目录

[引言 2](#_Toc531772465)

[1.1 编写目的与基线 2](#_Toc531772466)

[1.2 参考资料 2](#_Toc531772467)

[1.3 术语定义及说明 2](#_Toc531772468)

[2 需求及设计决策 2](#_Toc531772469)

[2.1 输入项 2](#_Toc531772470)

[2.1.1 输入项的需求 2](#_Toc531772471)

[2.1.2 输入项的决策 2](#_Toc531772472)

[2.2 输出项 3](#_Toc531772473)

[2.2.1 输出项的需求 3](#_Toc531772474)

[2.2.2 输出项的决策 3](#_Toc531772475)

[2.3 约束条件 3](#_Toc531772476)

[3 配置项详细设计 3](#_Toc531772477)

[3.1 配置项体系结构 3](#_Toc531772478)

[3.1.1 单元划分 3](#_Toc531772479)

[3.1.2 层级结构 7](#_Toc531772480)

[3.2 全局数据结构 8](#_Toc531772481)

[3.2.1 常量 8](#_Toc531772482)

[3.2.2 变量 9](#_Toc531772483)

[3.2.3 数据结构 9](#_Toc531772484)

[3.3 接口设计 9](#_Toc531772485)

[3.3.1 图片浏览与操作组件接口设计 9](#_Toc531772486)

[3.3.2 接口标识与接口图 16](#_Toc531772487)

[3.3.3 XX接口 17](#_Toc531772488)

[3.3.4 XX接口 17](#_Toc531772489)

[3.4 单元详细设计 17](#_Toc531772490)

[3.4.1 XX单元 17](#_Toc531772491)

[3.4.2 XX单元 17](#_Toc531772492)

[3.4.3 XX单元 17](#_Toc531772493)

[3.4.4 XX单元 17](#_Toc531772494)

[4 需求可追溯性 18](#_Toc531772495)

[5 注解 18](#_Toc531772496)

[6 外包要求 18](#_Toc531772497)

[6.1 检测分析模块-图片浏览与操作组件 18](#_Toc531772498)

[6.1.1 组件示意图 18](#_Toc531772499)

[6.1.2 组件功能描述 19](#_Toc531772500)

[6.1.3 组件接口方法描述 21](#_Toc531772501)

# 引言

## 编写目的与基线

*【--说明编写详细设计方案的主要目的。*

*说明书编制的目的是说明一个软件系统各个层次中的每个程序（每个模块或子程序）和数据库系统的设计考虑，为程序员编码提供依据。*

*如果一个软件系统比较简单，层次很少，本文件可以不单独编写，和概要设计说明书中不重复部分合并编写。*

*方案重点是模块的执行流程和数据库系统详细设计的描述。--】*

***该文档针对XX设计的XX模块的XX版本进行编写的***

## 参考资料

*【--列出详细设计报告引用的文献或资料，资料的作者、标题、出版单位和出版日期等信息，必要时说明如何得到这些资料。--】*

## 术语定义及说明

*【--列出本文档中用到的可能会引起混淆的专门术语、定义和缩写词的原文。--】*

# 需求及设计决策

## 输入项

### 输入项的需求

### 输入项的决策

## 输出项

### 输出项的需求

### 输出项的决策

## 约束条件

安全性、保密性、私密性的需求及选择的方法

# 配置项详细设计

## 配置项体系结构

暂无

### 单元划分

通用工具栏主要分两个程序集

#### Ty.Base.WpfBase程序集：

**主要功能：**图片浏览与交互WPF底层

(与通用工具栏组件公用一套底层)

#### Ty.Component.ImageControl

**主要功能：**图片浏览组件实现程序集

1. **ImageOprateCtrEntity.xaml**

**主要功能：**图片浏览组件的接口实现和主要操作窗口



**示例：**红色框部分为此控件



**功能列表：**

IImgOperate接口全部功能实现

展示标定信息列表

展示放大区域

展示全屏窗口

**2、ImageView.xaml**

**主要功能：**具体的图片展示控件和标定控件



**示例：**红色框内为本控件



**功能列表：**

1. 标定缺陷区域
2. 标定样本区域
3. 标定放大区域

**3、ImagePartView.Xaml**

**主要功能：**放大区域展示窗口



**示例：**红色框内为本控件



**功能列表：**

1. 展示放大区域
2. 展示放大区域在全图中的位置

**4、ImageFullScreenWindow.Xaml**

**主要功能：**图片浏览与操作交互的全屏展示页面



**示例：**红色框内为本控件

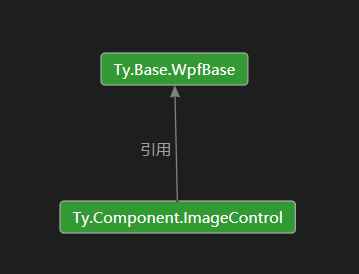


**功能列表：**

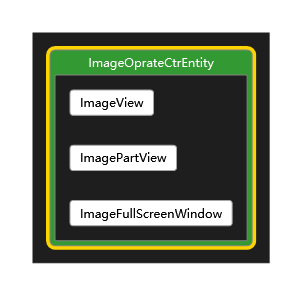
1. 全屏展示
2. 退出全屏展示

### 层级结构

#### 程序集引用层级结构



#### 工具栏控件层架结构关系



**说明：**ImageOprateCtrEntity 为最上层控件，包含了外部接口实现，包含了图片浏览、局部放大和全屏显示的控件

## 全局数据结构

暂无

### 常量

### 变量

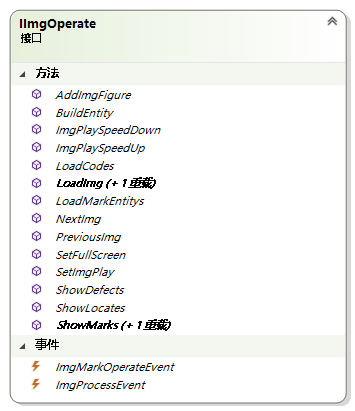
### 数据结构

## 接口设计

### 图片浏览与操作组件接口设计

**外部调用接口定义说明**

**1、图片操作类接口 (名称与定义参考第六部分外包需求中定义)**



**2、图片标定实体**



**3、图片自动播放顺序**



**4、标定实体操作类型**



**5、图片处理类型**



**6、标定内容变更委托，包括：新增、修改、删除**



**7、图片风格化处理委托**



**内部实现接口定义说明**

**1、图片控件内部按钮定义：控件内的操作按钮**



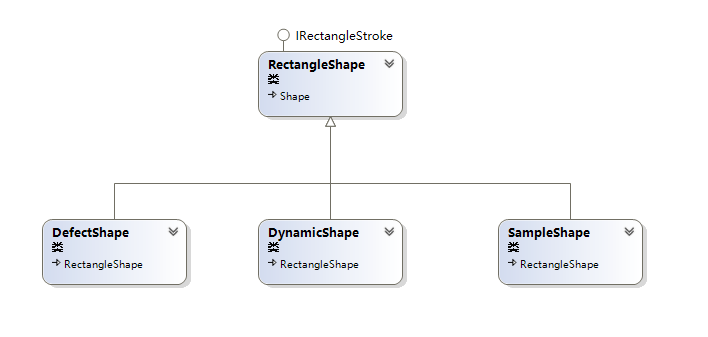
**示例：**红色区域标识此控件



**2、控件内部定义的绑定命令CommandBindings支持**



**3、图片内部绘制矩形图类**



**a、IRectangleStroke：矩形接口定义**



**参数说明：**

Visibllity:绘图区域的可见性

Clear：清除绘图区域

Draw:绘制绘图区域

**b、RectangleShape：矩形底层实现类**



**参数说明：**

Code：标识样本代码

DefiningGeometry：Shape中定义Geometry模型

RenderedGeometry:Shape中渲染Geometry模型

Name:标识样本名称

Postion：标识模型在绘图区域的位置

Rect：标识模型的矩形范围

Clear：清除绘图

Draw：绘制图形

**c、图形中动态展示的矩形区域**



**参数说明：**

HeightMatch：标识绘图区域高的有效范围

WeightMatch：标识回去区域宽的有效范围

BeginMactch：当鼠标按下触发、表示开始检验区域范围是否可用

IsMatch：当鼠标抬起检验、表示绘图范围是否有效

Refesh：当鼠标移动时、更新绘图区域

**示例：**红色区域标识此控件



**d、缺陷的绘图形状**



**示例：**红色区域标识此控件



**e、样本的绘图形状**



**示例：**红色区域内标识此控件



### 接口标识与接口图

暂无

### XX接口

#### 接口的优先级别

#### 接口类型

#### 元素及集合特征

#### 通讯方法

#### 协议的特征

#### 兼容性

### XX接口

## 单元详细设计

暂无

*【与3.1.1一一对应】*

### XX单元

#### 限制条件以及非常规特性

#### 软件逻辑

#### 启动时内部起的作用

#### 控制交予其他配置项的条件

#### 对每个输入的响应及响应时间，包括数据转换、重命名、数据传送操作

#### 操作序列和动态控制序列

#### 异常与错误

### XX单元

### XX单元

### XX单元

# 需求可追溯性

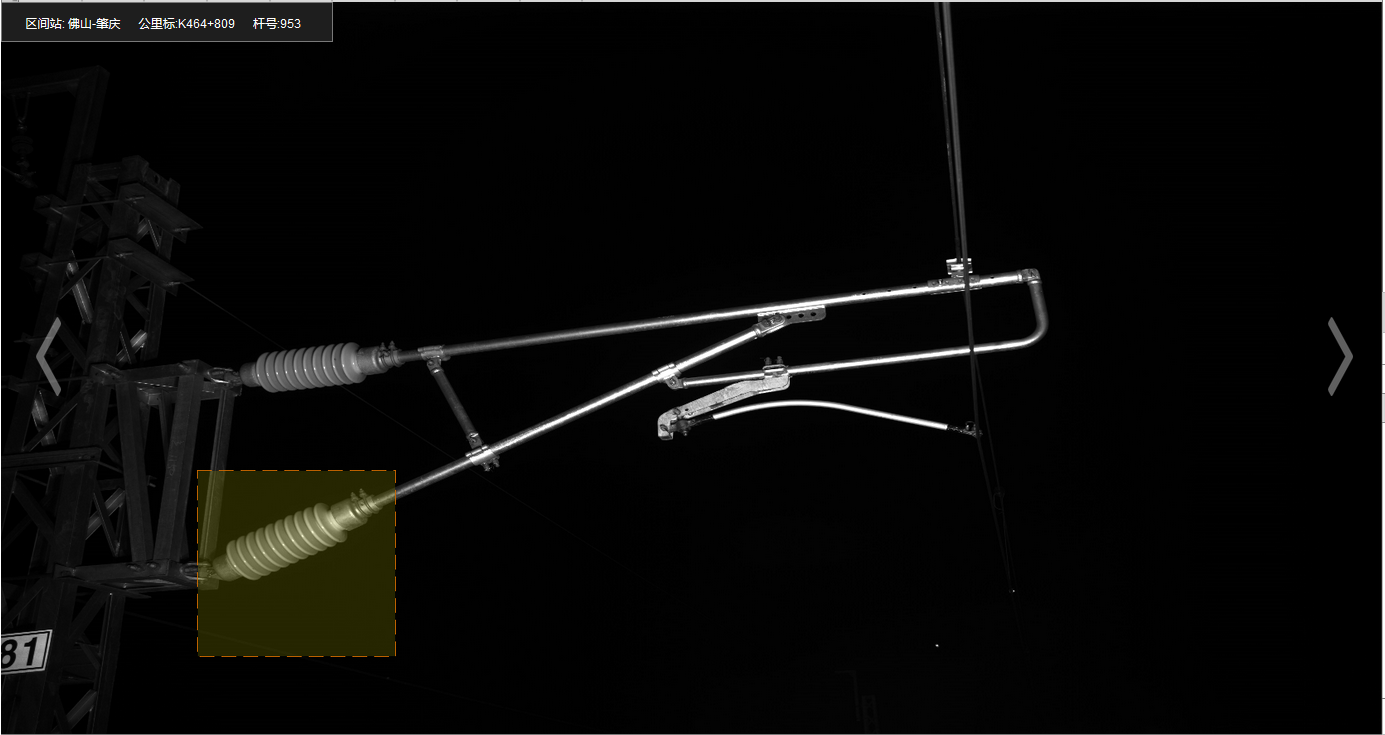
# 注解

*【--本章应包含有助于理解本文档的一般信息（例如原理本章应包含为理解本文档需要的术语和定义，所有缩略语和它们在文档中的含义的字母序列表--】*

# 外包要求

## 检测分析模块-图片浏览与操作组件

### 组件示意图





### 组件功能描述

1. 加载并展示图片；
2. 可以在图片上框选一块矩形区域，而后**选择将这块矩形区域标定为区域识别样本**。选择区域识别样本标定选项后，在矩形框合适位置（一般是矩形框下方，当屏幕上矩形框下方空间不足时，展示在矩形框右侧或者上方等）弹出样本编辑框，提供下拉检索框，确定标注类型代码；
3. 可以在图片上框选一块矩形区域，而后选择将这块矩形区域标定为缺陷，选择区域识别样本标定选项后，在矩形框合适位置（一般是矩形框下方，当屏幕上矩形框下方空间不足时，展示在矩形框右侧或者上方等）弹出样本编辑框，提供下拉检索框，确定缺陷类型代码；
4. 默认情况下，所有标定内容都展示在图片之上，且样本区域标定和缺陷标定采用不同颜色的标定框展示；
5. 当鼠标移动到标定框内部时，标定框高亮，右侧的缺陷列表行高亮（接口），同时展示该标定的详细信息（标定类型，标定内容，标定代码等）；鼠标移出后，标定详细内容隐藏，只展示标定框。
6. 提供接口，支持只展示区域标定信息；
7. 提供接口，支持只展示缺陷标定信息；
8. 提供接口，支持展示指定类型（通过标定时选择的代码指定）的标定信息；
9. 选定的标定信息支持修改和删除；
10. 提供接口，支持加载历史标定信息；
11. 可以在图片上框选一块矩形区域，而后**选择将这块矩形区域放大展示**。放大展示方式：框定矩形区域长宽按照同等比例放大到当前图片控件展示，图片居中，无法填充的位置显示为黑色；
12. 放大展示局部区域时，需要同时展示放大区域在原图中的位置。
13. 框定的矩形区域如果长/宽小于10px或面积小于50px2，则自动放弃本次框定内容。该标准中定义的**指标值通过配置读取**，可在不修改代码的情况下修改指标。
14. 提供接口，支持由局部放大展示模式切换为全图展示模式；
15. 提供接口，支持图片风格化处理请求（滤镜、对比度、曝光补偿、夜视、锐化、边缘锐化等处理）；
16. 图片展示控件左上放支持展示图片信息，要展示的信息通过配置方式添加以及调整。不同指标的展示方式，颜色可配置。默认情况下采用信息展示框背景为黑色，字体颜色为白色的配置；
17. 具有通用图片浏览工具的基础功能：前一张，后一张，提供快捷键接口。
18. 提供全屏展示切换功能；

### 组件接口方法描述

/// <summary>

/// 标定内容变更委托，包括：新增、修改、删除

/// </summary>

/// <param name="markEntity">标定实体，实体内包含了实体是新增、修改还是删除的相关信息</param>

public delegate void ImgMarkHandler(ImgMarkEntity markEntity);

/// <summary>

/// 图片风格化处理委托，

/// </summary>

/// <param name="imgPath">图片路径</param>

/// <param name="imgProcessType">风格化处理类型</param>

public delegate void ImgProcessHandler(string imgPath, ImgProcessType imgProcessType);

/// <summary>

/// 图片操作类

/// </summary>

public interface IImgOperate

{

/// <summary>

/// 创建图片浏览展示控件

/// </summary>

/// <returns></returns>

ImgOperateCtrlEntity BuildEntity();

#region 数据加载方法

/// <summary>

/// 图片控件加载标定使用代码和代码名称

/// </summary>

/// <param name="codeDic">key：代码（唯一） value：代码名称</param>

void LoadCodes(Dictionary<string, string> codeDic);

/// <summary>

/// 加载图片

/// </summary>

/// <param name="imgPath">图片路径</param>

void LoadImg(string imgPath);

/// <summary>

/// 加载图片列表

/// </summary>

/// <param name="imgPathes">图片路径列表</param>

void LoadImg(List<string> imgPathes);

/// <summary>

/// 加载图片的标定信息

/// </summary>

/// <param name="markEntityList">图片已标定内容</param>

void LoadMarkEntitys(List<ImgMarkEntity> markEntityList);

#endregion

#region 事件

/// <summary>

/// 新增/修改/删除图片标定事件

/// </summary>

public event ImgMarkHandler ImgMarkOperateEvent;

/// <summary>

/// 图片风格化处理事件

/// </summary>

public event ImgProcessHandler ImgProcessEvent;

#endregion

#region 设置方法

/// <summary>

/// 展示全部缺陷标注

/// </summary>

public void ShowDefects();

/// <summary>

/// 展示全部区域定位标注

/// </summary>

public void ShowLocates();

/// <summary>

/// 展示全部标注，包括缺陷和定位标注

/// </summary>

public void ShowMarks();

/// <summary>

/// 展示指定编码标注

/// </summary>

/// <param name="markCodes"></param>

public void ShowMarks(List<string> markCodes);

/// <summary>

/// 是否全屏展示

/// </summary>

/// <param name="isFullScreen">true：全屏展示 false：正常展示</param>

public void SetFullScreen(bool isFullScreen);

/// <summary>

/// 图片详细信息展示

/// </summary>

/// <param name="imgFigures">要展示的指标和相应值</param>

public void AddImgFigure(Dictionary<string, string> imgFigures);

/// <summary>

/// 前一张

/// </summary>

public void PreviousImg();

/// <summary>

/// 下一张

/// </summary>

public void NextImg();

/// <summary>

/// 设置图片播放

/// </summary>

/// <param name="imgPlayMode">正序，倒叙，停止播放</param>

public void SetImgPlay(ImgPlayMode imgPlayMode);

/// <summary>

/// 加快图片播放速度

/// </summary>

public void ImgPlaySpeedUp();

/// <summary>

/// 减慢图片播放速度

/// </summary>

public void ImgPlaySpeedDown();

#endregion

}

*其中，*标定实体操作类型*定义如下：*

/// <summary>

/// 标定实体操作类型

/// </summary>

public enum ImgMarkOperateType

{

/// <summary>

/// 新增标定

/// </summary>

Insert=0,

/// <summary>

/// 修改已有标定

/// </summary>

Update=1,

/// <summary>

/// 删除标定

/// </summary>

Delete=2,

}

*图片处理类型如下：*

/// <summary>

/// 图片处理类型

/// </summary>

public enum ImgProcessType

{

滤镜 = 0,

对比度 = 1,

曝光补偿 = 2,

夜视 = 3,

锐化 = 4,

边缘锐化 = 5,

}

*图片自动播放顺序枚举值如下：*

/// <summary>

/// 图片自动播放顺序

/// </summary>

public enum ImgPlayMode

{

正序 = 0,

倒叙 = 1,

停止播放 = 2,

}

*图片标定实体定义如下：*

/// <summary>

/// 图片标定实体

/// </summary>

public class ImgMarkEntity

{

/// <summary>

/// 由大括号包起来的全部大写的36位guidID，与表中的记录保持一致

/// </summary>

public string ID;

/// <summary>

/// 标定矩形框左上角X坐标

/// </summary>

public int X;

/// <summary>

/// 标定矩形框左上角Y坐标

/// </summary>

public int Y;

/// <summary>

/// 标定矩形框高度

/// </summary>

public int Height;

/// <summary>

/// 标定矩形框宽度

/// </summary>

public int Width;

/// <summary>

/// 标定类型代码（代码唯一，从代码表中查询）

/// </summary>

public string Code;

/// <summary>

/// 标定类型名称（通过标定类型代码到表中查询）

/// </summary>

public string Name;

/// <summary>

/// 标注操作类型，枚举值，默认是新增

/// </summary>

ImgMarkOperateType markOperateType;

}