

# 一种实现控件与多语言文本动态绑定的方法

叶 蓉<sup>1,2</sup>, 陈 榕<sup>1</sup>

(1. 同济大学 基础软件工程中心, 上海 200092;

2. 上海第二工业大学, 上海 201209)

**摘 要:**现代应用中大多已有进行多语言切换的需求,但对于控件本身文本显示信息,要实现多语言实时切换还有一些难度,文中提出一种能够根据用户的选择将控件与显示文本进行动态绑定显示的方法,通过对控件文本信息的多语言实时切换。该方法具有执行效率较高、定制选择灵活方便、性能稳定可靠、适用范围较为广泛的特点。该方法已在基于CAR构件的Elastos操作系统上实现并运用于智能手机应用。

**关键词:**多语言;动态绑定控件;CAR构件;Elastos操作系统

**中图分类号:**TP311

**文献标识码:**A

**文章编号:**1673-629X(2008)01-0037-03

## A Method of Multi-Lingual Dynamic Banding Component

YE Rong<sup>1,2</sup>, CHEN Rong<sup>1</sup>

(1. System Software Engineering Centre of Tongji University, Shanghai 200092, China;

2. Shanghai Second Polytechnic University, Shanghai 201209, China)

**Abstract:** Nowadays, more and more software application want the control can display text real-time and multi-lingual. In this paper a dynamic banding of control and text display method is proposed, through real-time and multi-lingual switch of text control information. This method has the advantage of better execution efficiency, customization and flexibility, as well as reliable performance and the wider applicable scope against traditional way. It has been implemented on the CAR based Elastos operation system, and has been applied to various smartphone applications.

**Key words:** multi-lingual; component dynamic banding; CAR component; Elastos operation system

## 0 引 言

现代的计算机系统中,已经越来越多地支持多语言功能,但是对于控件本身的文本显示信息,要实现多语言的实时切换,是非常困难的。不仅如此,在手机的应用中,经常也会涉及到控件文本的动态变换,比如多语言的切换。在现有技术中,传统的控件无法方便地做到文本的动态绑定,必须由应用开发者自己去处理这些琐碎的文本替换工作,这样就大大限制了计算机多语言技术的进一步发展。

文中将提供一种能够根据用户的选择将控件与显示文本进行动态绑定显示的方法<sup>[1]</sup>。该方法包括系统进行多语言控制模块加载操作、将预定义的文本传递给控件、控件检查所述的文本是否属于特定编码,并判

断多语言控制模块是否加载成功、根据判断结果分别进行普通显示文本保存和显示处理操作或者将该文本转换成相应的数字标识并保存同时根据该数字标识调用多语言控制模块取得对应的显示文本信息行保存并显示、系统根据用户请求进行文本语言种类的设置操作并触发相应的控件、控件调用多语言控制模块取得对应的显示文本信息保存并显示<sup>[2,3]</sup>。

该方法具有执行效率较高、定制选择灵活方便、性能稳定可靠、适用范围较为广泛的特点。

## 1 控件与多语言动态绑定方法的实现过程

通过如下四个步骤,可实现本方法中控件与多语言文本动态绑定显示:

(1)第一步:加载多语言模块。

首先从指定的位置加载多语言模块,如果成功,则保留该接口指针,如果失败,则接口指针为空。其具体实现如图1所示。

(2)第二步:设置控件初始文本。

收稿日期:2007-03-31

基金项目:国家“863”计划资助项目(2001AA113400)

作者简介:叶 蓉(1980-),女,江苏南京人,硕士研究生,研究方向为嵌入式操作系统、系统软件支撑技术;陈 榕,教授,博士生导师,研究方向为嵌入式系统、构件技术。

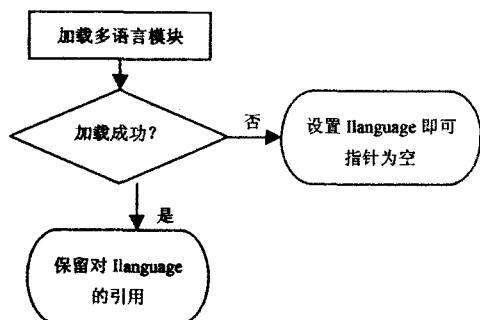


图1 加载多语言模块

当控件初始化时,需要设置一个初始的文本。用户可以将预定义好的特定编码的文本设置给控件,设置文本时,控件检查该文本是否属于特定编码的,如果多语言接口为空,或 ConvertToID 转换失败,则当作普通文本保存起来。如果检查成功,则将该编码转换成相应的数字 ID,并调用多语言接口的 GetText 方法,得到相应的文本,并保存起来。其具体过程如图 2 所示。

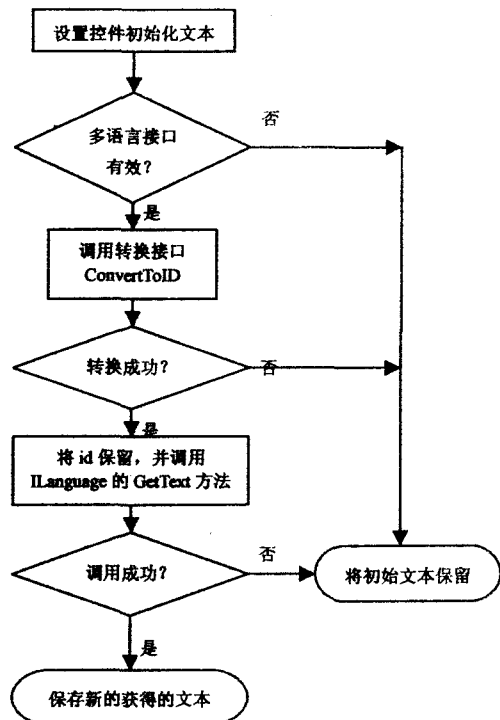


图2 设置控件初始文本

(3)第三步:运行时设置控件文本。

该步骤同第二步,需要说明的是,设置新文本时会替换掉原先保存的文本。

(4)第四步:语言的切换。

调用 IGShell 接口的 SetLanguage 方法,多语言接口为空,则调用失败,如果 SetLanguage 调用成功,则该方法就会激发一个文本改变的事件,如果 SetLanguage 调用失败,则不激发事件。其实现如图 3 所示。

(5)第五步:事件的响应。

当控件收到该事件时,首先检查多语言接口 ILan-

guage 是否有效,如果无效则不作任何操作,如果有效,则调多语言的 GetText 方法,如果该函数执行成功,则替换掉原来保留的字符串,如果该函数执行失败,则不做任何操作。其过程如图 4 所示。

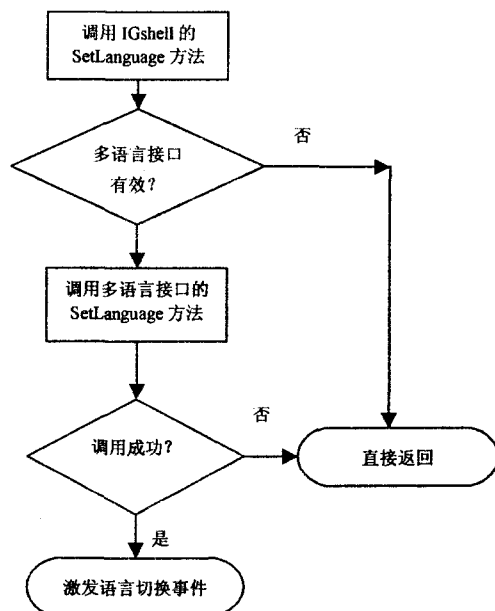


图3 语言的切换

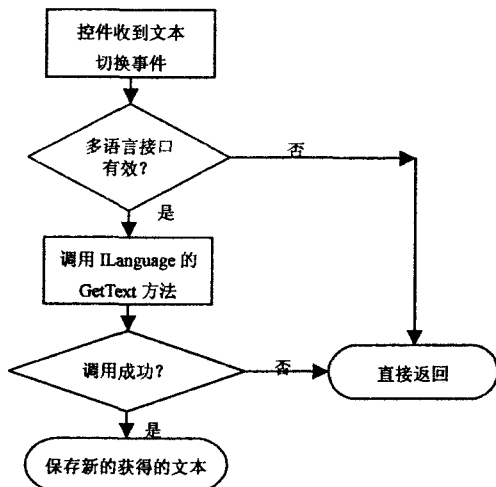


图4 事件的响应

(6)第六步:控件的刷新。

当完成一次语言切换时,控件自己做一次更新,将当前的文本画出来。

## 2 实现多语言动态绑定的接口定义

### 2.1 接口定义

定义 ILanguage 接口实现数字 ID 到字符串之间的转换:

```
interface ILanguage
```

```
{
```

```
    SetText([in] INT nTextID,[in] EzStr esText)//通过 ID 设置
```

字符串资源

```
GetText([in] INT nTextID,[out] EzStrBuf esbText)//通过 ID
获取字符串资源
ConvertToID([in] EzStr esText,[out] INT * pTextID);//把
特定编码格式转换成整型 ID 值
GetLanguageID([out] INT * pLanguageID);//获取当前语言
ID
SetLanguageID([in] INT nLanguageID);//设置当前语言 ID
```

其中各接口的实现过程说明如下:

\* SetText 方法可以动态地建立字符串和数字 ID 的映射关系。

\* GetText 方法获得数字 ID 对应的字符串。

\* ConvertToID 方法将一个特定编码格式转换成一个数字 ID,比如,将字符“1234”转换成数字的 1234。

\* GetLanguageID 获得了当前语言的类型,如中文,英文。

\* SetLanguageID 设置了当前语言的类型,如中文,英文。

同时,ILanguage 构件本身只是提供一个抽象的接口。用户只要实现该接口的指定语义,就能被系统正确地调用。系统调用方并不关心接口实现方是使用文件或数据,还是从网络,只要返回数字 ID 对应的字符串就可以了。

接口实现的示例代码片断如下:

```
SetText([in] INT nTextID,[in] EzStr esText)
{
    database.add(nTextID,esText);//在数据库中添加一个映射关系
}
GetText([in] INT nTextID,[out] EzStrBuf esbText)
{
    database.get(nTextID,esText);//从数据库中获得一个映射关系
}
ConvertToID([in] EzStr esText,[out] INT * pTextID)
{
    * pTextID = decode(esText);//返回特定编码格式到数字 ID 的转换
}
SetLanguageID([in] INT nLanguageID)
{
    currentLanguageID = nLanguageID;
    //根据设置的当前语言环境,选择不同的数据库
    if (nLanguage == CHINESE) {database = database.. Chinese;}
    if (nLanguage == ENGLISH) {database = database.. English;}
```

```
.....
}
GetLanguageID([out] INT * pLanguageID)
{
    return currentLanguageID; //返回当前的语言环境
}
```

## 2.2 定义激发多语言切换事件的接口

IGShell 接口实现多语言的切换并激发相应的事件:

```
Interface IGShell
{
    SetLanguage(INT nLanguageID); //设置当前的语言
    AddEventHandler([in] INT nEvent,[in] EzDelegate edHandler); //注册文本改变事件
    RemoveEventHandler([in] INT nEvent,[in] EzDelegate edHandler); //注销文本改变事件
    ...
}
```

## 3 多语言与控件动态绑定方法的具体运用

以和欣智能手机拼图游戏为例,说明应用实施过程<sup>[4,5]</sup>:

```
ECODE CPuzzleMainForm::Init()
{
    .....
    ec = m_ pLanguage -> GetText (IDS_ GameSteps, m_ wsb-
    Step); //在初始化时获取
    CForm::AddLanguageChangedCallback(m_ pForm, this,
    &CPuzzleMainForm:: OnLanguageChanged); //在主窗体上
    注册语言的回调事件
    .....
}
ECODE CPuzzleMainForm::OnLanguageChanged(/* [in] * / I-
    Form * pSender)
    //当语言发生变化时将触发该回调事件
{
    .....
    ec = m_ pLanguage -> GetText (IDS_ GameSteps, m_ wsb-
    Step);
    //将重新获取当前语言对应的控件文本
    .....
}
```

图 5 及图 6 给出了在拼图游戏中,菜单控件的中英文切换的效果图<sup>[6]</sup>。

## 4 总 结

采用了上述的计算机系统中控件与多语言文本动态绑定显示的方法,由于系统中将控件和对应的显示

(下转第 43 页)

容忍上限或低于容忍下限的评价,系统将认为这种评价为非准确性评价,从而自动进行剔除。

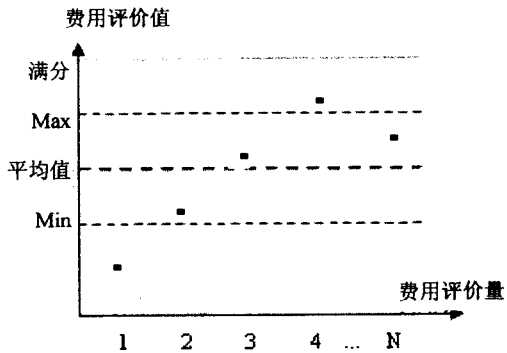


图1 某网格服务的费用评价控制示意图

对于8个评价指标都可以采用上述方法来进行非准确性评价控制。至于是采用定时还是定量进行处理可以根据系统特定情况进行选择,灵活处理。

#### 4 总结及展望

网格服务的质量好坏是能否成功构建服务网格的基础。文中通过建立一种评价网格服务质量的拥有8个具体指标的指标体系,分别涉及了服务提供者、服务使用者 and 系统三个角色,分别对8个指标进行打分或者从系统取值,并根据服务质量问卷获取的权值最终确定每个服务的具体指标值。同时,还可以对打过分

程中可能存在的非准确性评价进行控制,剔除超出容忍的评价,起到了保证评价质量,进而保证网格服务质量,更好为网格服务用户和网格服务提供商建立一个良性的服务网格环境的作用。文中多侧重于单网格服务的评价策略上,对于组合网格服务以及其中单个网格服务的作用大小没有进行深入的研究,这是后续工作中的重点。

#### 参考文献:

- [1] Foster I, Kesselman C. The Grid2: Blueprint for a new Computing Infrastructure[M]. [s.l.]: Morgan Kaufmann Publishers, 2004.
- [2] Foster I, Kesselman C, Nick J M, et al. Grid services for distributed system integration[J]. Computer, 2002, 35(6): 37-46.
- [3] Foster I. On recent changes in the grid community[J]. IEEE Distributed Systems Online, 2004(5): 2-3.
- [4] Ran S P. A model for Web services discovery with QoS[J]. ACM SIGecom Exchanges, 2003, 4(1): 1-10.
- [5] Mani A, Nagarajan A. Understanding quality of service for Web services[EB/OL]. IBM, 2002. <http://www-106.ibm.com/developerworks/library/ws-quality.html>.
- [6] Cardoso A J S. Quality of service and semantic composition of workflows[D]. Georgia: University of Georgia, 2002.

(上接第39页)



图5 中文版菜单控件

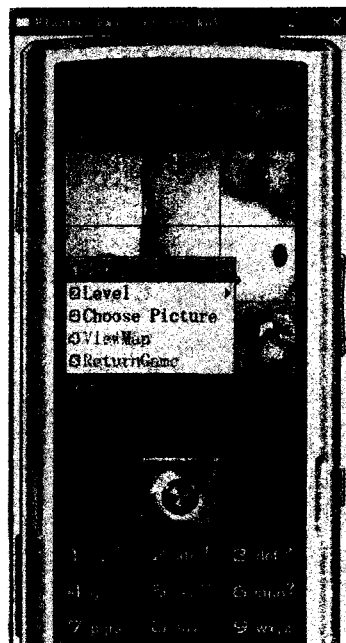


图6 英文版菜单控件

文本信息根据不同的语言种类进行了分离,并在控件中设置了对应于显示文本信息的数字标识,根据设置

的语言种类和数字标识利用多语言控制模块进行文本信息的提取,从而实现了控件与多语言文本的动态绑定和显示;同时,该方法不仅执行效率较高,而且用户的定制选择灵活方便,性能稳定可靠,适用范围较为广泛,为计算机系统和手持设备的嵌入式系统的多语言技术的进一步发展奠定了坚实的基础。

#### 参考文献:

- [1] Koretide. Elastos2. 0Manual[EB/OL]. 2006. <http://www.koretide.com.cn/download/download.php?id=2> 2007.
- [2] Pan Aining. COM's Principle and COM's Application [M]. Beijing: The Tsinghua Press, 1999.
- [3] Rogerson D. Inside COM: Microsoft's Component Object Model[M]. [s.l.]: Microsoft Press, 1999.
- [4] Eckel B. Thinking in C++ [M]. 2nd ed. [s.l.]: Prentice Hall, 2002.
- [5] Koretide. Website[EB/OL]. 2007. <http://www.koretide.com.cn>.