

# 平台+插件模式下的中文软件国际化

刘伟<sup>1</sup>, 蒋胜平<sup>2</sup>, 张玉刚<sup>3</sup>

(1. 中国地质大学 信息工程学院, 湖北 武汉 430074;

2. Autodesk 公司, San Rafael 94903;

3. 华中师范大学 计算机学院, 湖北 武汉 430074)

**摘要:**近年“平台+插件”软件开发模式广泛应用于各大软件的开发应用中。随着经济全球化的发展,中文软件要在国际软件业中更好的发展,关键要走中文软件国际化这条道路。讨论了平台/插件的模式结构,根据软件国际化的理论及规则,提出了该模式下中文软件国际化的一般方法,给出一个基于DLL实现的平台/插件模式中文软件多语言版本的实例。

**关键词:**平台/插件;软件国际化;Unicode

**中图分类号:**TP311.5

**文献标识码:**A

**文章编号:**1673-629X(2009)03-0151-04

## Internationalization of Chinese Software in Platform/Plug-in Model

LIU Wei<sup>1</sup>, JIANG Sheng-ping<sup>2</sup>, ZHANG Yu-gang<sup>3</sup>

(1. Faculty of Information Engineering, China University of Geosciences, Wuhan 430074, China;

2. Autodesk Company, San Rafael 94903, USA;

3. School of Computer, Central China Normal University, Wuhan 430074, China)

**Abstract:** In recent years, the “platform/plugin-in” model of software is widely used in various software development and applications. Accompanying with the development of economic globalization, in order to develop Chinese software better in the international software industry, the key is the internationalization of Chinese software. In this paper, discuss the structure of the platform/plugin-in model, according to the theory and rules of international software, raise the general approach of internationalization of Chinese software in this model, and achieve a multi-language interface by extracting resource of Chinese software. Finally, give an example of multi-language version of the platform/plugin-in model by using a set of resource-only DLL.

**Key words:** platform/plugin-in; software internationalization; Unicode

## 0 引言

随着经济全球化进程的推动,软件设计技术的迅猛发展,国际市场需求的高速增长,促使软件走向国际市场。在大型的国际软件开发中,超过60%的利润来自国际市场。软件能否进入国际市场并占有较大的市场份额,与软件的国际化及本地化紧密相关。软件国际化的实现是软件开发的重要部分。软件的国际化程度越高,实现软件本地化越容易。

而近几年出现的“平台+插件”软件开发的模式,已广泛应用于各大软件的开发应用中,例如,PhotoShop等常见的软件中为实现一些辅助的功能而嵌入

的各种各样的插件;国产GIS软件MapGIS7.x的开发等等,都广泛应用了“平台+插件”的软件开发模式。

文中是在现有的平台+插件模式下的软件架构的基础上,根据软件国际化的理论及规则,提出了平台+插件模式下中文软件国际化的一般方法。

## 1 基本概念

### 1.1 平台/插件软件架构的思想

“平台+插件”模型的主要思想是将扩展功能以插件的形式通过平台统一地管理起来,在平台内部提供平台和插件之间以及不同插件之间完备的消息机制(包括系统消息转发、框架内部自定义消息),对不同扩展功能进行分类并定义标准接口,从而把不同的功能插件有机地集成到一起,有效地协同工作<sup>[1]</sup>。

平台+插件软件结构是将一个待开发的目标软件

收稿日期:2008-07-12

作者简介:刘伟(1982-),男,硕士研究生,从事中文软件国际化与遥感数字图像处理方向研究;蒋胜平,美国Autodesk公司软件高级工程师,博士,教授,硕士生导师,从事软件国际化理论研究。

分为两部分,一部分为程序的主体或主框架,定义为平台,另一部分为功能扩展或补充模块,可定义为插件。

插件的本质在于不修改程序主体(平台)的情况下对软件功能进行扩展与加强。当插件的接口公开后,任何公司或个人都可以制作自己的插件来解决一些操作上的不便或增加新的功能,也就是实现真正意义上的“即插即用”的软件开发,从而很方便地对应用程序进行扩展。

## 1.2 软件国际化与本地化

软件国际化是开发一个程序内核的过程。在软件设计和文档开发过程中,使功能和代码设计能处理多种语言和文化。其功能设计和代码设计不是基于某种语言或单一地域的信息,而是支持多种语言和多个地域的数据,从而在创建不同语言版本时,不需重新设计或修改源程序代码。软件的本地化是针对某一地域所支持的编码字符集、语言和地域习惯为软件建立符合本地要求的信息的过程。包括转换用户界面、调整对话框、用户化功能,以及测试结果以保证程序仍然工作。

软件国际化和软件本地化是国际化软件的开发包括两个互相联系的阶段。软件国际化是软件全球化的实现基础,是有效实现软件本地化的根本保证。软件本地化是国际化软件的另一个关键过程,它是本地化项目管理、软件界面和文档翻译,软件手册的桌面印刷排版,本地化软件编译,软件测试和质量保证等一系列软件工程活动的集合。

国际化软件开发的一般过程见图 1。

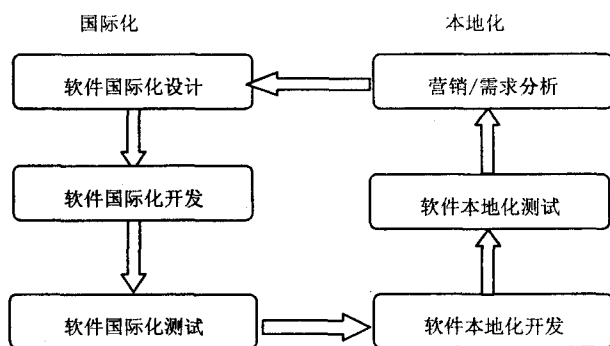


图 1 国际化软件开发的一般流程

## 2 软件内核的国际化

中文应用程序软件要具有国际化能力,必须遵循国际化标准,即 Unicode 标准<sup>[2,3]</sup>。在中文软件中处理的字符即是中文简体字符,是双字节格式<sup>[4]</sup>。一旦从中文双字节格式转换到 Unicode,支持英文和其他语言将会很容易做到。因此,平台+插件模式的中文软件其内核也分别要支持 Unicode 标准。

### 2.1 平台和插件分别创建 Unicode 的应用程序,并实现 Unicode 化

(1)定义字符变量或字符常量时使用 Unicode 数据类型。

Windows 针对 16 位的 Unicode 字符定义了一些 Unicode 数据类型。它方便了人们在开发应用程序时 Unicode 的编程。比如这些类型有:Unicode 字符类型 WCHAR,指向 Unicode 字符串的指针变量类型 PW-STR,指向 Unicode 字符串常量的指针变量类型 PSW-STR 等。

(2)使用 String 操作函数的通用类型。

转换基于 ANSI 的 String 操作函数为通用类型。对于 String 函数,这些宏都以 \_tcs 为前缀,根据正确的预处理器定义映射到适当的 str、\_mbs 或 wcs 函数。如:调用 \_tcsncpy 代替 strcpy 和 wcsncpy,用 \_tcslen 代替 strlen 和 wcslen 等。对于 String conversion 函数,同样在移植过程中用 TCHAR.H 所定义的通用函数。如:用 \_ttoi 代替 atoi 和 wtoi<sup>[5]</sup>。

(3)调整指针算法。

ANSI 类型的 Char \* 的值相减产生一个字节数;Unicode 类型的 wchar\_t 的值相减产生一个 16 位块的个数。在编写 Unicode 应用程序时,当确定字节的个数时,乘以 sizeof(TCHAR)。当由字节数来确定字符个数时,除以 sizeof(TCHAR)。如果在编程时,为了方便也可以为这两种操作定义宏,这样在程序中可以保证 ++ 和 -- 操作将按照数据类型的大小进行增加和减少<sup>[5]</sup>。

例如:获得驱动器时:

```

TCHAR szBuf[100];
memset(szBuf,0,100);
DWORD len = GetLogicalDriveStrings ( sizeof ( szBuf )/sizeof (TCHAR),szBuf);
  
```

(4)检查假设字符长度总是一个字节的代码。

Unicode 的编码方式要求每个字符占用 2 个字节,因此,要检查并修改假设字符值总是小于 256。例如,把字符值作为索引,对大小为 256 项的表进行索引的代码。另外,确保定义的 NULL 的长度为 16 位。

(5)创建通用版本的数据结构。

对于结构中的字符串或字符域的类型定义,应该正确地基于 Unicode 编译时的标志加以处理。在书写串处理函数或字符处理函数,或是书写以字符串为参数的函数,就应该创建这些函数的 Unicode 版本并为它们定义通用的接口或原型。

(6)添加必要的 Unicode 代码进行处理。

大量的应用程序仍然是基于代码页的,而又要实

现内核的 Unicode,所以有必要处理在代码页编码与 Unicode 之间的转换的情况。一对 Win32 API MultiByteToWideChar 和 WideCharToMultiByte 允许在代码页与 Unicode 之间和 Unicode 与代码页之间分别进行转换。例如:

```
char sText[20] = {"栅格数据集"};
MultiByteToWideChar(CP_ACP, 0, psText, -1, sText, dwSize);
wchar_t wText[20] = {L"矢量数据集"};
WideCharToMultiByte(CP_OEMCP, NULL, lpcwszStr, -1, psText, dwNum, NULL, FALSE);
```

## 2.2 移植或修改非 Unicode 应用程序为 Unicode

在平台+插件软件开发模式中,大型的应用程序的开发,还需要除平台与插件外的大量的底层 DLL 或子系统 EXE。而这些应用程序在创建时,也应该按照上述 Unicode 化的要求。但是,在实际情况中,由于历史原因或其他原因,也存在着些非 Unicode 化的应用程序。根据软件国际化的要求,可以按照上述 2.1 的方法,把非 Unicode 的应用程序移植或修改过 Unicode,从而实现应用程序内核国际化。

总之,Unicode 化的操作包括字符的变化,数据类型,数据结构,API 的名字,工程的设置,指针的算法,字节和字符的长度,在 Unicode 与非 Unicode 的 I/O 之间的代码的转换等。Unicode 化程序的高低,它将影响着你的整个应用程序国际化能力的高低。

## 3 移植资源文件实现多国语言界面

在 Win32 应用程序提供多语言用户界面的最佳解决方案是在应用程序内核的基础上为每种语言提供一组资源 DLL 的核心二进制文件。而在基于 Win32 平台+插件模式的应用程序的最佳方案也应采用此种方法。实现这一方案的思路是利用纯资源 DLL,把平台应用程序与插件应用程序的资源文件都分离到纯资源 DLL 中。而这种纯资源 DLL 是仅含资源而没有可执行代码。从而实现了需要本地化的资源与源代码的分离。应用程序中的中文、英文、日文等不同语言的资源可放在不同的纯资源 DLL 中,从而根据不同地区或语言实现软件的本地化<sup>[5~7]</sup>。

### 3.1 建立相应的纯资源 DLL 动态库工程

针对不同地区或语言建立相对应的工程,比如,中文简体相应的工程名为:Chinese;相对应于英文的工程名为:English。

### 3.2 把应用程序中的资源文件都移植于纯资源 DLL 工程中

应用程序包括平台应用程序、插件应用程序、很多

封装有底层不同功能算法的 DLL 应用程序和子系统 EXE 应用程序。资源文件包含菜单、对话框、位图、工具条、消息、光标、字符、音效等资源文件。

(1)处理字符串资源。把用到的字符串,分别放到资源中的 String Table 文件中,分别赋予不同的 ID,而在代码中进行如下处理:

比如:

处理前的代码:

```
CString strInput;
strInput = TEXT("请打开一幅影像!");
```

处理后的代码:

```
CString strInput;
strInput = LoadString(ID_OPEN_IMG_FILE);
```

把处理前的字符串“请打开一幅影像!”放到资源中的 String Table 文件中,并赋予一个唯一的 ID 值 ID\_OPEN\_IMG\_FILE,分离到不同语言版本的 DLL 中,可以根据不同的需求进行翻译。而在处理后应用程序代码中, strInput 变量的值用 LoadString() 函数获得。

(2)防止资源文件中的不同的资源具有相同的 ID,并且在不同的纯资源 DLL 中的要具有相同的 resource.h 文件。在一个大型的平台+插件模式的应用程序中,有大量的插件应用程序,大量的功能 DLL 应用程序和大量的子系统 EXE 应用程序。因此,在整个应用程序中,存在着大量的资源,而这些资源都分离到一个版本的纯资源 DLL 中,所以要避免在 resource.h 有相同的 ID 冲突。另外,无论是哪个语言的版本的纯资源 DLL,在主应用程序中,消息针对 ID,所以,每一个语言版本的纯资源 DLL,要有相同的 resource.h 文件。

(3)避免使用硬编码。所谓硬编码就是把需要本地化的字符串放在程序的主体代码中,而不是放在外部可以易于本地化的资源文件中;基于假设的字符串长度,设定字符串的长度数值常量;在代码中对需要本地化的关于语言或文化方面的任何假设(例如日期和时间格式等);不使用基于源语言的数字常量、屏幕位置、文件和路径名。软件国际化设计应该避免任何硬编码,以提高本地化能力。

### 3.3 翻译、调整资源文件

要根据不同地区、不同语言、不同的风俗习惯进行调整,甚至于更改资源文件。

(1)翻译与中文相对应的文字。

针对不同的语言版本,把其相对应的纯资源动态库中分离出来资源,其中与中文文字相对应的文字翻译成外文,而真正具有其本地化能力。

(2)调整界面的大小。

平台与插件应用程序中,一些界面中的交互文字经过翻译后会扩大很多。即在中文中的交互界面比较合适,而编译后交互文字已经不适应以前的中文界面。比如,中文语言“编辑”翻译成德文“Bearbeiten”,其德文文字长度增加了很多。因此需要调整界面大小,有时甚至要重新设计。

(3)调整界面的格式。

日期格式、日历格式、字符串格式、货币和数字格式、字体、图片标识等,这些格式针对不同的国家与地区,肯定有所不同,所以在进行本地化时,要根据当地的风俗习惯进行相应的修改与调整。

#### 4 程序实例

平台+插件模式的应用程序的实现方式有很多,例如基于 COM 组件开发的方式,基于 DLL 开发的方式等等。无论用什么样的语言与开发工具实现,基本思想都是一样的。文中在实验时,给出了一个在 VC++2005 编译器下用 DLL 实现的平台+插件的简单的应用程序。

在该程序中,采用了三个层次。

(1)最上层:总框架平台程序——EXE,单文档—视图程序。实现图像与图形的显示。

(2)中间层:界面接口模块——DLL 实现,包含图像图形显示视图的基类,与插件 Frame 的基类。

(3)最底层:各种功能的具体实现模块——DLL 实现,一些平级的基本的插件,实例中提供了两个基本的插件,一个是提供操作栅格数据功能,另一个提供矢量数据编辑的功能。

该程序实例中,总框架平台程序,界面接口模块,

栅格操作插件,矢量编辑插件,四个工程,全部实现了 Unicode 化,具备中文软件国际化的能力。而后,又利用纯资源 DLL,分别把这四个工程文件中的资源文件都分离到中文纯资源 DLL 与英文纯资源文件中。最终实现了中文软件的多国语言版本。

#### 5 结束语

平台/插件模式已广泛应用在软件开发,而中文软件国际化是当前中国软件业应对经济全球化竞争的必然趋势。因此,在平台/插件的模式中文软件开发时,应该根据软件国际化的理论及规则进行软件开发,以适应国际市场,提高中文软件的国际竞争能力。

#### 参考文献:

- [1] 吴亮.基于插件技术的 GIS 应用程序框架的研究与实现[J].中国地质大学学报,2006,9(5):609-614.
- [2] 马林,黄文培.Struts 应用程序中文显示方案及其国际化技术[J].计算机技术与发展,2007,17(2):31-34.
- [3] Yeo A. Software Internationalization and Localization[C]//6th Australian Conference on Computer-Human Interaction (OZCHI'96). Washington, DC, USA: IEEE Computer Society, 1996.
- [4] Adams G. Internationalization and character set standards[J]. Standard View, 1993, 1(1): 20-22.
- [5] 张阳,蒋胜平.中文软件的国际化方法研究[J].电脑与信息技术,2006,12(6):44-46.
- [6] Jiang Shengping, Globalization of Non-English Software[J]. International Journal of INFORMATION, 2006, 9(4): 36-41.
- [7] Belge M. The next step in software internationalization[J]. Interactions, 1995, 2(1): 54-62.
- [8] Set and Systems, 1995, 74(2): 213-217.
- [4] Hong D H, Kim C. A note on similarity measures between vague sets and between elements[J]. Information Sciences, 1999, 115(1): 83-96.
- [5] 李凡,徐章艳.Vague 集之间的相似度量[J].软件学报, 2001, 12(6): 922-926.
- [6] 李艳红,迟忠先,严德勤.Vague 集相似度量与 Vague 熵[J].计算机科学, 2002, 29(12): 129-132.
- [7] 夏少云.Vague 集之间相似度量的分析与研究[J].北方交通大学学报, 2004, 28(1): 95-99.
- [8] 张诚一,党平安.关于 vague 集之间的相似度量[J].计算机工程与应用, 2003, 39(7): 92-95.
- [9] Li Zhi-zhen, S Peng-Fei. Similarity measure on intuitionistic fuzzy sets[J]. Pattern Recognition letters, 2003, 24(15): 2687-2693.
- [1] Zadeh L A. Fuzzy sets[J]. Information and control, 1965(8): 338-356.
- [2] Gau W L, Buehrer D J. Vague sets[J]. IEEE Trans SMC, 1993, 23(2): 610-614.
- [3] Chen S M. Similarity measures between Vaguess set[J]. Fuzzy

(上接第 150 页)

相似度量的三个方面的因素,在此基础上,充分考虑了未知信息在相似度量中的作用,解决了传统相似度量方法的中把未知信息看作肯定和否定各占一半的问题,提出了一种新的相似度量方法,并证明其满足公理化定义的所有条件,从而说明了它的正确性,并举出实例论证了它的有效性。

#### 参考文献: