****

**毕业设计（论文）中期报告**

**题 目：软件可靠性模型性能综合评测系统**

**专 业 软件工程**

**学 生 郭睿**

**学 号 131110206**

**班 号 1311102**

**指导教师 张策**

**日 期 2017.04.29**

目录

[1. 论文工作是否按预期进行、目前已完成的研究工作及结果 3](#_Toc481509630)

[1.1 论文工作是否按预期进行 3](#_Toc481509631)

[1.2 目前已完成的研究工作及结果 3](#_Toc481509632)

[2. 后期拟完成的研究工作及进度安排 12](#_Toc481509633)

[2.1 后期拟完成的研究工作 12](#_Toc481509634)

[2.2 后期进度安排 13](#_Toc481509635)

[3. 存在的问题与困难 13](#_Toc481509636)

[4. 论文按时完成的可能性 13](#_Toc481509637)

[5. 参考文献 13](#_Toc481509638)

# 论文工作是否按预期进行、目前已完成的研究工作及结果

## 论文工作是否按预期进行

首先，再次明确本SRGM评测系统的应用目标：在完成研究准备工作的基础上，进行本系统的设计和实现，最终达到通过本系统展示、比较不同SRGM在不同失效数据集上建模描述测试过程的差异，帮助初学者更直观、更具体的了解不同模型的测试过程和差异所在；也为研究人员验证、比较模型性能，从而验证得出性能更优的模型，和软件工程师在实际工程中针对特定的失效数据集选择相应的SRGMs提供一定的参考和便利。

目前，根据之前确定的需求分析、研究方法和技术路线，已经按预期完成了相应的工作。

在理论研究方面，通过阅读软件可靠性增长模型的有关文献，结合数学分析的知识，大体上理解了SRGM模型的原理和不同模型之间的区别，明确了现有失效数据集的含义及其所适用的范围，熟悉了在不同的数据集上验证不同模型的相关流程，并且已在matlab上实现了相关的验证流程、绘制出了拟合、预测曲线、计算出了评测的数值。此外，还能够将多个模型的拟合曲线绘制在一起予以比较。

在软件开发方面，根据V字开发模型，进行了系统的详细设计，完成了类及其方法的设计、数据库表的详细设计；同时，认真听取并采纳老师的建议，系统采用更符合实际需求的C/S架构。在详细设计的基础上，进行了系统的代码实现，目前为止除了在软件可靠性模型与失效数据集的匹配及模型的导入上存在一点小问题，系统选择、绘图、求值、比较、导入、管理模块的基本功能都已实现。

## 目前已完成的研究工作及结果

下面将大体上根据研究工作所进行的顺序，分详细设计和代码实现两个部分介绍目前所完成的有关工作及其相应的结果。

一、详细设计部分

* 类的设计：

1.类之间的层次关系：根据概要设计中的ERD体现的实体之间的联系，结合业务逻辑中的各种功能模块，实现如下所示的类图。



Figure 1‑1类图

2.类之间的逻辑关系：类之间的逻辑关系，体现在各个功能实现过程中，信息的传递流程，通过时序图来表示，以下为各个用例的时序图。



Figure 1‑2注册&登录

上图是用户注册和登录的时序图，此处以SRGM研究人员为例，对于软件工程师同样适用，而系统管理员只存在登录的时序。用户在界面输入相应的用户名和密码，来完成注册和登录。



Figure 1‑3选择模块

上图为选择模型和数据集的时序图，此处以SRGM初学者为例。首先，系统界面将可选的模型和数据集呈现给用户，之后用户可以选择先选模型or先选数据集，然后在选择与前一项相匹配的后一项，最后确认选择。



Figure 1‑4演示子系统（除选择模块）

上图为演示子系统的时序图，其中除了求拟合参数外，其他时序顺序可以互换。该时序图中涉及几乎整个系统中最关键的方法。



Figure 1‑5导入&删除数据集

上图为导入&删除数据集的时序图，此处以软件工程师为例，对于同样SRGM研究人员同样适用。



Figure 1‑6导入&删除模型

上图为导入&删除模型时序图。



Figure 1‑7管理模块

上图为系统管理员管理用户和系统自带模型、数据集的时序图。其中，对模型的管理与对数据集的管理相同，此处以模型为例。

* 数据库表的设计：

数据库名称：srgm

支持的软件：SQLserver2005~2012；VS2013

数据库表的定义：根据概要设计中的ERD进行拆分，将相应的多对对关系拆分为一对多关系，建立相应的中间表，并进一步对数据库中的表进行定义，用以规范建立数据库的表格，具体定义如下:

Table 1‑1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 软件工程师表 | | | | | | |
| 数据项 | 列名 | 数据类型 | 长度 | 主键 | 允许空 | 默认值 |
| 用户名 | username | char | 8 | 是 | 否 |  |
| 密码 | password | Varchar |  |  | 否 |  |

Table 1‑2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SRGM研究人员表 | | | | | | |
| 数据项 | 列名 | 数据类型 | 长度 | 主键 | 允许空 | 默认值 |
| 用户名 | username | char | 8 | 是 | 否 |  |
| 密码 | password | Varchar |  |  | 否 |  |

Table 1‑3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 系统管理员表 | | | | | | |
| 数据项 | 列名 | 数据类型 | 长度 | 主键 | 允许空 | 默认值 |
| 用户名 | username | char | 8 | 是 | 否 |  |
| 密码 | password | Varchar |  |  | 否 |  |
| 姓名 | name | Varchar |  |  | 否 |  |
| 手机号 | phonenum | char | 11 |  | 否 |  |

Table 1‑4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 失效数据集表 | | | | | | |
| 数据项 | 列名 | 数据类型 | 长度 | 主键 | 允许空 | 默认值 |
| 名称 | dsname | varchar | 8 | 是 | 否 |  |
| 存放路径 | path | Varchar |  |  | 否 |  |
| 来源 | source | Varchar |  |  | 是 |  |
| 发布时间 | date | date |  |  | 是 |  |
| 操作权限 | permission | int | 1 |  | 否 | 0 |
| 创建者 | username | char | 8 |  | 是 | null |
| 类型号 | typeID | int |  |  | 否 |  |

注：permission为0表示系统管理员，1表示软件工程师，2表示SRGM研究人员。

username=null表示由系统创建，其等价于permission=0。

Table 1‑5

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集类型表 | | | | | | |
| 数据项 | 列名 | 数据类型 | 长度 | 主键 | 允许空 | 默认值 |
| 类型号 | typeID | int |  | 是 | 否 |  |
| 时间类型 | time | char | 1 |  | 否 |  |
| wt | wt | boolean | 1 |  | 否 | 0 |
| ct | ct | boolean | 1 |  | 否 | 0 |

Table 1‑6

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SRGM模型表 | | | | | | |
| 数据项 | 列名 | 数据类型 | 长度 | 主键 | 允许空 | 默认值 |
| 名称 | modelname | varchar | 8 | 是 | 否 |  |
| 存放路径 | path | varchar |  |  | 否 |  |
| 操作权限 | permission | int | 1 |  | 否 | 0 |
| 创建者 | username | char | 8 |  | 是 | null |
| 形状 | shape | varchar |  |  | 是 |  |
| 类型号 | typeID | int |  |  | 否 |  |

Table 1‑7

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 模型类型表 | | | | | | |
| 数据项 | 列名 | 数据类型 | 长度 | 主键 | 允许空 | 默认值 |
| 类型号 | typeID | int |  | 是 | 否 |  |
| 类型名 | name | varchar |  |  | 否 |  |

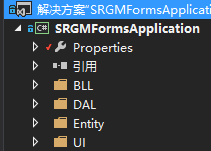
Table 1‑8

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 参数初值表 | | | | | | |
| 数据项 | 列名 | 数据类型 | 长度 | 主键 | 允许空 | 默认值 |
| 编号 | ID | int |  | 是 | 否 |  |
| 模型名 | modelname | varchar | 8 |  | 否 |  |
| 数据集名 | dsname | varchar | 8 |  | 否 |  |
| 参数初值 | value0 | varchar |  |  | 是 |  |

二、代码实现部分

* 系统的搭建

本系统是采用c#实现的C/S架构的桌面程序，将该系统分为以下几层，BLL：业务逻辑层，DAL：数据库访问层，Entity：封装、传递数据的实体类，UI：负责系统界面交互和数据、图片的显示。



* matlab部分的代码实现及DLL文件的生成

根据之前matlab中模型验证的有关工作，自行根据数据集的特点对其进行简单的分类：

1.不含测试工作量+不含变化点。

2.含有测试工作量+不含变化点。

3.含有测试工作量+含有变化点。

参考文献中模型分类的明确规定：

1.完美模型

2.ID

3.TEF

4.ID+TEF

5.ID+TEF+CP

观察和理解不同模型在不同数据集上的验证过程，发现：

1. 1、2类模型在1类数据集上的验证方法相同；

2. 3、4类模型在2类数据集上的验证方法类似；

3. 5类模型在3类数据集上的验证方法类似。

根据以上分析，联系matlab部分的相关代码，分别完成以上3类验证过程的实现，3种方法的大致流程如下:



Figure 1‑8

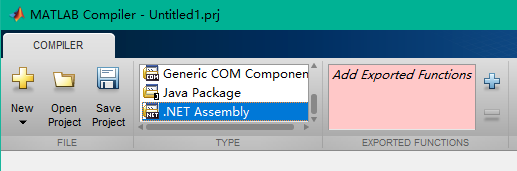
因模型和失效数据集的类型不同，其传入的参数和具体实现细节有所不同。以下为三种验证过程的函数：

T1MnDSn(modelName,dataSetName,xmd0String)

T2MnDSn(modelwtName,modelmtName,dataSetName,xmd0wtString,xmd0mtString,xmd0Num)

T3MnDSn(modelwtName,modelmtName,dataSetName,xmd0wtString,xmd0mtString,xmd0Num,cp)

完成了matlab中验证过程的代码，需要将其生成为DLL，才能被c#项目所调用，下面介绍生成DLL文件的具体操作：



1.在matlab的command window中输入deploytool命令

2.选择Library compiler来生成共享库和组件

3.选择Type：.NET Assembly

4.在Exported functions 中添加所编写的.m文件

5.命名DLL文件名为Matlab，类名为：MnDSn

6.在Files required for your library for run 中添加所要处理的模型文件

7.点击 Package ，等待生成DLL文件

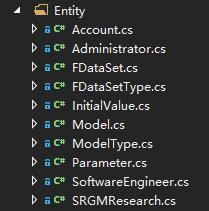
至此，生成的Matlab.dll文件中，含有一个类MnDSn,类中的方法即为你所添加的.m文件。

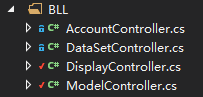
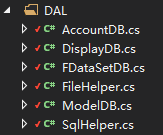
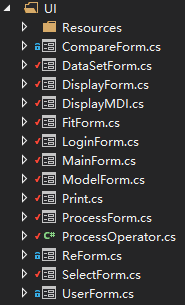
另外，因为本系统需要对不同模型在同一数据集的拟合结果进行比较，所以此时需要实现多个模型在同一数据集上拟合图的绘制，以更直观地显示出不同模型拟合曲线的差别。为了节省生成图像的时间，在matlab中，我直接采用拟合所生成的ymd来进行绘图，从而避免再次拟合所耗费的大量时间，其具体实现分为两类：

1、单文件模型（完美、ID） T1MnDSnMixD(modelNames,ymdpaths,dataSetName)

2、多文件模型（模型3、4、5）T2MnDSnMixD(modelwtNames,modelmtNames,ymwtdpaths,ymmtdpaths,dataSetName)

* 软件系统的代码所实现的功能





以上为在各个层中所编写的相应代码，其中entity中为实体类；BLL中的各个controller分别处理用户、数据集、展示、模型的相应业务；DAL中的DB分别处理用户、失效数据集、模型等在数据库中的相应操作，sqlhelper负责数据库的连接和一些基本操作，FileHelper负责处理文件的相关操作；UI中的各个窗口分别为：比较功能、系统（用户）数据集管理、演示功能、拟合、登录、主窗口、系统（用户）模型管理、打印、进度条显示、选择、用户管理功能的界面。*除了界面未做优化和用户模型的导入存在技术难点外，以上的各个功能的代码编写都已基本完成。*

matlab混合编程（c#中对matlab所生成DLL的调用）：

在c#项目中的引用中添加DLL文件，之后就可以在BLL层的Display Controller中调用DLL中类的方法，在调用时涉及到c#向matlab的数据类型的转化，因为matlab中是以矩阵形式来组织数据，所以必须通过MWArray类来进行转换。

# 后期拟完成的研究工作及进度安排

## 后期拟完成的研究工作

1.完成界面的美化和代码结构的优化，有选择性的进行代码重构。

2.解决系统必须引用DLL的技术难点，实现用户模型的导入功能。

3.增加模型和失效数据集之间匹配的自由度，即模型只要能在该失效数据集上进行验证，就允许用户进行相关的参数初值设置和验证。

## 后期进度安排

Table 2‑1 后期进度安排

|  |  |  |  |
| --- | --- | --- | --- |
| 起始时间 | 完成时间 | 计划工作内容 | 备注 |
| 2017-05-05 | 2017-05-07 | 实现用户模型的导入功能 |  |
| 2017-05-08 | 2017-05-11 | 增加模型和失效数据集之间匹配的自由度 |  |
| 2016-05-12 | 2016-05-15 | 界面的美化和代码结构的优化 |  |

# 存在的问题与困难

目前，有如下存在的问题和困难，需要解决：

1.c#项目需要添加DLL文件的引用后，才能调用DLL文件进行模型的验证。一个DLL文件在c#项目中相当于一个类，其中的方法即为添加的.m文件，而.m文件运行所需的文件（模型文件）需要在生成DLL的同时添加到其中。*所以，生成的DLL只能处理其生成时所添加的模型。当添加新的模型时，需要生成相应的DLL文件，而DLL文件需要被添加到引用，这是无法在系统内实现的。因此，需要找到无需添加dll文件引用即可处理新导入模型或考虑其他方法。*

2. 模型只要能在该失效数据集上进行验证，就允许用户进行相关的参数初值设置和验证。这里需要在一定限制条件（如：数据集格式）内，通过在matlab上编码来判别数据集的类型，从而决定模型能否在该数据集上验证，如果不能，需要给出不能验证的理由。因此，需要进一步加强对matlab的学习。

# 论文按时完成的可能性

在解决上述困难的思路可行的前提下， 能够大致按照后期进度安排，完成之后的任务。

# 参考文献

指导教师评语：

指导教师签字： 检查日期：