基于 BBNs 的软件故障预测方法

罗云锋,贲可荣
(海军工程大学计算机系,湖北武汉 430033)

摘 要: 本文在分析已有软件故障预测方法后指出:单纯从软件开发过程的某个阶段或基于几种度量来预测软件故障是不充分的.提出综合利用软件开发过程信息构建基于 BBNs 软件故障预测模型.本文从一个基本的贝叶斯信念网(BBNs)故障预测模型出发,扩展基本节点,得到了一个较完善的故障预测模型,结合已有的关于软件度量的研究成果,提出利用软件度量和专家知识确定节点状态概率分布.仿真实验结果表明该模型与实际情况相符合,具有一定的故障预测能力.

关键词: 软件故障预测; 贝叶斯信念网; 软件度量
中图分类号: TP311 **文献标识码**: A **文章编号**: 0372-2112 (2006) 12A-2380-04

BBNs-Based Software Fault Prediction Method

LUO Yun-feng, BEN Ke-rong
(Department of Computer Science, Navy University of Engineering, Wuhan, Hubei 430033, China)

Abstract: It was not sufficient to predict the software fault based on partial phases of software development process or some metrics by analyzing the current fault predicting methods. This paper proposed to build a fault predicting model based on BBNs (Bayesian Belief Networks) by using the information of software development process. Based on a simple BBNs fault prediction model, a general fault prediction model was proposed by expanding the basic node. We brought forward that the software metrics and experts' knowledge could be used to determine the node status probability distribution. The simulation experiment proved that this model with good fault predicting ability coincided with the real case.

Key words: software fault prediction; bayesian belief networks; software metrics

1 引言

当前关于软件故障预测的研究大都集中于软件工程领域的某个方面,如面向对象系统中利用各种度量属性建立模型预测故障数和故障倾向,利用测试过程中用例的覆盖率预测模块故障,利用专家经验建立专家知识库预测故障等等.软件故障的原因贯穿于软件开发全过程,仅从一个方面来考察软件故障是不充分的.近十几年备受关注的贝叶斯网络(BBNs)对于解决复杂系统不确定因素引起的故障具有很大的优势,被认为是目前不确定知识表达和推理领域最有效的理论模型.本文提出基于 BBNs 的故障预测方法,综合利用软件开发过程信息预测软件故障.

2 软件故障预测的研究现状

预测故障的方法可以分为两大类:(1)基于数量的技术,关注预测软件系统中的故障数;(2)基于分类的技术,关注于预测哪些子系统具有故障倾向.第一类已经有一些研究,但是开发一个有效的模型比较困难.第二类方法更成功一些.利用软件度量来预测故障倾向是一个重要的趋势和研究内容,当前的预测模型涉及软件设计度量,代码度量和测试度量.软件维护的历史数据,例如软件改变历史[1]和过程质量数据[2]也被用于软件故障预测.很多专家认为开发过程的质量是产品质量(这里默认是残留故障密度)最好的预测器.Ahmed E. Hassan 等人提出利用启发式规则预测软件子系统故障倾向[3].还有文献提出利用测试过程中的各种数据(如测试覆盖率)来预测故障[2].

分析已有的故障预测模型,它们大多基于软件开发过程中的某一个或几个阶段的数据,或者基于一种或者几种度量,如软件复杂性度量和测试度量.但显而易见,影响软件质量的关键因素不仅仅是其几个度量.软件故障与软件开发全过程往往具有不确定的因果关联关系,导致软件故障的因素很多,单纯从软件开发过程的某个阶段或基于几种度量来预测软件故障是不充分的.BBNs 本身是一种不确定性因果关联模型,具有强大的不确定性问题处理能力,能有效进行多源信息表达与融合.因此本文提出基于 BBNs 构建软件故障预测模型,综合利用软件开发过程信息预测软件故障.

收稿日期:2006-08-15;修回日期:2006-11-10
基金项目:国防预研基金(No. 513270104)

3 贝叶斯网络

一个 BBNs 是一个有向无环图,由代表变量的节点及连接这些节点的有向边构成. 节点代表随机变量,可以是任何问题的抽象,如问题复杂度,观测现象,意见征询等. 节点间的有向边代表了节点间的相互关联关系. 有向图蕴涵了条件独立性假设,用 $A(v_i)$ 表示非 v_i 后代节点构成的任何节点集合,用 $\prod(v_i)$ 表示 v_i 的直接双亲节点集合,则 $P(v_i|(A(v_i)\cup\prod(v_i))) = P(v_i|\prod(v_i))$. 用条件概率表(conditional probabilities table,CPT)来描述点与点之间关联,条件概率表可以用 $P(v_i|\prod(v_i))$ 来描述,它表达了节点同其父节点的相关关系——条件概率. 没有任何父节点的节点概率为其先验概率. 图 1 用 BBNs 描述了一个简单的关于软件产品质量的例子[4],产品质量由管理能力和开发能力确定,表 1 为其 CPT. BBNs 对构造者的信念(专家知识和经验)建模,基于这个模型它能够提供精确的数学计算和预测.

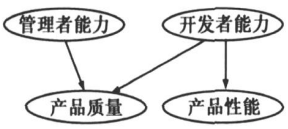


图 1　一个 BBNs 例子

表 1　产品质量的 CPT

管理者能力	高		低	
开发者能力	高	低	高	低
P(产品质量="高")	0.9	0.85	0.35	0.15
P(产品质量="低")	0.1	0.15	0.65	0.85

4 基于 BBNs 的软件故障预测方法

将 BBNs 应用于软件故障预测的步骤是:(1)确定变量及其顺序;(2)建立 BBNs 结构;(3)确定 BBNs 的参数(CPT). 本文从软件开发过程来建立一个 BBNs 基本模型,并以此模型为基础扩展节点.

4.1 一个 BBNs 故障预测的基本模型

影响软件项目风险的基本因素可分为两组,一是与组织相关的因素,包括组织文化,管理经验和能力以及过程成熟度. 二是与项目相关的因素[4]. 影响软件故障的基本因素可以描述为图 2 的基本模型. 方框是可以扩展的基点."项目特征"和"验证和确认"影响到软件开发的需求分析,设计,实现和测试过程,软件故障受开发过程的影响,这个模型涵盖了软件开发过程.

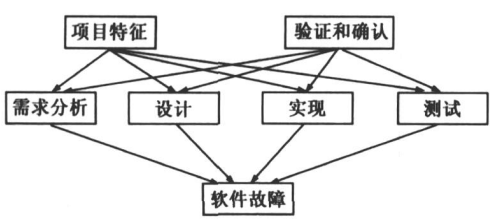


图 2　BBNs 故障预测的基本模型

4.2 扩展的 BBNs 故障预测模型

我们用已探测的故障数,残留故障数,残留故障密度和测试中故障密度四个节点来描述软件故障,分别用"问题复杂度","设计功效"和"测试功效"节点描述需求分析,设计和测试过程. V&V[4]与问题复杂度,设计功效和测试功效三个变量关系紧密,因此本文去掉 V&V 节点,将这些描述 V&V 节点的变量(如测试覆盖率,员工能力等)用来确定问题复杂度,设计功效和测试功效的参数.

本文采用如图 3 所示的 BBNs 故障预测模型,这个模型可以解释为两个阶段:第一个阶段覆盖了软件生命周期的规约,设计和编码;第二个阶段覆盖了测试. 设计规模和缺陷数节点为整数或者一个限定的范围,故障密度为实数,其他节点有下面的状态:很高,高,中等,低,很低. 问题复杂度表示待开发问题集中内在的复杂度,这些问题是规约中离散的功能需求,问题复杂度和设计功效之间的不匹配将导致引入故障数和设计规模增大. 测试阶段在设计阶段之后,实践中实际分配的测试功效比所要求的少得多. 测试功效和设计规模之间的不匹配将会影响已探测故障的数目,引入故障是其边界条件. 已探测故障和引入故障之差是残留故障数. 测试中故障密度是已探测故障和设计规模的函数(已探测故障/设计规模),同样,残留故障密度是残留故障数/设计规模.

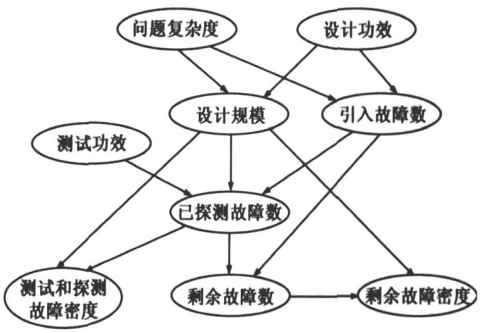


图 3　贝叶斯信念网络故障预测模型

这里的问题复杂度,设计功效和测试功效的粒度仍然较大,不利于确定其状态,将其继续扩展,建立相应子网来描述这些节点:(1)问题复杂度子网(图 4);(2)设计功效子网(图 5);(3)测试功效子网(图 6).

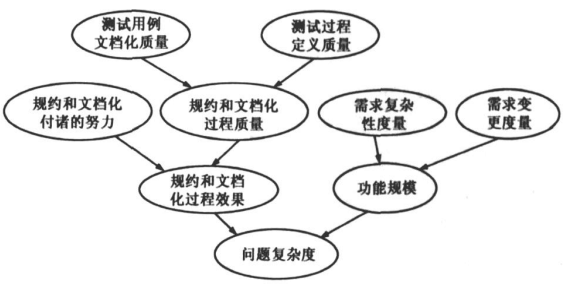


图 4　问题复杂度子网

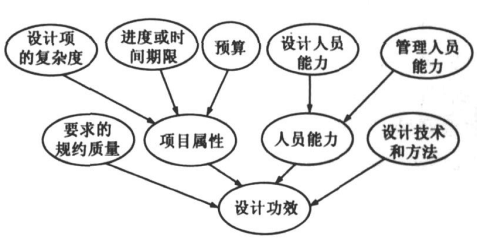


图 5　设计功效子网

图 6 测试功效子网

4.3 确定 BBNs 参数

接下来的问题是确定变量状态的概率和变量之间关系的强度. 从对软件开发过程的各种文档记录中我们可以得到一些确定性知识. 对于不确定性知识,传统的方法是根据专家经验主观确定. 研究人员定义了大量软件度量描述软件质量[2,5,6],将这些研究与专家知识和经验结合起来确定 BBNs 参数.

4.4 推理规则

采用应用最广的随机模拟采样法(Pearls and Gibbs 算法). 首先,为网络上的节点做初始实例化,证据节点实例化为观察值,非证据节点实例化为随机值;然后,开始遍历图,对每一非证据节点 Y,计算在其他节点给定值的情况下 Y 的后验概率分布:

$$P(Y|\ W_Y) = \alpha P(Y|\ Pa(Y)) \prod_i P(s_i|\ Pa(s_i))$$

式中,W_Y 表示除 Y 的节点集合,S_i 表示 Y 的第 i 个子女,为正规化因子,其余乘积项为条件概率. 公式表明了本节点的概率仅与其父母节点,子节点及其子节点的父母节点有关;Pearl 使用上式结果对节点进行采样,结果作为 Y 的新实例化,反复进行,直到近似过程收敛(设进行了 m 次遍历),这时查询结果为:$P(Y|\ e) = \frac{1}{m}\sum_{i=1}^{m} f_i$,$f_i$ 为第 i 次遍历 Y 的条件概率,e 为证据向量的观察值.

5 仿真实验

本文在 AgenaRisk[7]系统中对该模型进行仿真实验. 实验部分采用了 AgenaRisk 中关于软件故障预测和软件项目风险管理的数据. 由于具体的项目数据难以收集,我们根据图 3 所描述的简化模型来做仿真实验. 在实验中我们用软件需求复杂性度量和软件需求变更度量来描述问题复杂度[6]. 利用各种度量来描述设计功效,包括对象(模块)之间的耦合数(耦合度量),不使用公共属性的方法的个数(内聚度量),继承树的深度和继承的平均深度(继承度量)[5]. 用代码覆盖度量来描述测试功效,定义一个相应策略的测试有效率(test effectiveness ratio,TER),TER1 是语句覆盖的测试有效率,TER2 是分支覆盖的测试有效率,TER3 是线性代码顺序和跳转覆盖测试有效率. 我们设定的是一个中等规模的系统,严格按照软件工程开发过程开发,花费了大量资源在设计和测试上,尽量减少耦合,增加内聚,TER1,TER2 达到 100 %,TER3 达到 90 %,因此可以判定设计功效为很高(概率为 100 %),测试功效很高(概率为 100 %),如图 7 所示. 从仿真结果可以看到设计规模较小,引入故障数较少(期望值为 17.8),已探测故障密度相对较高,剩余故障数较小(期望为 6.6),这与实际情况是相符合的. 当我们将设计功效设置为较低时(概率为 100 %),如图 8 所示,明显设计规模变大,引入故障数增加(期望值为 43.1),相应的剩余故障数增加(期望值 13.0),已探测故障密度减少. 表 2 是两者的对比结果. 在实验中我们分别对问题复杂度,测试功效和设计功效赋值,以检查模型对各种环境下的变化,其结果与实际较为符合,说明了模型的合理性.

表 2 仿真结果

条件 / 节点	设计功效很高	设计功效较低
设计规模	较小	较大
引入故障数	较少(期望值为 17.8)	增加(期望值为 43.1)
已探测故障密度	较高	减少
剩余故障数	较小(期望值为 6.6)	增加(期望值 13.0)

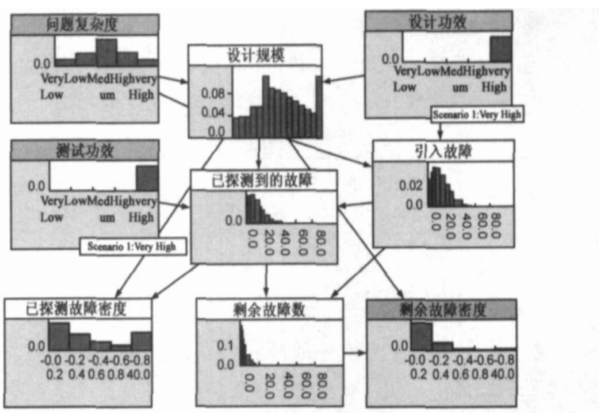


图 7 BBNs 故障预测模型的仿真图 (a)

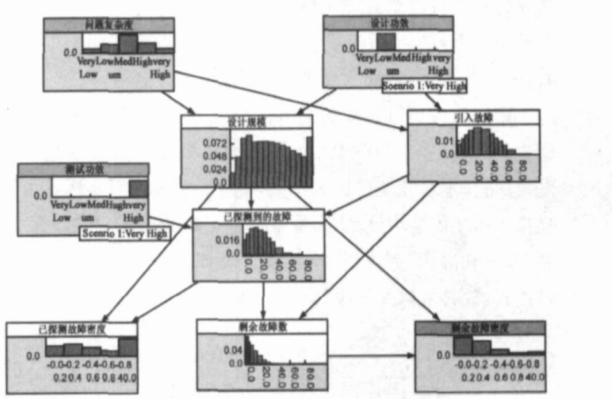


图 8 BBNs 故障预测模型的仿真图 (b)

6 结语

本文从软件开发全生命周期来考察故障,给出了一个 BBNs 故障预测原型系统,并在 AgenaRisk 系统中对该模型进行仿真实验. 从实验结果可以看到,BBNs 能够使用来自主观和客观的概率分布和不充分的数据预测软件故障数. 仿真实验还只是基于一个简化的模型,将实际项目数据应用于模型,探讨建立完备网络结构和确定节点状态的方法,建立适应具体项目便于数据收集和确定节点状态的网络是需要进一步探讨的问题.

参考文献:

[1] Todd L Graves, Alan F Karr, J S Marron, Harvey Siy. Predicting fault incidence using software change history [J]. IEEE Transactions on Software Engineering, 2000, 26(7): 653 - 661.

[2] Fenton, NE, Neil M. A critique of software defect prediction models [J]. IEEE Transactions on Software Engineering, 1999, 25(5): 675 - 689.

[3] Ahmed E Hassan, Richard C Holt. The top ten list: dynamic fault prediction [A]. Proceedings of the 21st IEEE International Conference on Software Maintenance (ICSM'2005) [C]. Budapest, Hungary: IEEE, 2005. 263 - 272.

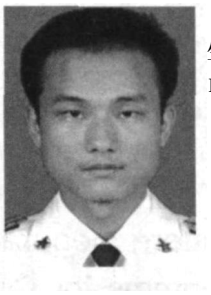
[4] Chin-Feng Fan, Yuan-Chang Yu. BBN-based software project risk management [J]. Journal of Systems and Software, 2004, 73(2): 193 - 203.

[5] Munson J C, Nikora A P. Toward a quantifiable definition of software faults [A]. Proceedings of 13th International Symposium on Software Reliability Engineering (ISSRE 2002) [C]. Annapolis, MD, USA: IEEE, 2002. 388 - 395.

[6] 王青,李明树. 基于 SPC 的软件需求度量方法[J]. 计算机学报, 2003, 26(10): 1312 - 1317.
Wang Qing, LI Ming-Shu. Measurement of Software Requirement Based on SPC [J]. Chinese Journal of Computers, 2003, 26(10): 1312-1317. (in Chinese)

[7] AgenaRisk. http://www.agenarisk.com [EB/OL]. 2005/2006-7-15.

作者简介:



罗云锋　男,1982 年生于湖北,硕士研究生,研究方向为软件维护.
E-mail: hehelyf@yahoo.com.cn

贲可荣　男,1963 年生于江苏,博士,海军工程大学教授,博士生导师,研究方向:软件工程、人工智能.