UFT

2019年11月11日

1:39

Unified Functional Testing:

* 1. 是HP公司的关键字驱动的功能测试工具
  2. 支持的脚本语言:VBScript
  3. 擅长:Java,Web,.Net
  4. 超过Winrunner的主流测试工具

测试过程:

定制测试计划->创建测试脚本(录制)->增强测试脚本功能->运行脚本->分析结果

* 1. 定制测试计划:是根据被测项目的具体需求,以及所使用的测试工具的自身特点而定制的,它完全用于指导自动测试过程.

它的两种体现方式是:以独立文档的形式出现,另一种是以项目计划附件的形式出现.

* 1. 创建测试脚本:自动录制机制能够将测试人员的每一个操作步骤及被操作的对象记录下来,自动生成测试脚本语句,还将被操作对象及相应的动作按照层次和顺序保存在一个基于表格的关键字视图中.
  2. 增强测试脚本功能:允许测试人员通过在脚本中增加或更改测试步骤来修正或自定义测试流程,
  3. 运行测试:从脚本第一行开始逐行执行语句运行过程中会对设置的检查点进行验证用实际的数据代替参数值,并给出形影的输出结果信息.
  4. 分析测试结果;运行结束后紫铜会自动生成一份详细完整的测试结果报告.

测试对象管理机制:

录制脚本时会自动获取被操作对象的一系列属性信息,然后将对象连同它的属性信息存在自己的对象仓库中.

每一个测试都对应着至少一个对象仓库,测试人员在新建或者打开一个测试的同事,也加载了相应的对象仓库.

按照关键字视图中的顺序回访脚本时,首先会在对象仓库中找到对象,然后根据对象仓库中描述的对象属性,在被测应用中定位到该对象,然后执行相应的动作.

如果运行脚本一个对象的一个或者多个属性发生了变化,也就是说对象仓库中的属性和被测应用中的属性不一致了运行测试结果就会失败.

为了实现成功的测试,测试人员就要经常维护自己的对象仓库,调整对象仓库中的对象属性,使之与被测应用一致.

总结:

创建测试:

        获取被操作对象的属性信息

* 1. 使用唯一的对象名在对象仓库中记录该对象.
  2. 将对象的全部属性信息存放在数据仓库中
  3. 标识关键属性信息
  4. 在脚本中记录对象名称和相应的动作

运行测试:

* 1. 从脚本中获取对象名称
  2. 在对象仓库中定位对象,并获取其关键属性
  3. 根据关键属性信息在被测应用中定位对象
  4. 根据脚本中录入的动作执行相应的操作.

对UFT的理解:

这个工具是一个能帮助我们提升效率的工具,把大量的手工测试的动作或者工作,交给工具去通过相应的测试脚本的方式,让工具去完成,提升了效率,减少了工作量.

这个工具的第一个特点是基于关键字驱动的测试脚本,第二个是拥有一个胖到的对象仓库,对象仓库的作用是在工具开启前就选择要用到的插件,选择插件的同时就意味着 加载了对应的对象仓库,有了对象仓库就可以录制该类型的软件的测试脚本,有了对象仓库后就可以录制同样类型的被测软件的所有操作,将被测软件中的控件看成一个一个的对象,单独的对象,给这个对象起一个关键唯一的名字,并且将它的全部属性,拿到保存到对象仓库中,是当前脚本专有的对象仓库中,并标识出其关键属性.所以在这个脚本中记录了我对控件(对象)做了什么操作,这样就完成相应的录制.

在脚本回放的时候,首先会通过对应的脚本中的名称去对象仓库中定位对象,找到对象的关键属性后,再到被测软件里面找具体的控件,定位好控件后,加载脚本中录制好的相关的操作,这就完成脚本的回放.

报错:

一:录制回放时关键属性值不一致就会报错

二:回放过程中找不到或发现之前没有识别的对象也会报错.

检查点:

用于验证应用程序是否在预定的状态上.

* 1. 标准检查点(standard checkpoint)

检查标准对象的属性信息,如某个复选框是否被选中.

* 1. 位图检查点(Bitmap checkpoint)

检查图片对象的属性信息,如某个图片的来源文件是否正确

* 1. 文本检查点(Text checkpoint)

检查网页或者窗口上面的文字信息是否正确

* 1. 可访问性检查点(accessibility checkpoint)

检查网页是否满足w3c的网页内容访问性指南的要求

* 1. 数据库检查点(DataBase checkpoint)

检查数据库的内容是否正确.如检查数据库的查询值是否正确.

检查点的添加有两种方式,一种是录制时添加,另一种是录制后添加(可访问性检查点)

超时等待:因为在运行脚本时,速度非常快,如果不做等待检查点会直接报错.页面加载慢,运行速度快,这种情况并不是脚本的问题,所以要设置一下等待时间.

检查点的插入位置取决于待检查的内容是否依赖于当前步骤.如果依赖必须放在当前步骤之后,如不依赖,前后都可以.

图片检查点:

检查图片是否一致,颜色是用RGB度量,检查的是数值.

文本检查点:

四个选项框,完全匹配:内容一致是通过,如果不勾选目标值包含期望值的内容时通过,不包含不通过.

还有区分大小写,忽略空格,未显示文本(取反,也就是不显示.)

数据库检查点:

手工添加数据源.数据源(数据库应用程序,或者数据库服务器)

添加一个数据库检查点有两种方式,第一种是需要下载微软的一个工具,还有就是手动指定SQL语句,next  创建选择需要的数据源,确定 输入SQL语句  next  选择字段,设置预期值,run.

数据库检查点可以做到和被测软件无关,也就是可以独立出来.

可访问性检查点:

检查的是属性,alt web页面中录制前,在tool下选择然后设置.

参数化:

就是一张包含多组数据的参数表,使之替换录制的脚本中相应的常量值,这样在脚本运行的时候就会使用参数表中的数据来代替录制的常量值.PS:参数化就是常量变量化.

脚本中使用数据的建议:

* 1. 将之前未产生过问题的数据放在检查数据中的前部,产生过缺陷或问题的数据放后部.
  2. 将正向测试的数据放在待检查数据的前部,反向测试的数据放在后部.

数据可以放在Global或者当前action表中,其区别是全局变量和局部变量.

输出值(output value):

建立输出值的过程就是让uft在执行脚本的时候,把一些执行结果中间结果数据输出到DataTable 中,查看输出值与检查点相比,是另一种测试形式,建立输出值不但是为了查看测试结果,这个运行时输出结果还可以作为参数,重新运行与脚本中.

输出值类型:

* 1. 标准输出值:输出标准对象的属性值
  2. 文本/文本区域输出值:输出窗口或者web页面中的文本信息.
  3. 数据库输出值:输出数据库的查询结果.
  4. xml输出值:输出xml文档的元素或属性信息.

注意:并不是所有的应用都支持上面列出的输出值,不同的应用环境支持不同的输出值.

验证码的测试;

* 1. 把相应的验证码输出到数据表中去.
  2. 真嘎验证码输入框做参数化,(数据来源就是数据表中存的验证码)

Action(动作):

    它是一系列操作步骤的集合,在uft中,允许将整个测试脚本分成多个action 每个action就是一个相应独立的逻辑单元,而一个测试脚本就是由顺序调用每个action的操作构成的.

action作用:

用action对一些相对凌乱的步骤进行分组,使整个脚本由功能相对独立的action顺序(或层次)组成,非常便于脚本的阅读和维护.

 当其他脚本需要本脚本中的一些步骤时,可以讲这些步骤组成一个action,然后由其他脚本调用,这个action来重复这些步骤.

在data table中,每个action拥有自己的sheet,测试人员,可以选择让action使用自己的sheet还是使用共有的sheet.

当脚本中的某些步骤需要参数化,可以将这些步骤组成一个action然后对这个action实现参数化,也就是仅让这个action反复执行,而不让所有脚本都反复执行.

action的种类:

* 1. 不可复用action-non-reusable action

 只能被本测试脚本调用的action,其他测试不能使用.

* 1. 可复用action-reusable action

可以被其他测试多次调用的action,在其他测试中以只读的方式使用,只有被调用的测试(即原测试)才有权限修改.

* 1. 外部action-external action

调用其他测试中的action,该action还是存储在被调用测试中,本地只能以只读方式查看.