

# The Design and Implementation of Software Testing System

Jiang Wen<sup>1</sup>, Zheng Ming<sup>2</sup>, and Li Hua<sup>3</sup>

School of Physics and nuclear energy engineering, Beihang University, Beijing, China 100191

1. jwxh1989@163.com, 2. zhengming@buaa.edu.cn, 3. lihua@buaa.edu.cn

**Abstract**—Testing and evaluation of software is an key work in the software development process. This article introduces a powerful Software Testing System from the practical point of view. This system can create the test routes and replenish test items automatically after users set relevant parameters as they wish, and finally it will derive test cases. This system has automatic error detecting function, regression test function and document output function and so on. The C/S architecture, C# language and .net framework 4.0 is used in developing this system. The system's database is SQL Server database and the technology of WPF is used for the pages of this system. The accuracy and the degree of automatic of software testing and evaluation will be highly improved by using this Software Testing System.

**Keywords**—software testing, C#, WPF, SQL server

## 软件测试系统的设计与实现

姜文 郑明 李华

北京航空航天大学物理科学与核能工程学院, 北京, 中国

**摘 要** 软件的测试和评估是软件开发过程中的一个关键环节。本文从实际应用角度出发,介绍了一套功能强大的软件测试系统。该系统可根据用户需求设置相关参数,自动生成测试路径并且补充测试项,最后生成测试用例。该系统自带错误检测功能,以及回归测试和文档输出等功能。系统采用 C/S 架构, C#语言编写,以.net framework 4.0 为开发平台,数据库采用 SQLServer 数据库,界面编写采用 WPF 技术。测试人员通过该测试工具对目标软件进行测试,提高了测试评估的精确度和自动化程度。

**关键词** 软件测试, WPF, 数据库 (SQL Server)

### 1. 引言

软件测试工具可以简化测试人员的工作,它将一些重复的手工测试变成由计算机自动完成的工作,这样既减轻了测试人员的负担并提高了工作效率,又使测试的准确度大大提高。目前国内还没有一款完全满足所有软件的测试工具。而对于测试人员来讲,拥有一款通用的测试工具,可以根据自己的需求进行测试,更加人性化地为测试工作服务,就变得尤其重要。

设计和开发该测试系统的主要目的是为测试提供方便,预期的工作过程是:测试人员通过用户端从数据库导入测试脚本库和变量库,在绘制结构图和测试结构图后,设置相关功能联系并设置测试脚本和变量,该测试系统对测试脚本进行分析组合生成测试项,然后测试人员设置参数,最终生成测试实例,将实际结果导入和预期结果作比较,

最后生成测试文档。

### 2. 系统设计

#### 2.1 系统流程设计

在用户进入编辑界面,也就是主界面后,应该主要有三大功能模块:工具栏,属性栏,画图工具区,画图编辑区,错误提示区。用户首先利用工具栏里提供的控件进行编制,包括编制模式,功能框架,然后编制功能类型,最后编制功能实例。编制完成后进行错误检测,再确认没有错误后让该系统自动生成测试组合即测试项。然后由用户对每一个测试项进行编写或者直接导入测试用例。系统提供自动对比功能,将实际结果和预期结果进行比较,最后生成测试文档。系统需带有回归测试的功能。

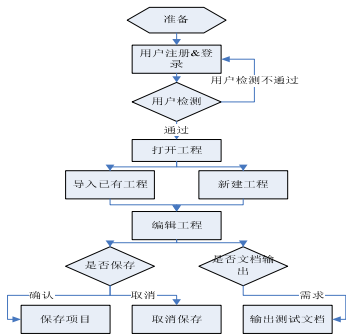


图1 系统流程图

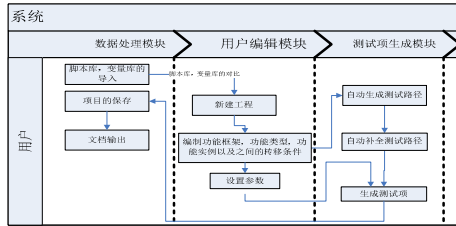


图2 系统功能框架

## 2.2 采用的设计模式

在该测试工具系统的设计中，主要开发了三个大的类：数据库类，数据操作类，绘图工具类。还包含若干小的结构体或者类，这些结构体或类用于记录临时数据。这三个大类之间的关系是，用户使用绘图工具类编辑，一些小的结构体或类负责记录这些数据，这些小的结构体或类与相应的控件联系，用于显示一些数据，数据处理类就负责将这些小的结构体或类中的有效数据提取出来转化成一定格式的文件如 xml，数据库类负责将这些数据结构体存入到数据库中和从数据库中取出这些数据。数据操作类在该项目中起到了承上启下连接用户和数据库的桥梁作用，是项目中至关重要的一环，采用设计模式中的开闭原则设计如图 2 所示，在数据操作类中，不是把所有的属性和方法写在这个类中，而是将链接，导入，保存，操作这四个功能分开，使用四个类分别实现，这样如果哪个功能的设计开发出现问题就可在这个功能的类中修改而不用查找整个数据操作类。

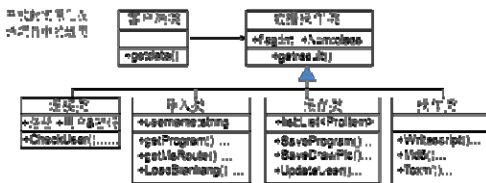


图3 开闭原则的应用

## 2.3 数据库设计

该测试工具采用 c/s 模式，将相关数据存储在服务器端的 MS SQL Server 服务器上。数据库设计了 12 张数据表，

分别存储脚本参数，绘图数据，文件名，模式路径，功能框架，项目基本信息，实例路径表格，模式转移路径表格，用户信息，变量库，脚本库，原始需求库。并设计了 11 个相关的存储过程，创建项目模板，删除所有，寻找指令，获取项目，获取项目名称，载入变量，载入脚本，载入原始需求，新建项目，重测试，保存项目[4]。

## 2.4 界面设计

该系统界面设计采用 WPF 技术，在使用 wpf 的技术中使用的比较多的两个很重要的控件，datagrid 和 treeview[2]，Datagrid 是数据显示成表格形式的控件，而 treeview 是数据显示成层次树结构的控件[3]。

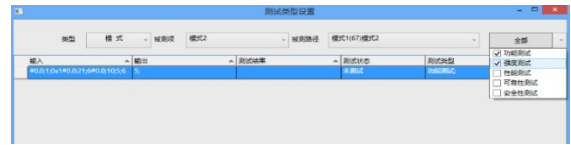


图4 datagrid的应用



图5 treeview的应用

```

1: <treeview Name="TreeView1" Width="250" Padding="5" Margin="5" BorderThickness="1">
2:     <treeview.ItemTemplate>
3:         <HierarchicalDataTemplate DataType="{x:Type local:PropertyModelItem}" ItemsSource="{Binding Path=Children}">
4:             <StackPanel Orientation="Horizontal">
5:                 <Image VerticalAlignment="Center" Source="{Binding Icon}" Width="30" Height="30" Margin="0,0,2,2"/></Image>
6:                 <TextBlock VerticalAlignment="Center" Text="{Binding DisplayText}"></TextBlock>
7:                 <Image VerticalAlignment="Center" Source="{Binding EditIcon}" Margin="2,0,0,0"/></Image>
8:                 <StackPanel.ToolTip>
9:                     <TextBlock VerticalAlignment="Center" Text="{Binding Name}" TextWrapping="Wrap" MaxWidth="200">
10:                     </TextBlock>
11:                 </StackPanel.ToolTip>
12:             </StackPanel>
13:         </HierarchicalDataTemplate>
14:     </treeview.ItemTemplate>
15: </treeview>

```

图6 WPF的编写

## 2.5 算法设计

在该系统中使用比较多的算法是递归算法，该算法使用在模式内路径的生成，框架内转移路径的生成，自动生成测试项，设置脚本参数生成测试用例等方面。递归算法是一种直接或者间接地调用自身算法的过程。在计算机编写程序中，递归算法对解决一大类问题是十分有效的，它往往使算法的描述简洁而且易于理解[1]。

由于在该测试工具中需要多次使用排列组合，因此需要多次使用递归算法。如图 7 所示：这段程序的目的是自动生成测试路径：



图 7

递归算法在该项目中的应用

## 2.6 其他技术

该系统中使用了 linq, 泛型, md5 加密等技术。

## 3. 功能介绍

(1)登录。用户通过设置账号和密码进行注册, 注册后的用户就可以通过账号和密码进行登录。点击服务器设置, 出现服务器设置的相关信息, 输入服务器的 IP 后就可以和指定的服务器进行连接。

(2)工程管理。系统首先进入打开工程界面, 该界面提供给用户最近的一些工程名称, 用户可以点击后直接打开最近的工程。用户可以新建工程, 还可以打开在某个位置的指定项目文件。界面的右侧是帮助文档, 方便用户查找相关功能和信息的介绍。点击查看可以看到相关脚本, 变量, 需求的信息以表格的形式显示出来。系统还提供变量检测, 脚本检测, 需求检测的功能, 对比新旧不同的变量, 脚本, 需求, 用户根据实际情况增删改。

(3)项目编辑主界面。该界面是用户进行项目编辑的主界面, 上面的图标功能从左到右以此为新建, 保存, 打开, 复制, 剪切, 粘贴, 调整样式, 操作, 设置, 工具。用户在右边的工具箱里先选择模式工具, 将模式图标拖到中间的显示区显示。将多个模式进行编辑。编辑完成模式后, 系统在左边的资源管理器中会自动生成相关模式。点击某个模式进入框架界面, 框架编辑界面中和模式编辑基本相同, 也是采用拖曳图标和连线法。编辑完成后可以在左边的资源管理树中右键点击一个节点, 选择自动生成, 就会出现若干条测试项。用户通过对需要测试的结构的理解, 可以在该工具的环境下通过拖曳自主组合测试实体和联系, 用圆形框代表测试实体, 单向箭头代表联系, 圆形框可以改变大小, 单向箭头可以改变长度, 起始点和终结点。圆形框和箭头都有个属性框, 其中属性可以由用户自己修改和设置。在圆形框上存在着若干个对称的锚点, 这些锚点为单向箭头提供了停靠处。用户在完成结构配置后需要对测试实例设置测试脚本和测试变量等属性, 还需要对转移条件设置时间戳和转移变量。如图 9 所示:

从一个功能实例到另一个功能实例之间的转移可以由用户自己采用拖曳的方式将线的起始点和终止点停靠在实

例的锚点上, 同样框架之间的联系也是采用这种拖曳线的方式完成。在同一个功能框架下, 不同功能类型中的实例需要自由组合形成测试项, 例如功能类型 a 中含有两个功能实例, 功能实例 1 和功能实例 2, 分别标记为 a1 和 a2, 其中功能实例 1 是起始实例, 用红色标出。功能类型 b 中含有三个功能实例, 功能实例 5, 功能实例 6, 功能实例 7, 分别标记为 b5, b6, b7, 功能实例 5 是起始实例, 用红色标出。而功能类型 a 到功能类型 b 存在一条转移路径。如图 8 所示:

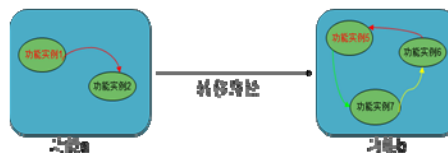


图8 功能演示图

那么由此测试工具自动生成六个基础测试项: a1-b5, a1-b6, a1-b7, a2-b5, a2-b6, a2-b7, 这是第一步。

第二步, 自动补全测试项: 即补全在同一个功能类型中的从起始实例到该实例的路径, a1-a2,

a5-a7-a6, a5-a7-a6-a5-a7-a6.....

a5-a7, a5-a7-a6-a5-a7.....

这一步非常重要, 因为会出现有环的情况。如上图所示, 功能类型 b 中含有环, 这样从起始实例 5 到终止实例 6 存在的路径就有 a5-a7-a6, a5-a7-a6-a5-a7-a6, .....无限循环下去就会有无数条路径, 此时只能规定路径步数, 假设在该功能类型中功能实例有 n 个, 那么最大路径步数为 n\*2, 这样就可以保证环能至少走一遍。

第三步, 自动组合前两项, 补充完整生成一个完整的测试项: 例如上图中 a1-a2---a5-a7-a6 即为一条完整的测试项。如上图, 所以的测试项应为: a1-a5, a1-a5-a7-a6, a1-a5-a7-a6-a5-a7-a6, a1-a5-a7, a1-a5-a7-a6-a5-a7, a1-a2-a5, a1-a2-a5-a7-a6, a1-a2-a5-a7-a6-a5-a7-a6, a1-a2-a5-a7, a1-a2-a5-a7-a6-a5-a7。

第四步, 根据前面设置的变量和脚本, 生成测试项 list, 例如 a1-a5-a7-a6 这条测试项, a1 测试实例中含有原来编辑和设置的测试脚本, 这样得到一系列的测试项 list, 这些 list 由测试脚本和变量组成。

第五步, 设置脚本参数, 生成测试用例: 对于生成后的测试项用户可以设置参数来生成不同的测试用例, 例如 #0.0;23;xxx;xxx 对前一个变量 xxx 设置参数 23, 21, 后一个变量 xxx 设置参数 67, 55, 这样该测试工具可以对这些测试项进行排列组合, 生成四个测试用例 #0.0;23;23;67, #0.0;23;23;55, #0.0;23;21;55, #0.0;23;21;67。

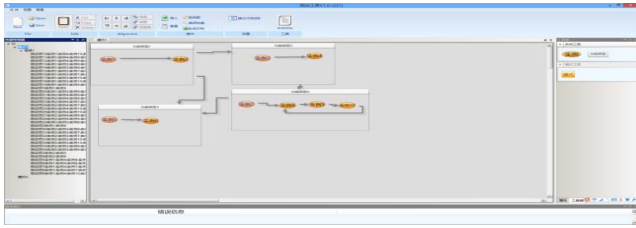


图9 测试项的生成

(4)功能路径转移设置。右键资源管理树中一个节点选择功能路径设计或测试项路径设计进入到路径设计界面，在该界面中最左边显示测试项的名称，点击某个就会在中间出现测试项的实例联系路径，方便用户直观的查看，最右边的就是路径设计的表格，初始情况下的测试路径为测试项中转移条件的排列组合。用户可以根据实际的测试需求进行增删改。

(5)测试项路径设计。这是在完成上步操作后对一条测试路径的进一步设计，用户可以根据实际情况进行测试的组合。如图 10 所示：

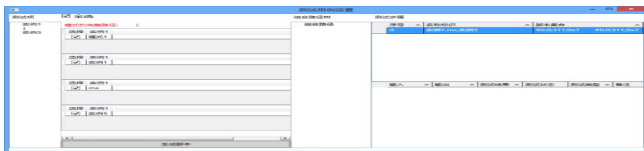


图10 测试路径的设置

双击测试步骤进行参数的设置，条件覆盖自动生成后会在下面的表格中显示测试的实际情况。左边的脚本信息记录这脚本和脚本的说明信息。

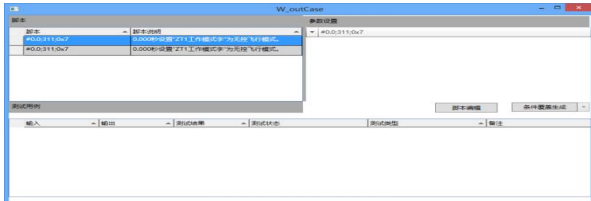


图11 参数的设置

脚本编辑及时参数的编辑，脚本中的大部分是些变量，实际测试中会给出一些测试值来进行测试。点击条件覆盖后就会生成相应的测试用例。

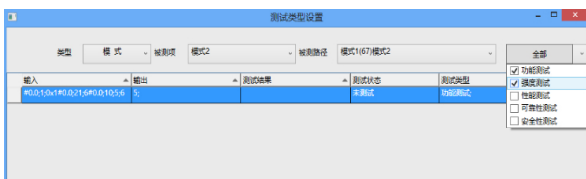


图 12 测试项的显示

(6)系统图是用户根据测试的实际情况设置的一些外部

设备以及接口，方便以后为数据的输入做准备。

(7)属性。模式，框架，功能类型，测试实例，转移条件都是有各自不同的属性的，用户可以对此进行详细的修改和设计，例如下图中，是对测试实例的属性描述。

(8)查看信息，用户在这里可以看到和编辑整个测试的状态和相关的信息。

## 4. 总结

该测试软件系统基本完成了测试人员对于一个软件测试工具的需求，减轻了测试人员的工作量，提高了测试人员的工作效率。本测试软件系统主要实现了五个方面的功能，第一，实现了多用户，多项目，以及项目保密功能。第二，实现了拖曳式操作功能，改变了原来由测试人员手工录入测试信息的方式。第三，实现了由用户编辑完后该测试软件系统自动生成测试路径和测试项的功能。第四，实现了错误检测功能。第五，实现了脚本的输出和测试结果和测试文档的输出。

该测试软件系统界面设计简约，可操作性强，数据和层次的显示清晰明了，并具有动态效果。该测试软件系统在安全保密方面也考虑得比较周全，不仅用户的安全，脚本库，变量库的数据库安全采用 md5 加密方式，而且编码内部的安全都采用 md5 加密方式，安全可靠。

经相关测试人员多次测试使用，该测试软件实现了上述的各个功能，在测试过程中使用稳定可靠，达到了预期的设计目标。

## References

- [1] WangXiaoDong. "Design and analysis of computer algorithms". BeiJing: Electronic Industry Press, 2001: 6-11
- [2] Lori "A MacVittie.XAML in aNutshell". O'Reilly, 2006
- [3] Eric Butow. "User Interface Design for Mere Mortals. Addison"-Wesley Professional, 2005
- [4] JiangYi Hu. "The performance optimization of database application system of SQL based on Server ". Computer engineering and Applications, 2001, 37(2): 95-97.