

|  |
| --- |
|  |
| 工程计算模板开发系统环境概要设计说明书 |



|  |  |  |  |
| --- | --- | --- | --- |
| **文档类型：** | 项目文档 | **文档标识：** |  |
| **文档状态：** | [√] 草稿  [ ] 正式发布  [ ] 正在修改 | **模板标识：** | PERA/ZC/F011\_ZL\_2013 |
| **当前版本：** |  |
| **作者：** |  |
| **完成日期：** |  |

修改历史

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **版本** | **作者/修改人** | **完成日期** | **修改内容简述** | **审核** | **批准** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

确认签字

|  |  |  |
| --- | --- | --- |
| **单位** | **签字** | **日期** |
|  |  | 年 月 日 |
| 安世亚太科技（北京）有限公司 |  | 年 月 日 |

目 录

[工程计算模板开发系统环境概要设计说明书 1](#_Toc378240836)

[1. 范围 7](#_Toc378240837)

[1.1. 标识 7](#_Toc378240838)

[1.2. 系统概述 7](#_Toc378240839)

[1.3. 文档概述 8](#_Toc378240840)

[2. 引用文档 8](#_Toc378240841)

[3. 工程计算模板产品介绍 9](#_Toc378240842)

[3.1. 平台架构 9](#_Toc378240843)

[3.2. 模板开发环境简述 10](#_Toc378240844)

[3.3. 模板执行环境简述 11](#_Toc378240845)

[3.4. 模板库简述 11](#_Toc378240846)

[4. 模板开发 12](#_Toc378240847)

[4.1. 基本元素 12](#_Toc378240848)

[4.1.1. 数值类型 12](#_Toc378240849)

[4.1.2. 控件 16](#_Toc378240850)

[4.1.3. 联动映射 19](#_Toc378240851)

[4.1.4. 布局 20](#_Toc378240852)

[4.2. 公式 20](#_Toc378240853)

[4.2.1. 表现形式 21](#_Toc378240854)

[4.2.2. 六种基本函数 21](#_Toc378240855)

[4.2.3. 三角函数 22](#_Toc378240856)

[4.2.4. 数组函数 22](#_Toc378240857)

[4.2.5. 分段函数 23](#_Toc378240858)

[4.2.6. 其它函数 23](#_Toc378240859)

[4.3. 曲线 23](#_Toc378240860)

[4.3.1. 曲线定义 24](#_Toc378240861)

[4.3.2. 曲线查询 25](#_Toc378240862)

[4.4. 模板视图 26](#_Toc378240863)

[4.4.1. 设计树 27](#_Toc378240864)

[4.4.2. 属性视图 29](#_Toc378240865)

[4.4.3. 参数列表 29](#_Toc378240866)

[控件工具条 30](#_Toc378240867)

[4.5. 模板管理 30](#_Toc378240868)

[4.5.1. 新建模板 31](#_Toc378240869)

[4.5.2. 打开模板 32](#_Toc378240870)

[4.5.3. 保存模板 33](#_Toc378240871)

[4.5.4. 关闭模板 33](#_Toc378240872)

[4.5.5. 删除模板 34](#_Toc378240873)

[5. 模板执行 34](#_Toc378240874)

[5.1. 执行环境配置 34](#_Toc378240875)

[5.2. 执行过程 34](#_Toc378240876)

[5.3. 执行控制 36](#_Toc378240877)

[5.3.1. 一次性执行（确定） 36](#_Toc378240878)

[5.3.2. 反复执行（应用） 36](#_Toc378240879)

[5.3.3. 取消执行 36](#_Toc378240880)

[5.3.4. 执行顺序 36](#_Toc378240881)

[5.4. 日志 36](#_Toc378240882)

[5.4.1. 前台显示日志信息 37](#_Toc378240883)

[5.4.2. 日志文件 37](#_Toc378240884)

[5.4.3. 日志设置 37](#_Toc378240885)

[5.5. 结果处理 38](#_Toc378240886)

[5.5.1. 在当前视图显示结果 38](#_Toc378240887)

[5.5.2. 导出模板 39](#_Toc378240888)

[6. 库 41](#_Toc378240889)

[6.1. 库管理 41](#_Toc378240890)

[6.1.1. 库的查看 42](#_Toc378240891)

[6.1.2. 常用库管理 42](#_Toc378240892)

[6.1.3. 库的导入导出 42](#_Toc378240893)

[6.1.4. 管理库分类 43](#_Toc378240894)

[6.1.5. 新建库分类 43](#_Toc378240895)

[6.1.6. 删除库分类 44](#_Toc378240896)

[6.1.7. 管理库中的内容 44](#_Toc378240897)

[6.2. 库的分类 44](#_Toc378240898)

[6.2.1. 材料库 44](#_Toc378240899)

[6.2.2. 型材库 45](#_Toc378240900)

[6.2.3. 曲线库 45](#_Toc378240901)

[6.2.4. 公式库 45](#_Toc378240902)

[6.2.5. 标准件库 46](#_Toc378240903)

[6.2.6. 自定义库 46](#_Toc378240904)

[6.2.7. 模板库 46](#_Toc378240905)

[6.2.8. 实例库 47](#_Toc378240906)

[6.3. 使用库 47](#_Toc378240907)

[6.3.1. 引用库的视图 47](#_Toc378240908)

[6.3.2. 引用库的数据 47](#_Toc378240909)

[6.3.3. 引用库的视图和数据 48](#_Toc378240910)

[6.3.4. 冲突的处理 48](#_Toc378240911)

[6.3.5. 库的接口 48](#_Toc378240912)

1. 范围
   1. 标识

本文档在整个项目开发中标识号为：**XXX-XX-PERA-XXXX**。

本文档在整个项目开发中完整的标题名称为：工程计算模板开发系统软件设计文档概要设计。

（1）本文档在文字描述中的术语和缩略语约定如下：

**模板：**本文所指的模板是指工程领域要完成一个相对固化的细粒度计算过程的统称。该过程可以由界面输入或者选择计算元素，并由后台公式逻辑将结果完成后再输出到界面上。

**子模板：**子模板是模板的一部分，一个模板可能由几个子模板拼接并由公式逻辑进行适当的参数连接而成。子模板可以是一个独立的逻辑单元，也可以是一个不完整的逻辑单元。

**控件:** 与普通的编程控件不同的是，本文所指的控件是指界面元素，且不需要对控件进行任何编程，即可完成业务逻辑。

（2）本文档适用于工程计算模板开发系统软件项目的各个子系统及系统本身。

* 1. 系统概述

像航空航天、船舶、汽车等行业的复杂系统设计工作，由于涉及到的专业范围广，工作量大，型号研制周期紧迫，容易造成分析失误；同时，又面临很多必须做的重复性工作。借鉴国内外经验，模板是解决这些问题的有效工具。

模板（template）总是面向某一项具体工作或任务，创建模板的目的在于提升工作质量和效率。模板一旦创建完成，就可以源源不断地发挥作用，老型号能用，新型号也能用。尤其是模板中凝聚的一些宝贵的工程经验和方法，对知识的传承和交流也有益处。

以航空结构强度专业为例，很多工作流程和方法都可以提炼为模板，比如以下几个大的方面：

* 建模工作：有限元建模中的截面属性计算等；
* 数据处理：全机载荷计算结果后处理，试验数据处理，疲劳载荷谱等；
* 强度校核：临界载荷、工作载荷、剩余强度、尺寸优化等；
* 强度报告：载荷计算报告、强度校核报告、试验报告等；

模板可大可小，螺栓、接头、耳片的强度计算工作，甚至曲线插值工作都可以创建模板。

* 1. 文档概述

本设计文档给出系统的设计说明，包括最终实现的软件必须满足的功能、接口和用户界面以及设计约束等。

本设计文档的目的在于：

* 为编码人员提供依据；
* 为修改、维护提供条件；

本设计文档的预期读者包括：

* 项目开发人员，特别是编码人员；
* 软件维护人员；
* 技术管理人员；
  + 执行软件质量保证计划的专门人员；

1. 引用文档

* 基于仿真的综合设计产品介绍-V1。7-20130902
* 细节强度软件Aircraft SAT V0。4y
* 强度工程细节分析方法集
* Flexware概要设计文档
* 公式库设计文档
* Simworkflow功能点列表
* 工具封装总结
* 用Flexware实现AAA软件结构重量计算模块
* 结构设计引导-外翼翼梁受力分析设计引导封装说明
* PeraStrength产品介绍

1. 工程计算模板产品介绍

安世亚太综合设计系统PERA。WS工程计算模板开发环境是精益研发平台PERA的核心组成模块之一，是一套面向高端研发企业的支撑研发模板开发、管理和应用的平台。

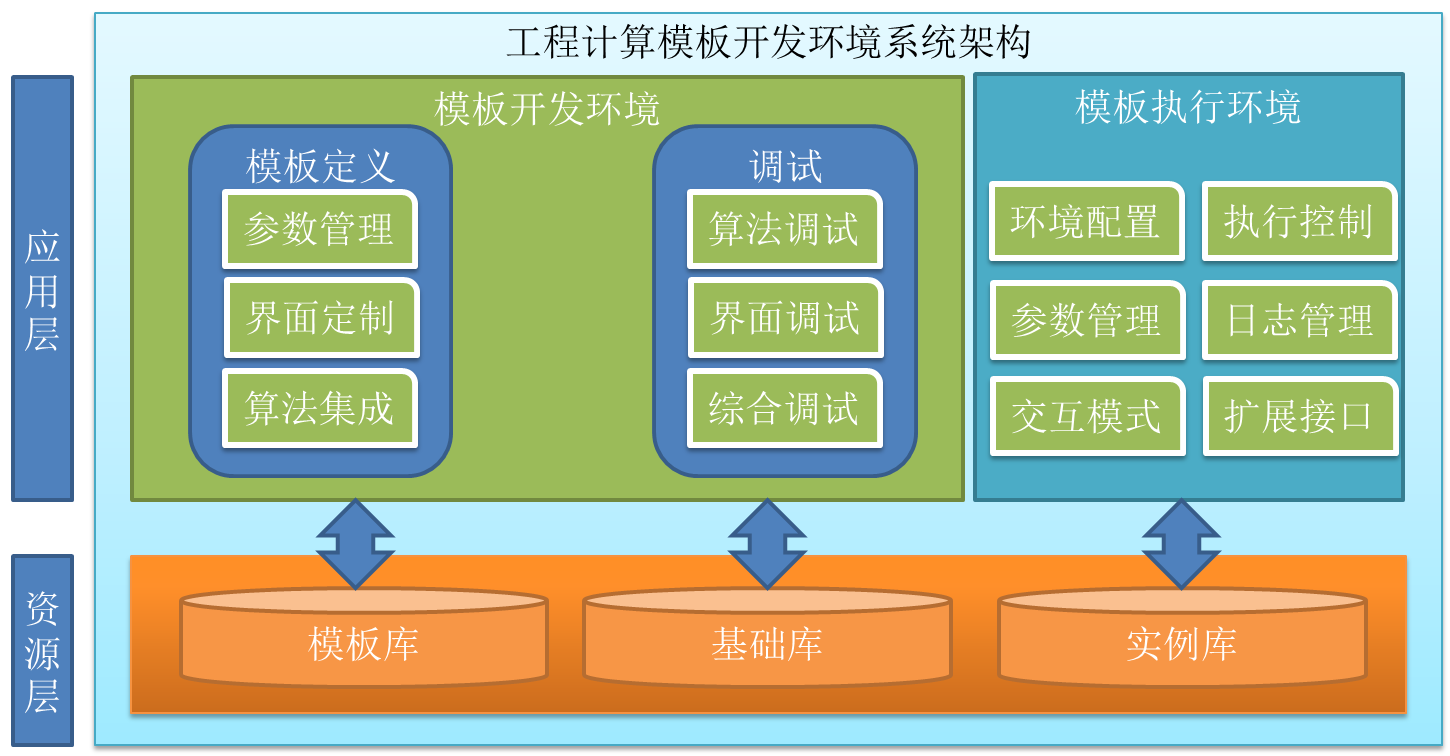
综合设计与仿真技术是现代企业实现高效高质产品研发的驱动力，通过全方位的协同设计，以及精确地建模来模拟真实世界，帮助企业优化产品性能，减少物理样机及试验，实现协同研发的高效模式，从概念设计到最终验证全过程驱动产品研发。

本文所涉及到的是工程计算模板开发和应用过程。

* 1. 平台架构

PERA。WS工程计算模板开发环境按照模板的生命周期逻辑分为模板开发、模板管理与模板应用三个层次，辅以基础库和模板库以及实例库作为资源支撑层。 该系统采用先进的软件用户界面开发技术与通讯技术，融合了业务专家和咨询专家的工程应用与开发经验，封装了各种仿真分析规范和分析方法，可定制开发出的一套快速工程计算模板系统。通过该系统，可以规范用户使用过程，减少用户机械化操作，缩短仿真分析周期，提高仿真分析效率，让用户将精力投入到产品的优化改进中，从而提高客户产品研发水平，进一步增强研发能力，对在研及未来产品的研制产生积极的技术创新与推动作用。

功能层次如下图所示：



PERA。WS工程计算模板具备以下特点：

* **易用性:** 通过免编程的手段来创建模式相对固定的模板，会大大减少科研价值较低的重复劳动，把精力放在真正的设计上来，提高设计效率，减少人为错误。
* **工程化:** 使用工程人员熟悉的业务情景和业务语言来对设计过程进行表达。
* **规范化:**将各行业的规范数据和规范封装到模板和基础库中，从而达到仿真的规范化;
* **标准化：**将工程人员熟悉的数据和标准通过元件库，物理属性库等进行封装，从而实现标准化；
* **开放性：**用户可以通过封装模板的形式，把自己的业务动态地增加到模板库中而无须手动编程，可满足绝大部分计算类需求；

**可扩展性：**构建虚拟的产品仿真设计系统，满足各种专业的需要，集成相关软件和工具，打通各自间的数据接口，实现仿真数据有效传递。

工程计算模板开发系统的架构类似文档处理器（生成、集成与整合）和分析程序编辑器的集成软件，目的是使工程师快速化、标准化完成计算分析，分析结果以用户的标准格式输出为计算分析报告。

* 1. 模板开发环境简述

设计模板是由两部分组成，界面层和业务逻辑层(算法或模型)。界面层由设计控件通过拖拽构成，如Tab页，表单，曲线，参数组等组成。用户将需要关注的输入输出元素以设计模板UI层面体现出来，然后通过参数映射来挂接一个或多个相应的业务逻辑单元，来实现整个设计过程。

设计模板开发环境主要由两部分构成，一部分是人机交互界面的定义层，即UI层，一部分是业务逻辑层，两部分之间通过参数来进行驱动。UI层可以将参数传递到业务逻辑层，而业务逻辑层在执行完成以后，可以将结果反映到UI层上。其中UI层可以由用户从头开始从基本控件元素来进行设计，也可以从库中选择和使用子模板UI。

用户自定义UI

自定义逻辑

子模板UI

关联

子模板逻辑

Framewrok

* 1. 模板执行环境简述

该模块主要用于动态解析生成的模板文件，生成模板供用户执行，用户可在模板界面上进行快速交互操作，由执行引擎驱动仿真模板或其它逻辑单元完成模板指定的功能，并将结果数据反映到UI层。

* 1. 模板库简述

模板库可以对模板进行分类和管理，在使用时可以按照不同的维度来展示，并支持模板搜索。

同一个模板，可以按照不同的维度进行分类，例如可以将一个模板归类为材料库、型材库、曲线库和函数库等等维度来进行分类维护。分类维护可以方便用户通过不同的分类来管理和使用模板。

1. 模板开发

模板开发框架为各个专业的工程人员提供统一的模板搭建框架，用户不需要掌握计算机编程，只需专注于业务的梳理，通过拖拽控件和简单配置即可完成整个模板的创建与执行。

* 1. 基本元素

经过对航空、航天、船舶和电子行业等业务领域的调研，抽象出来一套工程计算模板上所需要的一些基本元素，通过这些基本元素所创建的模板能满足大多数专业需求。

* + 1. 数值类型

根据实际工程场景的需要，参数具有以下分类：

1。按照参数类型分为：实型，整数，字符；

2。按照组织形式分为：标量和数组；

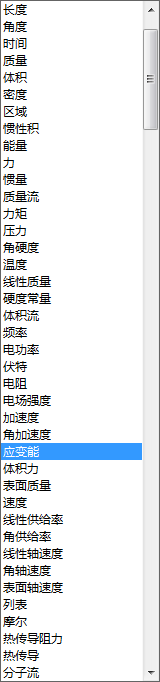
3。按照方向分为:输入，输出，输入兼输出。

1）实数类型：当鼠标光标定位到空白模板的任意位置时，单击图标 ，则弹出如图所示实数定义对话框：



在弹出的对话框中有以下几种属性：

* 变量名，是指显示在模板中变量的名称；
* 类型，可以是如下图所示



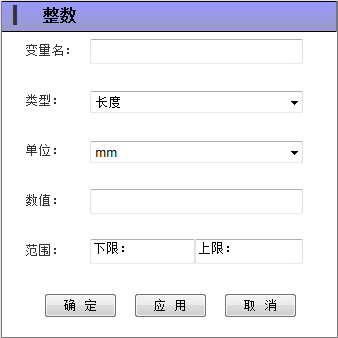
* 单位，单位的下拉菜单根据选定完变量类型自动判断；
* 数值，变量的初始值；
* 范围，限制变量值的范围，如果超出限制，在光标失去焦点时会弹出警告对话框，并将值置灰，不可用。

最后在模板上的显示的实数变量如图所示



其它数值类型属性同上。

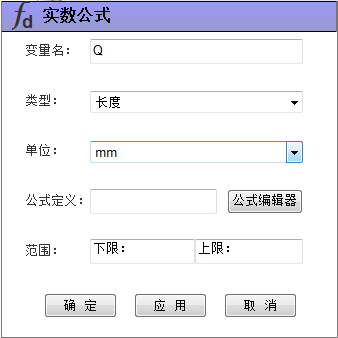
2）整数类型：单击图标 ，则弹出如图所示整数定义对话框：



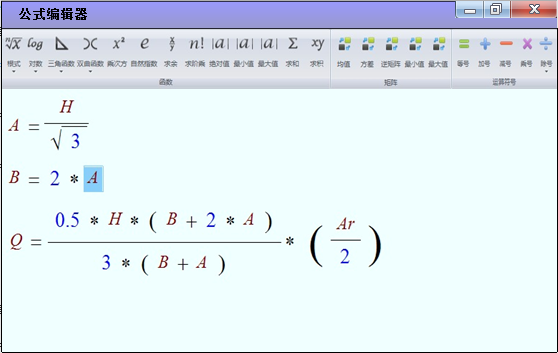
3）字符类型：单击图标 ，则弹出如图所示字符定义对话框：



4）实数类型公式：单击图标 ，弹出如图所示对话框



* 变量名，是指公式等号左边的因变量名；
* 类型和单位同实数类型
* 公式定义，单击公式编辑器，弹出如下图对话框



公式编辑器详情见4。2节。

在编辑器中敲入公式，保存关闭后系统自动截取公式图片，并插入到公式定义文本框中，点击确定后，在模板上生成如图所示的公式类型



如图所示公式并不能编辑，只是以图片格式显示。

* + 1. 控件

如图所示，控件包括以下几种类型：

本图不包括曲线工具和Tab页签

1. 文本框

使用文本框可以定义标题、副标题、摘要和说明性文字等等，可以通过工具栏更改字体和字体大小，还可输入希腊字母表等，工具栏如图所示



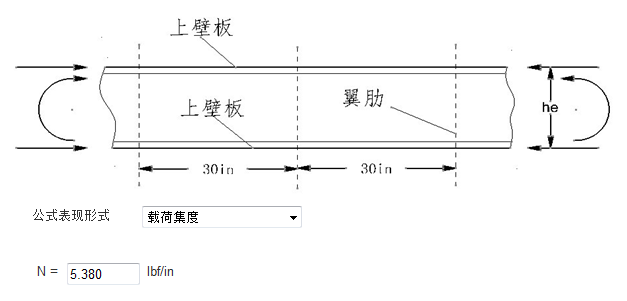
1. 表格

表格设计采用的是FlexCell表格控件（待定），FlexCell是一款灵活、易用的表格和报表控件，提供了多种单元格和图表类型，提供了合并单元格、虚表、打印和打印预览等功能，支持剪贴板操作、导入和导出XML文件、导出CSV/HTML/PDF文件，不需要Excel运行库支持就可以导出为Excel WorkBook。软件包中包含表格设计器，可以极大地减少编程人员的工作量，使用少量代码即可设计出复杂的表格。如图所示



1. 图片

工程人员在制作模板的过程中会需要一些配图加以说明，能更直观的将模板展示出来。如图所示

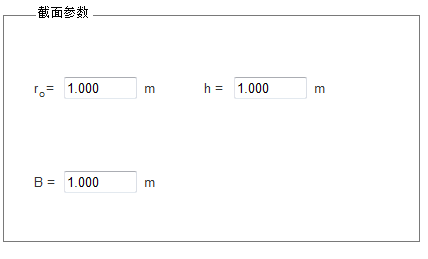


1. 超链接

在配图或文字说明的时候可以增加链接功能，可以指向其它Web页面，超链接也可以在报告中生成。

1. Group参数组

Group参数组具有参数分类的功能并且一次定义同一类型的多个变量，定义完毕后，可以双击每个参数并定义属性，如下图所示



1. 下拉菜单

见4.1.3节

1. 文件夹

文件夹控件功能是归类模板文件，当点击文件夹图标后，在左侧设计树上会增加一个节点，在新建模板后设计树上会自动生成一个文件夹节点，双击图标可以重命名，在导出的报告里，文件夹名称相当于一个标题。

1. 曲线工具

见4.3节

1. 3D模型显示

3D模型显示可以藉由第三方显示控件,对常见CAD,CAE模型进行展示.

1. Tab 页

Tab页签的功能也是归类模板内容，如果模板计算内容较多，可以分成几部分进行计算，每个部分都放到一个Tab页签中，Tab页签可以增加或者删除；

Tab页签与设计树之间的关系：每一个Tab页签都对应一个设计树，切换页签，设计树也对应变化，并且每增加一个Tab页签，相当于在设计树上增加一个文件夹节点。如下图所示。

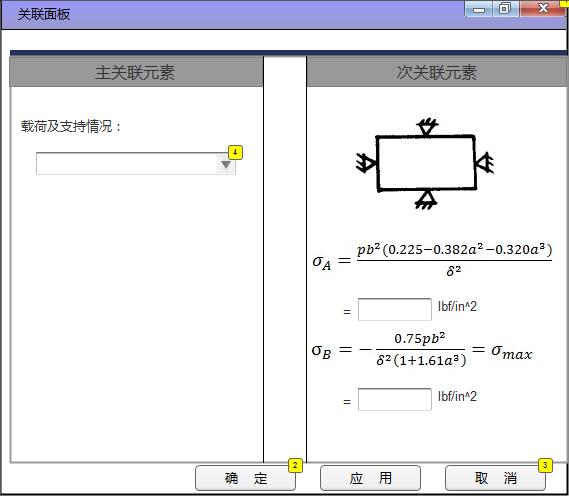


* + 1. 联动映射

联动映射是指将下拉菜单中的某一项和数值类型变量、数值类型公式或者其它控件关联起来，其本质是将if或switch语句通过UI图形界面可视化显示出来，鼠标左键+Ctrl点选需要关联的控件，点击 图标，将下拉菜单自动增加到左面的主关联元素上，其它控件增加到右边的次关联元素，并弹出如下图所示对话框



下拉菜单关联图片控件和数值类型公式，如下图所示



* + 1. 布局

布局分为自动布局和手动布局两种方式（类似于综合设计环境中自定义界面组件中的布局模式）：

1）自动布局，在后台代码中根据空白模板的长度和宽度，给出一个合理的行间距和列间距，依次顺序往下排列；

2）手动布局，采用左右对齐，上下居中常用的控件排列方式，如下图所示



* 1. 公式

为了减小开发工作量和难度，公式部分内容要需要找一些功能强大的公式库，首先想到的是MATLAB和Mathematica，由于MATLAB提供的API不是开源而且收费，并且Mathematica的公式库体积庞大，所以需要另找开源公式库，经过调研，可以被。Net所调用的Python语言下的Numpy和Scipy公式库具有体积小，免费开源的特点，并且功能可以与MATLAB相媲美。

作为Python Tools for Visual Studio项目的一部分，NumPy和SciPy程序库已经迁移到。NET上了。这项迁移通过本地的C核心组合了C#和C接口。IronPython对NumPy和SciPy的端口是完整的。NET端口，并且包含了针对一般本地C核心的自定义C#/C接口。这意味着不仅IronPython能够使用所有功能，而且所有。NET语言——像C#和F#——也可以使用，它们只需要访问C#接口对象，或者有时需要从其它。NET语言解析IronPython表达式。这意味着多维数组对象（ndarray）可以在IronPython和C#或者F#代码之间平滑地传递。此外，ndarray对象实现了标准的IEnumerable接口，这让数组对象可以由现存的代码经常使用，即便不是特别针对NumPy的代码也没问题。

NumPy是一种很低级别的API，用于在大型、多维数组和矩阵上执行数学运算。SciPy是在其基础之上构建的。SciPy包含很多模块，能够用于优化、线性代数、整合、插入、特定函数、FFT、符号和图像处理、ODE解析器以及其它科学和工程领域的一般任务。”

NumPy和SciPy的组合为一般的。NET代码提供了很多显而易见的优势。。NET的垃圾回收程序能够比手动的内存管理提供更好的性能，还有就是，通过高度优化的C代码，可以获得很好的计算速度。

* + 1. 表现形式

公式可以通过公式组件进行创建,编辑,完成后的公式可以以图片的方式嵌入到模板页面中,作为参考,使公式编写人能够第一时间看到公式的组织形式.

* + 1. 六种基本函数

六种基本函数是指：正比例函数、反比例函数、一次函数、二次函数、指数函数、对数函数、幂函数，如下图所示



* + 1. 三角函数

可以计算正、余弦，正、余切，如下图所示



还可以计算双曲函数，如下图



* + 1. 数组函数

可以定义向量或者数组，还可定义矩阵，数组函数包括以下几个：

* 求逆
* 转置
* 计算行列式的值
* 计算特征值和特征向量
* 可以计算点积和向量积
* 可以求向量的和
  + 1. 分段函数

分段函数采用if语句，例如，在公式编辑器中敲入k(x)=if(1<x<4，f(x)，-f(x))

它的含义是当1<x<4时k(x)=f(x)，否则k(x)=-f(x)。

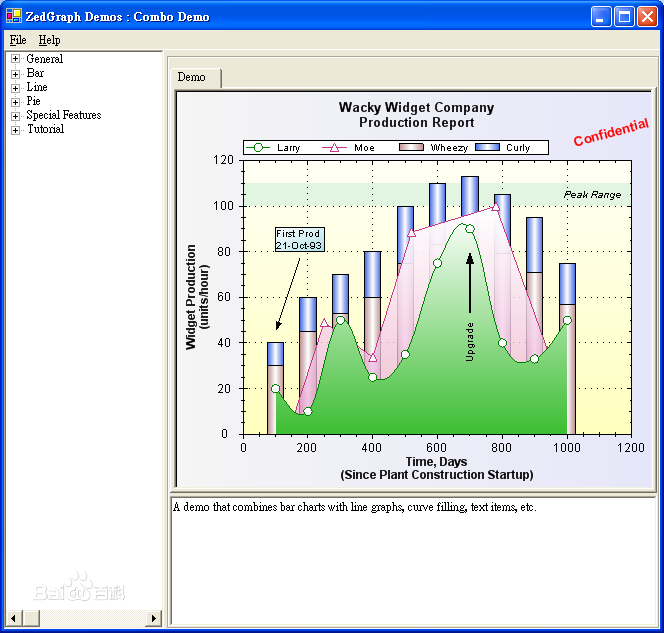
* + 1. 其它函数

依据Scipy和Numpy的强大功能，可以计算阶乘、绝对值、最大最小值、连乘和连加等，如下图所示



* 1. 曲线

根据对曲线开源控件的调研，曲线控件采用ZedGraph的开源曲线图库（待定），ZedGraph 是一个开源的。NET图表类库， 全部代码都是用C#开发的。它可以利用任意的数据集合创建2D的线性和柱形图表。ZedGraph 的类库具有很高的灵活性。几乎图表的每个层面都可以被用户修改。同时，为了保证类库的易用性，所有的图表属性都提供了[缺省值](http://baike.baidu.com/view/116467.htm)。类库中包含的代码可以根据被划分的数据来选择适应的比例范围和步长、尺寸。 ZedGraph 继承了Framework中的UserControl接口，所以允许用户在VS 的IDE 环境中进行拖放操作。如下图所示



* + 1. 曲线定义

1）在模板库中，可以通过曲线工具定义曲线，如下图所示，曲线定义有两种方式：

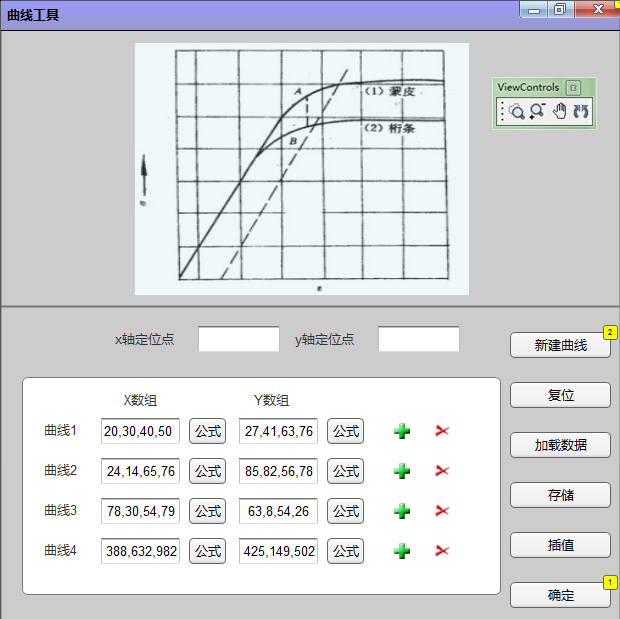
①。通过公式定义曲线，但是要指定横坐标x的取值范围；

②。通过离散的X和Y坐标点定义曲线，但是要指定拟合方法并插值。

2）在同一个坐标系中可以定义多条曲线，点击“+”号添加曲线，点击“×”号，也可以删除曲线；

3）在曲线工具中可以通过ViewControl小工具放大缩小曲线的局部，也可以平移和旋转曲线；

4）在建立好曲线后，在x轴定位点和y轴定位点中调试所建立的曲线，看计算出来的x或y值是否符合预期要求，不符合则可更换拟合方法等。



5）右侧按钮依次是新建曲线、复位、加载数据、存储、插值、确定

①新建曲线，是指新建一幅曲线图，不是指单条曲线；

②复位，是指重置整幅曲线图；

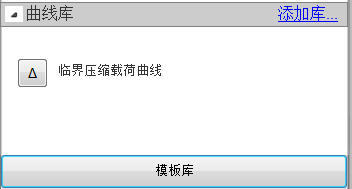
③加载数据，点击后弹出文件浏览对话框，直接加载一幅定义好的曲线图文件，例如“临界压缩载荷曲线。prjplot”；

④存储，点击后弹出文件保存对话框，保存成曲线格式文件，例如“\*。prjplot”

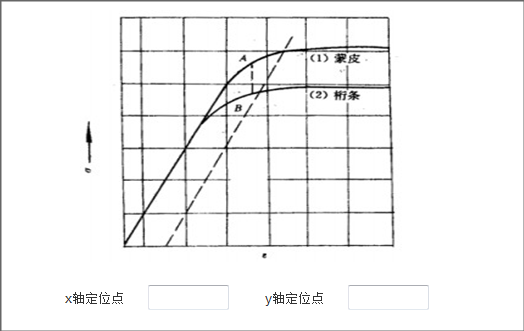
⑤插值，选择拟合方法，例如最小二乘法、拉格朗日插值法、牛顿插值法、牛顿迭代法、区间二分法、弦截法、雅克比迭代法和牛顿科特斯数值积分法等等。

* + 1. 曲线查询

在曲线工具中保存完成定义的曲线之后，曲线会保存在曲线库中，如图所示



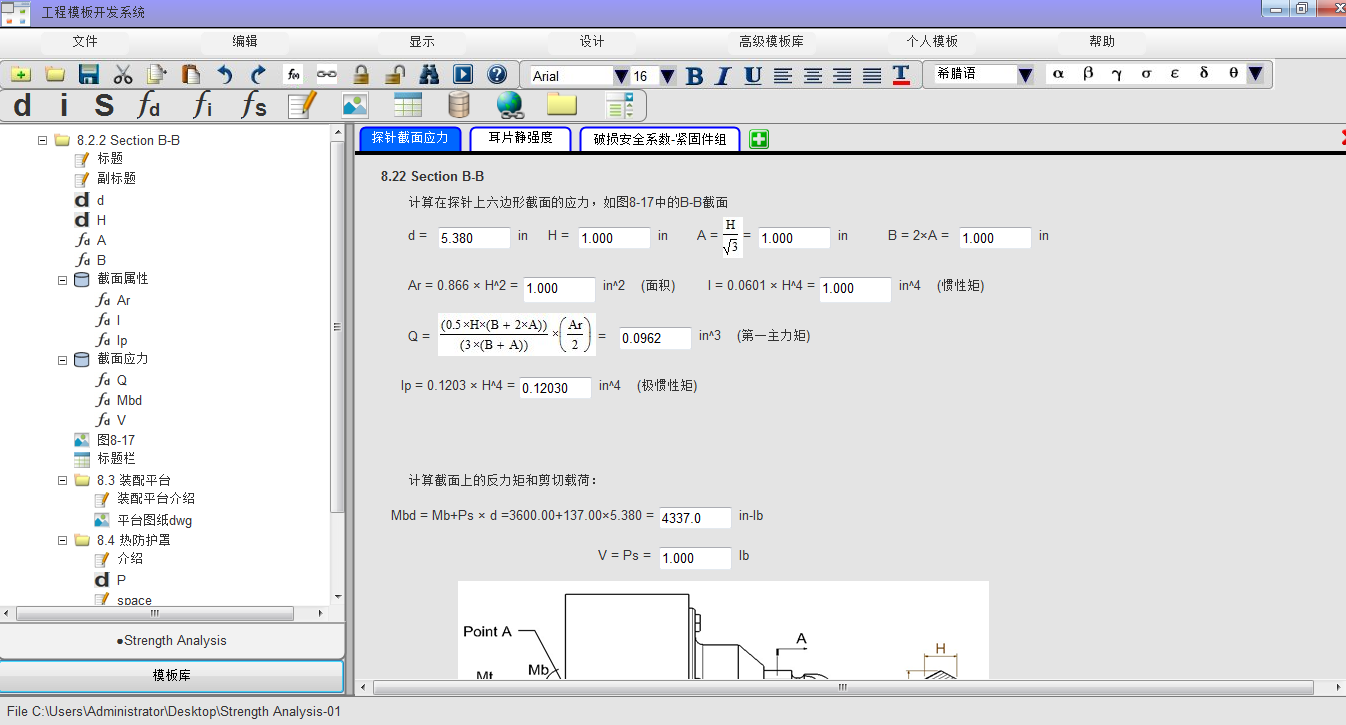
点击曲线“临界压缩载荷曲线”，则弹出曲线图，如下图



在计算x或y之前需要点选其中一条曲线，当曲线高亮显示时，即可计算所选曲线的x或y值。当x或y值计算出后，可以当做输入值供后续公式调用。

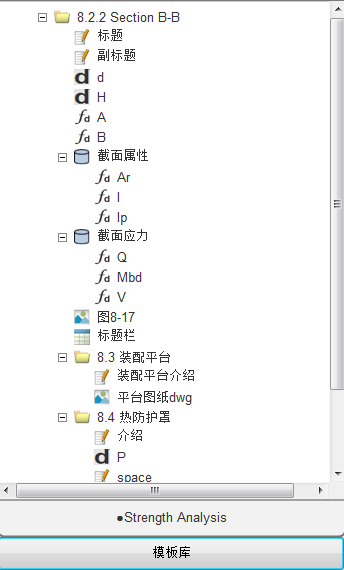
* 1. 模板视图

模板开发环境整个视图从上到下分为菜单栏、工具栏、设计树、模板开发区、模板库和状态栏。整个系统如下图所示



* + 1. 设计树

在空白模板区上建立的变量和控件在设计树上都能记录下来，按照建立的先后顺序或者按照分类（待定）在设计树上展现，如下图所示

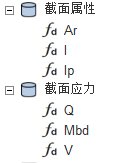


1）在最上层所展示的文件夹的图标相当于一个模板的标题（如果这模板生成一个报告，相当于一个章节），在这个节点下面是模板里的内容，比如标题文本框、数值类型、公式、图片控件和标题框等。

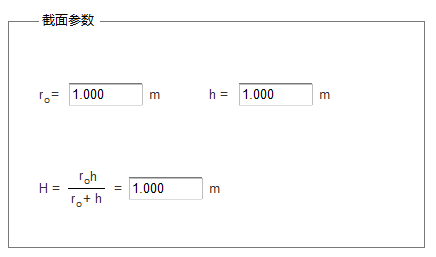
2）这里首先要说明一下参数组的概念（Group），参数组可以一次定义多个参数，如图所示



例如在“数据组名”输入“截面属性”，数据个数输入3，则直接生成3个参数，如图所示



在模板开发区上的显示如下图所示



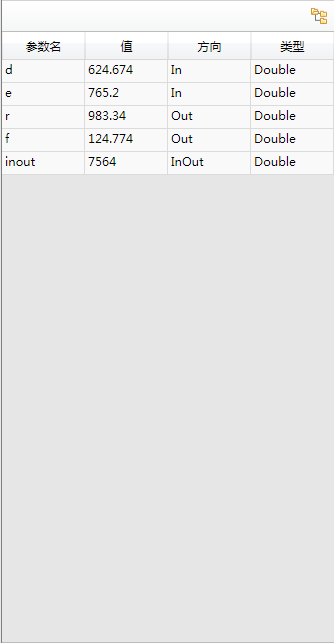
3）从模板库拖入模板时，模板库的参数在设计树上的是按参数组的显示形式来展示，在模板开发区的展示形式如上图一样，模板库的名称就是参数组的名称。

* + 1. 属性视图

每个定义的参数或者控件，在双击之后都会弹出对应的属性框，可以在设计树上双击，也可在模板开发区上双击。

* + 1. 参数列表

在设计树上设置一个按钮，点击此按钮则列表显示出当前模板中的变量，如下图，包含参数名、值、方向和类型，类似于封装环境中的参数列表。



参数列表中的参数不可同名，如果在命名时有同名参数，会弹出提示对话框，使用希腊字母进行参数命名，则在参数列表中自动转换成英文字母。

控件工具条

控件工具条可以拖动，既可以吸附到工具条的位置上，也可拖出来，放置到自己合适的位置，如下图所示



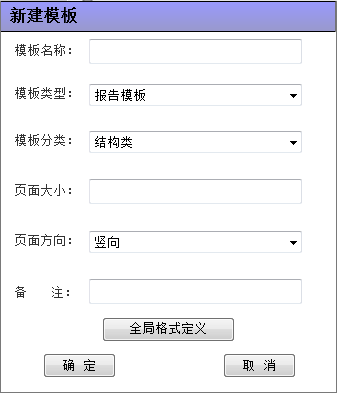
* 1. 模板管理

模板管理的内容放在主菜单下，如图所示

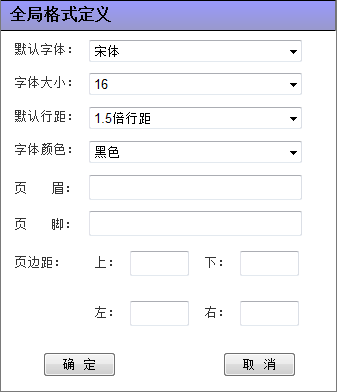


* + 1. 新建模板

在菜单项“文件”下点击新建模板，则弹出如图所示对话框



可以定义模板名称、类型、分类、页面大小和页面方向等等，点击全局格式定义后，弹出如图所示对话框



可定义模板的全局格式。

* + 1. 打开模板

打开模板弹出窗口如下图所示



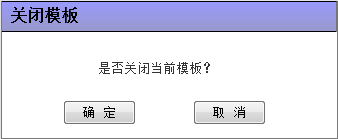
* + 1. 保存模板

保存模板如图所示



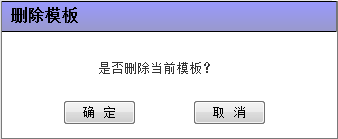
* + 1. 关闭模板

关闭模板如图所示



* + 1. 删除模板

删除模板如图所示

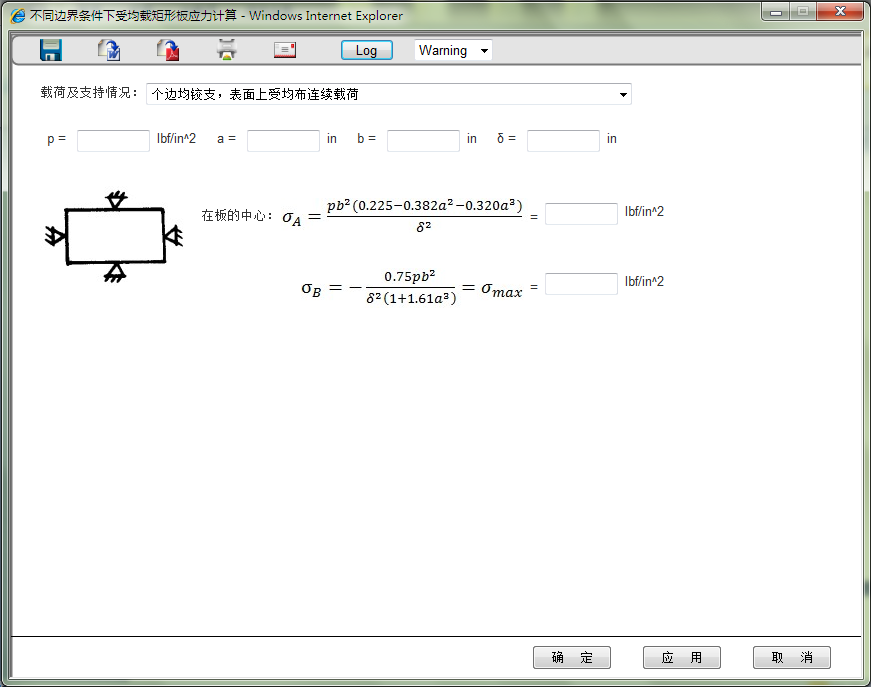


1. 模板执行
   1. 执行环境配置

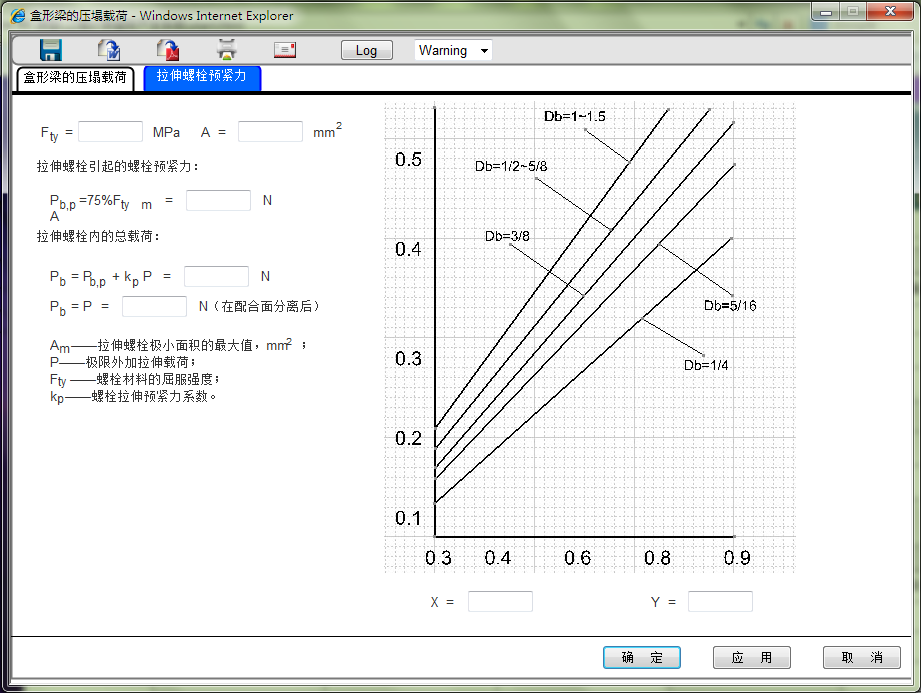
在模板执行之前，可以对模板执行进行一些配置操作，具体的有:

* 1. 默认执行级别设置
  2. 是否自动打开运行目录
  3. 数字的显示模式(有效数字的长度，科学计数法等等)
  4. 执行过程

模板建立完成并保存为“\*。prjtemplate”文件（或者命名其它后缀名待定），此文件在安装有工程计算模板开发系统环境的计算机上自动被关联，双击此文件，则弹出模板执行窗口，如下图所示



带Tab页签的模板运行，如下图所示



* 1. 执行控制

在模板开发环境下生成的模板文件（\*。prjtemplate文件）是一个可以点击执行的程序（类似于\*。bot文件），运行的原理也和bot文件相似，都是基于Flexware框架，但需要增加以下内容。

* + 1. 一次性执行（确定）

在模板文件执行的窗口上有3个按钮，如下图所示



当点击“确定”按钮时，模板运行，运行完成后整个执行窗口关闭，并在模板文件所在的目录下生成“out。xml”文件，此xml文件记录模板文件运行时的信息和输出参数的信息和结果。根据前面的设置，在运行后有可能会自动切换到结果文件目录。

对于含有多组公式的模板，按照公式引入的顺序来执行。因此，在创建模板时，要象平时推算数学式子一样，有顺序地书写公式。对于绝大部分的情况，这一点不难做到。

* + 1. 反复执行（应用）

当点击“应用”按钮时，执行窗口不关闭，并且在当前目录生成“out。xml”文件，并将xml文件中的结果参数返回到执行窗口中并显示，如多次点击应用，可以多次试算，并且out。xml多次更新。

* + 1. 取消执行

点击“取消”按钮，模板文件不运行，并且执行窗口关闭。

* + 1. 执行顺序

执行顺序按照结构树,由上而下进行,类似于代码由上而下执行.

* 1. 日志

生成跟踪日志是检测模板运行是否正常的必要手段，一旦出错，可以快速通过日志查找出问题出在哪里。

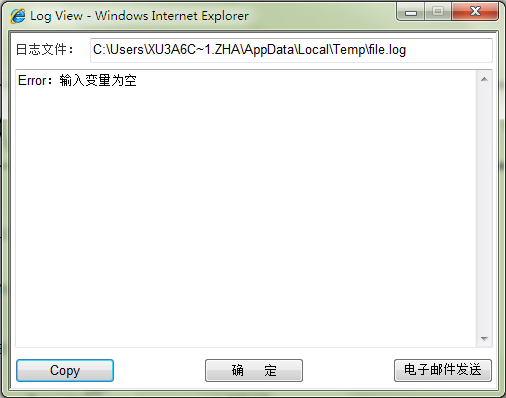
日志根据日志级别的不同，使用不同的标识来显示。错误日志使用红色，警告日志使用黄色，信息和Debug使用正常的底色。

* + 1. 前台显示日志信息

在执行窗口的工具栏中，有显示log文件的按钮，如下图所示



点击后，弹出如下图所示窗口



log窗口上显示日志文件的路径，文本框上显示日志，并可以复制，或者通过内部或外部邮件系统发送邮件。日志内容显示为中文。

* + 1. 日志文件

通过日志文件，可以跟踪一段时间以来系统的运行状态以及用户的操作过程。

* + 1. 日志设置

通过完善日志记录器程序，将日志级别分成以下几类，该级别可以在执行环境配置中进行设置。除了可以设置日志级别以外，还可以设置默认日志文件的生成位置。

⑴ SysError: 表示非常严重的错误等级，记录极有可能导致程序终止运行的致命错误信息，每个严重的错误事件将会导致应用程序的退出；

⑵ Error: 表示较严重的错误等级，但仍然不影响系统的继续运行，输入参数在计算的过程中可能会有问题；

⑶ Warn: 表示可能对系统有损害的情况，会出现潜在错误的情形；

⑷ Info: 默认的等级，突出强调应用程序的运行过程；

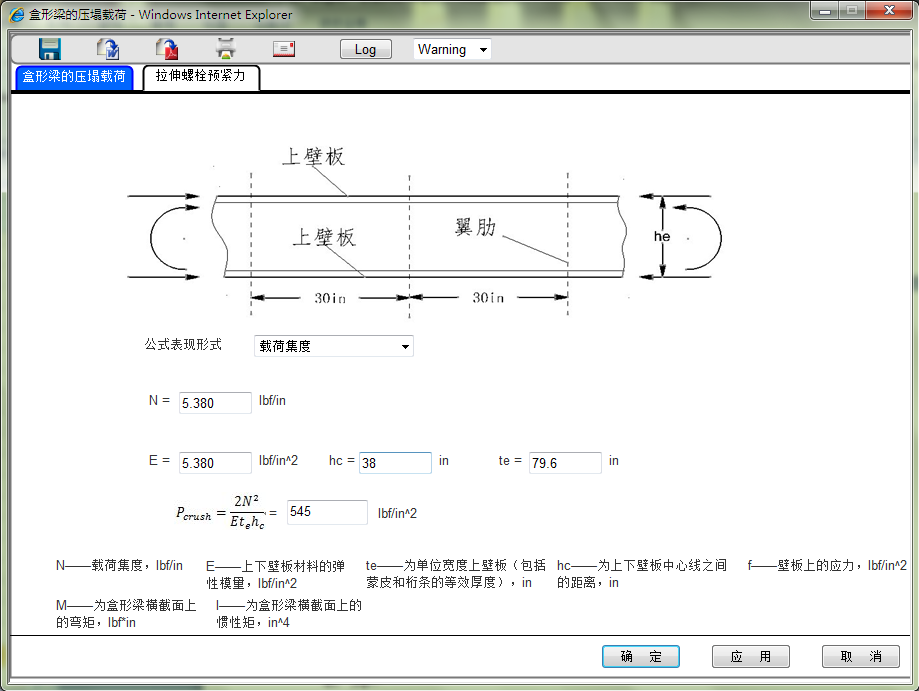
⑸ Debug: 这个等级表示用于调试程序的正常的事件信息；

* 1. 结果处理

模板在运行时过程中会产生一系列输入输出参数，参数的输出和模板的保存有以下几种方法。

* + 1. 在当前视图显示结果

模板在运行时，程序执行完成后，在工具栏上点击保存按钮后，当前的输入和输出参数会保存在当前界面上，模板关闭后再打开会将最后一次执行的结果显示在界面上，如图所示

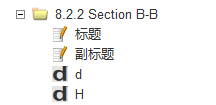


* + 1. 导出模板

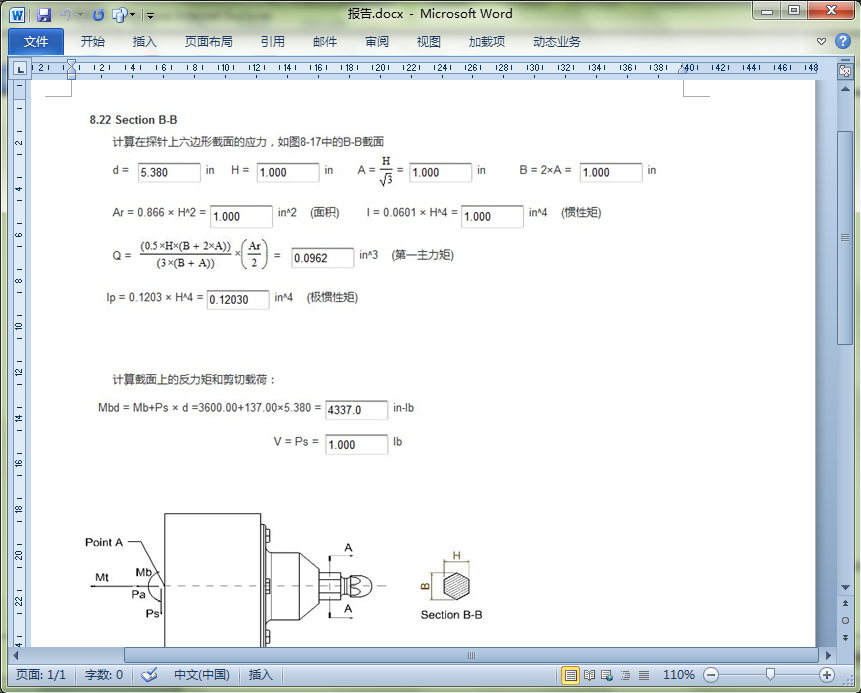
模板可以导出为doc和pdf格式报告，如图所示



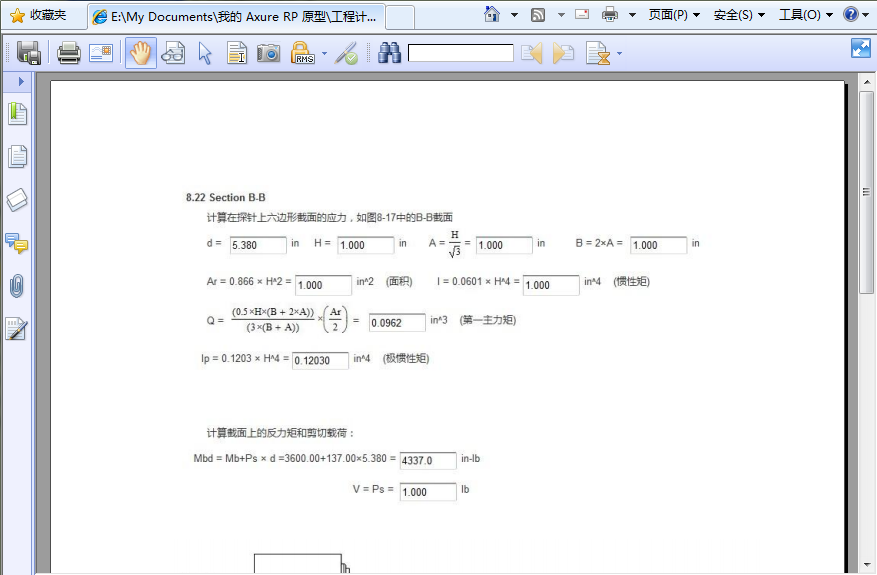
设计树上每一个带文件夹图标的节点会被当成word或pdf文档的二级标题使用，模板的名称是一级标题，如下图所示



word报告格式如下图所示



pdf报告格式如下图所示



模板还可以直接打印成word或者pdf文档格式，还可以通过内部或外部邮箱系统发给指定的人员。

1. 库

模板是实现功能重用的一种机制，使用模板可以抽象化业务，达到重用的效果。若干模板的集合组成模板库。用户可以在模板库中方便地选择，调用。

一般来讲，库里的内容是模板的片断，类似于程序中的代码片断一样。频繁被调用的功能被归类整理，形成功能子模块，这些子模块即库中的内容。这些具有特定功能的子模块管理和维护是藉由库来完成的。库的概念类似于文件系统中的文件夹，是文件存放的载体，更形象地讲，库是子模板或者说模板片断的容器。

子库，是库的细分，相当于子文件夹或子目录的概念。

库或子库的本质是操作系统上的文件夹，而库中的内容，即子模板是具有视图和数据的逻辑单元。该逻辑单元存在于文件夹中，构成了整个库。

* 1. 库管理

库管理可以分为导入库，导出库以及管理当前库中的内容，包括库的导入和导出（以实现库的共享功能），管理库的分类，如添加或者减少库的分类，以及管理某个库中的具体内容。如下图所示，



* + 1. 库的查看

通过库的查看功能可以方便地浏览库中的元素，在没有引用之前就可以知道各元素的参数以及视图。

* + 1. 常用库管理

把经常用到的元素放到常用库里，免去每次都要去各个库中进行查找的功能，类似于WEB浏览器的”收藏”功能。

* + 1. 库的导入导出

将已有的库导入到当前系统中。对于重合的部分，用户可以覆盖或者略过，类似于文件夹到文件夹的拷贝。

将当前系统中的库有选择性地进行导出，导出的载体是压缩包。例如，导出材料库时，材料库的内容以material.db形式导出。

导出库

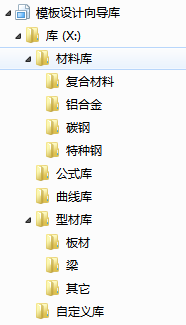
导入库

当前系统

库的压缩包文件

* + 1. 管理库分类

通常来讲，系统中内置的库的内容不能够覆盖所有领域的需求，即我们对库要进行一定的扩展，例如在电子行业，需要增加新的电子元器件库，而电子元器件库中又有可能再细分。



* + 1. 新建库分类

新建库分类分为两种情况:

1. 在根分类下新建一个大的分类；
2. 在某个分类下面，建一个更细分的分类。
   * 1. 删除库分类

删除库分类意味着删除该分类以及该分类下所含的内容，类似于文件系统上删除一个文件夹一样。

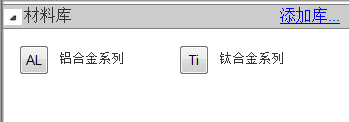
* + 1. 管理库中的内容

1. 新建子模板: 在库的某一分支下新建模板片断
2. 导入导出子模板: 将库中的一个或者多个子模板进行导出
3. 删除子模板:在库的某一分支下删除子模板
4. 移动子模板:将子模板从一个库移动到另外一个库，类似于文件的剪切粘贴
5. 重命名子模板:给子模板重新命名，子模板不允许有重名现象存在。
6. 编辑子模板:修改子模板的视图或数据，保存或另存成更新后的子模板。
   1. 库的分类

库基本上分为两大类: 系统内置库和用户自定义库。其中系统内置库中又包含四类，分别是材料库，型材库，曲线库和公式库。这四个库不能够被删除或重命名，但是可以增加子库。也不能移动位置，属于系统自带的库。用户自定义库目录下，用户可以自由在添加子库。

* + 1. 材料库

材料库属于内置库，其基本的作用是在计算过程中对材料进行查询和引用，该库提供基本的页面布局，可插入到当前模板中。如图所示

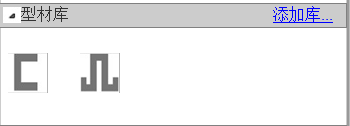


材料库的显示方式可以是图标式还可以列表显示，图标可以自定义，例如导入图片等，点击添加库之后，弹出材料库的管理界面，如下图所示，具体功能详见6.1节



* + 1. 型材库

型材库属于内置库，其基本的作用是在计算过程中对型材进行查询和引用，该库提供基本的页面布局，可插入到当前模板中。与材料库不同的是，型材库的每一个子模板带有型材的图片。如下图所示

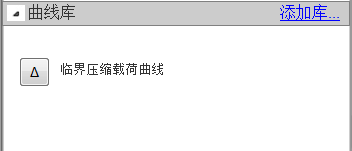


* + 1. 曲线库

曲线库相对来讲是比较复杂的，因为除了引用曲线以外，还要负责实现对曲线的编辑，具体请参考前面关于“曲线”的章节。

* + 1. 公式库

公式库是内置库中应用最多，同时也是含有数据计算关系的库。该库中的元素含有两个层次，一是视图层，二是业务逻辑层。当用户将公式库中的元素插入到当前模板中，这两层关系也将随之被调用。具体请参考6.5使用库。如图所示



* + 1. 标准件库

用户在建模或设计过程中经常会用到标准件库中的零件或元器件，因此在进行模板开发和使用的时候不可避免地要访问这些标准件，因此标准件库的存在使这个过程大大简化,且有利于协同工作中保持数据一致性。并且，直接调用标准件库中的零件的参数，还可以大大降低查询手册过程中可能出现的偏差，从这个角度上来讲，标准件库还起到了消除人工手动查询带来的错误因素。

* + 1. 自定义库

根据专业和分工的不同，可自定义库，这样可根据用户需求拓展库的分类，新建一个自定义库相当于新建一个文件夹，自定义库里面也可以进行细分类，操作过程和文件夹操作类似，可以新建自定义库，删除自定义库，如自定义库内有文件，则提示是否确定删除，删除空库则直接删除没有提示。

* + 1. 模板库

模板库是对模板进行组织管理的载体。模板库中的模板是指具有一个完整分析的模板，而不是像材料库和型材库中保存的一个数据片段。

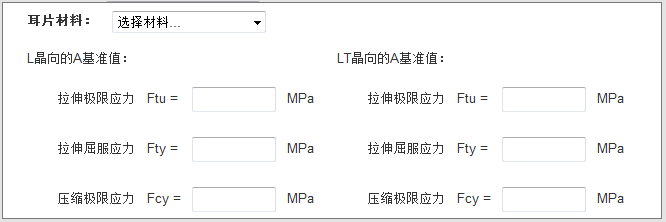
模板的创建，保存和修改均会记录在模板库中。用户可以将定义好的模板或者有代表意义的模板存储到模板库中，在以后的工作中如果遇到类似的问题可以直接从模板库中调用，或者在原有的基础上进行删除、添加和修改就可以达到目的，也可以从这些模板中挑选部分片段，加入到当前的工程计算模板中，重新生成新的模板。而不需要一步一步重新建立新的模板和设置属性信息了。

* + 1. 实例库

对于一个模板而言，不同的输入对应着相应的输出，为了能够保存用户在不同的设计下的结果，需要由实例库来保存这些不同的设计。有了实例库，用户可以对模板运行过程对模板产生的实例进行管理，把不同的设计，即不同的运行实例进行保存和对比,从而能够把模板的运行结果作一个统一的管理。

* 1. 使用库

库的使用类似于对函数的调用，即成为当前正在设计模板的一部分。但不同的是，库是拷贝式调用，即一旦库成为设计模板的一部分，原来库变化对现在的模板没有影响。 例如从材料库中引用铝合金材料属性的过程，点击鼠标左键拖拽图标，拖拽到模板开发页面上，松口鼠标左键，在当前页面上自动生成控件，如下图所示



对应的后台参数直接生成在参数列表中。其中既可以引用视图也可引用数据，或者两者都可以被引用。

* + 1. 引用库的视图

如果仅需要库中的输入输出表单，则可以选择性地引用库的表现部分，而忽略其内部数据逻辑关系，即只引用库的视图。

* + 1. 引用库的数据

如果仅需要库中的数据，以及数据之间的逻辑关系，而不需要视图，即只引用库中的数据。

* + 1. 引用库的视图和数据

综合以上两种情况，如果即要视图，又需要数据，则引用库中的视图和数据。

* + 1. 冲突的处理

在引入库内容的时候，有可能与现有模板的内容冲突，即出现了同样名称的数据，解决的方法有两种:

1. 自动映射。将相同的变量名自动合并，成为一个变量，前提是输入输出关系以及类型都能匹配上。
2. 提示冲突，并取消当前操作。直到数据冲突解决完成后，再引入。
   * 1. 库的接口

从已有的其它软件的库中，导出到本系统库的格式时，需要使用库的接口进行转换。例如，原有的系统中有一个Oracle数据库，库中存放了一批标准件。这些标准件可以通过库的接口进行开发，导出大量数据，生成本系统可以识别的中间格式。再由这种中间格式导入到本系统中，同6.1.1中所讲述的导入行为。

DB

当前系统

文件

Data