QuickAnalysisSys  
详细设计

# 文档概述

本文档适用于有一定软件开发经验的工程师或者系统维护人员，用于了解软件的详细设计思路和实现方法，从而指导代码开发和维护工作。

# 软件模块设计

系统详细设计，主要分为以下几个模块：

* 数据层设计
* 业务层设计
* 主界面UI设计
* 2D曲线绘制设计
* 3D模型绘制设计

# 数据层设计

本章节主要介绍了软件的数据层设计，包括模型文件与solution文件格式解析方式，数据层的基本设计思路及其接口设计。

## 模型文件

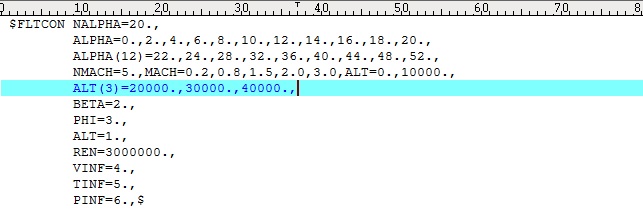
模型文件是快速分析系统的核心文件，用户根据该文件记录的信息进行solution计算，生成3D模型图。

模型文件是以dat为后缀的文件，使用FORTRAN代码数据格式（80A1）进行编码。

### 模型文件格式解析

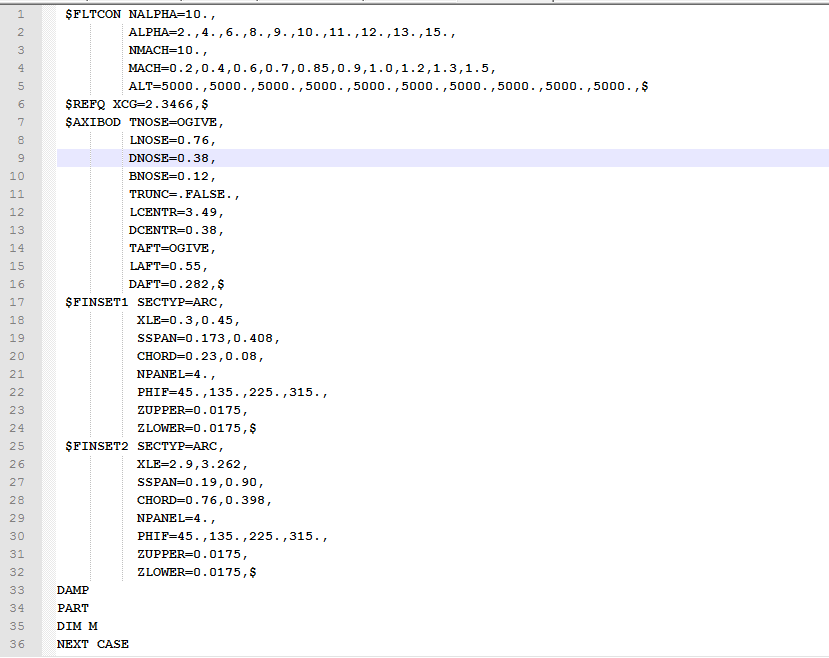
模型文件每行最多80列，包括符号均包括在内。每个子项以“$”作为开始和结束标志，每个变量必须以“，”作为变量值分隔符，如果有多个变量则会有多个“，”，每个模型文件结尾必须包含“NEXT CASE”字符作为模型文件结束标志。

单项子项数据超过80列，则在下一行加上数组下标继续，如下图“ALPHA”子项超过80列则如下图所示。



整个模型文件的输入数据可以是浮点型数据或者逻辑常量，所有的变量名都必须为大写英文字符，这对于“E”编辑符的数值同样适用，如REN=6.0E06是正确的，而REN=6.0e06则不对。

模型文件ANSI格式编码解析文本截图如下：



### 模型文件类图



## solution文件

solution文件是用户的程序计算后得到的若干结果文件，在模型文件在同一目录下的solution文件夹中。每个solution文件的文件名是计算结果名称。由于各业务上的差异性，solution文件的读取方式，解析可能会不同，为更大程度的方便用户扩展，这里使用工厂模式的方式，抽象出了曲线图的solution接口，更多的solution接口将在今后的开发中进行扩展。

### solution文件管理类图



# 业务层设计

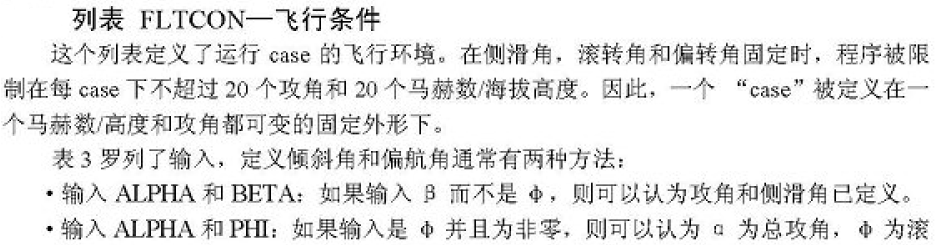
本章节主要介绍了软件的业务层设计，包括项目的树状结构，各子项的参数结构，业务层的主要功能是将模型文件中的数据组织起来形成一定规律的结构供UI层调用。

## FLTCON数据

FLTCON数据来自模型文件，读取模型文件数据层的ModelContent并解析，以下是FLTCON数据结构类图。



Checkdata用于检查当前数据信息并调整数据结构及设置程序计算条件，根据用户提供的业务描述，对变量进行检查，并根据当前用户输入数据对计算方式定义，多种计算方式可同时存在。





## REFQ数据

REFQ数据来自模型文件，读取模型文件数据层的ModelContent并解析，以下是REFQ数据结构类图。



## GEOMETRY数据

GEOMETRY包含两种数据类型AXIBOD与ELLBOD，GEOMETRY只能为其中的一种，根据模型文件解析后，得到GEOMETRY数据类型，以下是GEOMETRY数据结构类图。



## FINSET数据

FINSET数据下包含零到四个FINSETn子项，根据模型文件解析后，得到FINSET数据类型，以下是FINSET数据结构类图。



## CONTROL数据

CONTROL数据来自模型文件，读取模型文件数据层的ModelContent并解析，以下是CONTROL数据结构类图。



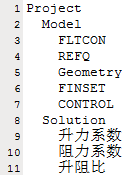
## DEFLCT数据

DEFLCT数据来自模型文件，读取模型文件数据层的ModelContent并解析，以下是DEFLCT数据结构类图。



## Project项目数据

项目树状结构如下图所示：



Project整合了模型文件与模型文件同目录下的solution文件夹中的文件，对Model中的数据进行了解析，并按照数据结构重新组织，供上层调用。

Model中涉及的四个必须的项：FLTCON、REFQ、GEOMETRY、FINSETn和CONTROL。

FINSETn：在一个FINSET下包含：FINSET1、FINSET2、FINSET3、FINSET4。根据模型文件中包含的项决定。

Solution中为SlotionMgr包含的solutionNames而定，每个子项对应一个solution文件夹中的文件。

以下为项目树类图结构。



Project通过解析ModelContent,生成不同的ModelItem(FLTCON,REFQ,GEOMETRY等)，在Project中这些数据对象都被抽象为ModelItem进行处理，并传递给UI层。

# UI层设计

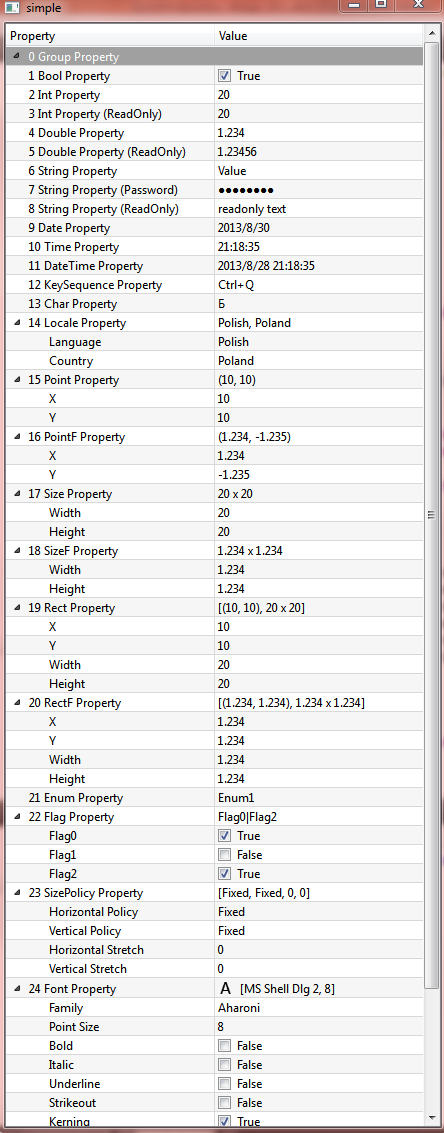
## 项目树

项目树采用QTreeWidget作为基类，使用QDockWidget包装后，交给MainWindow使用，项目树类图如下。



## 属性框

属性框使用开源项目Qt-solutions中的qtpropertybrowser为基础进行封装。属性在经过包装函数包装为QVariant交给封装后的qtpropertybrowser，即可显示在属性框中，属性框UI示例如下所示：



## 中央图形框

中央图形框分为两部分组成，使用QTabWidget作为基类，插入2D曲线图/3D模型图形框，和一个文本框显示文本文件内容，类图如下：



## 主界面

主界面使用QMainWindow作为基类，整合整个UI窗口，添加动作及菜单，下图为主界面类图。

# 2D曲线绘制

2D曲线绘制采用开源项目Qwt的QwtPlotWidget作为绘制基础，结合具体业务，绘制曲线，类图如下。

