Generating Remote Control Interface Automatically into Cellular Phone for Controlling Applications Running on PC[*]

DENG-JYI CHEN, SHIH-JUNG PENG AND CHIN-ENG ONG
Department of Computer Science and Information Engineering
National Chiao Tung University
Hsinchu, 300 Taiwan

Generally speaking, we could conveniently and directly operate ordinary user interface by hand controller. But along with the progressing techniques and increasing functions of communication equipment for family use, the operation and interaction styles between people and device have become more and more abundant. Most of the multimedia contents can be run and displayed on different kinds of platforms without possessing remote control ability originally. If people can just use some simple instruments, such as cellular phone or PDA, to remotely control the multimedia application module running on the PC or digital TV, then the control would become vivid and interesting. But there are various controlled instruments, display devices, and different kinds of methods. People who want to make a specified device to have remote control ability functions must write many complex procedures or programs into the device. Even in just adding or modifying a new function, people must also find the original source program and design function procedures repeatedly, which make developing application systems a cost burden, waste of time, and inefficient.

In this paper, we will construct a remote control interface generator system, under which designer can easily transfer specified control object of Java application system running on PC, such as control buttons and labels, and generate remote control interface with objects into cellular phone automatically. This will simplify the development process of creating a control interface and make the control system development and modification more flexible and elastic. Through directly generated controlling objects and functions into the cellular phone by interface generator, we could directly and easily control the Java application system running on the PC without needing to write complex programs into the cellular phone.

Keywords: cellular phone interface, cellular phone programming, remote controller, remote control interface, interface generator, bridge interface

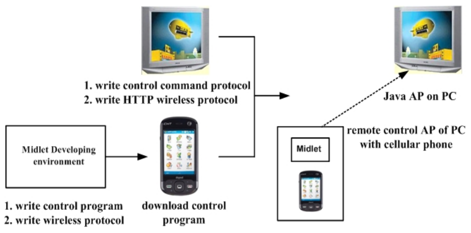
1. INTRODUCTION

As it is known, we could conveniently and directly operate ordinary user interface by hand controller. But along with progressing existing techniques and increasing functions of communication equipments for family use, operation and interaction styles between people and device become more and more complex. Most of the multimedia contents can be run and displayed on different kinds of platforms without possessing remote control ability originally, so, people think if they can just use some simple instruments, such as cellular phone or PDA, to remotely control the multimedia application module

Received March 26, 2008; revised July 15, 2008; accepted September 18, 2008.
Communicated by Chih-Ping Chu.

[*] This research was supported in part by the National Science Council of Taiwan, BestWise International Computing Co., CAISER (National Chiao Tung University, Hsinchu, Taiwan), and Ta Hwa Institute of Technology (THIT, Hsinchu, Taiwan).

running on the PC or digital TV, then the control would become vivid and interesting. But because there are various control instruments, display devices, and different kinds of methods, it is not easy to make a specified device to have remote control ability. To do this, we must repeat the design process: (1) Write control command protocol and HTTP wireless protocol into the applications running on the PC; (2) Write interface program and wireless control protocol into cellular phone, as in Fig. 1, for every application system one after another. But this designing process is hard work; we must write many complex procedures even for just adding or modifying a new control function. After that, there still remains a big problem that if we do not have the original source program of application system, it could become impossible to make the specified device have remote control ability. Such repeated design procedures would make developing application systems a cost burden, waste of time, inelastic, and inefficient.


1. write control command protocol
2. write HTTP wireless protocol
Java AP on PC
Midlet Developing environment
remote control AP of PC with cellular phone
Midlet
1. write control program
2. write wireless protocol
download control program
Fig. 1. General developing method.

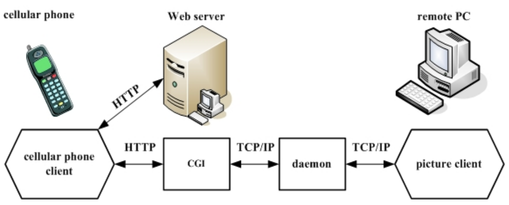


Fig. 2. Rajicon system architecture.

2. RELATED WORK

Remote control methods have been discussed recently for many years; in 2001, it [1] was proposed that cellular phone be used to control remote GUI running on the PC. Cellular phone is a mobile device possessing a small screen but cannot show the same GUI as on the PC screen. So we must analyze GUI and develop some function programs in the cellular phone for its use in the remote control interaction with GUI on the PC. In view of remote controller, client-server is exactly a simple framework. At first, the client sends request to the server through OTA (on the air). After receiving this, the server parses the command and accesses it, then sends the result response back to the client. This interact agreement is developed for a specific device; for establishing this, we must consider about the device operating ability, whether it supports multimedia content and relative remote control function. When the interaction functions aimed at become more and more continual, the remote control interface and interaction protocol development would become more complex and difficult.

Rajicon System [2] was developed in 2002. Its main function is using cellular phone to control the remote PC, as in Fig. 2. It works on a specific device, so, if we want to do some complex operation functions, we must define a set of interactive rules. Rajicon System uses macro command to formulate every operating function, so, the more operating functions there are, the more macro commands it would need, as described in Table 1.

Through pressing keypad of cellular phone in order to input command to control application running on the PC, these operation methods are complex and undirected, and the control commands are not easy to memorize. On the other hand, this system is

Table 1. Part remote control functions with cellular phone keypad mapping.

Key	Zoom mode	command	result
1	View entire desktop	?a	Hold down Alt key until next key pressed
2	Dec cursor height	?c	Hold down Ctrl key until next key pressed
3	Refresh	?n	Pressed Enter key
4	Dec cursor width	?m	Press left mouse button
5	Execute macro	?^	Maximize the window at the cursor

designed for a specific device, if there are some control functions to be created or modified; such designed procedures should be restarted repeatedly.

In 2002, Jeffery Nichols and Brad A. Myers issued a topic 'Generating remote control interfaces for complex appliances' [3]. They proposed a method of personal universal control (PUC) and the document content is about how to create a control interface with graphics or speech by downloading a functional specification explanation of equipment for family use, as in Fig. 3. After that, PUC system would analyze specification document and use decision tree algorithm [5] to create a specification group tree, as in Fig. 4, then according to this group tree structure to establish an appropriate control interface. But there are many design factors needed to be considered, including: (1) Before establishing an interface, it had to download a functional specification explanation for a specific equipment, but this specification description is not easy to establish; (2) It is tedious to design a control interface, because interface designed algorithm need to consider user's requirement with the presentation of control device objects.

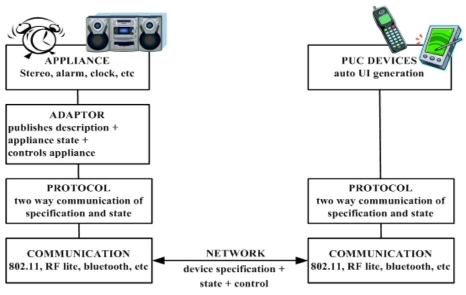

APPLIANCE
Stereo, alarm, clock, etc
ADAPTOR
publishes description +
appliance state +
controls appliance
PROTOCOL
two way communication of specification and state
COMMUNICATION
802.11, RF lite, bluetooth, etc
NETWORK
device specification + state + control
PUC DEVICES
auto UI generation
PROTOCOL
two way communication of specification and state
COMMUNICATION
802.11, RF lite, bluetooth, etc

Fig. 3. PUC system architecture.

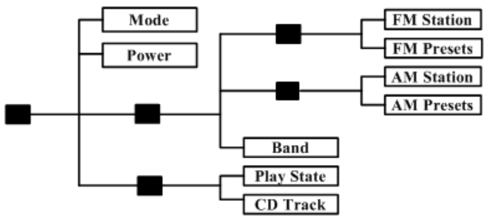


Fig. 4. A sample group tree for a shelf stereo.

In [7], because traditional remote control typically allows users to activate functionality of a single device, they present qualitative and quantitative results from a study of two promising approaches creating such a remote control: end-user programming and machine learning. In end-user programming, users manually assign the buttons they believe are sufficient to accomplish their tasks to graphical 'screens'. They then work with a single, handheld remote control that can display those screens. Machine learning also uses a single, handheld remote to display screens; however, it uses the recorded history of a user's actual remote interactions to infer appropriate groups of buttons for the performed tasks. There may be some questions in the following [7]: (1) The end-user programming will be very complicated while carrying on the design of remote control to the device with graphical screens. User must define the components needed in operating on a screen that can control many device interfaces remotely at the same time, and all the control procedures need a large number of programs to be written; (2) The ML can record the operation of user automatically and utilizes the performing algorithm to produce the operating component. Perhaps because of the difference of device, it cannot completely define each function of component of the device; (3) The method of automatically producing remote control interface needs to accord to the characteristic of the device and the favorite of user, but it also needs a large number of procedures to be written in order to design algorithm automatically. There are some merits with adopting our interface generating system: (1) the designer can simplify program and design the complicated control procedures that have the characteristics of code reusable by using interface-generating system to define the object and generate operating interface automatically; (2) Analyze the objects that users defined automatically, and produce procedure control and corresponding remote control operation interface; (3) Standardizes the way that the user defines the object, to help user to produce the operation procedures and interface on remote control with the device. Such procedure does not have the need to write complicated mathematical calculations or design under a specific interface.

In [9], they described a new widget and interaction technique, known as a "Frisbee", for interacting with areas of a large display that is difficult or impossible to access directly. It consists of a local "telescope" and a remote "target". This design satisfies five design principles of: (1) minimizing physical travel, (2) supporting multiple concurrent users, (3) minimizing visual disruption while working, (4) maintaining visual persistence of space, and (5) application independence. However, it needs to write a large number of procedures for the specific and large-scale showing device. Moreover, because of the limitation of hardware specification and the difficulty in obtaining the relevant information, it is comparatively difficult and complicated to design the interface. Our interface generating system could automatically produce remote control system and relevant operating method; it will not be limited to the specification of the hardware, and it can offer many different devices to carry on remote control interaction.

Along with the functions of periphery equipment gradually increased, the design of remote control interface in the cellular phone has become more and more complex and difficult. In this paper, we hope to construct a remote control interface generating system. Through that, designer can easily transform control objects of JAVA application program running on the PC, such as control buttons or labels, into cellular phone automatically. Finally, using the keypad or touching panel of cellular phone controls the JAVA applications running on the PC directly and easily. The key functions of this interface system include: (1) It can wrap operate mode and static endue each operation item a control command, so there is no need to define complex operating instructions; (2) It can analyze operating objects of JAVA applications, and produce description file to the interface generator. New operating functions can be developed quickly and effectively to generate interface operating program into the cellular phone; (3) There are many written kinds of specific abstract procedures of template, and these can be reusable by inheriting. System designer can develop new Java multimedia function and transform new service through inheriting template procedure easily, quickly, and effectively, even add or modify remote control function without needing to know the source program of applications. This will simplify the development process and make the control system development and modification more easy, effective, and flexible.

3. PROPOSED METHOD AND ALGORITHM

3.1 Structure

On the traditional developing of remote control functions into the cellular phone, designer need to write the MIDlet of cellular phone at first, and then write the JAVA AP controlling statement including interaction between PC and cellular phone. This designed process is hard work, a waste of time, and inefficient.

Our solution for solving this problem is to construct an interface generating system; with it the designer can easily develop remote control interface program into cellular phone without even needing to write the program of cellular phone. Finally, with using this cellular phone as a remote controller, user can interact with the JAVA application running on the PC directly and easily. The framework of remote control interface system is as in Fig. 5.

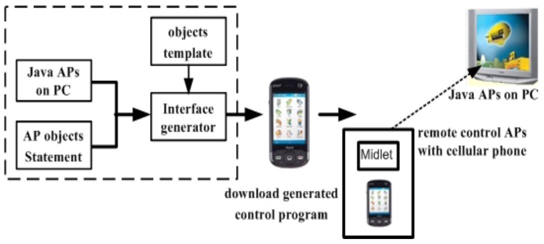


Fig. 5. Framework of remote control interface system.

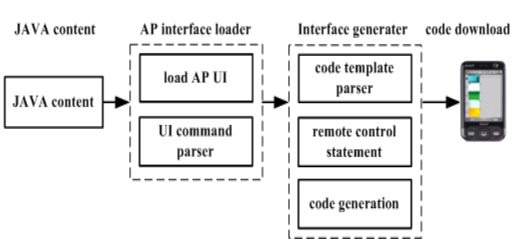
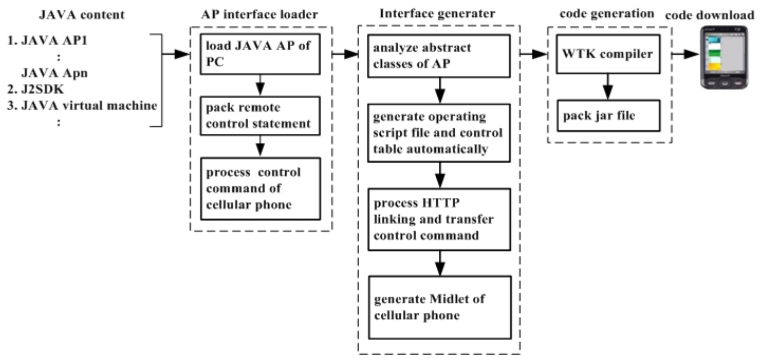


Fig. 6. Cellular phone interface generated modules.


JAVA content
1. JAVA AP1
:
JAVA Apn
2. J2SDK
3. JAVA virtual machine
:
AP interface loader
load JAVA AP of PC
pack remote control statement
process control command of cellular phone
Interface generater
analyze abstract classes of AP
generate operating script file and control table automatically
process HTTP linking and transfer control command
generate Midlet of cellular phone
code generation
WTK compiler
pack jar file
code download
Fig. 7. Cellular phone interface generated procedures.

The generator modules include JAVA content, AP interface loader, and interface generator, as in Fig. 6. Module "JAVA content" means the JAVA applications running on the PC to be controlled with cellular phone. Module AP interface loader includes "AP interface loader", which loads JAVA application into and runs it on the PC, "packs remote control statement", and "UI command parser" which processes the interface control command of cellular phone. Finally, module "interface generator" includes "code template parser", which analyzes abstract classes of AP, and according these classes to generate operating script file and control command table, "remote control statement", which processes HTTP linking and transfer control command, and "code generation", which generates MIDlet of cellular phone automatically according to the operating script file. Finally, it executes the Wireless Toolkit (WTK) compiler and packs program into jar file, then designer downloads this jar file into the cellular phone, as in Fig. 7.

3.2 Algorithm

Control interface systematic framework is composed of application program interface loader, cellular phone interface generator, and a Java application program. Application program interface loader is a graphic user interface (GUI) for designer which helps loading JAVA applications running on the PC into interface generator easily. It is a JAVA server procedure; system user makes JAVA application program template and provides this to the interface loader. After loading application interface program, it links and transforms operation objects, such as Javax.swing.JButton and Javax.swing.JRead Button, into commands ID, such as '001#' for play_btn_1 and '002#' for play_btn_2. In the following, we describe algorithms of these modules individually.

3.2.1 Algorithm of application program interface loader

Algorithm 1.1 Loading AP UI

```
1. Initializing java swing GUI component, such as JFrame, JSplitPane, JScrollPane,
   JMenuBar, JMenu, JRadioButtonMenuItem and JDesktopPane.
2. Create look and feel properties by calling setLookandFee (String) function.
3. Add all kinds of GUI components into JFrame by calling JFrame.add (Component).
4. Set default size and look&feel of JFrame.
5. Waiting for launching Java Application Program.
6. Initializing Java Application Program.
7. Loading the JInternalFrame of the target Java Application Program.
   If (ActionListener of JMenu Received JMenuItem ActionEvent)
   {  Switch (ActionEvent)
      { Case 'Launch Java _AP_X':                    // X: 1 ~ N
        LoadAP (Java_AP_X);
           :   } }
8. Parsing the code template (abstract class) of the target Java Application Program by
   calling BufferedReader (FileReader) and FileReader (abstract class name: String).
   BufferedReader br = new BufferedReader (new FileReader ('JavaAPName'));
   While (br.ready ())
   {  String s = br.readLine ();
      Parsing s and retrieve the String Token;
      Switch (String Token)
      {  Case 'play_btnN':                            //N: 1 ~ n
            Get the command ID value of play_btnN;     // ex: play_btn1 = '001#'
         Case 'btnN_Icon':
            Get the Icon path of play_btnN;           // ex: btn1_Icon = '1new.jpg';
            :   } }
```

```
9. Generating the Operation File ‘AP_ControlTable.txt’ by calling BufferedWriter
   (FileWriter), FileWriter (AP_ControlTable.txt: String).
   BufferedWriter bw = new BufferedWriter (new FileWriter (‘AP_ControlTable.txt’));
   While (br.ready ())
   {  String s = br.readLine ();
      If (s.startsWith (‘protected String’))
      {  st = new StringTokenizer (s, ‘’);
         ValueOfCmd = st.nextToken ();
         bw.write (Command ID + ‘’ + ValueOfCmd);
         bw.newLine (); } }
10. Load the JTable of the target Java Application Program.
    br = new BufferedReader (newFileReader (‘AP_ControlTable.txt’))
    While (br.ready ())
    {  String content = st.nextToken ();
       Switch (content)
       {  Case ‘TYPE’:
             Assign content to CurrentTableData[X] [0];    // X: 0 ~ n
          Case ‘ICON’ | ’LABEL’:
             Assign content to CurrentTableData[X] [1]; } }
```

According to system framework of this paper, each Java application operating screen can be load in by application program interface loader, which is designed by adopting JAVA swing groupware. From lines 1 to 4, before loading applications, we first initialize actuating devices of interface loader which is like a vessel containing application interfaces, such as Menu, Menu Bar, Radio Button, Split Pane, and Scroll Pane. When application is loaded in, we parse program code and find out all command ID of actuating devices on this application at line 8, then record the searching result into file AP_Control Table.txt as at line 9.

Algorithm 1.2 Parsing UI Command

```
1. Waiting for HTTP URL request from the MIDlet Program in mobile phone.
2. Parsing the HTTP parameters to retrieve the command ID.
   response.setContentType (‘text/html’);
   PrintWriter out = response.getWriter ();
   String command = request.getParameter (‘message’);
3. Retrieve command value in JTable by calling getValueAt (row index, column index);
4. Compare HTTP command ID and value in JTable by String.equals (String).
   For  (int i = 0, i < row count, i++)
   { If (command.equals (JTable.getValueAt (i, column index)))
        rc_setXXXActionPerformed (command); }
5. If command is identical, call remote control functions ‘rc_setXXXActionPerformed’.
```

In fact, interface loader exactly plays a server role. After recording all operating project and command ID, interface loader then waits for HTTP connection from remote cellular phone procedure. From lines 1 to 4, it illustrates that while interface loader received HTTP command, it would compare with the field value in JTable to find what kind of remote control command is delivered. This is named as rc_setXXXActionPerformed which supplies programmer with being able to define and write its functions, as at line 5.

3.2.2 Algorithm of interface generator

Algorithm 2.1 Code Generation

```
1. Generate the Remote Displayable Form.
   Import javax.microedition.lcdui.*;
   Import java.io.*;
   Public class RemoteDisplayable extends Form implements CommandListener
2. Generate MIDlet class.
   Public class remoteMIDlet extends MIDlet
   {  static remoteMIDlet instance;
      RemoteDisplayable displayable = new RemoteDisplayable2 ();
      Display display; }
3. Decide the Component Types (ex: Button/RadioBtn) in the command table.
4. Generate the Main List included possible list items of Component Types.
   If (parsing the command types in table = BUTTON|RADIOBTN|COMBOBOX)
   {  String Component_Array [] = new String [] {"Button", "RadioBtn", "ComboBox"};
      mainList = new List ("main menu", List.EXCLUSIVE, Component_Array, null);
      mainList.addCommand (mainList_OK_cmd);
      mainList.addCommand (mainList_EXIT_cmd);
      mainList.setCommandListener (this); }
5. Generate the operation items in operation list; a main list may include several opera-
   tion lists-Button/RadioBtn/Combobox).
   For (int i = 0; i < NumOfButton; i++)
   {  form btn_form[i] = new form ();
      ChoiceGroup choiceGroup[i] = new ChoiceGroup ();
      For (NumOfOperations_IN_ ith_Button)
          choiceGroup[i].add ("the i-th Button Operations");
      form[i].append (choiceGroup[i]);
      form[i].addCommand (opForm_OK_cmd);
      form[i].addCommand (opForm_EXIT_cmd);
      form[i].setCommandListener (this);
      add choiceGroup[i] into the i-th button operation form - form[i];
      add all button operation forms into mainList; }
6. Send HTTP Commands according to the command table.
   Protected void sendCommand (String command)
   {  set url = "http://140.113.208.118:8080/MyWeb/ResponseTest?message=";
      set url = url + command;
      call readyConnect (); }
   Public void readyConnect ()
   {  connectThread = new Thread (this);
      Try { call the HTTP connection thread and execute run () method;
             connectThread.start (); }
      Catch (Exception ex); }
```

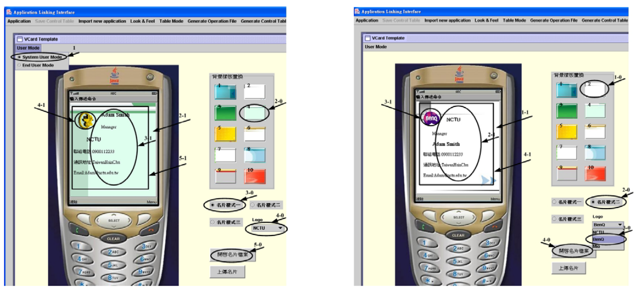
```
Public void run ()
{  Try { send HTTP connection request with a command by calling connect (url);}
   Catch (Exception ex); }
```

After mathematical process of Algorithms 1.1 and 1.2, we fully understand the sample version and class definition of Java application program. So users need not develop remote control program on cellular phone, they can easily and directly get remote MIDlet procedure of cellular phone by analyzing program code through interface generator. In Algorithm 2.1, MIDlet procedure inherits Form class and CommandListener in order to control the whole mobile interface. Lines 1 and 2 take charge of loading relative groupware, such as Button, Radio Button, and ComboBox, and analyzes what kinds of operation project are used in application program. At line 4, it would then generate the main list which belongs to cellular phone interface, such as Button, Radio Button, and ComboBox, which would produce sub-project on cellular phone screen at line 5. Finally, we need HTTP connection procedure to send remote control commands from cellular phone to GUI system, and this is performed at line 6.

4. EXAMPLE

4.1 Example of Generating Interface of Program VCard

If there is a JAVA program 'VCard' running on the PC environment, we can use mouse and keyboard to control the operating functions of this program; the main functions of this program are to edit the name card format with different background, logo, and card format. And the operating procedures are as shown in the Fig. 8.



(a) (b)

Fig. 8. Executing JAVA applications VCard under interface generator system.

In this case, we hope that we can use a cellular phone, which has GUI function automatically generated by proposed interface generator system, as a remote controller to control the JAVA program on the PC environment through HTTP wireless network. The interface generator procedures are separated into multiple steps as depicted in Figs. 9-11:

Step 1: Loading Target Application

The first step of loading a JAVA application into an interface generator system is to choose a desired application for the proposed system. After that, the target application is executed under the interface generator environment and laid on the top of the application's GUI, as in Fig. 9.

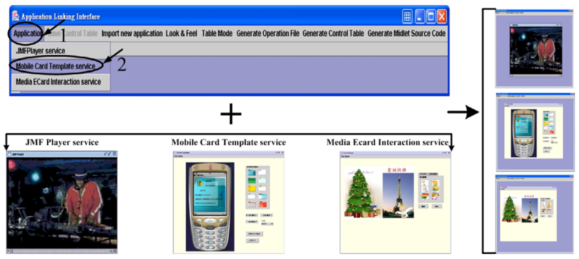
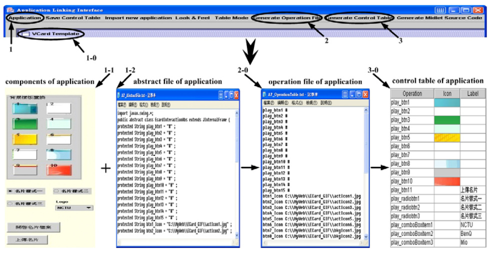

Application Linking Interface
Application
Import new application
Look & Feel
Table Mode
Generate Operation File
Generate Control Table
Generate Midlet Source Code
JMFPlayer service
Mobile Card Template service
Media ECard Interaction service
1
2
JMF Player service
Mobile Card Template service
Media Ecard Interaction service

Fig. 9. Loading JAVA AP running on the PC into proposed GUI system.

Step 2: Generate JAVA MIDlet file

The second step is to automatically generate MIDlet program of control interface from the control button of chosen JAVA application into cellular phone. After loading application into system, interface designer can just press command button "generate operation file", as stage 2 of Fig. 10, and the system would create operation file of application as stage 2-0 of Fig. 10. In the same way, designer can press command button "create control table", as stage 3 of Fig. 10, and the system would create control table of application as stage 3-0 of Fig. 10. Finally, designer presses command button "generate MIDlet source code", and the system generates MIDlet source code of control interface of application VCard on the cellular phone, as in Fig. 11. Figs. 10 and 11 show the major steps below.


Application Linking Interface
Application
Save Control Table
Import new application
Look & Feel
Table Mode
Generate Operation File
Generate Control Table
Generate Midlet Source Code
VCard Template
1
1-0
1-1
1-2
2
2-0
3
3-0
components of application
abstract file of application
operation file of application
control table of application

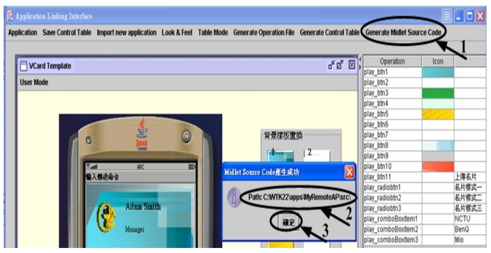


Fig. 10. Flow of generating control table of contrc interface automatically.

Fig. 11. Flow of generating MIDlet code of remote control interface of AP VCard automatically.

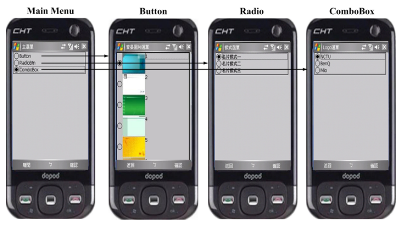


Fig. 12. Generated control interface under different objects view on cellular phone.

Step 3: Download executed file into cellular phone

After downloading and running the executed file in the cellular phone, the control interface of application VCard on the cellular phone under different objects view such as 'Main Menu', 'Button', 'Radio', and 'ComboBox' is shown in Fig. 12.

Step 4: Remotely control application program running on the PC with cellular phone

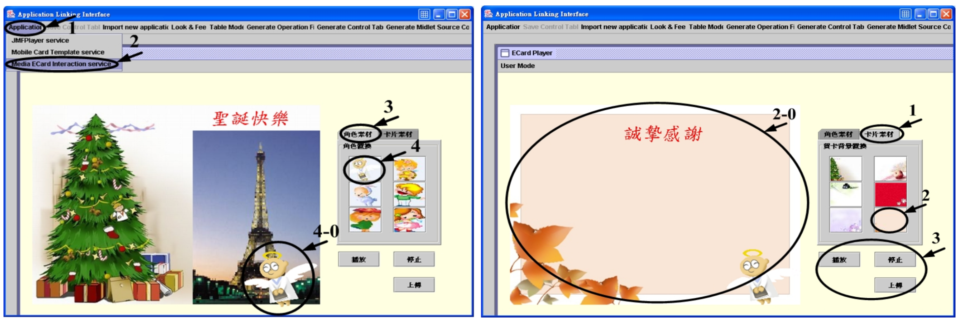
Finally, this cellular phone would become a remote controller; we can use it to remotely control the application program running on the PC through HTTP wireless network by directly touching the command objects, as we have seen, on the control interface of cellular phone, and we need not memorize complex compound commands or complex operating procedures.

4.2 Example of Generating Interface of Program ECard

If there is a JAVA program "Ecard" running on the PC environment, we can use mouse and keyboard to control the operating functions of this program; the main function of this program are to make an electronic congratulatory card with different background and role. And the operating procedures are as shown in the Fig. 13.

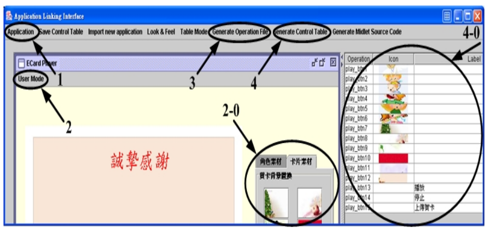
The major steps of generating MIDlet source code of control interface of application ECard on the cellular phone are shown in Figs. 14 and 15 below.

After downloading and running the executed file in the cellular phone, the control interface of application ECard on the cellular phone under objects view is shown as in Fig. 16.



(a) (b)

Fig. 13. Executing AP ECard under interface generator system.



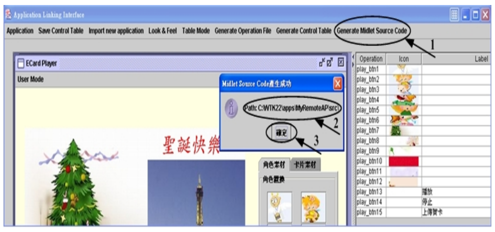


Fig. 14. Generating control table of AP ECard remote control automatically.

Fig. 15. Generating MIDlet code of interface of AP Ecard automatically.

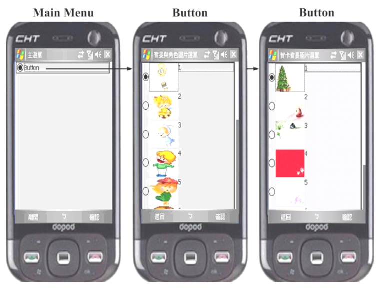


Fig. 16. Generated remote control interface on cellular phone.

5. CONCLUSION

In this paper, we overcome some common problems suffered by developers when bridging a remote control interface system of cellular phone with the JAVA applications running on the PC and have created an interface generator system to generate interface automatically into cellular phone, with which system designer can easily develop control interface of JAVA MIDlet of cellular phone. For more convenience, we define many kinds of template procedures, such as JButton, JRadioButton, and JComboBox, with which system designer can easily and quickly produce a new serve function. JAVA interface generator of cellular phone is the kernel of proposed system. It can parse abstract class written by the application designer, produce an operation script file and control table according to the template procedures, transfer specified control object of Java application system on the PC, such as control buttons and labels, into cellular phone automatically, and then give every operating object in the control table a command ID, exempt from developing remote agreement on cellular phone repeatedly. Finally it produces and compiles the MIDlet program and wraps the result of jar file into cellular phone.

The control interface also combined touch panel and key press operating ability of cellular phone; it can be generated by the interface generator system automatically, even functions of linking and transferring control command protocol. System designer need not define complex macro commands and operating procedures, all the designer has to do is define event procedures when receiving the requested control commands. Through directly pressing interface command object on the touch panel or single keypad of cellular phone, user can remotely control the JAVA application programs running on the PC directly and easily. As we have done, the proposed approach will simplify the development process, abbreviate development time, and make the control system development and modification more flexible and efficient.

We adopt Java Objected-Oriented program analysis based on control patterns as our developing methods, and use that to define the inherited relation and offer the making of the abstract classification to reach the improvement of productivity maintainability. In Table 2, we compare developing application system in traditional method with in our proposing interface generating mechanism in productivity, maintainability, flexibility, and efficiency. We could see that writing the abstract classification to standardize the model of the procedure, offering good encapsulation and characteristic inherited. So it could produce Java and MIDlet procedure automatically for the goal device, the designer needs less to write the relative procedure, especially to make use of GUI to present the application software of the service function, and we could make the development on application interface system more fast and elastic than current designed way.

Table 2. Comparison of interface system development in current method with proposed method.

	Current Java AP Development Approaches	Proposed Interface Generating Mechanism
Productivity	While developing new application program, the procedure designers need to write the whole system procedure on the mobile phone.	While developing new application program, the designers only need to write the abstract classification to offer to the interface system according to the defining model rule of procedure. It could Produce MIDlet procedure automatically by the interface generating system, so the designer does not need to write the interface operation procedure on the mobile phone.
Maintainability	When the designers just want to increase new function or modification, they must write the completely new functions again, and need to write or rewrite the interactive control procedure again, it is difficult to maintain under development.	When the designers just want to increase the new function or modification, they can reuse the written good procedure model again, and do not need to write the interactive control procedure again, it is easier to maintain under development.
Flexibility	While developing the new application program, the procedure designer needs to write the whole system procedure.	While developing the new application program, the procedure designer only needs to write the abstract classification to offer to the interface generating system according to the defining model rule of procedure.
Efficiency	When designer wants to increase new function or modification, they need to write the whole new function mould again.	When designer wants to increase the new function, they can reuse the written good procedure model again.

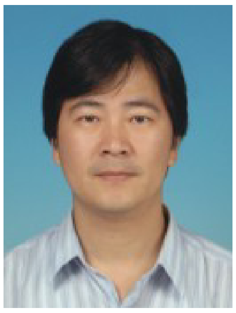
REFERENCES

1. H. Okada, K. Kato, T. Ikegamai, Y. Tatusmi, and T. Asahi, "Proposal of PC remote control system by mobile devices," *Information Processing Society of Japan*, *Sig Notes*, Vol. 2001, 2001, pp. 1-16.
2. N. M. Su, Y. Sakane, M. Tsukamoto, and S. Nishio, "Rajicon: Remote PC GUI operations via constricted mobile interfaces," in *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, 2002, pp. 251-262.
3. J. Nichols, B. A. Myers, M. Higgins, J. Hughes, T. K. Harris, R. Rosenfeld, and M. Pignol, "Generating remote control interfaces for complex appliances," in *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology*, 2002, pp. 161-170.
4. M. Abrams, C. Phanouriou, A. L. Batongbacal, S. M. Williams, and J. E. Shuster, "UIML: An appliance-independent XML user interface language," in *Proceedings of the 8th International Conference on World Wide Web*, 1999, pp. 1695-1708.
5. D. J. M. J. de Baar, J. D. Foley, and K. E. Mullet, "Coupling application design and

user interface design," in *Proceedings of Conference on Human Factors and Computing Systems*, 1992, pp. 259-266.

6. J. Nichols, B. A. Myers, M. Higgins, J. Hughes, T. K. Harris, R. Rosenfeld, and S. Shriver, "Requirements for automatically generating multi-modal interfaces for complex appliances," in *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, 2002, pp. 377-382.
7. O. Omojokun, J. S. Pierce, C. L. Isbell, and P. Dewan, "Comparing end-user and intelligent remote control interface generation," *Personal and Ubiquitous Computing*, Vol. 10, 2006, pp. 136-143.
8. N. R. N. Enns and I. S. MacKenzie, "Touchpad-based remote control devices," in *Proceedings of Conference on Human Factors in Computing Systems*, 1998, pp. 229-230.
9. A. Khan, G. Fitzmaurice, D. Almeida, N. Burtnyk, and G. Kurtenbach, "A remote control interface for large displays," in *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, 2004, pp. 127-136.
10. E. Schwalb, "Synopsis − Books and software iTV handbook: technologies & standards," *Computers in Entertainment*, Vol. 2, 2004.
11. A. K. Dey and G. D. Abowd, "Towards a better understanding of context and context-awareness," *Lecture Notes in Computer Science*, Vol. 1707, 1999, pp. 304-307.
12. S. J. Peng, J. K. Ruzicka, and D. J. Chen, "A generic and visual interfacing framework for bridging the interface between application systems and recognizers," *Journal of Information Science and Engineering*, Vol. 22, 2006, pp. 1077-1091.
13. S. J. Peng and D. J. Chen, "A generic interface methodology for bridging application systems and speech recognizers," in *Proceedings of International Conference on Information, Communications and Signal Processing*, 2007, pp. 1-5.
14. S. K. Huang, "Objected-oriented program behavior analysis based on control patterns," PhD. dissertation, Computer Science and Information Engineering, National Chiao Tung University, Taiwan, 2002.
15. Microsoft Corporation, Universal plug and play forum, http://www.upnp.org/.
16. API specification for the Java2 Platform, Standard Edition, version 1.4.2., http://java.sun.com/j2se/1.4.2/docs/api/.
17. Design Patterns in Java, http://www.fluffycat.com/java/patterns.html.
18. Java TV API 1.1 (JSR-927), http://java.sun.com/javame/reference/apis/jsr927/.
19. J. Nichols, B. A. Myers, and B. Rothrock, "UNIFORM: Automatically generating consistent remote control user interfaces," in *Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems*, 2006, pp. 611-620.
20. T. Ha, J. Jung, and S. Oh, "Method to analyze user behavior in home environment," *Personal and Ubiquitous Computing*, Vol. 10, 2006, pp. 110-121.
21. J. Hess, G. Küstermann, and V. Pipek, "Premote: A user customizable remote control," *ACM CHI Extended Abstracts on Human Factors in Computing Systems*, 2008, pp. 3279-3284.
22. S. J. Lee, Y. H. Kim, S. S. Kim, and K. S. Ahn, "A remote monitoring and control of home appliances on ubiquitous smart homes," in *Proceedings of the 1st International Conference on Mobile Wireless Middleware*, *Operating Systems*, *and Applications*, 2008.

Deng-Jyi Chen (陳登吉) received the B.S. degree in Computer Science from Missouri State University (Cape Girardeau), U.S.A., and M.S. and Ph.D. degrees in Computer Science from the University of Texas, Arlington, U.S.A. in 1983, 1985, 1988, respectively. He is now a professor at Computer Science and Information Engineering Department of National Chiao Tung University, Hsinchu, Taiwan. Prior to joining the faculty of National Chiao Tung University, he was with National Cheng Kung University, Tainan, Taiwan. So far, he has been publishing more than 130 referred papers in the area of software engineering (software reuse, object-oriented systems, and visual requirement representation), multimedia application systems (visual authoring tools), e-learning and e-testing system, performance and reliability modeling and evaluation of distributed systems, computer networks. Some of his research results have been technology transferred to industrial sectors and used in product design. So far, he has been a chief project leader of more than 10 commercial products. Some of these products are widely used around the world. He has been received both research awards and teaching awards from various organizations in Taiwan and serves as a committee member in several academic and industrial organizations.



Shih-Jung Peng (彭士榮) received the B.S. degree in Electronic Engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan, and M.S. degree in Computer Science and Information Engineering from National Central University, Taoyuan, Taiwan, in 1990 and 1994, respectively. He is now a Ph.D. student at Computer Science and Information Engineering Department of National Chiao Tung University, Hsinchu, Taiwan, and he is also a Teacher at Information Management of Ta Hwa Institute of Technology (THIT), Hsinchu, Taiwan. His research interests include e-learning, window's application, performance and reliability modeling.

Chin-Eng Ong (翁浚恩) received the B.S. degree in Computer Science and Information Engineering from Fu Jen Catholic University, Taipei, Taiwan, and M.S. degree in Computer Science and Information Engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2006 and 2007, respectively. He has been involved in the field of Window Mobile software and Java application. Now he is a Software Engineer in Mobile Communication Business Unit in ASUSTek COMPUTER INC. His research interests include Java application in mobile devices, Window mobile programming, and mechanisms about context awareness.