# 故障模式建模算法详细设计说明书

**陈碧欢，宋文博，孙庆浩，涂轩涵，彭鑫**

**复旦大学，计算机科学技术学院**

目录

[一. 系统结构 3](#_Toc530063800)

[二. 模块介绍 3](#_Toc530063801)

[1.1 模拟模块 3](#_Toc530063802)

[1.2 算法模块 3](#_Toc530063803)

[1.2.1 子模块介绍 3](#_Toc530063804)

[1.2.2 算法流程 4](#_Toc530063805)

[1.2.3 具体策略 4](#_Toc530063806)

## 一. 系统结构

根据智能测试平台系统架构图可知需要实现的核心部分为算法模块和模拟模块。现将这两块细化如图1，并在下文中分别阐述两个模块的详细设计。

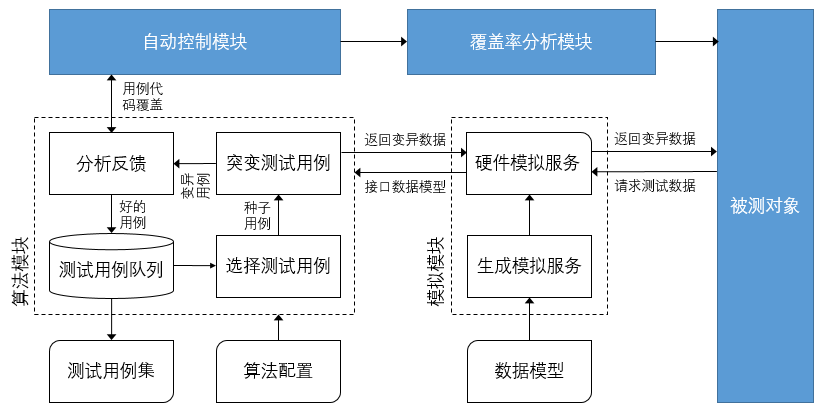


图1

## 二. 模块介绍

### 1.1 模拟模块

模拟模块模拟硬件层的输入输出，主要功能是生成硬件模拟服务。

基于数据模型中的接口、输入/输出取值范围等信息，同时解析复杂接口的输入格式，建立从接口+输入信息到输出取值范围的映射关系供硬件模拟服务使用。

硬件模拟服务的作用主要有请求测试数据和返回变异数据两种。收到被测对象的测试请求时，根据被测对象的接口和输入向算法模块发送对应的接口数据模型（输出取值范围）；当算法模块返回变异数据时，将变异数据打包成合适格式发送给被测对象。

### 1.2 算法模块

#### 1.2.1 子模块介绍

算法模块采用遗传算法的思想，大致思路为根据模拟模块传入的数据模型及测试用例队列里的现有用例生成新测试用例，将新测试用例发送给被测对象获取代码覆盖率信息，不断迭代此过程直到总代码覆盖率达到目标值。该模块分为分析反馈模块，测试用例队列模块，选则测试用例模块和突变测试用例模块。

分析反馈模块用于接收被测对象发回的对应模拟模块所发送的变异数据的反馈，并将反馈中所包含的覆盖率信息和对应测试用例相结合。去除未覆盖独特分支的测试用例后将得到的包含覆盖率信息的用例放入测试用例队列中。

测试用例队列为保存测试用例以及其覆盖率信息的数据结构。生成的初始测试用例和突变算法生成的新测试用例全部都加入到队列中进行维护。当队列中的用例数量过多时，会进行重新评估过程去除没有覆盖独特分支的冗余测试用例，在不影响总覆盖率的情况下对队列中的用例进行精简。

选择测试用例模块用于从测试用例队列中选择出遗传算法所需要的种子测试用例。选择方式基于某些价值判断函数。选择出种子测试用例后将其传递给突变测试用例模块进行突变。

突变测试用例模块将选择出的种子用例突变为变异用例，再经由模拟模块发送给被测对象以获取覆盖率信息。可用的突变策略有加1、减1、取边界值、交叉等，还将根据具体的数据类型设定针对性的突变策略。

#### 1.2.2 算法流程

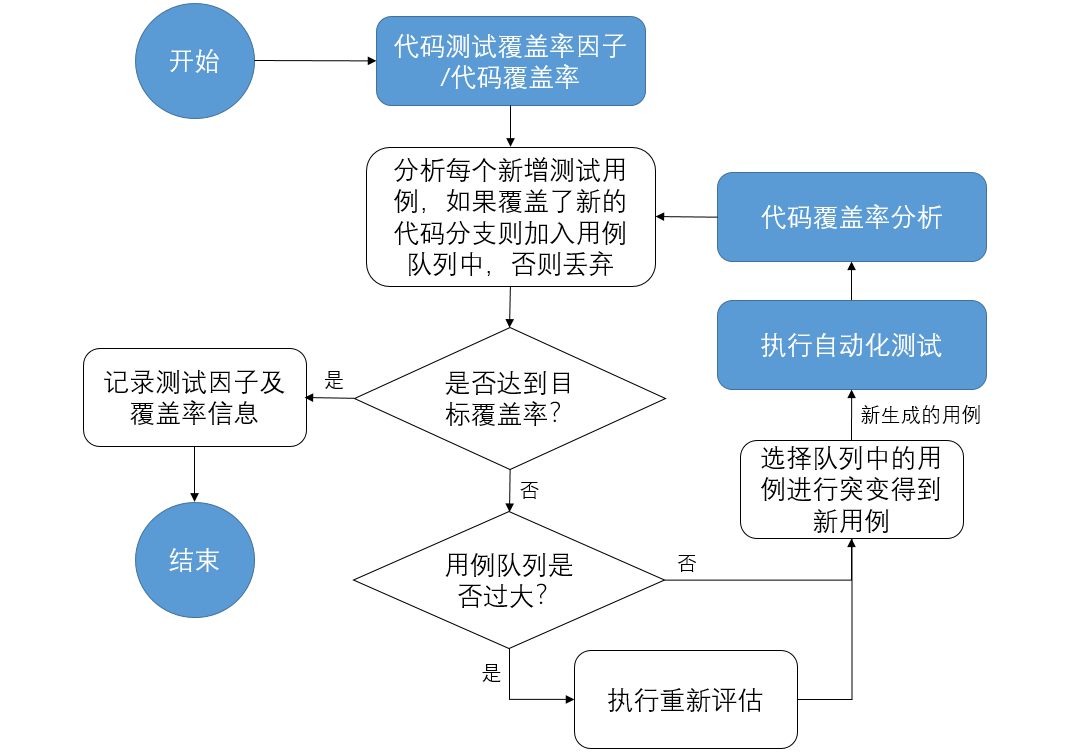


图2

完整的算法流程图如图2。接收到模拟模块发送的接口数据模型后，算法模块生成一批基础测试用例，作为最初的变异数据经由模拟模块发送给被测对象。分析覆盖率后在分析反馈模块将得到的覆盖率信息和对应的变异用例结合。分析每个新增测试用例，如果覆盖了新的分支则加入测试用例队列中，否则丢弃，此时队列中的用例即为初始的代码测试覆盖率因子/代码覆盖率。对总覆盖率进行判断，若未达到目标值，在完成对测试用例队列的精简后从队列中选择出一批种子测试用例，基于模拟模块传入的数据模型，利用一定的突变策略生成新的测试用例。将新生成的测试用例发送给被测对象执行自动化测试，丢弃未覆盖新分支的用例后判断此轮突变是否使代码覆盖率达到目标值。若没有则重复循环，若达到目标值则输出测试因子集、总代码测试覆盖率以及每个测试因子与代码覆盖率的关系并结束测试。

#### 1.2.3 具体策略

需要详细叙述的策略包括选择测试用例模块中使用的选择策略和突变测试用例模块的突变策略两种。下面将分别叙述二者可能使用的策略。

a. 测试用例选择策略

可能的测试用例选择策略有两种，第一种是简单的顺序策略，即选择队列中最新加入的用例进行突变。这样做优点是可以在不影响效率的情况下保证每次的种子测试用例大部分是未进行过突变的新用例。第二种是优先级策略，即每次在往测试用例队列中加入新的测试用例时，根据一定的适应度算法对测试用例进行排序，再选出适应度最高的用例作为种子用例进行突变。这样做的优点是突变出的新测试用例覆盖新分支的可能性更高，缺点是每次的排序过程可能消耗较多时间。在实际开发过程中会根据突变效果和运行时间等因素选择出较优的方法。

适应度算法主要是为了衡量用例突变出新测试用例的能力。前文中的测试用例队列中保存的为当前所有有效用例（包括需要计算适应度的用例自身）， 同时，设计数据结构保存每个用例生成的新用例数量。现需要计算用例a的适应度f(a)，令其生成的有效用例总数为m，由a生成的用例总数为n，则衡量适应度的公式为：

在生成的总用例数n上加1的原因是新突变出并被放入用例池的用例的n值为0，加1可以防止除0情况的出现。如此计算适应度可以保证每个新加入用例池的用例的初始适应度恒定为最大值1，即一定有机会进行一次突变，不会直接沉到测试用例队列底部。同时每次迭代中每个进行了突变的测试用例的适应度都会发生更新，从而不断地将更好的用例排列到队列前端，以达到更好的效果。这样做的缺点是每次迭代都要更新所有用例的子代用例数、适应度等信息，还要对用例队列进行排序，时间上的开销会比较大，可能需要设计特殊的数据结构减小开销。

b. 突变策略

接口分为简单接口和复杂接口两种。简单接口直接根据输出取值范围进行突变，复杂接口则根据获得的输入信息先从数据模型内取出正确的输出取值范围，再根据范围进行突变。

无论是简单接口还是复杂接口输出格式均固定，包括三个字段：info，result和value

Info字段只需原样传入传出，不需要进行突变。

Result字段有true和false两个可能取值，突变时只需要在true和false之间转换，或者取其他值（如空值）测试非法值情况。

Value字段有以下三种情况：

1、      数字类型  0-100

2、      字典类型  { "Bbl\_Second BOOT Version": "1C.004.10.019",  "Bbl\_First BOOT Version": "1C.004.10.019",  "Bbl\_Host Software Version": "50.270.HA.110", "Bbl\_Current Running Version": "50.270.HA.110",  "Bbl\_Equipment Software Version": "NULL"}

{"data": [[0, 256, 4650160, 100000, 2970, 0 ], [1, 0, 0, 0, 0, 0, 0, 0], [0, 256, 4750160, 100000, 2970, 0], [0, 0, 0, 0, 1, 0, 0, 0], [139 ], [136, 3], [22364160]], "length": 4, "type": [6, 5, 6, 5, 133, 134, 144]}

3、      字符串：SMW200A

三种情况的突变方法如下表：

|  |  |  |
| --- | --- | --- |
| 类型 | 示例 | 突变方式 |
| 数字 | "value": 0-100 | 1.取边界值  2.加1/减1  3.取特殊值，如0，1等  4.取范围外的值，如INT\_MAX |
| 字典类型 | {"Bbl\_Second BOOT Version":"1C.004.10.019",   "Bbl\_First BOOT Version": "1C.004.10.019",  "Bbl\_Host Software Version": "50.270.HA.110", "Bbl\_Current Running Version": "50.270.HA.110",  "Bbl\_Equipment Software Version": "NULL"} | 1.取其他随机值  2. 取相邻值  3. 取字典中未包含的值 |
| 字符串 | 字符串：SMW200A | 据了解得知单个字符串对覆盖率影响很小，故不对其进行突变或是仅将其突变为空值测试非法输入。 |

表1

现举两个例子说明具体的突变方法：

1. 接口：GET /rru/0/download-file-percent HTTP/1.1

接口描述：查询目前下载文件的进度

接口类型：简单接口

输入取值范围：空

输出取值范围：

{

"info": {

"en": 英文字符,

"cn": 中文字符

},

"result": true/false,

"value": 0-100

}

假设种子测试用例为：

{"info": {"en": 英文字符, "cn": 中文字符}, "result": true, "value": 20}

可对result字段进行突变，即将true突变为false，变异后的用例为：

{"info": {"en": 英文字符, "cn": 中文字符}, "result": false, "value": 20}

采用取边界值的方法对value字段进行突变，变异后的用例为：

{"info": {"en": 英文字符, "cn": 中文字符}, "result": true, "value": 100}

采用加1/减1的方法对value字段进行突变，变异后的用例为：

{"info": {"en": 英文字符, "cn": 中文字符}, "result": true, "value": 21}

采用取范围外的值的方法对value字段进行突变，变异后的用例为：

{"info": {"en": 英文字符, "cn": 中文字符}, "result": true, "value": INT\_MAX}

2. 接口：PUT /rru/0/upload-file HTTP/1.1

接口描述：上传文件

接口类型：复杂接口

输入取值范围：

{"file\_type": 5, "oper\_type": 0}

{"file\_type": 23, "oper\_type": 0, "save\_file\_name": "certificate\\

2102312AHQ10J8000019\\CertReqFile.req"}

{"file\_type": 12, "oper\_type": 0, "timeout": 300}

输出取值范围：

{

"info": {

"en": "",

"cn": ""

},

"result": true/false,

(5)"value": "/$[ArchivesInfo Version]\r\n/$ArchivesInfoVersion=3.0

\r\n\r\n\r\n[Board Properties]\r\nBoardType=WDHMGAP38B01\r\nBarCode=

2102312AHQ10J8000019\r\nItem=02312AHQ\r\nDescription=Function Module,HAAU5213,WDHMGAP38B01,TX26500~29500MHz/RX26500~29500MHz,-48VDC,25G eCPRI,4T4R,65dBm\r\nManufactured=2018-08-29\r\nVendorName

=Huawei\r\nIssueNumber=00\r\nCLEICode=\r\nBOM=\r\n"}

(12)value："[hex]9a a2 99 c9"

(23)value：null

}

假设输入为：

{"file\_type": 12, "oper\_type": 0, "timeout": 300}

根据输入中的"file\_type": 12，可以得到当前的输出取值范围：

{

"info": {

"en": "",

"cn": ""

},

"result": true/false,

value："[hex]9a a2 99 c9"

}

假设此时种子用例为：

{"info": {"en": "", "cn": ""},"result": true, value："[hex]9a a2 99 c9"}

可对result字段进行突变，将true突变为false，变异后的用例为：

{"info": {"en": "", "cn": ""},"result": false, value："[hex]9a a2 99 c9"}

将result字段突变为范围外值，变异后的用例为：

{"info": {"en": "", "cn": ""},"result": null, value："[hex]9a a2 99 c9"}

将value字段的字符串突变为范围外值（空值），变异后的用例为：

{"info": {"en": "", "cn": ""},"result": true, value：""}

每次突变时选取测试用例队列中适应度最高的用例作为种子用例。