行政院國家科學委員會補助專題研究計畫成果報告

設計及製作介面產生器用於整合連結前認知辨識系統及後端應用軟體系統

計畫類別：個別型計畫
計畫編號：NSC 97-2221-E-009-062-MY3
執行期間： 97 年 08 月 01 日至 100 年 7 月 31 日

執行機構及系所：國立交通大學資訊工程學系(所)

計畫主持人：陳登吉

成果報告類型(依經費核定清單規定繳交)：完整報告

本計畫除繳交成果報告外，另須繳交以下出國心得報告：
□赴國外出差或研習心得報告
□赴大陸地區出差或研習心得報告
□出席國際學術會議心得報告
□國際合作研究計畫國外研究報告

處理方式：涉及專利或其他智慧財產權，二年後可公開查詢

中 華 民 國 100 年 10 月 20 日

設計及製作介面系統產生器用於整合連結前端認知辨識系統及後端應用軟體系統

The design and Implementation of Interface Interfacing generator for integrating and bridging front-end recognizers and back-end application software systems

計畫編號：**NSC**97-2221-E-009-062-MY3
執行期限：97 年08 月01 日至100 年07 月31 日
主持人：陳登吉交通大學資訊工程系教授

成果報告目錄

設計及製作介面系統產生器用於整合連結前端認知辨識系統及後端應用軟體系統

The design and Implementation of Interface Interfacing generator for integrating and bridging front-end recognizers and back-end application software systems

計畫編號：**NSC**97-2221-E-009-062-MY3
執行期限：97 年08 月01 日至100 年07 月31 日
主持人：陳登吉交通大學資訊工程系教授

一、摘要

中文摘要

在[49]的文獻中指出大部份的應用軟體系統裡幾乎有80%的軟體程式碼是和介面系統有相關的。介面系統的花費是開發應用軟體時必需面對的議題，應用軟體的使用者常以人機介面的好壞來評定該應用軟體的好壞。如何快速並簡化介面系統的開發或修改是一重要的研究主題。一般而言，介面系統可分為人機介面系統和應用軟體及應用軟體之間的整合介面系統。前者是目前較常見的議題，我們已在上期國科會三年期研究計畫有開發一使用者人機介面產生器，而後者是強調在兩種應用軟體之間的整合時所設計的介面系統。本研究針對後者提出一個為期三年的研究計畫。期待能達到有彈性且易延伸的介面之介面系統架構Interface Interfacing System (如圖1 所示)，期待能給軟體開發者在製作應用軟體的介面系統時能有較彈性且易維護的解決方式。此計畫的完成我們將予提供一簡易的橋樑，來協助系統開發者將兩種應用軟體(前端的認知器與後端GUI的應用軟體)整合成以前端認知器為人機介面的應用系統，將可和前期的國科會計畫整體連成一無縫式的軟體介面系統的整體解決方案。對軟體介面將有重要的貢獻。

圖1為一介面整合的interface interfacing system應用例子。此例子之後端為windows 環境內的傷心小棧應用軟體，前端為語音辨識應用軟體。透過此介面系統可將現有的傷心小棧(solitary)軟體透過語音的方式來做人機介面的操作。其他應用軟體若要使用相同語音方式來操控應用軟體亦可。此interface interfacing system 比傳統的方法在製作介面系統時可以快速且不易出錯，同時可減輕介面軟體程式的維護。

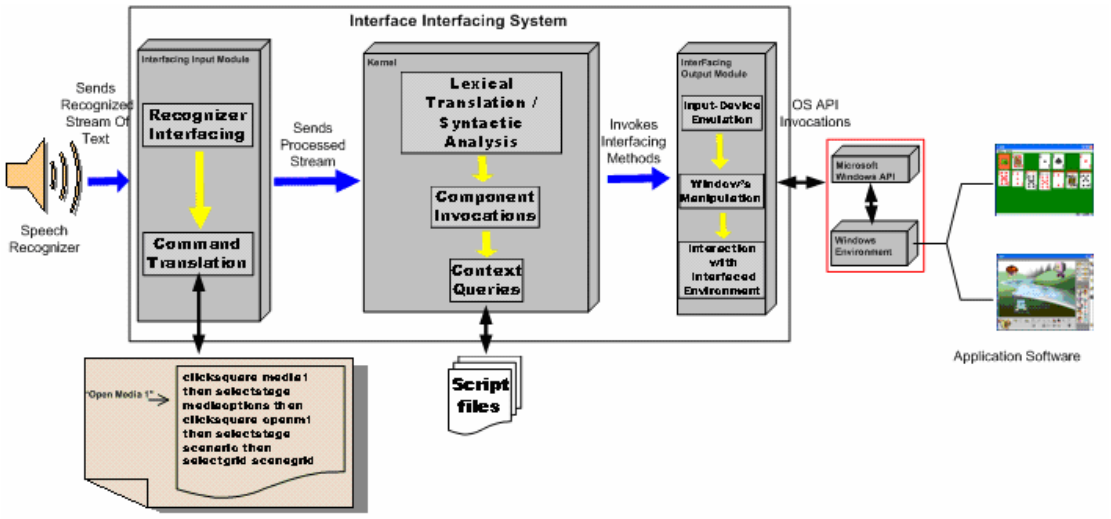


圖 1. The proposed Interface Interfacing System

此Interface Interfacing System，我們已做了先期研究。部份內容已發表在 Journal of Information Science and Engineering 期刊上。本計畫基於此研究成果上，重新作一較完整的規畫並實作完成此理想的架構及系統。

在第一年的計畫裡我們將先期研究成果重新規畫並定義其系統架構，包括前端的語音命令語言(command language)、parser，以及後端的應用程式的介面整合模組。建立基本架構與定義所需的Script 語言及GUI 模組，先以滑鼠進行測試，接著提供一個有系統的方法利用語音辨識操控應用軟體來和滑鼠模組結合以達使用語音來控制滑鼠的互動工作。第一年的計畫已經執行完畢，請參閱下列網址 https://nscnt12.nsc.gov.tw/prQUERY/ShowPDF.asp?url_www=298$296$308$242$297$299$310$297$305$291$247$285$273$246$250$244$253$244$244$265$245$246$246$246$251$253$243$265$243$251$253$243$312$310$307$308$297$310$243$263$275$264$250$249$280$274$243$315$312$242$314$307$299$242$295$311$306$242$246$245$312$306$295$311$306$243$243$254$311$308$312$312$300$6190

在第二年裡我們將把第一年在PC 上開發的觀念架構修整使其能使用到手持裝置上(如PDA、smart phone 等環境)，專注於設計及製作一個應用程式介面載入器用以載入PC 上Java AP 與處理來自手機操作介面程式的控制命令。設計及製作一個手機內Java 程式之介面產生器用以產生手機內Java應用軟體之介面設定使其遙控PC 上相同之應用軟體。前端的認知器改為遙控方式而後端的應用程式則在PC 環境模擬成功後再移植到手持系統的環境。本研究以Java Midlet AP 為例子，透過Smart phone 的手持系統實際遙控應用軟體，圖2 為一各模組間的 Remote Interfacing System 架構。第二年的計畫已經執行完畢，詳細請參閱下列網址 https://nscnt12.nsc.gov.tw/prQUERY/ShowPDF.asp?url_www=266$264$276$242$247$285$273$246$250$244$253$244$244$265$245$246$246$246$251$253$243$26

5$243$252$253$243$312$310$307$308$297$310$243$263$275$264$250$249$280$274$243$315$312$242$314$307$299$242$295$311$306$242$246$245$312$306$295$311$306$243$243$254$311$308$312$312$300$6190

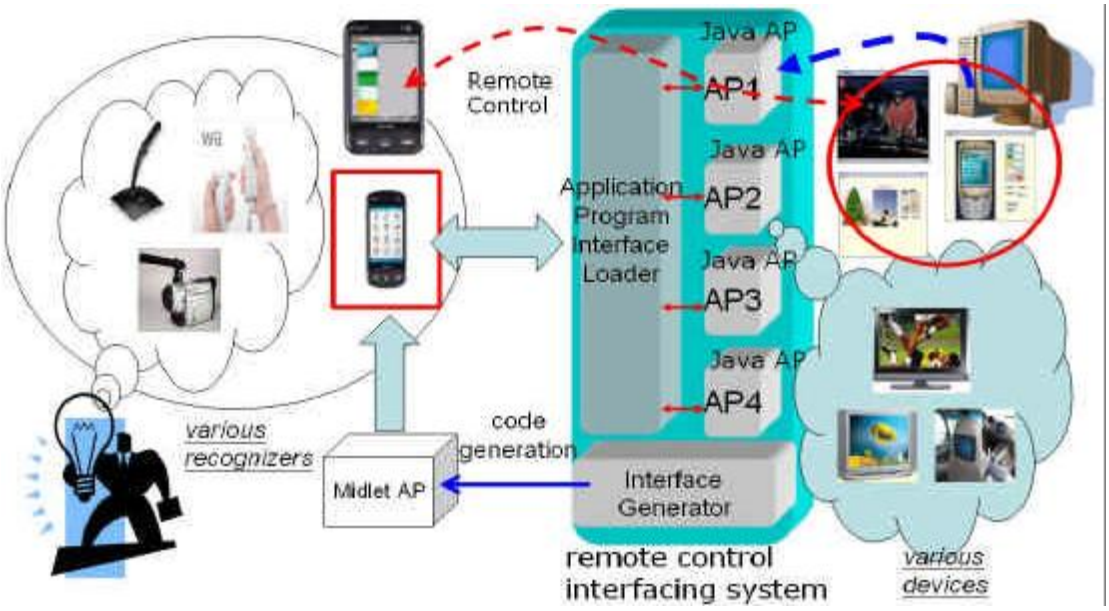


圖2、Remote Control Interfacing System Overview

第三年我們將前兩年的研究成果加以應用到多媒體內容的製作及不同的多媒體播放平台上。大部分的多媒體內容，例如廣告、動畫、簡訊等，都能在各式的平台上播放。如果使用者能藉由一些簡易的操作裝置(例如手機、PDA 等)來與顯示裝置(例如 PC、Digital TV)上的多媒體應用系統進行遠端遙控的溝通，這樣的服務與操作會變得很生動有趣。但由於操作的目標裝置種類很多且控制方式不同，如果要為每一個既有裝置上的多媒體應用軟體，進行遙控互動程式之撰寫或局部功能的新增修改時，勢必要知道每個被遙控的多媒體應用系統之原始碼，才可能為每個互動多媒體系統裝置量身訂作一套遙控溝通的程式。但由於具有互動功能之多媒體系統種類相當多，這樣的開發步驟與方法，非常浪費時間且沒有效率。本期計畫提出一個介面產生器(Interface Generator)，像一個程式碼的剖析器(Parser Generator)的運作功能，能夠為特定裝置上的應用軟體，自動產生手機操作程式，如此，開發者不需了解及撰寫手機內的遙控溝通程式，這樣的機制能夠簡化複雜的開發步驟，而使系統之開發與修改更有彈性。本期計劃運用此介面產生器及其方法實際製作產生三種不同的手機應用軟體為例，使其能和 PC 內相同之應用軟體得以透過遙控的方式來達成互動溝通。透過此實際應用的例子，我們展示此介面產生器及其方法的可行性及應用性。第三年計畫較詳細的資料，請參閱第 10 頁至第 88 頁。

關鍵詞：See-through interface、程式碼的剖析器(Parser Generator)、介面產生器、認知辨識、語音辨識(Speech Recognizers)、軟體工程

英文摘要

It has been shown that the major effort spent on the design and implementation of the application system software is the user interfaces (UI) [49] (or human-machine interface (HMI)). If UI can be developed in a short time, it will be a great help to reduce development time for application software systems. Therefore, many researchers have been seeking better solutions to aid UI designers to create UI systems.

In general, there are two kinds of interface system: human- machine interface and interface for bridging application software as one. The former concerns the GUI design and implementation for the application software. The later concerns with the integration of recognizer and application software together to form a new application software that uses the recognizer as the front-end system. In this proposal research, we layout a three-year integration project that focus on the later interface technology, called generic Interface Interfacing system, Figure 1 depicts the detailed components.

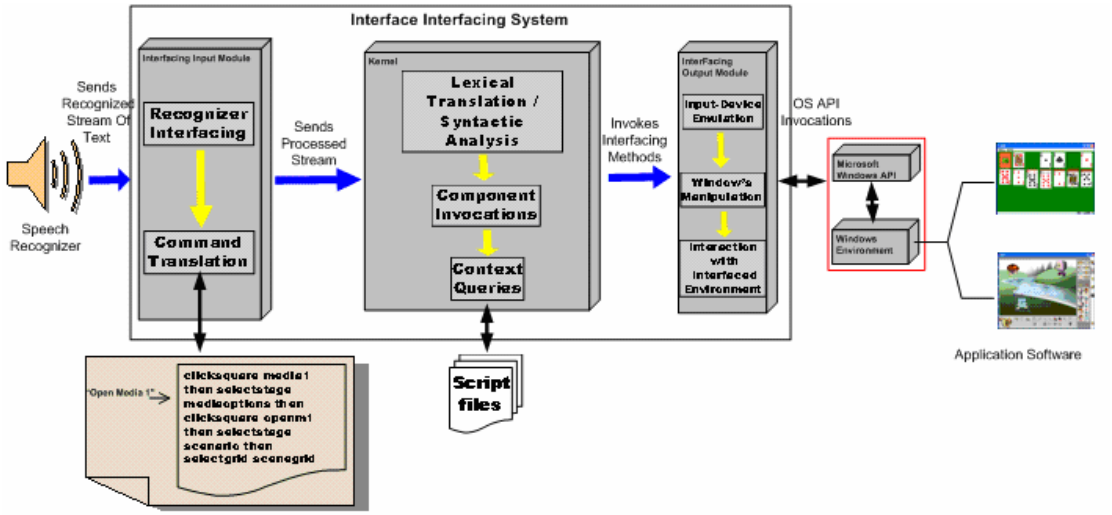


Figure 1、The proposed Interface Interfacing System

Basically, application systems that utilize recognition technologies such as speech, gesture, and color recognition provide human-machine interfacing to those users that are physically unable to interact with computers through traditional input devices such as the mouse and keyboard. Current solutions, however, use an ad hoc approach and lack of a generic and systematic way of interfacing application systems with recognizers. The common approach used is to interface with recognizers through low- level programmed wrappers that are application dependent and require the details of system design and programming knowledge to perform the interfacing and to make any modifications to it. Thus, a generic and systematic approach to bridge the

interface between recognizers and application systems must be quested.

In the first year of this integration research work, we propose a generic and visual interfacing framework for bridging the interface between application systems and recognizers through the application system's front end, applying a visual level interfacing without requiring the detailed system design and programming knowledge, allowing for modifications to an interfacing environment to be made on the fly and more importantly allowing the interfacing with the 3rd party applications without requiring access to the application's source code. Specifically, an interfacing script language for building the interfacing framework is designed and implemented. The interfacing framework uses a see-through grid layout mechanism to position the graphic user interface icons defined in the interfaced application system. The proposed interfacing framework is then used to bridge the visual interface commands defined in application systems to the voice commands trained in speech recognizers. The proposed system achieved the vision of interface interfacing by providing a see-through grid layout with a visual interfacing script language for users to perform the interfacing process. Moreover such method can be applied to commercial applications without the need of accessing their internal code, and also allowing the composition of macros to release interaction overhead to users through the automation of tasks. Figure 1 also indicates an example that a solitary game or an authoring system in window system can be played using the speech recognizer in window system after the integration using the proposed approach.

The main contributions of such interface interfacing system include 1) Productivity is reasonable good: system developers no need to trace the low level code (without requiring the detailed system design and programming knowledge) while integration the recognizer with the application software, 2) Maintenances effort is low: allowing fo r modifications to an interfacing environment to be made on the fly, and 3) Flexibility is good: allowing the interfacing with the 3rd party applications without requiring access to the application's source code. The 1st year project has been implemented, please see the following website. (https://nscnt12.nsc.gov.tw/prQUERY/ShowPDF.asp?url_www=298$296$308$242$297$299$310$297$305$291$247$285$273$246$250$244$253$244$244$265$245$246$246$246$251$253$243$265$243$251$253$243$312$310$307$308$297$310$243$263$275$264$250$249$280$274$243$315$312$242$314$307$299$242$295$311$306$242$246$245$312$306$295$311$306$243$243$254$311$308$312$312$300$6190)

In the 2nd year project, we continue the concept used in the first year to investigate the handheld device environment such as PDA or Smart phone. In this case, we use the remote control capability in the smart phone as the front-end

recognizer and java program as the back-end application software. The choice of the java as the implementation language is rested on its heterogeneous platform adaptation features. Specifically, we will propose an interface generator, similar to the concept of the parser generator, to automatically generate remote control programs for a specific multimedia application in the smart phone. With this generator, designer does not need to write the textual remote control programs in the smart phone. This will simplify the development process and make the control system development and modification more flexible. Figure 2 depicts the detailed components. In Figure 2, it indicates that a interface generator (the interface interfacing system) can proceed to perform a code generation (Java Midlet AP) after the back-end applicatio n software in the PC environment has been integrated with the remote control module using the proposed approach. Of course, the remote control module can be replaced by Wii-like recognizer if it is needed.

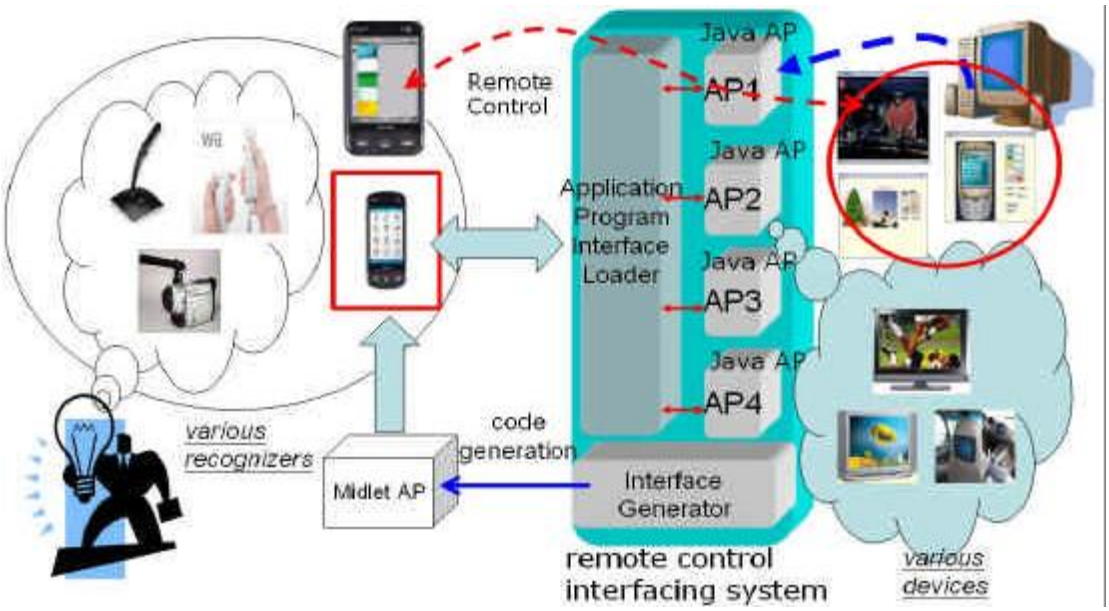


Figure 2、Remote Control Interfacing System Overview

With the quick advance of technology, screen display of digital TV and mobile system becomes more and more elegant and is able to present fine and vivid multimedia contents. Most of the multimedia contents, such as advertisement, motion pictures, messages, etc., can be displayed on different kinds of platforms. If user can use some simple instruments (such as smart phone, PDA, etc.) to remotely communicate with the multimedia application module in the display device (such as PC monitor, digital TV, etc.), then the control becomes live and interesting. But there are various control instruments and display devices, and different kinds of control methods. If one wants to write the control program or partially modify the control

features for the multimedia application module in the display device, then he needs to know the software source code in the multimedia application module that will be remotely controlled, so that he can custom-design a set of remote control programs for each multimedia application. However, there is a lot of multimedia application; a custom design for each of these applications becomes time consuming and less efficient. The 2nd year project has been implemented, please see the following website. (https://nscnt12.nsc.gov.tw/prQUERY/ShowPDF.asp?url_www=266$264$276$242$247$285$273$246$250$244$253$244$244$265$245$246$246$246$251$253$243$265$243$252$253$243$312$310$307$308$297$310$243$263$275$264$250$249$280$274$243$315$312$242$314$307$299$242$295$311$306$242$246$245$312$306$295$311$306$243$243$254$311$308$312$312$300$6190)

Most of the multimedia contents, such as advertisement, motion pictures, messages, etc., can be displayed on different kinds of platforms. If user can use some simple instruments (such as cell phone, PDA, etc.) to remotely communicate with the multimedia application module in the display device (such as PC monitor, digital TV, etc.), then the control becomes alive and vivid. But there are various control instruments and display devices, and different kinds of control methods. If one wants to write the control program or partially modify the control features for the multimedia application module in the display devices, then one has to know the software source code in the multimedia application module that will be remotely controlled, so that he can custom-design a set of remote control programs for each multimedia application. However, there are a lot of multimedia applications, a custom design for each of these applications becomes time consuming and less efficient. This research proposes an interface generator, similar to the concept of the parser generator, to automatically generate remote control programs for a specific multimedia application in the cell phone. With this generator, designer does not need to write the textual remote control programs in the cell phone. This will simplify the development process and make the control system development and modification more flexible. This research demonstrated this interface generator and its algorithm to produce three different kinds of application softwares in the cell phone. These three softwares can remotely communicate with similar application softwares in PC. With this practical example, we demonstrate the feasibility and applicability of this interface generator and its algorithm. For more detail information of the 3rd year project, please refer from page 10 to page 88.

Keywords：See-through interface, Software Engineering, Parser Generator, Interface generator, Recognizers, Speech Recognizers

二、計畫緣由與目的

In a Windows environment, the commonly used traditional method, which allows developed window application programs with Human Computer Interaction (HCI) control ability, is directly to write the control procedures into the application programs while using low-order designing formula to package procedures into single application system. To apply such devising method, the designer must possess certain knowledge about application system designing and programming in order to devise an application system with HCI control functions. Particularly when the design is completed, it is relatively difficult to revise or add any system functions to it without the primitive code, as shown in figure 1.

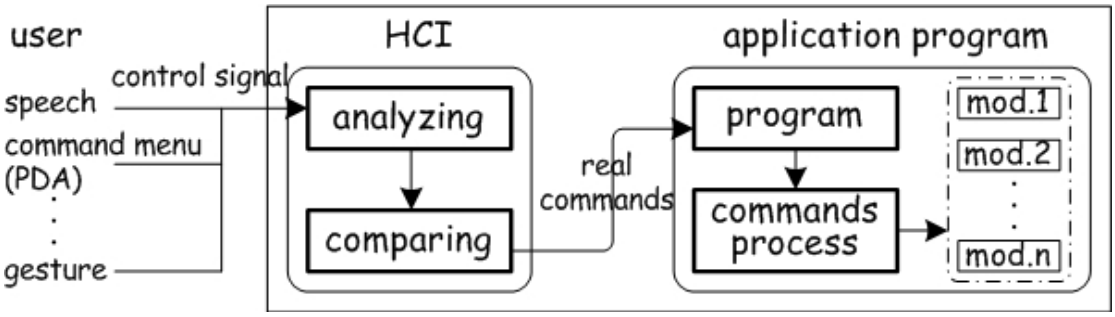


Figure 1 General developing method of HCI control

Three major problems may exist under such development of HCI control procedures. First, system designers must be equipped with abundant knowledge about the design of HCI and programming languages in order to design an application with HCI function. Second, if we want to design an application, which lacked interaction ability before, we need to obtain the primitive code of the particular application due to the difficulty in modifying new programs without the code. Third, even if we have obtained the code, we need to re-analyze the entire structure of application in order to write a suitable control program. These tasks will leave the designer with much trouble and seemingly resulting in less flexibility and efficiency.

To overcome these issues, we emphasize on the research of Software Engineer Methodology to develop a visual generic interface bridge (GIB) system and introducing this system into two parts: First, the “Integration of GIB and Speech HCI,” and secondly, “GIB-based Application Interface (GAI) generation,” in which a PDA device is taken as an example. The GIB provides visual operating interface, under which designers draw recognizing square object at any corresponding position on the windows and name each square object. Subsequently, we can easily use speech command to control mouse and keyboard actions corresponding to the position of square object. By increasing the operation of application with more flexibility and

expandability, we use macro command to define and combine the control commands. One macro command may be combined with several control commands; this will avoid noise effect between long commands and make the application control more flexible with grammar analysis technology. Through this process we can make any application, which did not have HCI control ability previously, with speech or wireless remote HCI control functions in an easier and more efficient manner and do not need to write any program code, as shown in figure 2.

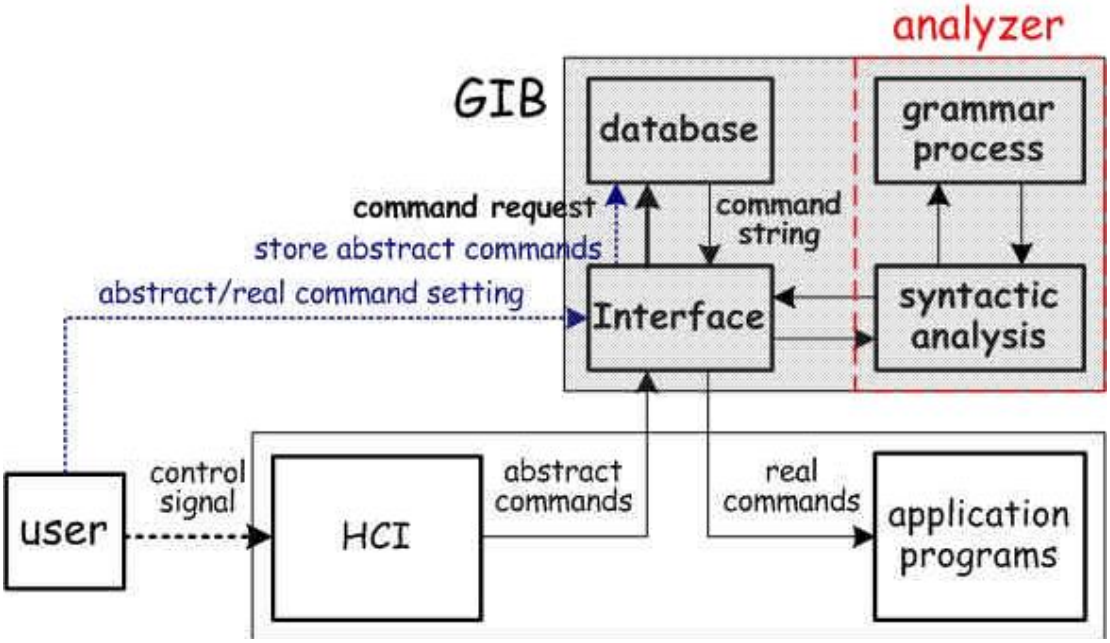


Figure 2 Architecture of GIB control system

三、結果與討論

第一年研究成果

在第一年的計畫裡我們將先期研究成果重新規畫並定義其系統架構，包括前端的語音命令語言(command language)、parser，以及後端的應用程式的介面整合模組。建立基本架構與定義所需的Script 語言及GUI 模組，先以滑鼠進行測試，接著企圖提供一個有系統的方法利用語音辨識操控應用軟體來和滑鼠模組結合以達使用語音來控制滑鼠的互動工作。第一年的計畫已經執行完畢，請參閱下列網址

https://nscnt12.nsc.gov.tw/prQUERY/ShowPDF.asp?url_www=298$296$308$242$297$299$310$297$305$291$247$285$273$246$250$244$253$244$244$265$245$246$246$246$251$253$243$265$243$251$253$243$312$310$307$308$297$310$24

3$263$275$264$250$249$280$274$243$315$312$242$314$307$299$242$295$311$306$242$246$245$312$306$295$311$306$243$243$254$311$308$312$312$300$6190

第二年研究成果

在第二年裡我們將把第一年在PC 上開發的觀念架構修整使其能使用到手持裝置上(如PDA、smart phone 等環境)，專注於設計及製作一個應用程式介面載入器用以載入PC 上Java AP 與處理來自手機操作介面程式的控制命令。設計及製作一個手機內Java 程式之介面產生器用以產生手機內的Java應用軟體之介面設定使其遙控PC 上相同之應用軟體。前端的認知器改為遙控方式而後端的應用程式則在PC 環境模擬成功後再移植到手持系統的環境。本研究以Java Midlet AP為例子，透過Smart phone 的手持系統實際遙控應用軟體，圖2 為一各模組間的Remote Interfacing System 架構。第二年的計畫已經執行完畢，詳細請參閱下列網址
https://nscnt12.nsc.gov.tw/prQUERY/ShowPDF.asp?url_www=266$264$276$242$247$285$273$246$250$244$253$244$244$265$245$246$246$246$251$253$243$265$243$252$253$243$312$310$307$308$297$310$243$263$275$264$250$249$280$274$243$315$312$242$314$307$299$242$295$311$306$242$246$245$312$306$295$311$306$243$243$254$311$308$312$312$300$6190

第三年我們將前兩年的研究成果加以應用到多媒體內容的製作及不同的多媒體播放平台上，例如行動多媒體名片樣板編輯器及播放器、電子賀卡樣板編輯器及播放器，最後考量這些已編輯完成的多媒體軟體內容在大型LCD 及LED 平台上可以撥放的多媒體協調技術，並實際以這些應用軟體為例來驗證所提出的介面整合系統的效益。以下為第三年計畫較詳細的資料。

3.1 Introduction

3.1.1 研究動機

人機介面的操作方式很直接，但隨著現有技術的進步及家用或通訊設備的功能變多，人與裝置間之操作及互動方式，變得越來越豐富。一般來說人們都可以直接與裝置進行各式各樣的互動，可是要利用特定的裝置與被控制的裝置間進行互動或操作，這樣的行為需要撰寫複雜的程式才能完成互動的功能，且針對在這

樣的特定裝置上所開發的控制協定，不易維護，因為若是增加新的操作程序，將導致系統常需要大量的修改，開發上而言成本負擔大、浪費時間且不夠彈性。

本計畫研究的動機，考量手機與 PC 上開發 Java 應用軟體的過程，提供一個手機程式的產生器，來為 PC 上的應用軟體產生出可以在手機上進行遙控的操作介面，因為能夠免除手機程式的撰寫，以及免除重複撰寫應用軟體上的遙控溝通協定，這樣一個有效的機制使其在軟體的製作上，能夠改善其生產力及維護能力。

3.1.2 問題描述與解決方法

進行裝置間互動的方式有很多，一般互動的方法，舉例來說，有以下幾項：

(1) 在數位家庭環境中，採用聲音輸入以控制家電。
(2) 在遊戲主機上，用藍芽連線的方式，透過特殊感應搖桿，來遙控遊戲(如：日本任天堂所開發的 Wii 遊戲平台)。
(3) 利用手機紅外線功能，遙控數位家電(如：Nokia 6708-內建紅外線遙控程式)。
(4) 透過網路進行遠端登入。
(5) 透過網路進行視訊電話操作。

在(1)至(5)這些互動的方式中，比較會遭遇的問題有：

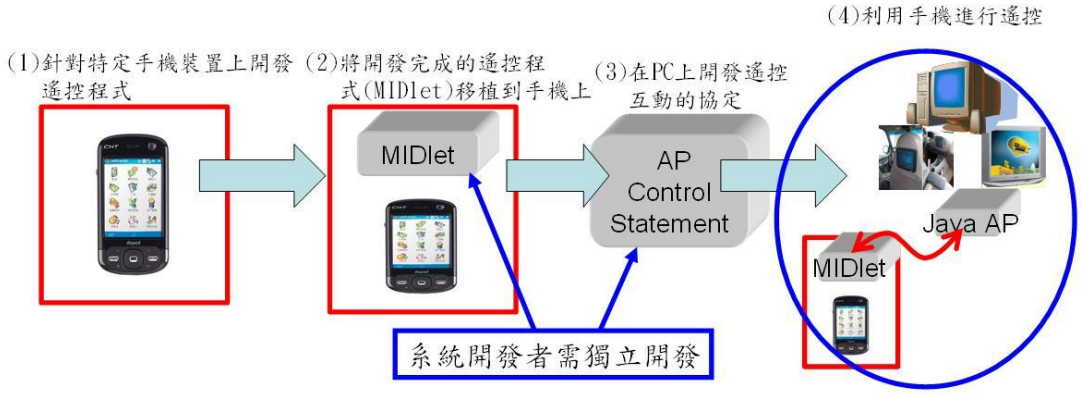
(a) 在特定的互動裝置與被互動的裝置間，開發互動溝通的程式，過程較為複雜。
(b) 需考慮裝置的運作能力以及本身的辨識系統(recognizer)支援，例如手機鍵盤、觸控板，麥克風等可否開發額外的互動控制。
(c) 操作上是否夠友善(friendly)，使用者能夠輕易上手。

依據本計畫的研究動機描述，在手機上進行遙控操作 PC 上應用軟體的機制，可能遭遇的問題就概括以上三點。而我們解決的方式會針對開發互動溝通過程上的簡化，主要是改善第(1)點所遇到的問題，即利用手機程式的產生器來降低複雜繁瑣的開發過程，且由於產生的程式是 Java MIDlet 手機程式，目前所有的手機裝置上都普遍具有支援 Java KVM(K Virtual Machine)平台，所以應用程式能透過手機的鍵盤或觸控板來進行操作，較容易與手機上的辨識系統結合，這樣的方式也能夠解決第(2)點問題，且產生出的手機操作介面與 PC 上的應用軟體操作介面的功能相同，使用者能夠輕易的操作，相對又能解決第(3)點問題。

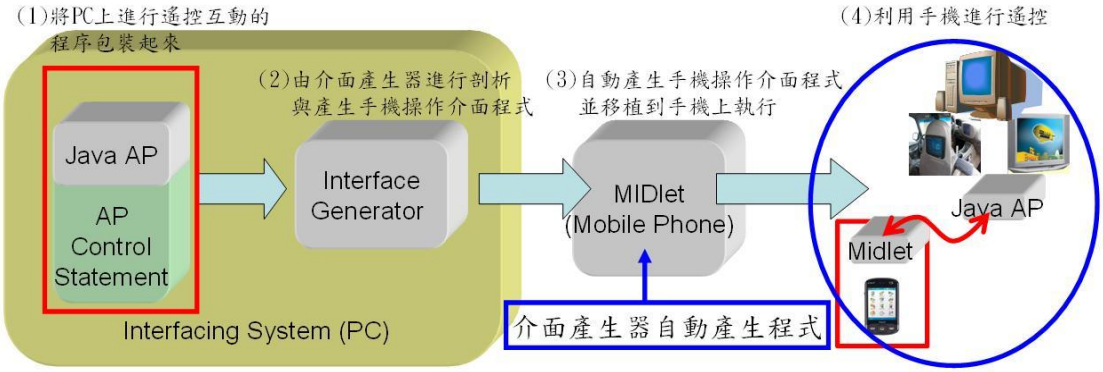
在特定的裝置上，開發遙控互動的應用方式，往往比較沒有彈性，當有新的互動服務產生時，應用軟體需要修改，甚至重新開發，為了能夠與操作控制器上的辨識系統(recognizer)結合，例如，鍵盤或是觸控板，應用軟體中還必須撰寫對應的互動操作功能，這個底層用來處理控制器訊號的程式碼，會因為一個新的應用軟體產生，而重新再被開發過，如此重複的控制程序(control statement)，就常常發生在開發過程中，使得開發應用程式變得沒有彈性及效率。

本計畫所提出的解決方法是利用一個手機內的 Java 程式之介面產生器(Interface Generator for MIDlet Program)，來為 PC 上的 Java 應用軟體產生手機操作介面程式，並能夠與該應用軟體進行遙控的互動，這樣的機制能夠簡化在特定的互動裝置與被互動的裝置間，開發互動溝通的程式複雜度，而這個手機內的 Java 程式之介面產生器是包含在本計畫所提出的遠端控制介面系統架構中，會於第 3.3 節詳細說明。

透過一個在通用介面系統(generic interfacing system)中的手機內的 Java 程式之介面產生器，來幫助產生這些重複的控制程序，並且能夠結合手機的觸控板辨識功能，自動產生手機操作介面的程式，成為一個遙控辨識系統(remote control recognizer)，如此，針對 PC 上的應用軟體以達到手機上的遙控操作介面之開發，就會變得較簡化，甚至不需撰寫手機應用程式，且降低系統開發者開發的負擔。圖 1(a)是一般手機上開發遙控服務的方式，而圖 1(b)是提供介面系統(interfacing system)來開發遙控服務的方式。在圖 1(a)中，手機上需獨立撰寫手機程式(MIDlet)，然後再撰寫遙控 Java 應用程式的控制程序(AP Control Statement)，才能夠完成遙控的服務，如此，開發的程序多且複雜。而圖 2(b)則是在 PC 上提供介面系統(Interfacing System)，而 Java 應用程式的控制程序(AP Control Statement)會自動被產生出來，並透過介面產生器(Interface Generator)，為一個 Java 的應用軟體產生手機操作的介面程式(Java MIDlet Program)，達到遙控互動的服務。如此，系統開發者透過這樣一個通用的介面系統平台，不需要撰寫手機程式(MIDlet program)，也不須重覆開發遙控的互動程序，能將手機操作程式與手機的觸控板辨識功能相結合，而系統使用者則可利用手機與跨平台的應用軟體進行遙控操作。



(a)、一般手機上開發遙控服務的方式



(b)、提供一個介面系統來為 Java AP 產生 MIDlet 手機程式

圖 1、手機上開發遙控服務的方式比較

以上是本計畫提出應用手機內的 Java 程式之介面產生器，為一個 Java 的應用軟體產生手機操作程式的解決方法，能夠解決重複且複雜的應用軟體開發流程，下一節中會針對我們欲進行的解決方法，分析其所要達到的研究目標有哪些。

3.1.3 研究目標

本計畫研究的目標是提供一個遠端控制介面系統架構(remote control interfacing system framework)，該架構的核心系統就是建構出介面產生器(interface generator)，來產生 Java MIDlet 手機程式。本研究目標以手機與 PC 的兩個異質平台為例，並利用 Java 程式語言來實作，開發該系統的目標是根據以下幾點來進行研究：

(1) 一般提供巨集指令(macro commands)的方式，利用手機按鍵的組合來操作一台遠端電腦上的系統介面[1]，這樣的操作方式複雜。本計畫之系統能夠包裝使用者的操作方式，並靜態的賦予每個操作項目一個控制命令(control command)，因此不需定義複雜的操作指令。

(2) 每個特定裝置上的操作項目不同，需要個別提供系統規格的描述文件給介面產生器，方能產生介面[3]。本計畫之系統能夠分析 Java 應用軟體介面上的操作元件，並自動產生描述檔案給手機內的 Java 程式之介面產生器，以產生手機操作的介面程式。

(3) 利用 Java 程式語言在手機或是跨平台的裝置上[16]，開發 Java 應用軟體系統以進行遙控互動的操作，過程複雜，透過本計畫之系統所提供的開發機制，能夠簡化複雜且彈性不佳的互動程序，這樣的開發機制稱之為介面化機制(interfacing mechanism)。該機制規範開發者為 Java 應用軟體撰寫一個特定的

抽象類別，稱之為程式樣板，並利用繼承的方式[27]能夠提供良好的可再利用性(reusable)，系統開發者能繼承該樣板來快速並有效的開發新的 Java 多媒體功能，而轉換出新的應用服務，開發者不需要再對特定裝置上的應用軟體進行重新開發。

本研究的遠端控制介面系統架構包含以下三個系統模組：手機內的 Java 程式之介面產生器(interface generator for MIDlet program)、應用程式介面載入器(application program interface loader)與 Java 應用軟體系統，透過這個架構會達成以上三點目標，來解決開發重複的控制流程與簡化複雜的程式撰寫。

3.1.4 設計理念

本研究的設計理念是開發一個通用的介面系統(generic interfacing system)，該系統提供手機操作介面的產生方法，比一般方式開發 Java 應用程式，能夠節省系統開發者的開發時間與減少程式的撰寫。首先在 PC 上，系統開發者會撰寫數個 Java 應用程式(Java AP，Java Application Program)，以執行不同的服務，而介面系統則提供一個介面(Interfacing Interface)以選取載入其中一個 Java 應用程式，並且能夠結合手機的觸控辨識功能，由介面產生器(interface generator)自動產生手機操作的介面程式(MIDlet AP，MIDlet Application Program)，透過這個手機程式，使用者能夠利用手機與跨平台的應用軟體進行遙控操作。在本計畫的系統實作中，Java 應用程式的執行環境是在 PC 上的 J2SDK(Java 2 Standard Edition Development Kit)平台中運作，所產生的手機 MIDlet 遙控程式，是用來操作 PC 上的 Java Content，由於 Java 具有跨平台的特性，目前都能夠結合數位機上盒中 MHP(Multimedia Home Platform)的 Java 規格(Java AWT/JMF Lite/Java TV)，以及其他相容的嵌入式系統裝置，因此未來都能在這樣的裝置上將介面系統的運作導入並且移植，如此要在跨平台的環境上去開發手機遙控的服務，就又能簡化許多複雜的開發流程。

由於該系統中提供一個手機內的 Java 程式之介面產生器(interface generator for MIDlet program)，能夠為 Java 應用軟體產生手機操作的介面程式，且利用 Java 物件導向的程式設計去規劃程式樣板，對 Java 應用軟體的開發上，提供良好的重覆開發(reusable)特性，因此針對手機以及 PC 上的 Java 軟體開發而言，程式設計師能夠簡化程式的撰寫，並提昇軟體製作上的生產力與維護能力，這就是本研究主要的設計理念與提供的貢獻。

3.1.5 研究步驟

本研究必須先了解製作一個遠端控制的方法，及如何利用手持行動裝置來進

行遠端遙控操作，並利用裝置上辨識器(recognizer)的操作，制定對應遠端系統的操作(keypad mapping)方式，並對目前手持行動裝置上的介面設計(interface generation)評估，進而規劃出整個系統架構、需求及實作。初步可以分為下列幾個步驟:

(1) 了解現存系統、及文獻探討:

本計畫會先介紹現存遙控操作的應用。以 Rajicon System[1]而言，該系統架構是藉由一個有限的行動裝置介面，對遠端 PC 的 GUI(graphic user interface)來進行遙控操作，該系統利用手機上的鍵盤辨識器(keypad recognizer)與巨集指令定義每一個操作，來操作遠端的 PC 介面，取代 PC 上的滑鼠操作。此外還會介紹個人化通用控制器系統 PUC (personal universal controller)System[3]，該系統藉由下載功能描述的規格文件來設計與產生手機呈現介面，在 3.2.2 到 3.2.4 節會說明本計畫之遠端控制介面系統與上述系統之不同處，另外在 3.2.6 節會進行 J2ME(Java 2 Micro Edition)環境以及互動電視(interactive TV，iTV)和 Java TV 的討論，以期將來能夠在數位電視平台上發展。

(2) 系統功能分析：

討論過目前現有的技術後，分析可否在現有的 Java Virtual Machine 環境上，搭配手機上的 J2ME 版本，開發多媒體互動的服務。並建立一個能夠和遠端 PC 的 Java Content 互動溝通的機制，規劃出手機上的遠端遙控操作介面(remote control interface)及 PC 上介面系統(interfacing system)的設計。

(3) 系統設計規劃與實作：

清楚目前需要建置的平台後，必須獨立規劃出新系統，依需求來設計相關的 Java 物件與類別，定義相關的套件與功能規格，並提供系統使用者撰寫抽象類別來定義程式樣板，如此能夠快速有效的開發一個 Java 應用軟體。並設計介面系統產生器，能夠幫助系統設計師產生手機操作介面程式，進行遠端遙控多媒體內容的服務。最後針對各個規劃的系統，選定開發工具進行程式撰寫，並同時進行單元測試工作，詳細的系統功能分析及設計實作會在第 3.3 節說明。

(4) 應用的實例與未來應用的評估：

針對手機介面平台，本計畫在 PC 上利用其他研究生(林賢忠、楊博鈞)實際開發了三個 Java 多媒體的應用程式，以提供手機端的使用者來進行實際遙控的操作示範。在第 3.4 節會介紹這三個 Java 應用軟體，並根據軟體工程製作的兩個維度：生產力及維護力，評估本計畫所提供的介面系統來進行應用軟體的開發，與一般的 Java 軟體開發，在這兩個維度上的比較，以及手機應用程式的開發對於生產力與維護力等特定項目上的比較結果，我們會根據質化的比較結果，來分析並評估他們彼此之間的好壞。另外在第 3.5 節結論及未來展望裡，也規劃在其它能夠支援相同規格的 Java 平台上(例如：互動電視)進行移植，而達到跨平台的橫向發展與技術擴增。

3.1.6 相關名詞

本研究相關名詞及參考範圍，敘述如下：

(1)遠端控制介面架構(remote interface interfacing framework)

連結手持行動裝置上的辨識系統與 PC 上的 Java 應用軟體系統的介面架構，此架構是由應用程式介面載入器、手機內的 Java 程式之介面產生器和 Java AP 所組成。

(2)應用程式介面載入器(application program interface loader)

將 Java 應用軟體系統的介面匯入，系統開發者能夠對畫面上的所有操作元件制定巨集指令，並處理手機介面所傳送的命令。

(3)手機內的 Java 程式之介面產生器(interface generator for MIDlet program)

產生手機遠端操作介面程式與操作控制表單。

(4)手機遠端操作介面(remote control interface)

一個手機遠端操作介面的程式，利用互動協定的機制來遙控 PC 系統上的 Java 應用軟體。

(5)操作控制表單(control table)

藉由介面系統產生器，根據操作描述檔案產生控制表單，能夠描述 Java 應用軟體介面上的所有操作元件，並給予每個元件一個控制命令。

(6)操作描述檔案(operation script file)

描述應用軟體上定義的程式樣板與操作元件的項目，介面系統產生器能根據該檔案來產生操作控制表單。

(7)互動協定(interactive protocol)

提供手機遠端操作介面與 PC 上的介面系統進行雙向溝通(two-way communication)，包含一個 HTTP 的 OTA(on the air)連線和巨集指令的傳遞控制。

(8)巨集指令(macro commands)

藉由手持行動裝置上所提供的觸控辨識器系統來傳送指令，這些巨集指令會對應於 PC 上的操作，以觸發控制 PC 上的 Java Content。

(9)手持行動裝置

本計畫採用的手機平台的規格是 Dopod CHT-9100 的 3G PDA Smart Phone，有內建的 Windows Mobile 5.0 作業系統及 Java MIDlet 的應用管理程式。

(10)J2ME 與 J2SE

J2ME(Java 2 Micro Edition)為本研究採用在手機平台上開發 Java 程式的函式庫版本，而 J2SE(Java 2 Standard Edition) 則是開發一般個人電腦上的應用程式，J2ME 需要有支援 KVM(K Virtual Machine)的裝置環境來執行，而 J2SE 則需要有支援 JVM(Java Virtual Machine)的裝置環境來執行。

3.1.7 章節說明

本計畫分成五個章節來說明，第 3.1 節主要是介紹說明以及研究的動機，第 3.2 節則是說明相關的研究、知識背景與文獻的探討，第 3.3 節則是系統架構與實作的部份，並針對應用程式介面載入器(application program interface loader)、手機內的 Java 程式之介面產生器(interface generator for MIDlet program)及 Java 應用軟體三個模組，詳細說明一個遠端控制介面系統架構(remote control interfacing system framework)如何運作，以及強調本計畫的研究與技術特點，第 3.4 節是一些實際開發的 Java 應用服務說明，以及進行系統開發上的評估，第 3.5 節是結論以及未來展望的探討。

3.2 Related Work

3.2.1 現有多媒體互動的方式

隨著多媒體的內容的變化越來越多，許多數位內容的業者就希望能夠提供豐富且生動的服務模式，來給一般普及的家電裝置或行動裝置上的使用者使用，所以使用者與多媒體內容的互動方式就變得越多，互動模式是人們現實生活中不可或缺的溝通行為，例如數位電視便提供了一個互動式服務，而互動數位電視(interactive TV, iTV)就是提供數位電視互動服務的商品。現實生活中，人們需要互動參與，需要商務交易，也需要資訊隨身化，互動電視即是實現此互動式的電視商務與娛樂的重要工具，控制互動電視需要一個遙控器來操作播放的內容，遙控互動的方式常見於各種行動裝置與家電用品中。以往較直接的人機介面操作以及語音輸入操作等互動，比較受限在目標裝置的支援能力，以及不能夠提供跨平台的應用服務，圖 2 是普遍的互動實例：

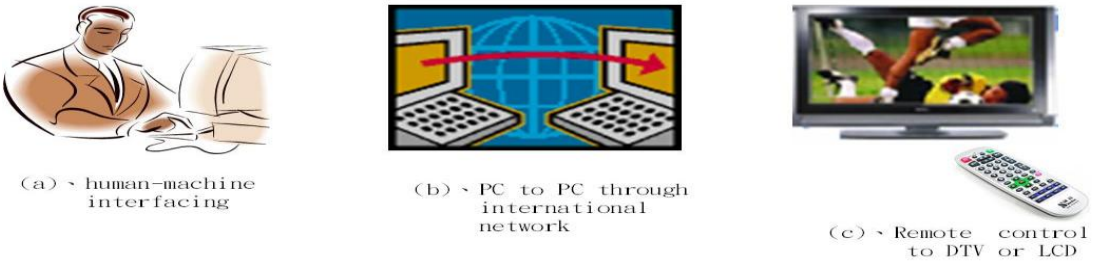
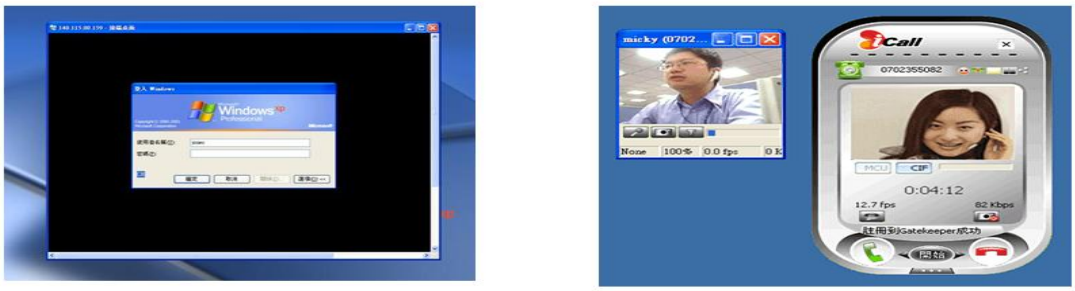


圖 2、普遍的互動方式

現存遠端控制的應用，針對圖 2(a)人機互動服務的例子，現有語音遙控的操作(圖 3)，針對圖 2(c)遙控數位電視的例子，現有如 Wii 的遊戲平台上利用動感式遙桿對電視進行遙控的操作(圖 4)，而圖 2(b)針對透過網路的操作，現有遠端登入或是遠端視訊電話操作(圖 5)：

圖 3、聲音控制

圖 4、遙控互動 wii



（a）、遠端電腦登入操作　　（b）、遠端視訊電話操作

圖 5、遠端操作-透過網路

由於行動計算系統的普及化，人們希望可攜性行動系統的功能越來越多，而逐漸變成希望能夠與週邊的配備同步互動或遠端操作，目前由於手機與 PDA 的使用普及化，甚至一般智慧型手機(Smart Phone)也已經結合了 PDA 的功能，和支援 Window Mobile 的作業系統，在特定的裝置上開發遠端互動與同步遙控的方式就變得十分需要，可是開發控制操作的過程，往往需撰寫許多系統底層之原始碼，才能達到互動，且當新的互動服務產生時，應用軟體必須重新開發。因此，若能結合介面產生器(interface generator for MIDlet program)與手持裝置上的辨識系統(recognizer)平台，使不同之多媒體系統皆能透過此操作平台自動產生特定的互動遙控介面，如此將會使系統之開發與修改變得方便與快速。

3.2.2 現有的遠端遙控機制與 Rajicon System

一般遠端遙控(remote control)的方式很多，在 2001 年有文獻提出利用行動裝置來遙控 PC 系統[2]的方式，作法是利用手機系統來操作 PC 上的圖形用戶介面(Graphical User Interface，GUI)，手機是一台行動裝置具有小的螢幕但不能顯示

相同的 GUI 作為 PC 的螢幕，因此需要分析行動裝置螢幕，來點擊螢幕以遙控 PC 上的畫面，這是一個行動裝置遙控 PC 系統的提議方法。針對遠端遙控而言，client-server 就是一個簡單架構，客戶端(client)透過 OTA(On the Air)連線的方式，送出請求(request)給伺服器端(server)，server 端接到請求後，進行 parsing 並處理這個請求，然後將結果回應(response)給 client 端。這是一種針對特定的裝置所開發的一個互動協定，建立這樣的遠端遙控機制，必須考慮裝置運作的能力、能否支援多媒體的內容呈現以及相對應的遠端操作功能，當目標裝置上進行遠端互動的功能越來越複雜時，控制器介面的設計以及互動的協定，就變得較難去開發。

本研究探討如何透過一個遠端控制介面系統 (Remote Control Interfacing System)與 PC 上的 Java 應用程式(Application Program，AP)，來產生手機操作的介面程式，以便與多元化的多媒體內容互動。該介面系統包含一個手機內的 Java 程式之介面產生器(Interface generator for MIDlet program)與應用程式介面載入器(Application Program Interface Loader)來產生手機操作程式。一旦手機操作程式與手機的觸控辨識系統結合，使用者將能夠利用手機與跨平台的應用軟體進行遙控操作，且透過這個介面系統，也能簡化 PC 上應用軟體的開發程序，表 1 是現行使用手機遙控操作的特色。

表 1、手機遙控操作的特色

	功能特色	搭配環境
紅外線遙控	透過紅外線遙控家用電器與設備。	電視遙控。
WiFi 網路	透過通用隨插即用(Universal Plug and Play，UPnP)技術[23]，各種數位電器、裝置，在連接上網路平台時，可以互相自動搜尋並且連接。	搭配各種數位電器的數位家庭環境。
藍芽連接	經過藍芽連接後，按下手機上的控制按鈕，能操作特定的裝置。	機器人、電子寵物等。

現有的遠端遙控機制很多，本研究以 Rajicon System[1]為例子來做說明與研究探討，該系統架構於 2002 年被提出，主要的功能是能以手機進行遠端遙控操作 PC 上的畫面，手機普遍能夠進行的功能雖然如上表所示，但是比較侷限在特定的目標裝置上，進行特定的操作功能，如果要進行許多複雜的操作方式，就必須定義一套互動協定的規則，Rajicon System 利用巨集指令(macro commands)來制定每一個操作，所以操作的功能越多，巨集指令就定義的越多，且不會不敷使用，這是現有的手機遠端遙控機制與 Rajicon System 的操作機制最主要的不同處，而使用這種方式制定操作，能夠使互動的功能不受裝置侷限。

在第 3.3 節會針對本研究所制定出的遠端遙控機制，來說明如何自動產生出互動控制的協定，以及如何來有效的開發欲互動的多媒體內容，以下 3.2.3 節我們藉由文獻上所記載的 Rajicon System 的運作，來說明如何利用 macro commands 來進行遠端遙控的操作。

3.2.3 巨集指令的制定方式-Rajicon System

採用手持行動裝置，例如手機或 PDA 等來進行遠端遙控的功能，主要會有三個需要考慮的問題：(1)遙控介面(remote control interface)的設計(2)溝通的協定(communication protocol)(3)裝置呈現能力(device capability)，第(1)項的探討會在 3.2.4 節來說明，我們先針對第(2)(3)項來討論，依據相關的文獻記載[1]，圖 6 中的 Rajicon System，以一個透過行動平台的介面來操作遠端 PC 的 GUI 架構，是一個遠端操作的例子，它不同於以往透過網路我們以本地端 PC 對遠端的 PC 進行遠端登入與操作，而是利用手機對遠端的 PC 進行操作與控制，本地端的 PC 以手機來代替。

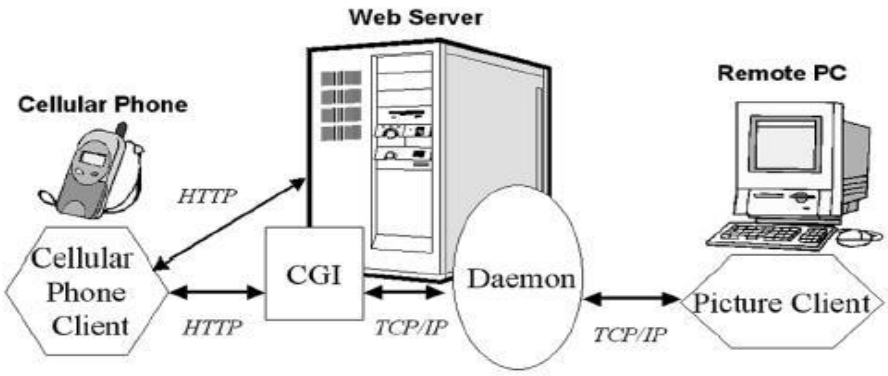


圖 6、Rajicon System Architecture

圖 6 的 Cellular Phone 中有一個 Cellular Phone Client 的模組，它是一個 Java program，負責提供介面的呈現給使用者使用，來進行對遠端 PC GUI 的操作，Web Server 包含了一個 CGI program，來處理接收到來自 Cellular Phone 所傳送的 messages 以及處理 HTTP 的協定，而 Web Server 中的 Rajicon Daemon，是負責保存 Cellular Phone 和 Remote PC 的資訊，處理對應 Remote PC GUI 的座標到 Cellular Phone 的介面上，而 Remote PC 中的 Picture Client，主要是負責完成 mouse/macro commands 的對應，以及產生桌面影像給手機端。

該系統的操作方式是利用手機上的鍵盤識別功能(圖 10)進行 macro/key 輸入(表 2~表 4)，Rajicon System 的溝通協定是利用巨集指令(多種指令)的組合來操作遠端 PC GUI，這樣就能在手機行動系統介面上，對應每個在 PC 介面上的座標來操作，取代了滑鼠或鍵盤的操作方式，圖 7 是進行 zooming、圖 8 是進行 scaling

以及圖 9 是進行 scrolling 的呈現結果。

圖 7、Zoom Movement Mode

圖 8、Scale Movement Mode

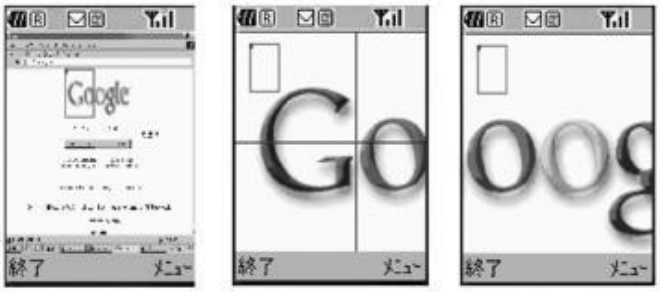


圖 9、Scroll Movement Mode

圖 10、The cellular phone's keypad

表 2、Keypad Mapping

Key	Zoom Mode	Scroll Mode	Scale Mode	Shrink Wrap
1	View entire desktop	Same	Same	Quit Shrink Wrap
2	Dec cursor height	Scroll up/Dec scroll height	None	PgUp
3	Refresh	Same	Same	Same
4	Inc cursor width	Scroll left/Dec scroll width	None	None
5	Execute macro	Same	Same	Same
6	Inc cursor width	Scroll right/Inc scroll width	None	None
7	Dec cursor size	None	None	None
8	Inc cursor height	Scroll down/Dec scroll height	None	PgDn
9	Inc cursor size	None	None	None
*	View previous image	Same	Same	None
0	Fit to window	Reset scroll lines	None	None
#	Zoom in	Toggle scroll lines	Scale out	None

表 3、Special Commands for key Input

Cmd	Result
?a	Hold down the Alt key until the next keypress.
?c	Hold down the Ctrl key until the next keypress.
?e	Press the Esc key.
?t	Press the Tab.
?n	Press the Enter key.
??	Write the '?' character.

表 4、Special Commands for Misc Operations

Cmd	Result
?m	Press the left mouse button at the cursor.
?^	Maximize the window at the cursor.
?_	Minimize the window at the cursor.
?r	Restore the size of the window at the cursor.
?!	Minimize all windows.
?x	Close the window at the cursor.
?*	Adjust the window's width at the cursor such that it can be easily viewed in the cellular phone's display.
?#	Adjust the window's height at the cursor such that it can be easily viewed in the cellular phone's display.
?*n*	Wait for *n* x 5 seconds.

如何用 marco command 來進行對 remote PC 的操作，舉一個例子，假設使用者想要在 PC 上 notepad 中輸入文字：

Hello

Word

根據表 3 則必須在手機的鍵盤上輸入： Hello?nWorld ，所以利用巨集指令的制定就能建立一套溝通的協定。

針對第(2)項，溝通的協定(communication protocol)的探討，Rajicon System 中所使用的 macro command 對應操作會有下列幾種問題：

- 這種 command-based 的 protocol 來當作 interactive protocol 需要定義許多的 macro commands，而利用手機鍵盤來輸入這些 commands，導致操作複雜且不夠彈性。
- 有新的應用軟體產生時，這樣的操作協定又需重新量身訂作，維護力與可再利用性不佳。

因此本計畫提供一個遠端控制介面系統(remote control interfacing system)，能夠自動產生互動的協定，並且利用系統化的方式來為每一種操作制定巨集指令，而該系統又能支援 Java 跨平台的裝置，且針對 Java 應用軟體上的開發，提供良好的可再利用性(reusable)，新的操作功能可以被快速並有效的重覆開發，這樣的方式能夠變換出新的多媒體服務。

至於第(3)項，對於裝置的呈現能力，在 Rajicon System 中 Cellular Phone Client 是一個 Java program，它必須佔用較少的 memory 資源，才能與 server 端互動，並將遠端 PC 桌面影像呈現出來。針對裝置呈現的能力，需要針對使用者對 screen size 的視覺觀感來進行協調，即是進行情境感知(context awareness)的協調，以呈現最佳的畫面，情境感知是指有能力去使用或擷取目標裝置的情境資訊(context information)，是一個能夠顯示關於使用者、載具、環境以及那些可以增加使用者在異質行動裝置上互動的資訊觀念[15]。

3.2.4 遠端遙控介面的製作-PUC System

根據上節所討論，採用手持行動裝置來進行遠端遙控的功能，可能會遭遇的第一個問題就是遙控介面(Remote Control Interface)的設計，因為隨著複雜的週邊配備功能日益增加，在手持行動裝置開發對應的遙控操作介面，就變得相當困難，在 2002 年 Jeffrey Nichols，Brad A. Myers 等人發表了 Generating remote control interfaces for complex appliances 的主題[3]，根據這篇文獻的內容，是提出個人化通用控制器系統 PUC(personal universal controller)的方法，來改良對於複雜設備上的圖形介面(graphical interface)與語音介面(speech interface)，PUC 是一個能夠與日常家用設備連結進行雙向溝通(two-way communication)的系統，由下載一份關於家用設備的功能的規格說明(specification)之後，並能自動的建立一個控制的介面，圖 11 是一個 PUC 系統的架構圖以及藉由網路雙向溝通的方式，圖中 PUC 裝置需要透過網路(network)將目標裝置規格(device specification)、呈現狀態(state)以及控制功能(control)下載，才能產生介面：

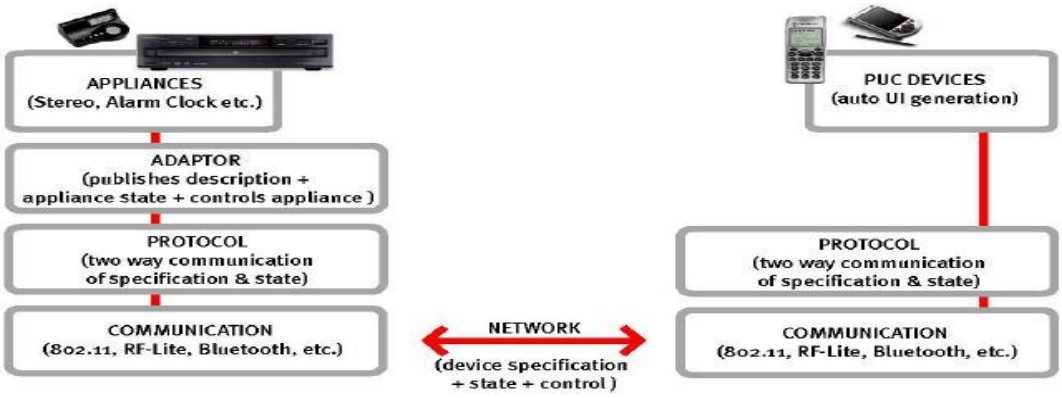


圖 11、PUC System Architectural Diagram

裝置規格的文件根據規格語言(specification language)(圖 12)[4]來描述，PUC 上的介面產生器(interface generator)會分析規格文件，根據這些資訊能建立出一個群組樹(group tree)(圖 13)，而裝置中有相同的功能相依資訊(dependency information)，能夠歸納在同一個 group tree 中，如此能夠根據規格的文件，表達使用者對相關設備功能的了解，而介面產生器則能夠根據此群組樹結構，來比較不同設備上的相依性(dependency)，並建立出合適的操作介面。

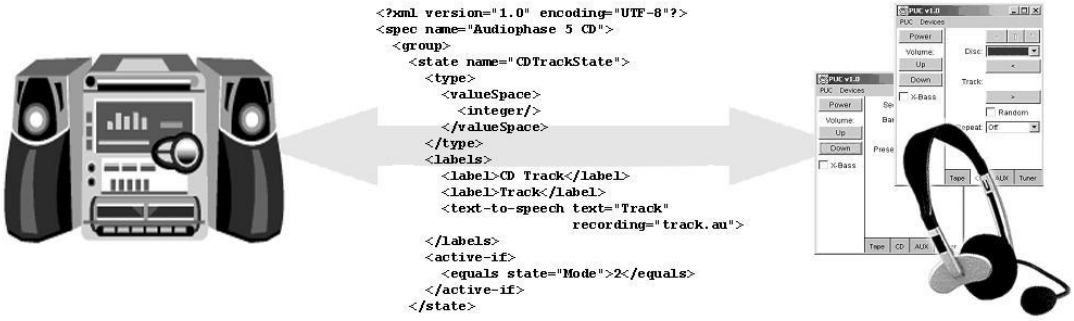


圖 12、XML-based specification language

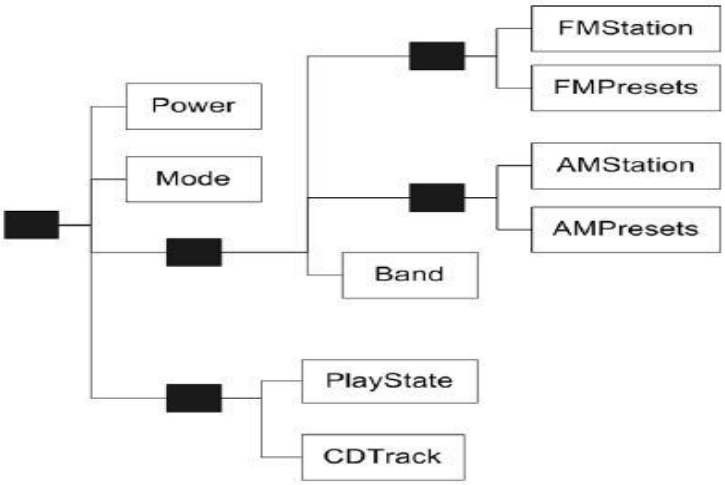


圖 13、A sample group tree for a shelf stereo with both a CD player and radio tuner

圖 13 是描述一個擁有 CD 播放器和廣播頻道調節器的書櫥音響，黑色的方塊是代表群組(groups)，有文字描述的白色方塊則是代表狀態變數(state variables)，而 mode variable 是表示透過喇叭播放出來的來源。利用建立 group tree 來描述 specification 的方式，需要一個決策的演算法[5]來進行分析，主要是利用 decision tree 的決策，根據這個研究的實驗比較結果[6]，用這樣的介面產生方式來建立介面比一般業者在目標裝置上所開發的真實介面來的較佳，介面產生的方法不用考慮外視感覺(look and feel)的設計，僅將功能的描述表現在介面上。另外要為複雜的硬體裝置，自動產生一個遙控的操作介面，需求上要有一些規範[7]，主要就是使用者的設計觀點，以及裝置上的應用能力考量，根據使用者的設計來產生遙控介面的方式[10][12]以及考慮觸控式的遙控裝置能力[9]或大型展示的裝置能力[11]，能總結出使用者的介面需要，結合使用者的設計觀點以及裝置協調特性[8]方能有較佳的呈現與充分表達功能的用途。

總結介面上的設計有許多的因素需要考慮，包括：

- 針對特定的設備，需下載一份功能的規格說明(specification)之後，產生器才能自動的建立一個控制的介面。
- 介面設計的演算法需要考慮使用者需求與裝置呈現，設計的方法繁瑣。

在本計畫研究的範圍中，手機遠端控制介面的設計是開發在 J2ME 的平台上，開發的系統能夠分析 Java 應用軟體介面上的操作元件，並自動產生描述檔案給手機內的 Java 程式之介面產生器，以產生手機介面的操作，如此不需為特定的裝置下載一份規格說明，在本研究中，我們為 PC 上的 Java 應用軟體，自動在手機上產生了手機操作介面的程式，透過手機的觸控與鍵盤識別器(keypad recognizer)能夠遙控操作 PC 上的 Java Content。

3.2.5 手機上的介面產生器之介面產生方式

在本計畫的研究目標中，是建構出介面產生器(interface generator)，來產生 Java MIDlet 手機程式，透過這樣的方式，最主要提供了三個特點：(1)免除重複開發手機上的遙控協定，與減少 PC 上應用軟體的程式撰寫來完成遙控的操作、(2)產生的手機應用程式(MIDlet)能在相容的 KVM 上執行，所以可以在不同的裝置上運作，遙控的裝置不易受限，也易於和不同的辨識器(例如：鍵盤、觸控板、麥克風、照相功能等)結合、(3)使用者不需利用巨集指令定義複雜的操作方式，介面產生器會自動包裝互動的程序，因此使用者透過手機程式的介面進行遙控操作時比較容易。

一般手機介面的產生方法，是需要考慮使用者的外視感覺(look and feel)來設計[7][8][9]，一般來說，製作手機的介面，可能需要介面的製作者利用編輯工具，

來編輯手機操作畫面，並自動產生對應的程式碼，本計畫實作的手機內的 Java 程式之介面產生器，如同一個剖析器(parser generator)，不需要進行編輯手機介面，而是剖析 PC 上的應用軟體之程式碼屬性，來自動產生描述檔案與對應的手機操作程式，因此能夠遙控 PC 上相同之應用軟體，該介面產生器系統是本研究針對所提出的問題描述上，所採用的解決方式，利用上述三種特色來達到研究的目標。

3.2.6 J2ME 與 Java TV

Java 語言是由美國昇陽公司於 1991 年所設計出來的程式語言，主要是讓同一種程式語言所寫出的程式，可以在不同的平台上運作，即所謂的跨平台的技術，對於不同的平台也有相對支援的 API 可以使用。Java 平台在演進到了 Java 2 之後，Java 針對不同領域的需求被分為四個版本：J2EE(Java 2 Enterprise Edition)、J2SE(Java 2 Standard Edition)[24]、J2ME(Java 2 Micro Edition)[25]和 Java Card。J2EE 是定位在伺服端的應用，J2SE 是定位在個人電腦上的應用，這個版本是 Java 平台的核心，其提供了非常豐富的 API 用來開發一般個人電腦上的應用程式，J2ME 則是定位在消費性電子產品的應用上，這個版本針對資源有限的電子消費產品的需求，精簡化了核心類別的函式庫，並提供了模組化的架構讓不同類型產品能夠隨時增加支援能力，Java Card 則是定位在智慧卡的應用上，Java Card 平台將精簡型的 virtual machine 嵌入卡片內，使 Java 程式能夠透過讀卡裝置下載至卡片內執行，如此一來就能延伸智慧卡的功能。這四種平台的關係可以圖 14 來說明，而本計畫所採用的開發環境是 J2SE 與 J2ME 的規格。圖 15 至圖 18 是三種不同 platform 的支援元件[32]

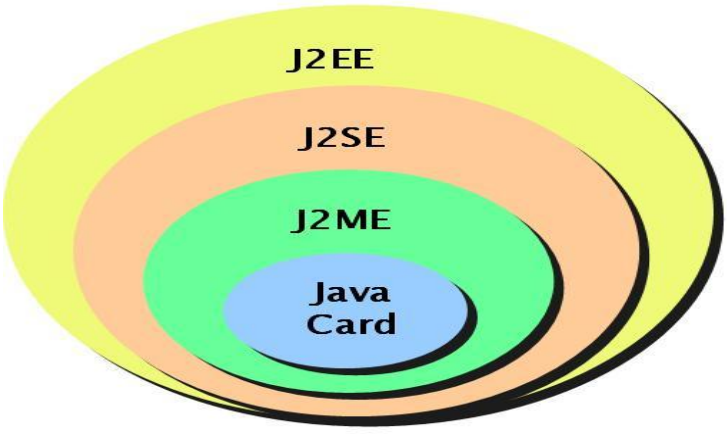


圖 14、各種 Java 平台核心類別函式庫關係

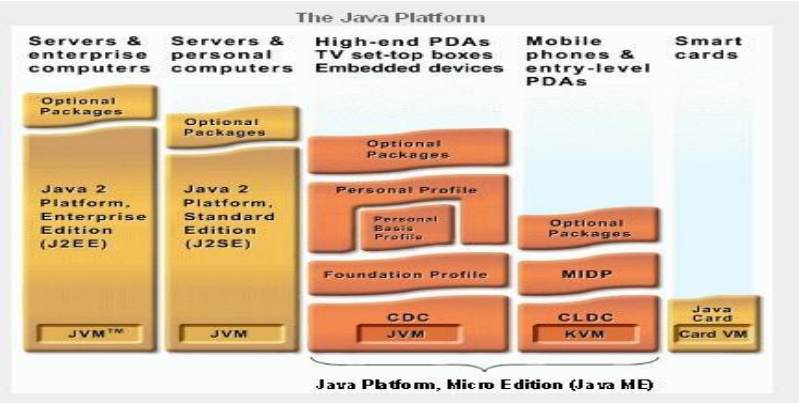


圖 15、The Java Platform

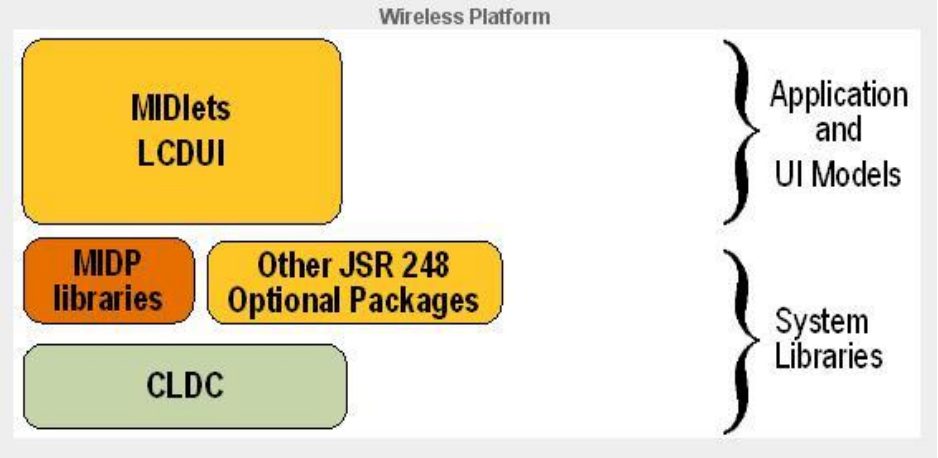


圖 16、The Wireless Platform

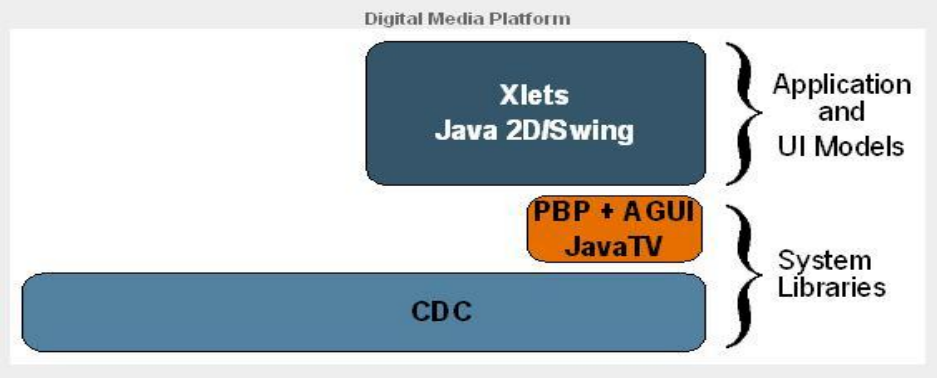


圖 17、Digital Media Platform

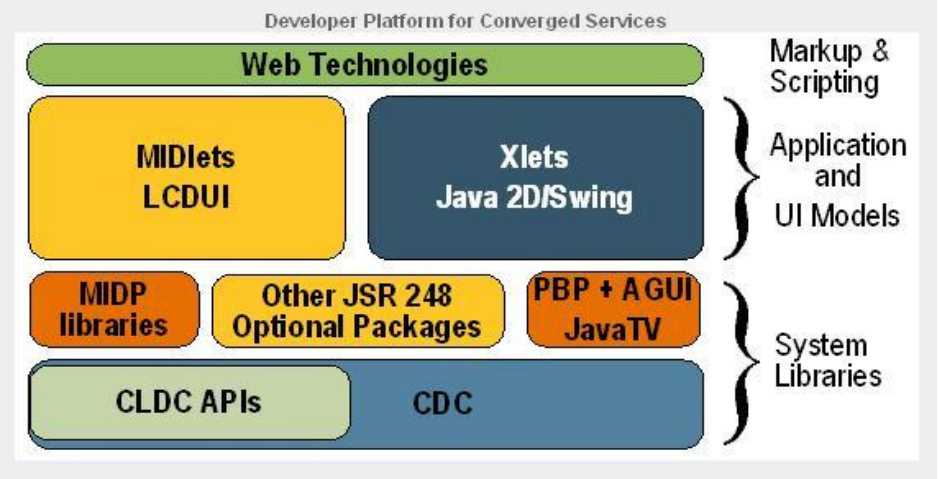


圖 18、Developer Platform for Converged Services

資料來源：http://java.sun.com/javame/technology/index.jsp

在本計畫中產生的手機 Java MIDlet 程式所應用的範圍，是屬於圖 15 中 Mobile phones & entry-level PDAs 所標示的部份，是在連接受限設備配置，即 CLDC(Connected，Limited Device Configuration)的規格下，並有 KVM(K Virtual Machine)的支援來執行。而本計畫在 PC 上所開發的應用程式，是屬於圖 15 中在 High-end PDAs，TV set-top boxes，Embedded devices 所標示的部份，是針對連接設備配置，即 CDC(Connected Device Configuration)組態的規格下，並有 JVM(Java Virtual Machine)的支援來執行。

在網路問世以來，Java 曾一度被計畫用於有線電視的機上盒(settop box)市場，但對於 Java 標準的 API 而言，運用在互動電視上相對過於累贅，早期對於 J2ME 的前身，Personal Java，已有新增了一些電視特有的功能，如影音串流、同步等等，雖然 Personal Java 功能比完整 Java 少了很多，但基本的網路功能還在，所以 Java TV 的實現，可藉由電視台隨著資料流將 applet 傳到電視機，讓觀眾能夠與節目互動，在當時 Java TV 的方案提出過早且與電視營運商的溝通有待商榷，但近年數位電視逐漸盛行，其中互動電視的腳步也已邁開。

互動電視(interactive TV, iTV)[14]係一透過地面無線、有線電視、電信網路或衛星等寬頻網路（broadband network），傳輸數位化（digitalization）的影音（video and audio）與加值服務（value-add services），乃科技匯流（convergence）後之新媒體。利用有線電視的線路伸入每個家庭，加上所謂的機上盒(Set-Top Box)，收視者不再只是被動的接受訊息，而是可以與電視節目互動。 互動電視的機上盒現階段隨著 MHP（Multimedia Home Platform）[28]應運而生，其為一免費使用、開放標準的互動電視中介軟體，提供具有不同數位內容、各種軟硬體的跨平台整合功能，在本計畫第 3.3 節所探討的遠端控制介面系統架構中，未來可以移植到一般具有相同 J2ME 規格支援的 MHP 數位機上盒[21][22]的平台上，目前該系統是以 LCD 的裝置來支援 Java Content 的呈現，但是對於大型數位電視及廣告的

看板有支援相同規格的 KVM，則能實現遠端遙控 Java TV 的技術[13][29][30]，本計畫在第 3.5 節的應用開發實例中，也有一個模擬廣告播放頻道的 Java TV Player，是根據 Java Media Framework[31]開發出的播放器，因應將來能運用在有支援相容 J2ME 規格的互動電視上。

3.3 Proposed Method and Algorithm

3.3.1 遠端控制介面系統架構

在本計畫實作的架構裡，我們夠結合手機觸控辨識功能並以 PC 上應用軟體介面的架構，來產生手機操作程式，這個系統稱為遠端控制介面系統(Remote Control Interfacing System)，系統概觀圖如圖 19 所示：

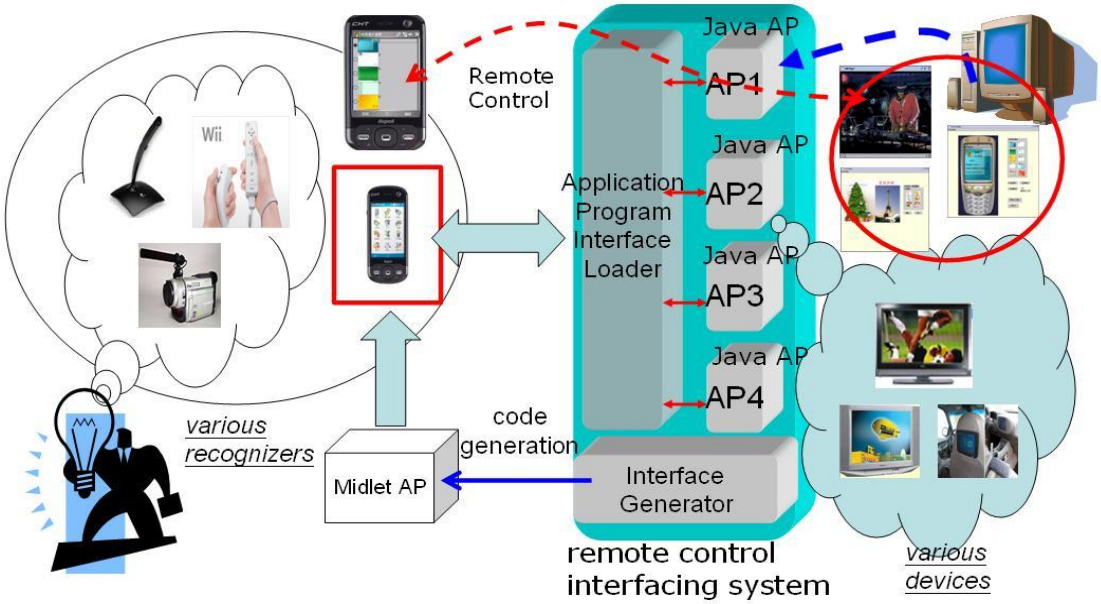


圖 19、Remote Control Interfacing System Overview

該系統是由應用程式介面載入器(application program interface loader)、手機內的 Java 程式之介面產生器(interface generator for MIDlet program)和 Java 應用程式(Java Application Program，Java AP)所組成，本系統係針對 PC 環境上的 Java 應用程式，來進行介面載入的機制。系統開發者在 PC 上所開發的 Java AP，為了能夠與手機遠端操作介面進行遙控互動，必須撰寫底層的 AP 控制程序(AP control statement)，方能處理遠端操作介面所傳送的訊息，並觸發軟體介面上的操作元件。在 3.1.2 節中曾經提及這樣重複的控制程序(control statement)，在開發另一個新的 AP 時，就常常需要被重複納入到開發過程中，變得較沒有彈性。根據圖 19 所示，應用程式介面載入器(Application Program Interface Loader)能夠將

開發好的 Java 應用程式的介面載入，並將需要撰寫的遙控程序(remote control statement)連結起來，如此這些要與手機互動的控制程序，就可以完全委託應用程式介面載入器來產生，之後由手機內的 Java 程式之介面產生器(interface generator for MIDlet program)，進行程式碼產生(code generation)，來為應用程式產生出對應的手機操作介面程式，即是一個 Java MIDlet 程式，部署到手機中即可執行。如此使用者就能夠利用手機上的應用程式與 PC 上的應用軟體進行遙控操作，而這樣的介面架構未來也可以擴增，結合前端更多元的辨識器系統，例如結合 Wii 遊戲平台上的動感式操作遙控，麥克風語音聲控及照相機取像辨識等。而由於 Java 具有跨平台的規格定義，未來遠端控制介面系統也能夠移植到數位電視(DTV)、廣告看板(LCD)以及汽車的媒體控制中心(Car Media Center)，使介面的機制能夠達到多元化的應用。

在本系統中，我們將產生手機操作介面給遠端的操作者，以便對 PC 上的多媒體內容進行遙控。使用 HTTP 的協定來進行連線及傳輸命令，當應用程式介面載入器接收到該命令時會處理並觸發對應的操作元件。本計畫應用實例中，利用 Java 語言開發 PC 上的多媒體程式，成為一個 Java 應用軟體並提供服務，透過一個遠端控制的介面系統(interfacing system)，來幫助該應用軟體產生手機程式。本系統分為兩種使用者，一種為用戶端使用者(end user)，另一種是系統端使用者(system user)，用戶端使用者是指操作手機介面系統的角色，能夠利用手機鍵盤或觸控操作手機畫面上的圖示，並控制遠端 PC 的 Java 程式。而系統端使用者則是操作介面系統的角色，能夠在 PC 端操作相關的 Java 多媒體程式。本系統架構中的遠端介面系統共分為三個模組來建置，即應用程式介面載入器、手機內的 Java 程式之介面產生器和 Java AP，3.3.2 節會詳細說明這三個部份的架構及運作。

3.3.2 系統架構說明

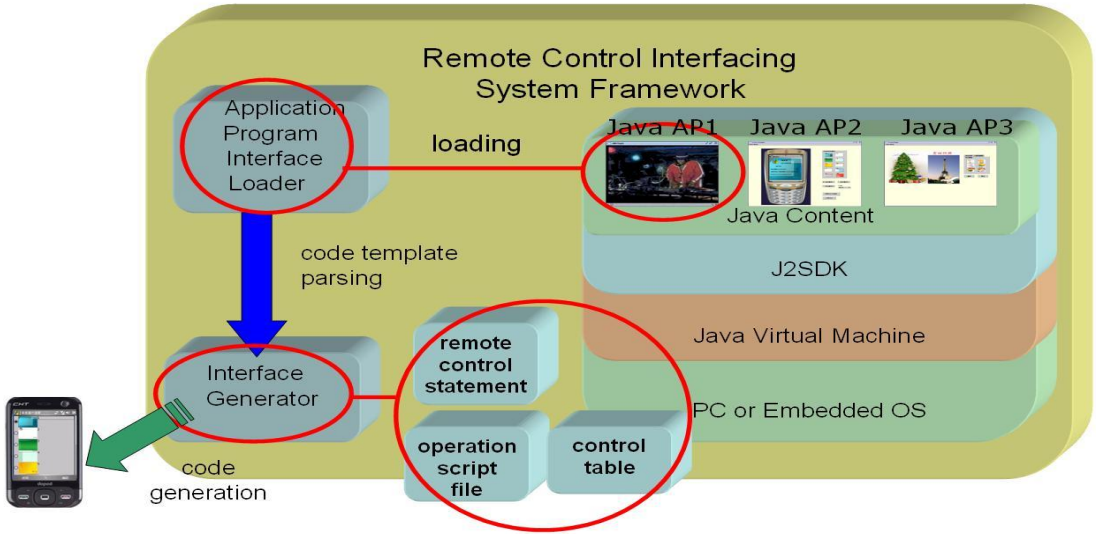


圖 20、Remote Control Interfacing System Framework

圖 20 為遠端控制介面系統的架構圖(Remote Control Interfacing System Framework)，該系統能夠進行連結 Java 應用軟體介面，幫畫面上的操作元件制定巨集指令(macro command)產生控制程序及進行程式碼產生(code generator)。而這些運作過程分別由系統中的三個模組來完成，第一個模組為應用程式介面載入器(application interface loader)，第二個模組為手機內的 Java 程式之介面產生器(interface generator for MIDlet program)或是稱之為介面產生器，第三個模組則是 Java 應用程式(Java AP)。應用程式介面載入器會載入 Java AP 的介面，並包裝控制程序(remote control statement)；手機內的 Java 程式之介面產生器則是根據包裝好的控制程序，操作描述檔案(operation script file)與控制表單(control table)資訊來產生手機 MIDlet 程式；而 Java 應用軟體則是會依據本計畫第 3.4 節的應用實例中，已開發的三個應用軟體來進行說明，分別是 Java 多媒體播放器系統，行動名片樣板編輯系統，電子賀卡樣板編輯系統。以下是遠端控制介面系統架構中各個模組與子模組的相關說明：

應用程式介面載入器(application interface loader)

- 使用者介面(user interface)-載入 PC 上的 Java AP。
- 命令剖析器模組(command parsing module)-處理來自手機操作介面程式的控制命令。

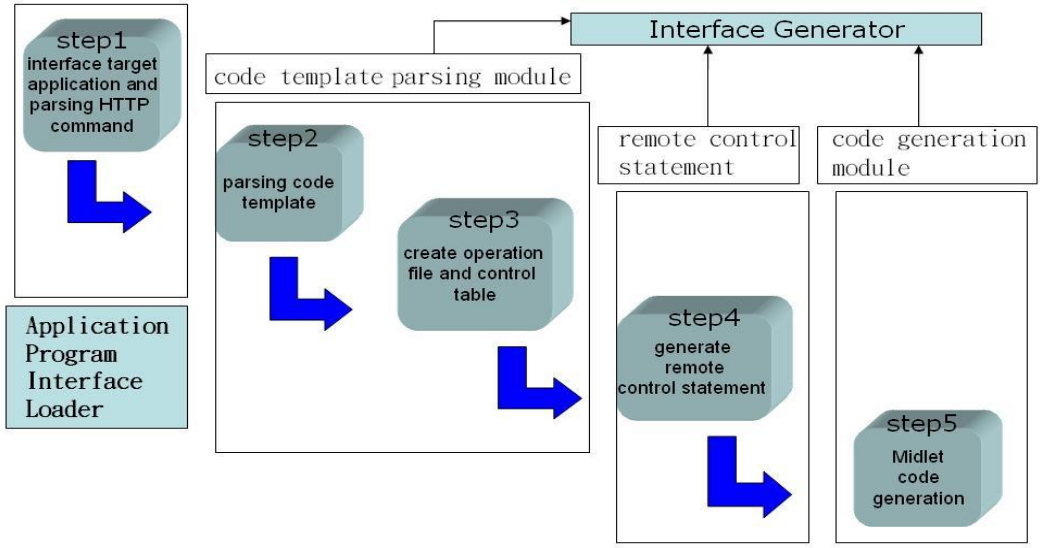
手機內的 Java 程式之介面產生器(interface generator for MIDlet program)

- 程式樣板剖析器模組(code template parsing module)-分析應用程式的抽象類別，根據該類別自動產生操作描述檔案與控制命令表單。
- 遠端控制程序(remote control statement)-處理 http 的連線與控制命令的傳遞。
- 程式碼產生模組(code generation module)-根據操作描述檔案，自動產生 MIDlet 程式與執行 WTK 編譯器。

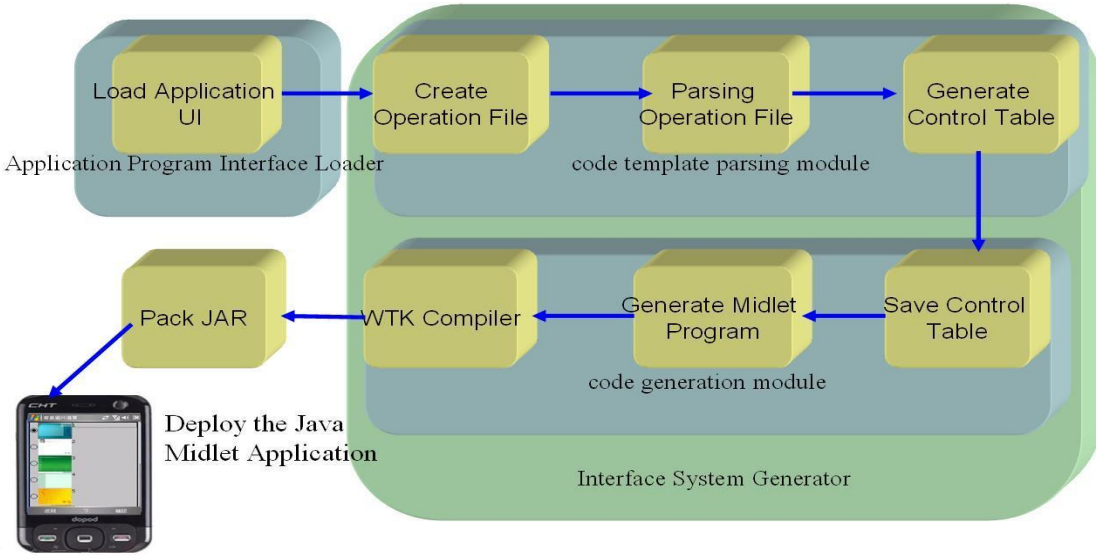
Java 應用系統(Java Application System)

- Java 多媒體播放器(Java Media Player)-能透播放影音多媒體串流的 Java 程式。
- 行動多媒體名片樣板編輯器-進行多媒體名片樣板套用的 Java 程式。
- 電子賀卡樣板編輯器-進行電子賀卡樣板套用的 Java 程式。

3.3.3 系統運作流程



(a)、Procedures for Interfacing a Java Application to a Remote Control Recognizer



(b)、手機介面操作程式的產生流程

圖 21、系統運作流程

遠端控制介面系統(以下簡稱為介面系統)的運作流程大致如圖 21(a)，由 3.3.2 節的架構說明，可以將其正規畫為五個步驟，首先介面系統去載入一個 Java 應用系統(step1)、分析程式樣版(step2)、產生操作描述檔案與控制表單(step3)、產生手機上的遙控程序(step4)與產生出一個 MIDlet 程式(step5)。而將圖 23(a)放大來看，產生一個手機操作的介面程式就如圖 21(b)所示，在介面系統載入完一個 Java 應用系統之後，應用軟體是根據系統開發者所定義的抽象類別來撰寫，這些

抽象類別是規範應用軟體的操作元件及屬性等，有了這個類別來提供程式的樣板，手機內的 Java 程式之介面產生器會分析應用程式的抽象類別，並根據該類別自動產生操作描述檔案(Operation Script File)，之後 parsing 這份文件產生 Java AP 的控制命令表單畫面，其中控制命令是採用系統化的方式，依 UI 上元件數目來給予每個操作元件一個控制命令，之後介面系統儲存這份控制命令表單資訊，並進行 MIDlet 程式的產生，連同這份表單寫出，系統最後會自動呼叫 WTK(Wireless Toolkit)編譯器來編譯產生 MIDlet 程式，編譯完成後，使用者可將程式包裝成 jar 檔，即可放到 PDA 手機上執行手機遠端操作介面的程式，並可以與 PC 上的 Java 應用軟體進行遙控的操作。

3.3.4 應用程式介面載入器

3.3.4.1 目的與功能

應用程式介面載入器(以下簡稱為載入器)是一個提供給使用者的 GUI(graphical user interface)介面，它是一個執行在 J2SDK 平台上的 Java 伺服器程式，系統端使用者(system user)可以利用應用程式介面載入器，來為 PC 上的 Java 應用軟體進行介面的載入，並處理來自手機操作介面程式的控制命令，系統端使用者必須先制定 Java 應用軟體的程式樣板，及提供應用程式給載入器系統，當載入器載入應用軟體的介面之後，會交由手機內的 Java 程式之介面產生器來進行分析程式樣板並產生手機程式。

3.3.4.2 需求分析

應用程式介面載入器主要的功能需求有：(1)連結 PC 上的 Java AP 與(2) 處理來自手機操作介面程式的控制命令。連結 PC 上的 Java AP，是需要系統開發者去撰寫程式的樣板，以及提供程式類別名稱給載入器，如此載入器才能載入對應的 Java 應用軟體，並且委任手機內的 Java 程式之介面產生器來進行分析程式樣板並產生手機程式，本計畫中載入器程式是利用 Java Swing 的套件，繼承其 JFrame 的外觀介面，而在第 3.4 節的 Java 應用實例中，制定的程式樣板是繼承 Java Swing 套件中的 JInternalFrame 介面[26]來規範 UI 的上的操作元件，未來也可以擴增其他 Swing 或是 Awt 的介面套件，以下是應用程式介面載入器提供連結的 Java AP 需求：

表 5、載入器與應用軟體介面元件的需求

載入器介面元件的需求	應用軟體介面元件的需求
javax.swing.JFrame	javax.swing.JInternalFrame
javax.swing.JMenuBar	javax.swing.JButton
javax.swing.JMenu	javax.swing.JComboBox
javax.swing.JMenuItem	javax.swing.JRadioButton

載入器除了連結 AP 之外，它也是一個 Web Server 端的應用程式，需要接收手機端的應用程式，利用 HTTP 的協定來進行連線及傳輸命令，手機端程式與載入器系統會有一份命令表單的記錄，在 3.3.5 節會詳細說明如何去產生控制命令表單，根據這份記錄，當載入器接收到命令時會進行比對，並觸發應用軟體上的操作元件，來達到遠端遙控的功能。載入器是 Web Server 端的應用程式，表 6 是相關的環境平台以及協定需求：

表 6、載入器的環境平台與協定需求

Web Server 的平台	Tomcat 4.1
作業系統	Windows XP
程式開發語言	Java、JSP
互動的協定	HTTP 1.1
編譯器環境	J2SDK 1.4

3.3.4.3 系統設計與實作

由上述的需求分析說明，應用程式介面載入器在實作上，可以利用一個 MVC(Model View Controller)的設計來描述它的功能模組，載入器的介面是用來呈現介面系統的外觀，需要載入的程式檔案就是 Java 的應用程式，而載入器的控制模組，就是進行剖析(parsing)手機端程式所傳輸的命令，圖 22 是一個 MVC 設計圖來描述載入器系統：

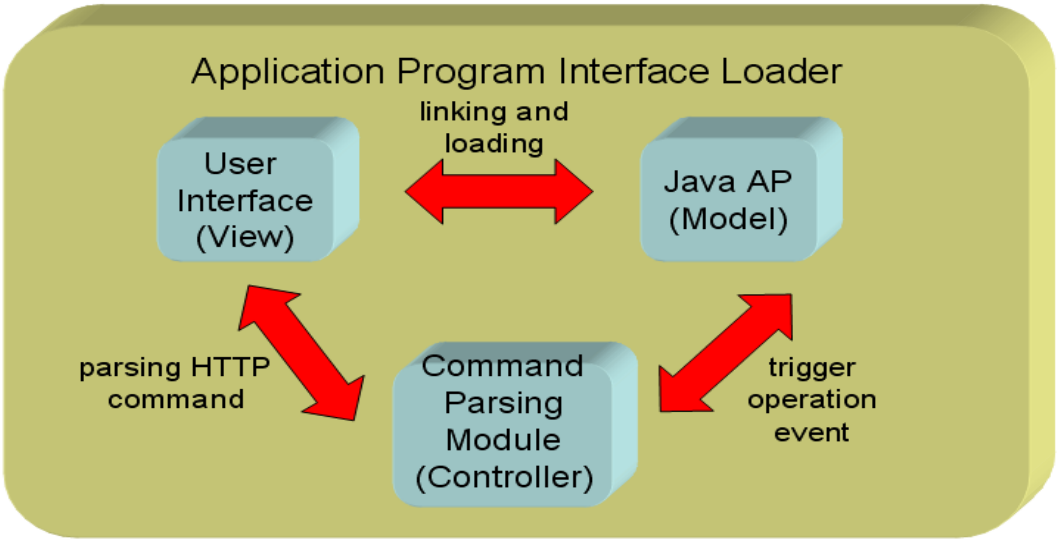


圖 22、應用程式介面載入器-MVC design pattern

應用程式介面載入器主要包含兩個子模組：使用者介面(user interface)、命令剖析器模組(command parsing module)。使用者介面(user interface)是提供介面的功能，能將應用軟體的操作畫面載入(loading)，並連結(linking)所有的操作元件，圖 23 是將第 3.4 節中所舉的三個 Java 應用程式畫面連結進來的結果，圖 24 是載入器系統所呈現介面畫面，在 Application 選項中，可以選擇載入系統開發者已經開發完成的三個應用系統，在這三個應用系統中，開發者已經提供了程式樣板的規範，及提供程式類別名稱給載入器，如此載入器才能載入對應的 Java 應用軟體。

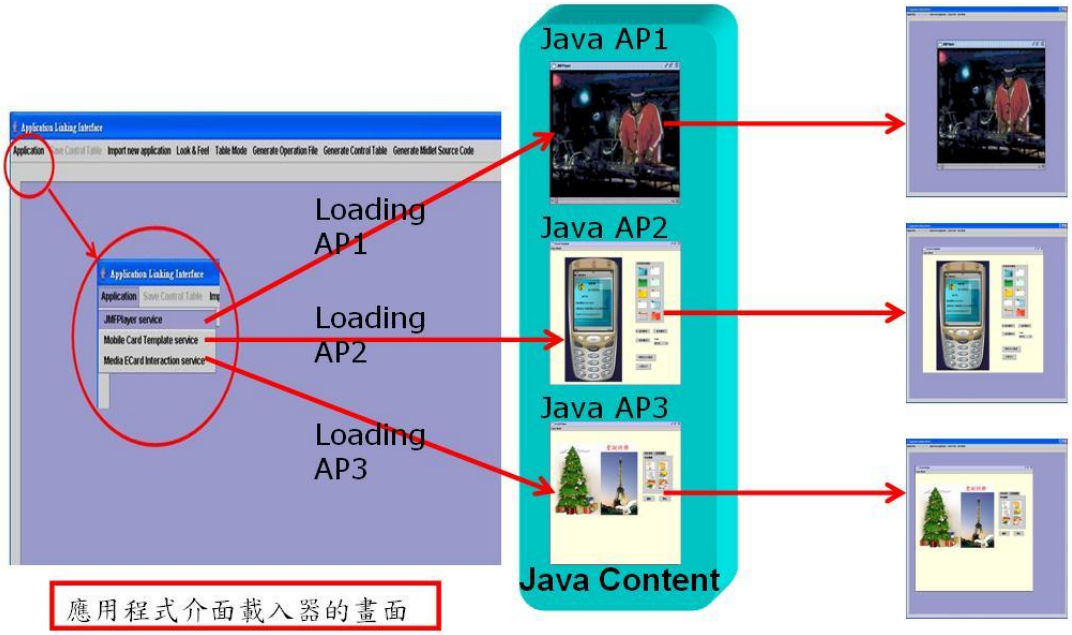


圖 23、連結 PC 上的 Java AP

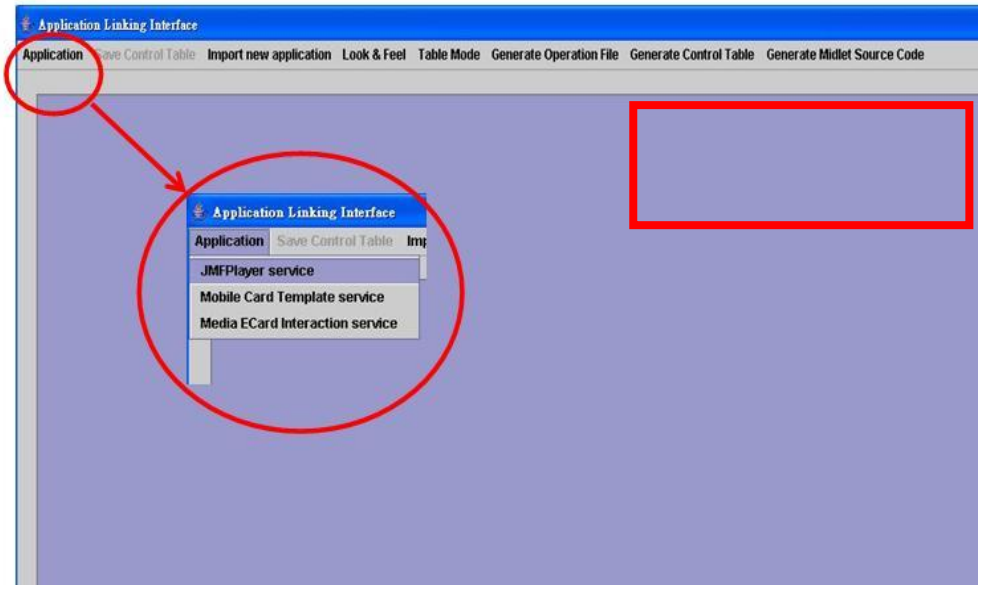


圖 24、應用程式介面載入器畫面中選擇載入的 Java 應用程式

載入器的命令剖析器模組(command parsing module)，是進行剖析來自手機應用端所傳遞的 http command，這個模組在進行剖析時，手機程式端與介面系統端，都會有一份相同的命令記錄表單，這是由手機內的 Java 程式之介面產生器所產生，在 3.3.5 節會說明手機內的 Java 程式之介面產生器如何產生控制命令表單，以及在 3.3.8 節會說明控制命令表單的制定，系統利用巨集指令的方式，賦予每個操作元件一個命令碼(command ID)，有了這份表單記錄，當用戶端使用者操作手機介面程式時，會根據這份記錄傳送 http command，而當剖析器模組接受到 http command 時，會去比對這份表單資料，進而觸發 Java 應用系統上的操作元件(trigger operation event)，如同用戶端使用者在 PC 上操作 Java 應用軟體的畫面，而這樣的操作就改以利用手機來進行遙控，圖 25 是載入器處理來自手機操作介面程式的控制命令。

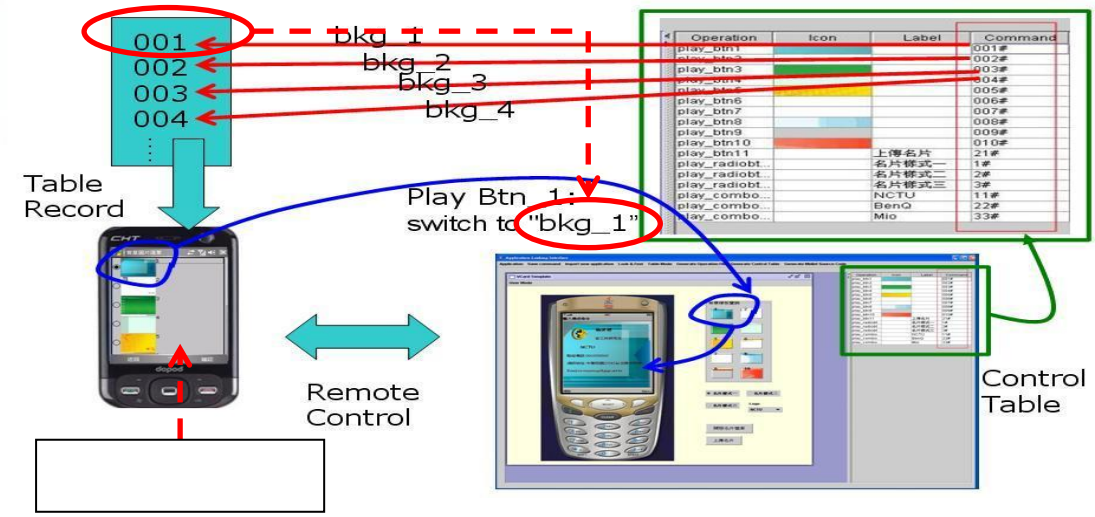


圖 25、處理來自手機操作介面程式的控制命令

當載入器連結一個 Java 應用程式(Java AP)後，它會記錄所有的操作元件資訊，圖 25 是以第 3.4 節中的行動名片樣板編輯器為例，當應用系統的畫面載入進來之後，手機內的 Java 程式之介面產生器會自動產生控制命令表單(control table)，並由載入器來將表單畫面呈現出來，畫面的內容記錄了所有的操作元件資訊，而由手機內的 Java 程式之介面產生器所產生的手機程式(MIDlet program)，也會有這份相同的表單記錄(table record)，記錄對應的操作元件命令碼，如此當用戶端使用者在手機操作介面程式中按下背景 1(bkg_1)的圖示切換(play Btn_1)，對應的命令碼 001 就會傳送給伺服器端的應用程式介面載入器，載入器會進行命令剖析及觸發對應的按鈕操作，將行動名片的背景進行切換。

以上我們歸納使用者介面(user interface)、命令剖析器模組(command parsing module) 與 Java 應用系統(Java AP)的運作：

- 使用者介面(user interface)
    1. 連結 AP 程式。
    2. 載入 AP 的操作畫面。
- 命令剖析器模組(command parsing module)
    1. 處理來自手機操作介面程式的控制命令。
    2. 觸發 AP 的操作元件。
- Java 應用程式(Java AP)
    1. 提供介面外觀與版型呈現給載入器系統。
    2. 提供操作元件的事件處理程序給載入器系統。

3.3.5 手機內的 Java 程式之介面產生器

3.3.5.1 目的與功能

手機內的 Java 程式之介面產生器是遠端介面系統的核心元件，其主要的功能如下：(1)分析應用程式的抽象類別，(2)根據該類別自動產生操作描述檔案，(3)自動產生 Java AP 的控制命令表單畫面，(4)根據操作描述檔案自動產生 MIDlet 程式。抽象類別是須由系統開發者依照程式樣板的規範去撰寫的，每一個需要利用介面系統進行連結的 Java 應用軟體，必須撰寫一個抽象類別，並提供相對應的抽象方法以及實作，將來利用這個抽象類別來當作程式樣板，可以進行重複開發，提供開發新的軟體應用服務。手機內的 Java 程式之介面產生器會先進行剖析(parse)這個抽象類別，並且產生操作描述檔案(operation script file)，操作描述檔案會描述程式樣板上所定義的操作項目，之後產生器會在介面系統的畫面產生 Java AP 的控制命令表單畫面，產生器自動產生 MIDlet 程式，將這個表單以及 MIDlet 程式包裝為一個 jar 檔，就能安裝在手機上並執行。

3.3.5.2 需求分析

手機內的 Java 程式之介面產生器主要負責的功能有以下四個步驟：

1. 分析應用程式的抽象類別。
2. 根據該類別自動產生操作描述檔案。
3. 自動產生 Java AP 的控制命令表單畫面。
4. 根據操作描述檔案自動產生 MIDlet 程式。

第一個步驟要進行分析應用程式的抽象類別，這個類別是用來描述程式的樣板，該類別主要提供一個可以繼承的物件及方法，須由系統開發者進行撰寫來定義。相關的物件成員說明如下 7~9：

表 7、程式樣板採用的開發元件

GUI 類別	javax.swing.JInternalFrame
操作元件型態	javax.swing.JButton javax.swing.JComboBox javax.swing.JRadioButton
操作元件描述	java.lang.String
操作元件抽象方法	protected function

表 8、操作元件項目及說明

操作元件型態	操作元件說明
javax.swing.JButton	提供 UI 上按鈕元件的操作
javax.swing.JComboBox	提供 UI 下拉式列表的操作
javax.swing.JRadioButton	提供 UI 上單一選項按鈕的操作

表 9、操作元件描述的內容

操作元件描述項目	描述的內容
操作元件的標籤	利用 string 描述標籤文字
操作元件的圖示(jpg ,gif)	利用 string 描述圖示檔案路徑

目前本計畫規範出一個常用的 GUI 程式樣板，開發者需撰寫一份抽象類別來描述應用系統的程式樣板，該 GUI 程式樣板中制定較常用的三個 java 操作元件，分別是按鈕元件(javax.swing.JButton)、下拉式列表(javax.swing.JComboBox)及單一選項按鈕(javax.swing.JRadioButton)。開發者在抽象類別中需利用字串(java.lang.String)定義操作元件的描述，元件的描述內容包括：操作元件的標籤文字、操作元件的圖示檔案路徑、及定義每個操作元件對應的抽象方法，提供開發者進行實作。

第二個步驟是根據抽象類別來產生操作描述檔案，產生器會剖析抽象類別所定義的成員及屬性，並產生一個操作描述檔案(operation script file)，這份文件會描述程式樣板上所定義的操作項目，描述的內容如下：

- 操作元件的項目(button string/combobox string/radiobutton string)
- 操作元件的標籤文字內容(label text)
- 操作元件的圖示檔案路徑說明(icon url)

當操作描述檔案產生後，第三個步驟就是自動產生 Java AP 的控制命令表單畫面，這個表單是以多媒體的資料內容來記錄，並呈現在介面系統的連結畫面上，資料內容包括文字描述還有按鈕圖示描述。表 10 是控制命令表單的欄位描述：

表 10、控制命令表單的欄位描述

Operation	操作元件項目的名稱
Icon	操作元件的按鈕圖示
Label	操作元件的文字內容

經過步驟一到步驟三之後，手機內的 Java 程式之介面產生器就能根據控制表單與操作描述檔案來產生 MIDlet 程式，產生器會呼叫 WTK 編譯的模組，進行編譯程式並產生 class 檔，使用者將 class 與資源檔案進行壓縮成 jar 檔，就可以部署到手機上執行，表 11 是 WTK 的環境需求。

表 11、WTK 的開發環境需求

開發平台	J2ME Wireless Toolkit
平台版本	Version 2.2
作業系統需求	Windows XP
作業系統版本	5.1
Java Version	1.4.2_01

系統開發者進行撰寫程式樣板的規範，會在 3.3.6 節詳細說明抽象類別的架構，並且會說明如何利用程式樣板來達到開發新的軟體應用服務，這樣重複利用程式樣板來開新的服務，可以減少開發者的負擔並節省開發的時間，在 3.3.7 與 3.3.8 節會分別說明操作描述檔案的功能以及控制命令表單的制定，最後 3.3.9 節會介紹 WTK(Wireless Toolkit)編譯器的功能。

3.3.5.3 系統設計與實作

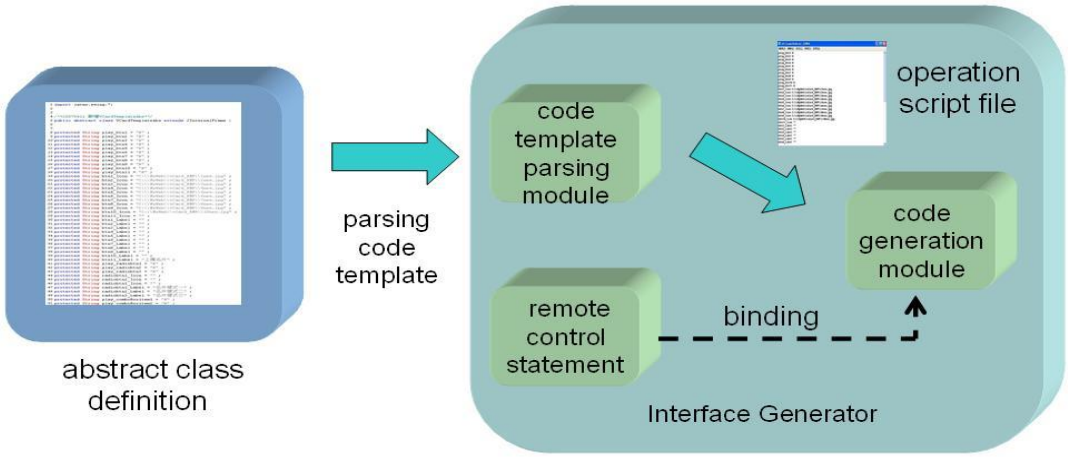


圖 26、手機內的 Java 程式之介面產生器架構圖

圖 26 是手機內的 Java 程式之介面產生器的架構圖，該模組的設計中包含三個子模組，分別為：

1. 程式樣板剖析器模組(code template parsing module)－分析程式樣板產生操作描述檔案與控制命令表單。
2. 遠端控制程序(remote control statement)－處理 http 的連線與控制命令的傳遞。
3. 程式碼產生模組(code generation module)－產生 MIDlet 程式並執行 WTK 編譯器。

程式樣板剖析器模組是進行分析應用程式的程式樣板，也就是依據系統開發者所撰寫的抽象類別進行剖析，之後產生一份操作描述檔案，用來描述操作項目及相關屬性，此時介面系統會在畫面上呈現一份以多媒體資料記錄的控制命令表單，而程式碼產生模組就能根據這些資訊進行 MIDlet 程式的產生，遠端控制程序是由一支 Java 伺服器程式(Java Servlet)所控制，它是一個在伺服器中固定執行的 CGI(Common Gateway Interface)程式，程式碼產生模組進行完剖析後，會將這支 CGI 程式連結進來並且產生對應的手機遙控程序，所以在軟體上需要重複開發的控制程序(control statement)，就可以委任給手機內的 Java 程式之介面產生器來完成，之後程式碼產生模組產生出 MIDlet 程式後，即可呼叫 WTK 編譯器進行程式編譯，使用者就可以包裝程式碼成為一個 jar 檔，並且安裝在手機上執行。

在程式樣板剖析器模組中進行程式樣板的剖析過程中，需要處理兩個階段的工作，第一個階段是分析抽象類別來產生操作描述檔案(operation script file)，操作描述檔案是一份描述應用系統畫面上的所有操作元件(operation components)，如圖 27 的呈現。另外根據這份文件，程式樣板剖析器模組會在介面系統的畫面

上呈現對應的控制表單(control table)。

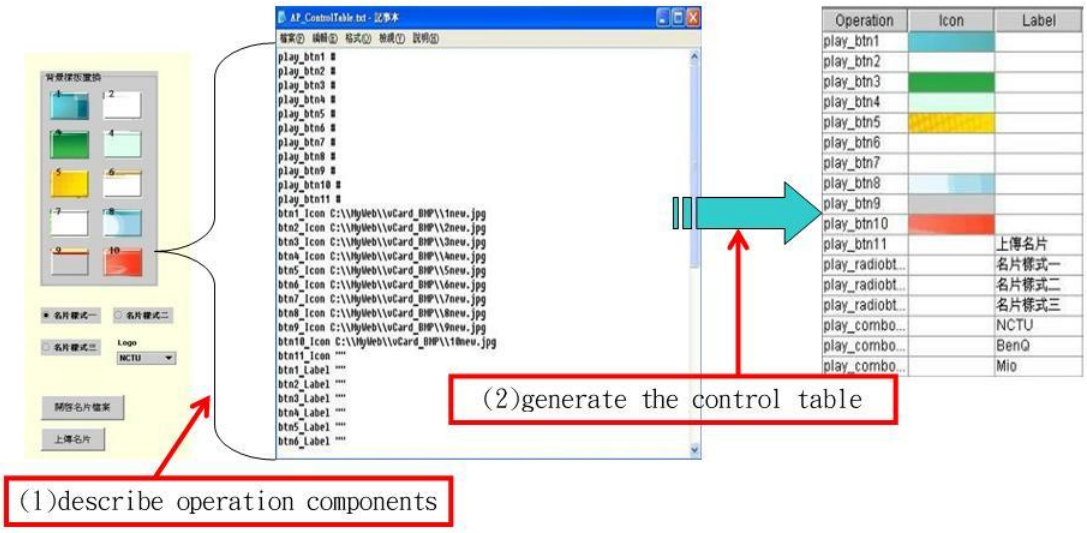


圖 27、剖析器模組的運作流程

第二個階段是轉譯遠端控制程序，而遠端控制程序是一個伺服器端的 CGI 程式，會在進行程式碼產生(code generation)時被程式碼產生模組連結進來。主要的設計功能包括：

1. 繼承一個 HTTP 的 servlet 類別(HttpServlet)
2. 提供 Get 與 Post 的方法(doGet/doPost method)
3. 處理 HTTP 的 request 訊息(HttpServletRequest 物件)
4. 處理 HTTP 的 response 訊息(HttpServletResponse 物件)

遠端控制程序會提供手機端程式與介面系統間的溝通互動，程式碼產生模組會將這段遠端控制程序轉成一支特定程式，提供手機操作介面系統來進行遠端遙控的工作，所以針對每個 PC 的 Java 應用軟體，這份相同的 AP control statement，就可以不需要經由系統開發者在進行開發，只需要處理接收到的命令並呼叫觸發的物件方法即可，如此進行程式碼產生時，可以取代掉重新撰寫控制程序的工作。圖 28 是一個遠端控制程序的主要程式片段說明：

圖 28、遠端控制程序的 servlet 程式

而程式碼產生模組所要設計的功能，就是提供程式碼比對，並且將特定的操作元件，描述轉成一個手機介面上的操作元件，會進行比對抽象類別的屬性及成員，分析抽象方法，轉譯成手機操作元件的事件處理(event handler)，相關的設計功能步驟如下：

1. 分析 abstract 類別名稱資訊(圖 29)
2. 產生視窗標題名稱(Frame Title) (圖 29)
3. 分析操作元件屬性的字串值(圖 30)
4. 提供操作元件圖示資源檔(.png) (圖 31)
5. 產生手機介面的操作元件(圖 32~圖 34)
6. 產生操作元件事件處理程序(圖 35)

在 3.3.4.1 小節，系統端使用者必須先制定 Java 應用軟體的程式樣板，及提供應用程式給載入器系統，系統開發者需要在應用程式介面載入器的程式中，進行程式樣板的資訊設定，所以程式碼產生模組才能根據這些設定資訊，分析抽象類資訊並產生 MIDlet 手機程式，圖 29~35 分別是陳述第 1-6 項的設計內容：

圖 29、分析 abstract 類別名稱資訊與設定視窗標題名稱(Frame Title)

程式產生的第一與第二步驟，是分析出應用軟體所繼承的抽象類別名稱，設定名稱資訊與設定視窗標題名稱，圖 29 所示。

圖 30、分析操作元件屬性的字串值

第三步驟則是需要分析操作元件屬性的字串值，如圖 30 所示，在擷取程式樣板剖析器模組所產生的控制表單的資訊後，這些操作元件資訊會相對的被轉譯成手機上不同的操作元件。

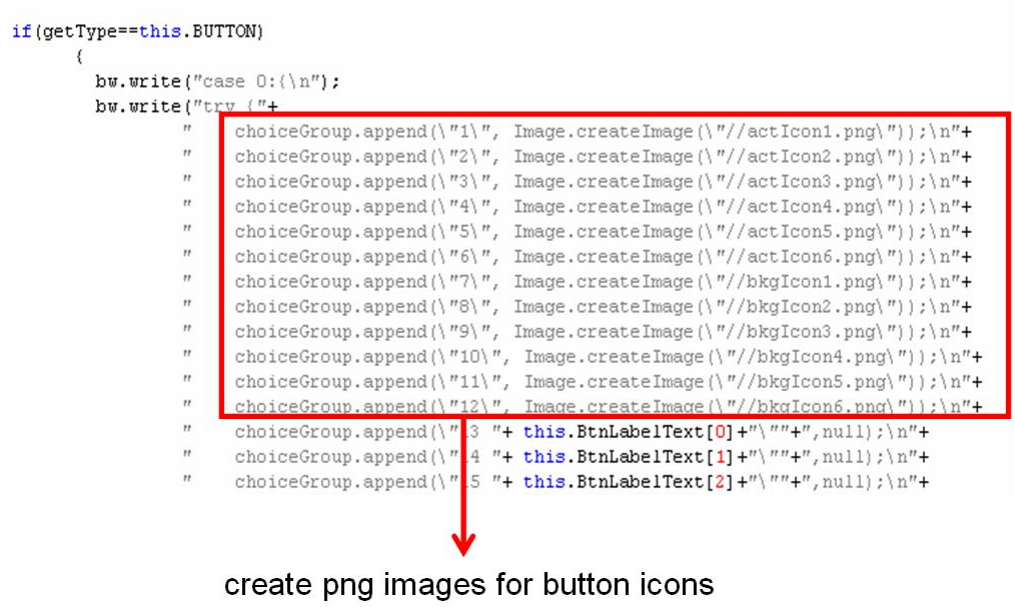


圖 31、提供操作元件圖示資源檔(.png)

第四步驟進行圖示資源檔案的程式產生，系統開發者需提供手機上的操作元件圖示(.png)，如圖 31 所示，手機上的操作元件分成兩種形式，一是以文字標籤描述(label)操作項目，二是以圖示標籤(icon)描述操作項目，在表 8 的控制命令欄位中，對三種操作元件：按鈕元件(javax.swing.JButton)、下拉式列表(javax.swing.JComboBox)及單一選項按鈕(javax.swing.JRadioButton)都有文字及圖示兩種描述，所以在第五步驟中，程式碼產生模組會根據這兩種形式轉譯出手機上的操作元件，如圖 32 至 34 所示：

圖 32、手機介面程式上對應的按鈕(Button)操作元件

圖 33、手機介面程式上對應的單一選項鈕(RadioButton)操作元件

圖 34、手機介面程式上對應的下拉列表(ComboBox)操作元件

在第六步驟中為每個操作元件產生操作元件事件處理程序，如圖 35 所示，手機程式中提供操作元件的觸發事件處理(operation event handle)，根據觸發的形式，會去比對控制命令表單(control table)來判斷命令形式(command type)，最後將命令由 http 請求發送給 PC 上的介面系統。


```
else if(command==opList2_confirmcmd)
{
  InputStream is;
  String CMD_TABLE_TYPE;

  switch(operationList.getSelectedIndex())
  {
    case 0: {

     CMD_TABLE_TYPE = "play_radiobtn1";
     is = getClass().getResourceAsStream("/Vcard_command_table");
     SendCMD(is,CMD_TABLE_TYPE );
      break;
    }
    case 1: {

    CMD_TABLE_TYPE = "play_radiobtn2";
    is = getClass().getResourceAsStream("/Vcard_command_table");
    SendCMD(is,CMD_TABLE_TYPE );
    break;
   }
   case 2: {

     CMD_TABLE_TYPE = "play_radiobtn3";
     is = getClass().getResourceAsStream("/Vcard_command_table");
     SendCMD(is,CMD_TABLE_TYPE );
     break;
   }

  }
```

圖 35、手機程式上產生操作元件事件處理程序

以上是介面系統中的手機內的 Java 程式之介面產生器(interface generator for MIDlet program)製作，在 3.3.3 節的圖 21(a)中，針對連結一個 Java 應用系統到產生手機 MIDlet 程式的五個程序裡，步驟 2 與 3 需要利用到應用程式樣板、操作描述檔案以及控制命令表單，這兩個步驟是由手機內的 Java 程式之介面產生器中的剖析器模組來執行，以下在 3.3.6 節中我們會描述應用程式樣板的分析內容有哪些，3.3.7 節中會描述操作描述檔案的內容，3.3.8 節會說明如何制定控制命令表單，以及 3.3.9 節描述程式碼產生模組中 WTK 編譯器的運作。

3.3.6 應用程式樣板分析

程式樣板是用來描述一個 Java 應用軟體的抽象類別，提供了一些 Java Swing 元件的屬性描述，以及提供一些抽象方法，這個抽象類別是需要系統開發者獨立撰寫與定義，如此手機內的 Java 程式之介面產生器中的程式樣板剖析器模組(code template parsing module)，就能進行剖析該類別，而產生出操作描述檔案與控制命令表單。針對 Java Swing 的元件，在實作本系統，我們定義了其中常用的三種操作元件與一個容器元件：

操作(Operation)元件

- javax.swing.JButton
- javax.swing.JComboBox
- javax.swing.JRadioButton

容器(Container)元件：

- javax.swing.JInternalFrame

操作元件代表了 UI 上的一些視覺化的按鈕或是選單等，在本系統中，我們定義了按鈕(JButton)，單一選項的選擇鈕(JRadioButton)以及下拉式選項的選單(JComboBox)，安排這三種視覺化操作元件排放位置的視窗，就是一個容器(Container)元件，這邊開發者需要繼承一個內部視窗(JInternalFrame)，須把上述三種操作元件包含進來。圖 36 是描述程式樣板的繼承架構，VCardPlayer.java 的執行結果是第 3.4 節的行動名片樣版編輯器的應用實例，它繼承的程式樣板就是 VCardPlayerAbs.java，需由開發者撰寫定義，而所採用的視窗容器就是繼承內部視窗(JInternalFrame)，最後由應用程式介面載入器，即 APLinkInterface.java，它繼承了 javax.swing.JFrame 視窗而將內部視窗包含進來。

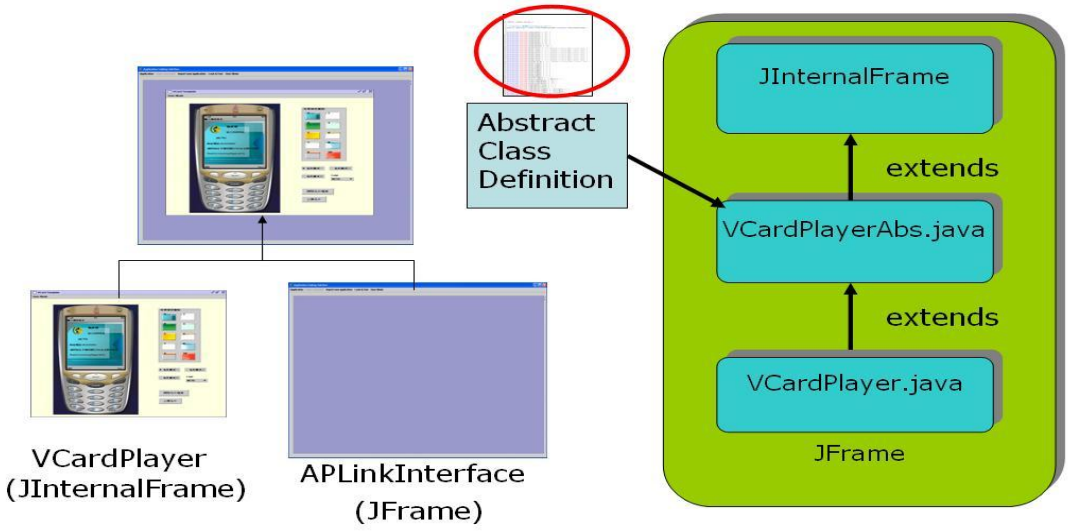


圖 36、程式樣板的繼承架構

從上圖得知 VCardPlayerAbs.java 就是行動名片樣版編輯器的程式樣板，而 VCardPlayer.java 則是進行實作這個抽象類別，圖 37 就是 VCardPlayerAbs.java 的抽象類別程式片段，它繼承了 javax.swing.JInternalFrame 的容器視窗，定義了 JButton、JComboBox 以及 JRadioButton 三個操作元件的屬性，下圖的字串值(String)是描述操作元件的標籤文字及圖示(jpg ,gif)檔案的路徑，這些關係可以用一個 Regular expression 來辨認字串變數，依屬於哪種型態之 token，而加以分析：

```
Identity :        [a-zA-Z_][a-zA-Z_0-9]*
Integer number：[0-9]+
String : \"([^"\n]|\\["\n])*\"
Keyword : "protected" | "String"…
```

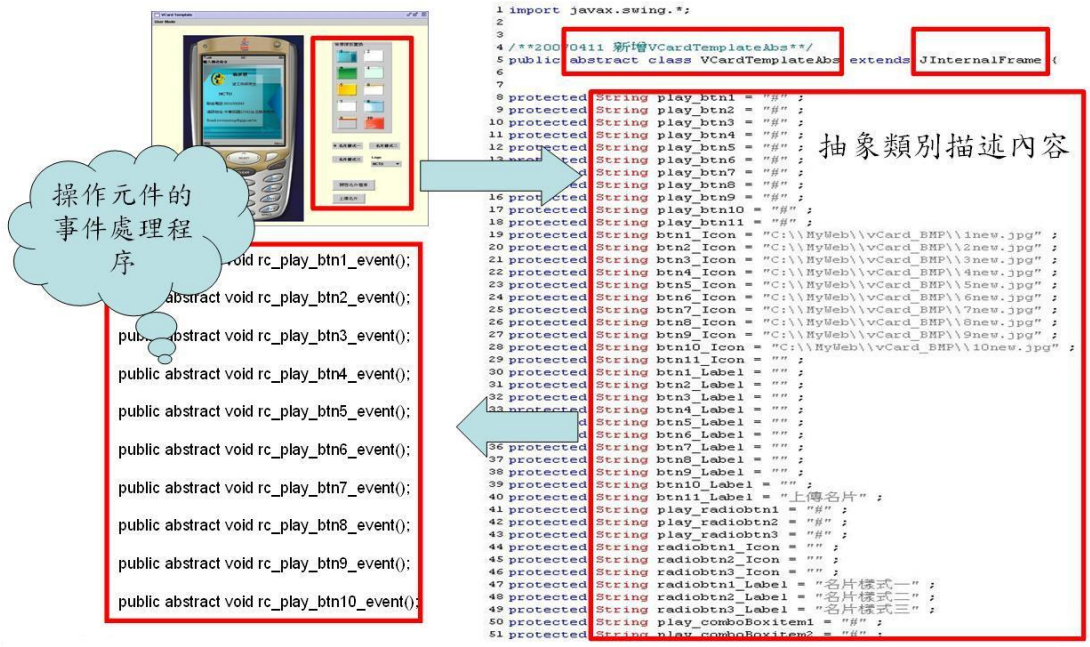


圖 37、抽象類別程式片段

當該類別定義好屬性變數與屬性值後，必須提供這些操作元件的事件處理程序，給開發者進行實作，以上述行動名片樣版編輯器的應用實例為例，VCardPlayer.java 這支程式就是進行實作這個抽象類別，運用這種模組化的方式規範程式的撰寫，對後期欲開發新的應用服務時，能夠利用繼承的概念提供多型的轉換，重複利用(reuse)該程式樣板的屬性而快速的完成一個新的應用程式。以下是系統開發者撰寫抽象類別程式的虛擬碼，開發者須依下述的規則撰寫程式樣板：

```
import javax.swing.*;

public abstract class“CLASSNAME”extends JInternalFrame {
//元件屬性的設定
protected String“播放的操作元件→(play_btnN | play_radiobtnN |
play_comboBoxitemN，N：[0-9]+) ” = “#” ；
protected String“操作元件圖示標籤→(btnN_Icon | radiobtnN_Icon |
comboBoxitemN_Icon，N：[0-9]+) ” = “ICON_FILE_URL” | ”” ；
protected String“操作元件文字標籤→(btnN_Label | radiobtnN_Label |
comboBoxitemN_Label，N：[0-9]+) ” = “LABEL_NAME”| ””  ；
//操作元件的事件處理程序
public abstract void rc_play_btnN_event();//N：[0-9]+
public abstract void rc_play_radiobtnN_event();//N：[0-9]+
public abstract void rc_play_comboBoxitemN_event();//N：[0-9]+
}
```

3.3.7 操作描述檔案說明

圖 38 是描述操作描述檔案與控制表單的關係，而在 3.3.5.3 節中，程式樣板剖析器模組(code template parsing module)在剖析完應用程式的樣板後，會產生操作描述檔案(Operation Script File)。

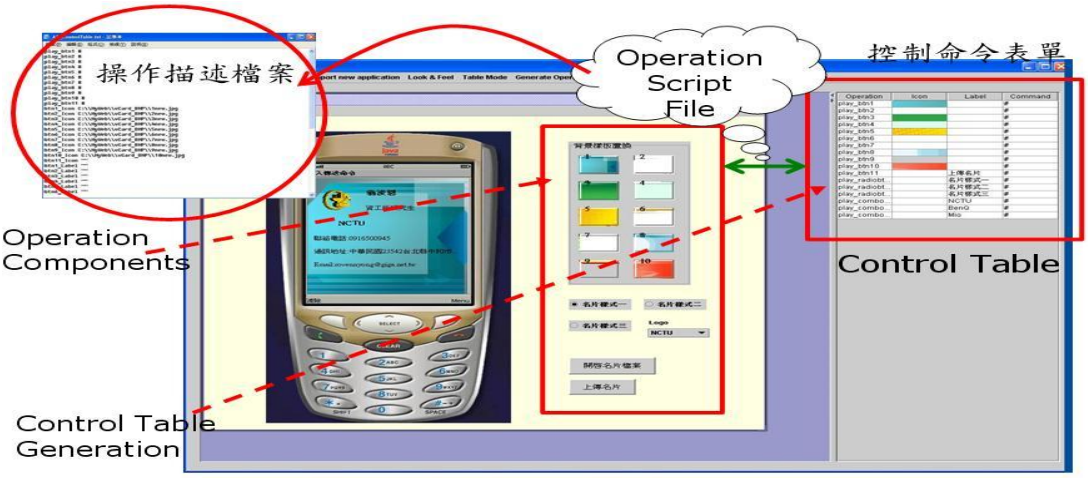


圖 38、操作描述檔案與控制表單的關係

而控制表單(Control Table)會根據操作描述檔案，自動產生在介面系統中應用程式載入器的畫面上，上圖是行動名片樣版編輯器的操作畫面，右列是所有的操作元件，包含按鈕(JButton)，單一選項的選擇鈕(JRadioButton)以及下拉式選項的選單(JComboBox)。而描述這些視覺化的操作元件，就是需要操作描述檔案來負責，最後程式樣板剖析器模組，再剖析該文件而產生控制表單畫面，如上圖右方的表單呈現結果。一份操作描述檔案的內容說明大致如圖 39 所示，共分成三個部份，第一個部份是操作元件項目的描述，共有以下三種元件項目：

- play_btnN ：代表按鈕(JButton)操作元件
- play_radiobtnN ：代表單一選項的選擇鈕(JRadioButton)操作元件
- play_comboBoxitemN ：代表下拉式選項的選單(JComboBox)操作元件 (N : [0-9]+)

第二個部份是操作元件的圖示檔案路徑，描述各個操作元件的圖示內容，每個元件可以有兩種描述資訊，一是以圖示內容描述，二是以標籤文字描述。兩種描述不能同時具有，操作元件的圖示檔案分為 jpg 與 png 檔案，jpg 圖檔(or gif)是提供給 Java 應用程式，而 png 圖檔則是提供給手機內的 Java 程式之介面產生器產生的手機程式，針對三種元件項目，共有以下三種圖示檔案路徑描述：

- btnN_Icon：代表按鈕(JButton)的圖示路徑
- radiobtnN_Icon：代表單一選擇鈕的圖示路徑
- comboBoxitemN_Icon：代表下拉式選項的圖示路徑

(N : [0-9]+)

第三個部份是操作元件的標籤文字，描述各個操作元件的文字內容，針對三種元件項目，共有以下三種文字標籤描述：

- btnN_Label：代表按鈕(JButton)的標籤文字
- radiobtnN_Label：代表單一選擇鈕的標籤文字
- comboBoxitemN_Label：代表下拉式選項的標籤文字

  (N : [0-9]+)

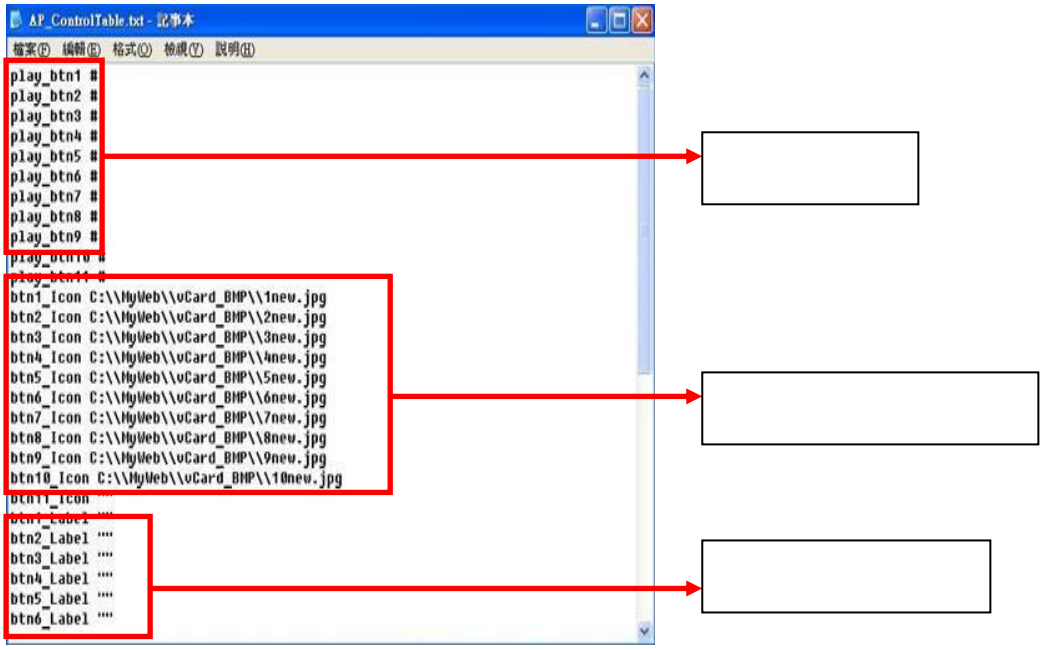


圖 39、操作描述檔案的內容說明

3.3.8 控制命令表單制定

上一節我們描述了操作描述檔案與控制表單的關係，而控制表單的內容是一個描述視覺化操作元件的多媒體欄位，除了描述所有操作元件資訊外，它需要為每一個操作元件制定一個巨集命令(macro command)。在本計畫開發的系統中，是利用命令觸發物件的方式來完成遙控操作 PC 上的元件，屆時使用者只要點選手機操作介面中的按鈕圖示即可傳送出命令，並根據控制表單來觸發 PC 上的操作元件，以表 12 舉例來說明：

表 12、使用命令字串觸發物件的方式

UI on Device	Command	Action
Btn_1	001	Focus object_1
Btn_2	002	Focus object_2
Btn_3	003	Focus object_3
Btn_4	004	Play object_1
Btn_5	005	Play object_2
Btn_6	006	Play object_3
……	……	……

假設在 Java 應用程式的 UI 畫面上，有數個按鈕元件(Btn_N，N：[0-9]+)，控制命令表單會賦予每個按鈕元件一個命令碼(command ID)，如同上表的描述 Btn_1 至 Btn_6 的命令碼分別是 001 至 006。當應用程式接收到 001 至 003 的命令時，對應的觸發事件行為分別是 Focus object_1 至 Focus object_3，而接收到 004 至 006 的命令時，對應的觸發事件行為分別是 Play object_1 至 Play object_3，以上就是一個利用命令來進行事件觸發的方法。

圖 40 至圖 42 就是本計畫所開發的三個應用實例與控制表單的操作描述，Operation 欄位是描述操作描述檔案上定義的操作元件項目，Icon 欄位是依據操作元件的圖示檔案路徑將圖示呈現出來，而 Label 欄位則是描述操作元件的標籤文字，Icon 欄位值與 Label 欄位值不會同時描述一個操作元件的內容。

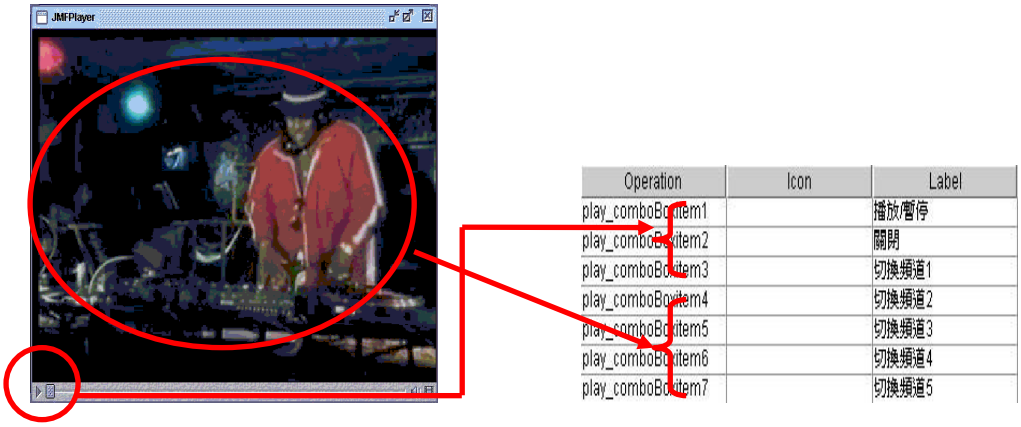


圖 40、JMF 播放器應用程式的控制表單操作描述

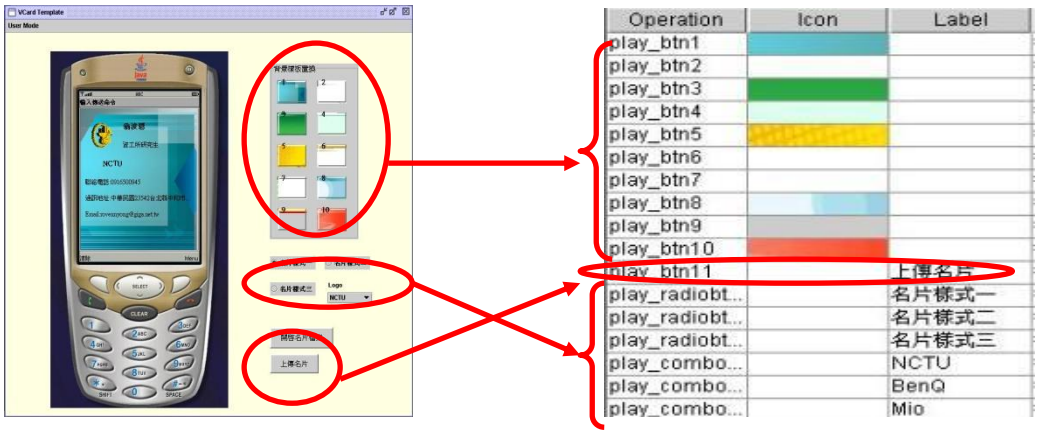
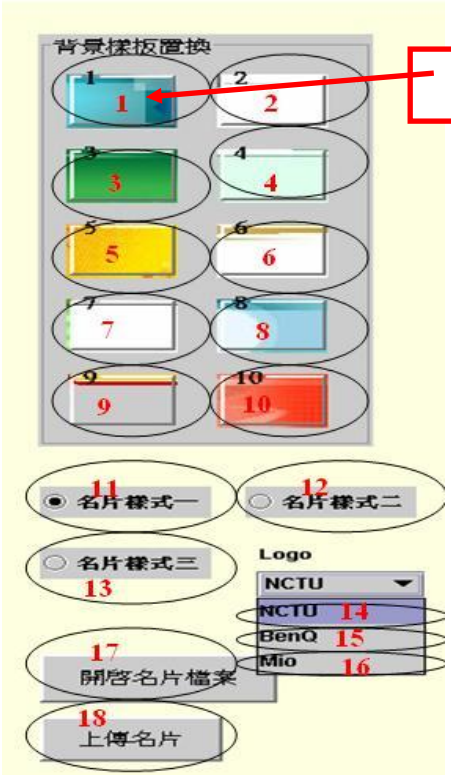


圖 41、行動名片樣版編輯器的控制表單操作描述

圖 42、電子賀卡樣版編輯器的控制表單操作描述

制定控制命令表單的方法有很多種方式，例如系統開發者於程式撰寫時，去制訂表單中的操作項目有哪些，並賦予一個固定的命令碼；另外一種方式則是依據 UI 上的元件數目，按順序動態賦予固定的命令碼。

在程式樣板剖析模組中，我們是依據 UI 上的元件數目，於程式撰寫中靜態賦予一個固定的命令碼，這是屬於系統開發者的製作方式，該方式會依據 UI 上的元件產生順序，來靜態賦予固定的命令碼，圖 43 是利用這種方式的賦予方法：



操作元件項目	命令碼
play_btn1	1
play_btn2	2
play_btn3	3
play_btn4	4
play_btn5	5
play_btn6	6
play_btn7	7
play_btn8	8
play_btn9	9
play_btn10	10
play_radiobtn1	11
play_radiobtn2	12
……	……

圖 43、以元件產生的順序來賦予命令碼

3.3.9 WTK 編譯器

在手機內的 Java 程式之介面產生器的程式碼產生模組中，當 MIDlet 的程式寫出的時候，該模組會呼叫 WTK(Wireless ToolKit)編譯器系統，這是一個可以編譯手機 MIDlet 程式的工具程式，並且會執行模擬器程式，模擬 MIDlet 程式實際執行在手機上的畫面。由於 WTK 自身並沒有附帶 Java 的運作環境，即 J2DK(Java 2 Standard Edition Development Kit)，所以在 WTK 安裝之前必需要安裝 JDK 的執行環境。

本計畫開發的 JDK 環境是 J2DK 1.4.2_01 版本，WTK 開發工具支援 MIDP(Mobile Information Device Profile)2.0 應用程序、WMA 2.0、MMAPI 1.1，File and PIM APIs (JSR 75)、Bluetooth、OBEX APIs (JSR 82)及 3D Graphics (JSR 184)，以下是 WTK2.2 所支援的這些 API 說明：

表 13、WTK2.2 所支援的 API 描述

WTK2.2 所支援的 API 種類	API 描述
WMA(Wireless Messaging API) 2.0	支援多媒體簡訊(Multimedia Messaging Service)服務。
MMAPI(Mobile Media API)	為資源有限的設備提供了音頻、視頻和其他多媒體支援格式。
File and PIM APIs(JSR 75)	提供檔案管理以 PIM(Personal Information Manager)的功能。
Bluetooth	藍芽連線的服務
OBEX(Object Exchange) APIs(JSR 82)	支援交換對象數據，諸如電子商業卡和日曆標籤之間以 vCard 和 vCalendar 的格式進行數據傳輸。
3D Graphics(JSR 184)	提供 3D 繪圖的功能。

無論哪個版本的 WTK 都會包括以下幾個目錄：

表 14、WTK 開發環境的工作目錄

目錄名稱	內容描述
appdb 目錄	RMS 數據庫信息
apps 目錄	WTK 附上的 demo 程序
bin 目錄	J2ME 開發工具執行文件
docs 目錄	各種幫助與說明文件
lib 目錄	J2ME 程序庫，Jar 包與控制文件
session 目錄	性能監控保存信息
wtklib 目錄	JWTK 主程序與模擬器外觀

開發的專案主要是放在 apps 目錄底下，以手機內的 Java 程式之介面產生器所產生的應用程式為例，編譯來源檔案路徑就是位於 C:\WTK22\apps\MyRemoteAP\src 底下，而所產生的目的專案名稱為 MyRemoteAP，以下我們利用 WTK 的系統畫面進行編譯手機內的 Java 程式之介面產生器所產生的 MIDlet 程式，來說明 WTK 的操作流程：

Step 1、執行 WTK(J2ME Wireless Toolkit)編譯工具(圖 44)

Step 2、使用者可以點選 New Project 以新增專案(圖 45)

Step 3、點選 Open Project 直接執行手機內的 Java 程式之介面產生器所產生的 MIDlet(圖 46)

Step 4、專案 MyRemoteAP 載入成功(圖 47)

Step 5、建置(build)與編譯 MyRemoteAP 專案(圖 48)

Step 6、建置專案完成(圖 49)

Step 7、 建置完成後的專案資料夾如圖 51 所示，src 是存放.java 的來源碼，當 Step 6 建置專案完成後，classes 資料夾會產生，編譯完成的 class 檔皆放於該資料夾底下(圖 50)。

Step 8、設定環境組態(圖 51)

Step 9、設定相關的屬性資訊(圖 52)

Step 10、執行 MIDlet 程式(圖 53)

Step 11、啟動手機模擬器以執行 MIDlet 程式(圖 54、圖 55)

Step 12、執行手機模擬器的操作畫面(圖 56)

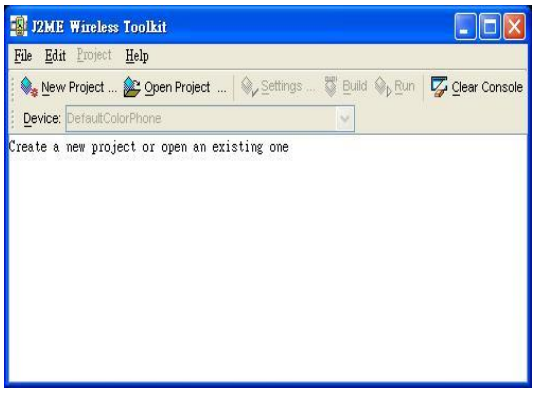


圖 44、執行畫面

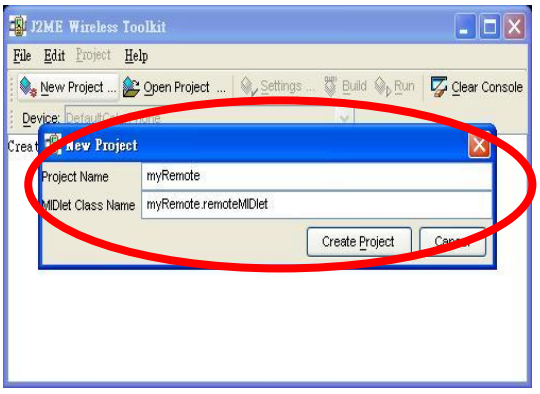


圖 45、新增專案

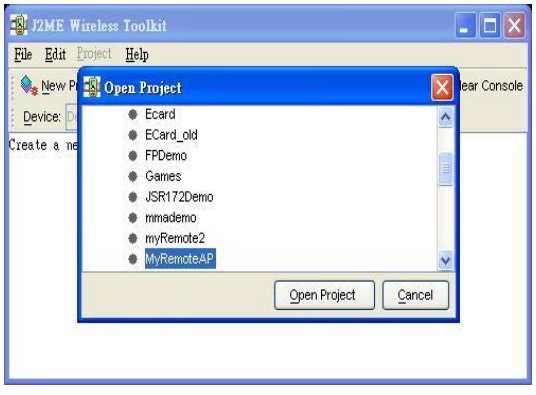


圖 46、打開專案

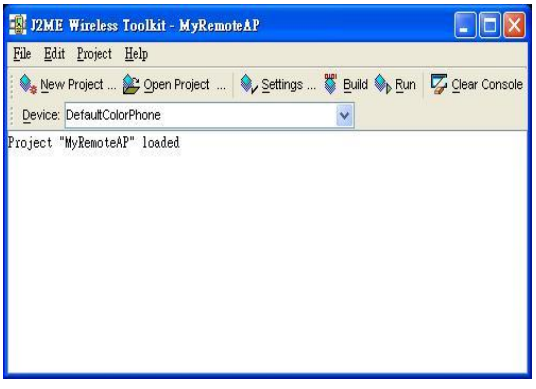


圖 47、載入專案 MyRemoteAP

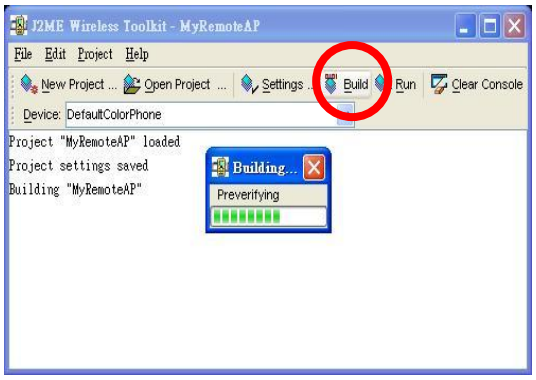


圖 48、建置專案 MyRemoteAP

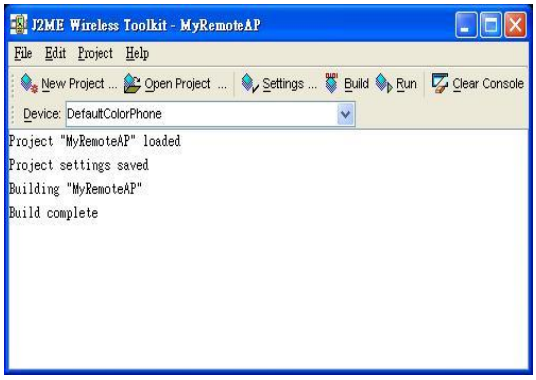


圖 49、建置專案 MyRemoteAP 成功

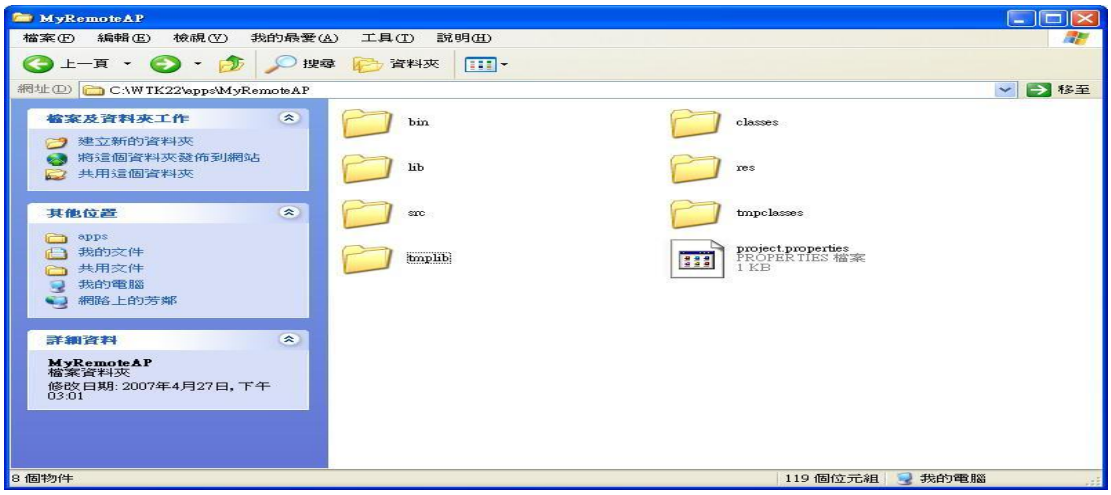


圖 50、專案 MyRemoteAP 的資料夾內容

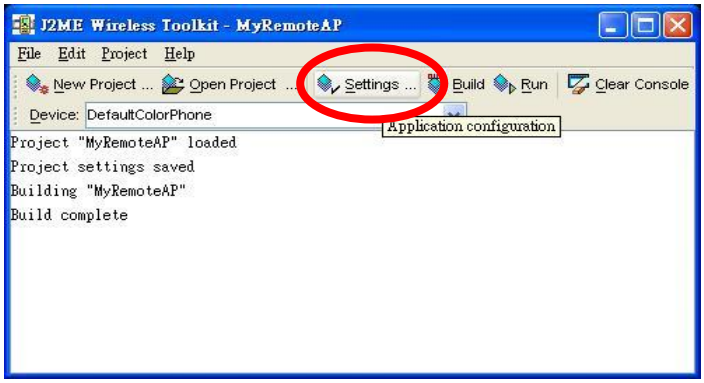
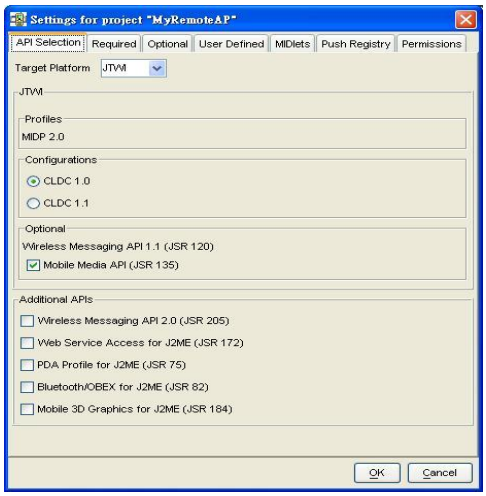
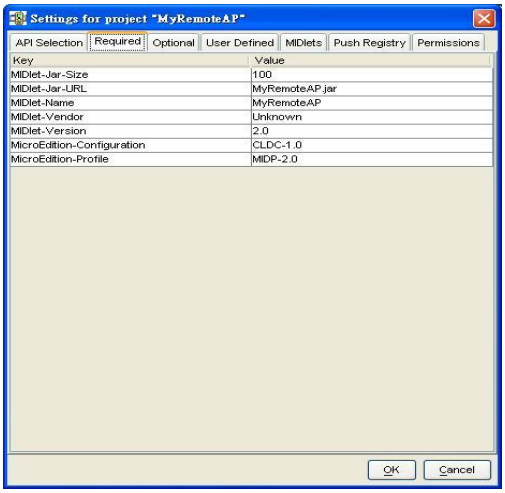


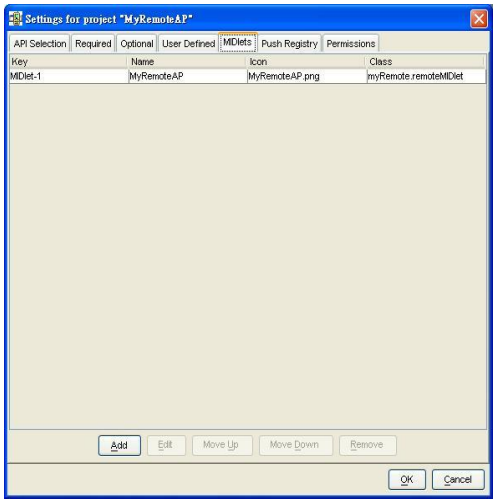
圖 51、按下 Settings 選項以設定環境組態



(a) 選擇 API 型式



(b)進行 MIDlet 檔案描述設定



(c)確認 MIDlet 程式的描述資訊

圖 52、相關的組態資訊內容

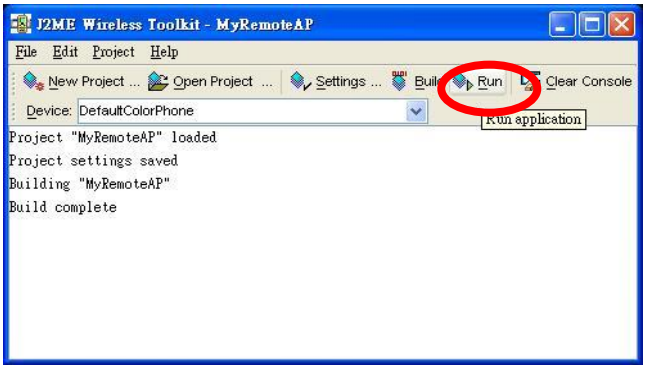


圖 53、點選 Run 執行按鈕，執行 MIDlet 程式

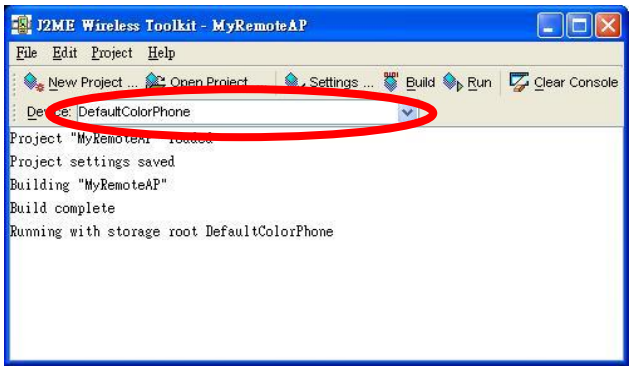
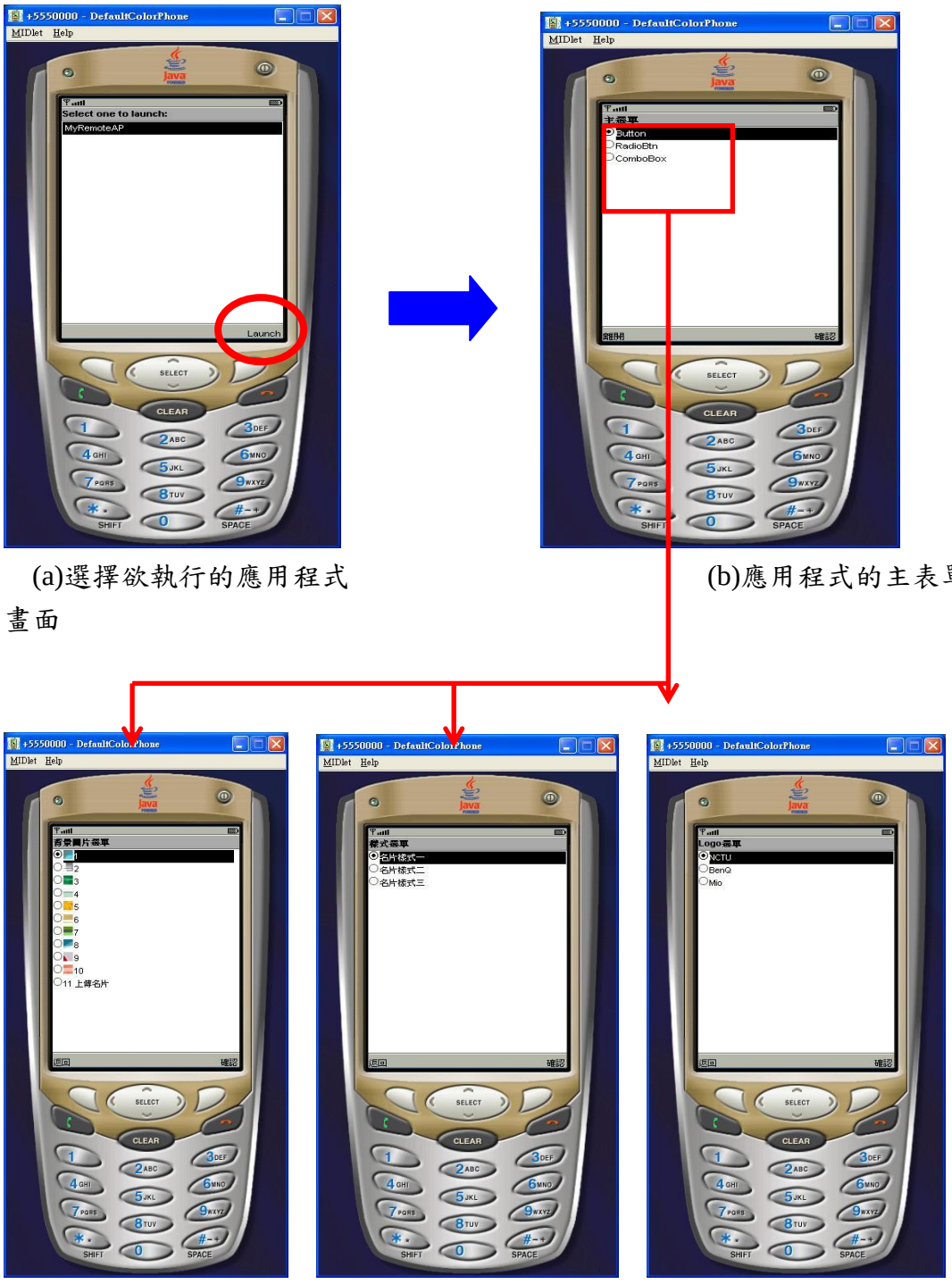


圖 54、以預設的模擬器(DefaultColorPhone)來執行

圖 55、DefaultColorPhone 模擬器畫面



(a)選擇欲執行的應用程式畫面　(b)應用程式的主表單

(c)Button 的操作選單　(d)RadioButton 的操作選單　(e)Combobox 操作選單

圖 56、模擬器主選單與操作選單的畫面

以上是在 WTK 開發工具下的操作說明，在系統實作中由手機內的 Java 程式之介面產生器所產生的 MIDlet 程式，會放置到 MyRemoteAP 的專案底下，該專案需要利用 WTK 工具進行編輯，當 MIDlet 程式碼產生完畢後，介面系統

(interfacing system)會自動執行 WTK 開發工具以建置 MyRemoteAP 的專案。

手機模擬器的執行結果會與目標裝置上的呈現不大一樣，那是由於每支手機的呈現能力不同導致，未來針對手機介面產生的改良，我們也可以進行適性化的情境感知(Context aware)協調來完成介面的改良。

3.4 Examples

3.4.1 介紹

在本章中會實際採用三個 Java 應用程式，來進行介面載入以及產生對應的手機介面操作程式，我們會根據 3.3.3 節的圖 23(b)步驟程序進行產生手機介面，這三個 Java 應用程式是在 PC 上的 J2SDK 平台上執行，以提供 Java 應用服務。

PC 上的三個應用服務範例分別是 Java 多媒體播放器(Java Media Player)、行動多媒體名片樣板編輯器及電子賀卡樣板編輯器，本計畫之系統操作分為兩種使用者，系統端使用者與終端使用者，3.4.2 至 3.4.4 節會分別進行這兩種使用者的操作。

3.4.2 Java 多媒體播放器

該播放器系統是利用 JMF(Java Media Framework)API 所開發的一個頻道播放器，採用的版本為 jmf2.1.1e，本系統能夠提供五個 mpg 影片的頻道播放，透由 JMF 的架構，將多媒體的影音資訊傳送給客戶端。JMF 是一個提供給使用者開發影音多媒體服務的一個套件，它不是標準 Java API 所制定的套件，由於 JMF 並不包含在 JDK 裡，所以先必須到 Sun 的網站下載該套件[31]，然後進行環境的安裝，利用 JMF 建構的影片視訊播放器能提供以下的格式播放：

表 15、JMF 支援的播放格式

Media Type	JMF 2.1.1 Cross Platform Version	JMF 2.1.1 Solaris/Linux Performance Pack	JMF 2.1.1 Windows Performance Pack
AIFF (.aiff)	read/write	read/write	read/write
AVI (.avi)	read/write	read/write	read/write
HotMedia (.mvr)	read only	read only	read only
MIDI (.mid)	read only	read only	read only
MPEG-1 Video (.mpg)		read only	read only
MPEG Layer II Audio (.mp2)	read only	read/write	read/write
Wave (.wav)	read/write	read/write	read/write

該系統是以三個廣告與兩個導覽簡介的 mpg 檔案當作播放的來源，以下的系統畫面是系統端使用者利用介面系統進行載入 Java 多媒體播放器，並產生出手機遙控介面的操作流程：

Step1：在系統畫面上選擇載入 Java 多媒體播放器(圖 57、圖 58)

Step2：點選 Generate Operation File 以產生操作描述檔案(圖 60、圖 61)

Step3：點選 Generate Control Table 以產生控制表單(圖 62、圖 63、圖 64)

Step4：點選 Generate MIDlet Source Code 並執行 WTK 編譯器(圖 65)

Step5：編譯產生的 MIDlet 程式並執行手機模擬器(圖 66)

Step6：執行手機模擬器的畫面(圖 67)

Step7：PDA Phone 上的 MIDlet 程式執行畫面(圖 68)

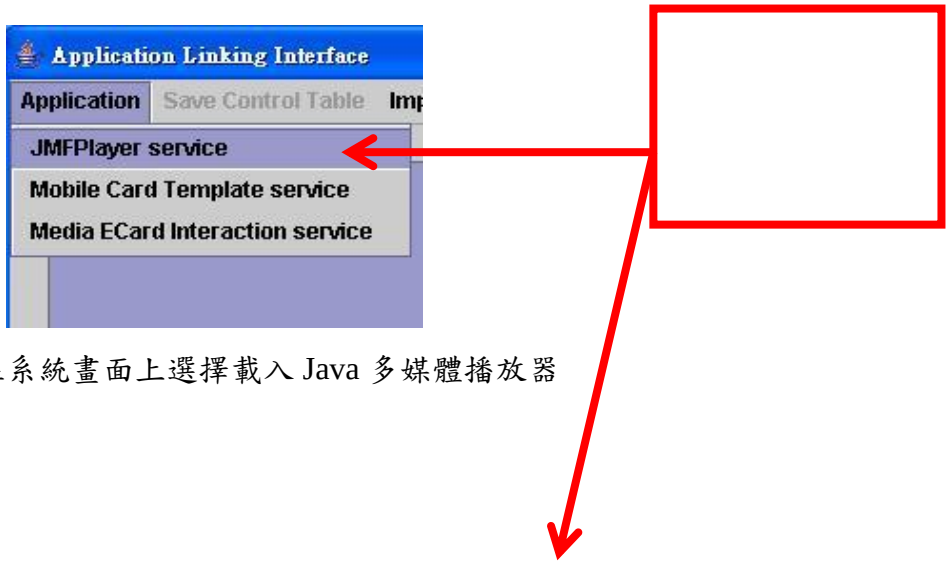


圖 57、在系統畫面上選擇載入 Java 多媒體播放器

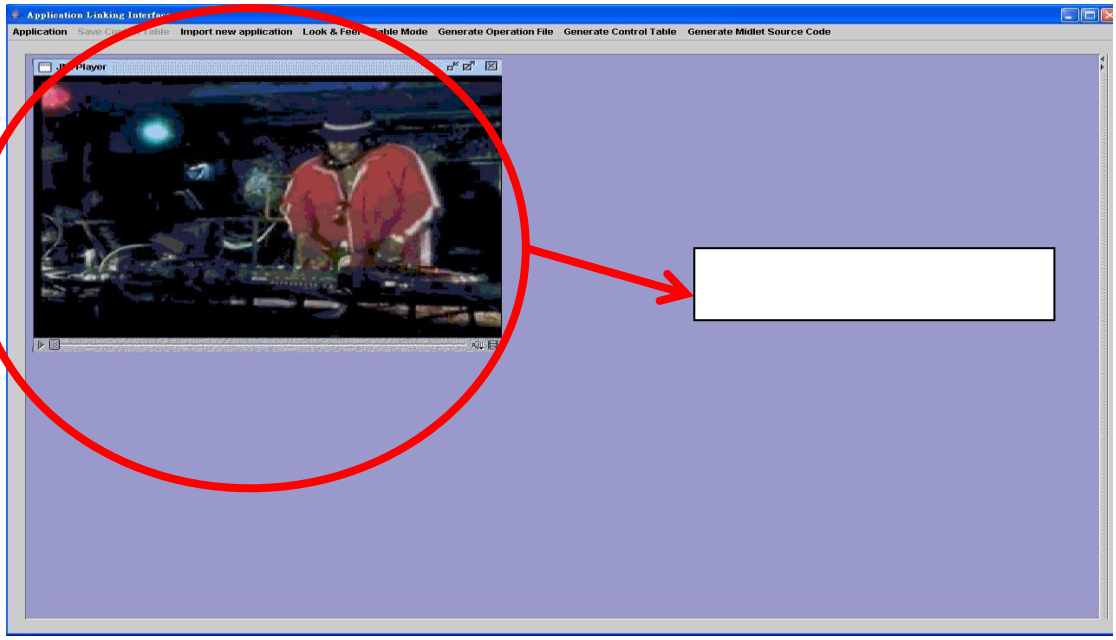


圖 58、Java 多媒體播放器系統畫面載入結果

資料來源：海尼根廣告網頁
(http://www.heineken.com/taiwan/lounge/estuff/download/videos/BirthOfScratch.wmv)

圖 59 為 Java 多媒體播放器系統的操作畫面，用以說明相關的操作元件。

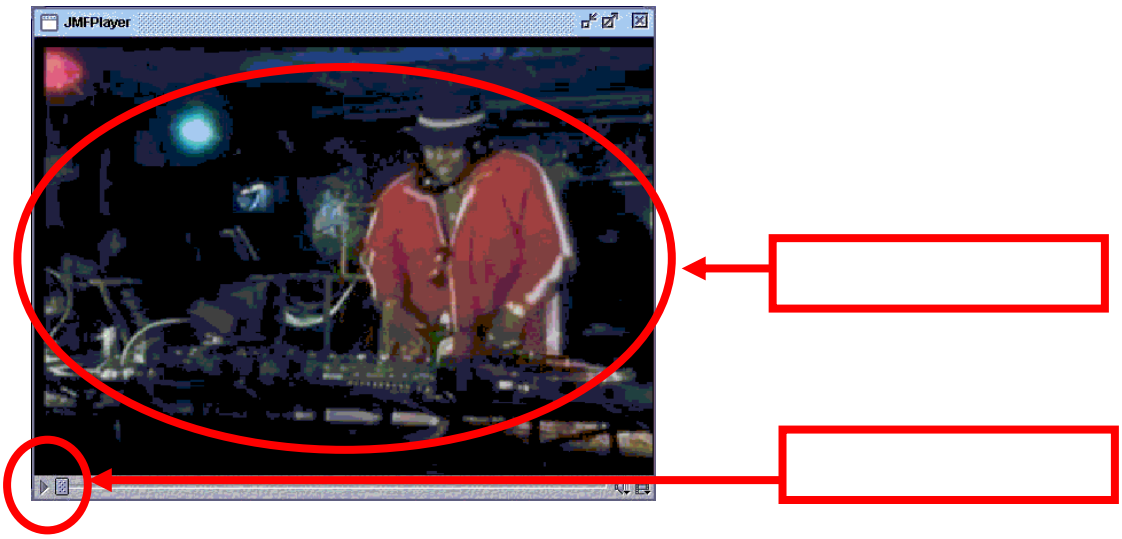


圖 59、Java 多媒體播放器系統的操作畫面說明

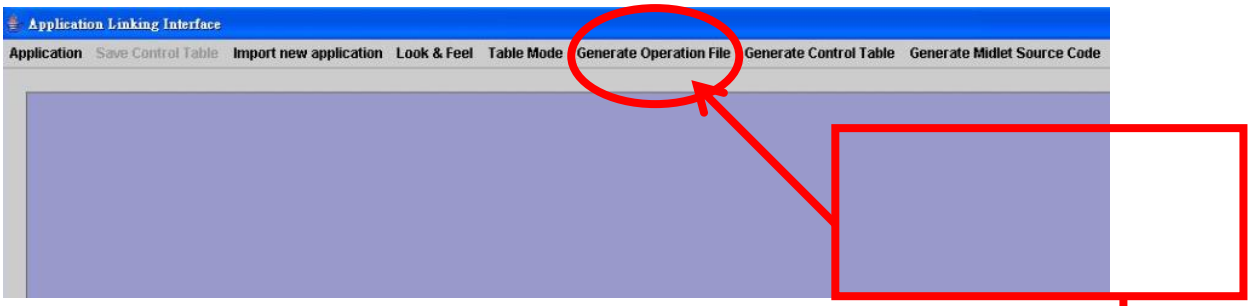


圖 60、點選 Generate Operation File 以產生操作描述檔案

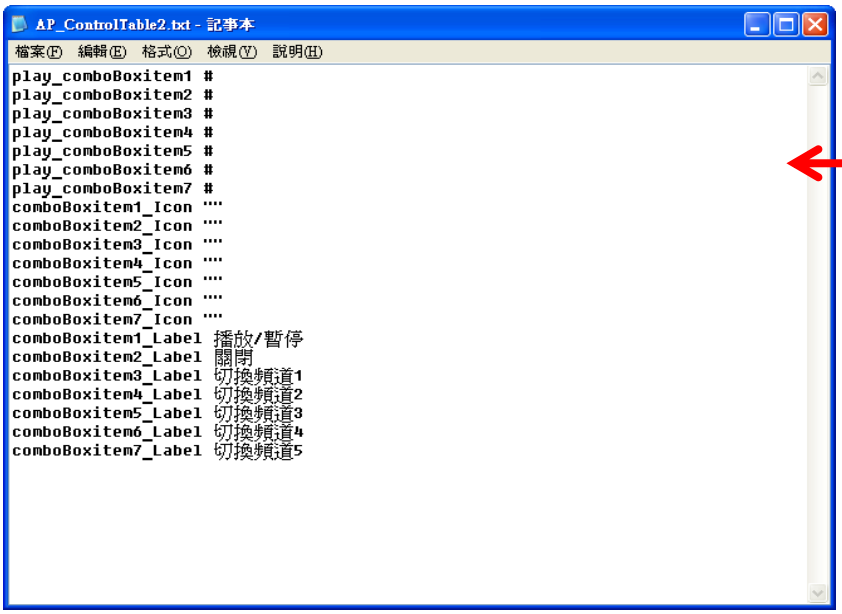


圖 61、描述檔案的內容

圖 62、點選 Generate Control Table 以產生控制表單

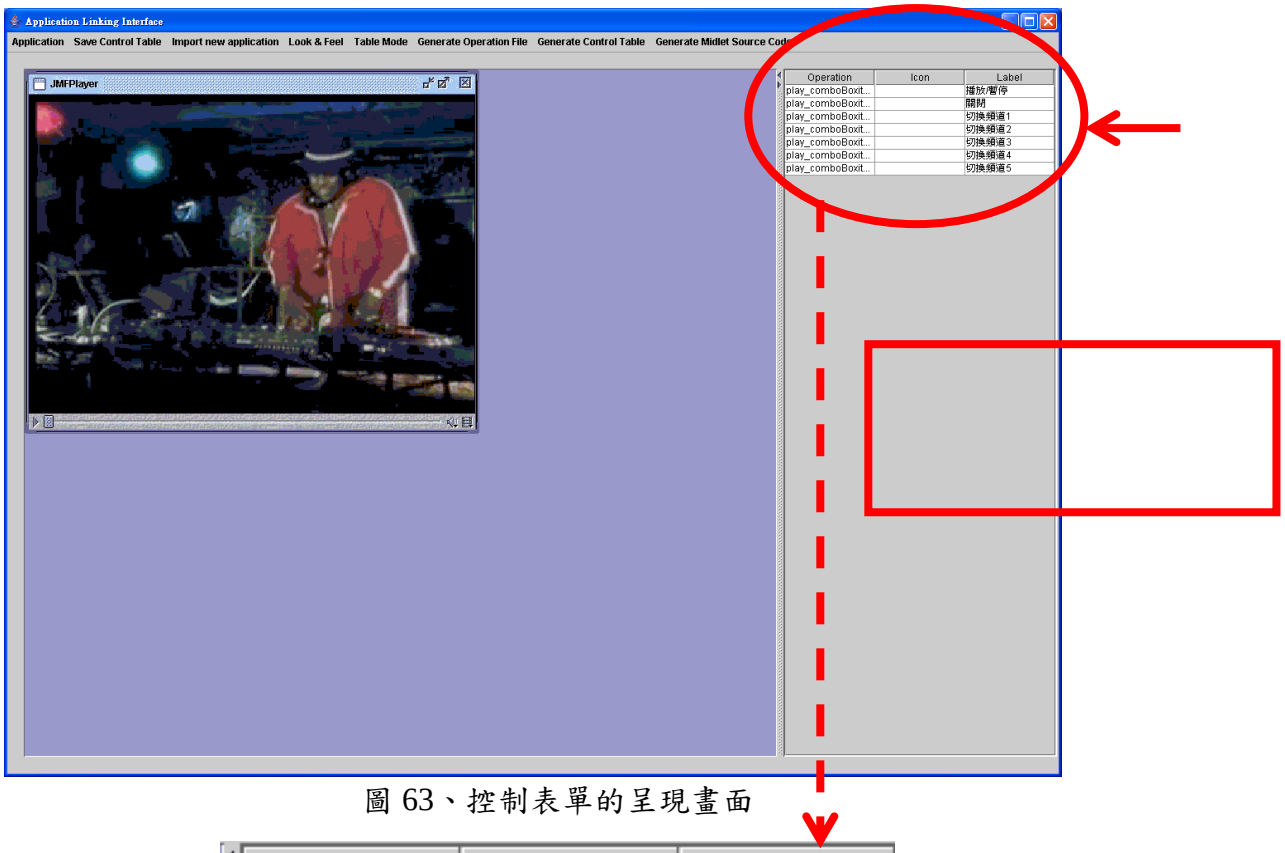


圖 63、控制表單的呈現畫面

圖 64、控制表單的內容

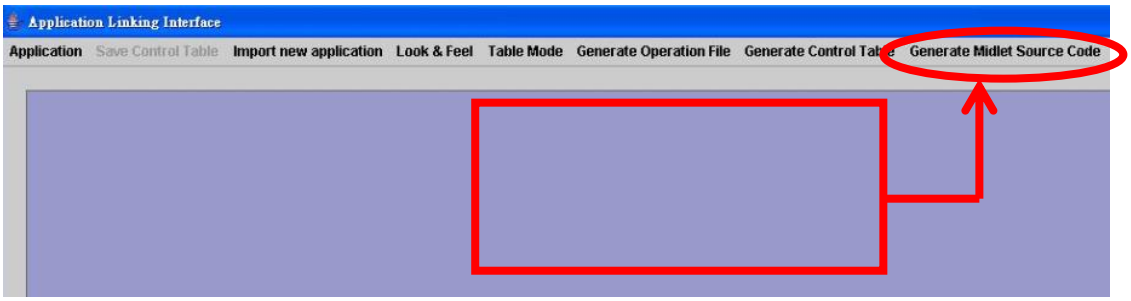


圖 65、點選 Generate MIDlet Source Code 並執行 WTK 編譯器

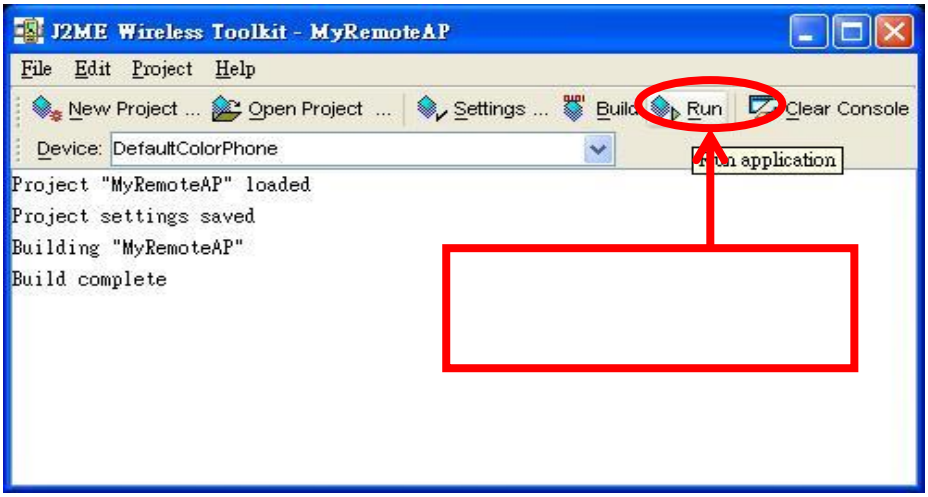
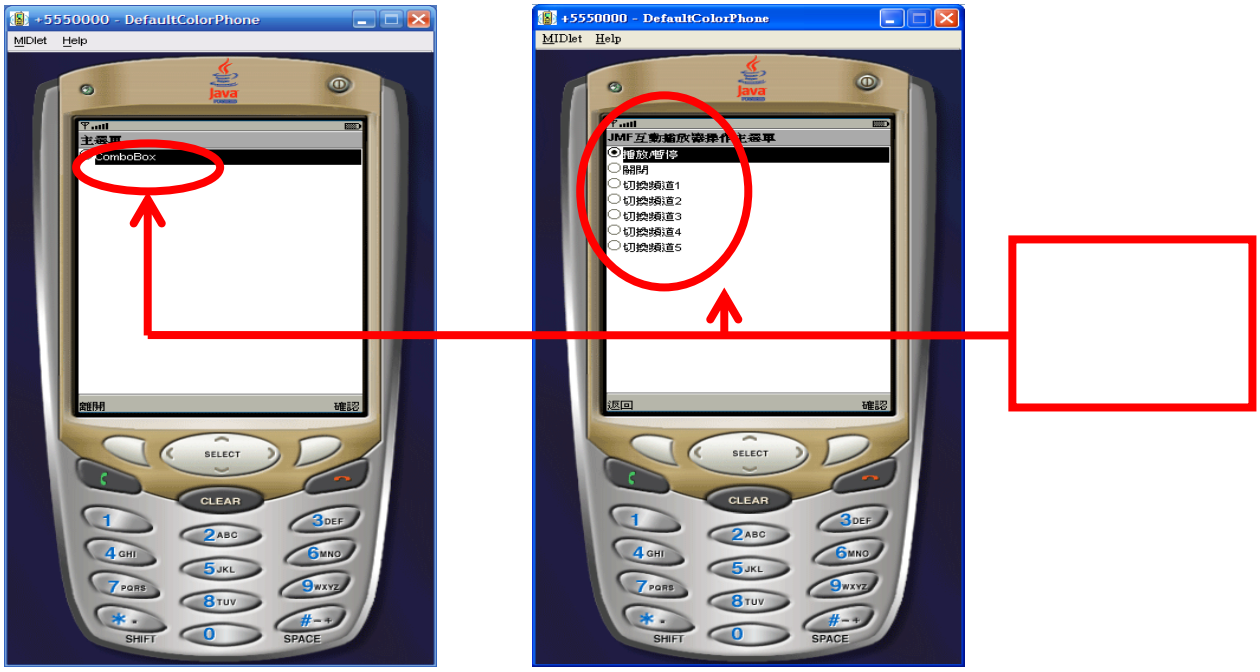
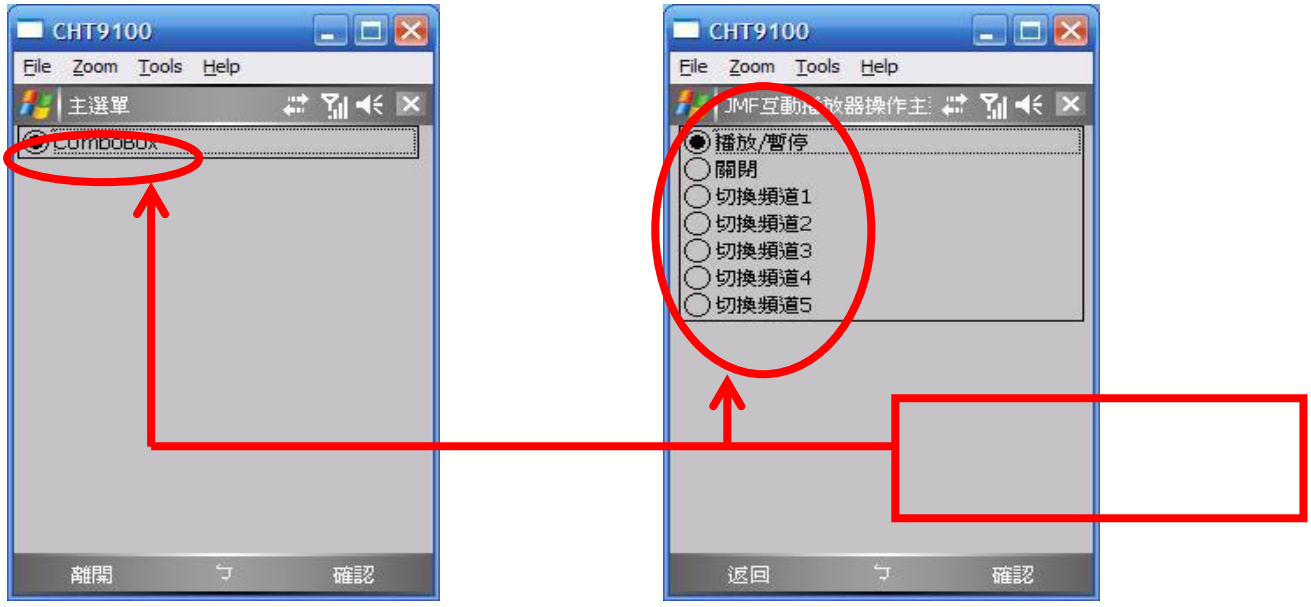


圖 66、編譯產生的 MIDlet 程式並執行手機模擬器



(a) 操作項目主選單-按鈕元件　　(b) 操作項目的內容選單-按鈕元件

圖 67、執行手機模擬器的畫面



(a) 操作項目主選單-按鈕元件　　(b) 操作項目的內容選單-按鈕元件

圖 68、PDA Phone 上的 MIDlet 程式執行畫面

圖 69 是終端使用者在手機上進行遙控操作 PC 上的 Java 多媒體播放器，在手機操作介面上的按鈕與 Java 多媒體播放器的操作畫面上，圈框起的部份是標示相同對應的操作功能：

圖 69、手機上遙控 Java 多媒體播放器-Combobox 選單操作

資料來源：海尼根、可口可樂、麥當勞商業廣告片段下載、原住民文化簡介導覽下載。(http://www.icoke.com.tw/，http://www.mcdonalds.com.tw/)

3.4.3 行動多媒體名片樣板編輯器

行動名片樣板編輯器是利用 Java Swing 套件[26]所開發的 GUI(graphical user interface)系統，提供名片製作者在 PC 上進行多媒體名片的製作，主要的名片製作功能，是進行背景樣板(background)、Logo 與呈現樣式(layout)的套用。在本計畫系統應用實例中，是利用手機來進行遙控編輯，換言之，編輯器上的編輯功能，將由手機介面程式來進行遙控操作，以下的系統畫面是系統端使用者利用介面系統進行載入行動名片樣板編輯器，並產生出手機遙控介面的操作流程：

Step1：在系統畫面上選擇載入行動名片樣板編輯器(圖 70、圖 71)

Step2：點選 Generate Operation File 以產生操作描述檔案(圖 73、圖 74)

Step3：點選 Generate Control Table 以產生控制表單(圖 75、圖 76、圖 77)

Step4：點選 Generate MIDlet Source Code 並執行 WTK 編譯器(圖 78)

Step5：編譯產生的 MIDlet 程式並執行手機模擬器(圖 79)

Step6：執行手機模擬器的畫面(圖 80)

Step7：PDA Phone 上的 MIDlet 程式執行畫面(圖 81)

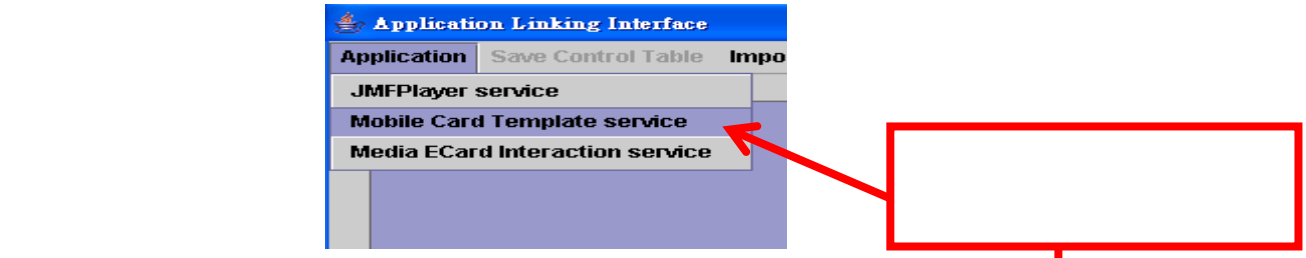


圖 70、在系統畫面上選擇載入行動名片樣板編輯器

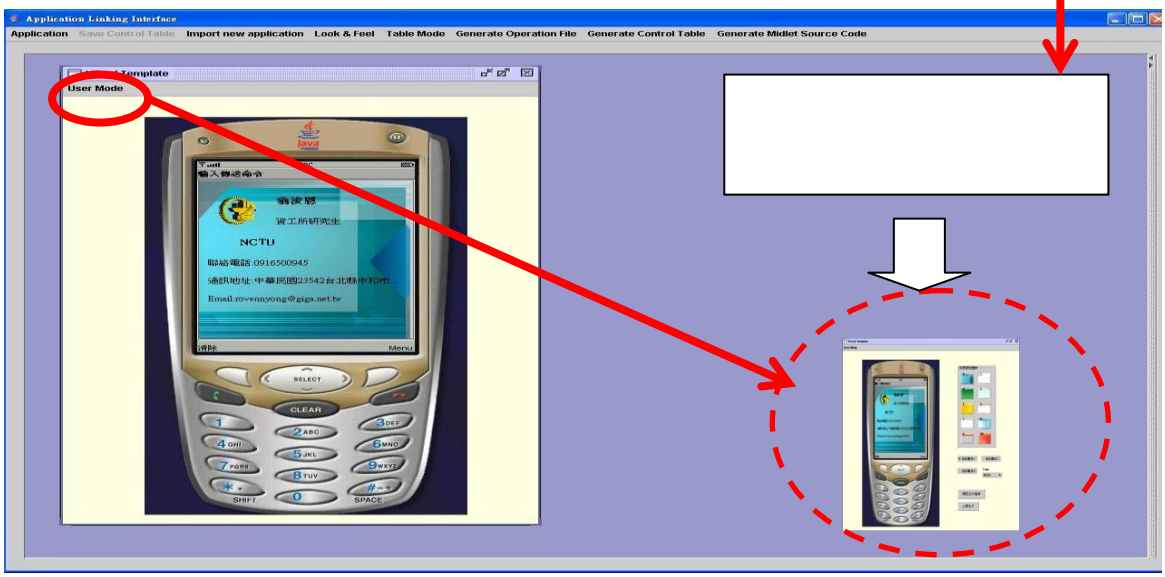


圖 71、行動名片樣板編輯器系統畫面載入結果

圖 72 為 Java 多媒體播放器系統的操作畫面，用以說明相關的操作元件。

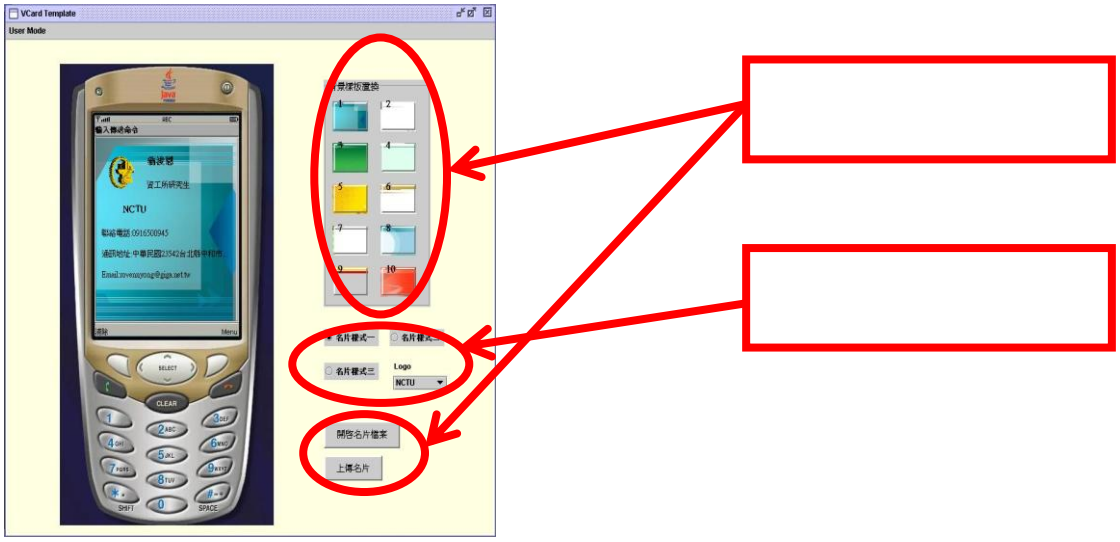


圖 72、行動名片樣板編輯器系統的操作畫面說明

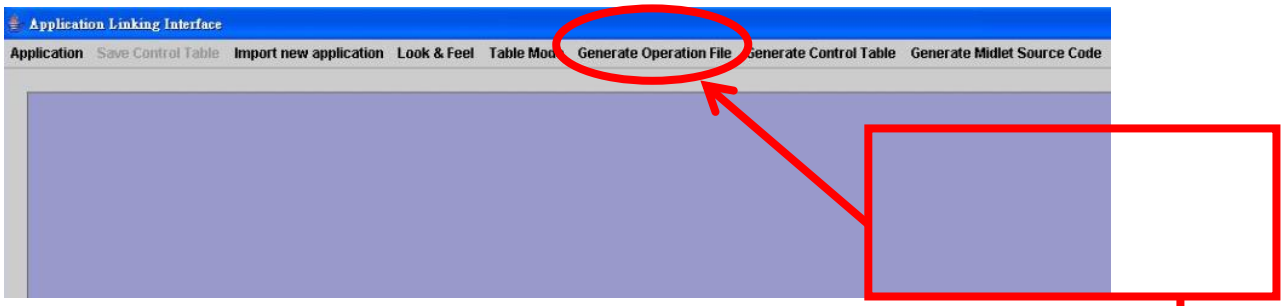


圖 73、點選 Generate Operation File 以產生操作描述檔案

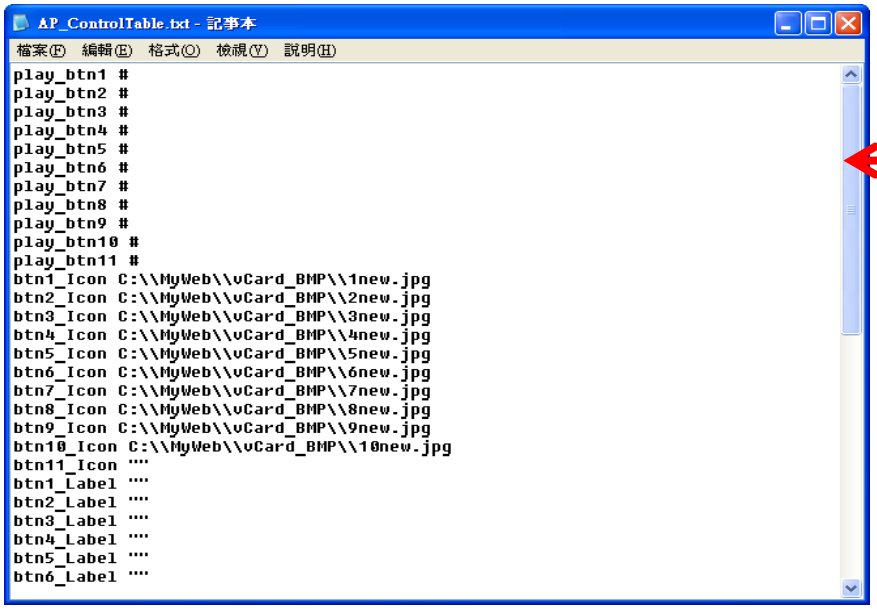


圖 74、描述檔案的內容

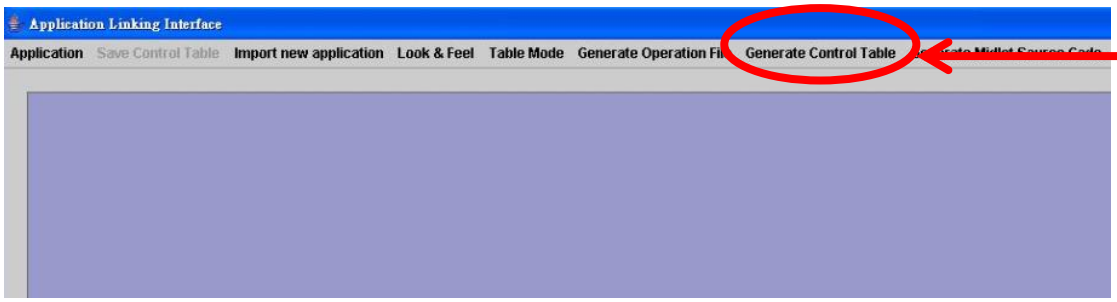


圖 75、點選 Generate Control Table 以產生控制表單

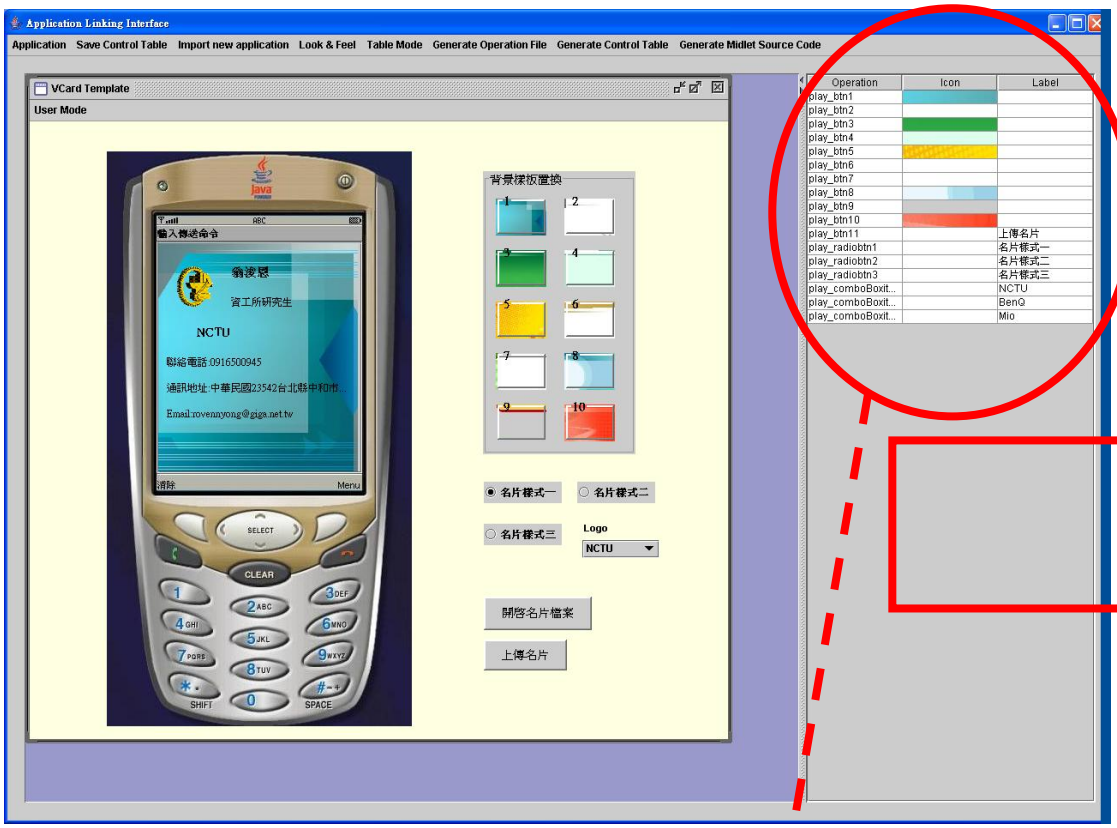


圖 76、控制表單的呈現畫面

Operation	Icon	Label
play_btn1		
play_btn2		
play_btn3		
play_btn4		
play_btn5		
play_btn6		
play_btn7		
play_btn8		
play_btn9		
play_btn10		
play_btn11		上傳名片
play_radiobt...		名片樣式一
play_radiobt...		名片樣式二
play_radiobt...		名片樣式三
play_combo...		NCTU
play_combo...		BenQ
play_combo...		Mio

圖 77、控制表單的內容

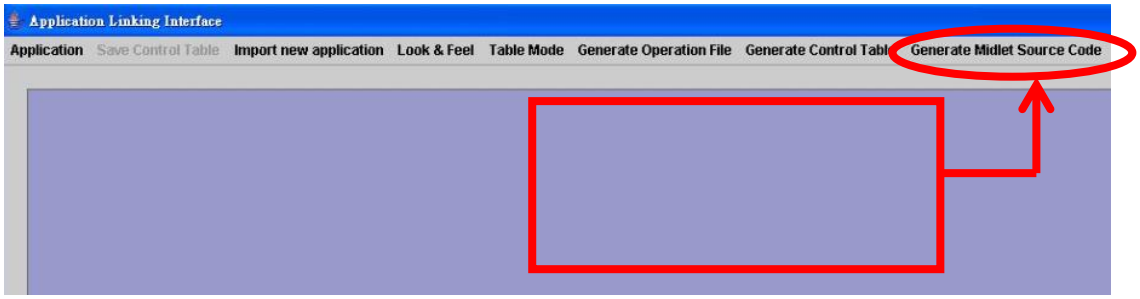


圖 78、點選 Generate MIDlet Source Code 並執行 WTK 編譯器

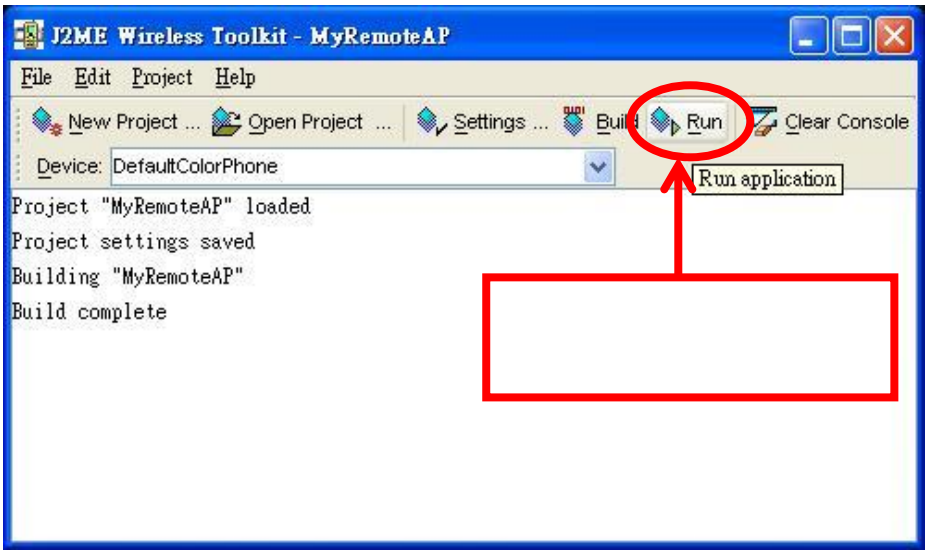
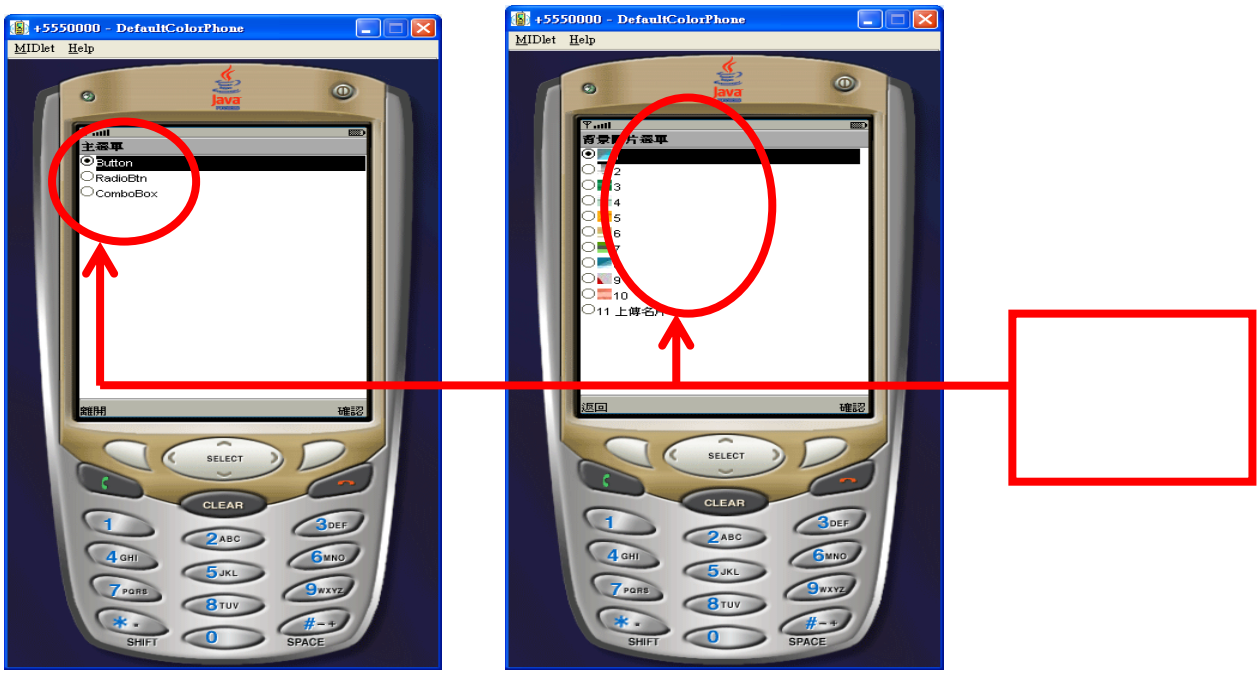
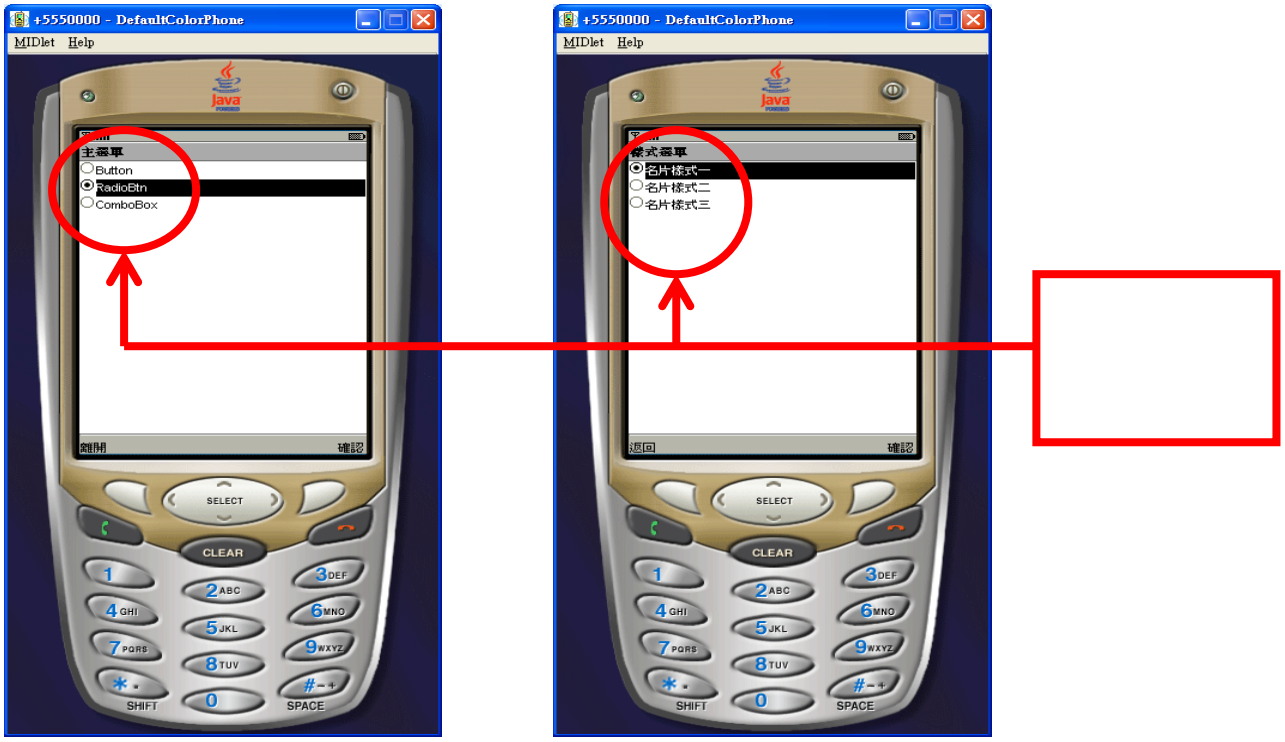


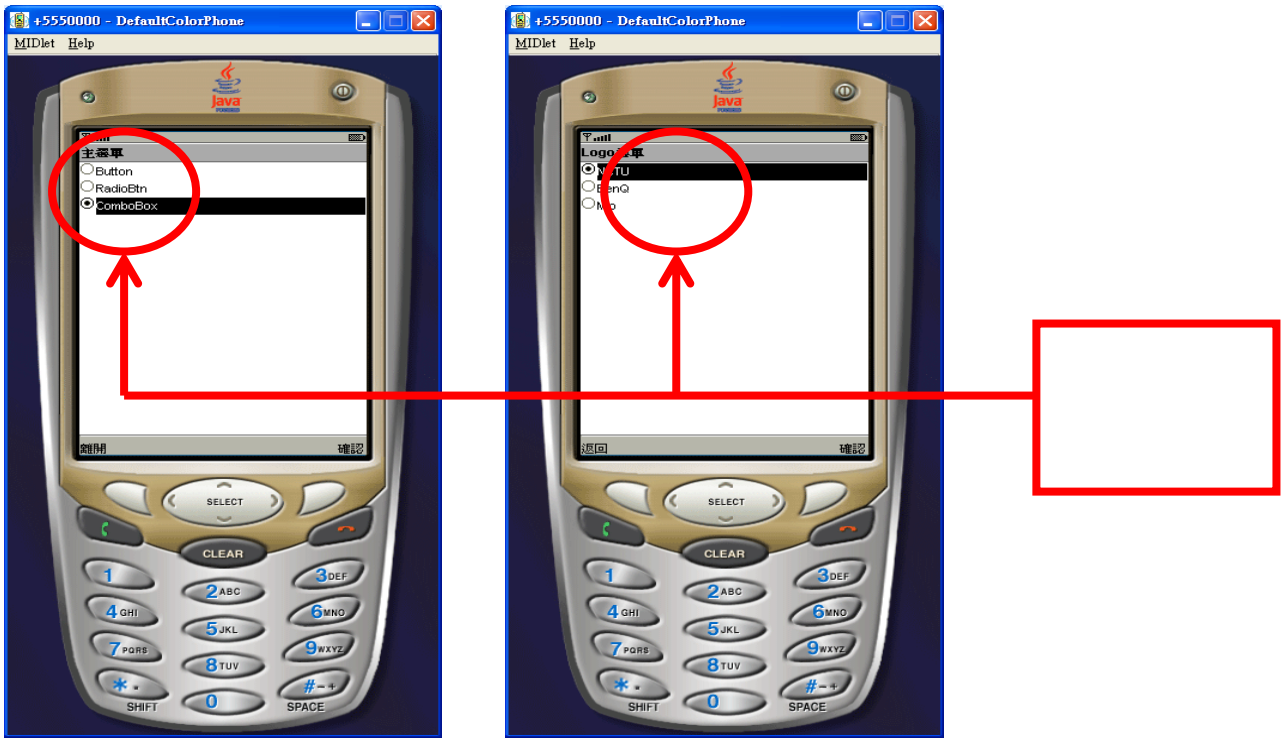
圖 79、編譯產生的 MIDlet 程式並執行手機模擬器



(a) 操作項目主選單-按鈕元件　　(b) 操作項目的內容選單-按鈕元件

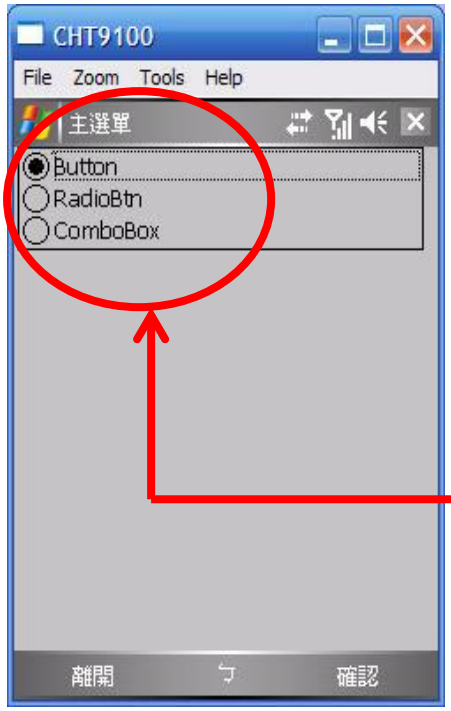


(c) 操作項目主選單-選項鈕元件　(d) 操作項目的內容選單-選項鈕元件

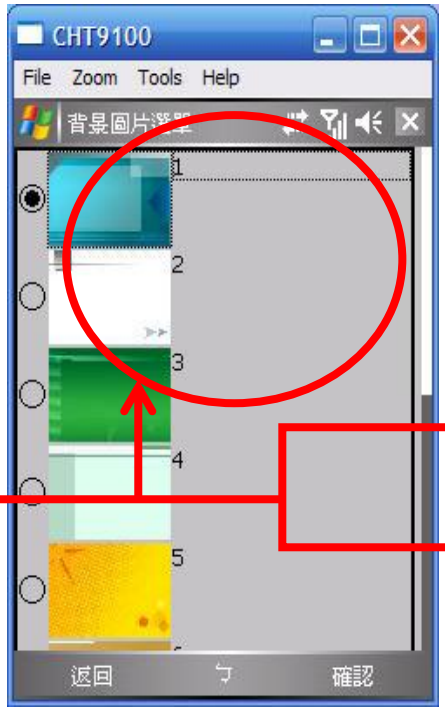


(e) 操作項目主選單-下拉選單元件　(f) 操作項目的內容選單-下拉選單元件

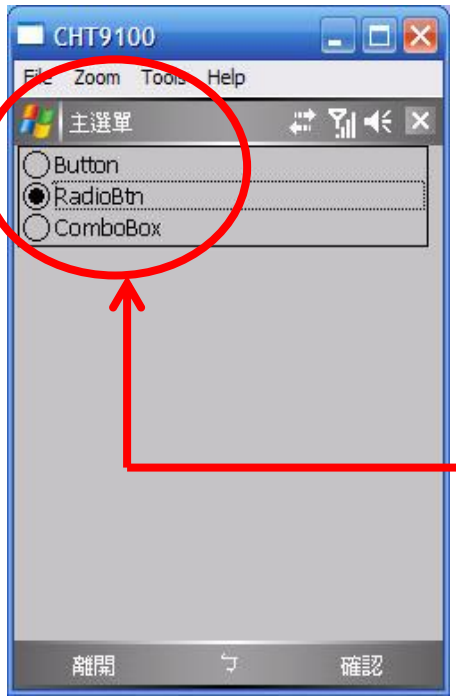
圖 80、執行手機模擬器的畫面



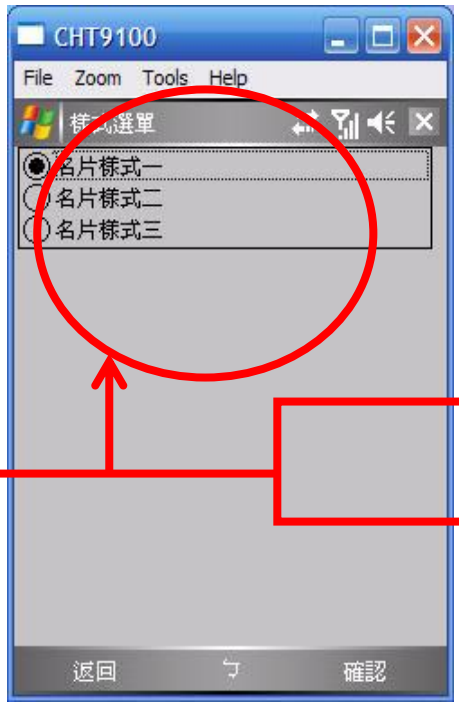
(a) 操作項目主選單-按鈕元件



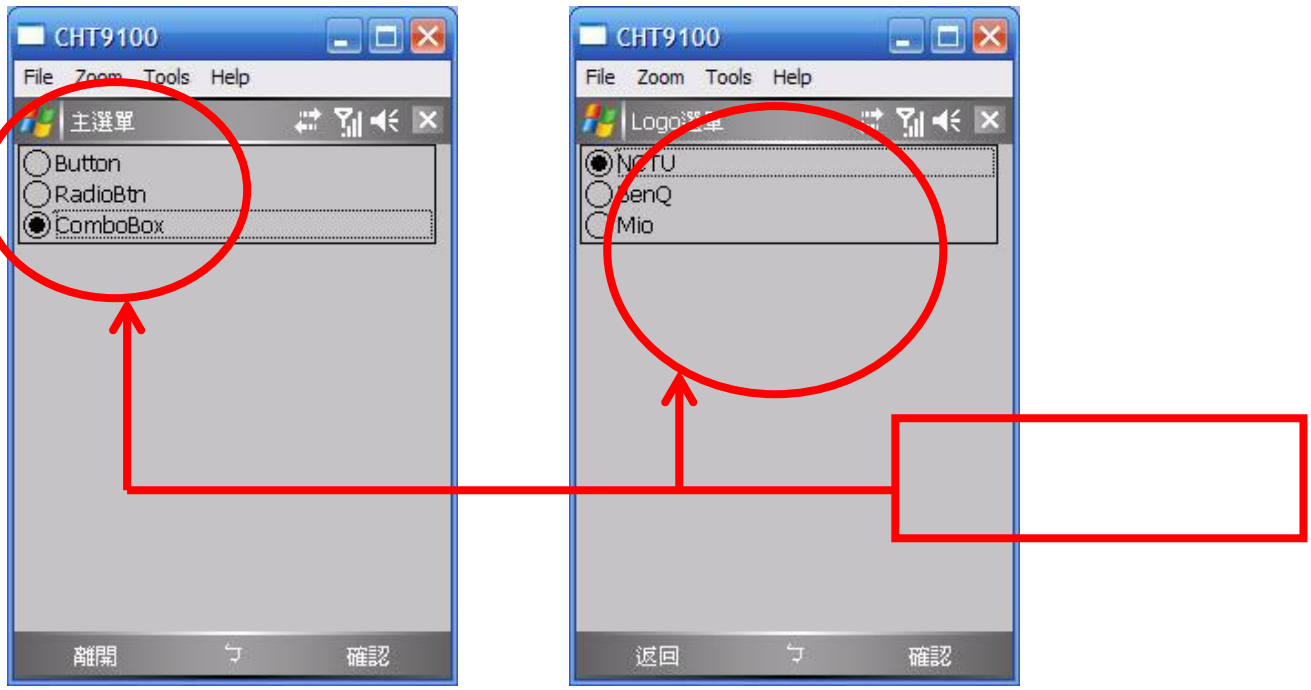
(b) 操作項目的內容選單-按鈕元件



(c) 操作項目主選單-選項鈕元件



(d) 操作項目的內容選單-選項鈕元件



(e) 操作項目主選單-下拉選單元件　(f) 操作項目的內容選單-下拉選單元件

圖 81、PDA Phone 上的 MIDlet 程式執行畫面

圖 82 至 84 是終端使用者在手機上進行遙控操作 PC 上的行動多媒體名片樣板編輯器，在手機操作介面上的按鈕與行動多媒體名片樣板編輯器的操作畫面上，圈框起的部份是標示相同對應的操作功能：

圖 82、手機上遙控行動多媒體名片樣板編輯器-Button 按鈕操作

圖 83、手機上遙控行動多媒體名片樣板編輯器-RadioBtn 按鈕操作

圖 84、手機上遙控行動多媒體名片樣板編輯器-Combobox 選單操作

3.4.4 電子賀卡樣板編輯器

電子賀卡樣板編輯器系統，提供多媒體賀卡的樣板套用功能，操作功能類似行動名片樣板編輯器，但具有動畫角色的套用功能。主要的賀卡製作功能是進行背景樣板(background)與動畫角色(actor)的套用，之後播放畫面會隨製作者的套用，動態呈現預覽的結果，最後可以進行上傳到伺服器平台進行協調，再下載至手機上播放，電子賀卡的編輯機制，搭配行動化，將來可以提供互動式多媒體簡訊的應用，以下的介面系統畫面是系統端使用者利用介面系統載入電子賀卡樣板編輯器，並產生出手機遙控介面的操作流程：

Step1：在系統畫面上選擇載入電子賀卡樣板編輯器(圖 85、圖 86)
Step2：點選 Generate Operation File 以產生操作描述檔案(圖 88、圖 89)
Step3：點選 Generate Control Table 以產生控制表單(圖 90、圖 91、圖 92)
Step4：點選 Generate MIDlet Source Code 並執行 WTK 編譯器(圖 93)
Step5：編譯產生的 MIDlet 程式並執行手機模擬器(圖 94)
Step6：執行手機模擬器的畫面(圖 95)
Step7：PDA Phone 上的 MIDlet 程式執行畫面(圖 96)

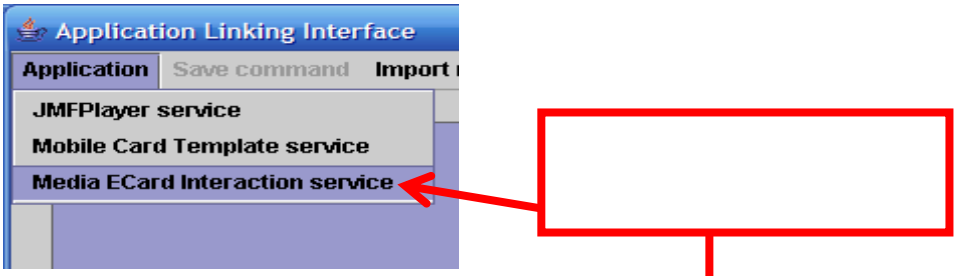


圖 85、在系統畫面上選擇載入電子賀卡樣板編輯器

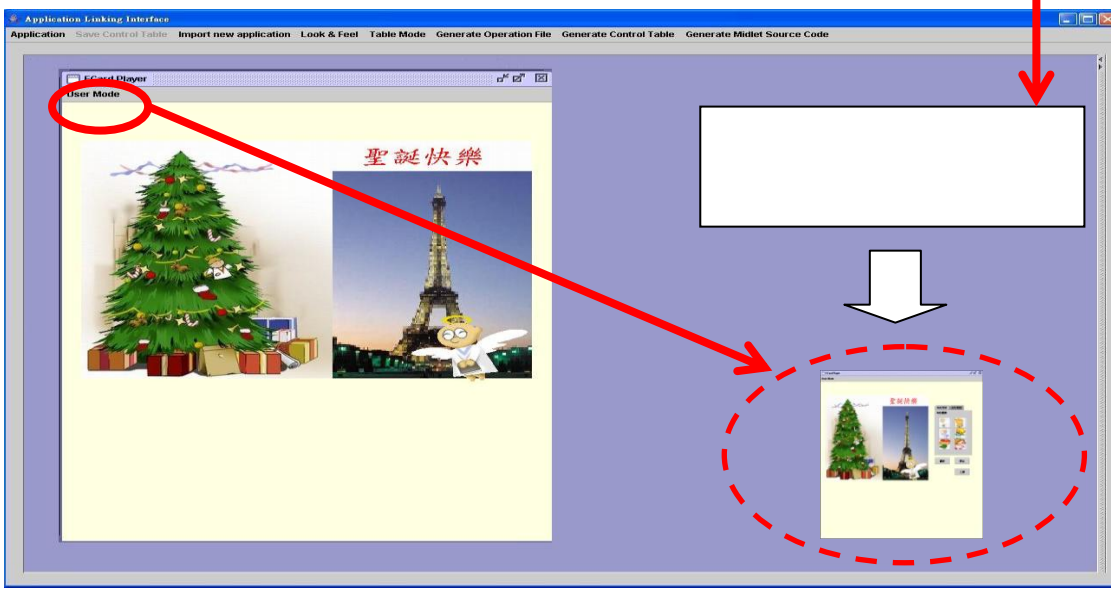


圖 86、電子賀卡樣板編輯器系統畫面載入結果

圖 83 為電子賀卡樣板編輯器系統的操作畫面，用以說明相關的操作元件。

圖 87、電子賀卡樣板編輯器系統的操作畫面說明

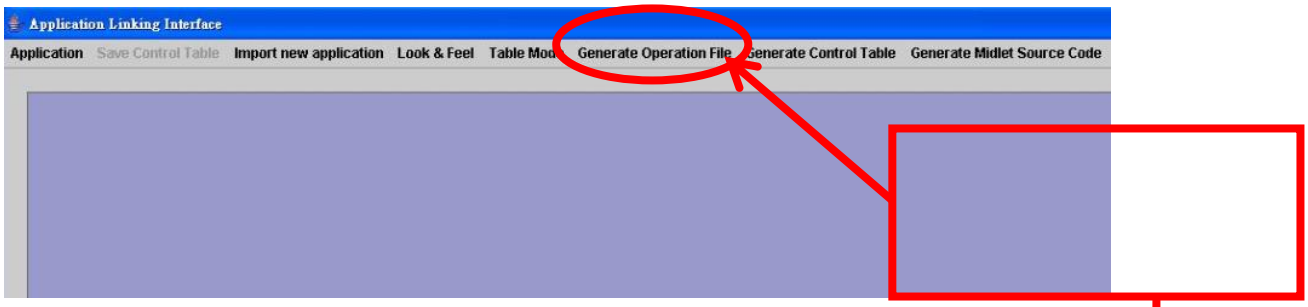


圖 88、點選 Generate Operation File 以產生操作描述檔案

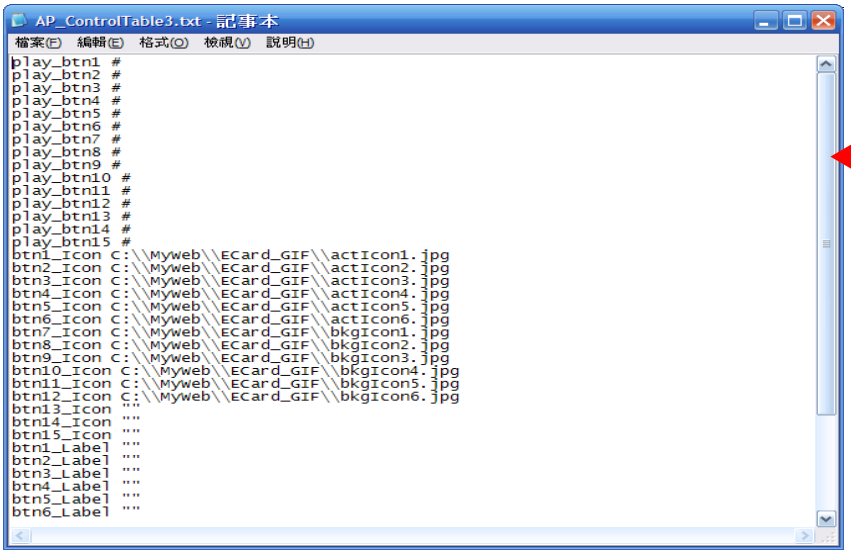


圖 89、描述檔案的內容

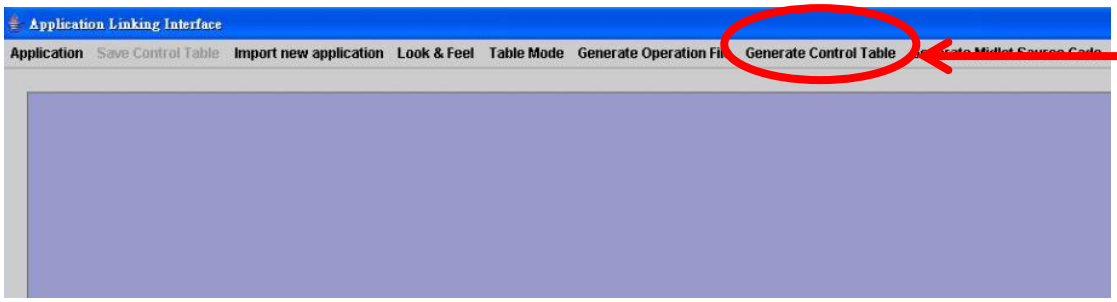


圖 90、點選 Generate Control Table 以產生控制表單

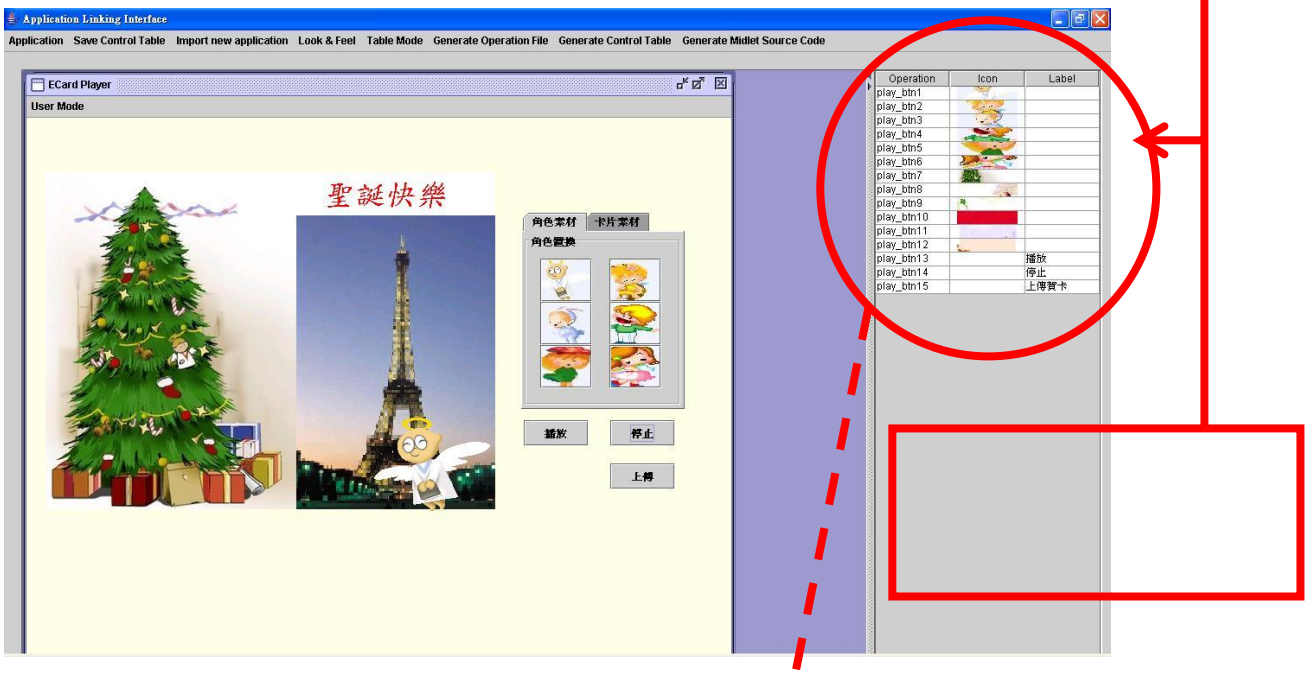


圖 91、控制表單的呈現畫面

Operation	Icon	Label
play_btn1		
play_btn2		
play_btn3		
play_btn4		
play_btn5		
play_btn6		
play_btn7		
play_btn8		
play_btn9		
play_btn10		
play_btn11		
play_btn12		
play_btn13		播放
play_btn14		停止
play_btn15		上傳賀卡

圖 92、控制表單的內容

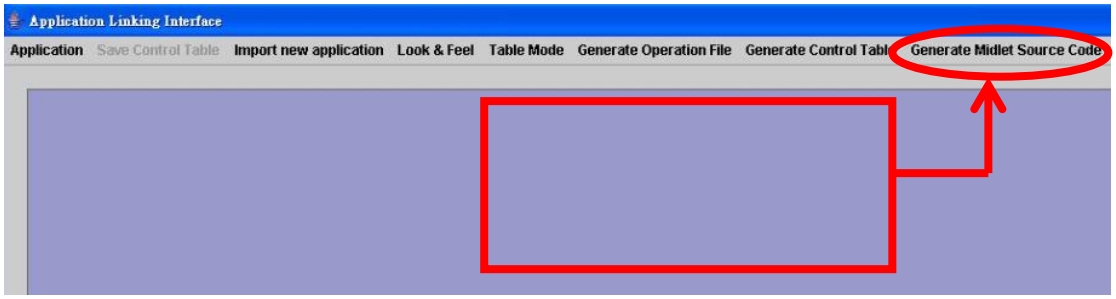


圖 93、點選 Generate MIDlet Source Code 並執行 WTK 編譯器

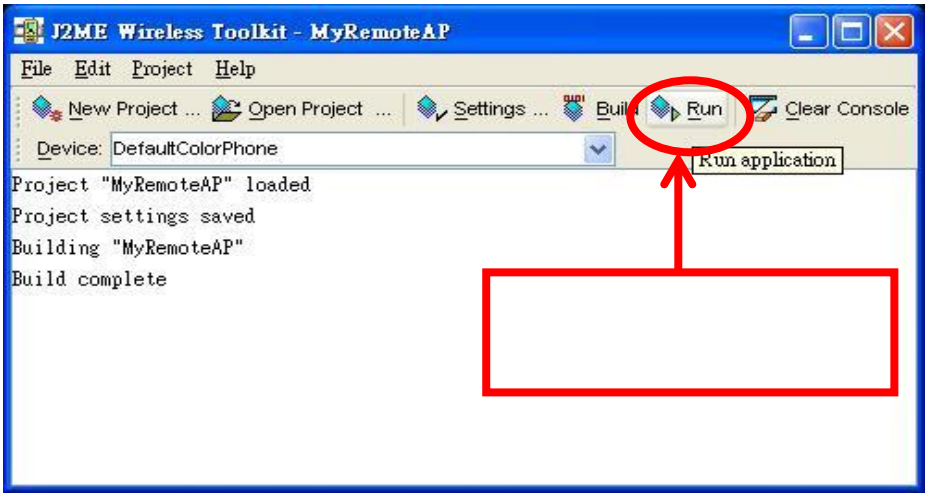
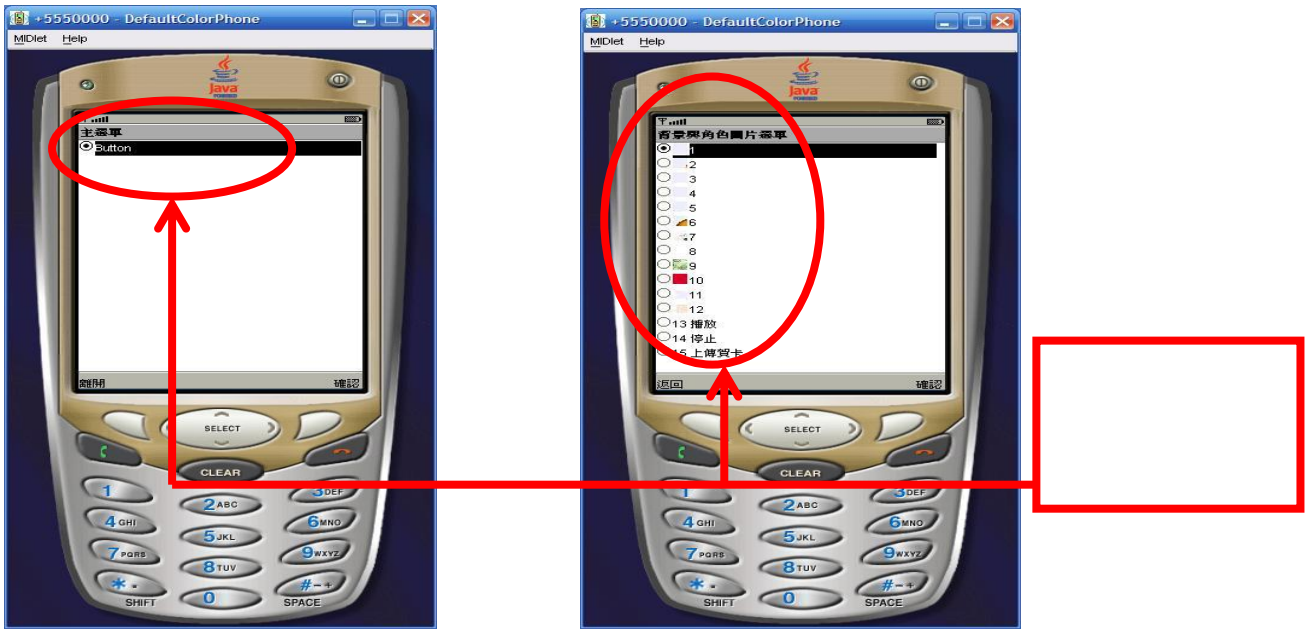
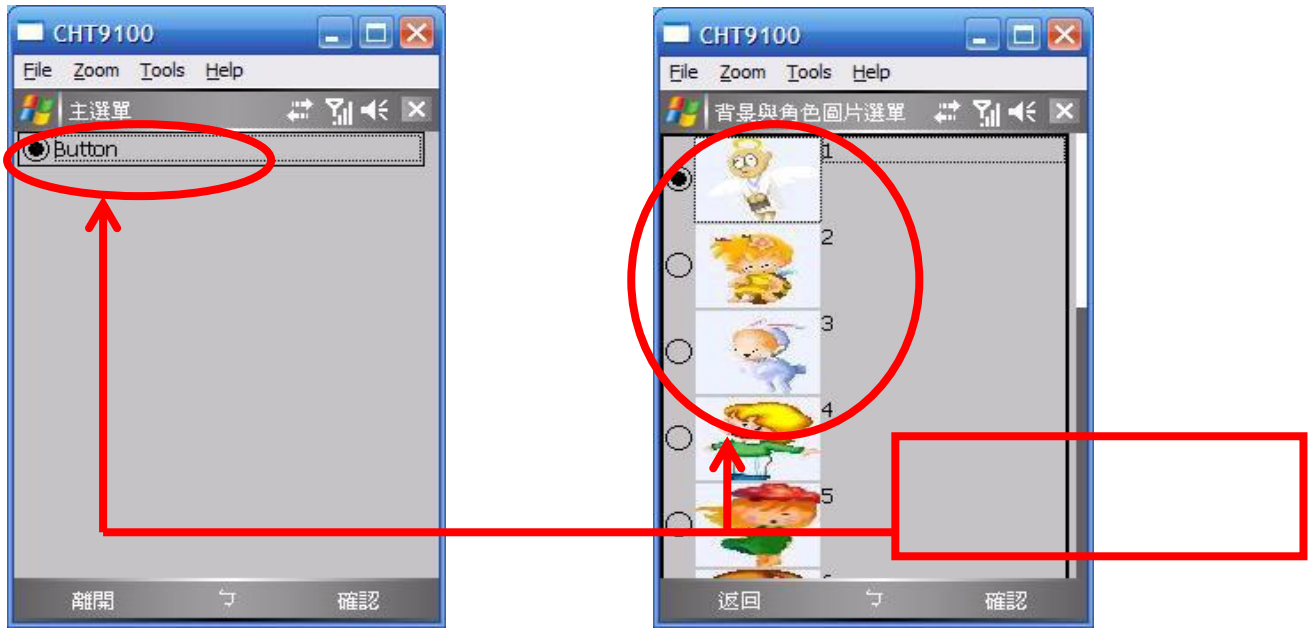


圖 94、編譯產生的 MIDlet 程式並執行手機模擬器



(a) 操作項目主選單-按鈕元件　　(b) 操作項目的內容選單-按鈕元件

圖 95、執行手機模擬器的畫面



(a) 操作項目主選單-按鈕元件　　(b) 操作項目的內容選單-按鈕元件

圖 96、PDA Phone 上的 MIDlet 程式執行畫面

圖 97 是終端使用者在手機上進行遙控操作 PC 上的電子賀卡樣板編輯器，在手機操作介面上的按鈕與電子賀卡樣板編輯器的操作畫面上，圈框起的部份是標示相同對應的操作功能：

圖 97、手機上遙控電子賀卡樣板編輯器-Button 按鈕操作

3.4.5 系統開發上的評估

表 16 是針對提供介面系統(Interfacing System)來開發 Java 應用軟體的機制，與一般開發 Java 應用軟體系統的開發方式比較，由於採用 Java 物件導向程式語言[17][19][20]的方式進行開發，我們可以定義繼承的關係以及提供抽象類別的製作來達成軟體製作上生產力(Productivity)與維護力(Maintainability)的改善。由表 16 的開發方式得知，撰寫抽象類別來規範程式樣板，提供良好的封裝與繼承的特性，程式設計師相對需要撰寫的程式碼變少，尤其是針對需要利用 GUI 來呈現服務功能的應用軟體，對於軟體介面上的製作，就更具彈性，開發也較快速。

表 16、開發方式的比較

開發方式 / 軟體製作	一般開發 Java 應用軟體系統的過程	提供介面系統(Interfacing System)來開發 Java 應用軟體
Productivity	程式設計師需要撰寫程式碼，才能設計出軟體。	根據定義的程式樣板規則，程式設計師只需撰寫抽象類別提供給介面系統，需要撰寫程式碼變少。
Maintainability	應用軟體欲增加新的功能時，需要重新撰寫新的功能模組。	應用軟體欲增加新的功能時，可能不需要重新撰寫。

表 17 是針對在手機上的 Java 應用程式開發比較，本研究區分了四個主要的評估項目來進行分析，比較利用本計畫所提出的介面化機制(Generic Interfacing Interface Mechanism)與現行 Java 的軟體開發(Current Java AP Development Approaches)，有何不同並具那些優點。

表 17、評估項目的比較

	評估項目	Generic Interfacing Interface Mechanism	Current Java AP Development Approaches
針對手機上的 Java 應用程式開發	Productivity	由介面產生器自動產生 MIDlet 程式，因此不需撰寫手機上的操作介面程式。	需要撰寫手機上的操作介面程式。
	Maintainability	不需要重新撰寫互動的控制程序，開發上較易維護。	需要重新撰寫或改寫互動的控制程序，開發上不易維護。
	Flexibility	能分析應用軟體的物件屬性，自動產生手機上對應的操作功能，開發方式較為彈性。	需要設計手機上對應的操作功能，開發方式較不彈性。
	Efficiency	不需要重新設計手機操作程式與重新改寫被操作裝置上的軟體。	手機與被操作的裝置上，都需要重新開發應用軟體，才能提供新的操作功能。

由表 17 的評估項目比較得知，針對手機上的 Java 應用程式開發，在生產力、維護力、彈性以及效率性這四個主要的項目上，由於提供了能夠產生手機內的 Java 程式之介面產生器系統，來為目標裝置上自動產生 Java 應用 MIDlet 程式，比現行開發 Java 軟體的方式，更能夠提升上述四點的能力。

3.5 Conclusion

3.5.1 結論

本計畫提出一個遠端控制介面系統的架構(remote control interfacing system framework)，可以為特定裝置上的 Java 應用軟體，自動產生手機上的遙控操作介面(remote control interface)，利用該架構開發的機制將使系統開發者免於重新撰寫手機應用程式，以及減少特定裝置上的 Java 軟體開發時間，以下是該系統提供幾項主要的特色：

(1) 手機內的 Java 程式之介面產生器

該介面系統(interfacing system)的核心元件，就是提供一個手機內的 Java 程式之介面產生器(interface generator for MIDlet program)，這個產生器能夠剖析開發者撰寫的程式樣板(即抽象類別)，並根據該程式樣板來產生操作描述檔案

描述檔(operation script file)與控制表單(control table)，利用控制表單賦予每個操作元件一個命令碼(command ID)，最後產生 MIDlet 程式及呼叫 WTK 編譯器進行編譯手機程式，將編譯好的程式包裝成 jar 檔，即可安裝至手機並且執行遙控的服務。

(2) 遙控操作的服務

本計畫所產生的手機操作介面程式，結合手機上觸控及按鍵的操作能力，能夠藉由手機內的 Java 程式之介面產生器所產生的操作介面，來遙控操作 PC 上相同的 Java 應用軟體。連線以及傳輸控制命令的協定，皆由手機內的 Java 程式之介面產生器自動產生出來，系統開發者只需在應用程式介面載入器中，處理接收到的 HTTP 連線，及撰寫接收到控制命令所需呼叫的事件程序即可。

(3) 程式樣板的規格制定

針對 Java 應用軟體的開發過程，本計畫分為手機上的應用程式與 PC 上的應用軟體為主，手機上的應用程式不需要規範撰寫程式的樣板，可由產生器自動產生，所以必須制定 PC 上的開發方式，本計畫針對三種操作元件(JButton、JRadioButton、JComboBox)，定義三種抽象類別的實作規則，當系統開發者依據這樣的規則去撰寫抽象類別，來定義 Java 應用軟體的程式樣板，則產生器就能自動產生出手機操作介面程式。

(4) 應用軟體的多元服務變化

由於提供抽象類別與抽象的方法實作，程式樣板可以進行重覆開發，提供良好的再利用性(reusable)，套用這樣的開發機制，我們可以重覆開發應用軟體，所以能夠快速並容易的產生出一個新的服務。

(5)減少開發上的負擔

在第 3.4 節中進行系統開發上的評估，本研究採用質化的方式，針對開發上幾點特定項目進行比較 ，由比較的結果得知，一般開發 Java 應用軟體系統與採用介面系統(interfacing system)來開發 Java 應用軟體，從軟體製作過程中的生產力與維護力主要的兩個維度來觀察，確實縮短了開發者的開發時間與程序，而針對手機上的 Java 應用程式開發比較，由於提供了能夠產生手機內的 Java 程式之介面產生器系統，來為目標裝置上自動產生 Java MIDlet 應用程式，比現行開發 Java 軟體的方式，更能夠提升生產力、維護力、彈性與效率性。

3.5.2 未來展望

本介面系統目前採用的 Java 開發平台是針對 PC 來進行實作，由於 Java 能夠跨平台的特性，及相關的 Java API 規格支援越來越多，本研究未來希望可以將介面系統(interfacing system)移植到數位電視、車用多媒體系統以及廣告看板系統

上，在此我們將提出未來可能發展的方向及尚未解決的問題：

(1) 介面系統能提供更多應用服務上的變化，例如擴增抽象類別的定義與操作檔案的描述規則，在本研究中只定義了三種基本的 Swing 操作元件：(JButton、JRadioButton、JComboBox)，未來則希望能夠制定更多的 Swing 操作元件。
(2) 考慮結合手機的語音或影像辨識系統，提供前端的行動裝置能有更多元的互動能力。
(3) 進行遠端遙控大型廣告看板及數位電視(Java TV)，支援其他相容的 Java 平台。
(4) 手機內的 Java 程式之介面產生器進行適性化的協調(Content Adaptation)機制，改善手機應用軟體介面的設計。

四、參考文獻

[1] *Norman Makoto Su,Yutaka Sakane, Masahiko Tsukamoto,Shojiro Nishio,*September 2002 ,“Systems Issues: Rajicon:: remote PC GUI operations via constricted mobile interfaces” , International Conference on Mobile Computing and Networking, Proceedings of the 8th annual international conference on Mobile computing and networking.

[2] *H. Okada, K. Kato, T. Ikegamai, Y. Tatusmi, and T. Asahi.* Proposal of PC Remote Control System by Mobile Devices. IPSJ Sig Notes(2001-HI-93), 2001(38):1–6, 2001.

[3] *Jeffrey Nichols, Brad A. Myers, Michael Higgins, Joseph Hughes, Thomas K. Harris, Roni Rosenfeld, Mathilde Pignol,* October 2002,“Papers: infrastructure for ubicomp: Generating remote control interfaces for complex appliances”, Symposium on User Interface Software and Technology, Proceedings of the 15th annual ACM symposium on User interface software and technology.

[4] *Abrams, M., Phanouriou, C., Batongbacal, A.L., Williams, S.M., and Shuster, J.E.* “UIML: An Appliance-Independent XML User Interface Language,” in The Eighth International World Wide Web Conference. 1999. Toronto, Canada.

[5] *de Baar, D.J.M.J., Foley, J.D., Mullet, K.E.* “Coupling Application Design and User Interface Design,” in Conference on Human Factors and Computing Systems. 1992. Monterey,California: ACM Press. pp. 259-266.

[6] *Nichols, J.W.* “Using Handhelds as Controls for Everyday Appliances: A Paper Prototype Study,” in ACM CHI'2001 Student Posters. 2001. Seattle, WA: pp. 443-444.

[7] *Nichols, J., Myers, B.A., Higgins, M., Hughes, J., Harris,T.K., Rosenfeld,R.,Shriver, S.* "Requirements for Automatically Generating Multi-Modal Interfaces for Complex Appliances," in ICMI. 2002.

[8] *Olufisayo Omojokun, S. Pierce, L. Isbell, Prasun Dewan* ,January 2006, "Comparing end-user and intelligent remote control interface generation", Personal and Ubiquitous Computing , Volume 10,Issue 2.

[9] *Neil R. N. Enns, I. Scott MacKenzie,*April 1998,"Touchpad-based remote control devices", Conference on Human Factors in Computing Systems , CHI 98 conference summary on Human factors in computing systems.

[10] *Jeffrey Nichols* ,April 2002 ,"Student Posters: Informing automatic generation of remote control interfaces with human designs", Conference on Human Factors in Computing Systems, CHI '02 extended abstracts on Human factors in computing systems.

[11] *Azam Khan, George Fitzmaurice, Don Almeida, Nicolas Burtnyk, Gordon Kurtenbach* October 2004 ,"Large public displays: A remote control interface for large displays", Symposium on User Interface Software and Technology, Proceedings of the 17th annual ACM symposium on User interface software and technology.

[12] *Desai N et al.* (2002) Automated selection of remote control user interfaces in pervasive smart spaces. In: Pervasive smart spaces HCIC winter workshop.

[13] *John Jones* ,February 2002 ,"Middleware: DVB-MHP/Java TV data transport mechanisms", ACM International Conference Proceeding Series; Vol. 21, Proceedings of the Fortieth International Conference on Tools Pacific: Objects for internet, mobile and embedded applications.

[14] *Eddie Schwalb* ,April 2004 "Synopsis - Books and Software: iTV handbook: technologies & standards", Computers in Entertainment(CIE), Volume 2 , Issue 2

[15] *Anind K. Dey , Gregory D. Abowd* ,"Towards a Better Understanding of Context and Context-Awareness",1999.

[16] *Janet L. Wesson,Darryn F. van der Walt,*2005,"Implementing mobile services: does the platform really make a difference?" ACM International Conference Proceeding Series; Vol. 150,pages: 208 - 216, Proceedings of the 2005 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries.

[17] *Shih-Kun Huang,* "Objected-Oriented Program Behavior Analysis Based on Control Patterns";*a Ph. D. dissertation, Computer Science and Information Engineering, National Chiao-Tung University, Taiwan, 2002.*

[18] *W. C. Chen,* "A Reuse-based Software Construction Paradigm for Visualized Reusable Components and Frameworks"; *a Ph. D. dissertation, Computer Science and Information Engineering, National Chiao-Tung University, Taiwan, 1998.*

[19] *J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy and W. Lorensen*; Object-Oriented Modeling and Design, *1991 Prentice-Hall.*

[20] *Grady Booch*; Object-Oriented Analysis and Design with Applications, *The Benjamin/Cummings Publishing Company, Inc., 1994.*

[21] *Jones J.*, DVB/MHP Java™ data transport mechanisms,Proceedings of the 40th International Conference on Tools Pacific: Objects for internet and embedded applications,Volume 10, 2002, pp: 115-121.

[22] *Cesar P. and Vuorimaa P.*, A Graphical User Interface Framework for Digital Television, Proceedings of the 10th International Conference on Computer Graphics, Visualisation and Computer Vision, February 2002, Czech Republic, Posters,pp. 1-4.

[23] Microsoft corporation. Universal plug and play forum. http://www.upnp.org/

[24] API specification for the Java 2 Platform, Standard Edition, version 1.4.2. http://java.sun.com/j2se/1.4.2/docs/api/

[25] Java ME Technology API Documentation, MIDP 2.0 http://java.sun.com/javame/reference/apis/jsr118/

[26] A Java/Swing GUI Framework http://www.newt.com/java/swing.html

[27] Design Patterns in Java. Available: http://www.fluffycat.com/java/patterns.html

[28] An Introduction To MHP http://www.interactivetvweb.org/tutorial/mhp/index.shtml

[29] Java TV API 1.1 (JSR-927) http://java.sun.com/javame/reference/apis/jsr927/

[30] Introduction to Digital TV Applications Programming http://java.sun.com/developer/technicalArticles/javatv/apiintro/

[31] Java Media Framework API http://java.sun.com/products/java-media/jmf/2.1.1/download.html

[32] Java ME Technology
http://java.sun.com/javame/technology/index.jsp
[33] B. Balentine, D. Morgan, and W. Meisel, How to Build a Speech Recognition Application, Enterprise Integration Group, 1999.
[34] Speech-Actuated Manipulator, http://www.research.att.com/history/89robot
[35] VSpeech 1.0, Team BK02 product, http://vspeech.sourceforge.net.
[36] Voxx 4.0, Voxx Team product, http://voxxopensource.sourceforge.net.
[37] Microsoft's Speech Recognizer V.6.1, Microsoft product, http://www.microsoft.com.
[38] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose, "Toolglass and magic lenses: the see-through interface," Xerox PARC, 3333 Coyote Hill Road, Palo Alto, CA 94304.
[39] Y. Boussemart, F. Rioux, F. Rudzicz, M. Wozniewski, and J. R. Cooperstock, "A framework for 3D visualization and manipulation in an immersive space using an untethered bimanual gestural interface," Centre For Intelligent Machines 3480 University Street Montreal, Quebec, Canada.
[40] S. K. Huang, "Objected-oriented program behavior analysis based on control patterns," a Ph.D. Dissertation, Department of Computer Science and Information Engineering, National Chiao Tung University, Taiwan, 2002.
[41] R. W. Sebesta, Concepts of Programming Languages, 5th ed., Addison-Wesley Publishing Company, 2002.
[42] J. Gosling, B. Joy, G. Steele, and G. Bracha, The Java Language Specification, 2nd ed., Sun Microsystems, Inc., 2000.
[43] BestWise International Computing Company, http://www.caidiy.com.tw.
[44] J. K. Ruzicka, "The design and implementation of an interfacing framework for bridging speech recognizer to application system," a Master Dissertation, Department of Computer Science and Information Engineering, National Chiao Tung University, Taiwan, 2005.
[45] S. J. Peng, "Bridging the interface between application systems and recognizers,"
Technical Report No. NCTU-CSIE-SE-TR-001, Department of Computer Science and Information Engineering, National Chiao Tung University, Taiwan, 2005.
[46] WinBatch Macro Scripting Language, http://www.winbatch.com/.
[47] B. P. Douglas, Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems, Addison-Wesley Publishing Company, 2003.
[48] Microsoft Speech SDK, Version 5.1 Documentation, Microsoft Corporation, 2001.

[49] E. Lee, “User-interface development tools,” *IEEE Software*, Vol. 7, 1990, pp.31-36.

五、計畫成果自評

有關計畫成果可分為論文發表及專利申請兩大方向來自評。有關論文發表方面，附於 5.1，成果發表如下

5.1 Papers publication

1) Shih-Jung Peng, Jan Karel Ruzicka and Deng-Jyi Chen, “A Generic and Visual Interfacing Framework for Bridging the Interface between Application Systems and Recognizers,” Journal of Information Science and Engineering, Vol. 22, No.5, September 2006, pp.1077-1091 .(SCI)

2) Deng-Jyi Chen, Shih-Jung Peng and Chin-Eng Ong, “Generate Remote Control Interface Automatically into Cellular Phone for Controlling Applications running on PC”, Journal of Information Science and Engineering, Vol.26, No.2, Mar. 2010, pp549-563

3) Shih-Jung Peng and Deng-Jyi Chen, “A Generic Interface Methodology for Bridging Application Systems and Speech Recognizers,” 2007 International Conference on Information, Communications and Signal Processing (IEEE ICICS2007), 10-13 December, 2007, in Singapore

4) Ming-Jyh Tsai, Deng-Jyi Chen, Chi-Chung Hung and Hao-Chun Lu, 2009, Context Aware and Content Adaptation of Template Based Multimedia Presentation on Handset Devices, Advances in Multimedia Information Processing - PCM 2009, December 15-18, Bangkok, Thailand.

5) Ah-Fur Lai, Deng-Jyi Chen, Web-based Two-tier Diagnostic Test and Remedial Learning Experiment, *International Journal of Distance Education Technologies*, Vol. 8, Issue 1, 2010, pp. 31-55.

6) Chung-Yueh Lien, Hsu-Chih Teng, Deng-Ji Chen, Woei-Chyn Chu, and Chia-Hung Hsiao, “ A Web-Based Solution for Viewing Large-Sized Microscopic Images” Journal of Digital Imaging, Published online: 27 June 2008, 0897-1889 (Print) 1618-727X (Online). doi: 10.1007/s10278-008-9136-x

7) Chen, D.J., Lai, A.F., Chiu, H.H., & Cheng, H.L., 2008, The Design and Implementation of On-line Video-based Oral Test Management System and its Experiment on English Speaking Proficiency Test, Proceedings of Technology Enhanced Learning Conference 2008, Dec 4-6,2008, Hanoi, Vientam.

8) Chu-Hsing Lin, Chen-Yu Lee, Deng-Jyi Chen, "Modified Autonomous Key Management Scheme with Reduced Communication/Computation Costs in MANET," accepted by Computing and Informatics, 2011. [SCI]

專利發表方面，附於 5.2，成果發表如下。

5.2 Patents

1) 陳登吉, 彭士榮, 蔣加洛, "介面系統、方法與裝置", patent No I 299457, 2008.08.01.~2025.12.19. For Taiwan.

2) Deng-Jyi Chen, Shih-Jung Peng, Jan Karel Ruzicka, "Interface System, Method and Apparatus," patent application No 20070150280A1, 2007.06.28. For USA. (pending)

3)多媒體簡訊樣板套用系統及播放系統、多媒體簡訊樣板套用方法及播放方法, 第 I 292667,中華民國(台灣),陳登吉 洪啟彰 楊博鈞,2008/01/11 to 2025/12/13.For Taiwan

4) D. J. Chen, Chen-Yu Lee, Liang-Hsiao Chao, Sang-Hue Yen, Chia-Hung Hsiao, Chien-Chao Tseng, Ti Chuang Chiang "多媒體註解書籤方法及其系統", 申請號: 099138516 (申請日 2010.11).