# 嵌入式软件系统界面可定制模型*

李小将, 胡正国, 陈启安

(西北工业大学 计算机科学与工程系, 陕西 西安 710072)

摘 要: 针对嵌入式软件界面子系统的设计特点, 介绍了进行嵌入式软件系统界面设计的可定制技术, 并提出了一个界面可定制模型。首先阐述了界面可定制模型的设计原理、模型结构和实现技术; 然后介绍了界面可定制模型在嵌入式软件系统界面子系统设计中的一个应用实例。实例证明, 界面可定制技术在嵌入式软件系统设计中是可行和实用的。

关 键 词: 嵌入式软件, 界面, 定制, 界面可定制模型

中图分类号: TP316.2 文献标识码: A 文章编号: 1000-2758(2001)03-0418-04

嵌入式软件系统被广泛应用在PDA(personal data assistant)、STB(set of top box)、HPC(handle PC)、汽车电脑、军用导弹等嵌入式系统中。尤其随着信息家电的出现和应用, 嵌入式软件系统作为信息家电的灵魂, 成为家电产品协同工作和连入Internet的重要软件系统, 具有内核小、组态性、可裁剪性、实时性、支持多任务、多线程、支持网络等特点。系统主要包括嵌入式操作系统核心(含驱动程序)、文件服务、窗口系统服务、网络服务、浏览器、汉字输入、电子邮件服务、界面等子系统。本文着重介绍界面子系统设计中的可定制技术。

所谓嵌入式软件系统的界面, 是指系统在运行过程中的外观表现形式, 以及用户与系统之间进行交互的接口。嵌入式软件系统是一种内嵌于信息家电等产品中的软件系统, 不同的产品界面特性差别很大, 针对不同的应用和不同的用户, 界面的表现形式和功能都有一定的差异, 传统的界面设计方法无法满足嵌入式软件系统的界面设计的需要。因此, 采用界面可定制技术进行嵌入式软件系统界面的设计和开发, 不仅可以满足嵌入式软件系统在不同应用中的界面要求, 也可以满足不同用户的界面要求。同时, 可以满足嵌入式软件系统在不同应用中的界面要求, 也可以满足不同用户的界面要求。同时, 可以节省界面子系统开发的投入, 提高开发效率, 符合软件设计过程中软件重用性、可扩展性和移值性的要求。

## 1 界面可定制模型

界面可定制模型是一种为了满足用户对界面可定制的要求, 遵循提高软件系统开发效率的原则, 依照市场经济的运作模式进行设计的界面模型。主要包括三个部分: 用户对界面语义的定义部分(界面需求); 系统对界面语义的实现部分(界面供应); 界面语义到界面实现的解释部分(界面定制)。

### 1.1 界面可定制的设计原理

根据嵌入式软件系统界面子系统的可定制性要求, 界面子系统必须抽象和概括为一个界面产生器通用模型, 而不是界面的一个具体实现。界面可定制模型的设计原理可以如下描述: 首先, 对界面进行分析和概括, 把界面的外观布局和交互行为抽象为一个个界面元素 $e$, 对每个界面元素 $e$ 用一条语义 $d(e)$(或称为描述规则)进行定义, 所有的定义语义 $d(e)$ 构成集合 $\boldsymbol{S}$, 在集合 $\boldsymbol{S}$ 中, 既定义了界面外观元素的语义 $d_1(e)$(以下称为描述语义), 又定义了界面交互行为元素的语义 $d_e(e)$(以下称为执行语义); 对于 $\boldsymbol{S}$ 中的任何一执行语义 $d_e(e)$, 定义一个实现语义 $i$, 所有的实现语义 $i$ 构成语义实现集合 $\boldsymbol{R}$; 设计一个 $\boldsymbol{S}$ 到 $\boldsymbol{R}$ 的映射函数 $\boldsymbol{F}$。用数学中的三元组表示如下

* 收稿日期: 2000-03-21

作者简介: 李小将(1973-), 西北工业大学讲师、博士生, 主要从事嵌入式系统、人机交互、软件测试的研究。

$$(\boldsymbol{S},\ \boldsymbol{R},\ \boldsymbol{F})$$

$\boldsymbol{S}$为界面布局和交互行为的语义定义集合;$\boldsymbol{R}$为语义实现集合;$\boldsymbol{F}$、$\boldsymbol{S}$ 到 $\boldsymbol{R}$ 的语义映射,即 $\boldsymbol{S}\rightarrow\boldsymbol{R}$。

式中,$\boldsymbol{S}$ 中的元素是 $\boldsymbol{F}$ 的输入,$\boldsymbol{R}$ 中元素是 $\boldsymbol{F}$ 的输出。$\boldsymbol{S}$ 中定义的任何一个执行语义对应一个实现语义的定义 $i$。$\boldsymbol{S}$ 中的描述语义的定义被 $\boldsymbol{F}$ 映射为实现语义定义中的参数。

**1.2**　界面可定制模型的结构

界面可定制模型的结构包括三部分: 界面描述器; 界面实现器; 界面解释器。界面解释器是该模型中实现可定制技术的核心部分, 又包括三个层次: 界面定义语义接口层、界面定义语义到实现的解释层(以下简称界面解释层)、界面实现接口层( 如图 1 所示)。

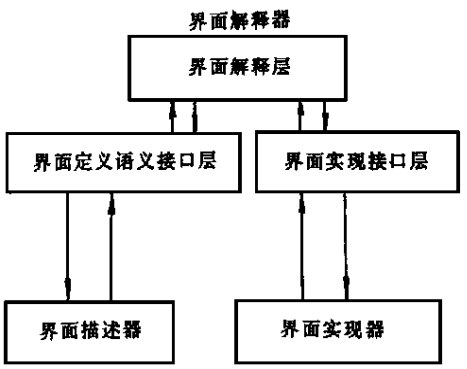


图 1　界面可定制模型的结构

界面解释器: 按照界面定义语义接口层的界面语义定义规范, 对界面定义语义进行解释, 并调用界面实现接口层的实现语义来实现。其中, 界面定义语义接口层相应于上述的集合 $\boldsymbol{S}$, 界面实现接口层相应于上述的集合 $\boldsymbol{R}$, 界面解释层相应于上述的 $\boldsymbol{F}$。

界面描述器: 用户按照界面定义语义接口层的语义规范, 对所需求的界面进行描述, 是上述集合 $\boldsymbol{S}$ 的子集, 对于不同的界面, 描述器具有不同的描述。

界面实现器: 用于提供界面解释器对界面描述器进行解释执行所获得的具体的界面结果。

**1.3**　界面解释器的实现

上述界面可定制模型中, 界面描述器由用户根据自己的需要按照界面定义语义接口的要求填写界面需求单( 界面控制脚本文件), 交给界面解释器, 界面解释器根据界面需求单自动生成相应的界面。下面对该模型中的界面解释器的实现进行介绍。

1.3.1　界面定义语义接口层的实现

嵌入式软件系统内嵌在各种家电设备上, 其用户界面根据设备的功能和用户的特点的不同而具有不同的表现方式。为了减少界面修改时的程序设计工作量, 做到界面的布局变化不用编程序, 界面的功能变化不动系统核心。为此, 对界面的外观布局和交互行为进行分析和抽象, 得到一个个界面元素, 并对界面元素进行定义。然而, 用户界面元素的定义是一个非常复杂的工作, 对复杂的用户界面进行完全描述的语言非常复杂, 只能尽可能地对界面外观布局和交互行为进行抽象和定义并逐步完善。

界面定义语义接口层就是对界面元素的语义进行定义, 通过一套规范的界面控制语言来实现。该语言足以描述界面元素的外观布局属性和交互行为。该语言语法的全集对应于三元组 $(\boldsymbol{S},\ \boldsymbol{R},\ \boldsymbol{F})$ 中集合 $\boldsymbol{S}$。任何一个具体的用户界面可由界面控制语言描述的一个界面控制脚本来实现( 参见 2.1) 。

1.3.2　界面解释层实现

界面解释层的实现包括两部分, 用算法描述如下:

(1)　初始化部分

① 面控制脚本进行词法分析并进行归类;

② 按照界面定义语义接口层定义规范进行语法检查, 如有错, 提示界面描述错误信息并退出系统, 否则执行③;

③ 按照界面控制脚本中的界面元素定义, 建立界面元素属性表;

④ 定位到脚本 START 标号位置, 用界面元素属性表中属性值做参数, 调用界面实现层接口, 按照执行语义逐条实现初始界面;

⑤系统等待接受界面驱动消息。

(2)　消息处理部分

① 系统接受界面驱动消息;

② 识别驱动消息类型;

③ 根据驱动消息类型定位到界面脚本执行起点;

④ 从脚本的执行起点, 按照执行语义逐条实现相应界面, 如执行到界面控制脚本中的系统退出语义, 退出系统, 否则系统等待接受界面驱动消息, 重复上述过程。

1.3.3　界面实现接口层的实现

该层对界面元素中执行语义的实现接口规范进行定义, 包括所有实现接口的函数名、参数类型、参数个数、返回值及接口功能的定义。该层不涉及接口的具体实现形式, 但接口的具体实现必须遵循规范中的定义。该层接口可以采用底层的图形用户接口

(GUI) 来实现。举例说明如下

假设界面定义语义接口层中定义打开一个新的浏览器页面的执行语义为

exec HTMLPAGE
URL = '表示新页面地址的字符串'
end exec

在界面实现接口层定义相应的实现接口为

void Browser _ NewPage(Browser * pBrowser, const char * pURL);

函数名: Browser _ NewPage

函数功能: 在浏览器中打开一新页面。

参数说明:

pBrowser: 指向一浏览器对象的指针, 为空给出出错信息

pURL: 指向页面地址字符串指针

返回值: 无

## 2 嵌入式软件系统界面可定制模型的应用实例

笔者在参与嵌入式软件系统研制过程中, 设计并实现了一个相对独立的用户可定制界面子系统 ICSS(interface of customization subsystem), 按照界面可定制模型的三层结构对 ICSS 介绍如下。

### 2.1 界面控制语言(interface control language)

该语言是上述模型中界面语义描述层的一个简单实现, 该语言对界面元素语法和界面控制脚本工作方式进行了比较完善的定义。它的语法定义主要包括两类定义, 即

· 基本定义

定义了常量、变量、运算符、表达式、赋值语句、条件语句、执行语句、返回语句、转移语句、菜单定义语句、消息处理定义语句、标号语句、注释语句等。

· 高级定义

定义了系统函数、系统预定义对话框、系统消息、系统工具栏、系统菜单等。

由于篇幅限制, 下面仅以执行语句和菜单定义语句为例说明该语言的语法定义。

执行语句::= exec 操作类型
{操作参数描述}
end _exec

操作类型::= MENU | DIALOGBOX | HTMLPAGE | REFRESH

操作参数描述::= 参数名 '=' 表达式

参数名::= 标识符

标识符::= 字母{字母| 数字| '_'}

表达式::= (由"表达式"语法定义给出, 下同)

(2) 菜单定义语句::= menu 菜单名字
菜单属性定义语句
{菜单项定义语句}
end _menu

菜单名字::= 字符串常量| 字符串变量

字符串常量::= '"' 除引号之外的任意内容 '"'

字符串变量::= '$' 标识符

菜单属性定义语句::= X 坐标', 'Y 坐标[', '栏数[', '宽度[', '高度]]]

X 坐标::= 表达式| LEFT| RIGHT| CENTER

Y 坐标::= 表达式| TOP| BOTTOM| CENTER

宽度::= 表达式

高度::= 表达式

栏数::= 表达式

菜单项定义语句::= menuitem[菜单项名字]
{菜单项属性定义}
{语句}

菜单项名字::= (同"菜单名字")

菜单项属性定义::= 菜单项属性名 '=' 表达式

菜单项属性名::= TEXT| ICON| SELECT| ENABLE
| DISPLAY| TEXTCOLOR

### 2.2 界面控制脚本解释器(interface control-script interpreter)

在进行解释之前, 为了保证界面控制脚本的正确性, 界面子系统提供独立的模块对界面脚本进行语法检查, 提示用户是否有语法错误, 提交给界面解释器解释的脚本为经过语法检查的正确的脚本。

界面控制脚本的工作方式是消息触发的方式, 它没有一个连续不断的执行路线。平时界面控制脚本处于静止状态, 只有发生能够导致界面活动的消息时, 界面操作程序才根据界面操作脚本的指示对消息进行处理。该语言定义了三种界面消息: 系统起动消息; 浏览器消息; 菜单项选择消息。界面解释程序接收到消息后, 根据消息的类型定位到界面操作脚本中的某一条语句, 然后从该语句开始顺序执行下面的语句, 直到执行到一条"执行语句"或"返回语句", 或一个语句定义块的终止时, 这一次消息处理操作完成。

鉴于界面控制脚本的工作方式只有三种消息驱动,界面控制脚本解释器实际上只采用了三个子模块,分别完成三种不同消息下语义的解释。同时,为了提高系统的执行效率,采用了预扫描、建立链表的方法,使系统经过几次扫描后可以多次使用。

**2.3** 界面语义实现静态库

所有界面元素执行语义通过嵌入式系统 API 来实现,所有的接口定义在静态库头文件中,所有接口的具体实现在语义实现静态库中。

该子系统在具体系统运用中有两种形式:(1) 直接作为嵌入式软件系统的界面子系统,界面解释器对用户定义的界面控制脚本进行解释,产生用户定制的界面;(2) 界面控制语言作为 HTML 的一个扩充,把界面控制脚本嵌入到 HTML 脚本中,在浏览器系统中嵌入界面解释器,产生用户的定制界面。上述两种形式的使用都是可行并且有效的。

## 3 结　论

以上对界面可定制模型的原理、结构和实现方法及应用实例进行了阐述,介绍了一种嵌入式软件系统界面设计的通用模型。该模型在实现用户的界面需求,节省系统开发时间,提高系统可维护性,可移植性等方面,与传统的界面设计方法相比,都具有明显的优势。该模型不但可以应用于嵌入式软件系统设计中,在普通的软件界面设计中也是有参考价值的。该课题在界面定义的充分性和完备性方面还需要进一步完善。进一步,在界面可定制模型中,建立用户界面行为的知识库,引入用户界面自适应机制,对于构造良好的人机界面将会具有更重要的意义。

## 参考文献:


[1] Magnus L. User-Interface Modelling——Adding Usability to Use Case. Int J Human-Computer Studies, 1999, 50: 243～262

[2] Harton H R, Hix D. Human-Computer Interface Development: Concepts and System for Its Management. ACM Computing Surveys, 1989, 21(1): 5～92

[3] Booch G, Jacobson I, Rumbaugh J. Version 1.0 of the Unified Modelling Language(on-line). Available: http://www.rational.com/uml/documentational.html

[4] Morse T, Jan T. Compositional Modelling of Reflective Agent. Int J Human-Computer Studies, 1999, 50: 407～431

[5] Constantine L. Essential Modelling: Use Cases for User Interfaces. Interaction, 1995, 2: 34～36


# An Interface Customizable Model for Embedded Software System


Li Xiaojiang, Hu Zhengguo, Chen Qi'an

(Department of Computer Science &Engineering, Northwestern Polytechnical University, Xi'an 710072)



**Abstract**: Traditional methods for designing interface, in our opinion, cannot satisfy the needs of interface design for embedded software system. We purpose a new technology of interface customization. In subsection 1.1, we discuss the design principles of interface customization. In subsection 1.2, we discuss the structure of interface customizable model(ICM). Fig. 1 shows that ICM consists of three modules: interface descriptor, interface implementor, interface interpreter. Interface interpreter is the core of ICM. In section 1.3, we discuss implementation technique of ICM interpreter. We used the new technology of interface customization to design and implement an interface customization subsystem, which, since the end of 1999, has formed a part of a successfully running embedded software system.




**Key words**: embedded software, interface, customization, interface customizable model