**参考文献记录**

目录

[1.《软件故障定位关键技术研究综述》 2](#_Toc522809412)

[2.《The DStar Method for Effective Software Fault Localization》 2](#_Toc522809413)

[3.《基于变异的错误定位优化技术研究》 3](#_Toc522809414)

[4.《针对基于变异错误定位的一种动态变异执行策略》 3](#_Toc522809415)

[5.《Metallaxis-FL: mutation-based fault localization》 3](#_Toc522809416)

[6.《基于变异技术的程序故障自动化修复方法》 3](#_Toc522809417)

[7.《基于错误分类谱的错误定位方法的研究》 4](#_Toc522809418)

[8.《基于频谱的软件多故障定位》 4](#_Toc522809419)

[9.《基于谓词切换的BPEL程序故障定位技术与支持工具研究》 4](#_Toc522809420)

[10.《面向BPEL程序的故障定位方法与工具研究》 5](#_Toc522809421)

[11.《面向BPEL程序的变异测试技术与支持工具研究》 5](#_Toc522809422)

[12.《基于切片的BPEL程序故障定位技术与支持工具研究》 6](#_Toc522809423)

1.《软件故障定位关键技术研究综述》

作者介绍：逢禹，上海理工大学。软件导刊（2017）。

主要内容：介绍了传统的故障定位技术以及2010年以来故障定位的相关成果，总结了常用的测试程序集。

故障定位常见技术有：

1. 基于覆盖的故障定位技术：采用可疑度描述来降低定位复杂度。

基于程序切片的故障定位技术：将程序抽象化，通过删除不相关的部分将程序用切片表示，切片保留原始程序的行为规范。

1. 基于程序频谱的故障定位技术：根据测试用例集合运行完成后的记录信息，使用风险评估方法对程序中的每一行语句计算故障可疑度。

最新推广的故障定位技术：

1. 基于数据挖掘的故障定位技术：从大量程序运行信息数据中提取相关信息，产生一个可使用的数据模型，从而达到定位效果。
2. 基于模型检测的故障定位技术：通过推导期望的程序行为模型，检测失效执行对期望行为的违背情况，从而捕捉故障行为。
3. 使用覆盖率数据集成技术，通过创建可疑语句排名列表的技术进行定位。
4. 以信息检索技术为切入点的故障定位技术：使用基于源代码的搜索得到故障相关报告的降序等级排列。
5. 算法调试的故障定位技术：将复杂度计算分解为一系列子计算。
6. 公式故障定位技术：依赖于对编码失败的程序公式执行轨迹跟踪。

常用程序集：来源于SIR和SourceForge，最常用Siemens、grep、flex。

2.《The DStar Method for Effective Software Fault Localization》

作者介绍：ic Wong, Vidroha Debroy, Ruizhi Gao, and Yihao Li。IEEE Transactions on Reliability, 2014, 63(1):290-308

主要内容：

介绍了一种技术叫DStar（D\*），可以给自动NCF化的故障定位提供可疑的位置，每条语句的可疑度计算公式为：

(NCF)\*

(NUF+ NCS)

\*代表指数，并且D\*这项技术在单错误和多错误场景中的错误定位精度高于其他类型

比其他故障定位技术更有效，当指数增长，有效性增强，直到达到平衡为止。

3.《基于变异的错误定位优化技术研究》

作者介绍：刘勇，北京化工大学。硕士论文（2018）

1. 在Simens程序套件上的实验结果表明，Tarantula技术的错误定位效果优于并集、交集和最近邻等方法，Jaccard, Ochiai, Op2错误定位效果与Tarantula相当或者优于它。
2. 怀疑度计算公式是基于程序谱的错误定位方法的研究内容之一，有30多个不同的怀疑度计算公式。
3. 大量实验结果显示没有一个公式能在所有情况下都比其他所有公式错误定位精度好，最新Wong等人提出来的DStar公式单错误和多错误场景中的错误定位精度高于其他类型相似的CBFL公式。
4. 一种基于变异的错误定位技术MUSE，实验证明错误比现有的CBFL：Jaccard, Ochiai, Op2等更好，Metallaxis-fl定位精度比MUSE更好。
5. MBFL是现有错误定位技术中定位精度最好的，但是时间和资源开销大。

4.《针对基于变异错误定位的一种动态变异执行策略》

作者介绍：龚沛、耿楚瑶、郭俊霞、赵瑞莲，北京化工大学。计算机科学（2016）

提出一种动态变异执行策略：

1. 首先被测程序执行测试用例集，搜集被测程序语句覆盖信息及测试用例执行结果。
2. 筛选出被失败测试用例所覆盖的语句，对这些语句植入人工故障，生成变异体。
3. 对所有变异体执行失败的测试用例，得到akf和anf值，然后选择一个基于频谱的语句怀疑度计算公式计算所有变异体的怀疑度上界值，并按从大到小顺序排列。根据排列顺序逐个对变异体执行通过的测试用例，【执行通过的测试用例的执行顺序也通过一定规则进行排列】，计算语句上变异体怀疑度的最大值，就是语句的怀疑度。
4. 被测程序的所有语句按照怀疑度值从大到小排序，得到怀疑度表。

5.《Metallaxis-FL: mutation-based fault localization》

作者介绍：Mike Papadakis, Yves Le Traon. Software Testing, Verification & Reliability(2015)

本文是最早提出基于变异的故障定位技术，将变异体和故障程序的位置联系起来，利用人工注入故障与实际故障之间的相似性（**核心思想**）来定位软件中的错误实体。被失败执行的测试用例杀死越多的变异体，越能预测一个错误的位置。实验验证基于变异的方法比基于语句的方法具有更高的定位精度。

6.《基于变异技术的程序故障自动化修复方法》

作者介绍：马春燕、刘杰：西北工业大学，赖文豫：中国人民大学。计算机应用研究（2014）

主要内容：采用故障定位技术，提出一种基于变异的程序故障自动化修复方法。具体是利用当时最有效的故障定位技术——基于频谱的故障定位技术计算故障程序的语句怀疑度，从高到低排序，依次从怀疑度最高的语句开始，选择该语句试用的变异算子集生成变异体，判断每一个变异体是否为潜在修复。

可改进之处：

1. 直接使用基于变异的故障定位计算可疑度，效率更高（2015年提出，早于2014），并且定位的同时可以省去生成变异体，直接修复即可。
2. 针对语句缺失的修复，也就是扩展变异算子。
3. 针对大型程序的改进，也就是不只是几百行。
4. 多故障系统的修复。

7.《基于错误分类谱的错误定位方法的研究》

作者介绍：刘秀静：北京化工大学。硕士论文（2017）

主要内容：针对基于程序谱的错误定位方法的错误定位精度不高的问题，提出一种基于语句特征的错误分类方法。利用统计学原理，计算与类型相似的历史程序版本中每类语句发生错误的概率，然后将基于语句特征的错误分类方法与基于传统的SBFL方法相结合，利用每类语句方法错误的概率，修正SBFL计算出的语句怀疑度值，提高错误定位精度。

8.《基于频谱的软件多故障定位》

作者介绍：宗芳芳：浙江理工大学。硕士论文（2015）

主要内容：提出一种基于二次定位策略的故障定位技术。第一次定位：首先使用工具生成函数调用图，从函数的覆盖信息中建立程序频谱，利用基于模型的诊断MBD的逻辑推理和构建行为命题模式来推导出函数候选集以及他们的概率，根据概率进行排序。第二次定位：利用DStar技术将函数候选集的语句进行语句的定位。

9.《基于谓词切换的BPEL程序故障定位技术与支持工具研究》

作者介绍：郑彩云：北京科技大学。硕士论文（2015）

主要内容：提出基于谓词切换的BPEL程序故障定位技术，通过谓词切换逐步缩小程序的故障定位范围，并采用与关键谓词（在谓词切换过程中，对于相同的测试用例，若切换后的程序的输出结果与预期结果一致[这是针对有故障的版本]，则该谓词为关键谓词）相关的程序切片来分析故障的根源。具体步骤如下：

1. 获取执行失败的测试用例集合
2. 记录执行路径中的谓词集合，分别更改每一个谓词的布尔值，强制改变判断分支，如果执行结果与预期不一致，则继续找下一个谓词，知道找到一致的，即为关键谓词。
3. 根据关键谓词来寻找后向数据切片，并根据后向数据切片来确定故障语句的集合。其中后向切片的步骤如下：
4. 找到关键谓词相关的变量集合V（包括变量与常量）
5. 寻找所有与V相关的后向切片Bv（直接或间接与V中元素相关）
6. 检测Bv中元素所在的块，若该语句块为非原子节点则深入到块内部去搜索和谓词相关的原子结点，这些原子结点的集合是故障怀疑度最高的块。

值得思考之处：在实验评估中，对于Tarantula的定位效率的比较值得商榷。

根据定义，定位效率（也叫定位准确率）指的是：给定一个故障版本集合总数为N，其中使用故障定位技术能被正确定位出的故障版本个数为M，那么，故障定位准确率(Success Rate)的计算公式为M/N。

定位精度：给定一个WS-BPEL程序，程序活动总数为A，应用基于故障定位技术定位到故障位置所需检查的活动数为C，那么，故障定位精度(Identification Ratio)的计算公式为C/A。

Tarantula最终结果是对语句的可疑度进行排序，供开发者从怀疑度最高的开始检查，最终一定能查到存在故障的语句（针对BPEL则是活动），不知道实验者是如何判断该定位效率的，更适合用定位精度来评估该方法。

10.《面向BPEL程序的故障定位方法与工具研究》

作者介绍：翟忆蒙：北京科技大学。硕士论文（2013）

主要内容：作为实验室研究BPEL程序的故障定位的开山鼻祖，首次提出适用于BPEL程序的故障定位技术，给出怀疑度最高的语句块，步骤如下：

1. 将BPEL程序分为一个个Block语句块，包括原子语句块和非原子语句块；
2. 运行测试用例，找出执行失败的测试用例集ts；
3. 对于ts中的每个测试用例，再次执行BPEL程序，记录每个测试用例经过的语句块，得到集合cs;
4. 用cs作为输入，利用几种基于频谱的故障定位方法计算每个Block的怀疑度得分，得到怀疑度集合mssb
5. 根据怀疑度排名集合mssb，得到最可能存在故障的语句块集合。

11.《面向BPEL程序的变异测试技术与支持工具研究》

作者介绍：王巧玲：北京科技大学。硕士论文（2014）

主要内容：根据BPEL程序自动生成变异体，分析与评估不同类型的变异算子产生变异体的难易程度，精简变异算子。具体步骤如下：

1. 对BPEL程序解析，识别可以变异的元素，记录匹配的代码；
2. 根据记录点找到相关变异算子模块，模块里存储着每个变异算子对应的错误植入规则；
3. 根据变异算子转换规则，生成对应变异体；
4. 识别变异体，得到精简变异体；
5. 启动BPEL引擎，输入测试用例，执行原始BPEL文件，并记录执行结果。
6. 将生成BPEL程序定义文件的精简变异体重新部署到BPEL环境中，输入上述测试用例，执行变异体文件，与原始BPEL结果进行比较，记录测试结果。
7. 将变异体执行完毕后，统计测试结果，对测试结果进行分析，找出变异算子故障类型被检测的难易程度，对变异算子进行精简，得出精简变异算子。

12.《基于切片的BPEL程序故障定位技术与支持工具研究》

作者介绍：冉玉凤：北京科技大学。硕士论文（2016）

主要内容：提出基于切片和谓词切换的故障定位方法。首先通过谓词切换技术找到谓词选择部分的错误，找到程序的关键谓词。对关键谓词进行后向数据切片，找到数据依赖部分的错误。

13.《基于场景的BPEL测试用例自动生成技术与工具研究》

作者介绍：赵彦：北京科技大学。硕士论文（2013）

主要内容：提出一种基于场景的BPEL测试用例自动生成技术。首先将BPEL转换为一种只包含活动与边的抽象图模型，之后依照一系列变换规则将图模型转换为一种中间表示形式——扩展的与或树，根据该中间表示形式并结合不同的覆盖策略生成测试路径集。针对每条测试路径运用基于约束求解策略得到每条路径的测试数据，最终得到测试用例集合。