

文章编号: 1000-5013( 2007)01-0107-02

# 多语言用户界面的实现技术分析

张剑达, 张全伙

( 华侨大学 信息科学与工程学院, 福建 泉州 362021 )

摘要: 分析和比较 4 种主要的多语言用户界面的实现技术, 列出实现的具体步骤, 评价其实现机制特点和应用场合. 以 Windows 为平台, 给出在应用系统中实施这些技术的方案.

关键词: 多语言用户界面; 动态链接库; 实现机制; Visual C++

中图分类号: TP 311. 11 文献标识码: A

## 1 界面实现技术方法分析

以 Visual C++ 为开发工具, 同时假定应用程序支持简体中文、繁体中文和英文 3 种语言, 分别对下面 4 种多语言用户界面的实现技术加以讨论并描述它们的实现过程.

### 1.1 多独立应用程序技术<sup>[1]</sup>

采用多独立应用程序技术, 必须为支持的每种语言建立一个项目. 每个项目包含相应的语言资源, 可以套用单一语言软件项目的开发方法. 采用这种技术的主要优点是实现简单, 能够比较彻底地实现整个软件的国际化. 其主要缺点表现在以下 4 个方面. (1) 每种语言都需要一套源代码. (2) 对资源所作的任何改变都需重新编译整个项目. (3) 不同语言界面间的切换需要启动不同的应用程序实例. (4) 每种语言都需要占用一个完整的应用程序的磁盘空间, 造成磁盘空间的严重浪费.

### 1.2 多语言的纯资源 DLL 技术<sup>[2]</sup>

采用该技术只需建立应用项目和资源项目, 前者包含了整个应用程序的应用逻辑, 后者包含了应用程序支持的所有语言的资源, 生成的是纯资源的 DLL (Dynamic Link Library) 文件. 该技术的主要优点是, 可以实现平滑地在不同的语言环境中切换, 版本更新不需要重新编译资源项目(前提是更新不涉及到资源). 其缺点主要在于, 添加新的语言需要重新编译整个资源项目, 比较费时; 纯资源 DLL 包含大量没有用到的资源, 导致磁盘空间尤其是内存空间的严重浪费. 该技术的主要实现步骤有 6 个. (1) 创建资源项目 MyRes, 编译链接生成纯资源 DLL 文件 MyRes. dll. (2) 在 CMResApp::InitInstance() 函数中, 调用 API 函数 LoadLibraryEx 以便应用程序 MyRes. exe 在运行时, 将 MyRes. dll 文件载入其进程空间. 同时, 该函数将返回 MyRes. dll 文件的句柄 hMyResDLL. (3) 选择需要切换语言的界面元素. 首先调用 EnumResourceLanguages 函数, 然后通过自定义的同调函数 EnumResLangProc, 可以得到用户界面包含的所有语言标识符. (4) 在主菜单初始化函数中, 将第 3 步得到的所有语言标识符对应的语言类别添加到界面语言切换菜单中. (5) 在响应用户选择界面语言切换菜单的各个菜单项的函数中, 根据用户选择的菜单项, 对所有需要切换语言的界面元素调用 FindResourceEx 函数, 可得到某种语言版本的所有界面元素的资源句柄. (6) 在获取资源句柄后, 加载并在用户界面上显示这些资源.

### 1.3 多独立单资源的纯资源 DLL 技术<sup>[3]</sup>

与前一种技术的基本思想大致相同, 主要区别是它把所有的语言资源统一封装在一个纯资源 DLL 中, 而后者把各种不同的语言从这个资源 DLL 中分出来, 为每种语言生成一个纯资源 DLL. 同样地, 它需要用 LoadLibraryEx 函数来装载资源 DLL. 所不同的是, 其装载选定的某种语言的纯资源 DLL. 该技

收稿日期: 2006-03-24

作者简介: 张剑达( 1978- ), 男, 讲师, 主要从事软件工程与软件架构的研究. E-mail: jian-da@hqu.edu.cn.

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

术除具备了多语言的纯资源 DLL 技术的所有优点外, 还容易加入对新语言的支持; 对某种语言资源进行更新只需重新编译链接其对应的资源项目; 用户可选择性地安装其中的一种或几种纯资源 DLL. 而其缺点主要是, 需要同步维护所支持的语言资源项目, 维护的工作量较大. 具体实现步骤有 3 个. (1) 根据应用程序支持的语言种类创建相应数量的资源项目, 编译链接生成对应的纯资源 DLL 文件; 并将这些 DLL 文件按照类型命名惯例进行命名. (2) 在应用项目的菜单初始化函数中, 提取纯资源 DLL 文件的语言标识符, 以语言标识符对应的语言类别为菜单项的文本生成语言切换菜单. (3) 在语言切换菜单的响应函数中, 根据用户选择的菜单项的语言标识符, 加载相应的纯资源 DLL 文件, 获取其句柄; 调用 AfxSetResourceHandle 函数, 将得到的句柄设置为应用程序的资源句柄, 同时将之前的纯资源 DLL 卸载, 释放其占用的资源.

1.4 引用外部语言包技术

采用该技术只需要创建一项目(即应用项目), 资源的文本保存在外部语言包中. 程序运行过程中, 以动态调用的方式来显示外部语言包中的资源文本. 因而, 该技术除具有前一种技术的所有优点外, 还有便于添加大量新的语言的优点. 正因为这些特点, 使得该技术非常适合个人或中小型软件企业开发多语言用户界面的共享软件. 其缺点主要表现在: 需要在应用项目中添加大量处理程序运行过程中动态显示外部语言包中资源文本的代码, 这对于大型项目来说, 开销相当可观. 该技术的主要实现步骤有 5 个. (1) 创建外部语言包, 语言包按照资源类型分类保存应用程序中资源的标识符与资源的文本, 并将其按“键-值”对的形式进行组织. (2) 在应用项目中创建使用外部语言包的类, 类主要包含加载外部语言包, 读取语言包中的“键-值”对等操作. (4) 在应用项目的菜单初始化函数中, 提取外部语言包文件的语言标识符, 以语言标识符对应的语言类别为菜单项的文本生成语言切换菜单. (5) 在语言切换菜单的响应函数中, 根据用户选择的菜单项的语言标识符, 加载相应的外部语言包文件. 通过读取语言包文件中的有关“键-值”对, 动态地生成所有需要更换界面语言的界面元素.

2 结束语

在实际应用这 4 种技术应当将下面一些问题考虑进去. 譬如, Windows 操作系统会优先按系统的语言版本显示软件的用户界面, 如果软件不包含该语言资源, 则系统通常会将用户界面的语言默认为英语, 这对于某些用户来说是难以接受; 在应用程序中调用的 Windows 通用对话框, 默认采用系统的语言来显示, 在切换用户界面的语言之后, 会产生与界面显示语言不一致的情况.

参考文献:

[1] 唐 勇, 李秀龙, 陈 晶, 等. 多语言用户界面的研究与实现[J]. 计算机应用研究, 2002, (4): 112-113.  
[2] 孙昌爱, 金茂忠, 段 彬. 开发 Visual C++ 多语言界面的方法[J]. 微型机与应用, 2001, (7): 7-8.  
[3] 谢又成, 任 萍, 章 兢. 基于纯资源 DLL 多语言编程的实现[J]. 微计算机应用, 2003, 24(3): 191-193.

Analysis of the Implementation Techniques for  
Multi-Language User Interface

ZHANG Jiar-da, ZHANG Quan-huo

(College of Information Science and Engineering, Huaqiao University, Quanzhou 362021, China)

**Abstract:** In this paper, implementation technique for four majors multi language user Interface were analyzed and compared. The implementation method, respective features and application situation were shown. The schemes of using these techniques to the application systems based on Windows platform were offered.

**Keywords:** multi language user Interface; DLL (dynamic link library); implementation mechanism; Visual C++

(责任编辑: 黄仲一)