

# 基于 GTK+ 的编辑界面自动生成设计与实现

王庆元, 周 宾

(中国电子科技集团公司第二十八研究所, 江苏 南京 210007)

**摘要:**通信系统中用户信息交互频繁,格式种类繁多,为解决软件编码定制用户信息编辑界面大部分工作重复繁琐的现状,应用 XML 方法对图形界面元素进行描述,结合开源 GTK+图形用户界面库的使用特点,给出了一种用户编辑界面自动生成的实现方法。通过解析 XML 文档并调用界面控件库接口动态生成图形用户界面,实现界面设计和生成实现的分离。实践证明,运用该方法实现多样化用户信息的编辑,能够大幅降低界面设计编程的工作量,提高了软件的开发效率和可维护性。

**关键词:**可扩展标记语言;界面描述;界面自动生成;GTK+

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2014)05-0180-04

doi:10.3969/j.issn.1673-629X.2014.05.044

## Design and Realization of Edit-interface Automatic Generation Based on GTK+

WANG Qing-yuan, ZHOU Bin

(The 28th Institute of China Electronics Technology Group Corporation, Nanjing 210007, China)

**Abstract:** There are numerous and frequent interaction user information in communication system. To solve the problem of repetitive and tedious work in customizing user information editing interface by software coding, a method for graphical user interface automatic generation is presented by applying the XML method in the description of graphical user interface, combining the characteristics of open source widget library GTK+. It dynamically creates graphical user interface by parsing the XML document and calling the API of widget library GTK+, this enables the separation of interface design from the generation. Practice proves by using this method in multiple information edit module, it can reduce the workload of interface design and coding greatly, and improve the developing efficiency and maintainability of software.

**Key words:** XML; interface description; interface automatic generation; GTK+

## 0 引言

自动化信息系统中的大部功能都与用户显控界面有关,据统计,用户界面在软件开发中占据 50% 左右的编码时间与运行时间<sup>[1]</sup>。传统的用户界面设计和开发一般是利用现有操作系统运行平台上的配套集成开发工具,使用其绑定的界面开发工具,通过手工选择控件进行布局与相应的操作处理编程实现的<sup>[2]</sup>,界面开发工作通常比较繁杂和重复,对界面的一个小小的改动可能会牵涉到整个界面其他相应元素的调整,有时还要编译整个工程。

概念模型和陈述模型分别从不同角度提出应用和表示分离的思想,但只能针对个性化的界面,不能将界

面的设计提供到统一的高度,在实际应用中缺乏灵活性<sup>[3]</sup>,用户界面与功能代码之间大多是紧耦合的,类似界面功能开发的可重用性较差。文中给出一种基于 GTK+图形用户界面库的用户编辑界面自动生成设计与实现方法,具有较好的可扩展性与可重用性。

## 1 GTK+简介

### 1.1 GTK+概述

GTK+是运行在 Linux 操作系统上的开源轻量级高级图形库,旨在为应用程序提供一套与平台无关的图形用户界面接口<sup>[4]</sup>,其使用 C 语言编写实现对象系统,具有面向对象特性,可实现继承、类型检验等技术,

收稿日期:2013-07-23

修回日期:2013-10-28

网络出版时间:2014-02-11

基金项目:江苏省自然科学基金资助项目(10KJB510014, BK2012832)

作者简介:王庆元(1981-),男,工程师,研究方向为嵌入式系统、指挥信息系统。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20140211.1617.041.html>

提供基于事件驱动和信号/回调函数的消息传递机制,在图像文件方面支持如 JPEG、BMP、GIF 等多种流行格式的图像显示;在文字显示方面支持如 ISO8859-1 ~ ISO8859-15、GB2312、GBK 等字符集以及 TrueType 等多种类型的矢量字体;在图形控件方面提供如窗口、按钮、标签、单行编辑框、多行文本编辑框、树、表格等较丰富的使用选择<sup>[5]</sup>,而且允许开发者通过继承方式创建自定义控件,可供程序员创建出适应各种应用场景的图形界面;总体而言 GTK+是一个可移植性好,小巧稳定且适应多平台应用的图形用户界面支持系统,在项目开发过程中可以保持桌面系统与嵌入式系统中相关界面功能处理模块的统一设计与开发。由于目前在嵌入式系统上并无相对统一且广泛使用的图形界面库,因此可以根据 GTK+的设计特点与实际需要对其进行裁剪与优化,将其移植到嵌入式系统进行界面显控功能开发,适应项目功能多平台运行的要求。

由于 GTK+经历了长期的开发及应用实践,是较成熟的开源项目,其一直得到一支强大开发团队的维护而处于不断的版本升级中,且由于它是基于 LGPL 授权的,在未来开发开源软件或商业的非自由软件领域中都将得到越来越广泛的使用。

1.2 GTK+库应用开发流程

GTK+系统使用事件驱动和信号回调机制作为应用程序的创建构架<sup>[6]</sup>。其中 gtk\_main()函数是 GTK+的主处理循环,整个系统应用只能一次调用,当用户对界面控件执行某种操作即激发产生一个事件,GTK+主循环捕获到事件后通知该控件并产生特定信号,此时控制权就转让给了由事先与该控件及信号建立关联的“回调函数”进行处理,整个 GTK+界面操控程序就一直处于“收到事件→发出信号→处理信号”的循环中,直到完全结束才退出程序。

下面对 GTK+的程序处理框架进行简要说明:

1)GTK+程序入口与初始化。

GTK+程序入口点采用了标准的 C 程序入口函数 main(),其原型为:int main(int argc,char \* argv[]),参数 argc 和 argv 分别表示命令行参数个数和参数字符串数组的指针。在 main 函数中可通过调用 gtk\_init 函数进行 GTK+的初始化,初始化函数原型为:void gtk\_init(int \* argc,char \* \* \* argv);该函数在 GTK+程序中是必需的,它初始化 GTK+程序并传递响应的命令行参数,根据命令行参数可设置系统相应的运行环境参数。

2)创建显示主窗体并进入主循环。

GTK+库提供了丰富的控件供编程者使用,应用可开发系统功能布局界面作为主控框架界面进行相应的系统功能调用,如可使用窗口配合功能按钮或菜单完

成主控界面,根据需要调用相应控件类的接口函数来创建控件、设置其属性、创建好信号连接器并调用显示函数 gtk\_widget\_show\_all(GtkWidget \* widget)显示全部主窗体,之后调用主处理循环函数 gtk\_main 即可进入主事件循环;当需要全部退出界面操作处理程序时,可在相应的关联回调函数中调用 gtk\_main\_quit()函数退出主循环结束 GTK+程序的运行。

3)信号处理函数的设置。

信号处理函数又称回调函数,其函数原型为:void func(GtkWidget \* widget,gpointer callback\_data);其中,widget 是指向发出信号的控件指针,callback\_data 是指向要传递给回调函数的用户附加数据。特定控件发出了特定信号后,应用程序需要确定调用哪个回调函数来处理信号,GTK+系统采用信号连接器来设置信号与回调函数之间的关联关系,信号连接器函数原型为:gint gtk\_signal\_connect(GtkObject \* object,gchar \* name,GtkSignalFunc func,gpointer func\_data);其中,object 是发出的信号的控件指针;name 是发出的信号名称,为字符串形式;func 是发出信号后要调用的函数名,需要用 G\_CALLBACK 宏来转换;func\_data 是要传递给回调函数的用户附加数据,这里只能传递一个参数,如果有多个参数可以先将它们定义为一个结构体,再将此结构体作为参数传递过去<sup>[7]</sup>。其他更广泛的应用可参阅文献[8]。

2 编辑界面自动生成技术的设计与实现

2.1 界面自动生成结构设计

由于 XML 具有数据重用、数据和表示分离、可扩展性、语义信息自描述性、基于树结构、易于编程,以及可以用来描述极复杂的数据等优势<sup>[9]</sup>,系统决定采用其特性来描述界面元素进行模块的设计与实现。编辑界面自动生成处理结构主要由界面描述定制工具、界面描述文件包(XML)、XML 解析器、界面自动生成引擎、界面元素控件库、界面应用处理、自动生成编辑界面七部分组成,其结构关系如图 1 所示。

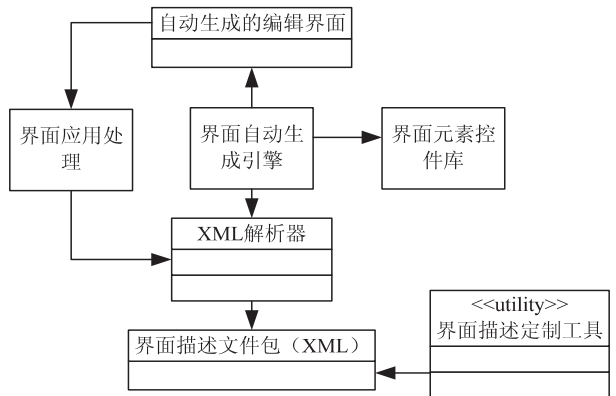


图 1 界面自动生成处理结构

其中界面描述文件包由界面描述定制工具生成,界面自动生成引擎通过 XML 解析器解析每一界面的描述内容文件,并调用系统图形界面库中的界面元素控件库,动态生成用户信息编辑界面;随后可由界面应用处理模块完成自动生成界面的编辑数据提取组装、接收格式数据的解析组合等实际应用处理。

## 2.2 界面模板信息的 XML 格式描述

界面模板是对用户界面的宏观构成、布局等总体属性的描述<sup>[11]</sup>,用来描述用户界面信息的 XML 格式文件可称为界面模板文件,界面上的每个控件都可以被描述成一个 XML 节点,而控件的那些相关属性都可以描述成这个 XML 节点的属性<sup>[12]</sup>。用户编辑界面的模板描述内容可包括界面的编号与名称、界面包含元素的类型和名称、相应界面元素初始值或来源等,下面的 XML 格式界面模板文件内容示例描述了界面表述元素的结构:

```
<? xml version="1.0" encoding="gb2312" ? >
<N 编号="1" 名称="编辑界面样例">
  <N 类型="标签" 名称="请配合" />
  <N 类型="编辑框" 名称="联系人" 数据类型="通信录"
序号="1" />
  <N 类型="标签" 名称="于" />
  <N 类型="编辑框" 名称="时间" 数据类型="时间" 序号
="2" />
  <N 类型="标签" 名称="到达" />
  <N 类型="编辑框" 名称="地点" 数据类型="字符" 序号
="3" />
  <N 类型="标签" 名称="进行" />
  <N 类型="组合选择框" 名称="作业类型" 序号="4" 初
始选项="0" >
    <N 类型="选择项" 序号="5">
      <N 类型="标签" 名称="作业 1" />
    </N>
    <N 类型="选择项" 序号="6">
      <N 类型="标签" 名称="作业 2" />
    </N>
  </N>
</N>
```

其中标签为文字显示内容标签,编辑框为用户填写内容项,根据数据填写类型需要可分为数值型、字符型、时间类型或通信地址等,组合选择框为可选择的下拉列表项,其他类型可根据实际需要进行相应扩展,如装备物资表等,根据数据量的大小可选择配置文件或数据库进行基础数据的存储使用。

## 2.3 界面描述定制工具

界面描述定制工具用来生成用于界面描述的 XML 文件,确定界面描述的 XML 格式后,可选择开发平台设计实现该工具,根据用户界面显示操作元素要

求,使用该工具进行界面元素描述设置,辅助生成指定界面的 XML 描述文件;由于嵌入式系统的资源限制与使用环境的局限,可在桌面 Windows 平台上应用 Delphi 等可视化开发工具进行设计实现;通过 Delphi 工具丰富的控件库(如树形控件、XML 文档控件等),对树形节点按照界面元素层次进行添加,并设置节点的属性值(如类型、名称等),配置完成后调用 XML 文档控件的操作过程生成界面描述文件。

由于用户信息界面种类繁多,可对其进行层次分类并建立文件编号索引,同时生成用户信息分类描述文件与对应的界面信息内容描述文件,完成后一起导入嵌入式平台进行使用,信息分类描述文件示意如下:

```
<? xml version="1.0" encoding="gb2312" ? >
<N 名称="用户信息分类">
  <N 名称="指示">
    <N 名称="作业协助" 编号="1">
      <N 名称="地点巡查" 编号="2">
        .....
      </N>
    <N 名称="请示">
      .....
    </N>
  </N>
```

其中编号可指定具体用户界面信息的描述文件,如 1.XML 文件可存储上述作业协助界面的信息描述内容;定制工具在桌面平台开发相对快捷,在此不作赘述。XML 解析器也有公开通用的类实现代码,可在多种平台方便移植,下面主要描述自动生成界面的设计处理流程及实现、应用方式。

## 2.4 界面自动生成引擎处理流程

界面自动生成引擎工作时可通过调用 XML 解析器提供的接口,解析界面模板描述内容对应的 XML 文件,形成界面信息元素的节点层次描述树;采用递归算法遍历层次描述树中的所有界面元素节点,根据节点描述信息调用界面元素组件创建相应的界面元素,并根据这些描述信息与规范约定对其进行合理布局、设置各界面组件的初始数据、关联其操作响应处理过程,最终生成对应的信息显示编辑界面,相应的处理流程如图 2 所示。

如整形数值输入框最多占用 10 个数字的界面宽度,关联的自动处理可对非法输入字符进行屏蔽;联系人选择框可根据最大名称长度设定界面占用宽度,初始值可设定为最近系统选择的联系人等。

## 2.5 应用 GTK+创建编辑界面方法

通过对 GTK+提供的界面控件库进行演示对比,考虑到其中的多行文本编辑框具有丰富的文本内容显示功能,能够包含显示多种类型控件且可以对内容自



动顺序显示,界面的生成设计采用在对话框中创建可以滚动显示的多行文本编辑框,通过在多行文本编辑框中动态插入控件的方式形成多控件显示编辑界面,编码实现的方式步骤如下:

a) 调用 `gtk_dialog_new` 函数创建对话框窗口,接着调用 `gtk_scrolled_window_new` 函数在对话框窗口上创建滚动窗口,在滚动窗口包含调用 `gtk_text_view_new` 函数创建的多行文本编辑框,调用 `gtk_text_view_get_buffer` 函数获取多行文本编辑框内容区指针,通过对内容区指针的操作准备在多行文本编辑框中动态插入相应的界面控件。

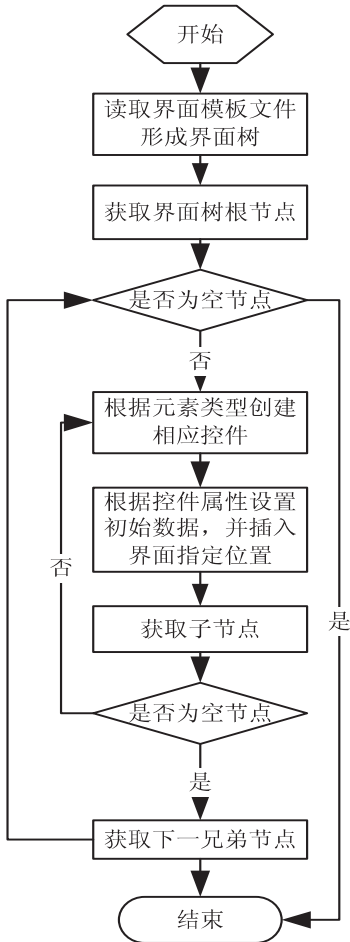


图2 界面生成引擎处理流程

b) 在调用 XML 解析器对 XML 格式的界面模板文件进行解析后,获得其节点描述树的根节点指针,通过 XML 节点类提供相应的成员函数可遍历访问指定节点对象的父节点、子节点与兄弟节点。

c) 针对每种控件类型设计相应的插入多行文本编辑框处理过程函数;如标签文本可直接调用 `gtk_text_buffer_insert` 函数插入相应的文本内容,对于其他如编辑框、组合选择框、按钮等用户操作控件可先创建该类型控件,再根据其对应节点的属性描述设置其初始数据、显示样式并关联相应的消息处理回调函数,然后调用 `gtk_text_buffer_create_child_anchor` 函数获取当前

多行文本编辑框的指定的可插入点,再调用 `gtk_text_view_add_child_at_anchor` 函数将当前控件放到该指定位置。

d) 设置递归函数 `CreateSiblingWidget (XMLNode * pNode, GtkTextBuffer * buffer, GtkTextIter &iter)` 对节点描述树进行递归遍历,针对节点类型调用上述相应的控件插入处理函数,将控件顺序布局到多行文本编辑框中,并可创建相应的控件链表记录创建的控件对象指针,以方便检查获取用户界面的最终填写值。

完成上述步骤后调用 `gtk_widget_show_all` 函数显示全部界面元素,调用 `gtk_dialog_run` 函数即可进入窗口模态对话框操作状态。

2.6 界面数据的应用处理

2.6.1 界面编辑数据的提取组装

在用户完成界面元素的编辑填写后,可根据创建界面时记录的控件链表进行填写数据完整性与合法性检查,不合规范的界面编辑元素提示用户进行重新输入;全部检查无误后可按顺序提取界面输入值,并按转换规则进行转换与组装,如联系人可转换为联系人编号,时间可转换为秒值(从 1970 年到填写时间的整秒数值),作业类型可填写相应的作业编号等;如此既可减少实际信息的通信量,又可增强信息传输的保密性。

2.6.2 格式数据的解析组合

用户信息接收端在收到编码信息后需要解析处理,可根据信息编号解析该信息对应的 XML 格式描述文件,按照描述文件约定的数据格式顺序拆分信息编码,根据编辑数据类型与转换规则逐一进行接收数据的逆变换,最终完成信息元素的解析与用户编辑内容的完整拼装,达到系统内用户多样化信息的一致互通。

3 结束语

针对实际项目应用中用户信息编辑界面内容需求多样繁杂的情况,文中结合应用 XML 特性对图形用户界面的表达元素进行规范化描述,通过具体示例介绍了应用 GTK+设计实现编辑界面自动生成的关键处理流程,实现信息编辑处理模块展现层和业务逻辑层的分离。开发者可根据用户实际需求提炼信息表述的要素点,快速定制相应的用户信息编辑处理界面,有效降低用户信息输入的工作量,快速编辑形成格式化描述信息,实际通信时只需传输动态编辑数据,降低信息传输的信道占用。通过基于某自主化嵌入式操作系统的车辆通信项目中的实际应用,有效解决了用户界面繁多而导致的程序代码庞大与界面重复开发的问题,降低了应用开发工作量及后期变更与维护成本。实际使用可根据需要进一步丰富界面的表达元素,满足不同

cmd 命令 arp-s 将 IP 地址 192.168.0.1 与 MAC 地址 00-1c-23-17-4a-cb 建立映射表。

gm\_rx\_d 为输入的 MAC 帧数据,当数据使能信号 gm\_rx\_dv 有效时,读入前导码 7 个 55h 和帧开始符 D5h,然后判断目标 MAC 地址 00-1c-23-17-4a-cb,如果正确,去除包头数据后,双口 RAM 的写使能信号 wren1,wren2 使能,并给出双口 RAM 的写地址 wrad-dr1,wrad-dr2,两个双口 RAM 是交替写入的。

h\_sync 和 v\_sync 是产生的 VGA 行同步和场同步信号,在数据有效时段产生读双口 RAM 的使能信号 rden1,rden2,并给出读双口 RAM 的地址 rdaddr1,rdad-dr2,将两个 8 bit 的数据 ram\_q1 和 ram\_q2 拼接成 16 bit 的数据 rgb\_data。VGA\_clk 的频率为 65 MHz,rx\_clk 的频率为 125 MHz。

运行系统后,将彩条数据通过千兆以太网传输给 FPGA,在显示器上可以显示相应的彩条。

6 结束语

该系统基于 FPGA 设计,完成了千兆以太网数据包的接收、解析,数据的缓存以及 VGA 的显示。相比于其他的基于 DSP 或者 ARM 的数据传输及 VGA 的显示系统,该系统具有可靠性高、性能稳定、开发灵活、调试简单、设计成本低、开发周期短、屏幕显示质量高的特点。

参考文献:

[1] Sridharan K,Priya T K. The design of a hardware accelerator for real-time complete visibility graph construction and effi-

(上接第 183 页)  
用户的需求。

参考文献:

[1] Lozamo M D,Gonzalez P,Romas I. User interface specification and modeling in an object oriented environment for automatic software development[C]//Proc of the 34th international conference on technology of object-oriented languages and systems. Santa Barbara,CA:[s. n. ],2000:373-381.  
[2] 李琦,李建成,张科峰. 基于 GUI4J 的界面自动生成技术[J]. 西安工程大学学报,2010,24(3):334-337.  
[3] da Silva P P,Paton N W. A UML-based design environment for interactive applications[C]//Proceedings of the 2nd international workshop on user interfaces to data intensive systems. Zurich,Switzerland:IEEE Computer Society,2001:60-71.  
[4] 宋国伟. GTK+2.0 编程范例[M]. 北京:清华大学出版社,2002.

cient FPGA implementation[J]. IEEE Trans on Industrial Electronics,2005,52(4):1185-1187.  
[2] Seifert R. 千兆以太网技术与应用[M]. 郎波,译. 北京:机械工业出版社,2000.  
[3] 夏宇闻. Verilog 数字系统设计[M]. 北京:北京航空航天大学出版社,2008.  
[4] 王建军,李勇帆,肖杰,等. 以太网协议数据封装动态仿真的设计与实现[J]. 计算机技术与发展,2010,20(6):133-136.  
[5] 徐洪波,俞承芳. 基于 FPGA 的以太网 MAC 子层协议设计实现[J]. 复旦学报(自然科学版),2004,43(1):50-53.  
[6] LAN/MAN Standards Committee of the IEEE Computer Society. IEEE Std 802.3-2005 telecom and information exchange between local and metropolitan area networks specific requirements[S]. 2005.  
[7] 王时雨,贺占庄,张盛兵. 基于 FPGA 的以太网 MAC 控制器的实现[J]. 微电子学与计算机,2012,29(2):171-175.  
[8] Stevens W R. TCP/I 详解[M]. 范建华,胥光辉,张涛,等,译. 北京:机械工业出版社,2000.  
[9] Marvell Semiconductor. 88E1111 integrated 10/100/1000 ultra gigabit Ethernet transceiver[M]. USA:Marvell Semiconductor,2004.  
[10] VESA and Industry Standards and Guidelines for Computer Display Monitor Timing[S]. 2004.  
[11] 张亚平,贺占庄. 基于 FPGA 的 VGA 显示模块设计[J]. 计算机技术与发展,2007,17(6):242-245.  
[12] 朱文伟,许忠仁. 基于 FPGA 的 VGA 图像控制器的设计与实现[J]. 贵州大学学报(自然科学版),2009,26(2):109-111.

[5] GTK+reference manual[EB/OL]. 2013-07. <http://doc.gnu-darwin.org/gtk20/>.  
[6] 黄瑛,刘少君,裴海龙,等. 基于 GTK+库数控机床界面的设计与实现[J]. 制造业自动化,2005,27(1):52-54.  
[7] 秦超,林为民,王力科. Linux 下图形配置界面的设计与实现[J]. 微计算机信息,2004,20(1):120-121.  
[8] GTK+ 2.0 Tutorial[EB/OL]. 2013-06. <http://oss.org.cn/man/develop/GTK+/tutorial/>.  
[9] 吴姗姗,彭向阳. 基于 XML 的通用数据库访问引擎的设计与实现[J]. 指挥信息系统与技术,2010,1(1):55-59.  
[10] 张银鹤,张秋香. XML 实践教程[M]. 北京:清华大学出版社,2007.  
[11] 吴桂阳,万建成,于清国,等. 基于 XML 的界面模板技术[J]. 计算机应用研究,2004,21(10):54-55.  
[12] 陈超,金晶. 态势显示软件人机界面建构方法[J]. 指挥信息系统与技术,2011,2(2):66-70.

# 基于GTK+的编辑界面自动生成设计与实现

作者：王庆元， 周宾， WANG Qing-yuan， ZHOU Bin

作者单位：中国电子科技集团公司第二十八研究所, 江苏 南京, 210007

刊名：计算机技术与发展

英文刊名：Computer Technology and Development

年，卷(期)：2014(5)

本文链接：[http://d.wanfangdata.com.cn/Periodical\\_wjfz201405044.aspx](http://d.wanfangdata.com.cn/Periodical_wjfz201405044.aspx)