合的方式计算出整体可靠性。这种方法可以运用在软件的设计阶段,也可以运用在软件的使用阶段。软件一般是基于分治(divide-and-conquer)思想进行开发的,而白盒方式的建模正是遵循了这种分解、分析的思想,在实践中有着更为广泛的适用性。

基于构件的软件通常把系统看作是一系列构件的组合,构件的基本特征是独立的、可替换的、在系统中满足一定功能的单元,此外还要求定义完整的接口来与其他构件通讯,系统本身也要求有良好定义的结构[1]。基于构件的可靠性模型可以充分利用构件现有的可靠性信息,采用组合的方法计算出系统的可靠性,化解了建立在统计学基础上的黑盒方式评估软件可靠性所面临的可行性障碍[2],是当前软件可靠性模型研究的焦点。根据现有文献可以将其分为两类:基于剖面的模型和基于架构的模型。

1.1 基于剖面的模型

Hamlet对构件的可靠性鉴定提出了一套比较完整的理论[3-4],实现了构件的可靠性的重用。模型的前提是用基于输入剖面的

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.60736017)。
作者简介:陈俊文(1984-),男,硕士生,主研方向:嵌入式系统,软件工程;谷建华(1965-),教授,博士;张凡(1979-),男,讲师,博士;董云卫(1968-),教授,博士。
收稿日期:2008-09-19 **修回日期:**2008-11-28

可靠度目录(catalogue)描述构件的可靠性，其核心思想是用从输入剖面到输出剖面的映射来描述构件，并用构件剖面之间的映射来描述胶合逻辑，通过剖面映射推导出各个构件的输入剖面，然而根据可靠度目录得出所有构件的可靠度，它们的乘积即系统的可靠度，这种方法本质上是一种理论上的程序分析，要求软件的输入输出可以在数学上良好的定义，针对的是理想的程序而不是现实的软件系统，把构件的失效概率当作一种依赖于输入子域的可组合的(compositional)属性来使用，精细地刻画系统中失效概率的组合过程。

1.2　基于架构的模型

基于架构的可靠性模型受到研究者的广泛关注。这类模型以系统的动态架构即运行时构件之间的动态转移关系为核心进行描述，将失效行为作为一种构件的属性或者架构上的属性合并进来，对系统的可靠性进行分析及评估。Goseva-Popstojanova 对现有基于架构的模型进行了全面的归纳和总结[5]，遍及算法上、实验上、模拟上和实践中的各种模型，并从理论上考察了各种模型的联系。总的来说，这些模型均要考虑系统分解、架构描述、失效行为描述及将失效结合到架构中四个方面的内容。根据将失效结合到架构中的方式的不同，可以将现有的基于架构的可靠性模型分为三类：基于状态的、基于路径的和可加模型。

(1)基于状态的模型。使用状态迁移图来描述软件的架构，将控制流的转移过程映射到系统状态空间，基于运行时调用关系的统计数据或者设计阶段的场景分析建立概率化的控制流图。模型假设迁移具有一阶马尔可夫属性，并且失效发生在每个状态的概率是相互独立的。常使用离散时间马尔可夫链(DTMC)、连续时间马尔可夫链(CTMC)、半马尔可夫过程(SMP)等随机过程进行建模。

(2)基于路径的模型。使用执行路径的集合来描述软件的架构，构件执行路径的组合实际就是用枚举的方式直观表达软件的运行时结构。枚举路径可以通过实验的手段得到，也可以通过分析规格说明或者场景模拟的方式得到。假设构件在执行路径的各个时刻都有可能发生失效，并假设发生概率相互独立，每条路径的可靠性可以用路径上所有构件可靠性的乘积来表示，而系统整体的可靠性可以平均各条路径的可靠性得到，计算时的权重由系统的操作剖面中各路径的概率决定。

(3)可加模型。强调综合各个构件的失效数据进行估计，而不是显式地对构件之间的转移关系进行描述。假设构件之间是简单的顺序关系单一路径模型，构件的失效行为具有相同的特征如可以用非齐次泊松过程(NHPP)描述，从而可以通过累加各个构件的失效强度(failure intensity)得到系统的失效强度。

1.3　模型的讨论

基于剖面的模型的特色是提出了基于子域的可靠性概念，从数据域而不是时间域的角度刻画构件的可靠性，使得用从输入到输出的映射来刻画出系统的动态结构成为可能。这种模型实际上是假设系统是一系列构件串行执行的序列，而且要求构件能用良好的映射形式来定义。它的一个根本缺陷在于，不可预期的失效会破坏预期的输入输出函数关系，当构件出错时，不管输入子域是如何划分的，错误的输出可能出现在任何一个输出子域上，剖面映射前后的子域并不能保持一致的失效概率，使得在剖面映射得到的新剖面上对失效概率进行组合的做法失去了意义。

而基于架构的两种主要模型的特色是将系统的控制流图概率化，从而可以直接适用于现实中的大多数软件。基于状态的模型虽然能够在数学上得到全面、精确的分析解[6]，但它的代价是用大量相互独立的状态来刻画系统内部结构，对于大规模系统存在状态爆炸的问题[7]；而基于路径的模型本质上是算法性的，只能得到数值上的解，但是这种算法具有可伸缩性，适合于对大规模系统进行求解[8]。实际上，基于路径的模型在计算每条路径的可靠性时其实并不局限于在假设范围内计算，而是可以融入各种实际因素进行修正，例如对于存在构件内依赖性(intra-component dependency)的系统[9]，可以通过“压缩”技术(collapsing)[10]或者选用时间相关的失效函数[11]来修正，使计算结果更加真实。

2　改进的可靠性模型

基于架构的可靠性模型假设基于构件软件的开发过程中构件开发者和架构设计者相互分离，将构件可靠性当作整个模型的可替换的、独立于架构且相互独立的“插件”对待，只刻画了构件失效行为的概率特征，把可靠性当作一种仅依赖于构件本身的静态属性。实际上，构件的可靠性本质上是一种软件属性，它与使用环境有关。在不同的系统操作剖面下，构件之间的转移概率发生变化，同时，构件的使用环境可能会出现较大差异，构件的可靠性也会发生变化，这两个参数直接影响到系统可靠性合成过程中结果的准确性。事实上，来自理论分析和实践经验的证据表明：在架构模型中，相比转移概率，构件可靠性对系统可靠性的方差影响更大[12]。基本出发点就是考虑如何刻画构件本身的可靠性与使用环境的关系，来提高基于架构的可靠性模型评估的准确性。

实际中基于构件的软件系统很难用良好的映射形式来定义构件，因而基于剖面的模型难以在现实系统中得到实施、验证。但是基于剖面思想的一个重要步骤是建立构件在不同输入子域上的可靠度目录，通过所有子域上可靠度的组合表示构件实际的可靠度。在不同输入剖面下这一组合的结果也不同，从而反映出构件实际可靠性随使用环境变化的特征，使得结果更加真实。基本思路就是通过引入剖面的概念来描述架构模型中构件可靠性与架构的关系。

架构模型中的每一对转移关系都代表着转移目的构件的一种使用环境，即输入剖面。当控制流从相同构件(记为构件 i)转移到某个构件(记为构件 j)时，构件 j 所面临的输入剖面可以认为是稳定的；而从不同构件转移到构件 j 时，构件 j 所面临的输入剖面往往差异较大。核心思想就是通过转移矩阵函数来描述构件的输入剖面，建立系统操作剖面和构件可靠性的联系。

改进后模型的各个因素的相互关系如图 1 所描述，图中加粗路径表示新引入的关联。

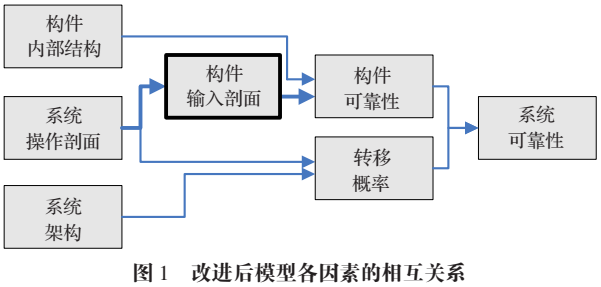

图 1　改进后模型各因素的相互关系

2.1　模型的定义

以文献[8]中提出的基于场景的模型为基础给出改进的可

靠性模型的描述,它本质上是基于路径的模型的一种。该文的思想也可以应用于基于状态的模型,但如何在基于状态的数学模型中描述构件的输入剖面暂不在探讨范围内,是下一步研究的内容。

模型的基础是一个构件依赖关系图 $G=\langle C,s,t,\boldsymbol{R},\boldsymbol{T},\boldsymbol{W}\rangle$,$C$ 是系统中所有构件的集合,$|C|=n$,s 和 t 分别是系统运行的初始状态和终止状态,R 是所有构件的可靠性的集合, 构件的可靠性由它各个输入子域上的可靠度组成的向量表示,$\boldsymbol{T}$ 是构件之间的转移概率矩阵,$\boldsymbol{W}$ 是转移目的构件的输入剖面的矩阵,W_{ij} 是从构件 i 转移到 j 时,构件 j 的输入剖面,用构件 j 各个子域的权重组成的向量表示。G 的一个示例如图 2 所示。

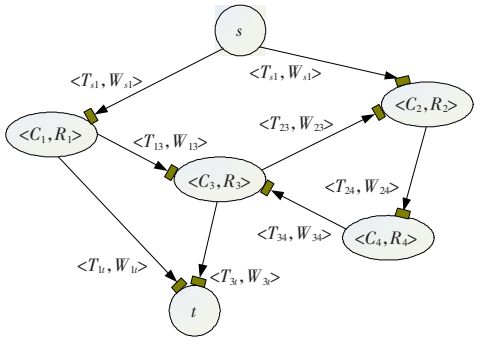

图 2 构件依赖关系图的一个示例

给定某个系统操作剖面, 从构件 i 向构件 j 转移时构件 j 的可靠度可以表示为 $T_{ij}\cdot(R_j\times W_{ij})$,其中 R_j 与 W_{ij} 之间是向量求内积。利用基于路径的模型中的可靠性合成算法框架[8],得到改进后的合成算法如图 3 所示。

```
Algorithm Enhanced Path-Based Reliability Composition(EPBRC)
Input
n:构件的数目
m_j:构件j的子域数目
T_ij:从构件i转移到构件j的概率
W_ij:从构件i转移到j时,构件j的各输入子域所占的权重组成的矩阵
R_j:构件j的各个输入子域1,2,…,m_j上的可靠度组成的向量
Output
R:系统的可靠度
Procedure traverse(R,T,W,n,m,i)
if i==n
  r=1;
else
  r=0;
  for j=1:n
    if T_ij≠0
      r+=T_ij* Σ_{k=1}^{m_j} (R_j(k)W_ij(k))*traverse(R,T,W,n,m,j);
    end
  end
end
Begin
traverse(R,T,W,n,m,1)
End
```

图 3 改进后的合成算法

相比文献[8]中的算法,该文在计算路径的可靠性时用因子 $\sum_{k=1}^{m_j}(R_j(k)W_{ij}(k))$ 取代了原来固定的构件可靠性 $\boldsymbol{R}_j$,即新的构件可靠性因子不仅仅是构件本身的函数,也是转移关系的函数。

2.2 模型的应用步骤

(1)构件开发者根据测试数据或者功能需求划分每个构件 j 的输入子域为 $S_1,S_2,\cdots,S_n$,以及各子域对应的可靠度 $R_1(1)$,$R_1(2),\cdots,R_1(m_j)$。

(2)依据系统架构设计得到系统内部的控制流图,用邻接矩阵 $\boldsymbol{M}$ 表示,如果存在从构件 i 到 j 的转移,则 $M(i,j)=1$;否则 $M(i,j)=0$。

(3)根据先前的经验、历史版本的数据或者系统设计阶段的规格说明刻画系统的使用场景,即赋予每对转移关系一个概率值 p_{ij},从而得到转移概率矩阵 $\boldsymbol{T}=[p_{ij}G(i,j)]$。$\boldsymbol{T}$ 实际上唯一确定了系统在给定操作剖面下的动态架构模型,它是一个一阶齐次有限离散马尔科夫链。

(4) 根据经验数据或者通过场景模拟识别出与转移关系 $\langle i,j\rangle$ 对应的构件 j 的输入剖面,即各输入子域所占的权重值 $W_{ij}(1),W_{ij}(2),\cdots,W_{ij}(m_j)$,得到 $W=[\langle W_{ij}\rangle]$。$W$ 相当于是联系 T 和 R 的纽带。

(5)运用 EPBRC 算法得到系统整体的可靠度。

3 案例分析

(1)假设某个系统在给定操作剖面(1)下动态架构模型的控制流图如图 4 所示。

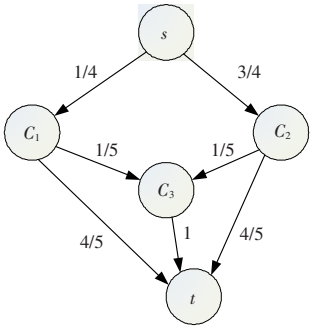

图 4 给定操作剖面(1)下的控制流图

其中状态 s 表示起始状态,状态 t 表示结束状态。图中的节点 C_3 有两条入边,说明构件 3 有两种可能的输入剖面。模型中的转移概率矩阵为:

$$\boldsymbol{T}=\begin{array}{c} \\ s \\ C_1 \\ C_2 \\ C_3 \\ t \end{array}\begin{array}{c} \begin{array}{ccccc} s & C_1 & C_2 & C_3 & t \end{array} \\ \begin{bmatrix} 0 & 1/4 & 3/4 & 0 & 0 \\ 0 & 0 & 0 & 1/5 & 4/5 \\ 0 & 0 & 0 & 1/5 & 4/5 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

为了简化分析不同操作剖面对于构件可靠性的影响,假设构件 1,2 都是正确的。不妨设构件 3 的输入域可以划分为两个子域,构件 3 的可靠度可以表示为:

$$\boldsymbol{R}_3=\begin{array}{c} \\ S_1 \\ S_2 \end{array}\begin{array}{c} \begin{array}{c} C_3 \end{array} \\ \begin{bmatrix} 0.91 \\ 0.72 \end{bmatrix} \end{array}$$

其中元素 $R(k)$ 表示构件 3 在子域 S_k 上的可靠度。特别的对于 T 中的状态 s 和 t,各子域上的可靠度认为是 1。对应转移矩阵 $\boldsymbol{T}$,转移时目的构件的输入剖面矩阵为:

$$\boldsymbol{W}=C_2\begin{array}{c}\begin{matrix}s & C_1 & C_2 & C_3 & t\end{matrix}\\\begin{bmatrix}0 & w_{s1} & w_{12} & 0 & 0\\0 & 0 & 0 & w_{13} & w_{1t}\\0 & 0 & 0 & w_{23} & w_{2t}\\0 & 0 & 0 & 0 & 1\\0 & 0 & 0 & 0 & 0\end{bmatrix}\end{array}$$

由于构件 1,2 是正确的,在任意输入剖面下的可靠度均为 1,所以这里只需要给出构件 3 在不同转移下的输入剖面。其中 w_{13}=[8/10 2/10],w_{23}=[9/10 1/10]。

(2)当系统的操作剖面变化为(2)后,构件之间转移概率也随之发生变化。假设 T_{s1} 和 T_{s2} 发生改变,如图 5:

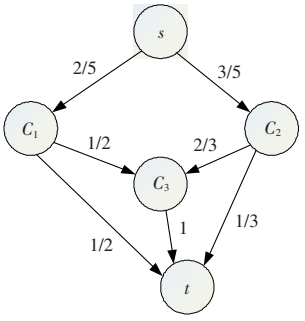

图 5　新的操作剖面(2)下的控制流图

新的转移概率矩阵如下:

$$\boldsymbol{T'}=C_2\begin{array}{c}\begin{matrix}s & C_1 & C_2 & C_3 & t\end{matrix}\\\begin{bmatrix}0 & 2/5 & 3/5 & 0 & 0\\0 & 0 & 0 & 1/2 & 1/2\\0 & 0 & 0 & 2/3 & 1/3\\0 & 0 & 0 & 0 & 1\\0 & 0 & 0 & 0 & 0\end{bmatrix}\end{array}$$

同时构件的输入剖面也随之发生变化,在新的转移目的输入剖面矩阵 $\boldsymbol{W'}$ 中,w'_{13}=[2/10 8/10],w'_{23}=[1/10 9/10]。

运用改进前的模型即基于场景的模型[8],系统中各个构件的可靠度不随输入剖面变化, 构件 3 的可靠度认为是一个常数,可以通过该构件在整个运行过程的输入剖面下,对各子域可靠度求加权和得出:

$$r_3=\frac{R_3\times w_{13}\cdot T_{s1}\cdot T_{13}+R_3\times w_{23}\cdot T_{s1}\cdot T_{23}}{T_{s1}\cdot T_{13}+T_{s1}\cdot T_{23}}=0.86$$

在系统操作剖面改变前后,构件 3 的可靠度和系统可靠度如表 1 所示:

表 1　改进前的模型的变量值

	构件 3 的可靠度	系统的可靠度
系统操作剖面 1	0.86	0.97
系统操作剖面 2	0.86	**0.92**

运用 EPBRC 算法, 构件 3 的可靠度随输入剖面的变化而变化。系统操作剖面改变前后,构件 3 的可靠度和系统的可靠度如表 2 所示:

表 2　改进后的模型的变量值

	构件 3 的可靠度	系统的可靠度
系统操作剖面 1	0.86	0.97
系统操作剖面 2	0.74	**0.84**

对比表 1 和表 2 可以看出,系统操作剖面改变后,改进的模型可以捕捉到构件 3 的可靠度随之发生的变化 (从 0.86 变为 0.74),而原来的模型不具备这种描述能力,使用的是固定不变的可靠度(0.86)。这使得最终合成出来的系统可靠度存在较大差异:改进后的模型得出的系统可靠度(0.84)显然更符合实际,而原来的模型得出的系统可靠度相对该值的误差达 9%,改进前后模型计算结果的差异是显著的。

4　结论

现有的基于架构的模型忽视了软件构件本身可靠性与系统操作剖面的相关性,在对不同系统操作剖面下系统可靠性进行可靠性分析时,只考虑到转移概率的变化而没有考虑构件本身可靠性的变化。将剖面的思想引入基于架构的可靠性模型,可以捕捉到系统操作剖面变化对构件可靠性产生的影响,提高了系统可靠性合成的准确性和可靠性分析的精确性。

模型的改进是以增加参数和计算复杂度为代价的,划分构件的输入子域、确定各个子域的可靠度以及获取转移目的构件的输入剖面都会增加开发的成本。然而,在系统设计阶段,这种可靠性分析的结果可作为构件选择和架构选择的依据,从而降低由于模型不准确引起的决策风险(在案例中带来的误差是比较显著的),因此这种代价是值得的。为了实证该模型对于可靠性合成结果准确性的提高,还需要针对实际的构件软件系统进行实验,这将是下一步的工作。

参考文献:

[1] Szyperski C.Component software:Beyond object-oriented programming[M].New York,NY:ACM Press/Addison-Wesley,1998.

[2] 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.

[3] Hamlet D,Mason D,Woit D.Theory of software reliability based on component[C]//International Conference on Software Engineering,2001.

[4] Hamlet D.Software component composition:A subdomain-based testing-theory foundation [R].Software Testing and Verification Research,2007.

[5] Goseva-Popstojanova K,Trivedi K S.Architecture-based approach to reliability assessment of software systems[J].Performance Evaluation,2001,45:179-204.

[6] Cheung R C.A user-oriented software reliability model [J].IEEE Transactions on Software Engineering,1980,SE-6(2):118-125.

[7] Gokhale S S.Architecture-based software reliability analysis:Overview and limitations[J].IEEE Transactions on Dependable and Secure Computing,2007,4(1):32-40.

[8] Yacoub S,Cukic B,Ammar H H.A scenario-based reliability analysis approach for component-based software [J].IEEE Transactions on Reliability,2004,53(4):465-480.

[9] 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(4):391-412.

[10] Krishnamurthy S,Mathur A P.On the estimation of reliability of a software system using reliabilities of its components[C]//Proc 8th Int'l Symp Software Reliability Eng,1997:146-155.

[11] Gokhale S,Trivedi K.Dependency characterization in path-based approaches to architecture based software reliability prediction[C]//Proceedings of the Symposium on Application-specific Systems and Software Engineering Technology(ASSET'98),1998:86-89.

[12] Goseva-Popstojanova K,Hamill M.Architecture-based software reliability:Why only a few parameters matter[C]//31st Annual IEEE International Computer Software and Applications Conference (COMPSAC 2007),Beijing,2007.