

基于XML的嵌入式Linux系统用户界面定制

靳 京, 桑 楠, 刘 一

(电子科技大学计算机科学与工程学院 成都 610054)

【摘要】针对嵌入式Linux系统GUI的可定制性需求,在分析了现有多款嵌入式GUI产品的基础上,提出了一种基于XML的用户界面动态定制方案。它通过一个定制模块为最终用户提供界面维护修改功能,并按照保存在XML文档中的定制结果来显示更新后的界面,实现界面设计数据与应用逻辑模块的分离。该方案在建立于MiniGUI上的嵌入式Linux系统用户界面中得以实现。

关 键 词 定制; 嵌入式Linux系统; 可扩展标置语言; 图形用户界面
中图分类号 TP311 **文献标识码** A

XML-Based User Interface Customization and Dynamical Modification of the Embedded Linux System

JIN Jing, SANG Nan, LIU Yi

(School of Computer Science and Engineering, University of Electronic Science and Technology of China Chengdu 610054)

Abstract Since the demand of the embedded Linux GUI system can be customized, an approach to customize and modify UI dynamically with XML is presented based on analyzing most of the embedded GUI productions. This approach provides a function to modify UI for the final user with a customizing modular, by which the data of UI design is separated from application logical modular. The information of the modification stored in the XML document will be used to display the new UI. This approach has been achieved in the embedded Linux system based on MiniGUI.

Key words customization; embedded Linux system; extensible markup language; graphical user interface

随着Linux操作系统在我国的不不断普及和迅猛发展,在Linux下开发图形用户界面(GUI)的应用程序已日渐成为一种需求^[1]。图形用户界面允许用户通过控制各类图标来完成各种计算机操作,综合利用计算机强大的处理能力、新型输入/输出设备以及先进的软件技术,使人机交互手段丰富而高效,为用户提供了一个直观的、友好的操作环境^[2]。

GUI在软件系统中占据着非常重要的位置,但却是软件系统中最容易发生变化的一部分^[3]。对实时性要求很高的嵌入式Linux系统而言,由于其本身硬件的条件限制,对GUI的基本要求除了要求占用资源少、高性能、高可靠性以外,可配置性也已经被提到相当重要的地位。

目前已经开发出的嵌入式GUI产品很多,如:Century Software 的 Microwindows (Nano-X)、Trolltech的QT/Embedded、XFree86 Project的XFree86以及飞漫公司的MiniGUI等。它们的性能特点各有

优势,有的在可配置性性能方面不理想,有的界面管理太复杂,不适用于多数嵌入式系统。在电子产品日趋个性化的今天,这一现状显然无法适应市场的需要。

另一方面,大多数系统把生成界面代码作为其主要目的,开发人员对界面的定义生成出界面形态的代码,然后在编程过程中进行组织与管理。这样,产生的界面信息与源代码处于相同级别,对界面信息的修改必须通过编译、连接之后,才能反映到可执行文件中。这种传统的界面管理方式只是简化了界面的生成与修改过程,对于用户界面的可扩展性并没有带来实质性的好处^[4]。

因此,本文提出了一种基于XML的可定制界面管理机制,实现对建立在MiniGUI上的嵌入式Linux系统用户界面进行配制及其动态更新,为最终用户提供维护修改功能而无需重新编译和连接,很好地实现了用户界面与应用语义的分离。

收稿日期: 2005-10-19

基金项目: 云南省省校省院合作科研项目(W060321)

作者简介: 靳 京(1975-),男,博士生,主要从事嵌入式实时系统方面的研究。

1 相关技术背景

1.1 XML及其GUI应用

XML是一种元数据语言,它允许开发者以标准统一的方式简便地描述,在任意应用程序中传输结构化数据,并且可以跨平台地进行数据内容的定义声明,由此可提供更有意义的搜索结果。XML还是一种通用的标准,适用于在异构系统、不同平台及应用程序间交换数据^[5]。XML程序无需编译即可执行,便于修改,为定制和更新用户界面提供了可能。

基于XML的用户界面描述将界面元素(例如对话框、菜单、工具条等)定义为XML标签(Tag),可方便地将界面元素以及相关信息写入XML文件中,此XML文件即定义了相应的用户界面。显然,基于XML的格式文档易于编辑,当然还可由此建立可视化界面设计工具。更为重要的是,此类文档可在程序运行时动态载入或修改,从而赋予了应用程序更为强大的界面处理能力^[6]。

1.2 MiniGUI

MiniGUI是一个专门为基于Linux的嵌入式系统提供的一个轻量级图形用户界面支持系统,是基于SVGALib及LinuxThread库的多窗口GUI支持系统。它采用了类Win32的API接口,实现了简化的类Windows98风格的图形用户界面。MiniGUI是遵循LGPL条款的纯自由软件,同时也是国内最早出现的几个自由软件项目之一,可以运行在任何一种具有POSIX线程支持的POSIX兼容系统上。MiniGUI有以

下主要特点:(1) 提供了完备的多窗口机制,实现了类Win32的消息传递机制;(2) 支持多字符集和多字体,以及全拼、五笔等汉字输入法;(3) 支持常见的图像文件,如BMP、GIF、JPEG、PCX、TGA等,以及Windows的资源文件,如位图、图标、光标等;(4) 小巧、可配置是MiniGUI最突出的特点;(5) 移植性好。

MiniGUI的目标是保持小的特点,并积极向模块化方向发展,为各种嵌入式系统或实时系统提供一个小型的窗口支持系统^[7]。

2 可定制GUI结构及其基本原理

用户界面定制(UI Customization)是指在软件运行过程中,用户可以根据自己的使用习惯,对界面元素(如菜单、工具栏、键盘命令等)的属性(如文字、图标、大小、位置)进行修改,当软件结束运行时,界面定制的结果将被保存到固定的界面配置文件中。

图1为XML的可定制界面管理机制,图中系统利用界面定制模块,对界面元素在允许的范围内进行个性化修改,并将修改后的界面数据及时写入到XML界面文件中的对应位置。当系统运行时,首先对XML文档进行解析,提取其中的数据并按照一定的机制存放于一个消息队列中;当MiniGUI运行时便从消息队列中逐个取出所用的数据,并赋值给相应的变量,这样系统界面就会按用户的设计的要求呈现在屏幕上。

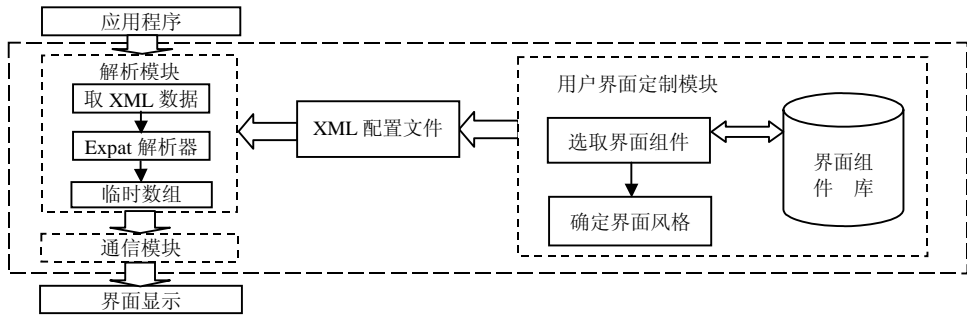


图1 基于XML的可定制界面管理机制图

3 重要模块的设计实现

3.1 用户界面定制模块

这是本系统的核心之一,其中的关键是界面组件库和界面风格的确定。

界面组件库由一组用于描述各种界面元素基本外观、特征、行为的组件,以及一整套用于界面管理的组件所构成,它是实现界面生成与定制的基础;而界面风格则定义了界面元素的个性化特点,使用户可以选择适合自己的界面样式。

在构造界面组件库时,首先对所有界面元素的外观、特征、行为进行分析,提炼它们的共性,并划分出类型和层次;然后采用面向对象的思想,用不同层次的类对它们进行描述,通过继承与派生维护共性。界面元素的外观是通过绘制而成,每个组件从其父类继承绘制方法,可获得标准、统一的视觉效果。界面元素的行为是通过消息来实现人机交互(如鼠标移动消息、点击消息、键盘击键消息等),在界面组件中,通过消息的映射、筛选、响应、转

发等操作处理，可以管理和扩充界面元素的交互方法和能力。

组件库中各界面组件均有各自的标准接口与Linux系统相连接，当被用户选中成为可用状态时，便可与下层系统对应的接口自动完成对接，无需重新设置，实现了用户界面设计数据与应用逻辑模块的分离^[8]。其关系结构如图2所示。

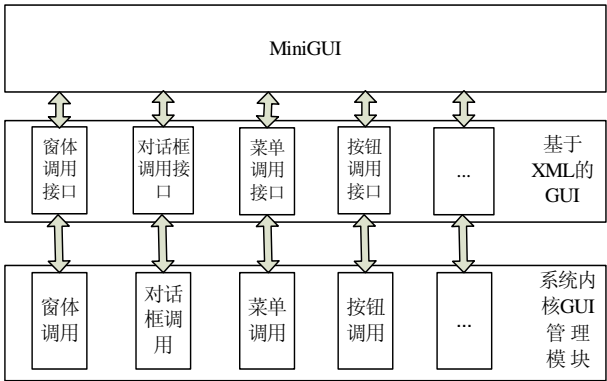


图2 定制模块与下层系统结构关系图

3.2 解析模块

XML文件解析流程如图3所示，其工作原理类似Expat工具包中的XML解析器^[9]。其中主要涉及到一个CxmlParser类，该类中封装了解析一个XML文档的所需函数。主要包括：

(1) Element(eleName): 解析名为eleName的元素的类别，根据其类别将其加入相应的链表；

(2) IterateAttributes: 解析结点的属性值，根据结点的类型调用相关类的OnAttribute函数，将结点的属性值存入相应的变量；

(3) Parser(fileName): 创建一个DOMDocument对象，载入名为fileName的XML文件。该函数利用DOM Document对象和文档对象模型(Document Object Model, DOM)应用程序接口生成一个结点的树型逻辑结构。可通过遍历此DOM树的每一个结点，达到浏览、查询和修改XML文档内容和结构的目的。

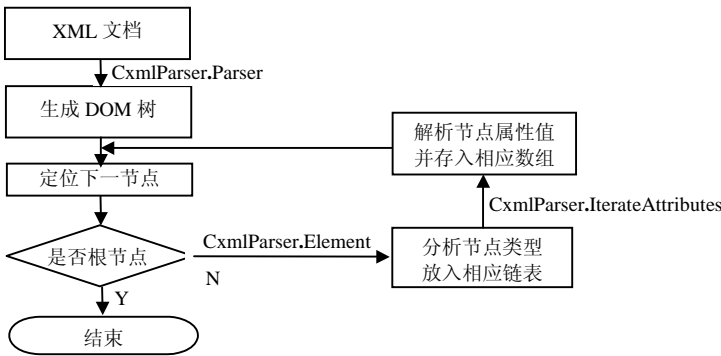


图3 XML文件解析流程图

DOM是一个文档对象组成的模型，对于XML应用开发来说就是一个对象化的XML数据接口。DOM提供了丰富的元素，使用操作灵活，而且可以完成对于XML文档的读取、修改、删除等功能。简单地说，DOM就是一个对象集合，通过操纵这些对象，就能操纵XML数据。

3.3 通信模块

在系统定制过程中，必然会有一定量的数据需要更新并在XML文档解析进程和界面显示进程间进行传递。由于系统定制是一个动态更新过程，其数据传输量不固定，而且考虑到嵌入式系统对实时性的要求，本系统选用了消息队列机制的方法，而XML进程间通信的过程如图4所示。

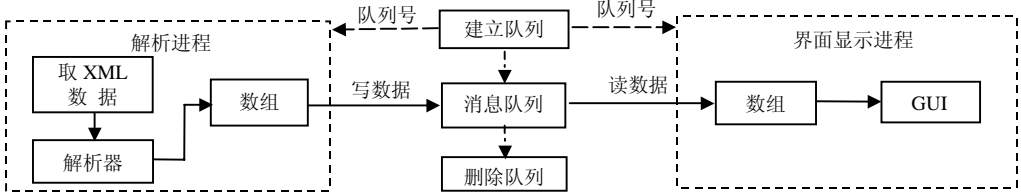


图4 进程间通信原理示意图

3.4 实现方法

下面以控制按钮为例介绍本系统的具体实现方法。XML文档结构如图5所示，图中整个解析实现过程描述如下

(1) 利用前述函数，解析生成如图6所示DOM树。

(2) 根据解析结果生成如下消息队列：
QUEUE ID = 123: “ID_BUT; 1; 确定; WS_

CHILD; 100; 100; 80; 30; ...”

(3) 消息传递给显示进程, 存入数组vary[i](i=0,1,2,3…), 其中vary[0]= “ID_BUT”; vary[1]= “1”; vary[2]= “确定”; vary[3]= “WS_CHILD” …

(4) 调用MiniGUI的按钮创建函数, 显示生成按钮。

...

CreateWindow (CTRL_BUTTON, vary[0], vary[1], vary[2], …);

...

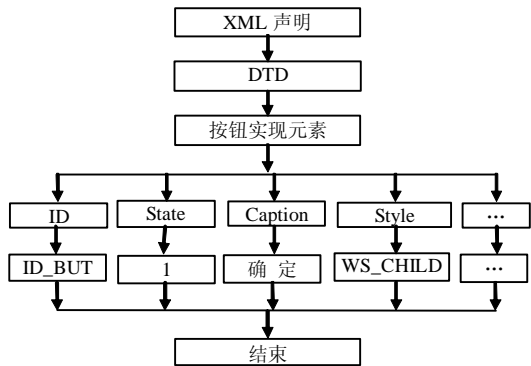


图5 XML文档结构

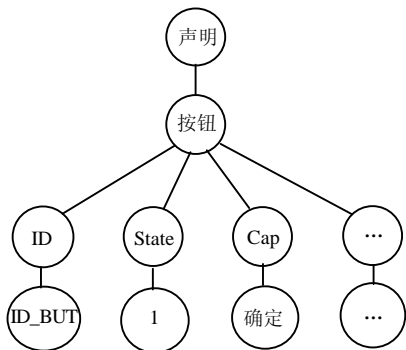


图6 DOM树

4 结束语

本文结合XML等技术, 为嵌入式应用提供了一

种新的用户界面开发和更新方案, 并介绍了基于实现该方案的核心技术。

这种基于XML的用户界面定制的方法不仅可以用户使用界面描述信息与功能实现代码相分离, 简化用户界面的开发与维护, 还可以使软件的最终用户获得与开发人员相同的界面操控能力, 能够在系统运行前和运行过程中对用户界面进行配置、定制和控制等操作。

基于XML本身特有的灵活性, 该方法在协同设计软件和网络服务型软件中也将有非常广阔的应用前景。

参 考 文 献

[1] 王子强. Linux下图形用户界面程序的开发与实现[J]. 计算机应用与软件, 2005, 6: 81-83.

[2] 施奈德曼. 用户界面设计——有效的人机交互策略 [M]. 第3版. 张国印, 译. 北京: 电子工业出版社, 2004: 114-117.

[3] FANG Zhi-gang, WU Xiao-bo, MA Wei-juan. The progress on the study of human-computer interaction technology[J]. Computer Engineering and Design, 1998, 19(1): 59-65.

[4] 祁献鹏, 郭玉东. MiniGUI——面向嵌入式系统的GUI系统[J]. 信息工程大学学报, 2001, 3(2): 8-10.

[5] 王 婷. XML技术在嵌入式系统图形界面开发中的应用 [J]. 武汉科技大学学报, 2005, 28(3): 75-77.

[6] 吴根祥. 基于XML的用户界面描述语言分析[J]. 计算机系统应用, 2004, 12: 28-30.

[7] TIAN Bian, DAI Hang, DAI Guanzhong, et al. Design and application of persistent user interface management system[J]. Chinese Journal of Computers, 2000, 23(6): 660-666.

[8] KASIK D J, LUND M A, RAMSEY H W. Reflections on using a UIMS for complex applications[J]. IEEE Software, 1989, 6(1): 54-61.

[9] 聂 丹. XML在脚本语言PHP中的应用[J]. 丹东纺专学报, 2005, 12(1): 42-44.

编 辑 刘文珍