

# 流程图编辑器

概要设计说明书 V2.1

“三+二” 开发小组

2009-07-19

修改历史纪录:

\*A - Add    M - modify    D - delete

版本	日期	类型(A*M*D)	修改者	备注
V1.0	2009-07-13	A	全体组员	概要设计初稿
V1.1	2009-07-17	A,M	任天辰	完善初稿，添加模块处理流程图
V2.0	2009-07-18	A	全体组员	添加主要功能设计，添加程序主要流程
V2.1	2009-07-19	M	任天辰	详细各功能算法及接口，修改封面

# 目 录

一、引言 .....	6
1.1 编写目的.....	6
1.2 项目背景.....	6
1.3 定义 .....	6
1.4 参考资料.....	6
二、任务概述.....	6
2.1 目标 .....	6
2.2 运行环境.....	6
2.3 开发环境.....	7
2.4 用户特点.....	7
2.5 假定和约束.....	7
三、流程说明.....	7
3.1 总体流程.....	7
3.2 输入——用户操作.....	9
3.3 输入——从文件输入.....	9
3.4 处理——图形绘制.....	9
3.5 处理——折线绘制.....	10
3.6 处理——图形移动.....	10
3.7 处理——图形.....	11
3.8 处理——图形.....	11
3.9 处理——文字添加.....	12
3.10 输出——自定义格式保存.....	12

3.11 输出——PNG 格式保存 .....	13
<b>四、总体结构与模块划分 .....</b>	<b>13</b>
<b>五、接口设计 .....</b>	<b>14</b>
5.1 外部接口 .....	14
5.2 内部接口 .....	14
<b>六、数据结构说明 .....</b>	<b>14</b>
<b>七、主要功能及函数描述 .....</b>	<b>15</b>
<b>7.1 输入 .....</b>	<b>15</b>
7.1.1 功能 .....	15
7.1.2 输入项 .....	15
7.1.3 输出项 .....	15
7.1.4 算法 .....	16
7.1.5 接口 .....	16
<b>7.2 画图形模块 .....</b>	<b>16</b>
7.2.1 功能 .....	16
7.2.2 输入项 .....	16
7.2.3 输出项 .....	16
7.2.4 算法 .....	16
7.2.5 接口 .....	17
<b>7.3 移动模块 .....</b>	<b>17</b>
7.3.1 功能 .....	17
7.3.2 输入项 .....	17
7.3.3 输出项 .....	17
7.3.4 算法 .....	17

7.3.5 接口..... 17

**7.4 删除图形模块..... 17**

7.4.1 功能..... 17

7.4.2 输入项..... 17

7.4.3 输出项..... 18

7.4.4 算法..... 18

7.4.5 接口..... 18

**7.5 折线绘制模块..... 18**

7.5.1 功能..... 18

7.5.2 输入项目 ..... 18

7.5.3 输出项目 ..... 18

7.5.4 算法..... 18

7.5.5 接口..... 18

**7.6 文字添加模块..... 19**

7.6.1 功能..... 19

7.6.2 输入项目 ..... 19

7.6.3 输出项目 ..... 19

7.6.4 算法..... 19

7.6.5 接口..... 19

**7.7 改变图形属性模块..... 19**

7.7.1 功能..... 19

7.7.2 输入项目 ..... 19

7.7.3 输出项目 ..... 20

7.7.4 算法..... 20

第 4 页

---

7.7.5 接口.....	20
<b>7.8 输出模块 .....</b>	<b>20</b>
7.8.1 功能.....	20
7.8.2 输入项目 .....	20
7.8.3 输出项目 .....	20
7.8.4 算法.....	20
7.8.5 接口.....	20
<b>八、人员分工.....</b>	<b>21</b>

## 一、引言

### 1.1 编写目的

流程图、示意图是人们撰写论文和演示文稿时经常需要用到的。我们希望设计并实现一个图形界面的软件，使用它可以绘制标准的程序流程图或其它示意图。

### 1.2 项目背景

该软件名称为 Flow Chart Editor；通过这款软件，用户便可以方便地绘制标准流程图，极大方便用户撰写论文和演示文稿。

### 1.3 定义

本系统：“三+二”小组——流程图编辑器 Flow Chart Editor

本文档：《概要设计说明书》

### 1.4 参考资料

《VC++深入详解》

孙鑫

《软件工程基础》

高等教育出版社

《MFC Windows 应用程序设计》

需求规格说明书

## 二、任务概述

### 2.1 目标

通过我们的这款软件，用户便可以方便地绘制标准流程图，极大方便用户撰写论文和演示文稿。

### 2.2 运行环境

Windows 操作系统。

## 2.3 开发环境

Win32, 工具选用: vc++6.0 (MFC)

## 2.4 用户特点

本软件的用户需具备一定计算机编程结构方面的知识, 对流程图构图规则有一个清晰的把握, 并且用户也应具备 Windows 等基本操作技能。

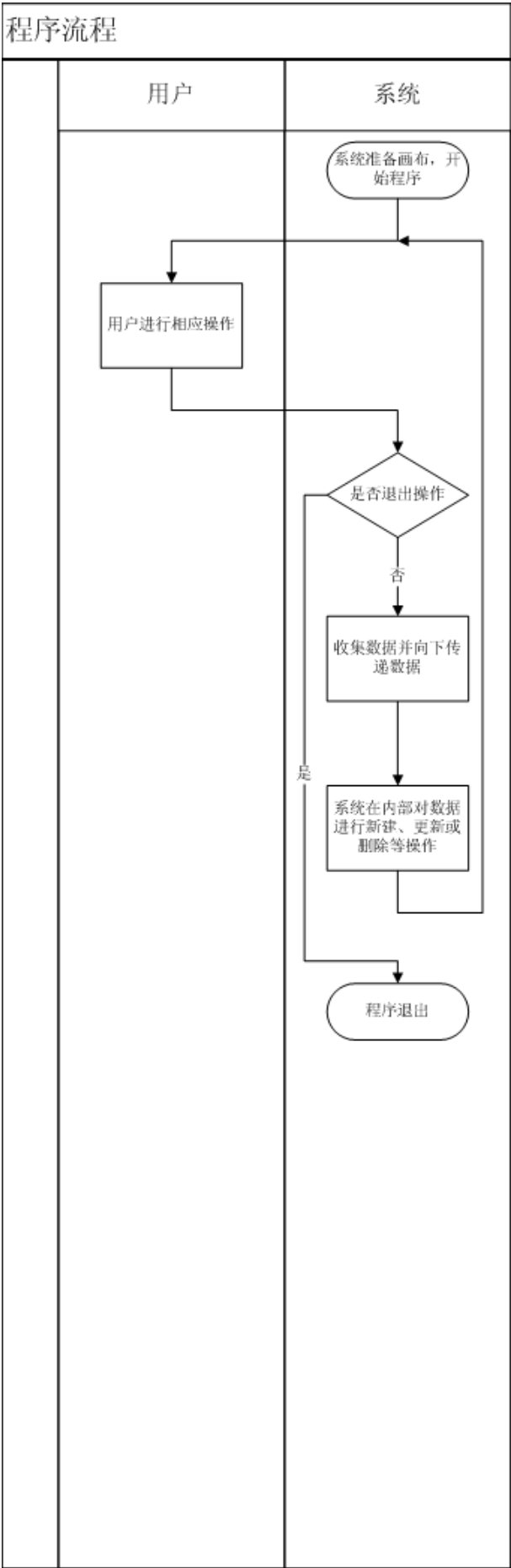
## 2.5 假定和约束

开发人员是第一次进行实际项目的开发, 有一个将所学知识转换为实用技能的过程。需要的开发时间较长, 在 2009 年 7 月底前完成。

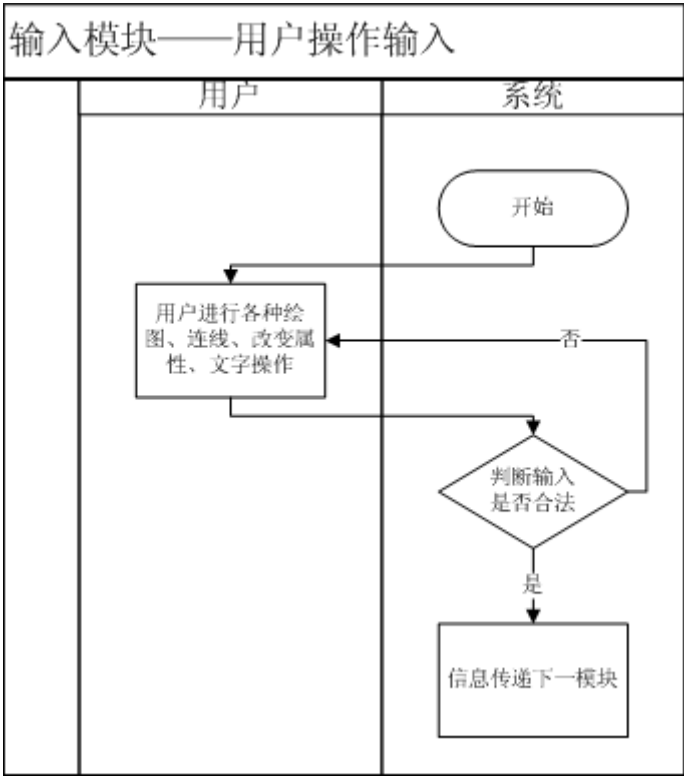
# 三、流程说明

## 3.1 总体流程

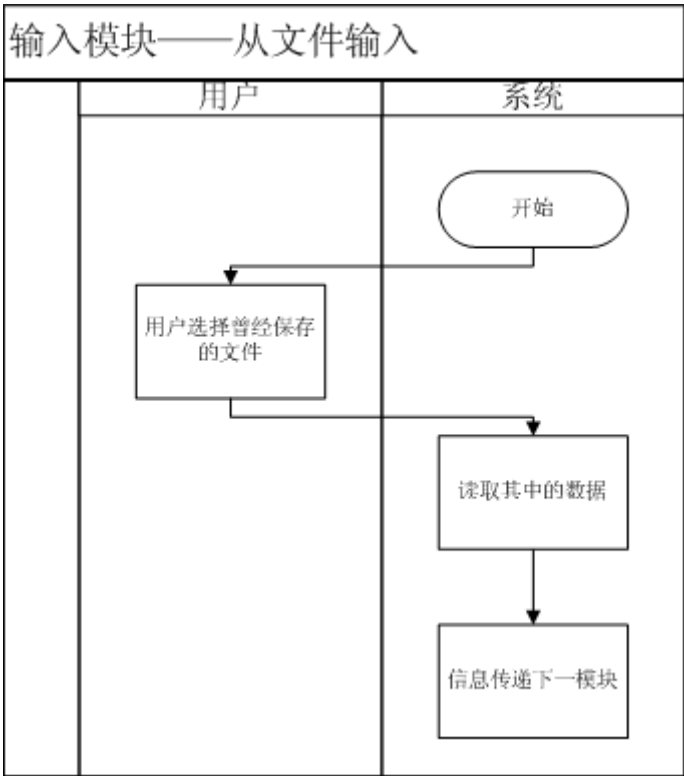




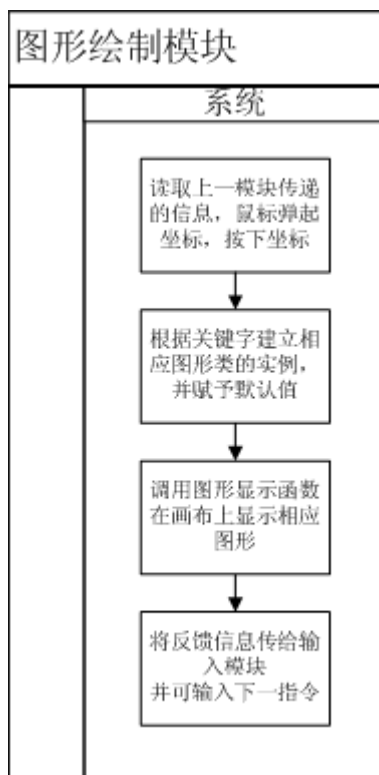
3.2 输入——用户操作



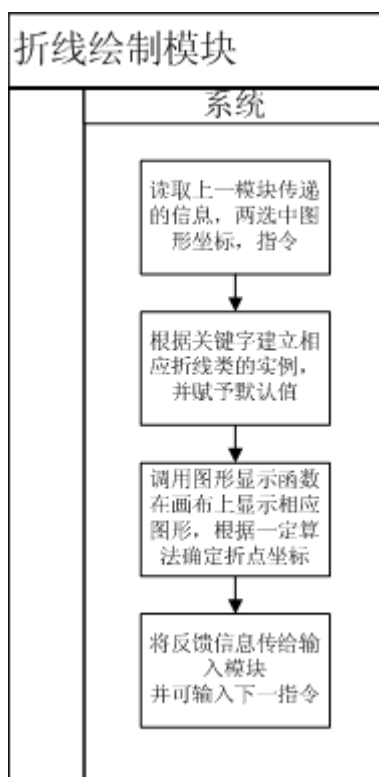
3.3 输入——从文件输入



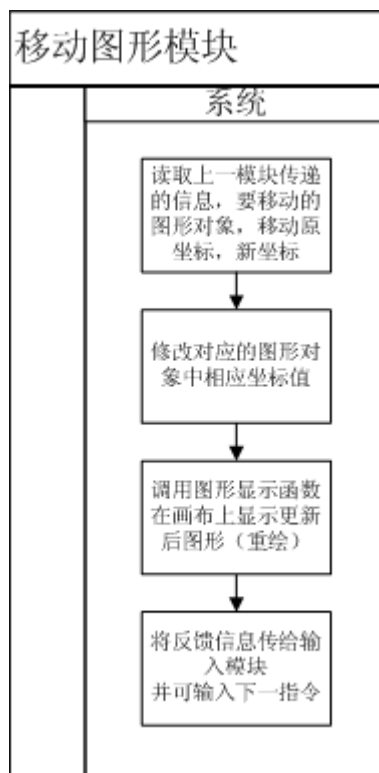
3.4 处理——图形绘制



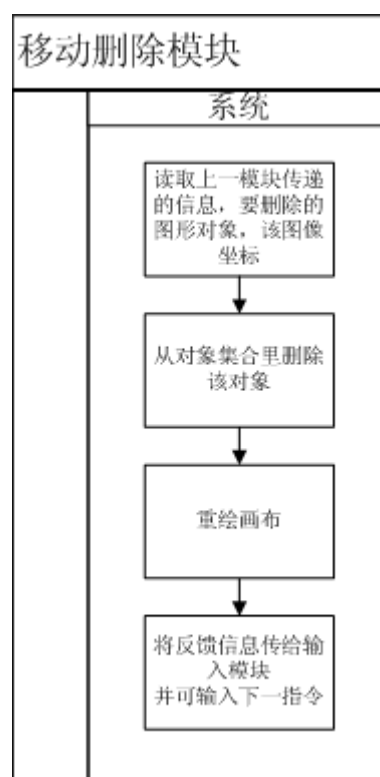
### 3.5 处理——折线绘制



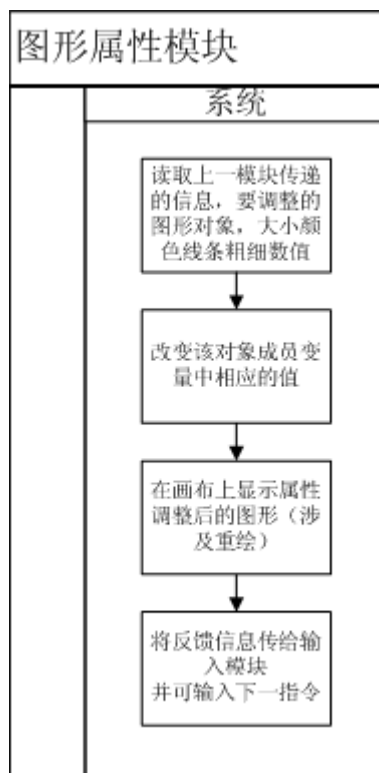
### 3.6 处理——图形移动



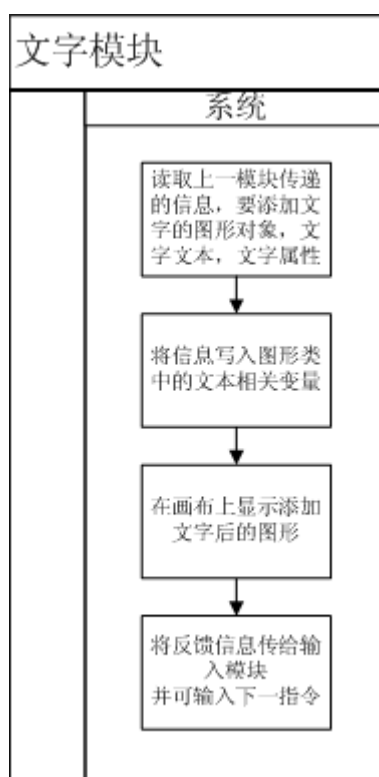
### 3.7 处理——图形



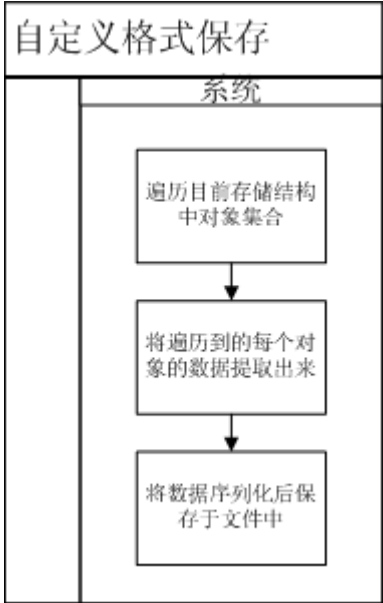
### 3.8 处理——图形



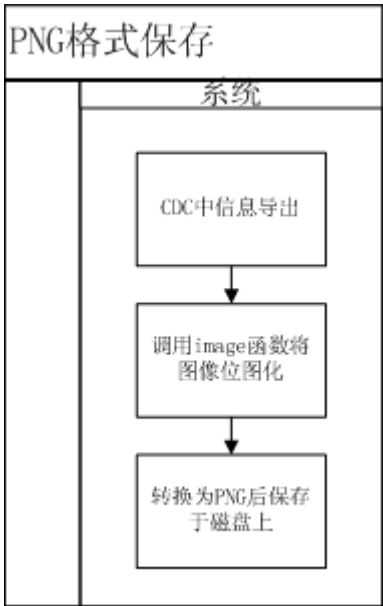
### 3.9 处理——文字添加



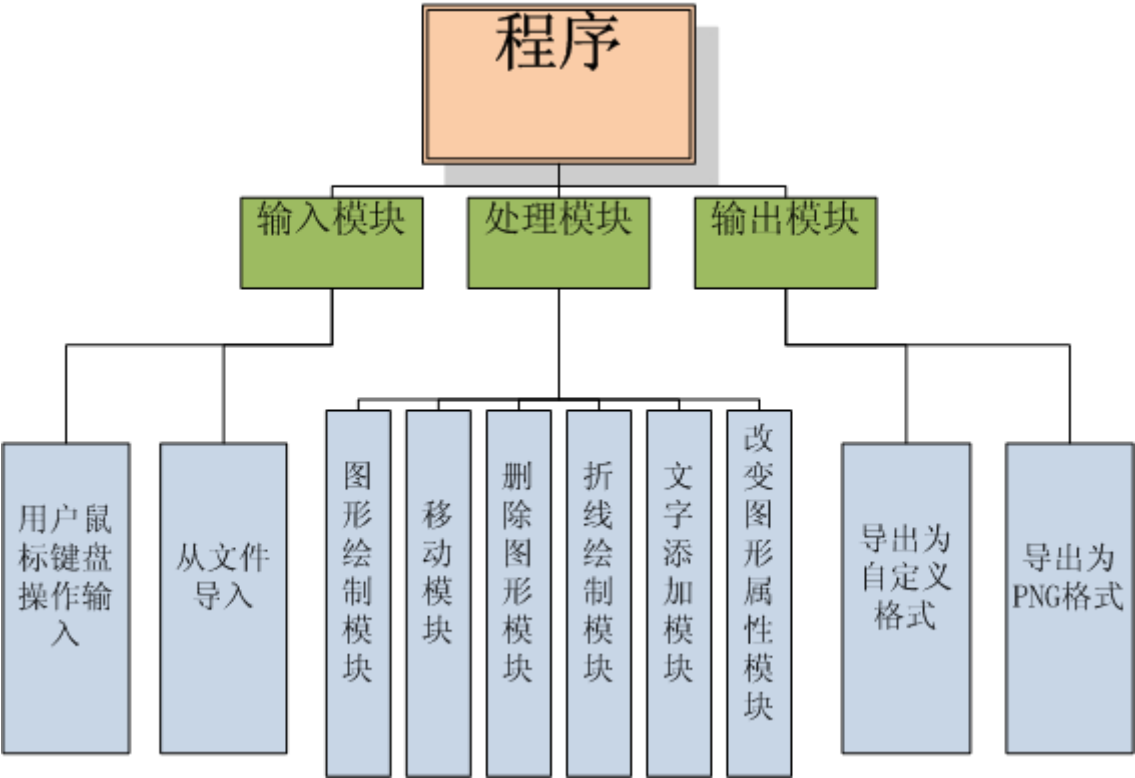
### 3.10 输出——自定义格式保存



3.11 输出——PNG 格式保存



四、总体结构与模块划分



五、接口设计

5.1 外部接口

本系统只针对本流程图编辑器使用，不与其他系统产生交互，因此不存在与外部的接口。

5.2 内部接口

本系统采用四层架构，分为图形表现层、数据结构层、界面操作层和文件操作层。内部接口是通过不同项目之间的引用实现的，主要表现为：界面操作层产生信息在数据结构层产生对象并在图形表现层显示；文件操作层通过读取内部数据层信息保存文件。

六、数据结构说明

该流程图编辑器根据需求和功能，详细设计了图形类作为其数据存储的结构。

图形类——CBasicShape

	名称	标识符	数据类型
属	图形类型	Tag	Int

性	左上坐标点	Point1	CPoint
	右下坐标点	Point2	CPoint
方法	构造函数	CBasicShape();	
	析构函数	Virtual ~CBasicShape();	
	标记图形类型	Virtual void SetTag(int mid_tag);	
	获取图形类型	Virtual int GetTag();	
	修改图形坐标点	Virtual void SetPoint(CPoint m_point1,CPoint m_point2);	
	绘图函数	Virtual void Draw(CDC* pDC);	
	序列化（存储用）	Virtual void Serialize(CArchive& ar);	

七、主要功能及函数描述

7.1 输入

7.1.1 功能

实现用户编辑流程图的输入从中获取有效信息并将之传递给下一模块。

7.1.2 输入项

通过界面上的按钮输入与所编辑的流程图相关的操作。

7.1.3 输出项

下表给出每一个输入对应的输出

输入模块输出项说明

输入	输出
画图形操作	鼠标点击的坐标，所画的是相应图形
选中两个图形画折线的操作	两个图形的输入输出边的中点坐标，所画的是折线



移动	要移动的图形原坐标以及移动后的位置坐标
删除	要删除的图形坐标
修改图形大小颜色线条	要修改的图形坐标，改成的图形坐标，颜色，线条
输入文本	要插入文本的图形，文本内容，字体

#### 7.1.4 算法

1. 用户点击或其他操作
2. 向其他模块传递信息

#### 7.1.5 接口

输入与界面各个按钮的消息宏映射函数 ON\_COMMAND 为接口

数据输出传递以具体的功能实现函数（ex. SetRectangle...）为接口。

### 7.2 画图形模块

#### 7.2.1 功能

输入模块的输出传递给相应新图形模块后，实现在内部进行数据结构的操作并把图形绘制在画布上的功能。

#### 7.2.2 输入项

画图指令，要画图形的位置坐标，图形种类

#### 7.2.3 输出项

在画布上显示要画的图形。

#### 7.2.4 算法

1. 接受相应指令。
2. 创建图形类的实例对象，存入存储结构（链表）。
3. 并调用画图函数，显示相应图形。

### 7.2.5 接口

各个图形类的创建方法（构造函数）并与 CPtrArray 函数相关程序的 OnDraw 函数。

## 7.3 移动模块

### 7.3.1 功能

用户选定图形后拖动，能实现将所画的图形移动到画布的任意位置。

### 7.3.2 输入项

移动指令，要移的图形原坐标，移动后的位置坐标。

### 7.3.3 输出项

将改变位置的图形绘制在画布上。

### 7.3.4 算法

1. 接受用户移动指令
2. 改变内部数据结构中该对象中的相关的数据成员
3. 在画布相应位置显示更新后图形。

### 7.3.5 接口

基本图形类的 SetPoint 方法

图形函数的 OnDraw 和 InvalidateRect

## 7.4 删除图形模块

### 7.4.1 功能

用户选定图形后删除，能实现将用户想删除的图形不在画布上显示。

### 7.4.2 输入项

删除指令，要删除的图形坐标。

### 7.4.3 输出项

在画布上删除选定的图形

### 7.4.4 算法

1. 接受指令。
2. 由获取的数据在内部存储结构中寻找相应的对象并将之删除。
3. 同时调用函数刷新整个画布。

### 7.4.5 接口

与数据结构中 `CPtrArray.At()`

图形显示与 `InvalidateRect`

## 7.5 折线绘制模块

### 7.5.1 功能

用户在选择两个图形后，点击菜单栏上的“自动连接”按钮即可触发该功能，自动连接其图形。

### 7.5.2 输入项目

连线指令，两图形的坐标。

### 7.5.3 输出项目

在屏幕上的两个图形间建立连线关系

### 7.5.4 算法

1. 接受指令。
2. 进行建立折线类的实例化对象。
3. 画布上根据一定规则确定其折点并显示。

### 7.5.5 接口

数据结构方面为构造函数和 `CPtrArray.Add`

显示方面与 `OnDraw` 和 `InvalidateRect`

## 7.6 文字添加模块

### 7.6.1 功能

该程序主要能使用户在流程图中添加文字注解其信息。用户在双击图形或者折线时弹出对话框输入文字并能同时选择文字的字体颜色大小等等，并在单击确定时触发该功能。

### 7.6.2 输入项目

文本以及其各种属性。

### 7.6.3 输出项目

在画布的合适位置显示用户特定的文字。

### 7.6.4 算法

1. 接受用户确定的文字以及各种属性
2. 改变各类中的文字相关的数据
3. 通过文本显示函数在画布相应位置显示文字。

### 7.6.5 接口

图形类的 `IsSelect` 和 `SetChar()`

显示方面 `OnDraw`。

## 7.7 改变图形属性模块

### 7.7.1 功能

用户能够改变每个图形的属性。用户在双击图形时弹出对话框，标签栏选图形属性，改变其图形的一些特征如大小颜色线条等等，单击确定完成输入并触发功能。

### 7.7.2 输入项目

用户确定的各种图形属性信息

### 7.7.3 输出项目

在画布上显示符合用户输入的特征图形。

### 7.7.4 算法

1. 接受用户所确定的图形各种属性
2. 改变该图形对象中的相关的数据成员
3. 通过图形绘制函数在画布相应位置显示更新后图形。

### 7.7.5 接口

图形类的 SetProp()

显示与 InvalidateRect

## 7.8 输出模块

### 7.8.1 功能

将所绘制的图形导出为自有格式以及 PNG 格式。

### 7.8.2 输入项目

目前在画布上的所有信息以及在系统内部的数据结构信息。

### 7.8.3 输出项目

导出为文件并储存在磁盘上。

### 7.8.4 算法

自定义格式，对内部数据结构的遍历以及数据的输出。

对于 PNG 格式的，读取目前在 CDC 上信息调用相应的函数并保存。

### 7.8.5 接口

自定义：Serialize 序列化函数

PNG: Image 相关函数。

## 八、人员分工

（组长）刘国良：统筹整个小组的进度，协调各组员之间的工作。新建图形、保存自定义格式文件功能的实现，对整体的研究。

任天辰：PNG 文件的导出实现，折线的建立折点选择算法的设计。

戴雯：图形选定和移动功能的研究及其实现。

廖黎姿：文字显示功能研究及其实现。

贾珊：文字显示功能研究及其实现。

待分配任务：图形删除、图形属性改变。