

# 数控系统人机界面二次开发平台的设计<sup>①</sup>

吴慧杰<sup>1,2</sup>, 王 品<sup>2,3</sup>

<sup>1</sup>(中国科学院大学, 北京 100049)

<sup>2</sup>(中国科学院 沈阳计算技术研究所, 沈阳 110168)

<sup>3</sup>(沈阳高精数控智能技术股份有限公司, 沈阳 110168)

**摘 要:** 数控系统自带的标准化的人机界面, 展现给用户大多是同一种形式, 不能满足用户特殊定制需求以及专用的数控系统的界面需求. 在分析用户的界面定制需求的基础上, 提出基于 Qt 的 Graphics View Framework 及 model/view 架构的可视化数控系统人机界面二次开发平台设计方法. 首先采用面向对象的二次开发方法将界面所需的组件注册到平台并绑定对应的属性对象, 然后采用参数化的二次开发方法完成界面的定制, 最后用 XML 文件作为配置文件描述定制的图形界面, 在开发平台与数控系统之间传递界面数据. 二次开发平台主要包括 PC 端配置平台以及 NC 端解析平台两个模块. 实验结果表明平台在数控系统人机界面的专项定制及复用方面取得良好的效果, 提高了数控系统的人机界面开发效率.

**关键词:** 数控系统; 人机界面; 二次开发; XML 配置文件

## Design of Secondary Development Platform for the CNC System Human Machine Interface

WU Hui-Jie<sup>1,2</sup>, WANG Pin<sup>2,3</sup>

<sup>1</sup>(University of Chinese Academy of Sciences, Beijing 100049, China)

<sup>2</sup>(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

<sup>3</sup>(Shenyang Golding NC Intelligence Tech Co. Ltd., Shenyang 110168, China)

**Abstract:** The CNC System with standardization HMI, mostly presents the same kind of form and cannot meet customer and dedicated CNC System's special requirements. Based on the analysis of the user interface customization demand, this paper designs a visual secondary development platform for the CNC System HMI based on Qt Graphics View Framework and model/view architecture. Firstly it registers components to platform and binds the corresponding property objects by using the object oriented secondary development method then completes the interface customization through the parameterization. Finally it uses the XML file as the configuration file to describe the graphical interface and transfer data between platform and CNC System. The secondary development platform mainly includes two modules which are PC configuration platform and NC analytic platform. The experimental results show that the platform has achieved good results in the special customization and reuse of CNC System's HMI, and the development efficiency of HMI is improved.

**Key words:** CNC System; HMI; secondary development; XML configuration file

随着工业化进程的不断推进, 一个国家的数控化率的高低已经成为衡量其机械工业技术水平的一个重要指标. 开放式数控系统要求开发出方便扩展、功能柔性并且对用户开放的数控系统, 便于用户基于其开放平台结合专有工艺进行二次开发. 人机界面(Human

Machine Interface, 简称 HMI)是人与数控系统之间传递、交换信息的媒介和对话窗口, 是数控系统的重要组成部分. 数控系统的人机界面发展也集中在专用机床的应用、特殊结构的引用以及用户操作简化等方面. 目前国内的数控系统自带的标准化的人机界面, 展

① 基金项目:“高档数控机床与基础制造装备”国家科技重大专项,航空发动机微孔冷加工超快激光数控机床(2014ZX04001041)

收稿时间:2016-06-20;收到修改稿时间:2016-07-25 [doi:10.15888/j.cnki.csa.005646]

现给用户的大多是同一种形式<sup>[1]</sup>, 这种标准化的人机界面对许多专用数控系统都不是最优化的设计<sup>[2]</sup>, 已经不能满足用户的需求. 其一, 不能满足用户对特定加工工艺的个性化定制需求; 其二, 如果用户在系统的使用过程中提出新的功能需求, 新功能的开发周期往往不能及时响应市场需求. 而且随着系统规模不断扩大、系统功能不断增加完善, 重新编写界面代码的时间和成本都将成倍递增. 因此, 需要开发一种数控系统人机界面二次开发平台, 用户基于此平台可以根据特定的功能需求快速便捷地开发定制人机界面, 降低开发和维护成本. 不仅满足用户对数控系统的专业化、个性化需求, 而且满足开放式数控系统的便于扩展、对用户开放等需求.

本文提出的数控系统人机界面二次开发平台, 该平台具有可视化、易扩展、易复用、易维护等特性. 用户基于平台可以自己配置人机界面上传到数控系统服务器, 用户所见即所得, 不需要学习任何的编程语言. 开发人员也可以利用平台进行系统界面的修改、升级, 不需要重新编写代码, 大大降低开发成本, 提高开发效率.

## 1 人机界面二次开发平台的结构

本文设计的数控系统人机界面二次开发平台主要包括 PC 端配置平台和 NC 端解析平台两个模块. 下面对两个模块作具体的描述.

PC 端配置平台是基于 Qt 的图形视图架构开发的可视化的界面配置软件, 用于注册并配置数控系统人机界面的组件及其属性, 并自动生成 XML 配置文件. 主要包括组件注册层、界面定制层、文件生成层:

(1) 组件注册层: 用于注册配置人机界面需要的系统的组件, 主要分为三类: 显示组件、编辑组件以及事件组件. 注册各种类型的组件到系统的组件盒, 并且将各种组件对应的不同的属性对象绑定到组件中. 在该层提供用户扩展系统所需的二次开发接口框架, 提高系统的可复用性与可扩展性.

(2) 界面定制层: 用于定制用户所需的个性化的、专用化的数控系统人机界面. 并且配置各个组件的属性值, 存储到组件绑定的属性对象中.

(3) XML 配置文件用于描述人机界面的图形化数据, 完成 PC 端配置平台和 NC 端解析平台的数据传递. XML 是纯数据描述, 与编程语言、操作系统或传输协

议均无关系, 因此可以将数据从以代码为中心的基础结构所产生的约束中解放出来, 可以做到将图形用户界面与结构化数据分割开来<sup>[3]</sup>, 具有良好的可扩展性和适用性, 是用来描述图形用户界面的最佳载体<sup>[4]</sup>. 基于 XML 的界面自动生成比基于模型驱动的软件界面更加灵活多变, 可以同时运行在不同的平台上, 而且对于开发后用户界面的控制也更加容易.

NC 端解析平台读取并解析配置文件数据, 完成数据与原数控系统的交互并显示 PC 端配置的人机界面. 包括文件解析层、调度管理层、通信服务层、系统接口层.

如图 1 所示, 为平台总体框架图. PC 端配置平台注册组件、属性、属性编辑器, 然后从注册的组件中选择需要的组件在页面中进行定制, 根据不同的数控系统人机界面需求定制不同的人机界面, 然后配置各个组件的属性值, 将配置完毕的人机界面信息保存到 XML 配置文件中; 通过以太网将配置文件发送到 NC 端解析平台; 数控系统解析配置文件, 根据属性值显示人机界面, 最终达到显示用户定制的人机界面的目的.

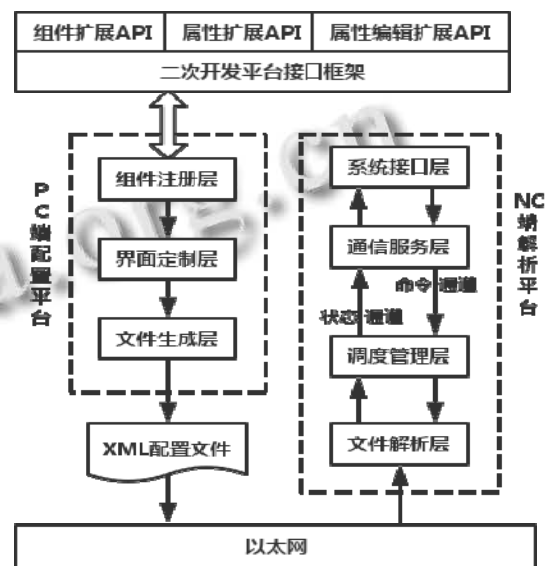


图 1 平台总体框架图

## 2 主要模块的设计方法

### 2.1 PC 端配置平台模块

PC 端配置平台是基于 Qt 的图形视图框架开发的可视化的应用程序, 工作在 PC 机上. 用户只需要从组件盒选择要添加的组件, 并且配置好组件的位置、名

称、文本、大小、字体、背景色等公共属性信息,然后根据组件的不同类型设置组件的特有属性,保存到配置文件中并利用系统集成的软件将配置文件上传到数控系统服务器。

为了满足数控系统对精度、实时性等方面的高要求,PC端配置平台允许用户配置数控系统对应的多个界面。按照系统不同的功能需求以及各个界面不同的性质,将界面设计成不同的模式,并且可以在各个界面之间方便的切换<sup>[5]</sup>。在新建工程时配置好系统的分辨率、页面数量、过程数、每个过程的轴数、轴名及主轴数、以及系统上电首页等系统属性。而且用户可以随时对上述属性做反复的修改,使用户配置的信息便于复用,提高系统可重用性。

### 2.1.1 组件注册层

组件是人机界面中最基本也是最重要的单元,人机界面由一个个不同类型的组件组合而成<sup>[6]</sup>。用户利用二次开发平台界面中的组件盒,通过选择拖拽一系列的操作以及属性的配置,生成满足专用需求的人机界面。组件注册层充分利用了面向对象的设计思想,采用基于面向对象的二次开发技术,充分发挥类的继承性、封装性、多态性等特点,实现平台中组件注册层的框架设计。将平台中涉及到的组件、组件属性、属性编辑器等一切客观的实体都看成对象,复杂的对象可以由简单的对象经过组合构成<sup>[7]</sup>,例如组件的几何形状属性是一个属性对象,它又是由 X、Y、宽度、高度四个属性对象组成。每一个属性对象都有各自绑定的属性编辑器对象。对象与对象之间通过统一的信号-槽的消息传递机制实现相互联系。

为了支持平台的可扩展性和可移植性,将各个对象的定义与实现分离,通过继承展现各个对象之间的关系,再通过各个对象提供的重载和虚函数的特性以及实体的动态性<sup>[8]</sup>,建立了面向对象的二次开发平台接口框架,提供了对于平台进行扩展的面向应用程序的二次开发接口 API,包括组件扩展接口 API、组件属性扩展接口 API 和属性编辑器扩展接口 API 等。用户基于平台提供的 API 可以实现对组件、组件属性、属性编辑器等对象和行为的扩展,使平台具有良好的可扩展性。

在分析数控系统人机界面的显示特点之后,根据面向对象的设计方法,组件的类层次结构如图 2 所示。将平台的组件归纳为以下几类:

(1) 显示组件:用于显示数控系统运行过程中的各个参数值、表达式以及界面组成的一些静态元素,如图片等。

(2) 编辑组件:用于用户的输入。

(3) 事件组件:用于对数控系统的各种控制事件,如页面跳转、文件打开等事件操作。

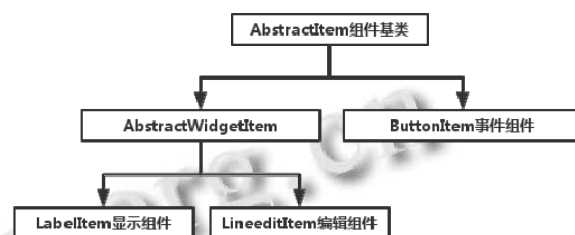


图 2 组件的类层次结构图

各种组件根据用途分别扩展上述三种组件类型,每一个组件都有 UUID 号用作程序的唯一标识。用户可以根据需要扩展多种类型的组件,本系统中设计了最基本的标签、输入框、按钮组件。

用户在扩展组件类型时,需要定义继承自组件基类的类,在类中实现组件定义、绑定属性等各种相关操作的接口,下面对组件扩展 API 主要的接口类型以及实现的功能进行介绍。

(1) createObject(): 用于创建图形视图框架中显示在 QGraphicsProxyWidget 上的控件,比如有 Label、LineEdit、Button 等,将定义的控件绑定到 QGraphicsProxyWidget 上。该接口无返回值。

(2) initProperty(): 用于将属性对象绑定到组件上。该接口也没有返回值。

(3) setDefault(): 将绑定到组件的每一个属性对象赋初始值。无返回值。

(4) setPropertyValue(const QString &name, const QVariant &value): 设置属性对象的值。参数 name 表示要设置的属性名称, value 表示属性值。无返回值。

(5) dirtyChanged(): 表示属性的变化,在页面上直接改变属性的值时,要通过该接口发送信号,用于属性编辑列表的更新。

通过分析数控系统的属性对象,归纳出各种类型对象的公有属性以及各个组件对应的特有属性,再将设计对象按照不同的特征结构进行分类设计。对于组件,设计了所有组件共同的基类 AbstractItem,它的属性可以分为两类:静态属性特征用于描述组件的数据;

动态属性特征用于描述对组件的操作. 这些属性都封装在顶层的结构中, 对于用户来说是不可见的, 体现面向对象的封装性. 平台注册的属性的层次图如图 3 所示.

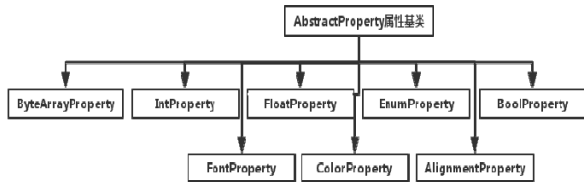


图 3 属性的类层次结构图

图 3 给出了平台注册的最基本的属性, 其他的属性都可以通过这些基本属性组合而成. 例如, 组件最基本的 GeometryProperty 属性, 用于描述组件的位置以及大小, 该属性就可以由四个 IntProperty 组成. 其他的一些属性构成的方法类似, 可以在属性中包含一个或者多个基本属性为子属性, 构成组件所需的复杂属性.

在将属性绑定到组件的过程中, 公有的属性放到结构的顶层组件中, 特有的属性放到结构的底层组件中. 下面对将属性注册到系统以及绑定到组件的注册绑定机制进行详细说明.

属性注册绑定机制的设计要考虑到系统的可维护、可扩展、灵活性等非功能需求. 考虑到用户后续可能对组件属性以及属性编辑器进行扩展, 充分发挥了面向对象程序设计的封装、继承、多态等特性降低系统的耦合度. 用户在对属性进行扩展时, 只需要编写要扩展的属性对象并实现预留的接口即可, 不会对已有的属性造成任何的影响. 本文采用最基本的简单工厂设计模式, 降低系统的耦合度, 完整封装属性对象. 属性注册机制的结构图如图 4 所示.

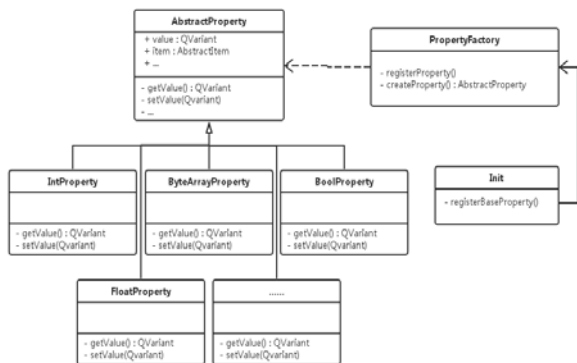


图 4 属性注册机制的结构图

具体的注册机制如下所述: 首先提供一个属性基类 AbstractProperty 类, 它包含了 value、attribute、property、children、parent、item 等属性, 还提供了用于文件生成层的 toXML()、fromXML()等方法, 然后其他的各种基本属性均继承自该基类, 按需重写基类的方法, 这样就大大降低了系统的耦合度, 修改或者添加新的属性类的时候不会影响到其他的属性类.

然后给出一个工厂类 PropertyFactory 用于得到指定的属性对象, 并将其绑定到组件上. PropertyFactory 类提供了两个主要的方法, 一个用于注册属性的 registerProperty()方法, 该方法需要提供的参数包括属性名称 name 以及要注册到的属性的元数据结构 metaObject, 然后将该属性注册到 metaObject 结构中, 用一个 Map 的二元集合存储该属性的名称以及元数据结构. 另一个用于得到指定的属性对象将其绑定到指定的组件上的 createProperty()方法, 这个方法需要提供的参数为属性的名称 name, 方法会根据名称从 Map 结构中查找对应的属性, 返回属性对象, 将其绑定到组件上, 并将属性的 item 属性设置为需要绑定的组件对象.

在属性注册时, 由于各种组件所需的属性类型较多, 本文采用了带参数的宏定义的方法来预处理注册机制, 这样在属性类型较多时, 不但可以很好的防止出错, 提高代码的可移植性, 而且可以更加顺利的整合代码规模提高运行速度. 本文的宏定义方法只在平台运行时执行一次, 将所有的属性对象一次性的注册到元对象 metaObject 结构中, 存储到 Map 集合, 避免多次调用宏产生的副作用所带来的不可预估的后果. 宏定义如下:

```

#ifndef registerProperty
#define registerProperty(name,type)
PropertyFactory::registerProperty(name,
                                &type::metaObject)
#endif
    
```

然后提供一个用于注册的 registerBaseProperty() 函数, 该函数在平台启动时执行, 将所有的注册函数包含在函数体中运行一次即可将所有的属性注册到平台的结构中. 例如要注册 Bool 类型的属性, 只需要将 registerProperty("Bool",BoolProperty) 包含在 registerBaseProperty() 函数体, 就会调用 PropertyFactory::registerProperty()方法注册属性对象.

将属性绑定到组件的绑定机制具体过程为：在组件基类 `AbstractItem` 类声明有一个用于存储描述组件的属性的 `Map` 集合，将属性的名称作为键值存储。然后声明一个 `InitProperty()`方法，将所有组件共有的属性，例如 `ObjectName`、`GeometryProperty` 等保存在 `Map` 集合中。在底层组件也就是各个子类中重写 `InitProperty()`方法，首先调用父类的该方法，将共有的属性先存储到子类的 `Map` 集合中，然后再创建子类自己的特有属性。然后就可以从 `Map` 集合中查找具体的属性进行配置。将属性编辑器注册到属性对象的注册机制类似，不再做详细描述。

### 2.1.2 界面定制层

界面定制层实现界面的定制功能，完成各种组件的添加、组件属性的设置；界面定制层包括页面视口、属性编辑视口、组件列表视口三个视口以不同的形式展现用户定制的人机界面，三个视口数据统一；属性编辑视口以树型结构显示组件的各个属性；组件列表视口以树型结构显示系统添加的页面以及每一个页面添加的组件。如图 5 所示为三个视口实时变化交互方式。

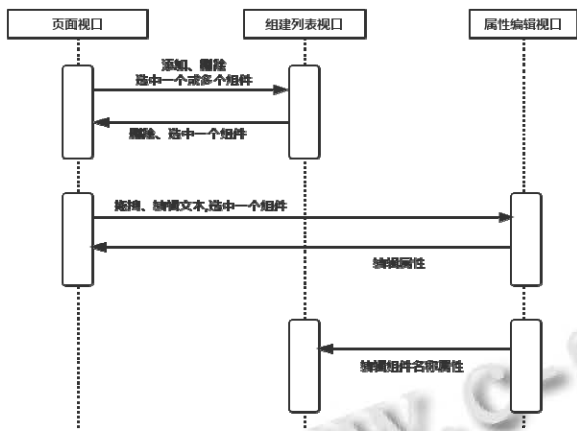


图 5 三个视口实时变化交互方式图

三个视口采用 Qt 特殊的信号-槽的消息传递机制实现数据的统一。Qt 摒弃了传统的消息驱动的机制，创新性的提出了一种在对象之间进行通讯的非常先进的 MOC 机制也就是信号-槽的消息传递机制<sup>[9]</sup>。这种传递机制削弱了各个对象之间的耦合度，信号发射的对象无需了解哪一个对象的槽函数对这个信号做出反应，同样对象的槽函数也无须知道是哪一个信号发射触发自己<sup>[7]</sup>，只需要用 `connect()`函数将信号槽关联，那么信号在发射时槽函数一定会被调用，这是 Qt 的机制

保证的。上述的三个视口均声明了多个信号、槽函数，保证视口间数据的统一。每个视口一旦有操作发生，该操作对应的信号就会发射，其他的视口的槽函数做出响应更新显示。

页面视口添加的组件包括显示组件、编辑组件、事件组件三种。添加组件后用户可以通过拖拽、编辑、水平等距、垂直等距等快捷操作配置组件通用属性。除了通用属性，还要设置各种组件特有的属性。如图 6 为各种组件的特有属性设置流程图。

页面定制层的定制过程全部采用参数化的设计方法。这种方法保持了各个组件结构之间的拓扑关系不变<sup>[10]</sup>，组件的所有属性一律采用参数来描述和控制，某一个参数值的变化将会直接导致三个视口中显示内容的自动变化，提高了组件的修改重构能力，极大的提升平台的柔性设计<sup>[10]</sup>。参数化设计特别适合用于界面的组件结构形式固定而结构尺寸经常变化的规格化、标准化、系列化的设计，利用参数化的设计来描述组件的形状特征<sup>[11]</sup>，可以极大的提高界面的配置速度。

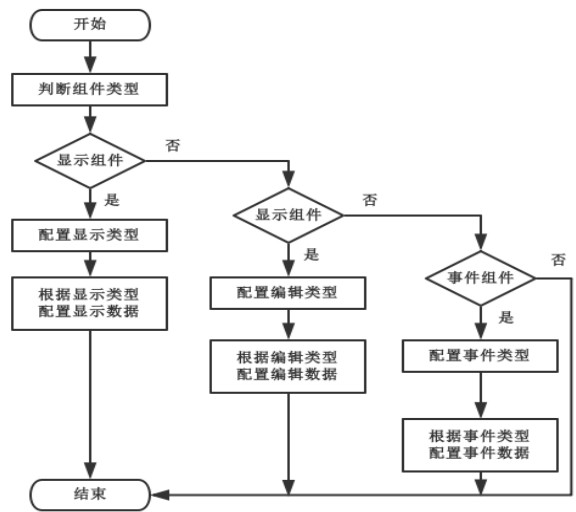


图 6 组件的特有属性设置流程图

### 2.1.3 文件生成层

文件生成层读取用户配制的人机界面数据，将数据封装到 XML 节点对象中，自动生成配置文件。

### 2.2 NC 端解析平台模块

PC 端将生成的配置文件通过以太网传送到数控系统中由文件解析层解析并通过基于数控系统的统一消息库和统一消息接口的模块交互机制完成与数控系统的融合通信。

文件解析层读取并解析 XML 配置文件数据, 采用基于预处理的解析技术<sup>[12]</sup>. 如图 7 所示.

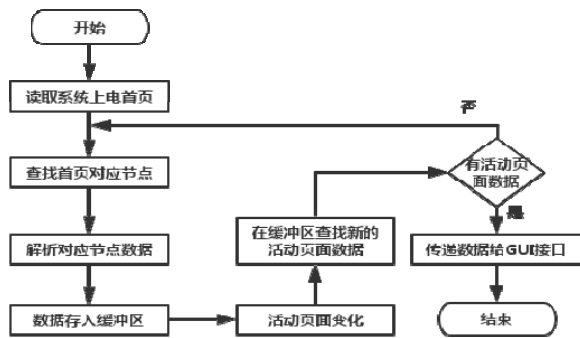


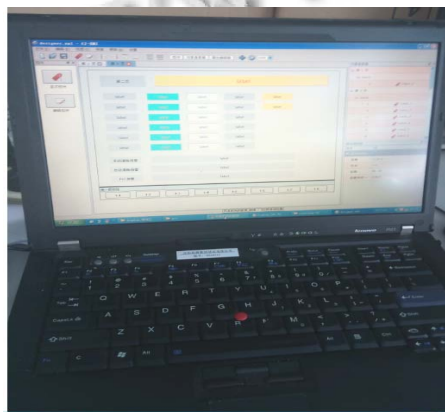
图 7 文件解析流程图

调度管理层实现模块初始化、功能模块调用、系统配置与模块间交互等功能, 将中性语言描述的数控专用控制功能、图形用户界面定制等功能转化成功能模块调用. 通信服务层实现 GUI 二次开发接口与数控系统的 HMI 功能模块的交互. 系统接口层调用基于试试控制系统库的专用数控系统二次开发平台的 GUI 二次开发接口, 将 PC 端配置的数据传递给接口, 实现与原有数控系统的融合通信.

### 3 实验测试

#### 3.1 功能测试

利用 Qt Creator 平台开发本文的数控系统人机界面二次开发平台的 PC 端配置平台, 蓝天数控高精数控系统作为 NC 端解析平台, 可以实现对数控系统的人机界面进行专项定制以及重复修改操作. 实验的 PC 端定制界面与数控系统的显示界面相同, 并实现了相关参数的配置, 可以监控机床状态, 编辑相应变量, 对机床外部单元进行控制, 如图 8 所示.



(a)PC 端配置平台实验测试图



(b)NC 端解析平台实验测试图

图 8 实验测试图

#### 3.2 界面的重复定制方法

开发人员可以利用平台进行界面的修改、重复定制. 数控系统人机界面二次开发作为数控系统的可选功能, 提供一个参数控制该功能是否打开. 如果打开的话, 要想重新定制界面或者进行修改操作, 只需要将原有界面的配置文件下载后在 PC 端打开, 然后重新定制需要修改的部分, 生成新的配置文件上传到数控系统服务器, 替换原有的文件即可. 配置文件名称固定为 designer.xml, 数控系统会在指定的界面目录下查找该文件, 上电后自动加载显示. 因此在修改时上传到数控系统的配置文件名称不可变, 必须是上述名称. 修改界面的操作流程图如图 9 所示.

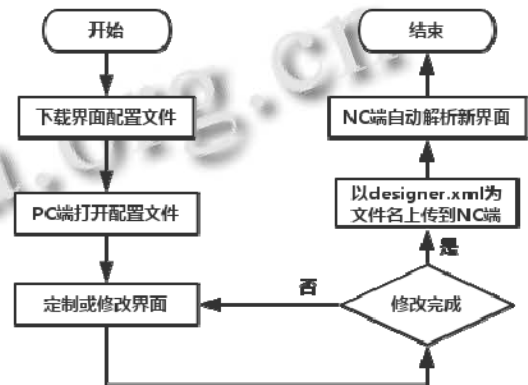


图 9 界面修改或重新定制流程图

### 4 结语

本文就数控系统人机界面不能满足用户个性化需求、开发维护周期长等不足之处, 设计了用于数控系统人机界面的二次开发平台, 并对该平台各个主要模块设计方法进行了详细的描述.

利用本文提出的数控系统人机界面二次开发平台对数控系统人机界面进行可视化定制, 具有以下优点:

①支持可视化配置. 用户在本平台配置界面的过程中, 所见即所得, 对界面有直观的把握, 用户配置方便, 无需学习编程知识. ②支持组件及组件属性的扩展. 用户可以对平台的组件、属性等进行特定的增加, 只需要实现预留的接口. ③提高人机界面开发与维护效率, 降低成本. 用户和开发人员可以利用平台随时对人机界面进行修改、定制, 缩短开发周期, 降低维护成本.

### 参考文献

- 1 杨丽娜, 马跃, 吴文江, 田军锋, 张建磊. 可定制的数控系统人机界面设计. 组合机床与自动化加工技术, 2011, 1: 57-60.
- 2 邴旭, 林剑峰, 徐兆成, 化春雷. 数控系统人机界面开发方法研究. 制造技术与机床, 2012, 11: 144-147.
- 3 王开丽. 基于自动生成 XML 的动态图形界面设计与实现 [硕士学位论文]. 成都: 电子科技大学, 2011.
- 4 黄洪, 林辉, 王奔. 一种图形用户界面的 XML 描述方法与工具开发. 计算机应用与软件, 2011, 10: 198-202.
- 5 金信琴, 裴照辉. 浅析数控系统的软件人机界面研究. 工业设计, 2013, 1: 61-63.
- 6 张琼琼, 马跃, 王志成. 组态式人机界面编辑与仿真技术. 计算机系统应用, 2015, 24(4): 26-31.
- 7 王哲. 基于 CAD 二次开发技术的晋城市供水管网管理系统的研究 [硕士学位论文]. 太原: 太原理工大学, 2012.
- 8 李咏红. CAD 二次开发方法研究与实现 [硕士学位论文]. 成都: 电子科技大学, 2004.
- 9 郑乃行. 基于 Qt\_OSG 的特征树结构的可复用式二次开发框架的研究和应用 [硕士学位论文]. 杭州: 杭州电子科技大学, 2014.
- 10 方贵盛, 王建军. AutoCAD 二次开发技术及其应用研究. 机床与液压, 2007, 6: 185-187, 212.
- 11 李咏红, 杜平安. 面向对象的参数化 CAD 二次开发方法研究. 电子科技大学学报, 2004, 5: 597-599, 610.
- 12 冯进, 丁博, 史殿习, 张曷熹, 许凯. XML 解析技术研究. 计算机工程与科学, 2009, 2: 120-124.
- 13 张中, 马国艳. FANUC 数控系统人机界面的二次开发. 金属加工(冷加工), 2014, 6: 62-64.
- 14 张曙. 数控系统及其人机界面的新进展. 机械设计与制造工程, 2014, 10: 1-5.
- 15 朱晓娟. 基于关系的 XML 数据存储技术研究. 计算机与现代化, 2009, 12: 51-54.