广西科学院学报
Journal of Guangxi Academy of Sciences
2006,22(4):279～283
Vol. 22,No. 4 November 2006

Windows 环境下软件国际化开发的解决方案
A Solution for Developing International Software Based on Windows

苏红帆[1],黄宁宁[2],韦录丰[1]
SU Hong-fan[1],HUANG Ning-ning[2],WEI Lu-feng[1]

(1. 南宁市公安局交通警察支队,广西南宁 530028;2. 南宁市公安局指挥中心,广西南宁 530012)
(1. Traffic Police Office,Nanning Bureau of Public Security, Nanning, Guangxi,530028,China; 2. Command Center,Nanning Bureau of Public Security,Nanning,Guangxi,530012,China)

摘要:介绍软件国际化的概念和主要特征,并分析软件国际化开发过程需要解决的多语种数据存储和多语种数据展示问题,提出一个 Windows 环境下软件国际化开发的整体解决方案。该方案包括指导准则、逻辑框架及其实现技术、数据库数据的国际化等内容,是一个通用方案,对采用任何具体实现技术都适用,可以指导用户开发出具有国际化特征的应用程序。

关键词:软件 开发 国际化 字符编码 多语种用户界面
中图法分类号:TP311 **文献标识码**:A **文章编号**:1002-7378(2006)04-0279-05

Abstract: The concept and main features of the software internationalization are introduced. The storage and revelation of multilingual data in developing international software are analyzed. An integration scheme for developing international software based on Windows is revealed. It includes guideline, logistic framework, implementation techniques, data internationalization of database. It is a universal scheme.

Key words: software, development, internationalization, character encoding, multilingual user interface

在国际化的背景下,为适应国际化业务,企业应用软件也需进行国际化,为不同地区分支机构或用户提供多语种的用户界面。很多开发语言都提供对国际化的支持,为软件国际化开发提供了基础。但是与单语种软件相比,软件国际化开发有其特点和难点,所使用的框架也有所区别。笔者从分析软件国际化开发的关键问题入手,提出一个通用的整体解决方案。

1 软件国际化的概念和主要特征

1.1 软件国际化概念

软件国际化(Internationalization,简称 I18N)就是在软件创建初期,在以完全支持英文的一般编程语言、编译、开发基础上,为适应更广泛的语言和文化习俗,在软件设计结构和机制上支持多语言的扩展特性[1]。软件国际化是不对软件系统使用者作任何假设的开发方法,软件产品的创建或修改,可在多个不同国家或地区和语言间使用。这包括了使用者的语言、文字编码方式、习惯数据表达方式、书写方式、宗教信仰等[2]。

1.2 主要特征

国际化软件和单一语言软件相比,主要具有以下几个主要特点。

1.2.1 全球可用性

全球可用性是软件国际化设计的目标,国际化软件为其不同语言版本提供一致的外观、风格和功能,具有在多种语言环境下运行的能力,能适应不同的国家区域和文化习惯。

1.2.2 本地化能力

本地化能力指国际化软件不用修改源代码即可

收稿日期:2006-07-03
作者简介:苏红帆(1971-),男,广西邕宁人,硕士,主要从事交通管理信息化工作和软件工程、软件国际化开发研究。

本地化为不同语言版本，并且保证本地化后仍具有正确界面和能被正确运行。在软件设计的开始阶段，就要遵循良好的软件本地化能力开发方法，以确保软件产品具有良好的本地化能力。

1.2.3 单一二进制

在软件创建初期，就在软件设计结构和机制上支持多语言的扩展特性，为各种不同语言版本创建单一二进制、全球可用的内核。完全全球化后的功能二进制文件，不用再做修改就可用于该软件的任何一种语言版本[3]。

1.2.4 多语言支持能力

国际化软件应该支持不同语言文字的输入、输出和显示。不仅要具有正确处理复杂文字和混合处理多种语言文字的能力，而且能使不同用户在不同语言版本之间顺畅地共享多种语言数据。

2 软件国际化的关键问题分析

软件国际化开发过程中，需要解决两个关键问题：多语种数据存储和多语种数据展示。计算机系统中，数据存储的基础是字符编码标准，因此多语种数据存储问题的实质就是字符编码问题。多语种数据展示实质上就是多语种用户界面的数据显示问题。

2.1 字符编码问题

字符编码(Character Encoding)是指为字符指定数字值的方法或系统，如ASCII、Unicode和Windows1252等。各国语言文字采用的字符编码标准往往是不一样的，如简体中文的编码标准是GB2312、日文的编码标准是Shift _ JIS等。在系统中以统一的编码方式处理所有字符是从根本上解决编码问题的一种简便而有效的方案[4]。而各国语言文字编码标准往往只支持本国文字和英文，无法实现多种语言文字的混合存储和处理，这给国际化软件开发带来了困难。Unicode编码标准的出现，使得多语言处理成为了现实。Unicode包含了世界上大多数文字的世界范围内的字符编码格式，包含了目前在计算机中广为使用的所有字符，它至少可以处理110万个编码点。现在，Unicode是被所有主要计算机公司接受的、事实上的字符编码标准，而ISO 10646是被所有ISO成员国家认可的、相应的全球性的法定标准，这两个标准包含同样的字符表和二进制表示[3]。目前，主流软件厂商的很多产品和标准都要求或支持、允许使用Unicode，如表1所示。由于支撑软件开发环境的主流产品从操作系统、开发语言到数据库、浏览器等大都支持Unicode。因此，Unicode提供了一种处理多文本和多语言的切实解决方案[3]。

表1 支持Unicode的厂商或产品

类别	支持Unicode的产品
操作系统平台	Mac OS 9.2，IBM AIX、AS/400、OS/2，Microsoft Windows CE，Windows NT、Windows 2000、Windows XP，SCO UnixWare 7.1.0，Sun Solaris及其以后版本
开发语言平台	Ada 95，JavaScript (ECMAScript)，Java，XML，XHTML，XSL，Pert、Sybase、Visual Studio 7.0及其以后版本
数据库平台	IBM DB2 (UDB、AS/400)，Ingres r3，Microsoft Access2000，Microsoft SQL Server7.0，Orcale7.2以后版本
浏览器	IE4.0、NetScape4.0以后版本

2.2 多语种用户界面问题

根据文献[3]可知，多语言用户界面的具体解决方案可能有很多种，但都可以归纳为表2中的3种类型：

表2 多语种用户界面解决方案[3]

方案	说明
1. 语言相关二进制文件	源代码和资源共存一体，每种语言都相应编译一个语言相关的二进制文件。
2. 多语言混合资源文件	资源与源代码分离，创建一个包含所有目标语言的所有本地化资源的文件，将源代码部分编译成语言无关的功能二进制文件。
3. 单语言独立资源文件	资源与源代码分离，为每种目标语言都创建一个独立资源文件，将源代码部分编译成语言无关的功能二进制文件。

方案1缺点是在本地化时需要修改与特定语言的需求和特征相关的源代码、资源。方案2缺点是某个语言的资源发生改变都必须更改/编译整个资源文件，同时也不便于添加新的语言。方案3克服了前两种方案的缺点，添加语言或更改某种语言的资源时，不需修改源代码、也不影响其他语言资源，因此，方案3可以作为应用程序国际化的首选解决方案，例如Windows的所有32位版本就是采用该解决方案。

用户界面的多语种数据包括两类型：静态数据和动态数据。静态数据包括用户界面中的文本数据和程序代码中的文本数据。我们可以采用方案3将静态数据从源代码中分离出来，并保存在资源文件中。动态数据就是程序运行时才产生的文本，我们可以采用数据库来存储。

3 软件国际化开发的解决方案

3.1 指导准则

根据国际化软件特点和上述两个关键问题的分析，并结合国内外软件工程理论和实践，作者对应用

软件国际化开发提出以下 5 点指导准则。其中,Unicode 编码准则适用于字符编码问题,区域中立准则和单一二进制准则适用于多语种用户界面和业务逻辑,文本处理国际化准则和多语言用户界面准则适用于多语种用户界面。

3.1.1 Unicode 编码准则

由于 Unicode 可以解决软件系统的数据国际化问题,国际化软件内部应该统一使用 Unicode 编码格式。在 WEB 系统中应该统一使用 UTF-8 编码(Unicode 编码格式之一)作为 HTTP 请求和响应的字符编码。数据库系统采用与前端应用程序相适应的 Unicode 编码方式,尽量避免前后端数据交互时发生复杂的字符转换。

3.1.2 区域中立准则

因为绝对不能假设全球可用的应用程序将以何种语言向用户显示信息,所有内部数据都应该以某种预定义的、区域中立的二进制或字符串格式存储。显示数据时,不要做区域特定的假设[3]。

3.1.3 单一二进制准则

在应用程序创建伊始,就在软件设计结构和机制上支持多语言的扩展特性。不要在应用程序内部包含本地化的消息资源、非英文字符的常量字符串,而应将它们存放在资源文件中。所有内部数据都应该以某种预定义的、区域中立的二进制或字符串格式存储,为各种不同语言版本创建单一二进制、全球可用的内核。

3.1.4 文本处理国际化准则

以一种一致的方式处理所有支持语言文字,包括文本的输入、输出和显示。确保应用程序具有区域意识和文化意识,使用用户设定的区域属性格式化区域意识和文化意识的数据。应用程序应提供不同的语言和方法来输入文本,能够处理复杂文字,依照与文字关联的语言特性,正确地显示所有支持的文字[3]。

3.1.5 多语言用户界面(MUI)准则

创建易于本地化的多语言用户界面,界面符合所支持的所有区域的本地化文化特征。通过合理的方式管理好应用程序的 MUI,允许用户切换用户界面(UI),使用户可通过简单配置或直接使用菜单、选项等方式选择用户界面的语言。

3.2 逻辑框架及其实现技术

3.2.1 国际化应用软件的逻辑框架

国际化应用软件框架与单语种软件在分层思想上基本一致,差异主要在于各层职责和实现机制上。软件国际化开发在框架上要遵循上节提出的指导准则,确保软件在架构上实现国际化的机制。基于 C/S 和基于 B/S 的国际化应用软件逻辑框架分别如图 1、图 2 所示。

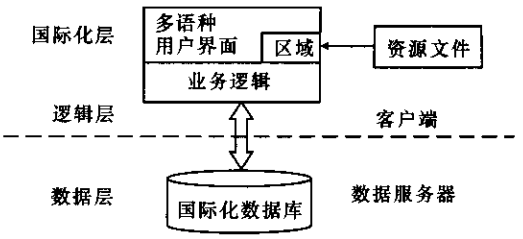

图 1 基于 C/S 的国际化应用软件逻辑框架

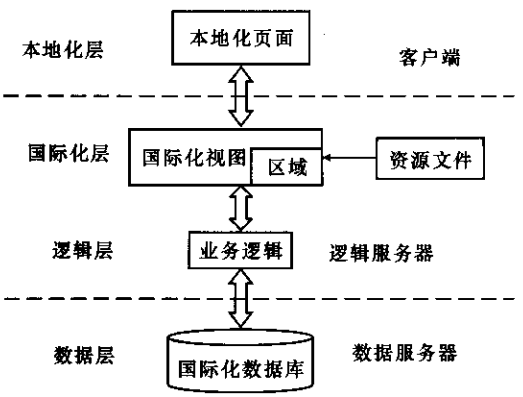

图 2 基于 B/S 的国际化应用软件逻辑框架

两种架构的国际化应用逻辑框架都可以划分为国际化层、逻辑层、数据层。所有层次都应遵循 Unicode 准则,即内部编码采用 Unicode 标准,B/S 架构推荐使用 UTF-8 编码。

3.2.1.1 国际化层

国际化层有 3 个部分:用户界面、区域对象和资源文件。

用户界面:C/S 架构称为多语种用户界面,B/S 架构称为国际化视图。两者区别主要在于本地化上不同,在运行时 C/S 架构的 MUI 直接在 UI 上本地化,而 B/S 架构的国际化视图则是先转化为本地化 HTML,然后发送到客户端的浏览器中显示本地化页面。遵循文本处理国际化准则,用户界面以一种一致的方式处理所有支持语言文字,包括文本的输入、输出和显示,并且要确保用户界面具备区域意识和接受处理复杂文字的能力。

区域对象:根据区域中立准则,区域对象在设计时不能以任何语言区域作为假设,而是设计成动态的。根据 MUI 准则允许用户通过改变区域对象属性来切换用户 UI。

资源文件:根据单一二进制准则,任何静态的本地化文本都必须从程序源代码中分离出来保存在独

立的资源文件中，资源文件根据语种分别设立。

3.2.1.2 逻辑层

根据单一二进制准则，所有内部数据都应该以某种预定义的、区域中立的二进制或字符串格式存储，不要在应用程序内部包含本地化的消息资源、非英文字符的常量字符串，而应将它们存放在资源文件中。

3.2.1.3 数据层

遵循 Unicode 准则，应选择支持 Unicode 标准的数据库系统，并选择与前端编码格式一致的字符集，避免前后端进行复杂的数据转化。数据库的设计详见 3.3 部分。

3.2.2 软件国际化开发的实现技术

很多开发语言都提供对国际化的支持，当前最主流的开发平台 Java 和. NET Framework 都提供了良好的国际化机制。这两种平台在解决多语种方面是相类似的，都是先对资源进行封装，然后根据区域属性格式化数据并在用户界面加以显示。这两种平台都可以很好地实现 3.2.1 中提出的逻辑框架。Java 具有良好的跨平台性、移植性和开放性，选择 Java 平台可使应用软件具有良好的异平台移植性。. Net Framework 集成度高，集成多种开发语言，便于应用软件的开发，但局限于 Windows 平台。Java 和. Net Framework 各具优势，选择何种开发平台视具体情况而定，也可选择其他支持国际化的开发语言。限于篇幅，以下只简单介绍 Java 和. Net Framework。

为实现国际化和本地化，Java 提供了 JIL(Java Internationalization and Localization)类库。应用程序启动时，JIL 代码从操作系统获取或由开发人员指定区域变量，然后再从资源包中获取特定区域的资源，最终将其显示在应用程序中[5]。Java 的内置国际化支持以 3 个主类为中心，如表 3 所示。

表 3 Java 实现国际化的三个主类

类	说明
java. util. Locale	封装一种特定语言环境所对应的语言、国家(地区)以及变体。
java. util. ResourceBundle	封装特定于语言环境的资源。资源文件扩展名为. properties。
java. text. MessageForamt	提供创建格式化的特定语言环境的消息的方法

. NET Framework 实现国际化的机制为：由 CultureInfo 类提供的区域性属性设置；然后 ResourceManager 类根据区域性检索特定语言的资源；最后使用 System. Globalization 命名空间提供的区域性识别类来操作和格式化数据。. NET Framework 实现国际化的命名空间如表 4 所示。

表 4 . NET Framework 实现国际化的命名空间

命名空间	说明
System. Globalization	包含定义与区域性相关信息的类，其中包括语言、国家/地区、所使用的日历、日期格式的模式、货币与数字以及字符串的排序顺序。
System. Resources	提供一些类和接口，它们使开发人员得以创建、存储并管理应用程序中使用各种区域性特定的资源。资源文件扩展名为. resources。
System. Text	包含表示 ASCII、Unicode、UTF-7 和 UTF-8 字符编码的类。

3.3 数据库数据的国际化

数据库数据的国际化包括 3 个主要方面的问题：字符编码和字符集问题、存储空间问题、国际化数据表设计方案。

3.3.1 字符编码和字符集问题

Microsoft SQL Server 和 Oracle 都支持 Unicode 编码。在 Microsoft SQL Server7 以后版本定义了 3 种 Unicode 文本类型：nchar、nvarchar、ntext，这些数据类型被定义为和 USC-2(UTF-16)等价[3]。通过全球支持，Oracle 可以支持多种语言以及字符集，从 7.2 版本开始支持 UTF-8 编码，提供对 Unicode 编码的支持[6]。字符集在创建数据库时指定，在创建后通常不能更改，所以在创建数据库时能否选择一个正确的字符集就显得尤为重要。根据 Unicode 准则，前后端的字符编码要相适应，避免前后端数据的字符转换。例如，Oracle9.2 中支持 Unicode 的字符集有 AL16UTF16、AL32UTF8、UTF8、UTFE、AL24UTFFSS，假设前端是 Web 程序，由于 Web 程序默认编码是 UTF-8，则应在数据库中选择 AL32UTF8 字符集，这样前后端之间传递数据就不用任何字符转换。

3.3.2 存储空间问题

在 Microsoft SQL Server 中，Unicode 数据类型所需的实际空间量是每个字符 2 个字节，是固定长度的。在 Orcale 中，UTF8 字符集并不是以固定长度进行编码的，例如 AL32UTF8 根据 Unicode 编码点采用 1～4 个字符进行编码。因此，在数据表结构设计时，必须认真考虑数据存储空间问题，即该如何设定字段长度的问题。根据文献[3]，Unicode 编码点和字节数以及编码的文字可归纳如表 5 所示。我们可以根据表中对应的 UTF-8 编码字节数来计算字

段所需的存储空间。例如，ASCII 字符的 UTF-8 编码是 1 字节/字符，阿拉伯语是 2 字节/字符，而中文则是 3 字节/字符。

表 5 Unicode 编码点分配简表

Unicode 范围	UTF-8 编码的字节	文字
0×00000000—0×0000007F	0×××××××	基本拉丁语。
0×00000080—0×000007FF	110××××× 10××××××	拉丁语扩展、希腊语、希伯来语、阿拉伯语、叙利亚语等。
0×00000800—0×0000FFFF	1110×××× 10×××××× 10××××××	包括汉字在内的世界上大部分语言文字、字符等。
0×00010000—0×00200000	11110××× 10×××××× 10×××××× 10××××××	古意大利语等古语，以及一些扩充、增补字符。

表 6 国际化数据表设计方案比较

方案	说明	优缺点
冗余数据方式	在数据库表中引入标识语种的 Locale 字段，每一条原始数据记录在国际化的数据表中都对应有多条各种语言的记录。	解决了国际化数据的存储和访问的问题，但是非国际化字段会有冗余数据，数据更新将涉及所有语种的记录以及维护数据的一致性，从而增加了系统运行时的开销。
单表方式	在数据库中用专门的数据表维护国际化数据，记为 LocaleTable。其他数据表的国际化字段存储 LocaleTable 表中记录的主键值，通过主键值与 LocaleTable 关联。	将国际化数据和非国际化数据分离开来，减少冗余数据，提高了数据一致性。不过，用单表存储会造成单表数据过多，从而增加对该表的访问时间，降低系统的执行效率。
分表方式	将国际化字段和非国际化字段分离成国际化表和非国际化表，两表均使用原始表主键，其中国际化表引入标识语种的 Locale 字段，并与原始表主键组合成复合主键。	解决了数据冗余和一致性的问题，但是由于分表后对原始表的访问就转变为两个分表的连接操作，也可能降低执行效率。

3.3.3 国际化数据表设计方案

支持国际化的数据库要满足 3 个要求：数据的一致性、国际化数据的完备性以及尽量减少数据冗余[4]。根据文献[4]可知，支持国际化的数据库的数据表的设计有 3 种方案，3 种方案各有优缺点，如表 6 所示，具体使用何种方案要视具体情况而定。

4 结束语

本文对软件国际化开发提出了一个通用解决方案。虽然在实际开发时，我们还需将逻辑框架细化为具体的实现框架，实现方法还需具体化，但是本文提出的指导准则、逻辑框架、数据库数据国际化等方针都是适用的，可以对整个国际化开发工作起到指导作用，从而确保应用程序具有良好的国际化特征。

参考文献：

[1] 曾凡锋. Linux 环境下应用软件国际化的实现[J]. 北方工业大学学报，2005，17(3)：30.

[2] 于千里. . NET 平台的软件国际化解决方案研究[J]. 农业网络信息，2005，6：14.

[3] International. 软件国际化开发[M]. 沈凤，译. 北京：机械工业出版社，2003：5，27，37，40，62，116，151-155，322.

[4] 许晖，李涓子. J2EE 系统国际化问题的解决方案[J]. 计算机工程，2005，31(18)：79-80.

[5] 李华宇. Java 的国际化和本地化原理及解决方法[J]. 微型机与应用，2001，11：32-33.

[6] 冯春培，盖国强，冯大辉，等. Oracle 数据库 DBA 专题技术精粹[M]. 北京：冶金工业出版社，2004：75.

（责任编辑：韦廷宗）

蔬菜农药残留监测信息管理系统通过技术鉴定

受广西科学技术厅的委托，南宁市科学技术局于 2006 年 5 月 17 日在南宁市组织召开“蔬菜农药残留监测信息管理系统”项目技术鉴定会。蔬菜农药残留监测信息管理系统包括检测结果、走势图、结果分析图、报表、宾馆检测和后台管理等六大模块，系统运行稳定、数据准确，具有较好的通用性、实用性、可靠性和易用性。该系统利用计算机、网络、通讯等现代高新技术，结合农产品检测的特点，实现对南宁市农产品农药残留检测信息资源的处理和共享，使信息的传递更加快捷和方便，系统能实时生成准确的检测结果走势图、分析图和报表，能实现蔬菜市场、蔬菜产地、农药检测超标蔬菜处理情况等信息的维护，能实现蔬菜市场、基地、检测员的考勤考核管理，提高了管理水平和工作效率，达到国内同类系统先进水平。该系统于 2004 年 7 月投入使用至今，社会效益和经济效益良好。

（陈友初）