

组态式人机界面编辑与仿真技术^①

张琼琼^{1,2}, 马跃², 王志成³

¹(中国科学院大学, 北京 100049)

²(中国科学院沈阳计算技术研究所 高档数控国家工程研究中心, 沈阳 110168)

³(沈阳高精数控有限公司, 沈阳 110168)

摘要: 人机界面在工业控制领域应用非常广泛. 针对传统的人机界面画面固定、可扩展性差的问题, 对组态式人机界面编辑与仿真技术进行了研究, 以实现画面的可定制. 首先, 介绍国内外组态式人机界面的现状及技术并设计了组态式人机界面的系统结构; 然后, 采用面向对象技术和设计模式相结合的组件技术实现画面编辑功能, 同时, 完成了具备用户更好体验的仿真功能方案设计; 最后, 采用融合了反射技术的工厂模式实现对不同厂家的兼容性设计. 该研究成果可用于指导组态式人机界面配置与仿真工具的技术研究和产品开发.

关键词: 组态; 人机界面; 设计模式; 画面编辑; 仿真

Editing and Simulation Technology for Configurable HCI

ZHANG Qiong-Qiong^{1,2}, MA Yue², WANG Zhi-Cheng³

¹(University of Chinese Academy of Sciences, Beijing 100049, China)

²(National Engineering Research Center for High-End CNC, Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

³(Shenyang Golding Tech.Co.Ltd., Shenyang 110168, China)

Abstract: HCI(Human-computer Interface) is widely applied in industrial control filed. In view of the traditional HCI picture which is fixed and has poor scalability problems, so editing and simulation technology of a configurable HCI is researched in this paper, which enables users to easily custom pictures for the HCI controller. Firstly, the current situation of domestic and foreign configuration and technology of HCI is introduced and the configuration of the HCI system structure is researched. Further, component technology making use of object-oriented technology and design patterns has been designed to achieve picture editing functions. At the same time, program design of simulation function is completed, which possesses a better user experience. Finally, a factory which is combined with reflection technology pattern is used to design for compatibility with different manufacturers. The research results can be used to guide the technology research and product development for configuration and simulation tools of the configurable HCI.

Key words: configuration; HMI; design patterns; picture editing; simulation

人机界面(HCI, Human-computer Interface)也称作用户界面, 最早从人机工程学发展而来. 随着计算机技术的迅猛发展, 人机界面的发展也快而广泛. 由于人机界面是直接为用户打交道的, 所以用户体验成为衡量人机界面的唯一指标. 近年来, 随着用户体验的不断升级, 目前的用户界面主要有: 批处理人机界面、命令行人机界面、图形人机界面、自然用户界面及面向未来的社会人机界面^[1], 但在工控领域的人机界面

多停留在图形人机界面, 甚至是命令行人机界面.

组态软件又称组态监控软件是工控领域数据采集与过程控制的专用软件^[2]. 组态软件大约在 80 年代中期出现在国外, 之后传入国内也渐渐发展了 20 多年. 国外的组态技术相对成熟, 但种种原因导致组态软件在中国并不受欢迎^[3]: 初始, 多数企业对组态软件并不充分了解, 也很少有项目跟进, 都宁愿投入人力物力针对具体项目做长周期的编程开发而不采用组态式

① 基金项目: 国家科技重大专项(2013ZX04012-021)

收稿时间: 2014-08-11; 收到修改稿时间: 2014-10-15

编程;况且,从国外引进的组态软件价格相对较高,对中小企业是个不小的风险;尤其在工控领域的工控软件常需要很高的可靠性,而组态软件技术相对不成熟,尚无法检测经长期使用是否自主可控,所以多数中小企业还在观望中;最重要的原因是工控软件一般功能专一,需求的变更不是很频繁.近年来,随着工控软件的快速发展,投入的资源越来越多,传统的工业控制人机界面已经无法满足用户不断增长的需求,而且不少中小企业还处在人机界面发展的保守阶段.据此,提出采用组态思想开发人机界面的编辑与仿真工具,旨在能为中小企业人机界面的配置与仿真工具的开发提供技术支持.

1 国内外组态式人机界面现状及技术

目前国内外最具代表性的组态软件^[4]如下所列:

In Touch: 是全球工业自动化软件的领先供应商.早期的 In Touch 软件采用 DDE 方式与驱动程序通信,性能较差,最新的 InTouch10 版以上的完全基于与 32 位或 64 位的 Windows 平台,并且提供 OPC(OLE for Process Control)支持,在控制领域得到了广泛使用.

Win CC: 西门子(Siemens)的 Win CC 是一套完备的组态开发环境,提供了类 C 语言的脚本以及调试环境,且内嵌 OPC 支持,可对分布式系统进行组态.

组态王: 是国内较早出现的组态软件产品之一,主要实现画面、数据、动画,并能充分利用 Windows 图形编辑功能,方便制作监控画面,动画显示设备状态,具有报警、实时趋势曲线等.且提供对仿真功能的支持.

世纪星组态软件: 拥有超级图形制作功能,被广泛用作绘图工具软件,拥有独特的功能,特别是图形设计、语音报警、组态报表及个性化功能在国内外组态行业中处于领先地位.

结合国内外组态式人机界面的相关技术,本文设计了组态式人机界面的系统结构,该系统结构由人机界面控制器和人机界面配置与仿真工具组成,并详细设计了人机界面配置与仿真工具的内部结构.该工具采用 USB 直接通信方式将用户定制的画面下载到人机界面控制器中.该工具采用的技术主要为人机界面编辑技术和人机界面仿真技术.其中,编辑技术主要在 MVC 模式的应用下研究功能组件的定制、画面编辑及兼容性的设计.而仿真技术则主要以用户体验为

目标,提出相应的仿真功能实现方案.

2 组态式人机界面的系统结构

本文主要研究用于沈阳高精数控研发的人机界面控制器的人机界面配置与仿真工具.通过需求分析得出该工具的开发主要分为三部分:编辑、仿真、下载.组态式人机界面的系统结构如图 1 所示:

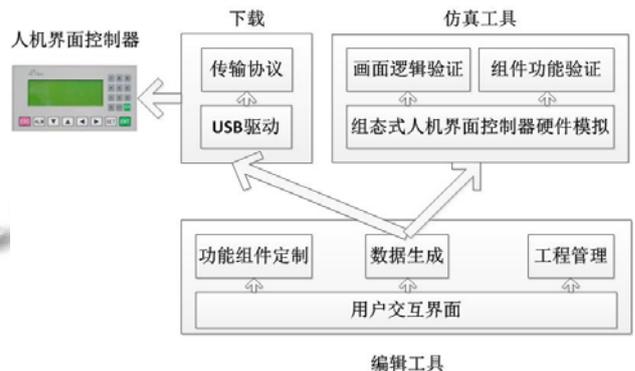


图 1 组态式人机界面系统结构

其中,编辑工具子系统在用户交互界面的基础上实现功能组件的定制、数据生成、工程管理等功能;而下载功能职责主要是通过 USB 数据线与组态式人机界面控制器相连后,将编辑工具子系统生成的数据下载到组态式人机界面控制器中,显然,下载子系统需要设计传输协议并实现 USB 驱动;同样的,仿真工具子系统也要利用这些数据实现仿真功能,从而有效的对画面逻辑及组件功能进行验证,而这两方面的验证需要对组态式人机界面控制器硬件逻辑进行模拟.

尤其在下载数据时,目前流行的通信方式主要有三种:串口通信、虚拟串口通信、USB 直接通信.三者各有优缺点:串口通信相对简单但不够灵活,通用性不强,正因其信号简单,目前仍普遍使用于 PC 机和众多工业设备中;随着 PC 与外围设备越来越少地使用串口,更多的设备偏向使用 USB 通信,而 USB 接口的最大特点就是易于使用、速度快、通用于多种常用设备;相比较 USB 直接通信而言,虚拟串口通信无需另外开发 USB 驱动,而且兼有串口通信的优点,该通信方式在实际的生产应用中拥有很好的借鉴和应用价值,但很难扩展到其他常见的外围设备中而用 USB 直接通信则很容易进行扩展,因此设计组态式人机界面配置与仿真工具的下载功能时使用 USB 直接通信.数据传输流程如图 2 所示.

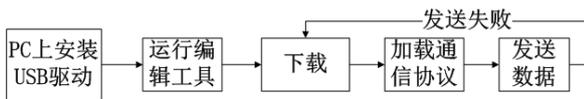


图 2 数据传输流程图

3 人机界面编辑技术

设计面向对象软件比较困难，而设计可复用的面向对象软件就更加困难，不仅要突出软件的功能性，也要对未来的问题和需求有足够的通用性^[5]。

面向对象理论^[6]的提出无疑为软件的开发注入新的生命。通过封装、继承、多态大大降低程序的耦合度，而采用合适的设计模式则使程序更加灵活，修改容易，重要的是拥有良好的复用效果，因此，组态式人机界面的设计采用面向对象的方式，并加入相应的设计模式，使之满足用户的需求。

在设计用户界面时采用各种各样的设计模型，目前最流行的是 MVC(Model-View-Controller) 模型。MVC 的作用是如何把用户界面分离成用来实现每一个特殊功能的组件，进而提高灵活性和复用性，其中，Model 定义了应用对象，View 是在屏幕上的表示形式，Controller 主要定义了用户界面对用户输入操作的响应方式^{[5][7]}。MVC 模型之间的交互关系如图 3 所示。

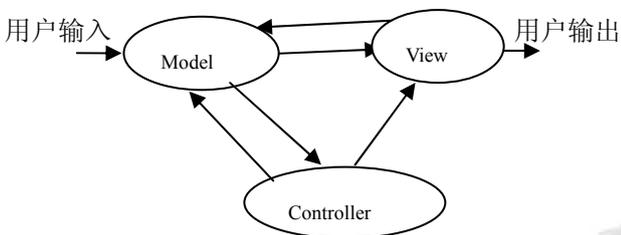


图 3 MVC 模型

虽然 MVC 模型很好的将应用与界面分离，但也有它的缺点，模型的引入使得系统的结构变得复杂，但一旦设计好后系统的耦合度低，复用性好，可维护性高。因些在该组态式人机界面编辑与仿真系统中采用 MVC 模式设计，主要考虑三方面的典型设计：功能组件的定制(Model)、画面的编辑(Controller)、兼容性的设计(View)。其中功能组件的定制是组态式人机界面的最为核心的技术^[8]，画面的编辑主要应用用户定制的功能组件进行画面的绘制，为使该人机界面调试与仿真工具更容易扩展，所以要兼容不同厂家的 PLC 产品。

3.1 功能组件的定制

每一个由用户定制的组件往往有多种显示方式，

如功能键就有如下显示状态: ESC 键、ALM 键、四个方向键、SET 键、ENT 键、0-9 数字键，除此，还有倍角、反相、手形、隐形四种状态，组合起来大概有近百种可能状态。如果直接绘制不同的状态将使代码变得杂乱且难以维护，因此可以将不同的显示状态封装成不同的策略类，可以采用策略模式与工厂模式结合^[5]的方式使代码容易扩展维护。

具体的模式设计如图 4 所示。

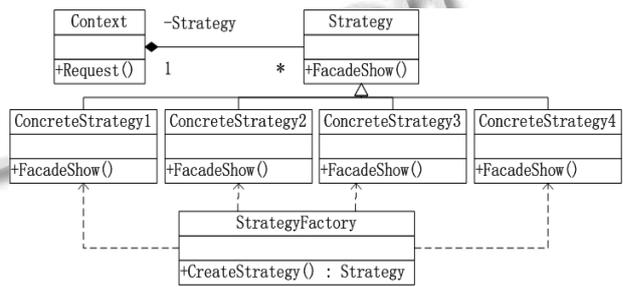


图 4 策略与工厂模式

关键代码如下：

```

abstract class Strategy
{
    //功能组件的抽象，显示不同外观
    public abstract void FacadeShow();
}

class ConcreteStrategy1:Strategy
{//具体策略类
    public override void FacadeShow()
    {
        //在组件中画具体的外观
        drawFacade();
    }
}

其他策略类与此相似

class StrategyFactory
{//策略工厂类
    public static Strategy createStrategy(string type)
    {
        Strategy strategy =null;
        switch(type)
        {
            case "type1":
                strategy = new ConcreteStrategy1();
            case "type2":
        }
    }
}
  
```

```

        strategy = new ConcreteStrategy2());
        .....
    }
}
}
}
class Context
{ //上下文类, 用于动态请求不同的策略
    Context(string type)
    {
        Strategy strategy = StrategyFactory.createStrategy(type);
    }
    strategy.FacadeShow();
}

```

采用策略与工厂模式使得在主窗体应用编程时只需要提供组件的不同类型参数, 由 Context 类动态选择执行哪个子类的策略. 如果需求变更, 只需要增加新的 ConcreteStrategy 类, 并在工厂中添加一个 Case 判断即可, 在主窗体程序中代码无需更改. 不仅如此, 这两种模式结合的运用简化了测试, 可针对每个具体策略进行单独测试, 修正其中的一个策略不影响其他的策略. 策略模式适用于完成相同的工作, 只是显示方式不同的情况, 因此开发者可以用相同的方式调用所有的策略, 简化不同策略间的耦合, 而在策略模式的基础上加入工厂模式则进一步大大简化主体程序的代码, 将繁杂的判断分支移到工厂中. 使客户端代码尽量隐藏类的信息, 以最少的信息量完成必要的功能.

3.2 画面的编辑

组态式人机界面配置与仿真工具的编辑模块主要研究画面的编辑, 该模块利用用户定制的组件编辑成一副副画面, 同时可以设置每个组件的参数. 这一模块的设计难点在于如何将组件与其参数关联. 首先, 需要通过链表的形式分别存储画面、画面属性、控件属性; 然后, 每当新加入一个组件需要设置该组件的 Tag 属性与在控件属性链表中的该组件属性相关联, 在设置参数时只需要单击相应的组件即可完成设置, 否则, 无法找到相应的属性; 这样做还有一个好处就是: 组件的外观总是与其部分属性有直接关系, 一旦这些属性变化, 组件的显示外观就要改变, 这时只需要通过 Tag 找到原组件的属性记录下来, 然后将外观改变后产生的新组件与其绑定即可.

完成画面编辑功能的关键代码如下:

```
List<PictureBox> listPBox = new List<PictureBox>();//记录画
```

面

```
List<GroupBox> listGBox = new List<GroupBox>();//记录画面属性
```

```
List<GroupBox> listGBox1 = new List<GroupBox>();//记录组件属性
```

```
//每当产生新组件, 则通过 tag 绑定该组件的属性, component 为组件对象
```

```
component.Tag = listGBox1[listGBox1.Count - 1];
```

```
//如果只需要更改组件外观, 将 tag 绑定到组件删除前属性在链表中位置
```

```
listPBox[currentRowNum].Controls.Add(component);
```

```
component.Tag = listGBox1[deleteFrontIndex];
```

```
//deleteFrontIndex 为组件删除前在链表中的位置
```

3.3 兼容性设计

由于人机界面主要与各类 PLC 配合使用文字或指示灯等形式监视、修改 PLC 内部寄存器或继电器的数值及状态, 而生产 PLC 的厂家采用的标准不同, 因此组态式人机界面的开发需要兼容各 PLC 厂家的产品^[9]. 人机界面中用户定制的功能组件属性除了与 PLC 相关的寄存器地址、位数等不太一致外, 其他的参数基本相同, 因此只需要封装不同厂家有差异的部分, 系统运行时做出合适的选择, 这一点只需工厂模式与反射技术^[5]结合即可做到.

具体的设计模式图如图 5 所示.

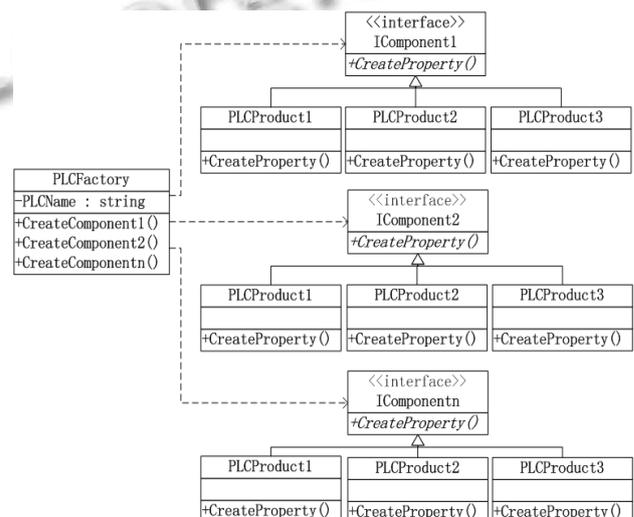


图 5 采用反射技术的工厂模式

关键代码如下:

```

//引用反射技术的命名空间
using System.Reflection;
class PLCFactory
{
//动态构建指定工厂的各组件
    private static readonly string AssemblyName="工厂模式";
    private static readonly string PLCName="三菱";
    public static IComponent1 CreateComponent1()
    {
    string className = AssemblyName + "." + PLCName + "Component1";
    return (IComponent1)Assembly.Load(AssemblyName).CreateInstance(className);//反射技术加载具体的工厂
    }
    public static IComponent2 CreateComponent2()
    {
    string className = AssemblyName + "." + PLCName + "Component2";
    return (IComponent2)Assembly.Load(AssemblyName).CreateInstance(className);
    }
    .....
}
}

```

采用反射技术后，如果要增加新的工厂类，在主体程序中只需要修改 private static readonly string PLCName="newFactoryName"即可，若增加新的组件，只需增加与其相关的类并修改 PLCFactory 类，增加 public static IComponentn CreateComponentn()方法即可。

采用反射技术的目的就是为了消除 switch 或 if, 从而消除分支判断带来的耦合，因此在功能组件定制的模式应用中也可以加入反射技术以消除分支判断。

4 人机界面仿真技术

仿真技术^[10]作为一门新兴的高技术，随着计算机技术的发展，其应用领域越来越广，作用也越来越大，因此，仿真是不可或缺的，通过仿真可以减少损失、节约经费、提高产品质量等优点。除此，为了提高人机界面编辑工具的开发效率，有时设计者在完成界面设计后，不知道结果是否和自己的初衷相符，据此，提出

开发系统仿真工具。该仿真工具主要是通过研究人机界面控制器的控制过程来模拟其相关操作(口令更改、画面跳转、组件的功能等)。在模拟组态式人机界面控制器硬件环境的基础上，该仿真工具实现画面逻辑和组件功能的验证，不通过通讯的方式而是直接在 PC 机上仿真通信，产生所需的数据和状态信息。

仿真工具的功能结构如图 6 所示。

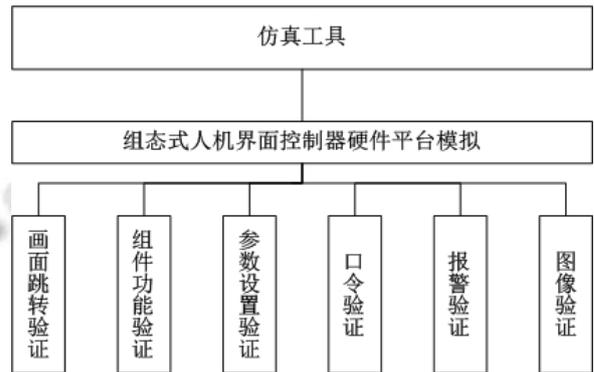


图 6 仿真工具结构图

这里给出简单的仿真实例，用于验证画面跳转、组件功能、口令及图像的验证，而参数设置与报警的验证需要获取工控机的实时数据。具体的仿真结果如图 7 所示：



图 7 仿真实例

仿真过程：首先，仿真开始会进入初始画面如图 6(a)，如果设置有口令，则点击任何按键提示请输入口

令, 点击 ENT 进入口令界面如图 6(b), 选择 1 可以输入口令如图 6(c), 如果口令正确进入界面 6(d), 按任意键返回主界面, 然后通过左键进入界面 6(e), 按右键进入界面 6(f).

该仿真工具能够从逻辑上验证用户在真实组态人机界面上的操作, 如果要动态实时的显示具体产品产生的数据, 可以通过 LabView 构建控制器的硬件监视系统^[11], 该监视系统与人机界面控制器相连获取实时数据, 然后将这些数据传递到仿真工具中, 从而实现更真实的仿真模拟, 进而使用户拥有更好的用户体验.

3 结语

本文利用 VC 开发了友好的用户交互界面, 方便用户使用. 通过研究画面编辑的功能组件模块与兼容性的设计技术, 实现了以组态的方式编辑画面的编辑工具. 该工具具有良好的扩展性、复用性、可靠性, 且对用户专业知识要求不高. 而仿真工具的实现, 则能够使用户及时校验二次开发后的产品, 大大的提高工作效率. 目前, 该组态式人机界面配置与仿真工具已成功应用于沈阳高精数控研发的人机界面控制器, 并取得良好的应用效果.

进一步, 利用这些技术可以实现更为复杂的触摸屏系列的人机界面控制器产品中, 也可以利用 LabView 工具实现联机仿真技术^[12].

参考文献

1 魏玮, 宫晓东. 基于用户体验的人机界面发展趋势. 北京航空

航天大学学报, 2011, 37(7): 868-871.

2 周浩, 黄雄峰, 李科, 付子鑫. 工业人机界面技术发展及其应用综述. 可编程控制器与工厂自动化(PLC FA), 2012, (10): 37-40.

3 王玉铎. 浅谈工控机及组态控制技术. 科技与企业, 2013 (23): 179.

4 石冠先. 基于 VC 的监控组态软件开发及其应用[学位论文]. 北京: 北京工业大学, 2013.

5 Gamma E, Helm R, Johnson R, et al. 设计模型. 李英军, 马晓星, 蔡敏, 刘建中译. 北京: 机械工业出版社, 2011.

6 段莹, 刘燕美. 浅谈面向对象技术在人机界面中的设计. 电脑知识与技术, 2013, 5: 1071-1073.

7 Pinheiro da Silva P, Griffiths T, Paton NW. Generating user interface code in a model based user interface development environment. Proc. of the Working Conference on Advanced Visual Interfaces. ACM. 2000. 155-160.

8 陈云云, 马捷中, 蒋泽军. 基于插件的组态软件设计与实现. 计算机测量与控制, 2013, 21(2): 436-438.

9 胡兆勇, 何汉武, 秦兵. 基于 Visual C++ 的虚拟 PLC 仿真软件开发. 广东工业大学学报, 2009, 1: 40-43.

10 郑健. PLC 仿真技术研究[学位论文]. 青岛: 青岛大学, 2007.

11 王欢. 基于 LabVIEW 的实时控制与仿真技术应用研究[学位论文]. 南京: 南京航空航天大学, 2009.

12 Popović B, Popović N, Mijić D, et al. Remote control of laboratory equipment for basic electronics courses: A LabVIEW - based implementation. Computer Applications in Engineering Education, 2013, 21(S1): E110-E120.