

Software Interface for Pocket PC Based Mobile Telephony

MIRCEA POPA¹, ANCA SORANA POPA², HORIA CIOCARLIE³

^{1,3}Computer and Software Engineering Department

²Department of Mechatronics

“Politehnica” University of Timisoara

No. 2, Piata Victoriei, 300006 Timisoara

ROMANIA

Abstract: - The widespread of the mobile communication is obvious in our days. More and more mobile devices are used in different domains: pagers, mobile phones, Pocket PCs etc. Pocket PC is a mobile computer included in the category of Personal Digital Assistant (PDA). It accomplishes all the functions of an organizer (address list, calendar, meeting list etc.) plus other more complex applications. An advanced version of a Pocket PC is the Pocket PC Phone Edition. Examples of models are: XDA and MDA. It is a Pocket PC plus mobile telephony capabilities. The need of connecting together more and more mobile devices and particularly the Pocket PCs to different networks (GSM, GPRS) became urged. The paper presents a complex software interface for the mobile telephony hardware module of the Pocket PC Phone Edition. The applications offer facilities such as: navigation in contacts data bases from the device and from the SIM card, sending SMSs to contacts from the data bases, indication if SMSs are received, visualization of all the received, missed or sent from the device calls, receiving and sending phone calls. A very simple and intuitive graphical user interface was developed.

Key-Words: - mobile communication, Pocket PC, mobile telephony, SMS, telephony services, class diagram

1 Introduction

The widespread of the mobile communication is obvious in our days. More and more mobile devices are used in various domains: pagers, mobile phones, Pocket PCs etc.

Pocket PC, [1], is a mobile computer included in the category of Personal Digital Assistant (PDA). It accomplishes all the functions of an organizer (address list, calendar, meeting list etc.) plus other more complex applications such as: text editor, data base, mathematical operations, clock, data transfer with a PC etc. They run an operating system such as Windows CE or Palm OS. There are many software applications running on PCs for Pocket PCs, [2].

An advanced version of a Pocket PC is the Pocket PC Phone Edition. Examples of models are: XDA and MDA. It is a Pocket PC plus mobile telephony capabilities. They have support for GSM through CSD (Circuit Switched Data), GPRS (General Packet Radio Service) connection, support for receiving and transmitting phone calls and SMSs etc. They have also extension possibilities using different solutions:

- Compact Flash cards, which offer: external memory, Wireless Ethernet (802.11b) network, GSM or GPRS modems, GPS (Global Positioning System) module, Radio module etc.;
- PCMCIA cards for connecting modules such as modems, network cards, wireless accessories etc.;
- SD (Secure Digital) card and
- Bluetooth support.

The need of connecting together more and more mobile devices and particularly the Pocket PCs to different networks (GSM, GPRS) became urged.

The paper presents a complex software interface for the mobile telephony hardware module of the Pocket PC Phone Edition. This software appeared from the need of a practical and a performing tool for the mobile telephony capabilities of a Pocket PC. The application offer facilities such as: navigation in contacts data bases from the device and from the SIM card, sending SMSs to contacts from the data bases, indication if SMSs are received, visualization of all the calls received, missed or sent from the device, receiving and sending phone calls. A very simple and intuitive graphical user interface was developed.

The next section presents similar work, the third section details the developed software and the last section outlines the conclusions.

2 Related Work

The mobile communication and particularly the Pocket PC's mobile communication were approached in different papers.

Reference [3] describes an application in which Palm (or Pocket PC) computers are used in the d-learning process. With the evolution of the mobile devices and technologies the d-learning process evaluated to e-learning and then to m-learning (mobile learning). The

Palm devices are connected to the communication media using infrared channels.

Another application based on mobile technology is described in [4]. A security manager was built to remotely monitor buildings for intrusions. PDAs are used as front-end devices.

Reference [5] shows that Web information is not yet as accessible from PDAs as from desktop computers. A mechanism for Web directory navigation for PDAs is described based on navigation actions locally done with intermittently communication with the server.

The application described in [6] was thought to provide mobile users with access to rich multimedia information and services and for that approaches were investigated for overcoming the display limitations of a PDA. The CMCE (Composite Device Computing Environment) framework was developed situating the PDA in the center of other surrounding computing resources (PCs, telephones, TVs etc.). A unified composite, virtual device is created from the PDA and the surrounding resources.

Reference [7] presents an application in which PDAs are used as electronic memory aids for disabled or elderly users.

The application from [8] is addressed to computer network managers. Often they must solve urgent network problems but have not access to a computer. A tool is described for network management that uses a WAP interface.

In [9] mobile technologies are used in a pervasive specific system. A solution for retail services based on wireless devices is described.

Reference [10] approaches an important disadvantage of the PDAs: the display limitations. Web pages do not display well on the small displays of the PDAs and the user becomes disoriented. Overview and detail concerns of viewing must be treated.

A software interface for the mobile telephony hardware module of the Pocket PC Phone Edition is described in this paper. Facilities such as: navigation in contacts data bases from the device and from the SIM card, sending SMSs to contacts from the data bases, indication if SMSs are received, visualization of all the calls received, missed or sent from the device, receiving and sending phone calls are provided. A very simple and intuitive graphical user interface was developed.

3 The Software Interface

3.1 The architecture

There are many applications for the mobile telephony capability of the Pocket PC but in most of the cases they

do not integrate more services. Fig. 1 shows this situation in a graphical representation.

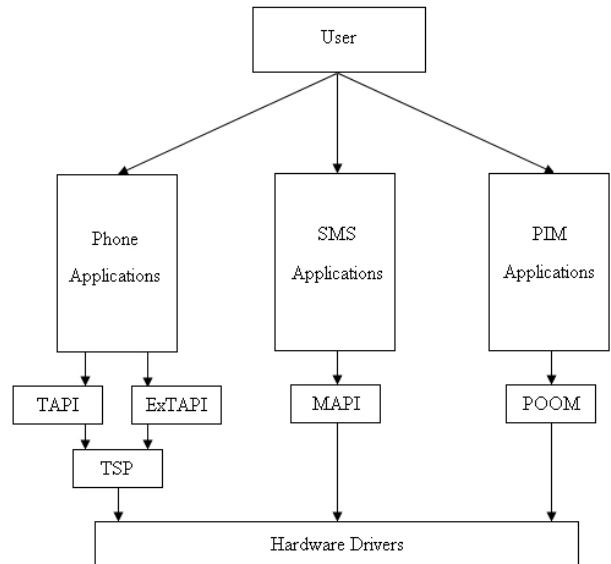


Fig. 1

The software interface from this paper unifies more mobile telephony services and facilities. Fig. 2 presents this approach.

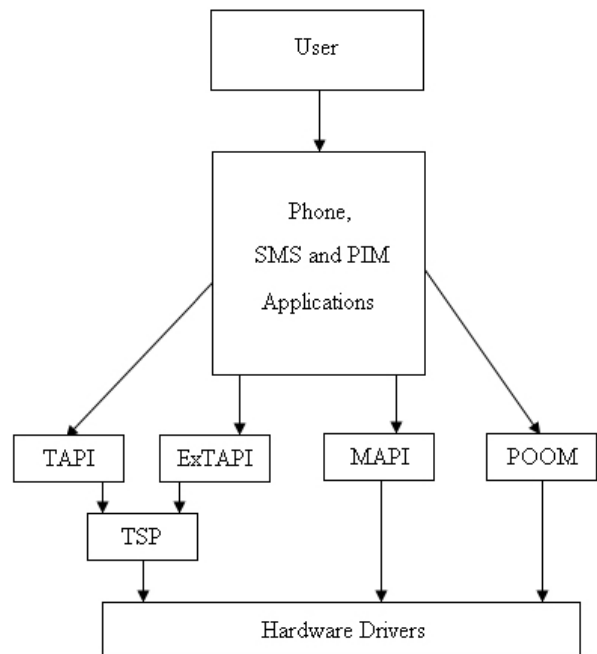


Fig. 2

Fig. 3 shows the structure of the application. The application is structured on two levels, providing flexibility, extension and reuse possibility of different components (modules, classes). The first level is made by the graphical user interface module and the second

level consists in four libraries dynamically linked, each one being responsible with different functionalities implemented. There are also levels inside the mentioned two levels. As part of the libraries the stratification is accomplished by the use of the *Bridge* design pattern which separates the abstractions from the abstractions them selves. As part of the graphical user interface the levels are done by class inheritance.

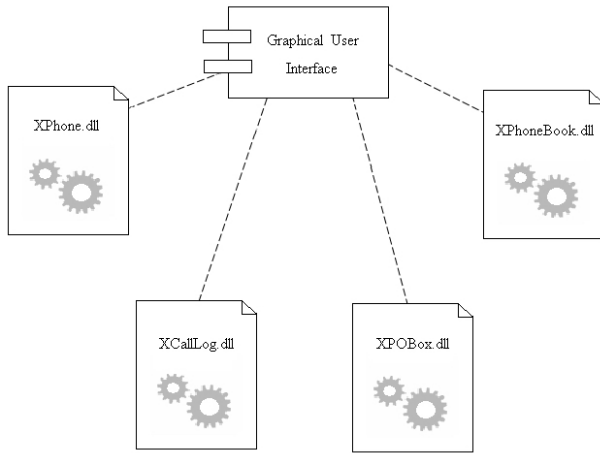


Fig. 3

The library XPhone.dll contains classes offering the possibility to receive and send phone calls. Other functions from the same library implements functionalities such as: deblocking of SIM cards protected by PIN codes, the name of the mobile telephony operator at which the device is connected, information about the level of the signal received by the device etc.

The library XPhoneBook.dll ensures the visualization and the use of the data base with the contacts, both Outlook type and from the SIM card.

The library XCallLog.dll contains classes for recording all the missed calls and the calls from the mobile device. Methods are defined for navigating between the recorded calls.

The library XPOBox.dll manages the SMSs. There are classes for receiving new messages, for reading already existing messages, for clearing messages and for editing and sending messages.

3.2 The library XPhone.dll

It has the following tasks:

- generates phone calls to a specified number;
- receives new calls;
- reports the state of the virtual phone;
- provides the name of the GSM operator and
- reports the signal level.

Figure 4 shows the class diagram of the library. As in the following diagrams too, only the most relevant classes for the respective module are presented and not all the presented classes are completely presented, some functions, private members or parameters being ignored.

The first two classes, *CObservable* and *IObserver* were built with the *Observer* design pattern. The *IPhone* class is derived both from the mentioned classes.

The second design pattern used was *Bridge*. The four classes based on this design pattern are *IPhoneImpl* and *IPhone* (with interface role) and *CXdaPhoneImpl* and *CbmwPhone*. Examples of functions are: *AcceptCal* for received calls, *DialNumber* for calls to be sent and *GetNumber* for obtaining information about the other person. In the *CXdaPhoneImpl* and *CbmwPhone* classes there are two methods, called *OnDialledID* and *OnCallerID*. They are callback type methods that is they will be accessed when the modules TAPI and ExtAPI obtain the number called and the number of the caller. An other function, *lineCallBackFunc*, receives and treats messages arrived from the inferior levels, referring the radio part of the device. The *CXdaPhoneImpl* and *CbmwPhone* classes implement the *Singleton* design pattern.

3.3 The library XCallLog.dll

The role of this dynamically linked library is to create an appropriate interface for the data base of the mobile device. The data base contains lists with different kinds of calls: lost, done, receipted. Its client is the graphical user interface module. The methods which can be used for navigating are: *First*, which sets the cursor on the first position from the calls list, *Next*, which decrements the cursor position on the selected list, *Previous*, which increments the cursor position on the selected list and *isDone* which returns a binary value indicating if the cursor is on the last position of the list.

The data of a real person are encapsulated in the class *CPerson*. The structure and its implementation are given below:

```

class CPerson
{
protected:
    CString m_strName ;
    CString m_strMobileNumber ;
    CString m_strHomeNumber ;
    CString m_strOfficeNumber ;
public:
    CPerson( CString strName
            CString strMobileNumber = ""
            CString strHomeNumber = ""
            CString strOfficeNumber = ""
            m_strName(strName) ,
            m_strMobileNumber(strMobileNumber) ,
            m_strHomeNumber(strHomeNumber) ,
    
```

```

    m_strOfficeNumber(strOfficeNumber{}
    CString GetName() {return m_strName;}
    CString GetMobileNumber() {return
m_strMobileNumber;}
    CString GetHomeNumber() {return
m_strHomeNumber;}
    CString GetOfficeNumber() {return
m_strOfficeNumber;}
};

```

As it can be observed, this class encapsulates four information about a person: name and three phone numbers. This class does not offers direct access to its members but only public methods through which the internal values can be obtained. It is a protection against the accidental modification of the contained values.

There are also three functions, *ADDFilter*, *RemoveFilter* and *SetFilter*, which offer the possibility to set, add or eliminate filters for the access to the data base. They receive an integer type parameter which can take one of three values defined in the program. The values are defined as it follows:

```

#ifndef FMISSED
#define FMISSED 0x000F
#endif
#ifndef FDIALED
#define FDIALED 0x00F0
#endif
#ifndef FRECEIVED
#define FRECEIVED 0x0F00
#endif

```

Combinations between the three constants, using the OR operation, are also possible. In order to effectively action on the data base the Refresh function was created. It can be used even if there are not modifications of the filter and then the effect will be the cursor positioning on the first position. Fig. 5 presents the class diagram of the *XCallLog.dll* library.

3.4 The library XPhonebook.dll

The role of this library is to offer to the client the possibility to access the two data bases with contacts. There are two formats for the contacts: POOM (Pocket Outlook Object Model) and SIM card format.

There is one class, *CBmwPhoneBook*, for accessing the two data bases. Its implementation is:

```

class XDAPHONEBK_API CBmwPhoneBook
{
private:
    CPersonList *m_pList;
    CPersonListElement *m_pElement;
    int m_iSource;
    CString m_strFilter;
    IPhoneBookImpl *m_pPhoneBookImpl;

```

```

    Bool GetNewPerson ();
public:
    void First()
    void Next();
    void Previous();
    bool isFirst();
    bool isLast();
    CPerson GetPerson();
    void SetFilter (CString Filter);
    void SetSource (int Source);
    CBmwPhoneBook(int Source, CString
Filter);
    ~CBmwPhoneBook();
};

```

A header file will be necessary, containing two constants, for setting the source of contacts: Outlook or SIM card. The file is presented below:

```

#ifndef F_XDZ_PHONEBOOK
#define F_XDA_PHONEBOOK 0x0F
#endif

#ifndef F_POOM_PHONEBOOK
#define F_POOM_PHONEBOOK 0xF0
#endif

```

Other values or combinations between the mentioned values are not allowed. A filter can be applied on the names of the selected contacts, using the function *SetFilter(CString Filter)*. This function receives a CString type parameter (a MFC class which abstracts a string of characters) which will be memorized as a private member and applied to the calls list of the method *GetPerson*.

There are two classes which implement the data bases with contacts: *CXdaPhoneBookImpl* for Outlook type contacts and *CPoomPhoneBookImpl* for contacts from the SIM card.

3.5 The library XPOBox.dll

It is responsible with the management of the SMSs, including the receipt and transmitting of messages, the visualization and the clearing of the existing messages.

This library uses VOMAPI modules for message services. These modules are situated above the MAPI level. MAPI (Messaging Application Programming Interface) are operating system's modules used by applications for creating, processing, transferring and recording e-mail and SMS messages. These functions provide the necessary tools for defining the scope and the content of the messages offering, at the same time, flexibility in managing the existent messages in internal data bases.

The VOMAPI classes were represented and the connections to the implemented classes were also

distinguished. The *CVOMAPIMessage* class encapsulates the data of a message and the operators applied to the data. The *CVOMAPI* class is the general class offering notifications when messages arrive and functions for managing the existent classes.

A new abstraction appears that of a message represented by the *CSms* class. It contains a *CPerson* type object in which the data about the sender and the content of the message, the arrival date and the number of the call center through which the message was received is memorized.

3.6 The graphical user interface module

It is responsible with the creation and the management of the graphical user interface. It calls all the facilities provided by the libraries created at user's request. All the four dynamically linked libraries were used for creating this module. The module was statically linked with the four .LIB type files generated at the libraries creation.

4 Conclusions

The created software has advantages from the user's and from the developer's points of view. The user's advantages are:

- the possibility to use all the mobile telephony capabilities of the Pocket PC Phone Edition and
- he has an integrated system which includes services usually offered as separate software packets.

The developer's advantages are: modularity and flexibility. The system has a decentralized architecture thus reducing the complexity and helping to isolate possible errors.

Further improvement directions are:

- the integration of a GPRS module for transferring data;
- synchronization services for the contacts memorized on the SIM card and in the Outlook memory of the system or memorized on two devices running the application or on two devices running two different applications;
- the portability of the created software on other Pocket PCs too, e.g. on a Pocket PC with an external modem which can be a PCMCIA or Compact Flash card.

Other improvements can be done at graphical interface level: the possibility to modify with configuration files the entire interface including the positions of the buttons, the states of the buttons, the colors, the fonts and the possibility to use graphical files with different kind of compressions.

References:

- [1] Pocket PC Developer's Network, <http://www.pocketpcdn.com>
- [2] Pocket PC 2002 Emulator, <http://www.pocketpcdn.com>
- [3] Sharma, S.K., and Kitchens, F.L., Web Services Architecture for M-Learning, *EJEL, Electronic Journal of e-Learning*, Vol. 2, Issue 2, December 2004
- [4] Sanna, A., and Fornaro, C., IMoViS: A System for Mobile Visualization of Intrusion Detection Data, *INFORMATION&SECURITY. An International Journal*, Vol. 12, No. 2, 2003, pp. 235 – 249
- [5] Cohen, D., Herscovici, M., Petrushka, Y., Maarek, Y.S., Soffer, A., and Newbold, D., Personalized Pocket Directories for Mobile Devices, *Proceedings of the Eleventh International World Wide Web Conference, WWW2002*, Honolulu, Hawaii, USA, May 7 – 11, 2002
- [6] Pham, T.L., Schneider, G., Goose, S. and Pizano, A., Composite Device Computing Environment: A Framework for Augmenting the PDA Using Surrounding Resources, *Proceedings of the ACM CHI2000 Workshop on Situated Interaction in Ubiquitous Computing*, The Hague, The Netherlands, 1-6 April 2000
- [7] Inglis, E.A., Szymkowiak, A., Gregor, P., Newell, A.F., Hine, N., Shah, P., Wilson, B.A., and Evans, J., Issues surrounding the user – centred development of a new interactive memory aid, *Universal Access in the Information Society*, Springer Berlin/ Heidelberg, Vol. 2, No. 3, 2003, pp. 226 – 234
- [8] Vieira Jr., A.C., and Anido, M.L., A Novel Tool for Network Management Using WAP Interface, *Proceedings of the 18th International Conference on Advanced Information Networking and Applications, (AINA '04)*, Vol. 1, Fukuoka, Japan, March 2004
- [9] Roussos, G., Kourouthanasis, P., Spinellis, D., Gryazin, E., Pryzbliski, M., Kalpogiannis, G., and Giaglis, G., Systems architecture for pervasive retail, *Proceedings of the ACM Symposium on Applied Computing*, New York, 2003, pp. 631 – 636
- [10] Rodden., K., Milic – Frayling, N, Sommerer, R., and Blackwell, A., Effective Web Searching on Mobile Devices, *Proceedings of HCI2003, the 17th Annual Human – Computer Interaction Conference*, Bath, England, 8 – 12 September 2003

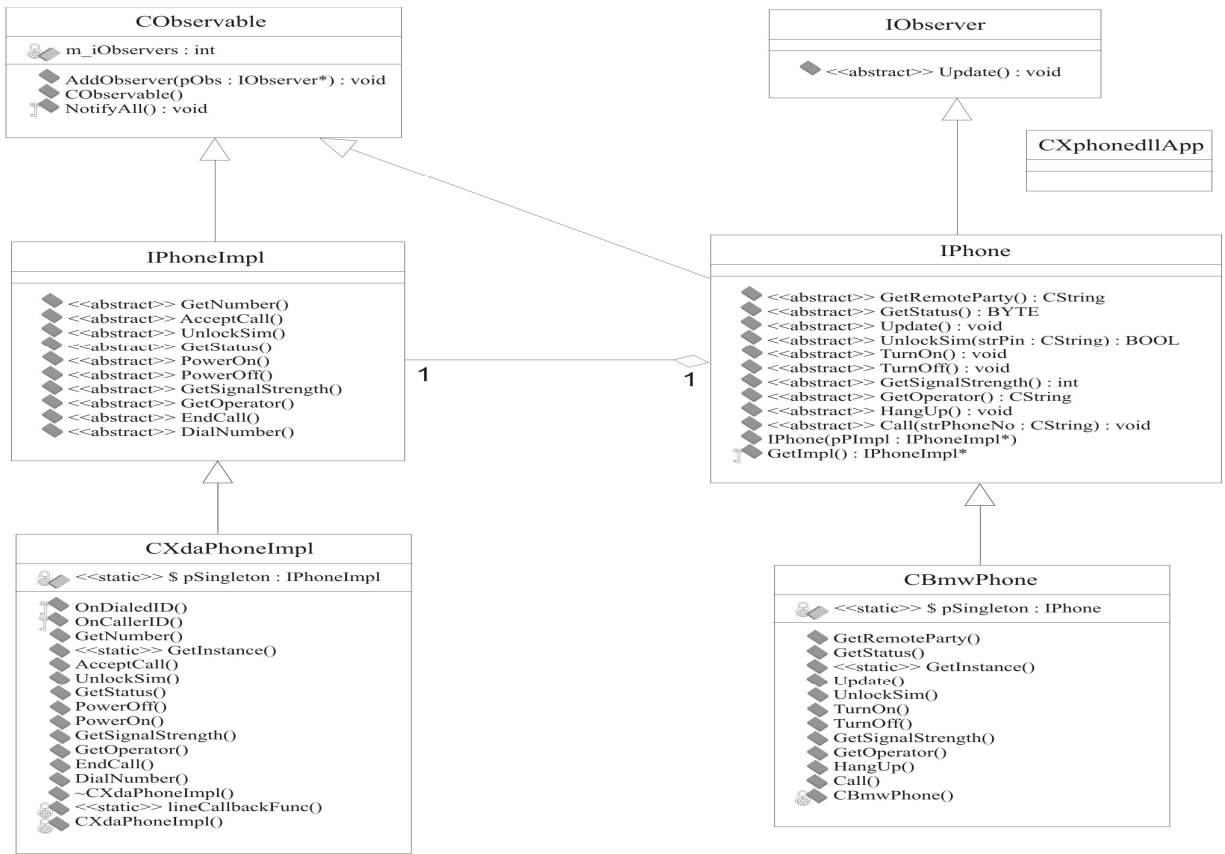


Fig. 4. The class diagram of the XPhone.dll library

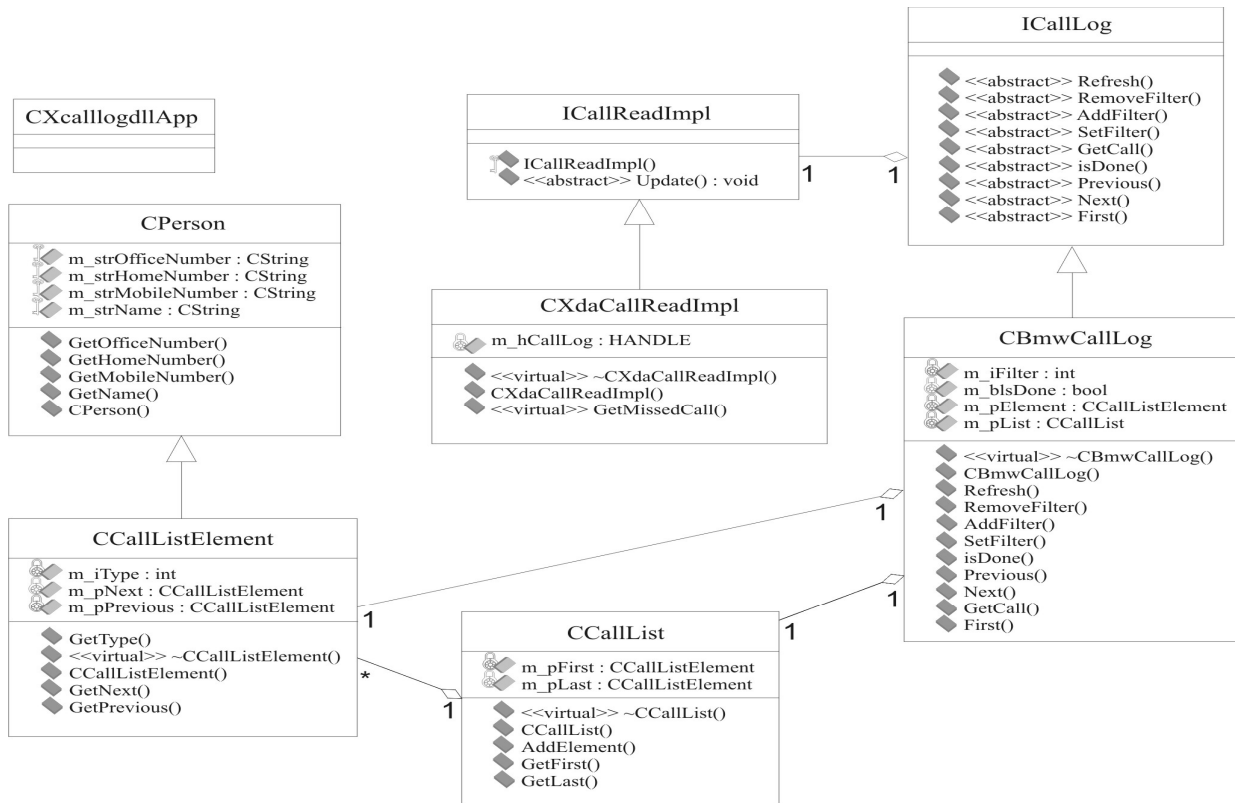


Fig. 5. The class diagram of the XCallLog.dll library