《英汉蒙电子词典》的设计与实现

吴红英，嘎日迪，赵小兵，韩东妹

（内蒙古师范大学计算机与信息工程学院，内蒙古 呼和浩特 010022）

摘　要： 本文介绍了《英汉蒙电子词典》的设计与实现方法，并给出相关的算法流程图。其中详细分析了屏幕取词的关键技术——截获 API 函数的调用、鼠标钩子和判断鼠标所停留位置的单词。同时，给出实现蒙古文字符的竖排显示方法。

关键词： 蒙古文；屏幕取词；应用程序编程接口；钩子

Design and Realization of "English-Chinese-Mongolian Electronic Dictionary"

Wu hong-ying, Garidi, Zhao xiao-bing,Han dong-mei

(College of Computer and Information Engineering, Inner Mongolia Normal University, Huhhot 010022)

Abstract: This paper introduces the design and realization method of "English-Chinese-Mongolian Electronic Dictionary", and give arithmetic flow chart of the realization. This paper analyzes the key technologies of capturing words from screen——API Hook, Mouse Hook and fixing the words on cursor. And expresses how to implement Mongolian characters vertical display.

Keywords: Mongolian; Capturing Words from Screen; API; Hook

1　引言

我国是一个多民族国家，蒙古族是民族大家庭当中的一员。全国约有 480 多万蒙古族，有八省区在使用蒙古文。在内蒙古自治区，蒙古文作为主体的官方文字，在社会各个领域中是必不可少的。例如在文化、教育、科研、宣传出版和媒体等许多行业中都在使用蒙古文。目前，内蒙古自治区使用蒙古语言文字教学的中小学有 1826 所，在校学生约 40 万人，他们不但需要学习蒙古语也需要学习汉语和英语。《英汉蒙电子词典》作为一种辅助教学软件，它的成功开发对蒙古族学生学习汉语和英语有一定的帮助。

2　蒙古文的特点及系统解决的关键问题

我们的研究目标是在 Windows 环境下实现具有英语词汇译为汉语和蒙古语以及汉语词汇译为英语和蒙古语翻译功能的词典，这里的蒙古文指的是传统蒙古文。词典包括屏幕取词、即时互译、词汇查询、词库维护等词典专有功能。

基金资助：内蒙古自治区教育厅科研基金项目（NJ3037）

作者简介：吴红英（1982-），女（蒙古族），内蒙古通辽市人，内蒙古师范大学在读硕士.

蒙古语言是阿尔泰语系的蒙古族语言文字。蒙古语是一种特殊的语言文字，是借用回鹘文字母，经过几百年的演变而形成的文字。蒙古语字母在组成单词时，是无缝隙连接显示；在排版时，是从上至下连接成词、从左向右换行的纵向排版显示方式。这些都要求程序给予特殊处理。尽管蒙古文在国际大家庭中也占有一席之地，有一定的知名度和影响力，但由于使用人口相对较少、书写格式比较特殊、进入国际标准编码又相对较晚，很多软件还无法顾及蒙古文的特殊使用要求，目前还不提供此类技术的支持。因计算机操作系统均不提供对这种特殊文字的底层函数支持，这就为蒙古语言文字的软件开发者提出了难题。目前的翻译词典尽管很多、功能也很强，但没有提供针对蒙古文的特殊处理。带有蒙古文(传统蒙古文)的词典很少，但是已出现了几部。在英汉蒙三语词典方面，都处于进一步完善和维护阶段。

《英汉蒙电子词典》需要解决的关键问题包括：(1)设计英-汉-蒙词库和汉-英-蒙词库。(2)设计词库查询功能，允许用户按英文和汉文关键字进行词库查询。(3)设计词库维护功能，允许用户在使用中随时修改、删除和添加词库中的单词。(4)设计符合蒙古文显示的功能模块。(5)设计屏幕取词功能模块，满足即点即译的功能要求。包括：截取 API 入口，获得 API 的参数；安全地潜入 Windows 内部；获取鼠标所在位置下的词单。

3 系统的设计

3.1 系统功能模块图

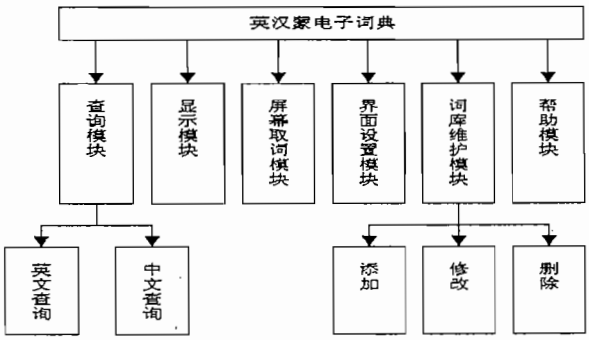


图 1《英汉蒙电子词典》功能模块图

3.2 词库设计

通常，电子词典是双语的，而《英汉蒙电子词典》是拥有三种语言的词典。考虑到如果只设计一个英-汉-蒙词库，对于中文的查询速度将会降低，因此要设计英-汉-蒙词库和汉-英-蒙词库各一套。设计思想：首先要录入英-汉-蒙词条，以英文为关键字形成英-汉-蒙词库；在此库的基础上建立汉-英-蒙词库，以汉语为关键字进行存储。

3.3 查询功能的设计

《英汉蒙电子词典》中查询功能在两处应用。一是在主界面，通过输入进行查询，另一个是在鼠标取到词之后的查询功能的实现。英文查询是对英-汉-蒙词库的精确查询。通过输入进行中文的查询同样是精确查询，如果汉-英-蒙词库中没有要查询的中文，将提示用户是否添加该中文。鼠标取词得到的中文并不是正确的中文词汇，通过查询汉-英-蒙词库，使用正向最大匹配方法才能最终确定正确的中文词汇。

3.4 屏幕取词技术

Windows 应用程序是通过应用程序编程接口(API)调用系统功能的，字符的显示也不例外[1]。经过分析知道：Windows 通过 ExtTextOutA、ExtTextOutW、TextOutA、TextOutW、DrawOutA、DrawOutW 这几个 API 函数显示大部分文字，其中以 A 结尾的是显示编码为 ASCII 码的字符的函数，以 W 结尾的是显示编码为 Unicode 的字

符的函数[2][3]。而且这几个函数都是在 Win32 子系统 GDI32.dll 中实现的，如果能想办法截获这几个函数的调用，加上适当的控制就可以实现屏幕取词。换一句话说，对于实现屏幕取词的关键问题是如何截获对这几个函数的调用。这技术看似简单，但其实在 Windows 系统中实现却是复杂困难的。

3.5 维护功能的设计

维护功能包括修改、添加和删除等功能。由于设计了两套词库，英-汉-蒙词库和汉-英-蒙词库，而且两套词库是保持一致的。因此，做维护功能的时候要考虑同时完成两个词库的维护，以便保证词库的一致性。

4 关键技术的实现

4.1 屏幕取词技术的实现

屏幕取词就是用鼠标取得屏幕上的文字，“金山词霸” 、“四通利方”和“南极星”等电子词典都使用了这个技术。目前，许多人对屏幕取词技术都有着浓厚的兴趣，网络上对此技术的讨论也很热。不仅是因为这项技术有着很重要的实用价值，而且还因为该技术涉及了许多的 Windows 内部知识，对进一步开发 Windows 程序有着很大的引导作用。

4.1.1 hook 的原理

Hook(亦称钩子)是 Windows 中经常使用的技术之一，它利用 Windows 中消息机制的处理特点，监视 Windows 系统中的消息的传递，并在消息到达目的地之前，将其截获并根据用户要求做出相应处理[4]。在屏幕取词时将用到鼠标钩子（WH_MOUSE Hook)。当用户在屏幕上移动鼠标时，系统就会调用鼠标 filter 函数 FilterMouseProc；进入 FilterMouseProc 后，就可以获得鼠标的当前的坐标了[5]。

4.1.2 屏幕取词原理

屏幕取词可分为两种实现方式：陷阱式和改引入表式[3]。在该电子词典中将使用第一种方式。程序的基本流程如下：a)判断鼠标是否在一个地方停留了一段时间。b)取得鼠标当前位置。c)以鼠标位置为中心生成一个矩形。d)挂上 API 钩子。e)让这个矩形产生重绘消息。f)在钩子里等输出字符。g)计算鼠标在哪个单词上面，把这个单词保存下来。h)如果得到单词则摘掉 API 钩子，在一段时间后，无论是否得到单词都摘掉 API 钩子。i)用单词查询词库，显示解释框[6]。

下面对上述一些关键步骤进行进一步的解释：

取得鼠标当前位置：只要装入一个 WH_MOUSE 类型的系统钩子，就可以截获所有的鼠标消息了[3]。应用程序可以调用 SetWindowsHookEx 来挂上鼠标钩子。安装了钩子函数之后，系统发给每个应用程序队列的消息都可以用鼠标钩子函数截获，我们就有机会在鼠标钩子内部截获各种鼠标消息，当然包括鼠标当前位置。具体实现的方法在 Hook 原理部分已给出。

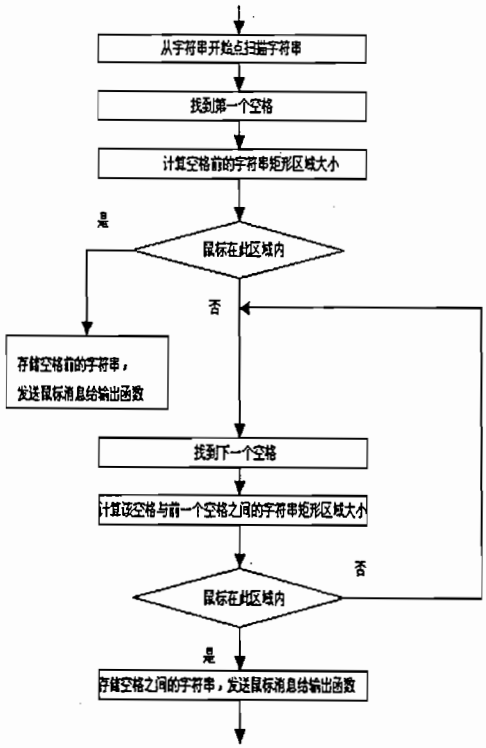
挂 API 钩子：通过直接改写 API 在内存中的映像，嵌入汇编代码，使之被调用时跳转到指定的地址运行来截获[1]。通俗地讲，就是在 Windows API 入口写一个 JMP XXXX 语句，跳转到自己的代码里。这就是陷阱式 API Hook。我们的目标是截获 GDI32.dll 中的六个 API 显示函数 ExtTextOutA、ExtTextOutW、TextOutA、TextOutW、DrawOutA、DrawOutW。陷阱式截取 API 就是在系统调用 API 显示函数（以 ExtTextOutW 为例）的时候，把这个函数开始执行处的几个字节改为 JMP XXXX，转到自己设计的陷阱函数中，这个函数的入口参数的类型、次序和返回值类型要和 ExtTextOutW 一样，在这个自己编写的函数中就可以分析显示字符串，做相应的处理了。方法是通过 VirtualProtect 修改进程的用户空间内存的读写保护属性[1][7]，例如把 ExtTextOutW 这个函数开始执行处的几个字节区域的保护属性改为可以读写（Page_ReadWrite)，接下来用 WriteprocessMemory 把它的第一条指令改为 JMP XXXX，跳转到自己设计的用来代替 ExtTextOutW 的函数入口处，这样在系统调用 ExtTextOutW 的时

候就可以跳转到自己写的函数中。

API 显示文字函数位于 GDI32.dll 中，它位于进程 2G 以上地内存空间，这个 dll 模块同时被多个应用程序使用，也就是在 Win32 系统中多个应用程序共享内存中的一个 dll 备份，但是 Windows NT/2000 有个 Copy On Write（写时备份）的机制，一个应用程序在此空间（2G 以上空间）中写的数据并不影响到其他进程，操作系统自动申请一块内存（该地址与原地址相同）让应用程序写[7]。也就是说上面的做法只能截获本进程的 API 函数。如果程序想截获其他进程中的 API 调用，就必须打破进程边界，把上面的代码注入到其他进程中，这个任务交给了挂钩函数 SetWindowsHookEx 来完成。由于鼠标钩子通过 SetWindowsHookEx 已经完成了注入其他进程的工作，所以不需要为注入而编写额外的代码，可以利用钩子函数注入、退出其他进程时，分别执行 Initialization、finalization 里的代码，进而把注入和退出代码写在 Initialization、finalization 里面就可以了[3]。可以看到，在屏幕取词过程中鼠标钩子充当了两个角色：代码注入与全局鼠标事件的捕获[2]。

向鼠标所在的窗口发重绘消息，让系统自动更新显示文字：重绘的消息只对窗口客户区（Client）有效，对非客户区（如标题、菜单等）无效。因此，还必须使用 Windows API（如 SetWindowsText 等）让其重绘[3]。

计算鼠标在哪个单词上面：通过调用自己编写的函数 NewTextOutA、NewTextOutW、NewExtTextOutA、NewExtTextOutW、NewDrawTextA、NewDrawTextW 输出的字符串还不是鼠标指向的英文单词或者中文词，还要经过判断鼠标位置进一步处理得到的字符串，来获得鼠标位置下的单词。在 Windows 的字符串显示函数中，以 A 结尾的是显示编码为 ASCII 码的字符的函数，以 W 结尾的是显示编码为 Unicode 的字符的函数。以 A 结尾的函数取出的字符串是 ASCII 码字符串，有相应的字符处理函数。但是对于以 W 结尾的函数取出的字符串除了上面的流程外，还要做一个特殊的处理，使用函数 widechartostring 将双字节编码的字符串转换为单字节编码的字符串，再做相应的处理就可以了。英文字母是单字节编码，及以 ASCII 码存储，是通过 ExtTextOutA、TextOutA、DrawOutA 等 API 函数显示的。仿照 ExtTextOutA、TextOutA、DrawOutA 的参数及调用协定来编写自定义函数。由于英文单词与单词之间是以空格分开的，所以用查找空格的方法，确定鼠标位置下的单词，如果有多个空格可以做循环处理。中文字是双字节 Unicode 编码，是通过 ExtTextOutW、TextOutW、DrawOutW 等 API 函数显示的。仿照 ExtTextOutW、TextOutW、DrawOutW 的参数及调用协定来编写自定义函数。由于中文的字词之间是连写，所以无法用空格、句号、逗号等一些特殊标志来切分。因此，对于中文的字、词的查询成为程序的难点。图 2 和图 3 分别为定位鼠标位置下英文单词的算法流程图和定位鼠标位置下中文字的算法流程图。



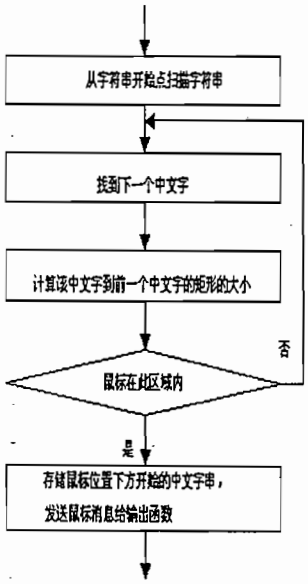


图 2 定位鼠标位置下英文单词的算法流程图　　图 3 定位鼠标位置下中文字的算法流程图

4.2 蒙古文字符的竖排显示实现

由于 Windows 系统不支持蒙古文字符的从上至下、从左至右的纵向显示格式，所以软件必须在文字显示时做加工处理。为了解决显示问题需要创建一个 LogFont 类型的逻辑字体，可以通过调用 Win32 API 函数 CreateFontIndirectt 来创建。Win32 API 即为 Windows 应用程序编程接口（Application Programming Interface），所有在 Win32 平台上运行的应用程序都可以调用这些函数。这个 API 函数在 Delphi 中的声明为 function CreateFontIndirect(const p1: TLogFont): HFONT; stdcall;该函数只有一个参数 p1: Tlogfont，所有有关字体的参数完全通过这个 TLogfont 结构来传送，Windows 将根据结构中的内容创建出相应的逻辑字体，在 Delphi 的 Windows.pas 中有 TlogFont 的定义。其中涉及到很多参数，对于完成蒙古文的竖排显示有两个重要的参数分别是 lfEscapement 和 lfFaceName。对前者赋值为-900 即可完成字型顺时针旋转 90 度。后者是指定采用的字体名称的参数，对其赋值为 MONGOLFD(蒙古文字体)。

逻辑字体创建之后，就可以把蒙古文字符串用当前逻辑字体写到指定位置了。使用的函数为：function TextOut(DC: HDC; X, Y: Integer; Str: PChar; Count: Integer): BOOL; stdcall;。

5 测试和总结

本词典的使用对象是中、小学学生，因此我们借助《英汉蒙学生实用词典》建立了英-汉-蒙词库，现已录入 6000 多条单词。词典设计实现后能够完成英语到汉语和蒙古语的翻译以及汉语到英语和蒙古语的翻译。其屏幕取词功能在 Windows 环境下的 Word 字处理软件、Excel 表格处理、记事本、桌面上进行了测试后能够将鼠标位置下的信息即时翻译成蒙古语。对汉语的取词翻译取决于词库中的词汇量，词汇量越大取词翻译准确率越高。

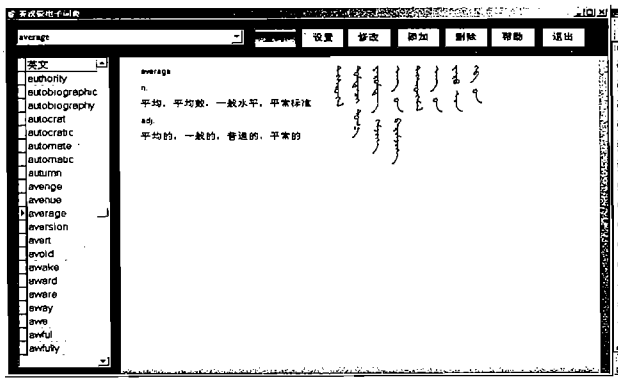


图 4 主界面

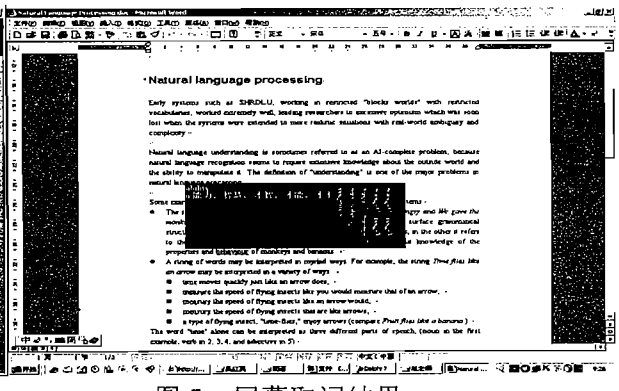


图 5 屏幕取词结果

本词典与其它成熟的电子词典相比还有相当大的差距，词典需要进一步完善，如在查询速度、IE 上鼠标定位准确等等。本词典的研发将对蒙古文自动识别以及英蒙机器翻译系统的成功开发奠定基础。

参考文献：

[1]顾平，刁红军.屏幕取词原理与实现[J].计算机工程与应用，2004 年第 28 期：109 页.

[2]王歆，张林山.windows 环境下的屏幕取词技术[J].实践经验，2000 年 4 月：62 页.

[3]飞思科技产品研发中心.Delphi 下深入 Windows 核心编程[M].北京：电子工业出版社，2003 年 1 月.

[4]杨宏宇.Hook 类型选型[J].计算机应用，2003 年 5 月，第 23 卷第 5 期：118 页.

[5]丘建雄，蔡放，方逵.Hook 技术及其在软件研发中的应用[J].计算机应用与软件，2003 年 02 期：7 页.

[6]吾守尔·斯拉木，热衣曼·吐尔逊.维吾尔文屏幕取词关键技术及其实现方法[J].新疆大学学报（自然科学版），2005 年 8 月，第 22 卷第 3 期：334 页.

[7]王社伟，朱如鹏.Windows 操作系统中的 GDI 坐标系统[J].计算基于现代化，2003 年第 4 期：14 页.