

二、相關研究

在這一章中，我們將對使用者介面，它的開發流程與開發方式做一個討論。在2.1節將對使用者介面做介紹。在2.2節中將討論使用者介面需求。在2.3節中將討論開發流程。在2.4節中將討論開發方式。

2.1 使用者介面

使用者介面 (User Interface)，為電腦系統或程式用來與使用者溝通的媒介，使用者可以透過輸入介面對電腦或程式下達指令，而電腦或程式再依指令執行相對應的功能。一個基本的使用者介面強調與使用者的互動，透過輸出介面讓使用者得知下達指令後的結果。[4]

因此基本的使用者介面需具備：

- 輸入
- 輸出



輸入方式能以多種方式，例如鍵盤，滑鼠，聲音…等，只要輸入能被機器認知即可。而同樣的，輸出方式也能以多種方式，只要輸出能被使用者認知即可。

從早期的使用者介面演變至今，已從單調的文字介面 (Command line)，進化至圖形化的使用者介面 (Graphic User Interface)，使用者可以輕鬆的點選圖形介面上的圖示 (Icon) 來執行指令，而不用像在早期輸入一長串的命令。

因此圖形化使用者介面已被廣泛的應用在個人電腦上，如 UNIX 作業系統的上的 X-window 視窗環境，Microsoft 的 Windows 系列，Apple 的 Mac OS 等。

好的使用者介面設計具有美觀，讓使用者減少輸入錯誤的發生，提供正確的文字或是圖像資訊，提供及時性的反應，與功能有著緊密的連結[3] [4]。能讓使用者能快速方便的使用機器的功能，減少學習的時間。

2.2.1 行動電話上之使用者介面

在行動電話上的使用者介面由於應用環境上的限制，講求體積小，攜帶方便，使用上的時間要長。

在輸入裝置通常採用小型的數字鍵盤或觸控式螢幕，輸出裝置則使用小尺寸的液晶螢幕，而為了長時間使用，其運算能力也受到了限制，以節省電力上的消耗。

因此在行動電話上的使用者介面呈現或操作上並不像桌上型電腦，提供了全尺寸的鍵盤及滑鼠來方便輸入，及提供了強大的多媒體功能，在行動電話上通常只能在小尺寸的數字鍵盤上作輸入，而輸出結果在小尺寸的螢幕上，因此在開發上限制較多。
[11]

且行動電話講求即時的反應，對使用者的輸入或系統內部發生的事件，都要能迅速的反應，因此大多採用核心 (Kernel) 較小的即時作業系統 (Real-time Operating System)，如 Nucleus OS、uC/OS-II 等，不過這類的 RTOS 只負責基本的功能，如記憶體管理、排程管理，若使用者介面要與其它硬體作溝通，如要控制撥號功能，則直接操作其硬體相關之函式呼叫 (function call)，並不透過作業系統。

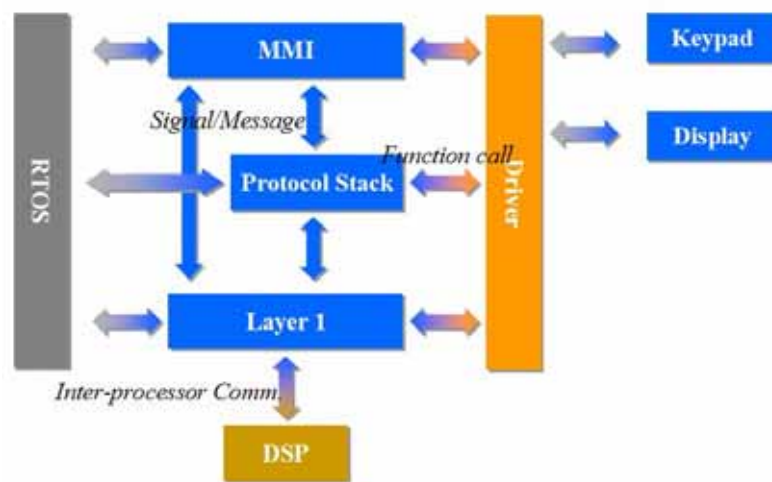


圖 3 現今行動電話軟體架構

如圖 3 所示，當使用者介面要操作或接收從按鍵(Keypad)或面板(Display)所傳過來的訊息時，便直接透過相對應的驅動程式作函式呼叫。

行動電話上的使用者介面除了要與作業系統溝通外，還需與硬體直接作溝通。在這樣的架構下，當硬體更動時，需更換相對應的驅動程式，或者是更換了作業系統，因此連帶的使用者介面也需要做調整，無法直接延用之前開發的結果。

2.2 使用者介面需求

2.2.1 需求

在建構系統的過程中，首先的步驟是將需求明確定義出來，系統將需要完成什麼功能。如圖 4 軟體工程上軟體開發方式的 Waterfall Model[2]，第一個步驟為定義需求：

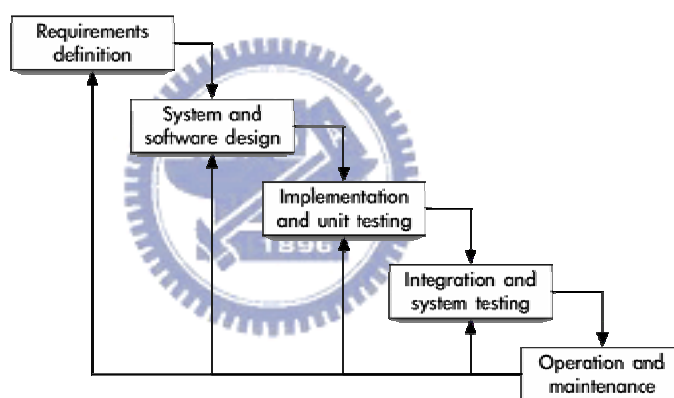


圖 4 Waterfall Model

而要明確的定義需求是相當困難的，使用者往往不知道他們要的是什麼，因此必需藉著 requirements engineering 的幫助。經過 feasibility、requirements elicitation and analysis, requirements specification, requirements validation 正規的步驟將需求找出，形成正式的需求文件。

2.2.2 視覺化需求

在 requirements elicitation 步驟中透過與使用者互動，分析來了解他們的需求，以文字、圖片等靜態方式記錄下來。但是透過以上的方式是否能百分之百找出使用者的需

求呢？唯有將系統實作出來後，經過使用者實際的接觸與操作系統後才能確定此軟體是否真正滿足使用者的需求。[16]

爲了儘可能將需求導引出來，可以將系統中的行爲、子系統間的互動，之間的關係，以圖示、文字、線條，將需求視覺化出來[17]，這樣稱之爲視覺化需求。透過這樣的方式，在需求定義階段就能夠給使用者得知系統未來上線的大致情形，讓使用者在此階段能夠完整將需求表達出來，不至於到實作完成後才發覺需求並未被滿足，再度修正需求，讓 requirements engineering 發揮最大的效用。

2.2.3 傳統的使用者介面需求

在傳統的使用者介面開發上，我們使用者對於使用者介面上的需求，在定義需求階段，以畫面和流程圖等介面規格 (UI Specification) 將使用者介面的運作情形以文字和圖片記錄下來[4]。

圖 5 爲使用者介面的畫面規格，包括了：

- 視窗大小。
- 圖示大小、位置。
- 文字大小、位置、字型。

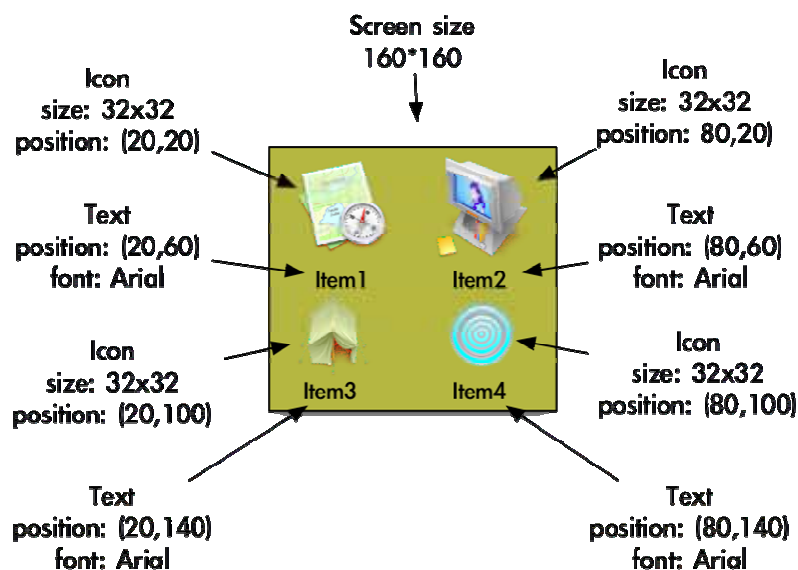


圖 5 畫面規格

我們用圖片與文字規格記錄了各個畫面的呈現方式，而各個畫面之間的轉移則用流程圖來表示。

圖 6 為使用者介面實際運作上的流程，整個系統由 6 個畫面所構成，而各個畫面存在著關係，若二畫面存在著關係則用箭頭連接，如 UI2 接受到儲存指令則跳躍至 UI6，則用箭頭從 UI2 連接至 UI6。

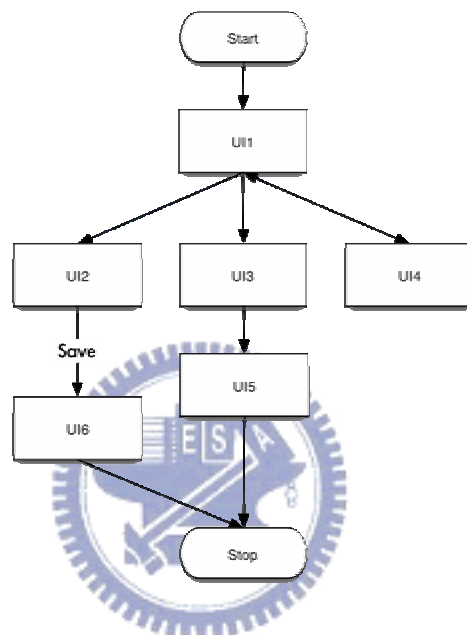


圖 6 使用者介面流程圖

用以上的方式來記錄使用者介面是屬於靜態的規格，會有 2.2.2 小節所提到的問題，無法完整的呈現實際上的運作情形，在實做後可能會與想像中的使用者介面有落差。

2.2.4 視覺化的使用者介面需求

為了解決使用者介面靜態規格與實作上的落差問題，我們可以利用視覺化需求的觀念，將這些使用者介面靜態規格視覺化動態呈現出來，並且能夠去操作，如圖 7。則在定義需求階段就能得知實作後的情形，對於使用者介面開發有相當的幫助。[6]

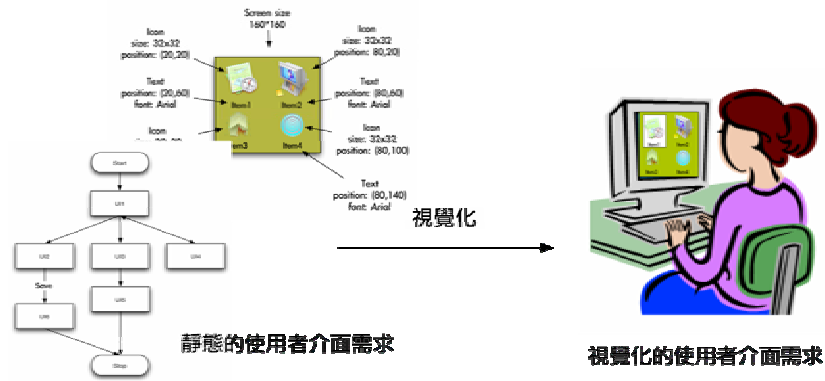


圖 7 視覺化的使用者介面需求

2.3 傳統使用者介面開發流程

在說明開發流程之前，首先說明在軟體開發過程中的人員組成。

一個專案團隊主要用下列成員構成:[9]

- 使用者：系統完成後的最終使用者。
- 市場人員：決定消費者需要什麼。
- 軟體設計者：將需求轉為實際上的規格。
- 使用者介面專家：又可稱之為使用者介面設計者 (UI designer)，協助開發使用者介面。
- 技術文件寫作者：撰寫系統技術文件。
- 軟體開發者：又可稱之為程式設計師，為實作系統。

圖 8 為成員關係圖：

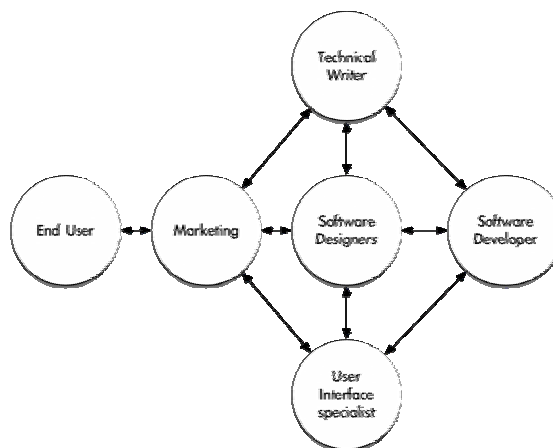


圖 8 成員關係

資料來源：[9]

使用者介面的開發，首先由軟體設計者與使用者介面設計者，將畫面的需求、操作流程等規格以文字、圖片之靜態的方式，定訂下來，並製作相關的媒體檔案，如顯示圖片，聲音等，之後交由程式設計師，依照規格將使用者介面實作出來，之後測試來驗證是否合乎需求。若實作出來不符合需求，則再更改規格，再交由程式設計師重新實作，直到需求滿足為止。其流程如圖 9：

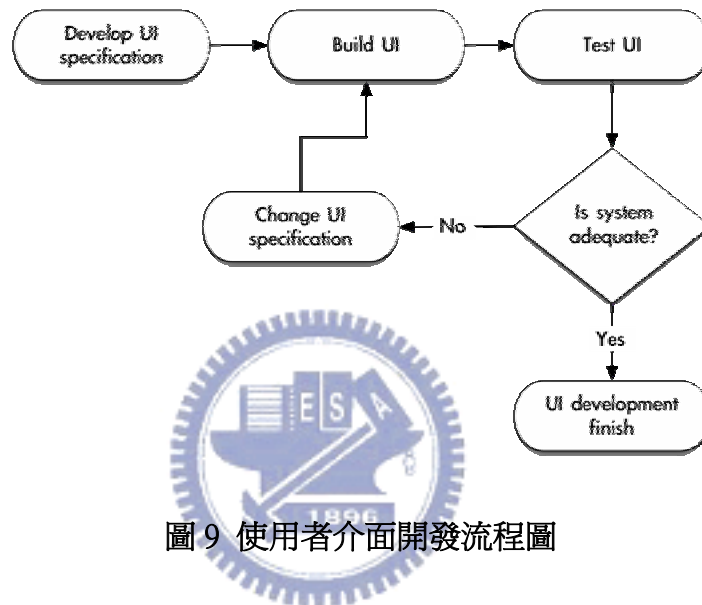


圖 9 使用者介面開發流程圖

在確認使用者介面是否符合需求是相當耗時的，需要等到程式設計師實作出來才能確認，雖然已有一些工具程式來協助使用者介面上的開發，如在 Windows 下開發程式用的 Microsoft Visual Studio、Borland C++ Builder，但仍需要程式設計師來操作與修改，於是將現今方法開發上的缺點歸納如下：

1. 靜態的規格並不能完整描述實際上運作的使用者介面，唯有實際操作後才能確定使用者介面是否符合需求。
2. 程式設計師除了設計系統功能外，還須要實作與修改使用者介面，工作負擔大。

2.4 相關系統分析與探討

現今在開發行動電話使用者界面的方法上，可分為三大類：

1. 傳統編程
2. 提供 framework 編程
3. 提供視覺化環境編程

在下面幾節一一作分析

2.4.1 傳統編程

在傳統編程中，整個程式是由程式設計師一行行將使用者界面建構起來，所需要的程式碼為最多，若規劃不良，則在修改維護時會非常的麻煩，如直接在 Nucleus OS 平台上編程。



2.4.2 提供 Framework 輔助編程

Framework 是一個完整的程式模型，具備標準應用軟體所需的一切基本功能，像是檔案存取、列印預視、資料交換…以及這些功能的使用介面[7]，利用 framework 輔助編程可降低寫程式所花的精力、加速開發效率、加強可維護性。

在目前在 Symbian OS 上，已有完整的 GUI framework 可供使用，利用它可迅速產生 Symbian OS 上的使用者界面標準元件，但需要以程式碼將使用者界面實作出來，無視覺化的環境來協助設計。

2.4.3 提供視覺化環境編程

視覺化環境編程提供了一個所見即所得 (What you see is what you get) 的開發方式，在設計階段所見到的使用者界面就等於執行時的畫面。程式設計師使用自行設計或者使用提供的標準使用者界面元件，以滑鼠以點選或拖拉的方式將使用者界面設計出來。如接下來介紹的兩套工具：

2.4.3.1 eMbedded Visual C++

Microsoft 除了在一般的個人電腦上，在行動電話也開發了適用於上的作業系統，稱之為 Windows Mobile 2003 Second Edition，它在此平台提供了完整的開發工具，如完整的 framework 與開發環境 eMbedded Visual C++，它還提供了模擬器可以用來實際驗證機器上的執行結果。如圖 10 所示。

在使用者介面上的編輯上，它提供了功能有限的編輯環境，只能用來設計選單部分，若欲要在畫面擺放一張圖片，則需要直接撰寫程式碼，無法自由的設計所想要的畫面。

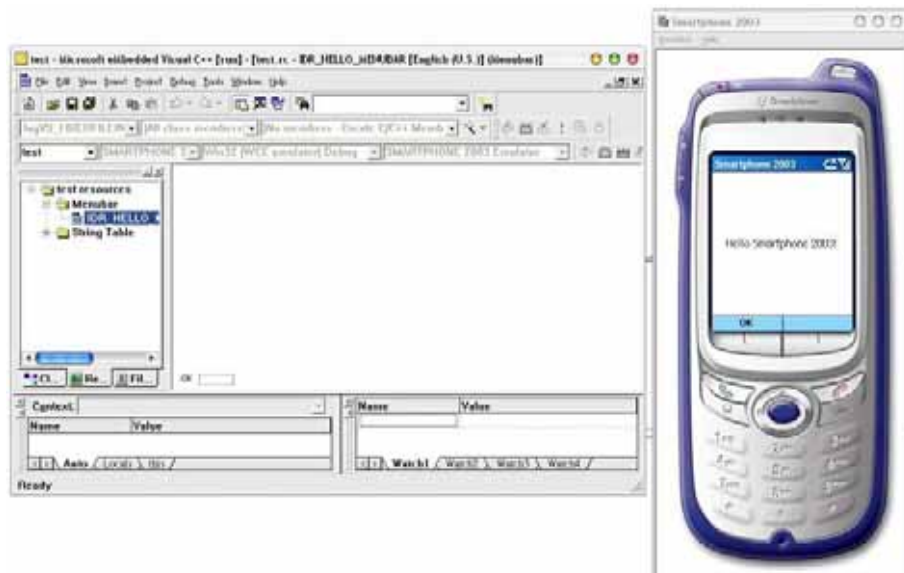


圖 10 eMbedded Visual C++

2.4.3.2 RAPID

Rapid 為 e-SIM 所開發出的一套產生跨平台嵌入式系統雛型及模擬的工具，在使用者介面上的開發上，它提供了以下的工具，如圖 11 所示。

- Object Layout：用來新增畫面上的物件或定義畫面上的物件位置
- Object Editor：用來修改物件的畫面

- Prototyper：用來驗證執行的結果

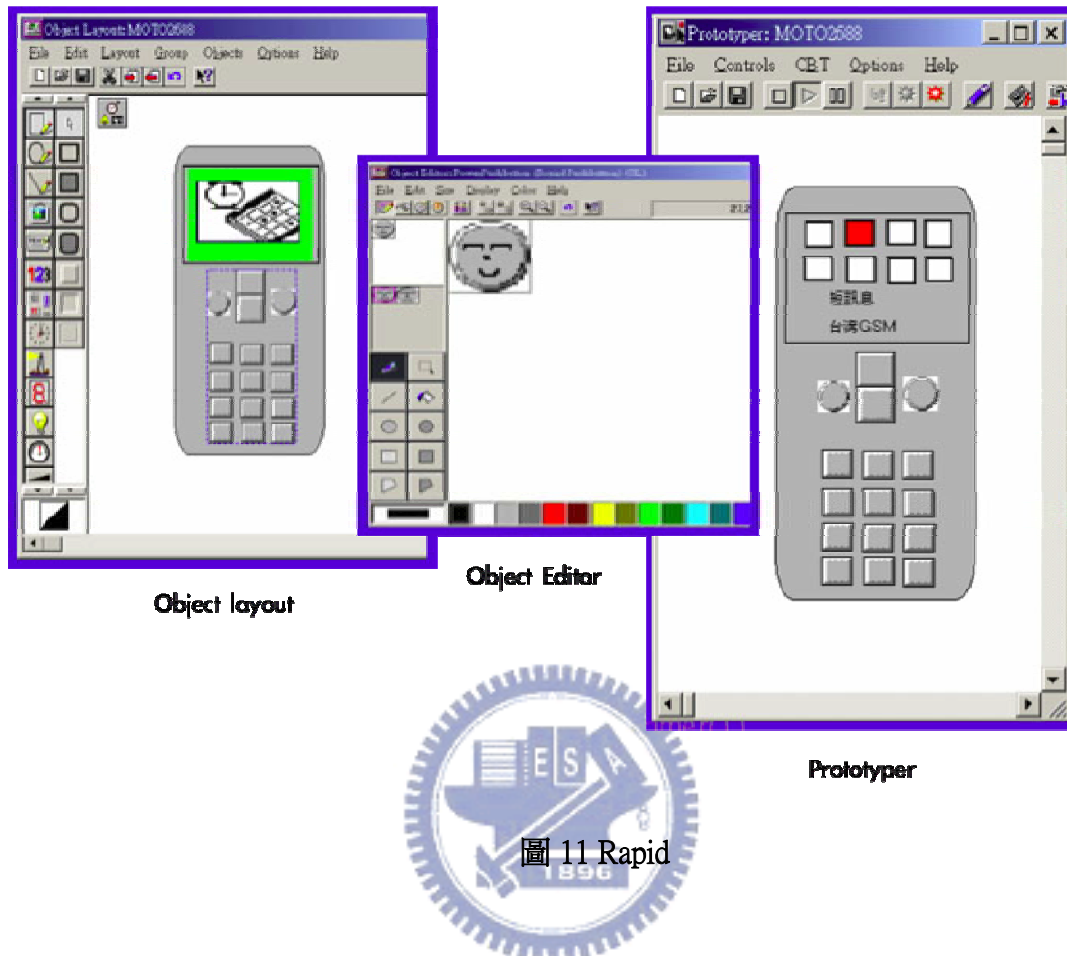


圖 11 Rapid

在使用 Rapid 開發使用者介面時，除了使用以上工具定義畫面上的物件外，還需要定義整個機器的各個狀態(state)，接下來定義各個狀態之間的轉移條件，如要完成一個撥號動作時，除了定義畫面上的按鍵外，還需定義出機器的撥號中狀態及未撥號中狀態，接下來設定按下畫面上的撥號鍵即觸發由未撥號中狀態進入撥號中狀態。

這種的開發方式，除了設計畫面，還必須了解整個機器的狀態與運作的細節，因此還是較為適合給程式設計師來使用。

2.5 使用者介面開發工具分析

從以上三種開發方式，可作一圖表來做比較

表 1 使用者介面開發方式比較表

	UI 設計難度	美術人員使用難度	後續 UI 維護難度
傳統編程	高	高	高
提供 framework 輔助編程	中	高	中
提供視覺化環境編程	低	中	低

由上述的使用者介面工具與開發方式分析，綜合以上可得到以下的問題：

- 使用者介面大部分仍需要以程式碼來完成
- 少部分雖提供了視覺化的開發方式，但仍需要由程式設計師來完成

一個適合開發行動電話使用者介面的方法，需要具備以下的條件：

- 充分達到專業分工，實作的責任不用完全放在程式設計師身上
- 設計、維護須方便
- 能快速的看到實際上的運作結果。
- 開發的成果能重用 (reusable)，增加開發的生產力。