

行政院國家科學委員會專題研究計畫 期中進度報告

手持裝置上的系統軟體之研製(2/3)

計畫類別：個別型計畫

計畫編號：NSC91-2213-E-009-044-

執行期間：91年08月01日至92年07月31日

執行單位：國立交通大學資訊科學學系

計畫主持人：袁賢銘

報告類型：精簡報告

處理方式：本計畫可公開查詢

中 華 民 國 92 年 6 月 2 日

摘要

無線網路技術與手持裝置的出現讓行動計算不再是遙不可及的夢想，不過由於目前並沒有一個統一的標準，所以呈現的是百家爭鳴的狀態，這種多樣性會導致軟體開發困難度的提升，如果想要開發一個能夠在各種不同平台上皆能執行的軟體，將會花許多的心力在移植與調整介面上，除此之外，手持裝置上的運算能力也不是十分強大，因此有些工作無法在上面進行處理。在本計畫中我們將提出一個可適性的行動應用程式開發平台，稱為 ART(Adaptive Remote Terminal)，來解決上述這些問題。

關鍵字：可適性，行動計算，終端機

Abstract

Mobile Computing is no longer an impractical dream due to the appearance of wireless network technology and handheld devices. However, there is a variety of platforms such that software development suffers from platform dependence. Lots of efforts have been wasted on tuning and adjustment for software to be adapted to different platforms. Furthermore, common handheld devices have weak computing power; more complex tasks hence cannot be worked out. In this project, we will propose a software development platform "ART", Adaptive Remote Terminal, to solve these problems.

Keyword: adaptation, mobile computing, remote terminal

第一章、前言

Remote Terminal 是一項很實用的技術，他可以讓使用者透過網路來操作遠端的電腦；不過目前這種技術卻被限制在 PC 上，雖然手持裝置和無線通訊正炙手可熱，但如果使用者想用手持裝置做到 Remote Terminal 的工作仍然是一件很困難的事。為了突破這樣的困境，讓用手持裝置的使用者也可以享受 Remote Terminal 的好處，於是 ART (Adaptive Remote Terminal) 便因蘊而生了。

我們所設計的 ART 希望能解決在手持裝置與無線網路上開發應用程式可能會遇到的幾個問題，如：無線網路的環境種類太多、資料傳輸目前並不可靠、計算、儲存、顯示能力比起 desktop 頗有不如、手持裝置規格多有不同、使用者可能希望在不同裝置上使用同一應用程式等等。

本報告在第二章闡述此研究的目標，第三章說明 ART 系統的分析與設計，第四章將說明 ART 系統整個的架構，最後一章會舉幾個應用於生活上的實例並作個總結與未來展望。

第二章、研究目的

我們在設計 ART 時，希望達成幾個目標：

1. 可以在不同的網路環境和裝置上執行，達到 **Any place, Any device** 的目標。初期我們是利用 MIDP 來達到這個目的。
2. 在同一個手持裝置上同時執行許多的應用程式，而不需擔心資源不足與無線網路斷線的情況發生。我們利用 remote terminal 的觀念，讓 client(手持裝置)只負責 UI 的顯示，而程式邏輯是在遠端的電腦上執行，大幅降低手持裝置的負擔；並且在 ART 中設計一個工作管理員，可以讓使用者輕易的在應用程式間切換。
3. 提供一個完整的無線網路應用程式開發平台，讓應用程式開發人員透過此一平台，更快速的開發無線網路的應用程式，大幅減少開發的時程。
4. 隱藏手持裝置種類與規格、網路環境、UI 等等與程式邏輯無甚相關，卻又必須耗費大量開發時間的工作，讓程式開發人員能夠有更多的時間和精神，設計更妥善完美的應用程式。

第三章、系統分析與設計

在我們設計的考量下，所有的應用程式邏輯都在 server 端上完成，等到執行完再把結果在 client 端的 UI 上做呈現。因此目前的 ART Framework 中，總共可分成五個角色：

1. ART Server (ART Application / Service Container) :

ART Server 負責 ART Application / Service 跟 ART Client 之間的溝通，並提供如 Application Management, Event Dispatch 等維護 Application 運作之必要功能。這個角色就有如 WWW 當中的 Web Server，如 IIS, Apache 等。

2. ART Client :

ART Client 便是真正在使用者手邊的手持裝置上執行的程式。任何使用者只要下載 ART Client 到自己的手持裝置上執行，就可以使用網際網路上的所有 ART Application/Service。ART Client 可在各式各樣的手持裝置上面執行，但就未來的展望而言，J2ME 將是最受歡迎的平台，於是我們目前先實作 MIDP 版本的 ART Client。這個角色就相當於 WWW 當中的 Browser，如 IE, Netscape 等。

3. ART Application / Service :

ART Application / Service 是跑在 ART Container 之中的應用程式，像是股票查詢程式，聊天室等等。ART Application / Service Provider 只要使用我們所提供的 API，就可以快速的開發出他們所需要的 ART Application/Service 給大家使用。這個角色就等同於 WWW 當中的網頁，如 HTML 檔案，或是用 JSP, ASP, PHP 撰寫的 Web 應用程式。

4. Mobile User :

任何持有手持裝置的使用者都可以叫做 Mobile User。Mobile User 透過 ART Client 使用網際網路上所有的 ART Application/Service。這個角色就等同於 WWW 當中的一般使用者。

5. ART Application/Service Provider (ART ASP) :

提供應用程式服務的公司或是程式開發者。ART ASP 透過我們所提供的 API 來提供 ART Application/Service。這個角色就等同於 WWW 當中提供網頁的公司或是網頁提供者，如 yahoo, 聯合新聞網等。

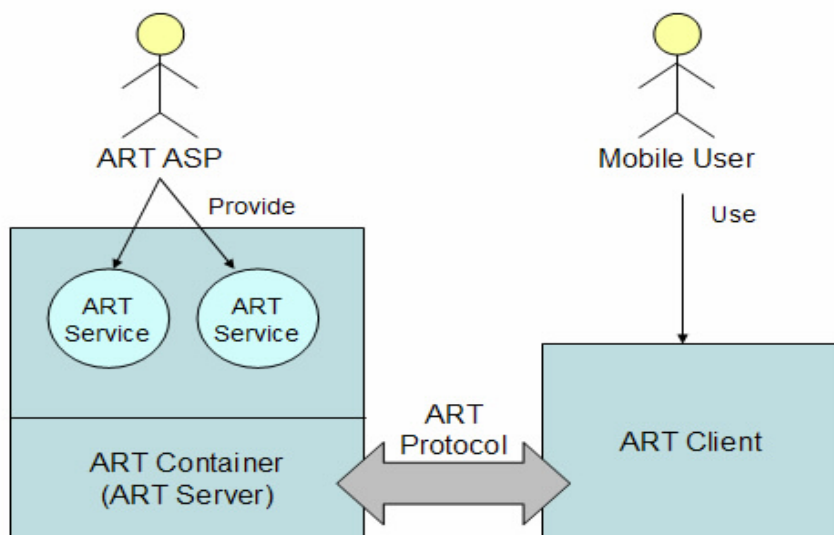


圖 3-1 ART Framework 中五個角色的關係

第四章、系統架構

ART 系統是採用 client-server 的架構，end-user 所接觸到的是 client(手持裝置)端，負責 UI 的顯示，而 server 端則負責程式邏輯的運算，client 與 server 兩者之間透過一個 asynchronous 的訊息傳遞機制來互相溝通。

為了未來的擴充性與彈性，我們將 ART 分成多層架構，每一層負責不同的工作，如圖 4-1 所示，如此一來就可以根據需求抽換某一層，而不需要全部改寫。其中 Transport 層負責網路上資料傳送的工作，目前 Transport 層我們提供 socket 和 HTTP 兩種版本，可以依照需求來選擇使用。Message Queue 則是提供訊息佇列的機制，提供給上層使用，而 ART server 提供應用程式的執行環境，ART client 則是負責顯示對應的執行畫面

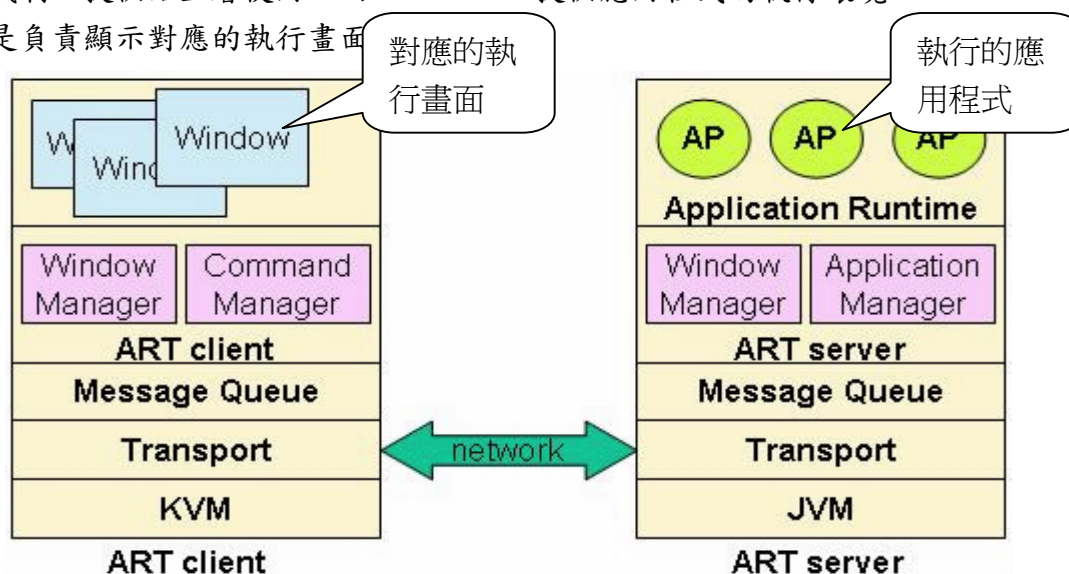


圖 4-1 ART 概念上的分層架構

接下來的內容將分別針對 message、server 和 client 這三大部分來作討論。

Asynchronous message delivery

這是 ART 系統中主要的溝通管道，在裡面傳送的資料稱為 ART message，可分成兩大類，一類為 ART 的系統訊息，另一類為應用程式訊息，分別介紹如下：

1. 系統訊息：簡單來說就是用來控制 ART 系統動作的訊息。
2. 應用程式訊息：負責將每一個應用程式訊息送到對應的應用程式去執行。這些訊息都會放置在 message queue 中等待 message handler 的處理。

ART Client (請參照圖 4-2 下半部)

1. Windows Manager(視窗管理員)：整個 ART server 中只有一個，負責控制 client 上所有的 widget 元件，並且設定對應的 listener。
2. Command Manager：負責提供一些基本的系統功能和產生 ART message

中的訊息，另外 listener 要送訊息回 server 也會透過它。

3. **Input Queue**：網路連線讀取到的 byte array 資料會轉換成 ART 制訂的 message 並且被放進來等待 Message handler 的處理。
4. **Output Queue**：相對於 input queue 從這個 queue 中讀取的 message，會被轉換成 byte array 然後送出去。
5. **Message Handler**：負責讀取 input queue 中的 message，然後呼叫 Windows Manager 或是 Command Manager 執行對應的功能。
6. **Asynchronous Message Delivery**：採用非同步的訊息傳遞機制來控管 input queue 和 output message queue。

因此一個 ART client 啟動後總共會有 3 個 threads，一個負責接收網路上的封包然後將之轉換成 ART message 後放到 input queue 中，另一個負責將 output queue 中的 ART message 轉換成封包傳送到網路上，最後一個負責處理 input queue 中的 ART message。

ART server(請參照圖 4-2 上半部)

1. **Windows Manager**：每一個應用程式的實體(instance)都會有自己的 Windows Manager，負責該應用程式的所有 UI 元件。
2. **Application Manager**：負責載入，連接，停止應用程式。
3. **System Queue**：與 ART client 上的 input queue 功能類似，不同的是，每一個 client 的連線都會有一個 thread 來負責接收該 client 傳送過來的資料。
4. **System Message Handler**：處理 System Queue 中的訊息，主要分成兩類，一類為處理系統訊息，另一類為分發(dispatch)訊息給對應的應用程式。
5. **Application Input Queue**：提供給 system queue handler 把屬於該應用程式的訊息各自放置於其中，然後等待 application input queue handler 的處理。
6. **Application Output Queue**：所有要傳送到網路上的訊息會擺放到各應用程式的這個 queue 裡，等待傳送；且各有一個 thread 在負責處理。
7. **Application Message Handler**：負責處理應用程式的訊息。
8. **Application**：每一個在 ART framework 上發展的應用程式都會繼承 Application 這個 class，它負責初始化應用程式的 input queue，output queue 和 queue handler，並分派每一個應用程式一個 thread 來負責 queue handler 的執行。

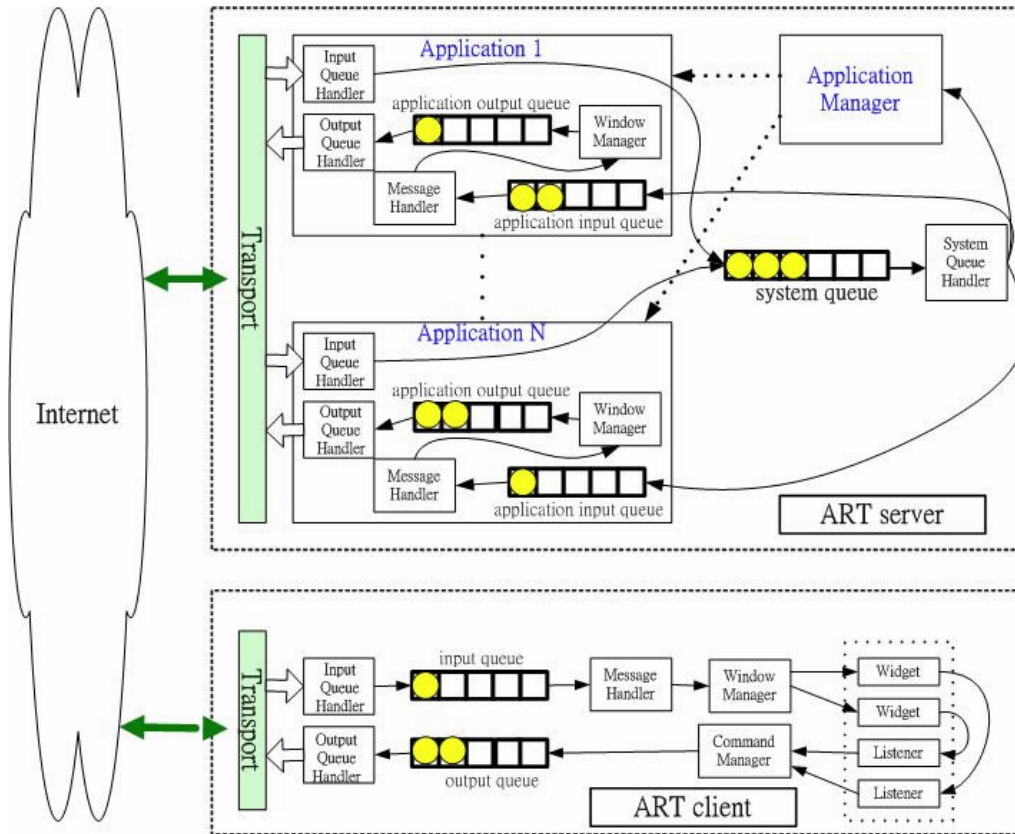


圖 4-2 系統架構圖與訊息流程

第五章、結果與討論

ART 平台發展主要有四大應用成就，分別說明如下。

一、遠端控制程式

概念類似目前 Windows Terminal Service：使用者可以透過手持裝置上顯示的 User Interface 來與遠端正在執行的程式溝通，進而達到遠端控制該程式的目的。其架構如圖 5-1 所示：

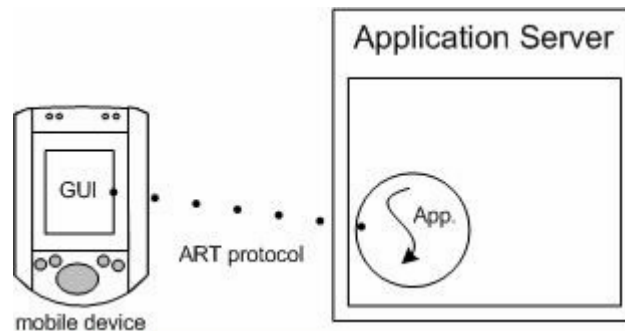


圖 5-1：遠端控制程式概念架構圖

以下我們以大家常用的 ICQ 作為範例。當我們在手持裝置上的 ART Client 選用 ICQ 時，便可進入以下傳訊畫面：



圖 5-2：ICQ 送訊方

輸入 testing message 並 send 後，destination 將會收到剛剛傳出的訊息。

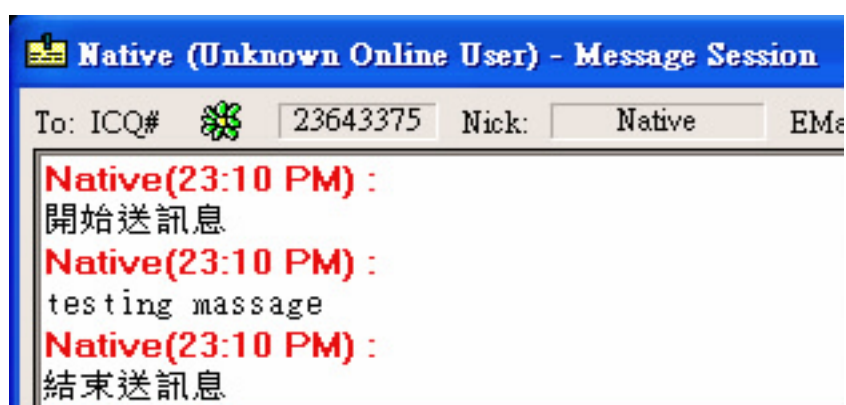


圖 5-3：ICQ 受訊方

二、遠端控制裝置

本項應用是指我們可以使用身邊的手持裝置，透過 ART 系統來控制各式各樣的裝置。舉例來說，對於一些家電用品（例如冷氣），我們可以在 server 端跑一些可以控制家電的 JINI 或 UPNP 的程式，並且提供 UI 給手持裝置。如此一來，使用者便可以透過手邊手持裝置所呈現的 User Interface，下命令給那些 JINI 或 UPNP 的程式，藉以達成隨時隨地控制掌握家電狀況的目的。上述的概念大概可以由圖 5-7 來表達：

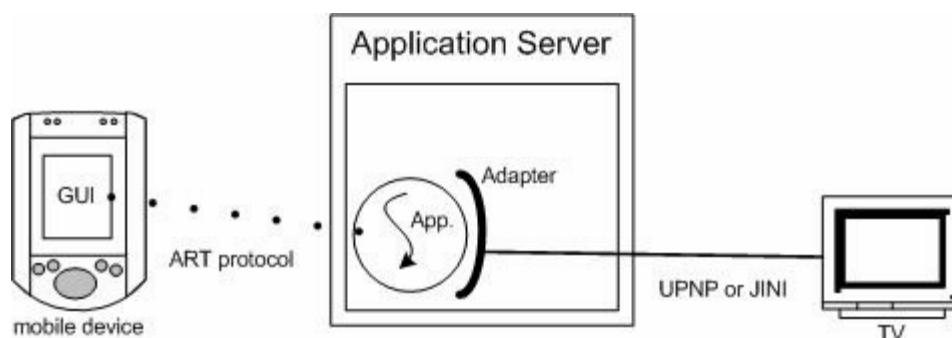


圖 5-4：遠端控制裝置概念架構圖

當然除了家電，只要是能寫程式控制的物品，都能透過本套系統來遠端操作。比如說我們在用 Notebook 透過單槍投影機向客戶在做 present 的時候，常常得在 Notebook 旁操作，導致講解不夠生動。今天我們可以使用這套系統，透過身邊的手機，來遠端操控 Notebook。圖 5-5 顯示了遠端操控中的實際情形：

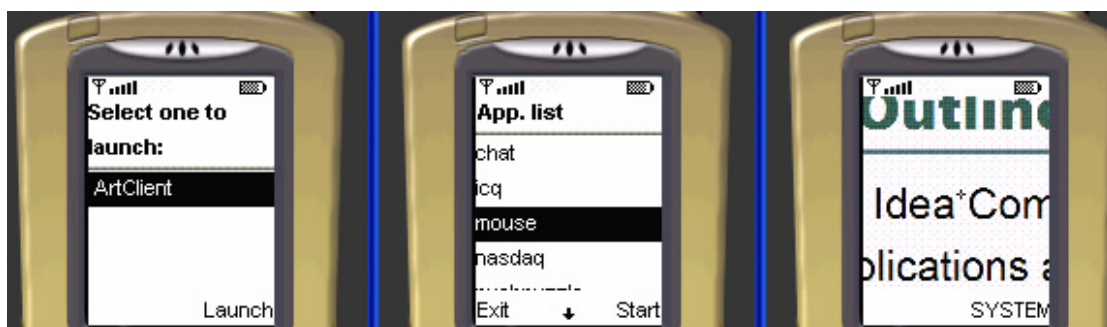


圖 5-5：遠端控制播放投影片

於是我們就可以依照手機上預設的上下左右按鈕來控制 mouse 的 cursor，達到控制投影片的播放的目的。

三、訊息交換

現今手持裝置種類不勝枚舉，各種異質平台（如手機、PDA、工作站、資料庫……等）也難以相容。若是對於每個不同的平台，都有一個相對應的 Application 跑在 ART server 上，那麼我們便可以透過這些相對應的 Applications，來達到異質平台溝通及訊息交換的功能。概念示意如圖 5-6：

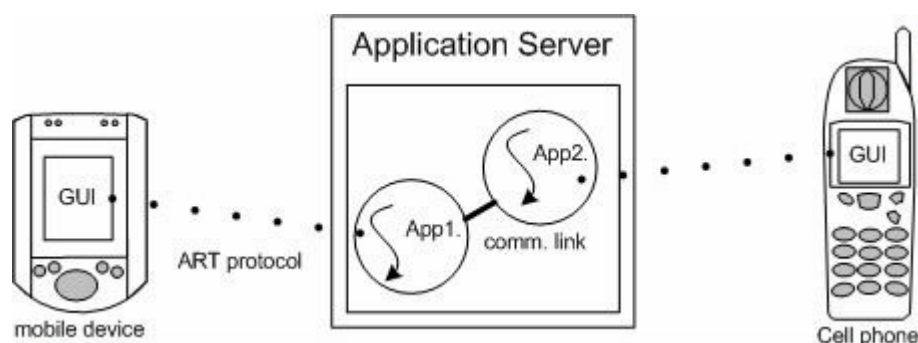


圖 5-6：訊息交換之概念架構圖

假設今天我們有兩位使用者 A 與 B，分別使用不同的手持裝置。兩個手持裝置皆有分別對應的 Application 跑在 Server 上，A 與 B 便可以透過本平台以各自的手持裝置交換訊息（例如聊天室功能）。

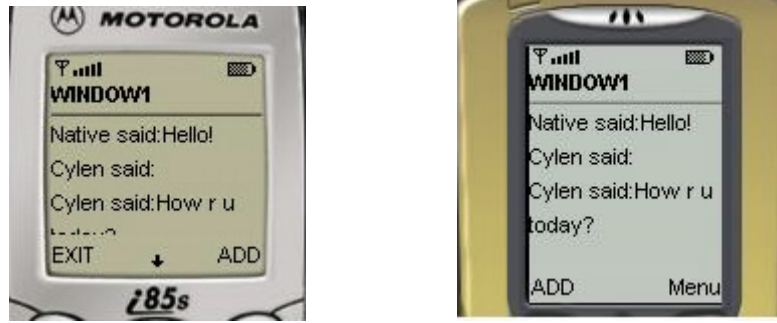


圖 5-7：手持裝置聊天室

四、協同合作

只要該 Application 允許的話，透過 ART Framework，我們也將可以做出如同如同一些像電子白板的群組軟體。也就是說，多個使用者可以同時連上一個使用一個 Application，使用者 A 與使用者 B 可以看到一模一樣的畫面。概念如圖 5-17 所示：

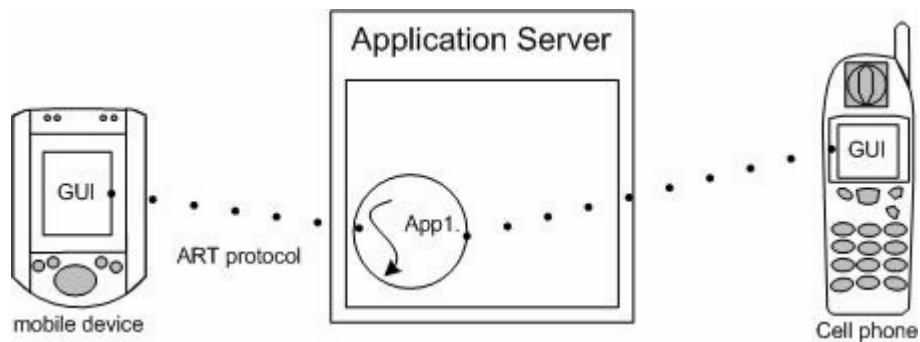


圖 5-8：協同合作概念架構圖

假設今天我們玩倉庫番的遊戲，使用者 A 解不出某個關卡，便可以向使用者 B 求助，使用者 B 就可以連到同一個遊戲的 instance，當場解給使用者 A 看。如圖 5-9 所示：

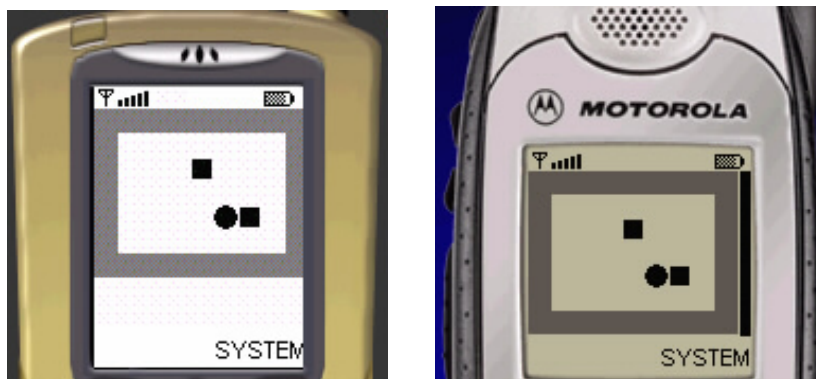


圖 5-9

平台優點討論

根據以上的介紹，我們歸納出本平台以下的優勢：

- **應用程式開發簡單快速**
ART Application/Service Provider 只需將心力專心放在製作應用程式，不需對 ART 系統內部如何運作費心。
- **程式邏輯與 U I 分離**
在 ART 中，開發者只要使用標準的 XUL 描述其 UI 部分，ART server 就會自動的在不同的機器上使用最適合的方式呈現。
- **打破硬體限制**
只要 server 端電腦運算能力夠強，手持裝置只要具備良好的網路連線品質，就能馬上擁有伺服器般的強大運算能力。
- **優異的多工特性**
由於 ART 具有 Windows Manager 與 Application Manager 的機制，在 server 端可以跑各種許多的應用程式，透過 Application Manager 的管理，使用者可在手持裝置上啟用多個各種不同的應用程式。
- **能與現有的網路技術及異質環境計算結合**
透過 ART server，ART client 可以存取不同的其他 Application Server 所提供的網路服務，例如現在正熱門的 Web Service；當然，ART server 上面的 Application 也可以是 JINI 的 Client，經由類似 JINI 的服務，ART client 更能操控更多無以數計的智慧家電和計算裝置。
- **Window Session 的概念**
在我們的設計中，使用者執行的工作狀態皆會紀錄起來，使用者下次連線開啟應用程式便可繼續上次的作業。

第六章、計畫成果自評

目前計畫所做的內容與原本提出的計畫有些許的出入，不過方向還是在研製手持裝置上的軟體開發，在原本的計畫中，第二年是作 context-aware 的研究，這部分我們提前在第一年的時候作，而今年我們是實作一套手持裝置上的軟體開發平台，目前這個平台可以適用在支援 J2ME MIDP 或 WAP 的手持裝置上，開發者只需開發一次就可以在不同的手持裝置上執行。本計畫的相關成果已經投到 2003 數位生活與網際網路科技研討會”。我們預計在第三年時，把 context-aware 併入我們的開發平台中，以期得到更好的效果。

