基于软件失效链的软件错误行为分类研究

刘义颖,江建慧
(同济大学 软件学院,上海 201804)

摘 要:目前软件应用广泛,对软件可靠性要求越来越高,研究软件的缺陷—错误—失效过程,提前预防失效的发生,减小软件失效带来的损失是十分必要的。研究描述软件错误行为的属性有助于独一无二地描述不同的错误行为,为建立软件故障模式库、软件故障预测和软件故障注入提供依据。文中基于软件失效链的理论,分析软件缺陷、软件错误和软件失效构成的因果链,由缺陷—错误—失效链之间的因果关系,进一步分析描述各个阶段异常的属性集合之间的联系。以现有的 IEEE 软件异常分类标准研究成果为基础,通过缺陷属性集合和失效属性集合来推导出错误属性集合,给出一种软件错误行为的分类方法,并给出属性集合以及参考值,选取基于最小相关和最大依赖度准则的属性约简算法进行实验,验证属性的合理性。

关键词:软件失效链;软件错误行为;错误行为分类;属性验证
中图分类号:TP311 **文献标识码**:A **文章编号**:1673-629X(2015)04-0001-05
doi:10.3969/j.issn.1673-629X.2015.04.001

Research on Software Error Behavior Classification Based on Software Failure Chain

LIU Yi-ying, JIANG Jian-hui
(Department of Software, Tongji University, Shanghai 201804, China)

Abstract: Software applications are more important than before. The requirements of reliability are more and more higher. It is very necessary to study the process of software defect-error-failure, to prevent failure happened in advance and reduce losses. It is helpful to describe the unique software error behavior and help developers to communicate about this field. It also provides more support with software fault pattern library, software fault detection and fault injection. Based on software failure chain theory, analyze the causal chain of software defect-error-failure, further analyzing and describing each stage abnormal relationships between attributes sets. Based on the existing IEEE software anomaly classification standard, give out software error attributes sets and reference values and a way to classify error behaviors. Verify rationality of attributes by the attribute reduction algorithm of minimal mutual information and maximal dependency.

Key words: software failure chain; software error behavior; error behavior classification; attribute verification

0 引 言

在当今社会,软件应用广泛,已经成为影响国民经济、政治乃至社会生活的重要因素。因此对软件的可靠性要求很高,尤其是对高可靠和复杂的软件系统而言。那么研究软件的缺陷—错误—失效过程,提前预防失效的发生,减小软件失效带来的损失是十分必要的。目前有一些对软件缺陷和软件失效分类的研究,但资料较为匮乏,而对软件错误行为分类的研究少之又少。而对软件错误行为的分类进行研究,给出合理的属性集合,更加准确有效地描述一类软件的错误行为,便于软件开发者之间沟通,为建立软件故障模式库、软件故障预测和软件故障注入提供依据。文中基于软件失效链的理论,以现有研究成果为基础,给出一种软件错误行为的分类方法,并通过实验数据验证属性的合理性。

1 研究背景

在软件可靠性的研究中,对软件缺陷、软件错误、软件失效的研究是必不可少的。本节首先结合软件缺陷、软件错误、软件失效三者的定义,介绍当前的相关

收稿日期:2014-06-19 **修回日期**:2014-09-24 **网络出版时间**:2015-02-23
基金项目:江苏省产学研联合创新资金项目(BY2013095)
作者简介:刘义颖(1989-),女,硕士研究生,研究方向为软件可靠性技术;江建慧,博士,教授,博士生导师,研究方向为可信系统与网络、软件可靠性工程、VLSI/SoC 测试与容错技术。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20150223.1241.038.html

研究成果,其中包括文中的研究基础。文中的“软件错误”也可理解为软件故障,在此意思相同。然后介绍软件失效链理论,给出软件缺陷、软件错误和软件失效三者之间的关系。

1.1　目前研究成果

根据国标 IEEE Std 610-12-1990(1990.9.28)[1],软件缺陷指计算机程序中一个不正确的步骤、过程或数据定义。它通常被认为是软件或系统产生失效的似然条件和推理上的原因。目前对于缺陷的分类方法有正交缺陷分类(ODC)方法[2],该方法主要用于软件开发过程的回溯,其优势在于分类标准不会随开发阶段的变化或开发产品的差异而发生改变。其正交性可保证缺陷的各类别均是截然不同的,且相互独立,因此能减少分类过程中的人为错误。还有早期基于现场数据得到的软件缺陷分类和扩展的缺陷分类方法。

软件错误,又称故障,是系统在运行时由于缺陷造成的非正常状态的表现和反映。软件系统运行时,在一定条件下偏离其预期设计的要求或规定的功能,这种现象称之为软件失效。通常,软件失效行为的监测依赖于系统的可观察性,并与其监测工具和集成运行环境有关。2010 年 IEEE 计算机协会在 IEEE Standard Classification for Software Anomalies[3] 中给出软件缺陷和软件失效的分类标准,既便于开发者和维护者之间沟通异常信息,也有利于描述和检测异常。但这个标准没有给出对错误行为的分类标准,不便于故障模式的描述,因此文中针对这一问题,基于现有的软件缺陷和软件失效的分类标准,结合软件失效链的理论,给出一种错误行为的分类方法。

1.2　软件失效链

软件缺陷、软件错误和软件失效构成一个因果链[4],其中当软件缺陷被激活时,会引起软件错误,当软件错误没有被软件系统容忍时,会引发软件失效,最终表现的是某一/某些功能丧失,甚至是软件系统崩溃或死机。

图 1 给出了缺陷、错误、失效三者之间的因果关系,也反映出当一个缺陷存在时,会导致零个或一个错误,当缺陷被激活,会产生一个错误行为。当一个错误行为没有被软件系统容忍,并引发一个或多个失效行为,最终表现出系统失效。由于缺陷—错误—失效链具有因果关系,那么描述各个阶段异常的属性集合之间应该也存在着一定的联系。下面就要利用异常之间的联系,通过缺陷属性集合和失效属性集合来推导出错误属性集合。

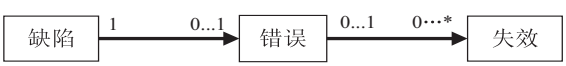

图 1　缺陷—错误—失效链

2　一种软件错误行为分类获取方法

2.1　剖析 IEEE 异常分类标准

首先,进一步剖析 IEEE 给出的软件异常分类标准中对软件缺陷和软件失效具备的属性的特点。在软件缺陷分类标准中,每个分类中包括描述缺陷或失效的属性及其定义,为得到错误行为的属性提供依据。

定义 1　共同属性:是指异常传播的几个阶段,具有相对相同的属性。例如:缺陷严重度和失效严重度,共同属性是严重度。

由定义 1 得到,IEEE 异常分类标准给出的关于缺陷和失效的属性集合中,具有共同属性。IEEE 分类标准中的共同属性,有 ID 号,行为描述,是否有准备处理方案,软件版本,严重度,最终处理状态,发现时间。除了这些共同属性之外,还有各自阶段特有的一些属性,在此只考虑与人员无关的属性。

2.2　基于软件失效链的属性分析

在缺陷—错误—失效链中,各个阶段的属性之间有怎样的关系,有一些问题需要讨论。例如:缺陷阶段具有的属性,在错误和失效阶段是否会有一些共同属性保持不变,是否会有一些属性在另两个阶段不需要,是否在后面的阶段中需要新的属性。诸如此类问题,在研究错误行为的属性过程中,需要考虑并给出答案。下面通过命题的形式来讨论缺陷、错误和失效各个阶段的属性之间的一些关系。

命题 1:共同属性集合,在缺陷—错误—失效链传递过程中保持不变。

分析:按照异常发生的顺序进行分析,在缺陷阶段,用来描述缺陷的基础属性有:ID 号,描述,严重度,软件版本,发现时间,处理状态,是否准备处理方案。当缺陷被激活引发错误行为时,当错误行为没有被容忍,产生失效行为时,都需要这些属性来描述。状态发生改变,但描述状态的角度不变。在这三个阶段中,都应具备共同属性集合中的属性。

命题 2:由缺陷—错误的关系,错误阶段需要跟缺陷阶段相关的属性。

分析:缺陷和错误两个阶段的关系是,缺陷被激活会导致错误行为,一个错误行为是由某一条缺陷导致的,那么对错误行为的描述,需要对引起错误行为的缺陷的类型和缺陷本身进行描述。因此错误阶段需要增加跟缺陷阶段相关的属性:引起错误行为的根源,即缺陷描述和缺陷的类型。

命题 3:由错误—失效的关系,错误阶段需要跟失效相关的属性。

分析:错误和失效两个阶段的关系是,错误行为没有被容忍,被激活的情况下会引起软件失效行为。一个错误行为可能会导致一个或多个失效行为,一个失

效行为是由一个错误行为引起的。那么对一个错误行为描述是,需要对错误行为后续是否引起软件失效进行描述。因此错误阶段增加跟失效阶段相关的属性:错误行为是否引起失效行为。

2.3　软件错误行为属性获取方法

基于 2.2 的分析,下面给出软件错误行为属性获取方法,并给出一套错误属性集合及其参考属性值。

从时间角度进行分析,软件异常行为有三个阶段的状态,软件错误行为阶段、软件缺陷阶段和软件失效阶段。这三个阶段在时间上是顺序进行的,那么对于软件错误行为阶段而言,除了自身阶段,还有之前和之后两个阶段与之有一定联系。那么就从这个视角进行分析,同时也根据 2.1 的分析,可以把错误行为属性分为三类:

(1)缺陷和失效的共同属性,有错误 ID 号,错误行为描述,是否对错误行为有准备处理方案,软件版本,错误严重度,错误最终处理状态,错误发现时间。

(2)与缺陷阶段相关的属性,有错误行为产生的根源,引起错误行为的缺陷类型。

(3)与失效阶段相关的属性,有错误行为是否引起失效行为。

最后总结成错误行为属性集合,如表 1 所示。

有了错误行为属性集合之后,需要讨论每一项属性的参考值,便于研究人员更加准确地描述错误行为。根据 IEEE 异常分类标准[3]和缺陷相关的参考值[5-6]给出错误行为属性的参考值,如表 2 所示。

表 1　错误行为属性集合列表

属性集合	解释	参考值
错误行为 ID 号	错误行为编号	
错误行为描述(Description)	描述具体错误行为	
软件版本(Version)	发现错误行为的软件的版本相关信息	
影响严重度(Severity)	错误行为对软件系统造成影响的严重程度	Critical,Major,Minor,Inconsequential
错误行为产生的根源(Root Cause)	引起错误行为对应的缺陷描述	例如:代码具体出错描述(代码中某一变量设为常数)/代码中某个函数出错
引起错误行为的缺陷类型(Defect Type)	缺陷代码错误种类分类	Data, Interface, Logic, Description, Syntax, Other
错误行为引起失效行为(Failure Behaviors)	错误行为是否引起了软件系统失效,如果是,写出相应的失效行为	无/引起的具体失效行为
对错误行为是否有准备处理方案(Prepared)	错误行为发生前已经准备好处理方案	Open,Closed
错误行为最终处理结果(Disposition)	对错误行为最终的处理结果	不处理,等待处理,已处理
错误行为发现的时间(Time)	操作人员发现这个错误行为的具体时间	

表 2　错误行为属性的参考值

属性	取值	注释
影响严重度(Severity)	严重(Critical)	基本操作受到严重影响无法继续
	中等(Major)	基本操作受到影响,但系统可以继续运行
	低等(Minor)	非关键操作被中断
	微小(Inconsequential)	对操作几乎没有什么影响
引起错误行为的缺陷类型(Defect Type)	数据(Data)	数据定义,初始化,映射访问或使用方面的缺陷
	界面/接口(Interface)	一个界面/接口实现或规范方面的缺陷
	逻辑(Logic)	在决策逻辑,分支,排序或计算算法等实现语言方面的缺陷
	描述(Description)	软件描述或它的使用、安装、操作方面的缺陷
	语法(Syntax)	语法规则不一致方面的缺陷
	其他(Other)	没有定义的其他类型的缺陷
对错误行为是否有准备处理方案(Prepared)	有(open)	出现错误行为之前已准备好进一步处理方案
	无(close)	没有准备进一步处理方案
错误行为最终处理结果(Disposition)	不处理	软件自身容错,未引起软件系统出现失效
	等待处理	危害度较低,或者还未找到出错原因
	已处理	对应缺陷已改正或移除

3　一个例子

下面以实际项目中的软件KVM为例,简单说明一下错误属性集合的用法[7-10]。首先分析研究者的需求,描述错误行为的属性应分为理论部分和实际需求部分。理论部分就是第二节中给出的最基础的属性,实际需求的属性是根据研究的需要,添加其他研究需要的属性,其中项目中的需求包括错误行为模拟对象和方法。那么这里添加属性操作对象和故障模拟方法,这些属性合在一起组成错误属性集合。

通过对软件失效机理分析和故障注入,按照错误属性收集相应信息,根据属性的不同角度描述错误行为。表3显示的是KVM的一条错误行为。

表3　KVM一条错误行为

属性集合	取值
错误行为ID号	1
错误行为描述	无法创建虚拟机列表
软件版本	qemu-img version0.12.1
影响严重度	高
错误行为产生的根源	KVM无法将用户态的MSR列表复制到内核态
引起错误行为的缺陷类型	数据
错误行为引起失效行为	无法获取相应信息
对错误行为是否有准备处理方案	有
错误行为最终处理结果	等待处理
错误行为发现的时间	2014.03.25
操作对象	MSR_List
模拟方法	修改ioctl(*,KVM_GET_MSR_INDEX_LIST,…)系统调用,具体为设置寄存器EAX值为-EFAULT,post_handler返回-1

4　实验验证

第2部分给出了描述错误行为的属性集合,除了理论分析之外,还需要通过实验数据来验证属性集合中属性的合理性。如果存在不合理的属性,那么需要进行属性的选取或约简。

在参考数据挖掘方面,对属性验证的算法[11-14],以及具体的应用环境需求,关于属性集合的合理性,在此问题中,选取最能够反映属性质量的参数进行分析和验证,即相关性和依赖性。在信息论领域中,互信息可以被用来衡量两个随机变量之间的相关性,也有用在属性验证方面的文章。在属性选取或者约简时,如果两个属性的相关性很大,那么已知其中一个属性时,增加另一属性对于增强集合的识别能力的作用不大。故需要最小相关性的属性,也就是属性之间的冗余最小。除了最小相关,还需要考虑属性的重要性方面,因此需要最大依赖度,属性的重要度大。由于错误行为属性的数据都是离散型的,因此选取基于最小相关和最大依赖度准则的属性约简算法[15]进行实验验证。

该算法是基于属性约简算法,和经典粗糙集基于属性重要度属性约简方法不同。在给定一个已选择的约简情况下,下一个条件属性产生方法是,使得它相对于类别的依赖度大,但它与已选属性的互信息平均值小。这样使得约简属性的依赖度大而相互之间的相关性小,减小属性冗余。

实验的软硬件环境如下:算法的实验环境都是PC机,英特尔双核CPU,主频2.93 GHz,1.98 G内存,Windows XP操作系统,应用软件Matlab 7.1。文中采用的实验数据来源于实验室云平台可靠性项目,基于Linux操作系统对Libirt、Openstack软件进行故障注入分析,收集与文中的属性一致的实验数据集为DB_1、DB_2。

将错误行为属性按表4从上到下的顺序,不考虑文字描述性的属性,包括错误行为描述、软件版本、错误行为产生的根源、错误行为引起的失效行为,把其他属性标记为$a_1,a_2,\cdots,a_6$。由于描述性的属性在描述错误行为中是必不可少的,同时在算法中对于文字性的属性,难以进行量化,因此这种属性目前无法通过数值进行衡量。

表4　实验所用2个数据集的基本信息

DB	数据条数	属性个数
DB_1	500	10
DB_2	500	10

表5是最小相关和最大依赖度准则的属性约简算

法的结果。

表 5　实验结果

	a_1	a_2	a_3	a_4	a_5	a_6
DB_1	0.527 5	1.372 7	0.654 8	0.986 4	0.218 3	1.308 7
DB_2	0.527 5	1.091 1	0.873 2	1.527 3	0.653 8	1.124 6

从表中可看出,每个属性的权重都是大于 0 的,说明属性间相关性较小,冗余性小,因此都是有效的,而且都保持在[0,2]之间,数值不大,说明它们之间有一定的依赖性,可以得出结论:当前的错误行为属性集合是合理的,有一定价值的。

5　结束语

文中基于软件失效链的理论,以现有研究成果为基础,给出一套用于描述软件错误行为的分类方法,并给出属性集合以及参考值,且通过实验数据验证了属性的合理性。有助于独一无二地描述不同的错误行为,为建立软件故障模式库、软件故障预测和软件故障注入提供依据。但同时也存在一定的局限性,比如文中的研究对文字描述型的属性不能转化成数值型属性,这样无法对文字型属性进行验证;而且,当前的分类方法适用于研究软件错误行为,对软件的其他行为的研究未必适用,有待进一步分析。

参考文献:

[1] IEEE standard glossary of software engineering terminology [S]. IEEE Std 610.12-1990,1990.

[2] 王　斌,吴太文,胡培培. 软件缺陷分类和分析研究[J]. 计算机科学,2013,40(9):16-20.

[3] IEEE standard classification for software anomalies[S]. IEEE Std 1044-2009,2009.

[4] 徐拾义. 可信计算系统设计和分析[M]. 北京:清华大学出版社,2006.

[5] Ko A J,Myers B A. A framework and methodology for studying the causes of software errors in programming systems[J]. Journal of Visual Languages and Computing,2005,16(1-2):41-84.

[6] Fenton N E,Ohlsson N. Quantitative analysis of faults and failures in a complex software system[J]. IEEE Transactions on Software Engineering,2000,26(8):797-814.

[7] Schneidewind N F. Software reliability model with optimal selection of failure data[J]. IEEE Transactions on Software Engineering,1993,19(11):1095-1104.

[8] Walia G S. Using error modeling to improve and control software quality: an empirical investigation[D]. Mississippi:Mississippi State University,2009.

[9] Parsa S,Vahidi-Asl M,Naree S A. Finding causes of software failure using ridge regression and association rule generation methods[C]//Proc of the 9th ACIS international conference on software engineering,artificial intelligence,networking,and parallel/distributed computing. Phuket: IEEE, 2008: 873 - 878.

[10] Avižienis 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.

[11] Jacobs J,van Moll J,Krause P,et al. Exploring defect causes in product developed by virtual teams[J]. Information and Software Technology,2005,47(6):399-410.

[12] Gu Jifa,Zhu Zhichang. Knowing Wuli. Sensing Shili,caring for Renli:methodology of the WSR approach[J]. Systemic Practice and Action Research,2000,13(1):11-20.

[13] Peng Yi,Kou Gang,Wang Guoxun,et al. Empirical evaluation of classifiers for software risk management[J]. International Journal of Technology & Decision Making,2009,8(4):749-767.

[14] Tan T,He Mei,Yang Ye,et al. An analysis to understand software trustworthiness[C]//Proceedings of the 9th international conference for young computer scientists. [s. l.]:[s. n.],2008:2366-2371.

[15] 万丽艳. 基于最小相关和最大依赖度准则的属性约简研究[D]. 保定:河北大学,2013.

基于软件失效链的软件错误行为分类研究

作者： 刘义颖， 江建慧， LIU Yi-ying， JIANG Jian-hui
作者单位： 同济大学 软件学院,上海,201804
刊名： 计算机技术与发展 ISTIC
英文刊名： Computer Technology and Development
年，卷(期)： 2015(4)