

# Elastos 智能手机动态多 UI 电话簿排序的实现

姜默涵, 陈 榕, 黄玉坤

(同济大学 基础软件工程中心, 上海 201203)

**摘 要:** Elastos 手机是一款可用于 TD-SCDMA 通信网络的智能手机, 它采用 Elastos 嵌入式操作系统, 运行于 Elastos 构件运行平台之上, 并使用 SQLite 嵌入式数据库作为公用存储。讨论了为 Elastos 智能手机提供动态多 UI 电话簿排序的数据处理技术, 介绍了在 SQLite 的 CAR 构件封装与移植的基础上, 如何实现各种中文排序以支持电话簿的排序功能。

**关键词:** Elastos 智能手机; 电话簿; 人机交互界面; SQLite; 中文排序

**中图分类号:** TP311.5

**文献标识码:** A

**文章编号:** 1673-629X(2008)01-0196-03

## Implementation of Dynamic UI Phone Book Sorting on Elastos Smart Phone

JIANG Mo-han, CHEN Rong, HUANG Yu-kun

(System Software Engineering Centre of Tongji University, Shanghai 201203, China)

**Abstract:** Elastos mobile is a smart phone that can be used on TD-SCDMA network. It adopts Elastos embedded operating system and runs on CAR component platform. It uses SQLite embedded database as public storage. Discusses data processing technology of the sorting of phone book on Elastos smart phone and implements Chinese sorting to support the sorting function of phone book after transplanting SQLite to Elastos.

**Key words:** Elastos smart phone; phone book; UI; SQLite; Chinese sorting

### 0 引 言

智能手机(Smart Phone)是集语音通信、数据通信、图像处理等多种功能于一体的个人移动媒体平台。智能手机在继承传统功能手机(feature phone)的一般语音通信、短消息、WAP等基础功能的同时,还发展了很多过去个人电脑才具备的处理能力和功能。例如强大的数据处理能力、支持动态加(卸)载应用程序、支持一定程度的软件功能扩展等。

作为一款支持中文的智能手机,对中文排序功能的支持十分重要,它牵涉到通话、短信、彩信、记事本等多个业务,它的功能强大、界面友好与否直接影响到终端用户的体验和该款智能手机的市场竞争力。

Elastos 手机是配合 TD-SCDMA 第三代无线网络而研发一款智能手机,是 TD-SCDMA 无线数据业务系统的重要组成部分。以 Elastos 手机中的电话簿为例,它的功能除了体现在其条目的数量和信息

丰富度外,还体现在其条目的排列、检索的方便、高效性和多策略性上。

文中讨论了如何在 Elastos 智能手机上,基于 SQLite 嵌入式数据库实现动态多 UI 电话簿排序核心模块的设计,以及 SQLite 的 CAR 构件封装和在 Elastos 操作系统上的移植。

### 1 Elastos 操作系统与 CAR 构件技术

Elastos 是上海科泰世纪公司开发的一款运营网络增值业务的嵌入式网络构件运行平台。它同时也是一个通用嵌入式操作系统,支持多进程、多线程、抢占式、线程多优先级任务调度等,其操作系统的内核叫 Zener,具有体积小、速度快、效率高的特点,适合于智能手机等消费类电子的嵌入式应用。Elastos 提供关于 CAR 编程模型和 CAR 构件运行的最优化支持。

CAR(Component Assembly Runtime)编程模型是科泰世纪自主设计、实现的一套网络构件编程规范。CAR 技术<sup>[1]</sup>的主要特色包括:支持可执行目标代码的自描述封装(Reflection);面向构件和面向方面编程(AOP);脚本语言自动化拼装网络构件。这使得在 Elastos 系统上实现的嵌入式网络应用软件具有可动态

收稿日期:2007-03-07

基金项目:国家“863”计划资助项目(2001AA113400)

作者简介:姜默涵(1984-),女,辽宁人,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,博士生导师,教授,研究方向为嵌入式系统、构件技术。

升级、通过配置实施新电信业务等优势,为实现动态多 UI 电话簿排序功能提供了有力的技术支持。

## 2 SQLite 的 CAR 构件封装与移植

SQLite 是用 C 语言编写的开源嵌入式数据库引擎<sup>[2]</sup>。它支持 SQL92 标准,可以在所有主要的操作系统上运行,并且支持大多数计算机语言。它可以直接在应用程序进程中运行,消除了与客户机服务器配置相关的开销,并提供零配置运行模式<sup>[3]</sup>,因此特别适合 Elastos 智能手机这样的嵌入式应用环境。

选定 SQLite 作为 Elastos 手机的持久存储服务,集中存储手机中需要动态共享、网络共享管理的各类信息。由于 SQLite 只支持 Windows 和 Linux,所以在使用它前首先需要对它进行移植。

首先开发了 SQLite 中与操作系统相关的代码,然后对 SQLite 做 CAR 构件封装。CAR 的描述代码(部分)如下:

```
module http://www.koretide.com.cn/sqlite3.dll
```

```
{
    interface IDatabaseManager;
    interface IDatabase;
    interface IStatement;
    interface IPreparedStatement;
    interface IResultSet;
    interface IResultSetMetaData;
    //其它接口定义信息等.....
}
```

最后形成的接口引用关系如图 1 所示。

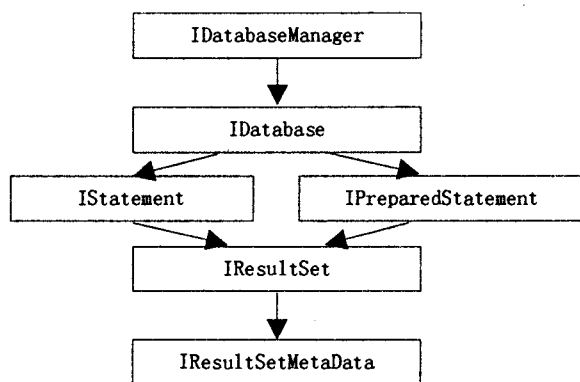


图 1 SQLite 持续存储构件接口关系图

## 3 动态多 UI 电话簿排序实现

### 3.1 电话簿排序的多个动态 UI 设计

图 2 所示的软件界面是 Elastos 手机向用户提供多种中文电话簿排序(如按音序、笔划数、常用优先等)方式的动态可替换 UI 之一。

通过触摸屏或软件可以在多个排序方法间切换。

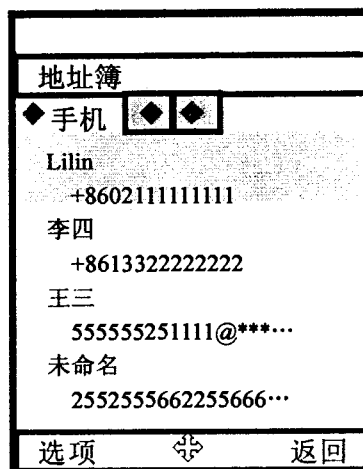


图 2 和欣手机 UI 示例

### 3.2 定义并安装比较函数

在 Elastos 手机的应用开发过程中,发现 SQLite 并没有提供对中文排序的支持,默认情况下,它对中文排序是按照汉字内码值的大小进行的,而汉字内码既不是完全按照拼音排序,也不是完全按照笔画排序。

幸运的是,SQLite 提供对用户自定义比较函数的支持,可在 SQLite 的官方网站上获得相关信息<sup>[4]</sup>。使用哪一个比较函数是由 COLLATE 子句决定的,该子句可用在表定义、索引或 SELECT 语句的 ORDER BY 子句中。在不指定 COLLATE 子句的情况下,即按内码排序。

查看源代码,在 SQLite 控制台程序 sqlite.exe 的 main.c 中,提供了两个用于创建比较函数的接口<sup>[5]</sup>:

```
int sqlite3_create_collation(
    sqlite3 * db,
    const char * zName,
    int enc,
    void * pCtx,
    int (* xCompare)(void *, int, const void *, int, const void *)
)

int sqlite3_create_collation16(
    sqlite3 * db,
    const char * zName,
    int enc,
    void * pCtx,
    int (* xCompare)(void *, int, const void *, int, const void *)
)
```

其中前者用于安装 UTF8 的比较函数,后者用于安装 UTF16 的比较函数。

在 main.c 里已经安装了一些内置的比较函数:

```
sqlite3_create_collation(db, "BINARY",
    SQLITE_UTF8, 0, binCollFunc)
sqlite3_create_collation(db, "BINARY",
    SQLITE_UTF16, 0, binCollFunc)
```

```
sqlite3_create_collation(db, "NOCASE",
```

```
SQLITE_UTF8, 0, nocaseCollatingFunc)
```

因此,可以参考系统内建的两个比较函数 binCollFunc 和 nocaseCollatingFunc 的实现,如下定义一个按照拼音排序的比较函数:

```
int pinyin_cmp(void *NotUsed, int nKey1, const void *pKey1,
int nKey2, const void *pKey2)
{
    int n = nKey1 < nKey2 ? nKey1 : nKey2;
    return pinyin_strcmp(pKey1, pKey2, n + 1);
}
```

并将其安装到系统中:

```
sqlite3_create_collation(db, "pinyin", SQLITE_UTF8, 0, pinyin_
cmp);
```

### 3.3 比较函数的具体实现

pinyin\_strcmp 函数的实现必须满足:如果第一个字符串小于第二个字符串,该函数返回一个负整数;如果第一个字符串等于第二个字符串,该函数返回 0;如果第一个字符串大于第二个字符串,该函数返回一个正整数。每个汉字的编码为两个字节(Elastos 的汉字内码统一为 wchar\_t,为一无符号 short 类型),我们的目的是要找出第一个不同的两个字节进行比较。

该函数的实现如下:

```
int pinyin_strcmp(const unsigned short *str1, const unsigned
short *str2, unsigned long length)
{
    unsigned long i = 0;
    if(str1 == NULL) return -1;
    if(str2 == NULL) return 1;
    for(i = 0; i < length; i++)
        if(!str1[i] || !str2[i] || str1[i] != str2[i]) break;
    return pinyin_charcmp(str1[i], str2[i]);
}
```

pinyin\_charcmp 函数的作用是对汉字的内码值做一个处理,因为 wch1 和 wch2 与实际的汉字内码的字节顺序是相反的,实现如下:

```
int pinyin_charcmp(unsigned short wch1, unsigned short wch2)
{
    wch1 = wch1 / 256 + wch1 % 256 * 256;
    wch2 = wch2 / 256 + wch2 % 256 * 256;
    return GET_CMP_CODE(wch1) -
        GET_CMP_CODE(wch2);
}
```

GET\_CMP\_CODE 宏首先通过 IS\_CN\_CHAR 宏判断该内码是否是一个汉字的内码,如果是则调用 GET\_PY\_CODE 宏,不是则按一般字符处理:

```
#define GET_CMP_CODE(wch) (unsigned int)(IS_CN_CHAR
```

```
(wch) ? GET_PY_CODE(wch) : (wch))
```

GET\_PY\_CODE 宏是实现该比较函数的关键,它的作用是由汉字的内码值得到经过排序的拼音码值,因此需要建立一张 GBK 编码到拼音顺序的映射表。为了保证效率,使用汉字内码减去一个常数 GBK-BEGIN1(最小的汉字内码值)作为 pinyin\_sort 数组的索引,而每个数组元素的值就是相应汉字的拼音顺序,这些值可以通过打印所有汉字并排序得到。另外,GBK 中存在大量没有编码的字位,但为了保证以内码作索引的高效,pinyin\_sort 数组中实际上约有 1/3 的空间是没有任何意义的。

GET\_PY\_CODE 宏的具体实现如下:

```
#define GET_PY_CODE(wch) (pinyin_sort[(wch) -
GBK-BEGIN1 + GBK-BEGIN1])
```

对于按笔画、使用频率等排序要求,其实现方式与上述的拼音排序一样,只需重新建立一张 GBK 编码到笔画顺序的映射表。

### 3.4 数据请求 SQL 语句

在具体手机开发中,实现了多个比较函数,并把它们封装为独立的 CAR 构件,在不同 UI 要求下,加载不同的比较函数 CAR 构件,就可以实现不同的数据处理要求,而且效率很高。例如,对于电话簿,如果想将其根据电话号码排序,可使用下面这条 C++ 语句:

```
ec = iStatement -> ExecuteQuery(L"select phoneBook. * from
phoneBook order by telephone;", &pIRS);
```

iStatement 接口用来执行静态的 SQL 语句,ExecuteQuery 方法<sup>[6]</sup>用于执行 select 语句,并通过 pIRS 返回查询的结果集的接口,通过该接口可以获得查询结果中各字段的信息。

如果想将电话簿根据姓名并按照拼音或笔画排序,可以使用如下的语句:

```
ec = iStatement -> ExecuteQuery(L"select phoneBook. * from
phoneBook order by name collate pinyin;", &pIRS);
ec = iStatement -> ExecuteQuery(L"select phoneBook. * from
phoneBook order by name collate bihua;", &pIRS);
```

## 4 总 结

文中所探讨的基于 Elastos 操作系统,通过在底层支持内容排序核心模块的方法具有一定的普遍性,不仅可以用在中英文电话簿中,还可以支持备忘录、歌曲名等多种检索排序的应用中。又由于 SQLite 数据库引擎和内容排序模块的 CAR 构件化,及 Elastos 操作系统对 CAR 构件的动态加(卸)载,以及应用软件的动态裁减和扩展的支持,使得 Elastos 智能手机对 3G 网络时代的数据业务的开展有相当的作用。

(下转第 201 页)

```

Text5.Enabled = True || Text6.Enabled = True
Text7.Enabled = True
Call XianShi
'调用段消息数据显示子程序
Call RecTemp '调用确认信息清除子程序
End If

```

End If

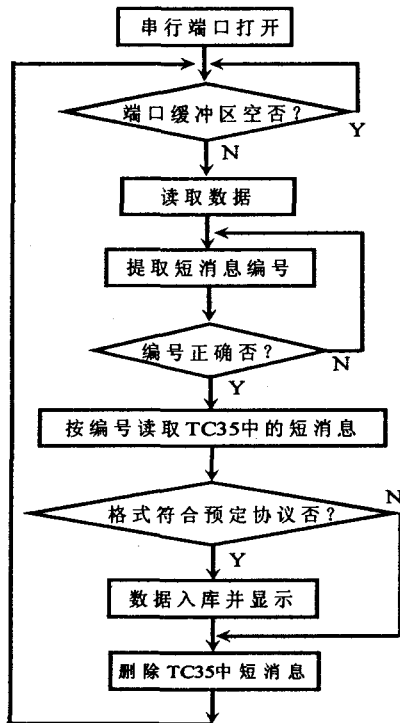


图 3 SMS 短消息接收和处理流程

### 2.2.2 监测数据管理模块

借助 VB.net 的 ADODC 模組调用 ACCESS 数据库实现,负责对后台监管记录数据库进行管理,又分为数据库连接与测试、新数据写入、已有数据检索、报表生成和打印模块。在此模块的代码行中关于数据库操作的语句采用 SQL 语法格式,以数据写入为例(其他操作实现方法类似),其代码如下:

```

Dim Ssql As String
Dim rs_ JILU As New ADODB. Recordset
Dim i As Integer, j As Integer, k As Integer, m As Integer
Ssql = "select * from 运营记录"
rs_ JILU. Open Ssql, conn, adOpenKeyset, adLockPessimistic

```

```

For i = 0 To (XinXiNum - 1)
For j = 0 To (ShuZuNum(i) - 1)
rs_ JILU. AddNew || rs_ JILU. Fields(0) = Date $ ||
rs_ JILU. Fields(1) = JiLuData(i, j).CheCi
rs_ JILU. Fields(2) = JiLuData(i, j).CheHao || rs_
JILU. Fields(3) = JiLuData(i, j).ZhuangTai
rs_ JILU. Fields(4) = JiLuData(i, j).ZhanDianBianHao
rs_ JILU. Fields(5) = JiLuData(i, j).DaoZhanTime
rs_ JILU. Fields(6) = JiLuData(i, j).ZhanZhanTime ||
rs_ JILU. Update
For k = 0 To 1000
Next k ' 时间延迟
Next j
Next i
rs_ JILU. Close

```

### 2.2.3 用户管理模块

本模块负责对软件用户(即操作员)信息的管理,借助 VB.net 的 ADODC 模組调用 ACCESS 数据库实现,其具体实现方法类似于监测数据管理模块,此处不再赘述。

## 3 结束语

经反复测试表明,本系统的软硬件在仿真环境和全真环境中均运行稳定,对公交车辆运营数据的采集和记录精度高、速度快,较为彻底地解决了公交车辆的不规范运营问题,促进了公交管理的规范化和自动化。

### 参考文献:

- [1] Pattison T, Dr Joe H. Building Applications and Components with Visual Basic. NET [M]. [s. l.]: Publishing House of Electronics Industry, 2004.
- [2] 刘文涛. Visual Basic + ACCESS 数据库开发与实例 [M]. 北京:清华大学出版社, 2006.
- [3] 李长林. Visual Basic 串口通信技术与典型实例 [M]. 北京:清华大学出版社, 2006.
- [4] 石磊. Visual Basic. NET 编程学习捷径 [M]. 北京:北京科海电子出版社, 2003.
- [5] 范逸之. Visual Basic 与 RS-232 串行通信控制 [M]. 北京:清华大学出版社, 2005.

(上接第 198 页)

### 参考文献:

- [1] 上海科泰世纪科技有限公司. CAR 构件技术大全 [EB/OL]. 2006-12-01. <http://www.elastos.com.cn/>.
- [2] Owens M. Embedding an SQL Database with Sqlite [DB/OL]. 2003-06-01. Linux Journal. <http://portal.acm.org/>.
- [3] Newman C. SQLite [M]. Indianapolis: Sams Publishing, 2004.
- [4] SQLite home page [EB/OL]. 2005-03-21. <http://www.sqlite.org/>.
- [5] Owen M. The Definitive Guide to SQLite [M]. Berkeley: Apress, 2006.
- [6] 上海科泰世纪科技有限公司. 和欣 2.0 资料大全 [EB/OL]. 2006-11-15. <http://www.elastos.com.cn/download/download.php?id=2>.