

行政院國家科學委員會補助專題研究計畫成果報告

eHome：電子家庭雛型之設計與實作--子計畫四：
資訊家電網路之微伺服器及
智慧型代理人軟體之設計與實作

計畫類別：☐個別型計畫 ☒整合型計畫

計畫編號：NSC 89-2218-E-002-076

NSC 90-2218-E-002-030

✓NSC 91-2218-E-002-008

執行期間：89 年 8 月 1 日至 92 年 7 月 31 日

計畫主持人：鄭士康

共同主持人：

本成果報告包括以下應繳交之附件：

- ☐赴國外出差或研習心得報告一份
- ☐赴大陸地區出差或研習心得報告一份
- ☐出席國際學術會議心得報告及發表之論文各一份
- ☐國際合作研究計畫國外研究報告書一份

執行單位：國立台灣大學電機工程研究所

中 華 民 國 九 十 二 年 九 月

摘要

結合資訊、通訊、消費電子的 3C (computer, communication, consumer electronics)產品愈來愈受重視。國立台灣大學電機工程研究所及資訊研究所的四位教授因此組成 3C 研究團隊，配合國科會之 3C 整合科技研究規劃，整合 OS (operating system)、Java、WAP (wireless access protocol) 技術，提出電子家庭 eHome 雛型設計製作之群體研究計劃，包含嵌入式即時作業系統核心程式之設計與實作、嵌入式 Java 及 Jini 系統之設計與實作、無線通訊協定(WAP)系統之設計與實作、資訊家電網路之微伺服器及智慧型代理人軟體之設計與實作等四項子計畫。

本研究計劃為此一群體計畫之第四子計畫，負責 eHome Center 之伺服器及智慧型代理人軟體之設計與實作。第一年利用 PC 環境發展完成接受 HTTP，送出 WML 網頁之伺服器，並藉 RS-232 有線介面控制電燈與冷氣機，驗證 eHome 以無線電手機 WML 瀏覽器控制家電之觀念。第二年以無線遙控電燈及冷氣機，加強伺服器功能，使其能接受 WAP，並開發完成具有學習使用者操作習慣能力之簡單智慧型代理人軟體，以便代替使用者進行日常家電操作。第三年延伸第二年完成之具有學習能力之智慧代理人軟體，搭配其他子計畫在視聽家電 Havi 技術之發展，開發完成能依使用者收視偏好推薦電視節目之智慧型代理人程式系統。本研究開發之技術與經驗，在智慧型家電及互動電視即將進入一般家庭的今日，具有相當大的實用價值。使用者偏好的學習及推薦技術也可應用到其他場合，如電影、音樂、書籍、電腦遊戲的推薦與過濾。

Abstract

The 3C products, especially the intelligent appliances, which combine computer, communication, and consumer electronics, have become a new and significant trend. Our team with members from Graduate Institute of Electrical Engineering and Graduate Institute of Information Engineering and Computer Science, National Taiwan University, thus proposed to integrate operating system, Java, WAP (wireless access protocol) into an eHome Center prototype system. This integrated project includes the design and implementation of an embedded real-time operating system kernel, design and implementation of an embedded Java system, design and implementation of a WAP system, design and implementation of a server and intelligent agent systems for information appliances network.

This research is the fourth subproject of the integrated project. The goal is to design and implement a server and intelligent agent systems for the eHome Center. In the first year, we have developed through the PC environment a server which can accept HTTP requests and send out WML web page. It can also control lamps and air-conditioners by RS-232 wired interface. This showed that controlling home appliances through a WML browser on a wireless mobile phone is possible. In the second year, we have replaced the wired RS-232 interface by a wireless link, and developed simple intelligent agents which can learn the user preferences to control lamps and air-conditioners. In the third year, we extend the intelligent agent system such that it can learn the user preference of watching TV programs and recommend new programs to the user. Our experience and the techniques developed in this research are believed to be valuable in the future, with the notion that the intelligent appliances and interactive TV are going to enter the living rooms of peoples' houses. The preference learning and recommendation techniques are also useful in other applications, like recommending and filtering films, music, books, or games to users.

目錄

摘要	
Abstract	
圖目錄	
表目錄	
第一章 導論	1
1-1 引言	1
1-2 文獻回顧	2
1-3 章節概要	2
第二章 資訊家電網路微伺服器之設計與製作	4
2-1 系統架構	4
2-2 WAP/WML 簡介	5
2-3 操作實例	6
2-4 系統 WML 卡疊設計	9
2-5 系統主要演算法設計	12
第三章 控制一般家電之智慧型代理程式設計與製作	15
3-1 系統架構	15
3-2 決策樹機器學習演算法	18
3-3 連續變數之處理	26
3-4 操作實例	28
3-5 系統主要結構設計	33
第四章 依使用者偏好推薦電視節目之智慧型代理程式設計與製作	37
4-1 系統架構	37
4-2 電視節目屬性	37
4-3 提供節目資訊之網站程式	40
4-4 使用者收看時間紀錄與分析	45
4-5 使用者偏好之學習	48
4-6 電視節目之推薦	50
4-7 操作實例	52
4-8 系統主要結構設計	53
第五章 結論	56
參考文獻	57

圖目錄

圖 1.1. eHome 結構雛型	1
圖 2.1. eHomeServer 程式軟體結構	4
圖 2.2. 無線上網結構	5
圖 2.3. SimpleLamp 及電路與天線	7
圖 2.4. RS-232 介面電路及天線	7
圖 2.5. 手機畫面	8
圖 3.1. eHome Center 程式軟體架構	15
圖 3.2. Personalization Agent 程式軟體架構	16
圖 3.3. Decision Tree Learning 演算法	16
圖 3.4. 程式 JDecisionTree 輸出一例	17
圖 3.5. 決策樹一例	19
圖 3.6. 部分完成之決策樹	20
圖 3.7. 美國國會議員投票立場與其黨籍之決策樹	21
圖 3.8. 教育支出法案分支子樹	22
圖 3.9. 修剪圖 3.8 子樹後所形成之單一分支	22
圖 3.10. 第一次修剪圖 3.7 後之決策樹	22
圖 3.11. 完全修剪圖 3.7 後之決策樹	23
圖 3.12. p 的估計範圍與信心水準	24

圖 3.13. 決策樹類別程式結構	26
圖 3.14. 範例之類別程式結構	26
圖 3.15. 由 n 個物品選取 m 個的組合產生程式	27
圖 3.16. 簡單燈具使用紀錄之一例	29
圖 3.17. 變光燈使用紀錄之一例	30
圖 3.18. 冷氣溫度設定紀錄一例	30
圖 3.19. 程式 eHomePlace1 之起始畫面	31
圖 3.20. 產生新代理人需填寫之表單	32
圖 3.21. 啟動先前存檔之代理人程式畫面一例	33
圖 3.22. 啟動 Lamp1Agent 後之主視窗畫面	33
圖 3.23. 介面 EHomeAgent 程式	34
圖 4.1. 電視節目推薦系統架構	37
圖 4.2. 電視節目表檔一例	41
圖 4.3. 電視節目資訊伺服器端主程式	42
圖 4.4. HTTPServer 類別程式碼	43
圖 4.5. 電視節目資訊客戶端主程式	45
圖 4.6. 收看時間紀錄檔一例	46
圖 4.7. 電視節目評分檔一例	47
圖 4.8. 電視節目收看時間紀錄及處理示意圖	48

圖 4.9. 由圖 4.7 之電視節目評分檔產生之戲劇範例檔	49
圖 4.10. 戲劇屬性資料檔	50
圖 4.11. 由圖 4.2 電視節目表檔產生的戲劇節目待評分檔	50
圖 4.12. 對圖 4.11 之檔案的評分結果	51
圖 4.13. 產生 TVAgent 後之主視窗畫面	52
圖 4.14. 推薦節目列表視窗	53

表目錄

表 3.1. 打網球的紀錄	18
表 3.2. EHomePlaceModel 主要方法成員	34
表 3.3. 各代理人程式的特殊方法成員	35
表 4.1. 電視節目分類及屬性	39
表 4.2. 電視節目推薦代理人程式 TVAgent 的特殊方法成員	54
表 4.3. 類別 RatingSetter 的主要方法成員	55

第一章 導論

1-1 引言

近來由於超大型積體電路(VLSI)及網際網路(internet)的快速發展普及，結合資訊、通訊、消費電子的 3C (computer, communication, consumer electronics)產品也愈來愈受重視。國立台灣大學電機工程研究所及資訊研究所的四位教授因此組成 3C 研究團隊，配合國科會之 3C 整合科技研究規劃，整合 OS (operating system)、Java、WAP (wireless access protocol) 技術，提出電子家庭 eHome 雛型設計製作之群體研究計劃，包含嵌入式即時作業系統核心程式之設計與實作、嵌入式 Java 及 Jini 系統之設計與實作、無線通訊協定(WAP)系統之設計與實作、資訊家電網路之微伺服器及智慧型代理人軟體之設計與實作等四項子計畫，目標為建立如圖 1.1 結構的雛型系統。

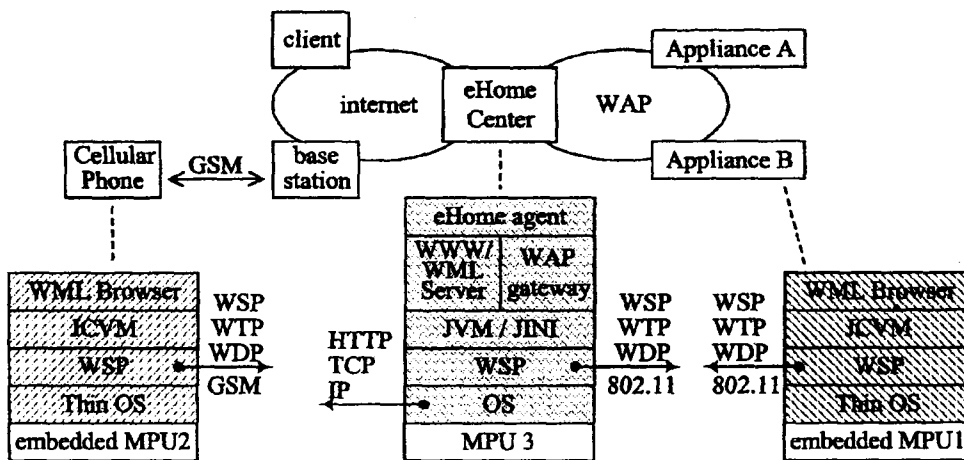


圖 1.1. eHome 結構雛型

此一系統應用實例可敘述如下：我們在離家十萬八千里遠的地方，用一個 WAP enabling 的大哥大手機，利用上頭的 WML (wireless markup language) 瀏覽器，透過 GSM 系統連結上大哥大基地台，再透過無遠弗屆的 internet 連結回家裡的 eHome Center。eHome Center 可能是一個 set-top box，對外有個 WWW 及 WML 伺服器，經由 WML 網頁可以讓我們得知家中的家電動態、溫濕度、保全系統...等等資訊；透過 WML 網頁我們也可以下達操控指令，如開啓冷氣調節室溫，或開關電燈音響以防盜賊。eHome Center 對內則用無線的方式與 Java enabling 的家電聯繫，通知家電執行指令。eHome Center 並內含智慧型代理人 (intelligent agent) 軟體，能夠學習使用者的使用習慣，自動操作家電，提供家事建議，或透過網路呼叫使用者手機，提醒使用者下班前採購某項食品等等。如此一來，這些家電不僅是資訊家電 (information appliances)，是網路家電 (internet appliances)，也是智慧家電 (intelligent appliances)。

本研究計劃為此一群體計畫之第四子計畫，負責 eHome Center 之伺服器及智慧型代理人軟體之設計與實作。第一年利用 PC 環境發展完成接受 HTTP，送出

WML 網頁之伺服器，並藉 RS-232 有線介面控制電燈與冷氣機，驗證 eHome 以無線電手機 WML 瀏覽器控制家電之觀念。第二年以無線遙控電燈及冷氣機，加強伺服器功能，使其能接受 WAP，並開發完成具有學習使用者操作習慣能力之簡單智慧型代理人軟體，以便代替使用者進行日常家電操作。第三年配合總計畫之修訂，將原訂發展之廚房資訊系統予以變更：延伸第二年完成之具有學習能力之智慧代理人軟體，搭配其他子計畫在視聽家電 Havi 技術之發展，研究開發能依使用者收視偏好推薦電視節目之智慧型視聽家電系統。

1-2 文獻回顧

本研究 eHome Center 內之伺服器需能應使用者要求，發送 WML 網頁。WML 為 WAP 科技一部份，最新及最完全之資料可至網頁[1]取得。本研究所屬總計畫主持人許清琦教授曾試製 WAP 系統，並有一簡單雛型於網頁[2]展示。

另一 eHome Center 內的重要軟體為智慧型代理人。此處所說的智慧型代理人(intelligent agents)與 WAP 網路的使用者代理人(user agents)涵義略有不同。使用者代理人通常泛指使用介面，如文字或多媒體之網路瀏覽器，可傳遞使用者意圖；而智慧型代理人則常為具有若干人工智慧及移動性能(mobility)之軟體，可主動(autonomous)代使用者處理資訊，並可能與其他智慧型代理人會商學習。智慧型代理人研究近年來漸趨熱絡，知名大學如 MIT [3]、Carnegie Melon [4]、University of Maryland of Baltimore County [5]、Stanford University [6] 等等，大公司如 IBM [7]、HP [8] 等等及新創高科技公司如 Cycorp [9]、General Magic[10] 等等，均投入大量人力資金研究。一九九四年七月 Communications of ACM 的 Special Issue on Intelligent Agents 收錄有數篇重要論文，其中由 MIT Media Lab 教授 P. Maes 發表之 Agents that reduce work information overload 一文 [11]，已公認為經典之作。該文敘述應用機器學習(machine learning)可發展出依使用者偏好，代使用者分類電子郵件、在龐大網路資訊中搜尋有興趣之論壇文章、建議購買書籍唱片等之智慧型代理人。該文的觀念已成功用於本計畫，發展資訊家電應用的智慧型代理軟體。台大電機研究所雷欽隆教授、王勝德教授及本人與資訊研究所許永真教授均在近年指導過智慧代理人方面之碩士論文，國內其他大學也有數位教授進行類似研究。國科會曾舉辦智慧型代理人研討會，有甚多國內學者參加。然而，利用智慧型代理人技術控制資訊家電的研究，除本研究外，似乎還沒有在國內外看到。

1999 年 IEEE Spectrum 封面即是 The Networked House，在其相關論文[12]內提及 Home Networking 的潛在市場龐大，故大小廠商均爭相投入。在網路上，我們也查到了一些相關的公司網站 [13]-[16]，可供參考。

1-3 章節概要

本報告依照執行順序，分年介紹研究成果。第二章包含第一年之 WML 微伺服器架構，程式大略說明，及操作實例。第三章說明第二年之家電控制智慧型代理人程式系統架構，決策樹機器學習演算法，程式碼解說，及操作實例。第四章

爲第三年成果，敘述電視節目收看偏好之量度與學習，電視節目之推薦方式，程式碼解說，與操作實例。第五章作一總結。

第二章資訊家電網路微伺服器之設計與製作

2-1 系統架構

整個 eHomeServer 的軟體架構以 UML 之類別圖(class diagram)表示如圖 2.1：Server 類別含有 ServletRequest、Servlet、ServletResponse 三個類別的 Field Data Member。其中 ServletRequest 包裝由手機發出，由閘口站(gateway) 轉來之 HTTP 協定形式之需求；Servlet 依需求發送訊息至各空間單位(SpaceUnit 類別，包括房間、入口、陽台等)的家電代表類別 Appliance (可以是房門開關狀態感應器 Door、簡易燈具 SimpleLamp、空調 AirConditioner、音響 Audio、視訊設備 Vedio 等)，家電進行指定之動作後，將結果以 WML 網頁描述，配合 Response 之設定送出。其中的 Appliance 類別是抽象類別，各種不同家電可以依不同特性，改寫 Appliance 所規定的方法成員(method members)。另外，Servlet 類別是我們仿照 javax.servlet 架構[2]自行撰寫，並沒有實際用到 javax.servlet 內的程式。另外，總計畫的助理廖文良先生以 Visual C++ 實作 RS-232 介面程式，並發展 Java Native Interface (JNI) 介面類別，稱為 RS232，使各種家電的代表類別得以利用 RS-232，與實際的家電通訊。

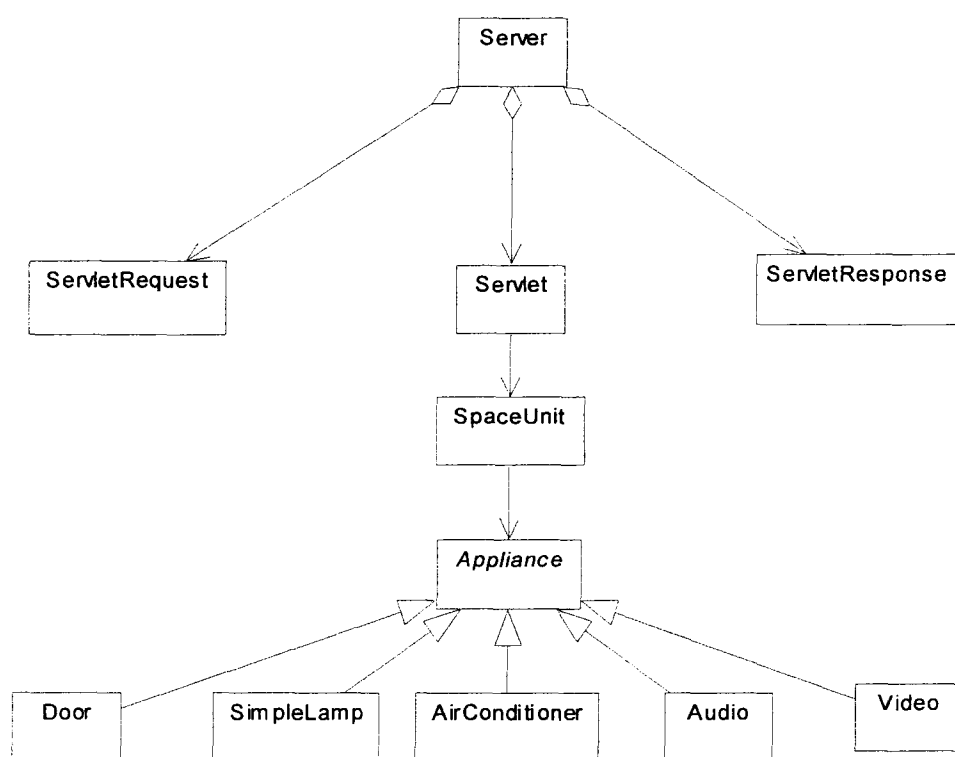


圖 2.1. eHomeServer 程式軟體結構

以上主要類別提供的功能，略述如下：

EHomeServer：主程式所在，傾聽(listen) 通訊插座(socket)內的一個通訊埠 (port)，每聽到一次使用者的連結要求即建立一個 Server。

ServletRequest：剖析 HTTP 訊息。

Server：建立執行緒，呼叫 Servlet 以完成 ServletRequest 需求的動作。

ServletResponse：傳送訊息標頭部分(header)及錯誤訊息等。

Servlet：建立維護 SpaceUnit 資料，處理 Request，發送 WML 主卡疊。

SpaceUnit：建立維護 SpaceUnit 內之 Appliance 資料，發送屬於 SpaceUnit 的 WML 卡疊。

Appliance：建立維護家電之狀態資料。具有三個抽象方法成員，分別為取得家電狀態(getStatus)、設定家電狀態(setStatus)、及發送對應家電的 WML 卡疊。衍生自本類別之具體家電類別如 SimpleLamp 必須覆寫(override) 這三個方法成員。

RS232：開啓及關閉 RS232 埠，藉由家電網路通訊協定(protocol)與家電聯繫。此一通訊協定目前因處理之環境較為簡單，採用自訂簡單格式。未來視相關科技發展，可採用較複雜之產業標準如 Bluetooth 之格式。

2-2 WAP/WML 簡介

手機連到我們伺服器(server)的方式如圖 2.2 [17]所示：手機先連上由電信業者或網際網路服務供應商(Internet Service Provider, ISP)提供之閘口站(gateway)，再由閘口站連到伺服器，然後伺服器提供網頁訊息，反向由閘口站傳至手機，顯示於手機的瀏覽器(browser)上。由於手機訊號的頻寬有限，因此連接閘口站時，不能使用一般網際網路的 HTTP 通訊協定，而必須採用專為無線傳輸開發之無線應用協定(Wireless Application Protocol, WAP)，而閘口站則須將 WAP 轉為 HTTP，經由網際網路連上伺服器，所以我們的伺服器只需處理 HTTP，而不必直接處理 WAP。此外，因手機的 CPU 能力、記憶體容量、及螢幕大小限制較大，手機瀏覽器無法瀏覽一般使用 HTML 撰寫之網頁，因此我們伺服器提供之網頁訊息須以功能較 HTML 簡化之 WML(Wireless Markup Language)呈現，無線上網裝置之瀏覽器才能觀看訊息內容。

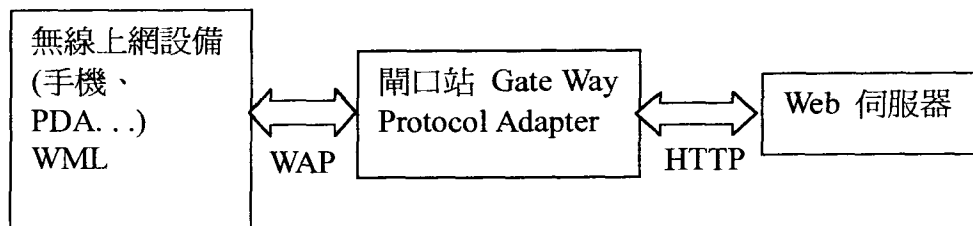


圖 2.2. 無線上網架構

WML 延伸自 XML(eXtensible Markup Language)，用來描述文字或圖形顯現於手機螢幕之方式。由於手機螢幕較小，一次能顯示的內容不多，因此其顯示的單位不再是 HTML 網頁(page)，而是卡片(card)，多張卡片組成的單位，則稱為卡疊(card deck)。

一個 WML 檔案由多張卡片組成之卡疊構成，其文件結構分檔頭及文件本體兩部分 [17]。檔頭固定為如下兩行(後兩行應併為一行)：

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM/DTD WML 1.1/EN
    http://www.wapforum.org/DTD/wml 1.1.xml>
```

以上由符號<和>以及二者所夾的字串稱為標籤(tag)。第一行由<?和?>構成的標籤為指令列，其中的 xml version="1.0"表示檔案為使用 XML1.0 標準之文件。第二行為外部文件格式定義，表示引用 <http://www.wapforum.org/DTD/wml 1.1.xml> 內對 WML 語法的規定。文件本體內的基本標籤格式有<tag> . . . </tag>(雙標籤)及<tag/>單標籤兩種。典型之文件本體及附帶之說明如下：

```
<wml>
  WML 文件開始
  <card id = "A"> 卡片 A 開始
    <p> 段落開始
      <do type = "accept" label = "next">
        按手機左鍵 options 出現"next"選項
        <go href="#B"/> 選 "next" 選項到卡片 B
      </do>
      This is Card A 顯示於手機螢幕
    </p> 段落結束
  </card> 卡片 A 結束
  <card id = "B"> 卡片 B 開始
    <p> 段落開始
      This is Card B 顯示於手機螢幕
    </p> 段落結束
  </card> 卡片 B 結束
</wml>
```

其餘用到的標籤在系統卡疊設計時說明。在撰寫系統卡疊時，我們當時由 Nokia 公司的 Nokia WAP 發展論壇下載 Nokia ToolKit 1.3 軟體(現在已升級到 Nokia Internet Mobile Tool Kit 4.0 [18])來模擬手機的瀏覽器，先模擬無誤後，再寫入伺服器程式之中。

2-3 操作實例

以下以實例介紹系統之操作過程：假設所要控制的環境有兩個 SpaceUnit，即 Entrance (入口)及 LivingRoom (起居室)，內有家電數個。圖 2.3 所示為可用的家電雛型之一，僅能開關之燈具 SimpleLamp，附於一簡單無線電路，以天線與接在個人電腦序列通信埠，並配有天線之 RS-232 介面電路(圖 2.4)通訊。這些電路及其他家電雛型如變光燈、小電扇(模擬冷氣機)、遙控器等之電路均由總計畫研究助理廖文良先生一手設計製作。

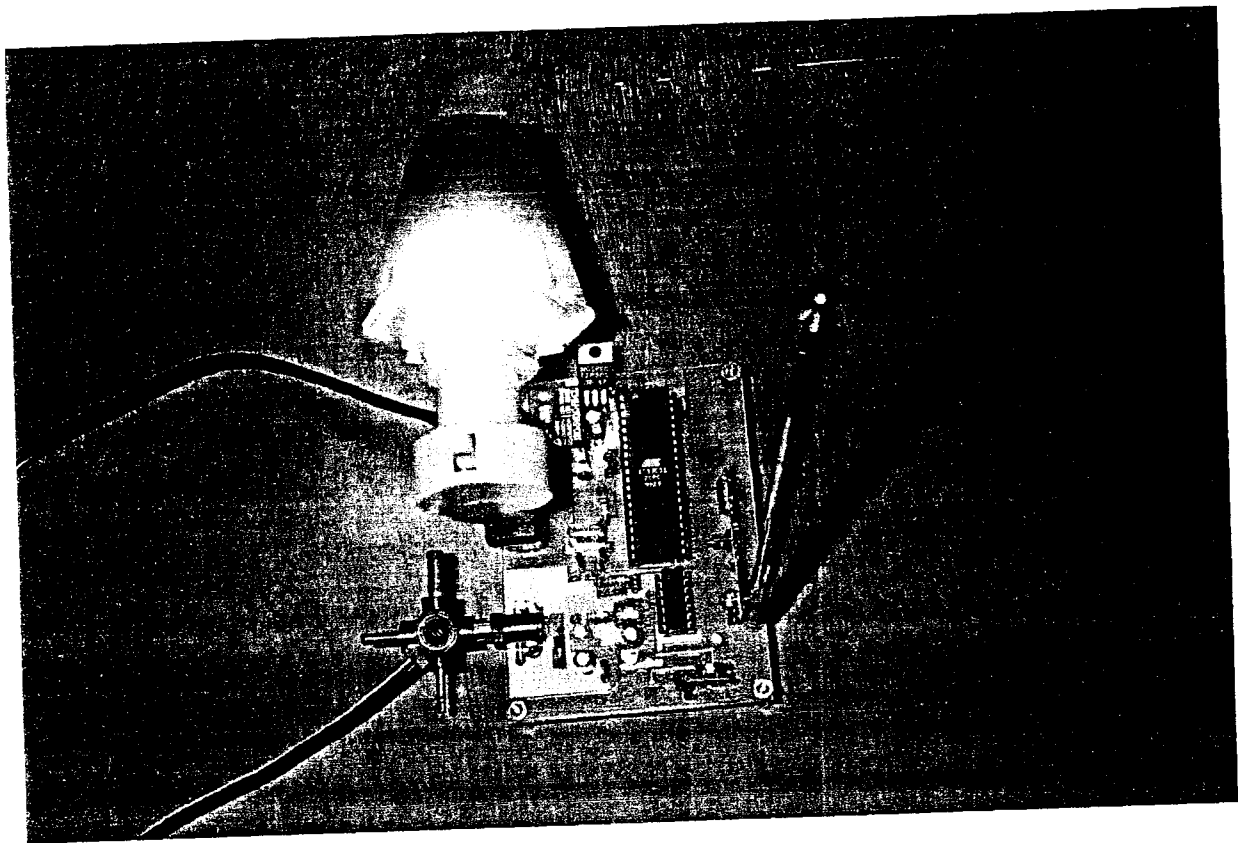


圖 2.3. SimpleLamp 及電路與天線

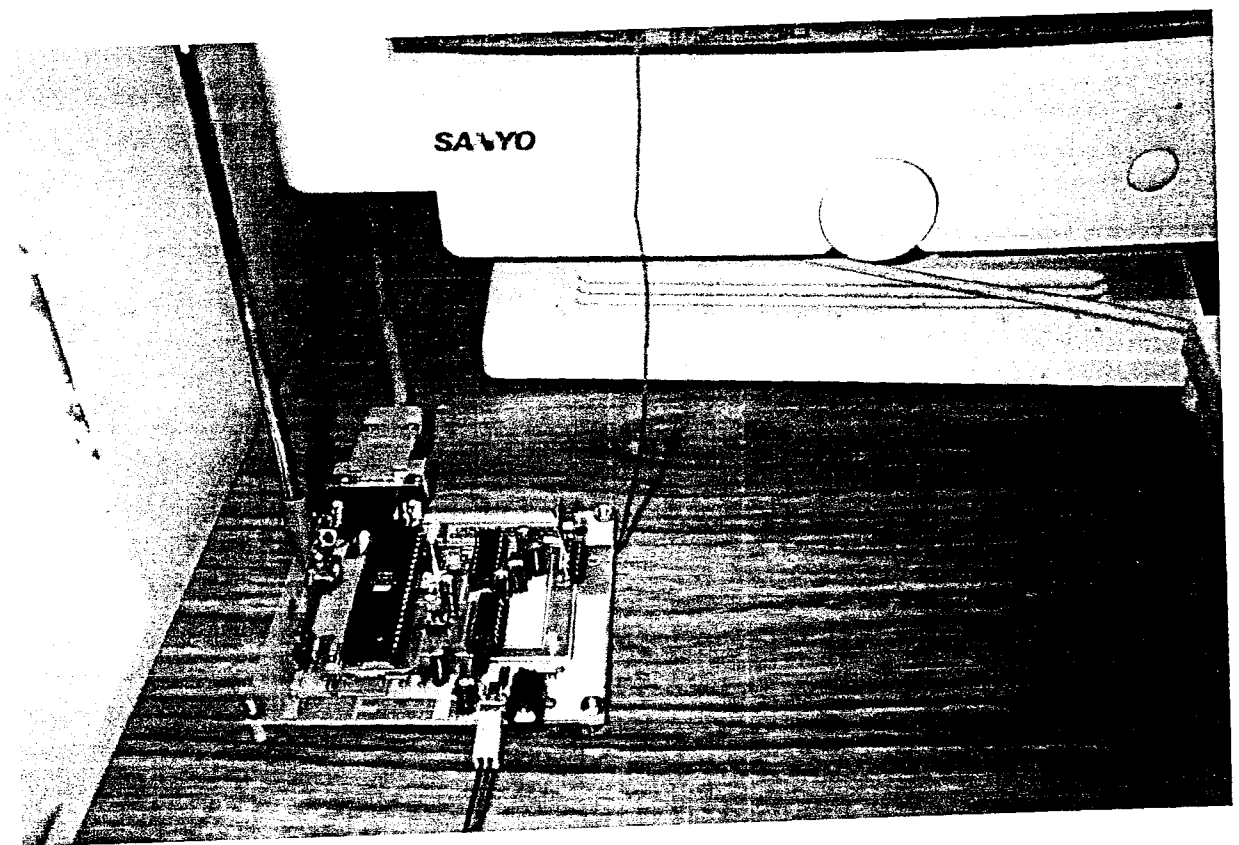


圖 2.4. RS-232 介面電路及天線

開始操作時，首先在接妥通訊網路之個人電腦執行 eHomeServer 程式。此一 eHomeServer 會產生一個通訊插槽(server socket)，進入一無窮迴圈，傾聽編號為 9200 之通訊埠。使用者接著於手機鍵入 eHomeServer 所在個人電腦的 IP Address，建立手機—Gateway—個人電腦連結。eHomeServer 聽到連結完成，產生一個 Server，送出 WML 主卡疊後結束連結，手機之 WML 瀏覽器(WML browser)解讀卡疊訊息，在手機螢幕顯示畫面如圖 2.5(a)。使用者依訊息按下手機之 Option 鍵後，畫面顯示所有 SpaceUnit 選項，如圖 2.5(b)。

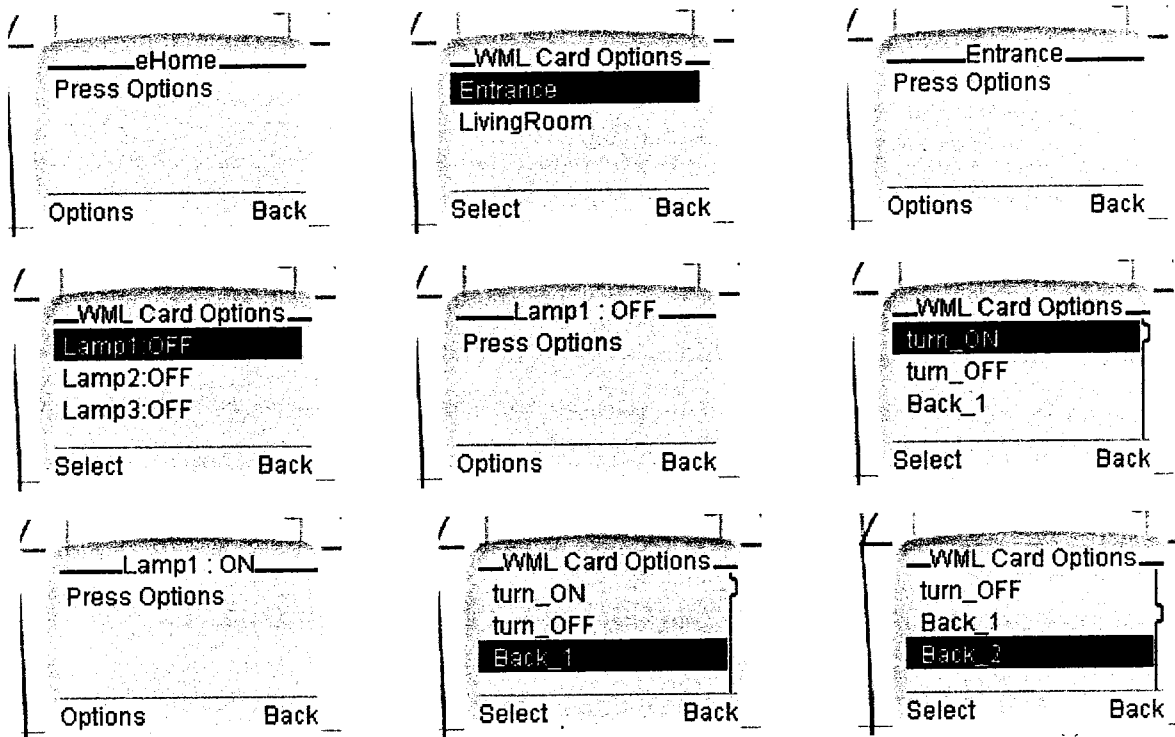


圖 2.5. 手機畫面

使用者其次點選 Entrance，又建立一個連結，產生新 Server，送回 Entrance 內所有家電選項，如圖 2.5(c)(d)，連結又結束。

使用者接續選擇 Lamp1，建立新連結及新 Server。eHomeServer 經由 RS-232 介面及無線電傳輸，取得 Lamp1 狀態，再送出含 Lamp1 狀態之卡疊後關閉連結。因 Lamp1 為一 SimpleLamp，僅有 ON 與 OFF 兩種狀態，故 Lamp1 為 OFF 時手機畫面如圖 2.5(e)。

至此使用者可改變 Lamp1 之狀態，例如，使用者可按 Option 鍵，由圖 2.5(f)點選 ON，建立連結。eHomeServer 於是產生 Server，透過 RS-232 介面及無線電傳輸打開 Lamp1 後，送回狀態更新之 Lamp1 卡疊，隨即中斷連結，使用者畫面如圖 2.5(g)。使用者按 Option 鍵後，可利用手機畫面上之 Back1 及 Back2 選項(圖 2.5(h)(i))，分別回到 SpaceUnit 或主卡疊，進行其他操作。

2-4 系統 WML 卡疊設計

以上描述之系統，使用者連上伺服器後，伺服器傳回文件主體如下之 WML 主卡疊，在手機顯示圖 2.5(a)之畫面，使用者按下 Options 鍵後，手機畫面成為圖 2.5(b)。

```
<wml>
<head>
  避免用到先前動作的記憶
  <meta http-equiv="Cache-Control" content="no-cache"/>
  <meta http-equiv="Expires" content="0"/>
  <meta http-equiv="Pragma" content="no-cache"/>
</head>
<template> 定義本卡疊所有卡片共同均擁有之部分
  設定返回鍵 prev，使用者可藉此返回前一畫面
  <do type="prev" name="Previous" label="Back">
    <prev/>
  </do>
</template>
<card id="eHome" title="eHome"> 標題設定為 eHome
<p>
  Press Options 顯示 Press Options
  接著對所有 SpaceUnit 重複如下之 do 敘述，用到 Java 字串語法
  name 為 SpaceUnit 名稱，localURL 為伺服器網址，port 為其埠號
  do type 為 submit，表示要由手機傳送資料
  label = name 顯示 SpaceUnit 名稱
  <do name= name type="submit" label= name>
    使用者點選 SpaceUnit 名稱之選項後，go 標籤可使瀏覽器將變數
    spaceUnit，appliance，status 等傳送至伺服器
    <go href="http://" + localURL + ":" + port + "/">
      postfield 定義欲傳送之變數及數值
      <postfield name="spaceUnit" value= name/>
      <postfield name="appliance" value="empty"/>
      <postfield name="status" value="empty"/>
    </go>
  </do>
</p>
</card>
</wml>
```

使用者點選 SpaceUnit 名稱，並由瀏覽器送至伺服器後，伺服器傳回如下顯示 SpaceUnit 內各電器名稱之空間單元卡疊(省略 wml 及 head 標籤)，手機螢幕顯示如圖 2.5(c)(d)：

```
<template>
  <do type="prev" name="Previous" label="Back">
    <prev/>
```

```

</do>
</template>
<card id= name title= name>
<p>
    Press Options
    對所有電器 Appliance 重複如下之 do 敘述，用到 Java 字串語法
    name 為 SpaceUnit 名稱，applianceName 為電器名稱
    status 為電器狀態
    <do name= applianceName type="submit" label=
        applianceName + ":" + status>
        <go href="http://" + localURL + ":" + port + "/">
        <postfield name="spaceUnit" value=name/>
        <postfield name="appliance"
            value=applianceName/>
        <postfield name="status" value= status/>
        </go>
    </do>
</p>
</card>

```

使用者點選電器名稱選項之後，伺服器傳回如下顯示電器名稱、狀態、及控制選項之電器卡疊(以 SimpleLamp 為例，省略 wml 及 head 標籤)，手機螢幕顯示如圖 2.5(e)(f)：

```

<template>
    <do type="prev" name="Previous" label="Back">
        <prev/>
    </do>
</template>

```

```

name 為電器名稱
<card id= name title= name+" : "+status>
<p>
    Press Options
    SimpleLamp 之控制選項為 turn_ON(開)，turn_OFF(關)，
    Back_1(回 SpaceUnit 卡疊)，Back_2(回主卡疊)
    name 為電器名稱，SpaceUnitName 為 SpaceUnit 名稱
    <do name= name+"ON" type="submit" label=" turn_ON ">
    <go href="http://" + localURL + ":" + port + "/">
    <postfield name="spaceUnit" value= spaceUnitName/>
    <postfield name="appliance" value= name/>
    <postfield name="status" value="ON"/>
    </go>
    </do>
    <do name= name + "OFF" type="submit" label=" turn_OFF ">
    <go href="http://" + localURL + ":" + port + "/">
    <postfield name="spaceUnit" value= spaceUnitName/>
    <postfield name="appliance" value= name/>
    <postfield name="status" value="OFF"/>

```

```

</go>
</do>
<do name= name+"Back1" type="submit" label=" Back_1 ">
<go href="http://" +localURL+":"+port+"/">
<postfield name="spaceUnit" value= spaceUnitName/>
<postfield name="appliance" value="empty"/>
<postfield name="status" value="empty"/>
</go>
</do>
<do name= name+"Back2" type="submit" label=" Back_2 ">
<go href="http://" +localURL+":"+port+"/">
<postfield name="spaceUnit" value="empty"/>
<postfield name="appliance" value="empty"/>
<postfield name="status" value="empty"/>
</go>
</do>
</p>
</card>

```

使用者選擇控制選項後，伺服器調整電器狀態，再送回上列卡疊，手機畫面如圖 2.5(g)，如使用者欲回到空間單元卡疊(圖 2.5(h))，可按 Back_1，欲回到主卡疊，可按 Back_2(圖 2.5(i))。

冷氣及變光燈之電器卡疊與 SimpleLamp 之電器卡疊相似，但使用時另需使用者輸入所欲設定的溫度及燈光強度，其傳送設定狀態之電器狀態卡疊，分別設計如下(省略 wml 及 head 標籤)，注意 SimpleLamp 之電器狀態卡疊與其電器卡疊相同：

冷氣

```

<template>
  <do type="prev" name="Previous" label="Back">
    <prev/>
  </do>
</template>
newcontext 為 true 時，瀏覽器每次讀到此卡，
其先前內容及變數會先被清除
<card id= name title= name+" : "+status newcontext="true">
  <p>
    <do type="submit" label="sendTemperature" >
      <go href="http://" +localURL+":"+port+"/">
        <postfield name="spaceUnit" value= spaceUnitName/>
        <postfield name="appliance" value= name/>
        傳送所欲設定之溫度(存於變數$(temperature)之中)
        <postfield name="status" value="$(temperature)"/>
      </go>
    </do>
    提醒設定溫度，0 代表關，1 代表開，15 ~ 40 為所欲設定之溫度
    Temperature setting (0: off, 1: on, 15-40: temperature):

```

```

輸入溫度，format = "*N" 表示型態為數字，
maxlength = 3 代表輸入數字字串長度為 3，
title = "temperature" 表示輸入時之提示字串為 temperature
emptyok = "false" 表示輸入資料不可空白
<input name = "temperature" format="*N" maxlength = "3"
    title="temperature" emptyok="false"/>
</p>
</card>

```

變光燈

```

<template>
  <do type="prev" name="Previous" label="Back">
    <prev/>
  </do>
</template>
<card id= name title= name+" : "+status newcontext="true">
  <p>
    <do type="submit" label="sendLightLevel" >
      <go href="http://" +localURL+" : "+port+"/">
        <postfield name="spaceUnit" value= spaceUnitName/>
        <postfield name="appliance" value= name/>
        傳送所欲設定之亮度(存於變數$(lightLevel)之中)
        <postfield name="status" value="$(lightLevel)"/>
      </go>
    </do>
    提醒設定亮度，0 (代表關)至 100(最亮)
    Light Level (0-100) :
    輸入亮度，format = "*N" 表示型態為數字，maxlength = 3 代表輸
入數字
字串長度為 3，title = "lightLevel" 表示輸入時之提示字串為
lightLevel
emptyok = "false" 表示輸入資料不可空白
    <input name = "lightLevel" format="*N" maxlength = "3"
        title="lightLevel" emptyok="false"/>
  </p>
</card>

```

2-5 系統主要演算法設計

本系統基本上為一個使用 HTTP 通訊協定之伺服器，故其主程式骨幹[19]如下：

```

static final int PORT = 9200; //default port
static String localURL;

```

```

public static void main(String args[]) {
    Servlet servlet;
    ServerSocket serverConnect;

    InetAddress address = InetAddress.getLocalHost();
    localURL = address.getHostAddress();
    servlet = new Servlet("TestConfigurationA.txt", localURL,
        PORT); // 建立 Servlet
    serverConnect = new ServerSocket(PORT); // 建立伺服器 socket
    System.out.println("\nListening for connections on port "+
        PORT + "... \n");
    while( true ) { // 聆聽等待客戶連線
        new Server( serverConnect.accept(), servlet )
            // 連上線時建立一個 Server
            System.out.println("Server constructed");
    }
}

```

上列程式中，建立 Servlet 的敘述呼叫 Servlet 的建構元 (Constructor)，讀入檔案 TestConfigurationA.txt，取得各空間單元及電器之資料。檔案 TestConfigurationA.txt 之一例如下：

2 兩個空間單元

Entrance	5	空間單元 Entrance 有五個電器
LightSensor1	5 3	電器 LightSensor1 為第 5 類電器，代號 3
Lamp1	0 4	電器 Lamp1 為第 0 類電器，代號 4
Lamp2	0 115	電器 Lamp2 為第 0 類電器，代號 115
Dimmer1	1 3	電器 Dimmer1 為第 1 類電器，代號 3
Safety	2 8	電器 Safety 為第 2 類電器，代號 8
LivingRoom	4	空間單元 LivingRoom 有四個電器
Lamp3	0 55	電器 Lamp3 為第 0 類電器，代號 55
Dimmer2	1 120	電器 Dimmer2 為第 1 類電器，代號 120
Airc1	3 5	電器 Airc1 為第 3 類電器，代號 5
TVCon	4 9	電器 TVCon 為第 4 類電器，代號 9

程式將電器分為六類，即第 0 類：簡易燈具 (SimpleLamp)，第 1 類：變光燈，第 2 類：保全感應器，第 3 類：冷氣，第 4 類：遙控裝置，第 5 類：光度感應器。

前述程式中，每個客戶 (client) 連線所建立的 Server 擁有一個獨立的執行緒，負責與客戶之間的通訊。執行緒執行時，取得客戶的 request，要求 Servlet 作出回應 (response)，並將回應傳回客戶。Servlet 能夠處理客戶的 HTTP GET 需求，其處理程式大要如下：

```
protected void doGet( ServletRequest req,
    ServletResponse resp ) {
```

```
    request 之空間單元沒有設定，傳回主卡疊
```

```
    if( req.getParameter( "spaceUnit" ).equals("empty") ){
        System.out.println( "Begin to send main page" );
        sendMain( resp ); // 傳送主卡疊
```

```
    }
```

```
    else
```

```
    {
```

```
        SpaceUnit spaceUnit = findSpaceUnit( req );
```

```
        if( req.getParameter("appliance").equals("empty") )
```

```
        {
```

```
            System.out.println( "send "+spaceUnit );
```

```
            request 設定空間單元，但未設定電器，傳回空間單元卡疊
```

```
            spaceUnit.sendCardDeck( resp, localURL, port );
```

```
        }
```

```
        else
```

```
        {
```

```
            Appliance appliance = findAppliance( req );
```

```
            System.out.println( "appliance =" +appliance );
```

```
            if( req.getParameter("status").equals("empty") )
```

```
            {
```

```
                System.out.println( "get "+appliance+" status" );
```

```
                request 設定空間單元及電器，但未設定狀態，傳回電器狀態卡疊
```

```
                appliance.sendStatusCard( resp, localURL, port );
```

```
            }
```

```
            else
```

```
            {
```

```
                System.out.println( "set "+appliance );
```

```
                request 要求設定電器狀態，先設定電器狀態
```

```
                setApplianceStatus( req );
```

```
                System.out.println( "send "+appliance );
```

```
                再傳回電器卡疊
```

```
                appliance.sendCardDeck( resp, localURL, port );
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

第三章 控制一般家電之智慧型代理程式設計與製作

3-1 系統架構

第二年目標為發展智慧型代理人程式，可學習使用者習慣，代為操作電燈與冷氣機。

本研究設計 eHomeCenter 之軟體架構如圖 3.1：使用者透過伺服器 Server 使用家電，其使用紀錄存入 User Profile 檔案，由代理人基地 Agent Place 中的代理人程式 Agent 學習，配合外來資訊，學得使用者習慣模式，隨後即可依照使用者授權程度，由代理人程式自動透過伺服器操控家電。此處之伺服器係由改進前一章之家電網路伺服器而得。

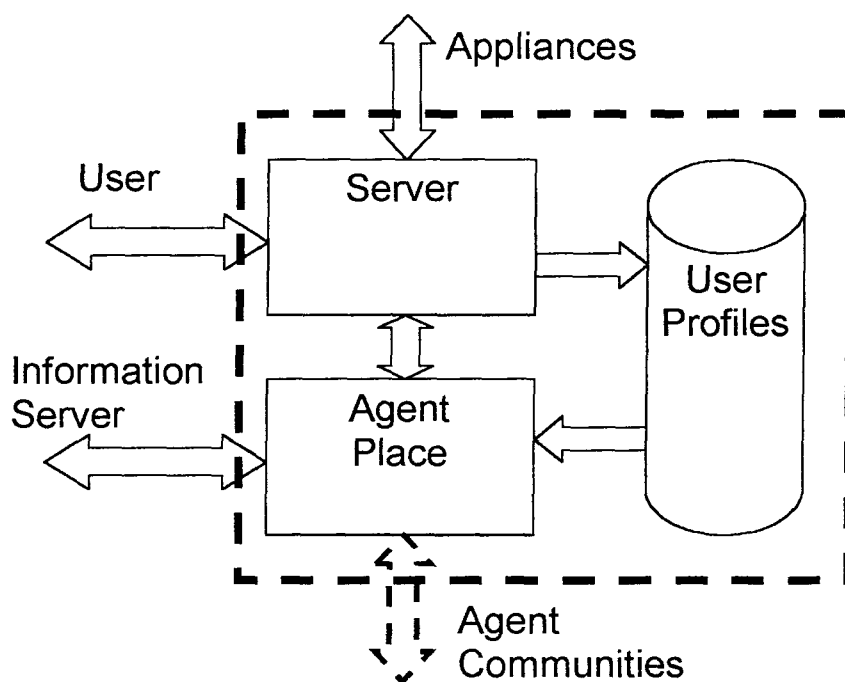


圖 3.1. eHome Center 程式軟體架構

此種代理人程式，可視為協助將家電個人化(personalization)之主要助手，其結構設計如圖 3.2：藉由學習單元 Learning Unit，將使用者使用紀錄整理歸納，得到使用者偏好紀錄檔 Preference，再藉環境資訊，發送操控訊息予 Server。

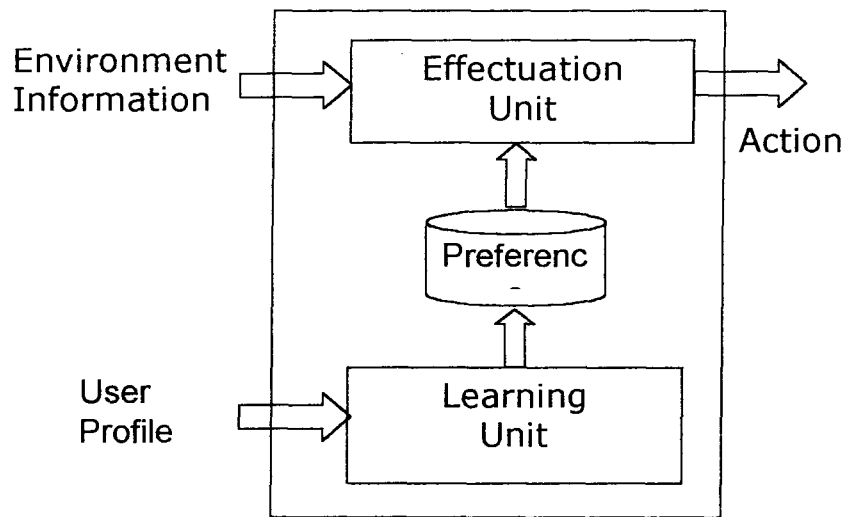


圖 3.2. Personalization Agent 程式軟體架構

圖 3.2 之結構中，最關鍵的便是學習單元。機器學習的方法甚多 [20]，如類神經網路、決策樹技術、貝氏學習法(Bayesian Learning)等等。其中以決策樹學習(Decision Tree Learning)技術較為單純，使用較易，因此採用為個人化代理程式的學習機制。其演算法參考知名之決策樹學習程式 ID3 [20]，顯示於圖 3.3。根據此一演算法，並參考另一以 C 撰寫之著名程式 C4.5 [21]，利用 Java 寫成一個決策樹學習之引擎 JDecisionTree。本計畫主持人並指導大學部學生團隊 Impeccable，利用此一引擎完成「智慧型代理人電視節目推薦系統」，獲得國家高速電腦中心主辦之第一屆全國軟體設計比賽佳作。圖 3.4 為 JDecisionTree 分析歸納某一大學部學生收看電視節目習慣所產生的決策樹。有關決策樹技術之細節將於下一節討論。

DecisionTreeLearning(*examples*, *attributes*, *default*)

0. if *examples* is empty return *default*
1. if all *examples* have the same classification return the classification
2. if *attributes* is empty return MajorityValue(*examples*)
3. *best* = ChooseAttribute(*attributes*, *examples*)
4. *tree* = a new decision tree with root test *best*
5. for each value *vi* of *best* do
 - *examples_i* = {elements of *examples* with *best* = *vi*}
 - *subtree* = DecisionTreeLearning(*examples_i*, *attributes*-*best*, MajorityValue(*examples*))
 - Add a branch to *tree* with label *vi* and subtree *subtree*
6. return *tree*

圖 3.3. Decision Tree Learning 演算法


```

*****  ProgramTest  *****

Read 21 cases ( 4 attributes )

Pruned Decision Tree

Type = Action : Yes ( 2.0 / 0.0 )
Type = Romance : Yes ( 4.0 / 0.0 )
Type = Funny : No ( 3.0 / 0.0 )
Type = History :
| Finish = Comedy : Yes ( 3.0 / 0.0 )
| Finish = Tragedy : Yes ( 0.0 / 0.0 )
| Finish = Insipid : Yes ( 0.0 / 0.0 )
| Finish = Unknown : No ( 1.0 / 0.0 )
Type = Horror : Yes ( 0.0 / 0.0 )
Type = Warm : Yes ( 0.0 / 0.0 )
Type = Science_fiction : Yes ( 3.0 / 0.0 )
Type = Art : Yes ( 2.0 / 0.0 )
Type = Story :
| Class = Movie : Yes ( 0.0 / 0.0 )
| Class = Series : No ( 1.0 / 0.0 )
| Class = Soap_opera : Yes ( 0.0 / 0.0 )
| Class = Cartoon : Yes ( 2.0 / 0.0 )
| Class = Animation : Yes ( 0.0 / 0.0 )

```

圖 3.4. 程式 JDecisionTree 輸出一例

對於電燈與冷氣使用習慣之歸納學習，則須將決策樹學習用於有大小順序之連續變數，這正是決策樹學習技術較弱的一環[21]。本研究團隊參考[21]的討論，構想一完整之解決辦法，說明於 3-3 節。

3-2 決策樹機器學習演算法

決策樹學習可以[20]中的例子來做說明：假設智慧型代理人程式觀察某使用者在各種情況下打網球的決定，製作一份觀察紀錄表如表 3.1。由表可以看出使用者在十四天內，各種屬性(attribute，此處包括天氣 Outlook、氣溫 Temperature、溼度 Humidity、風力 Wind)狀況下打網球的決定(decision)情形。例如，D3 日，天氣陰天，氣溫熱，溼度高，風力弱，使用者決定打網球。我們的目的是希望讓建立一個簡潔的決策模式，可以讓代理人程式依照此一決策模式，快速預測使用者在各種情形下是否決定打網球，以便進行配合動作，如詢問使用者是否須預定網球場地，或邀約球伴等。此種概念亦是近年風行之資料探勘(data mining)的主要研究目標之一。

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

表 3.1. 打網球的紀錄

各種決策模式中，決策樹為一種簡單明瞭的表示法，其形式如圖 3.5。其決策模式以圖 3.5 為例，先看天氣，如果天氣為陰(Overcast)，即決定 Yes。如果天

氣為晴(Sunny)，則再看溼度，若溼度高，則決定為 No，否則為 Yes。如果天氣為雨(Rain)，則檢查風力，如風力強則決定為 No，否則為 Yes。此一決策樹可發現與表 3.1 所作之決策完全相同，但是表達及應用方式要簡便的多。問題是如何由使用習慣紀錄表如表 3.1 學習出如圖 3.5 的決策樹。

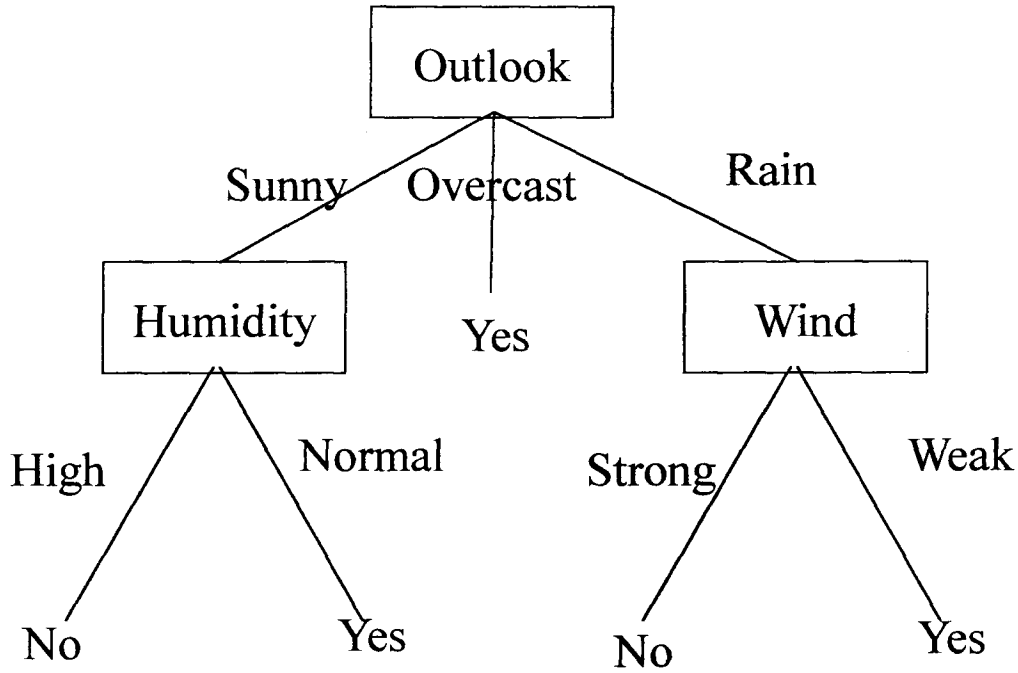


圖 3.5. 決策樹一例

建立決策樹首先要決定哪一個屬性是最佳的判斷標準，這時資訊亂度 (information entropy) 的觀念就很有用了。依據資訊理論 (information theory)，某一事件集合 S 的資訊亂度定義為

$$Entropy(S) = - \sum_{i=1}^c p_i \log_2 p_i \quad (3.1)$$

其中 p_i 為 S 內第 i 個事件的發生機率， c 為 S 內的事件數。關於 (3.1) 的定義推導可以參考 [22]。事件集合 S 的資訊亂度可以看成其資訊混亂之程度，因此如果選好一個屬性作為判斷標準之後，其資訊混亂程度應該降低，因此可以定義屬性選擇前後之亂度增益 (information entropy gain) 為

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (3.2)$$

其中 S 為屬性選擇前之事件集合， A 為所選擇用來分類的屬性， $Value(A)$ 是 A 屬

性的可能值， S_v 為 S 內事件中所有 A 屬性值為 v 之事件所構成之集合。以表 3.1 為例，所有 14 天資料構成集合 S ，分成決定為 Yes(+)和 No(-)的兩個事件，所以其資訊亂度為

$$Entropy([9+,5-]) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} \approx 0.94$$

若是以風力作為分類依據，則風力弱之事件集合為 $S_{Weak}=[6+, 2-]$ ，風力強之事件集合為 $S_{Strong}=[3+, 3-]$ ，其資訊亂度增益為

$$\begin{aligned} Gain(S, Wind) &= Entropy(S) - \\ &\quad (8/14)*Entropy(S_{Weak}) - (6/14)*Entropy(S_{Strong}) \\ &= 0.94 - (8/14)*0.811 - (6/14)*1.00 \\ &= 0.048 \end{aligned}$$

同法可算出若以天氣、溼度、氣溫為分類依據時，資訊亂度增益分別為 0.246、0.151、0.029。因此知道第一個分類屬性最好是天氣，因為其資訊亂度增益最大，代表以其分類時，資料的混亂程度降低最多。此時可得部分完成之決策樹如圖 3.6，由圖可看出，當天氣為陰時，所有範例(example) D3、D7、D12、D13 均屬於 Yes 事件，資訊亂度為 0，因此可直接判斷為 Yes，而天氣為晴或雨時，則無法直接判斷，必須進一步分類。

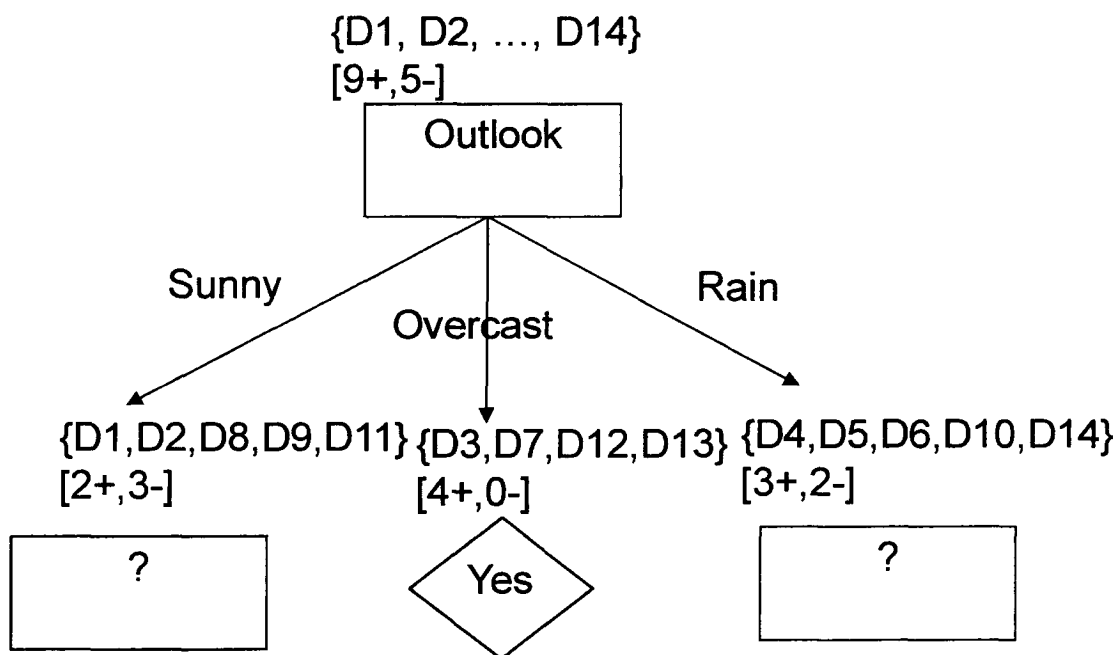


圖 3.6. 部分完成之決策樹

接著考慮天氣晴的情況，此時可以考量的進一步分類屬性為溼度、氣溫、風力，所對應的資訊亂度增益分別為 $Gain(S_{Sunny}, Humidity) = 0.970$ 、 $Gain(S_{Sunny}, Temperature) = 0.57$ 、 $Gain(S_{Sunny}, Wind) = 0.019$ ，因此其進一步的分類屬性以溼度為最佳。同理可找出天氣雨時最好的分類屬性為風力。再往下考慮時發現分類均已確定，可不必再繼續分類，即得出圖 3.5 之決策樹。

以上之過程可總結為圖 3.3 之遞迴演算法。演算法之第 3 步呼叫 **ChooseAttribute** 函數，即在找尋最佳分類屬性。在以上打網球的例子中，我們使用資訊亂度增益作為找尋最佳分類屬性之依據，但是由實際經驗得知 [20][21]，若有某一屬性其可能值相當多時，其資訊亂度增益往往相當高。為減少此種偏向，我們將資訊亂度增益除以分類屬性的分割亂度(split entropy)

$$SplitEntropy(A) = - \sum_{v \in Value(A)} p_v \log_2 p_v \quad (3.3)$$

作為決定最佳分類屬性的標準；方程式(3.3)中的 p_v 代表所考慮的範例中 A 屬性值為 v 的比例。

上述演算法在資料量繁多，且可能包含許多錯誤訊息時，會形成甚多多餘不應有的決策分支，使產生的決策樹異常龐大蕪雜，其錯誤率反較簡單的決策樹高，稱為過份配合(overfitting)，有必要修剪(prune)多餘之分枝。我們在程式中沿用[21]的悲觀估計修剪法(pessimistic pruning)，說明如下：

假定美國國會議員法案投票立場與其黨籍之歷史資料可建立一個決策樹，決策樹的一部分如圖 3.7：

```

physician fee freeze = n:
| adoption of the budget resolution = y: D (151)
| adoption of the budget resolution = u: D (1)
| adoption of the budget resolution = n:
| | education spending = n: D (6)
| | education spending = y: D (9)
| | education spending = u: R (1)

```

圖 3.7. 美國國會議員投票立場與其黨籍之決策樹

由圖可看出在凍結醫師費用法案(physician fee freeze)投反對票(n)者，在預算決議接受案(adoption of the budget resolution)中投贊成票者(y)有 151 位民主黨員

(D(151))，未投票或棄權(u)者有 1 位民主黨員(D(1))，投反對票者，其對教育支出案(education spending)之投票行為為：反對，6 位民主黨員(D(6))；贊成，9 位民主黨員(D(9))；未投票或棄權，1 位共和黨員(R(1))。

使用悲觀估計修剪法時由最末的分支開始，所以我們先考慮圖 3.8 所示的子樹(subtree)

```

|      |      education spending = n: D (6)
|      |      education spending = y: D (9)
|      |      education spending = u: R (1)

```

圖 3.8. 教育支出法案分支子樹

此一子樹所有分支決策錯誤的發生數期望值悲觀估計總和為

$$6*U_{25\%}(0,6) + 9*U_{25\%}(0,9) + 1*U_{25\%}(0,1) = 3.273 \quad (3.4)$$

其中 $U_{25\%}(0,6)$ 代表信心水準 50% 時，從 6 個案例出現 0 個錯誤、6 個正確決策的狀況反推出的決策錯誤機率估計，其細節說明於後。另一方面，如將整個子樹剪除，代以圖 3.9 的單一分支(D(16/1) 代表 16 個例子中，有一個錯誤)

```

|      adoption of the budget resolution = n: D(16/1)

```

圖 3.9. 修剪圖 3.8 子樹後所形成之單一分支

則其錯誤的發生數期望值悲觀估計為

$$16*U_{25\%}(1,16) = 2.512 \quad (3.5)$$

較(3.4)之數值為小，表示修剪後其錯誤發生數期望值變小，因此可將圖 3.8 之子樹以圖 3.9 之單一分支取代，圖 3.7 之決策樹因此變成如圖 3.10 所示：

```

physician fee freeze = n:
|      adoption of the budget resolution = y: D (151)
|      adoption of the budget resolution = u: D (1)
|      adoption of the budget resolution = n: D (16/1)

```

圖 3.10. 第一次修剪圖 3.7 後之決策樹

此一決策樹錯誤的發生數期望值悲觀估計為

$$151*U_{25\%}(0,151)+1*U_{25\%}(0,1)+16*U_{25\%}(1,16) = 4.642 \quad (3.6)$$

若將所有分支修剪，成為單一支如圖 3.11 所示：

physician fee freeze = n: D(168/1)

圖 3.11. 完全修剪圖 3.7 後之決策樹

其錯誤的發生數期望值悲觀估計為

$$168*U_{25\%}(1,168) = 2.610 \quad (3.7)$$

因此完全修剪之決策樹其可能之錯誤數較原先決策樹為小，所以圖 3.7 之決策樹應完全修剪為圖 3.11。

其次說明 $U_{25\%}(k,n)$ 的意義。考慮一隨機實驗，其出錯之機率為 p ，則進行 n 次實驗，出現 k 次錯誤之機率為

$$f(k,n) = \binom{n}{k} p^k (1-p)^{n-k} \quad (3.8)$$

其中 $\binom{n}{k}$ 代表 n 取 k 的組合數。但是我們的情況不同，我們並不知道每次的錯誤機率 p ，所以要倒過來，假設已知 n 次實驗中出現 k 個錯誤，則其機率 p 的最佳估計值為 $y = k/n$ ，以 y 為中心，估計 p 的範圍。估出來的 p 範圍如果很小，則 p 落在此範圍的機率就小，若範圍大，則 p 落於此範圍的機率就大。假定我們希望找出的範圍可使 p 落於其中的機率(稱為信心水準 confidence level)為 50%，則此範圍之上限與下限 p_{UL} 及 p_{LL} 如圖 3.12 所示，其中的上限 p_{UL} 即為 $U_{25\%}(k,n)$ 。圖中縱軸的標示 PDF 代表 p 的機率密度函數(probability density function)，信心水準即為 p_{UL} 及 p_{LL} 之間機率密度函數曲線下的面積。

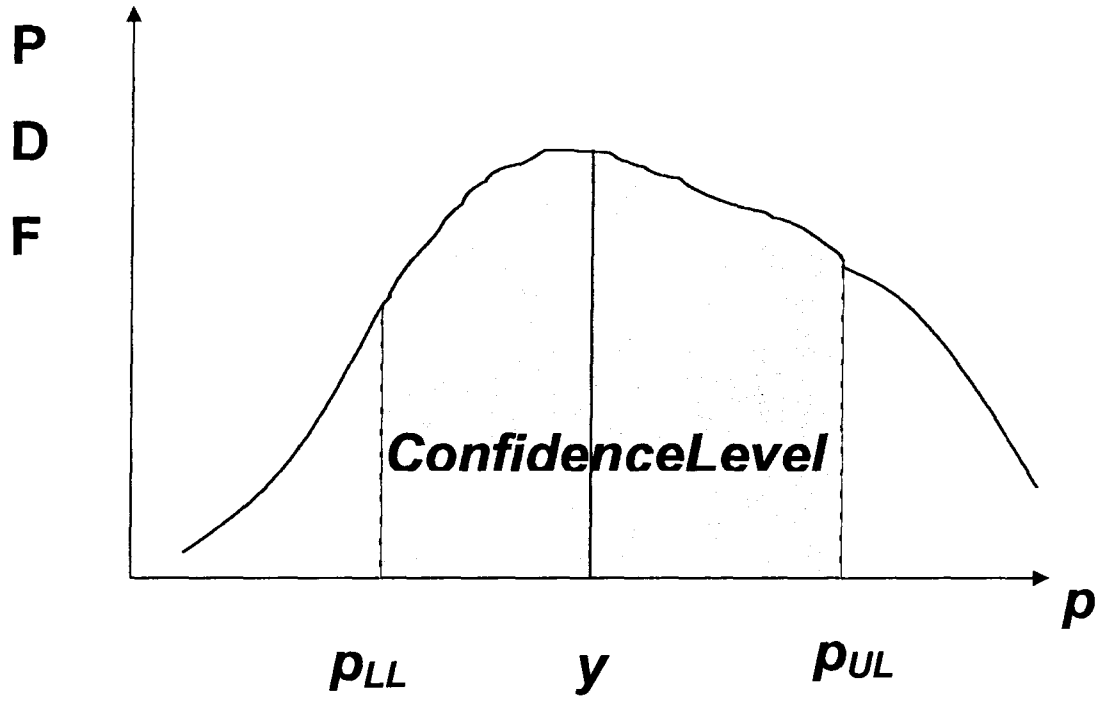


圖 3.12. p 的估計範圍與信心水準

由[23]的推導，可知 p_{UL} 及 p_{LL} 分別滿足以下兩式的 p 值：

$$\sum_{r=0}^k \binom{n}{r} p^r (1-p)^{n-r} = (1 - \text{ConfidenceLevel}) / 2 \quad (3.9)$$

及

$$\sum_{r=k}^n \binom{n}{r} p^r (1-p)^{n-r} = (1 - \text{ConfidenceLevel}) / 2 \quad (3.10)$$

另一方面，在 [23][24] 可以找到如下的關係：

$$\sum_{r=0}^k \binom{n}{r} p^r (1-p)^{n-r} = 1 - F(p; k, n-k-1) \quad (3.11)$$

$$\begin{aligned} \sum_{r=k}^n \binom{n}{r} p^r (1-p)^{n-r} &= 1 - \sum_{r=0}^{k-1} \binom{n}{r} p^r (1-p)^{n-r} \\ &= F(p; k-1, n-k) \end{aligned} \quad (3.12)$$

其中的累計 Beta 分布函數定義為

$$F(x; \alpha, \beta) = \frac{(\alpha + \beta + 1)!}{\alpha! \beta!} \int_0^x t^\alpha (1-t)^\beta dt = I_x(\alpha + 1, \beta + 1) \quad (3.13)$$

而不完全 Beta 函數定義為

$$I_x(a, b) = \frac{1}{B(a, b)} \int_0^x t^{a-1} (1-t)^{b-1} dt \quad (3.14)$$

$$B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} = \frac{(a-1)!(b-1)!}{(a+b-1)!}$$

所以(3.9)可表為

$$1 - F(p_{UL}; k, n - k - 1) = (1 - \text{ConfidenceLevel}) / 2 \quad (3.15)$$

$$F(p_{UL}; k, n - k - 1) = (1 + \text{ConfidenceLevel}) / 2$$

而(3.10)則成為

$$F(p_{LL}; k - 1, n - k) = (1 - \text{ConfidenceLevel}) / 2 \quad (3.16)$$

不完全 Beta 函數的計算可以利用 [24] 內的公式計算列表，再由表可計算其反函數，解出(3.15)、(3.16)中的 p_{UL} 及 p_{LL} ，即可算出 $U_{50\%}(k, n)$ 。

本研究實作的決策樹類別程式結構如圖 3.13 及 3.14。圖 3.13 顯示決策樹包含樹根 (root)，為一個節點 Node。介面 Node 可分為中間節點 IntermediateNode 及葉節點 LeafNode，葉節點僅包含決策值 Value，而中間節點則包含該節點的判斷屬性 Attribute，及此一屬性各數值連接下一層節點的連結 Link。此種架構是一種設計樣式 Composite [25]，而在巡行(Traverse)整個決策樹時，也用到另一個設計樣式 Visitor [25]。圖 3.14 為決策樹建立過程中所用到的範例 Example 類別程式架構，一個範例包括一群屬性與數值建立的集合，及所採取的決策值。另外有一個類別 Examples，包含類別 Example 的集合，並能計算其子集合的資訊亂度。節點類別內並實作了修剪功能，決策樹建立後，可用以修剪決策樹。在全部的實作中，發現最基礎的程式類別為集合。Java 的程式庫 java.util 雖有定義集合 Set 的介面，但因其中不包含集合基本運算如交集、聯集、差集等，因此我們依需要自行撰寫類別程式 JSet、LSet，分別以 ArrayList、LinkedList 實作具有集合基本運算之集合類別，另外我們也實作了類別 MSset，除基本集合運算外，並可將個別元素依出現次數多寡給予權重，以便在輸入資料有欠缺時，以出現較多的元素代替。

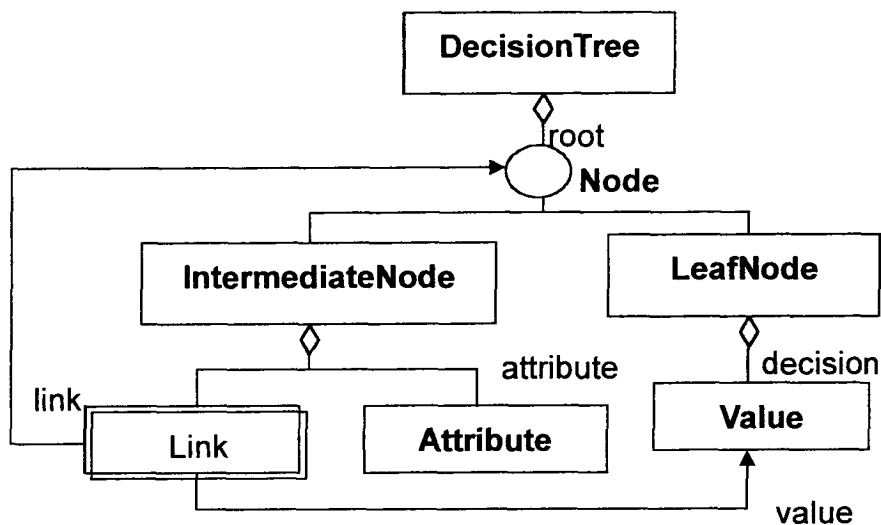


圖 3.13. 決策樹類別程式結構

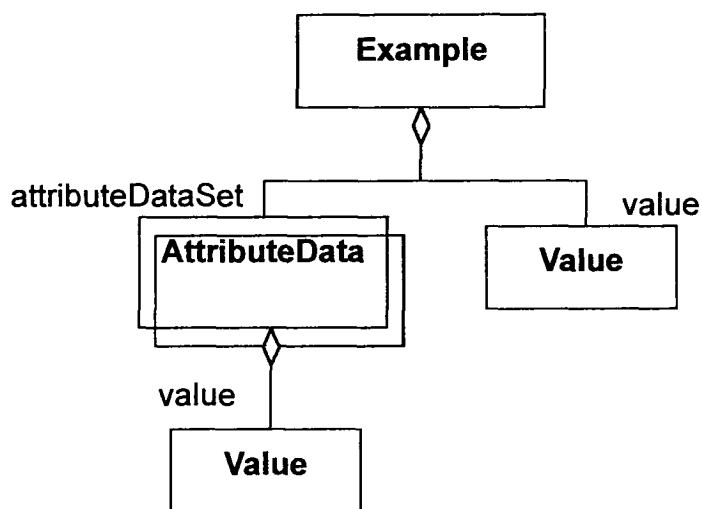


圖 3.14. 範例之類別程式結構

3-3 有序連續變數之處理

前一節決策樹學習的討論假設屬性為分類型變數(categorical variable)，僅有有限的幾個離散(discrete)值，而且這些值之間不必有大小順序之關聯；例如，屬性氣溫(Temperature)，其可能值為熱(Hot)、溫(Mild)、涼(Cold)三種，其間大小順序關聯對決策樹分類並不重要。但是實際氣溫可能是溫度計量到的數字，在某一範圍內連續變化，而且有大小的順序，稱為有序連續型變數(ordered continuous variable)。有序連續型變數的可能數值無窮多，因此不能直接套用決策樹學習的演算法，必須將其數值分為若干個區間，每一個區間給一個標籤，標籤成為分類型變數之值，即可使用決策樹學習了。例如，將氣溫 18°C 以下訂為涼，18°C 至

28°C 為溫，28°C 以上則為熱，即可將資料以分類型變數呈現。問題是這些區間的邊界如 18°C、28°C 應當如何決定？

參考文獻 [21] 提出如下做法：首先將數據由小而大排列，例如氣溫數據有以下六個(單位為 °C)：17.8、20.1、23.5、26.3、29.0、31.4，則假設將於相鄰數據之中點(18.95、21.8、24.9、27.65、30.25)之中選擇兩個作為熱、溫、涼的分界線；而選擇之方法為產生五個中點任選兩個的所有組合，利用(3.2)計算每種組合的分類結果對於原問題的資訊亂度增益，再選出亂度增益最大的組合決定熱、溫、涼的分界。

從 n 個中點選取 m 個的所有組合，可以用如圖 3.15，參考[26]所完成的 Java 程式，在利用建構元(constructor)建立物件後，逐一呼叫 next() 產生。

```
public class CombinationGenerator {
    private int n;
    private int m;
    private int c[];
    private int copy[];
    private boolean flagMore;
    private int j;
    public CombinationGenerator( int n, int m ) {
        this.n = n;
        this.m = m;
        flagMore = true;
        c = new int[m];
        copy = new int[m];
        for( int k=0; k<m; ++k ) {
            c[k] = k;
        }
        j = m - 1;
    }
    public boolean hasMore() { return flagMore; }

    public int[] next( ) {
```

```

if( flagMore ) {

    // make a copy
    for( int k=0; k<m; ++k ) {
        copy[k] = c[k];
    }

    // update
    boolean updated = false;
    while( !updated ) {
        // check if there is no more combination
        if( j < 0 ) {
            flagMore = false;
            return copy;
        }
        if( c[j] < n-m+j ) {
            c[j] = c[j] + 1;
            for( int i = j+1; i < m; ++i ) {
                c[i] = c[i-1] + 1;
            }
            updated = true;
            j = m - 1;
        } else {
            j = j - 1;
        }
    }
}
return copy;
}
}

```

圖 3.15. 由 n 個物品選取 m 個的組合產生程式

3-4 操作實例

依據前述三節的描述，我們完成了一個 Java 代理人基地程式 eHomePlace1，配合第二章的微伺服器，構成一個初步的 eHome Center。在 eHomePlace1 中包含

有學習使用者開燈與使用冷氣習慣的代理人程式 LampAgent、DimmerAgent、AirconditionAgent。其中 LampAgent 可以依據使用者在各種不同光度下的開燈關燈行為紀錄(圖 3.16 為一例，第一行為使用例編號，第二行為光度計讀數，第三行為開關紀錄)，利用決策樹學習技術決定開關燈的亮度界限，而在使用者授權之狀況下，自動開燈或關燈。DimmerAgent 可以由使用者在不同光度下調整變光燈亮度的使用紀錄(圖 3.17 為一例，其中第三行為使用者調整之變光燈亮度，100 代表開至最亮，0 代表關燈，一共分為五個等級 100、75、50、25、0)，學習各亮度等級的對應光度邊界，進而自動調整變光燈亮度。AirconditionAgent 的動作原理與 DimmerAgent 相似，其在各種溫度下的冷氣溫度設定紀錄範例如圖 3.18，其中第二行為室內溫度(°C)，第三行為冷氣溫度設定(°C)，在本例中只將其設為 40(高於一般室溫，開冷氣)及 15(低於一般室溫，關冷氣)。

配合這些代理人程式，前一章的微伺服器已經改寫，可以依照設定，紀錄電器使用情況，產生使用紀錄檔案(user profiles)，供智慧型代理程式使用。此一初步 eHome Center 雛型，業經實際以小型燈具及模擬冷氣之小電扇實際測試，各智慧型代理程式確實可以依照學習所得，自動開關或設定電器狀態。

```
C1 10 ON
C2 93 OFF
C3 15 ON
C4 18 ON
C5 21 ON
C6 25 ON
C7 14 ON
C8 12 ON
C9 28 OFF
C10 3 ON
C11 87 OFF
C12 89 OFF
C13 20 ON
C14 85 OFF
```

圖 3.16. 簡單燈具使用紀錄之一例

```
C1 10 100
```

C2	93	0
C3	15	75
C4	18	75
C5	21	50
C6	25	50
C7	14	75
C8	12	75
C9	28	50
C10	3	100
C11	87	0
C12	89	0
C13	20	50
C14	85	0

圖 3.17. 變光燈使用紀錄之一例

C1	20	40
C2	33	15
C3	25	40
C4	28	15
C5	21	40
C6	26	40
C7	24	40
C8	22	40
C9	27	40
C10	31	15
C11	23	40
C12	29	15
C13	19	40
C14	35	15

圖 3.18. 冷氣溫度設定紀錄一例

eHome Center 系統啟動時，首先執行伺服器，其次執行 eHomePlace1，出現圖 3.19 之起始畫面，主視窗顯示尚未有代理人程式啟動。此時可於選單 Agent 中選擇產生(Create)新代理人，或啟動(Activate)先前存檔之代理人。選單 List 中可以列出目前使用中(Active)或已存檔(Saved)之代理人清單。

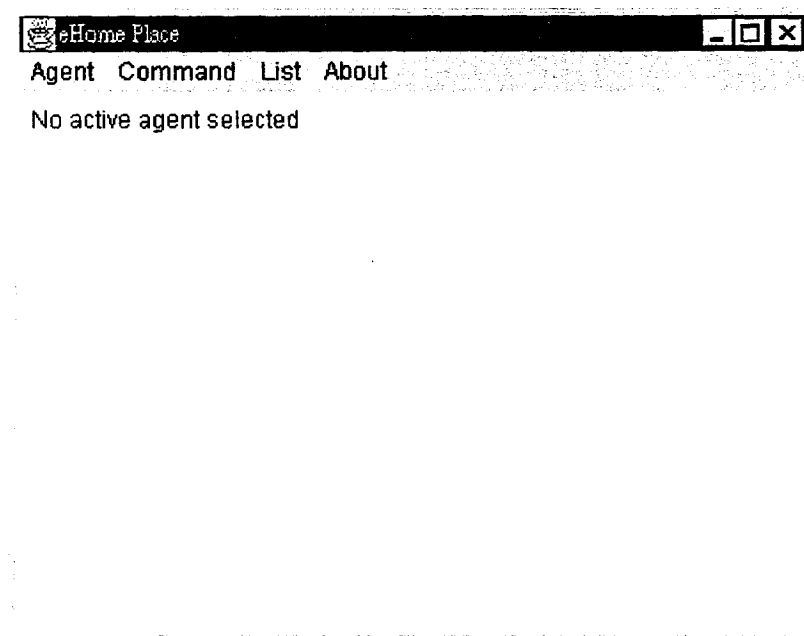
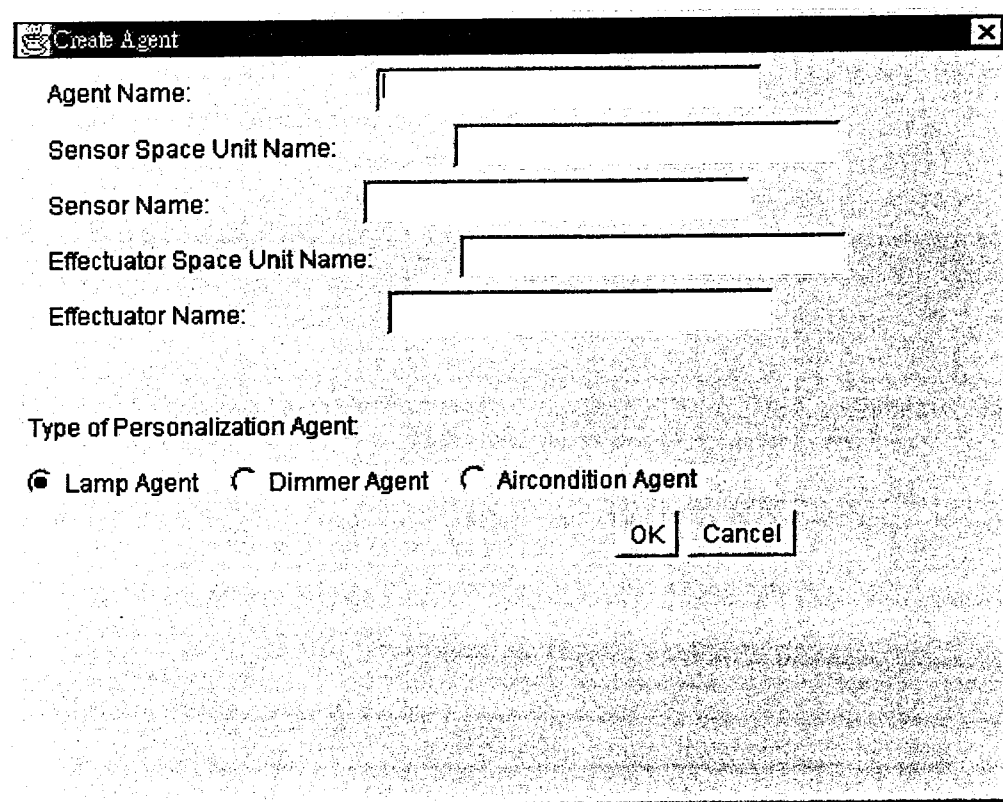


圖 3.19. 程式 eHomePlace1 之起始畫面

假設選擇產生新代理人，則會出現圖 3.20 之表單，使用者需填入代理人名稱、感應器(Sensor)如光度計所在之空間單位與感應器名稱、控制電器(Effectuator)如簡單燈具所在之空間單位與電器名稱、代理人種類等資料，按下 OK 按鈕後即可產生新代理人。代理人產生時，同時完成學習之動作，因此產生代理人前須利用伺服器準備好對應之電器使用紀錄檔，檔名為代理人名稱加上"Profile"再加副檔名"exs"。

A screenshot of a 'Create Agent' dialog box. It features five text input fields for 'Agent Name', 'Sensor Space Unit Name', 'Sensor Name', 'Effectuator Space Unit Name', and 'Effectuator Name'. Below these fields is a section titled 'Type of Personalization Agent' with three radio button options: 'Lamp Agent' (which is selected), 'Dimmer Agent', and 'Aircondition Agent'. At the bottom right of the dialog are 'OK' and 'Cancel' buttons.

Create Agent

Agent Name: _____

Sensor Space Unit Name: _____

Sensor Name: _____

Effectuator Space Unit Name: _____

Effectuator Name: _____

Type of Personalization Agent

☒ Lamp Agent ☐ Dimmer Agent ☐ Aircondition Agent

OK Cancel

圖 3.20. 產生新代理人需填寫之表單

若是選擇啟動先前存檔之代理人，則會出現如圖 3.21 之畫面，顯示先前已存有 Lamp1Agent、Dimmer1Agent、Aircondition1Agent 三個代理人，使用者可以滑鼠點選，按下 OK 按鈕，即可啟動所指定之代理人。假設指定 Lamp1Agent，則主視窗顯示目前處理之代理人為 Lamp1Agent，感應器為 LightSensor1，控制電器為 Lamp1，如圖 3.22。

產生或啟動代理人後，即可由選單 Command 選擇選項 Run，使代理人開始由感應器接收訊息，並依照已學習所得的動作邊界，判斷設定電器狀態，例如開關燈具，或設定變光燈亮度及冷氣溫度。使用者亦可隨時利用 Command 中的另外兩個選項，暫停(Suspend)或回復(Resume)代理人之工作。亦可以 Agent 中的選項，儲存(Save)代理人資料或停止(Dispose)代理人工作，或結束(Quit)整個程式。

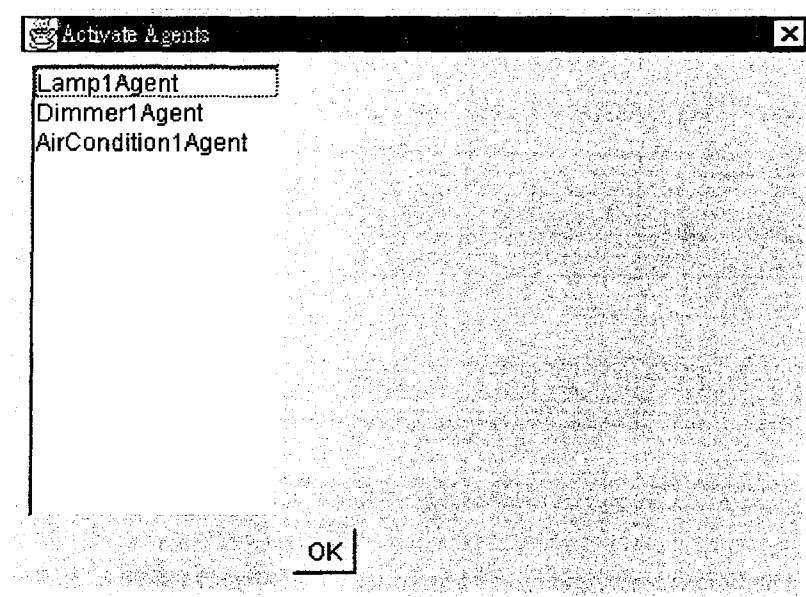


圖 3.21. 啟動先前存檔之代理人程式畫面一例



圖 3.22. 啟動 Lamp1Agent 後之主視窗畫面

3-5 系統主要結構設計

整體系統架構如圖 3.1，其中伺服器已在上一章討論，此處僅略作修改，因此不在此討論。代理人基地為產生及啟動代理人程式之所在，在此介紹代理人基地程式 eHomePlace1 之結構及演算法。

程式採取 Model-View 架構，主程式包含視窗(View)及代理人基地模型(Model)兩部分，好處是視窗與模型切割清楚，易於維護。代理人基地模型類別(class)程式稱為 EHomePlaceModel，含有兩個鏈結串列(Linked List) activeAgents 及 savedAgents，分別儲存已被啟動之代理人與已存檔之代理人，另外有一個變數 currentAgent，存放目前處理的代理人。主要的方法成員(method)如表 3.2。

方法成員名稱	功能
createAgent	產生新代理人，令為 currentAgent
learnProfile	currentAgent 由使用紀錄學習
runAgent	currentAgent 開始工作
disposeAgent	停止 currentAgent (無法回復)
suspendAgent	暫停 currentAgent (可以回復)
resumeAgent	回復 currentAgent
activateAgent	啟動代理人，令為 currentAgent
saveAgent	儲存 currentAgent
setCurrentAgent	重設 currentAgent
readInStoredAgents	讀入已存檔之代理人

表 3.2. EHomePlaceModel 主要方法成員

代理人類別在此有 LampAgent、DimmerAgent、AirconditionAgent 三種，均實作介面(interface) EHomeAgent，介面程式如圖 3.23。

```
public interface EHomeAgent {
    final int eHomePort = 9200;
    final String created = "created";
    final String learned = "learned";
    final String running = "running";
    final String suspended = "suspended";
    public String getName();
    public String getType();
    public String getSensorSpaceUnit();
    public String getSensorName();
    public String getEffectuatorSpaceUnit();
    public String getEffectuatorName();
    public String getState();
}
```

```

public void learn( String profileName ); // 學習
public void go(); // 執行工作
public void activate(); // 啟動
public void save(); // 存檔
public void suspend(); // 暫停
public void resume(); // 回復
public void stop(); // 停止
}

```

圖 3.23. 介面 EHomeAgent 程式

三種代理人程式除實作介面 EHomeAgent 外，並實作介面 Runnable，以便每一個代理人均能有一條執行緒(thread)單獨執行。除 EHomeAgent 定義之方法成員外，各代理人程式另外定義一些各自內部使用的特殊方法成員，列表如表 3.3。

類別	方法成員	功能
LampAgent	senseLightSensor	取得光度資料
	senseSimpleLamp	取得簡單燈具開關狀態
	turnOn	開燈
	turnOff	關燈
DimmerAgent	senseLightSensor	取得光度資料
	setDimmer	設定變光燈亮度
AirconditionAgent	senseTemperatureSensor	取得室溫資料
	setOnTemperature	設定冷氣溫度

表 3.3. 各代理人程式的特殊方法成員

各代理人執行工作時，在其執行緒之方法成員 run() 之內，進行工作之步驟細節略有不同，大致如下：

1. 與伺服器建立連線
2. 取得感應器資料
3. 依照學習所得之動作邊界資料，決定如何設定電器
4. 設定電器新狀態

各代理人內均有學習單元(Learning Unit)，代理人基地模型在建立代理人時，即要求代理人的學習單元完成學習動作，因此代理人執行工作時已學得動作邊界資

料。學習單元以 3-2 節及 3-3 節之討論為基礎，主要功能為由讀入的使用紀錄檔案，學習感應器資料分類及動作對應之資料邊界。注意此處實作之三種代理人，其使用紀錄中均只有一個屬性變數及一個決策變數，因此其對應的決策樹為只有一層的簡化決策樹，此處實作的學習單元因此只實作一簡化的決策樹，降低複雜度。第 3-2 及 3-3 節的完整決策樹將應用於下一章的電視節目推薦代理人程式。

第四章依使用者偏好推薦電視節目之智慧型代理程式設計與製作

4-1 系統架構

第三年工作為開發能依使用者偏好推薦電視節目之智慧型代理程式，此一工作將可為未來的互動電視，以及電腦與視聽家電結合的系統添加頗高的使用價值。

圖 4.1 為電視節目推薦系統的架構，與本人指導，獲得第一屆全國軟體設計比賽之大學生團隊 Impeccable 所完成之「智慧型代理人電視節目推薦系統」[27]相同，但將在一些細節加以改進。架構可分成電視節目播放、智慧型代理程式、電視節目資訊伺服網站三部分。電視節目播放部分可以是裝有機上盒(Set-Top Box)的有線電視或具有電視收視功能的電腦或另一子計畫所研究的 Havi 電視。智慧型代理程式裝置於機上盒或電腦內，可以紀錄統計使用者收看電視的時間，提供分析學習使用者偏好之用，並可依據學習結果，及電視節目資訊伺服網站提供之電視節目時間表，比對與使用者偏好相符合的各個節目，回傳並通知使用者節目時間及內容型態，方便使用者收看。電視節目資訊伺服網站提供最新節目訊息，每週更新，由電視節目提供廠商設計與維護，可提供一個統一的窗口，供各使用者之智慧型代理程式讀取當週節目資訊，此外，此一網站也提供各個節目的屬性資料。

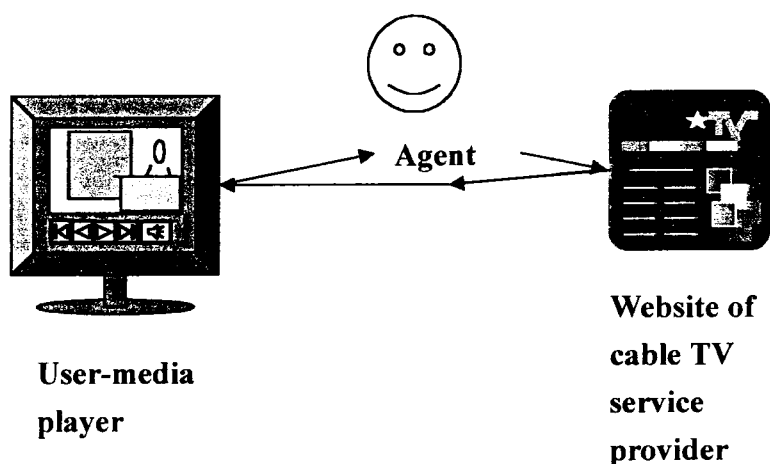


圖 4.1. 電視節目推薦系統架構[27]

4-2 電視節目屬性

欲分析使用者偏好，必須找出電視節目的屬性。但電視節目包羅萬象，很難如前一章打網球的例子訂出通用於所有節目的通用屬性。Impeccable 設計的電視

節目推薦系統將電視節目分成 Class (種類)、Language (語言)、Type (型式)、Ending (結局) 四項屬性。種類的資料值分成 Movie (電影)、Series (影集)、Soap_opera (連續劇)、Animation (卡通)、Show (綜藝)、Talk_show (訪問座談)、News (新聞)、Report (報導性節目)等八種，語言的資料值有 English (英語)、Japanese (日語)、Mandarin (國語)、Cantonise (客語)、Taiwanese (台語)、Korean (韓語)、Others (其他)等七種，型式資料值則為 Action (動作)、Romance (愛情)、Comedy (喜劇)、Detective (偵探)、Horror (恐怖)、Warm (溫馨)、Science_fiction (科幻)、Knowledge (知識)、News (新聞)等九種，結局有 Comedy (喜劇)、Tragedy (悲劇)、Insidious (平淡無奇) 三種資料值。這種屬性的分法並不理想，因為其中不同屬性的資料值有相當強的關聯性，例如型式的資料值大部分僅與戲劇性節目有關，而小部分只與訪問座談、新聞、報導性節目等相關，而大部分訪問座談、新聞、報導性節目等很難說它的結局是喜劇還是悲劇。另外，有些人偏好看某些演員或導演的作品，有人愛看三大男高音的演唱會轉播，這種資料值事先不確定的屬性沒有涵括在 Impeccable 的系統之中。除此之外，Impeccable 系統規定一個電視節目的一個屬性只能有一個資料值，而一般節目有時可能同時擁有多個資料值，例如，電影臥虎藏龍兼具動作與愛情成分。另一方面，其種類的屬性值也太少，有些節目例如空中英語教室勉強分入報導性節目或訪問座談都不太妥當。

基於上述說明，本研究改用如下方式來規劃電視節目的屬性分配：首先將電視節目分為若干大類，每一大類有其自己的屬性描述，如此就各大類的屬性均與其分類緊密相關，屬性與某些節目無關的情況可以消除。另外我們也加入了不特定值的屬性，處理如電影演員或節目主持人的偏好，並且允許一個屬性有多個資料值的情形，因此對屬性的處理應較 Impeccable 系統完整。

首先考慮電視節目的分類，依照傳播學者蘇燕輝的說明[28]，可以有以下幾種分法：若以法規規定內容區分，依照我國廣播電視法分類，有新聞及政令宣導節目、教育文化節目、公共服務性節目、大眾娛樂節目四種。以發行方式來分，有錄影帶、影片、現場直播三種。以節目內容分，有戲劇性節目(分單元劇、連續劇)、新聞性節目(分播報、評論、與深度報導)、綜藝性節目(如歌唱、競賽、才藝表演等)、訪問座談性節目(以國是論壇、社教、及公共政策為主題)、教學性節目(空中教學、親子教室、社會推廣教育為主)、音樂性節目(音樂欣賞、風景旅遊搭配及 MTV 等)、介紹性節目(如各國國情、旅遊、文化等內容)、宗教性節目(如某某大師講解經文、交換宗教上之經驗等)等八種。另外，也可以訴求對象來區分，例如：兒童節目、青少年節目、婦女節目、成人節目、老人節目、軍人節目、農業節目等。蘇先生另外還提到“Writing for Television and Radio”一書將電視節目分類為廣告與聲明(commercial and announcement)、採訪與脫口秀節目(interview and talk programs)、新聞和體育節目(news and sports)、人物專輯及紀錄片製作(features and documentaries)、音樂、連續劇及喜劇(music, variety and

comedy)、公益、教育以及兒童類節目(corporate, educational, and children's programs)等六類。

參考以上的分類方式，以及黃葳葳教授在[29]提到的節目型式，我們將電視節目分為戲劇、新聞、綜藝、訪問座談、教學、音樂、知識、宗教人生、體育運動等九類，每一類具有的屬性及資料值如表 4.1：

類別	屬性	資料值
戲劇	種類	影片、單元劇、連續劇
	呈現方式	電影、動畫、電視劇、偶戲、歌仔戲
	語言	中、閩、客、英、日、韓
	型態	動作、冒險、警匪偵探、社會寫實、戰爭、西部、男女之間、同性戀、喜劇、歌舞、歷史、傳記、恐怖、親情倫理、科幻、懸疑、教育、勵志、情色
	觀眾設定	普、護、輔、限
	主要演員	(不定)
	導演	(不定)
	製作單位	(不定)
新聞	種類	播報、評論、社會新聞深度報導、國際新聞深度報導、財經股市
	語言	中、閩、客、原(住民)、英、日
	主持人	(不定)
綜藝	種類	歌唱、益智競賽、才藝表演、短劇、模仿表演、娛樂新聞、綜合
	觀眾設定	普、護、輔、限
	主持	(不定)
	製作	(不定)
訪問座談	主題	國是論壇、公共政策、趨勢發展、社會八卦
	語言	中、閩、客、原(住民)
	主持	(不定)
教學	科目	英語、電腦、國文史地、商業企管、美術工藝、插花烹飪、美容瑜珈、體操韻律、拳術氣功
	對象	社會人士、大專青年、青少年、兒童、幼兒
	主持人	(不定)
	製作單位	(不定)
音樂	種類	現場錄影轉播、音樂欣賞、MTV、卡拉 OK
	樂種	聲樂、器樂、音樂劇、國樂、本國流行音樂、

		日韓流行音樂、西洋流行音樂
	主持人	(不定)
	主要表演者	(不定)
	主要表演曲目	(不定)
知識	種類	旅遊地理、文化藝術民俗、歷史傳記人物、自然生物、科學技術、美食、醫藥健康
	主持人(主述人)	(不定)
	製作單位	(不定)
宗教人生	種類	大師說法、經驗交換、儀式轉播
	宗教	佛、基督、天主、民間宗教、回、儒、人生體驗
	主持人	(不定)
體育運動	種類	棒壘球、籃球、撞球、高爾夫、保齡球、賽車、體操韻律溜冰、水上運動、搏擊
	主要參與隊伍	(不定)
	主要參與運動員	(不定)

表 4.1. 電視節目分類及屬性

利用上述分類及屬性資料值，我們曾比對實際的電視節目表，發現可以包含大部分的電視節目特性。

4-3 提供節目資訊之網站程式

以上電視節目分類及屬性確定後，我們假設電視節目提供廠商會建立電視節目資訊伺服網站，每週於固定時間更新節目表資訊，供各使用者的智慧性代理人程式下載。節目表資訊包含頻道號碼、播映日期及時間、節目名稱、節目分類、節目各屬性資料值等，其格式如下

```
{ <channel> + <numberOfPrograms> +
  { <date> + <time> + <programName> + <category> +
    { <numberOfValues> + {<attributeValue>} } } }
```

其中大括弧{ }代表其中的資料項目反覆。圖 4.2 為一個電視節目表檔的例子，有些資料如主要演員、導演、製作單位等，我們手邊沒有確切資料，就用演員一、導演一、日本公司一等名稱取代。

12 2

2/18/2003 18:00:00 名偵探柯南 戲劇 1 影片 1 動畫 2 日 中
1 警匪偵探 1 普 1 演員一 1 導演一 1 日本公司一

2/18/2003 19:00:00 華視晚間新聞 新聞 1 播報 1 中 1 李四端
47 3

2/18/2003 18:00:00 搶救生態--搶救非洲萬獸之王 知識 1 自然生物
1 主持人一 1 美國國家地理雜誌

2/18/2003 19:00:00 國家地理饗宴:非洲毒王 知識 1 自然生物 1
主持人二 1 美國國家地理雜誌

2/18/2003 23:00:00 亞洲花絮集:七月遊魂 知識 1 文化藝術民俗 1
主持人三 1 美國國家地理雜誌

48 2

2/18/2003 18:00:00 穿梭時空談科技:終極冒險家 知識 1 歷史傳記
人物 1 主持人四 1 製作單位一

2/18/2003 23:00:00 穿梭時空談科技:終極冒險家 知識 1 歷史傳記
人物 1 主持人四 1 製作單位一

53 3

2/18/2003 18:00:00 我的這一班 戲劇 1 單元劇 1 電視劇 1 中
1 教育 1 普 1 演員二 1 導演二 1 公共電視

2/18/2003 19:00:00 我們的島 知識 2 自然生物 文化藝術民俗 1
主持人五 1 製作單位二

2/18/2003 20:00:00 孽子 戲劇 1 連續劇 1 電視劇 1 中 1
親情倫理 1 普 1 演員三 1 導演三 1 公共電視

56 1

2/18/2003 23:00:00 慾望城市 戲劇 1 單元劇 1 電視劇 1 英
1 男女之間 1 輔 1 演員四 1 導演四 1 美國公司六

57 1

2/18/2003 23:00:00 小鬼當家3 戲劇 1 影片 1 電影 1 英 1
喜劇 1 普 1 演員五 1 導演五 1 美國公司一

59 1

2/18/2003 23:00:00 電子情書 戲劇 1 影片 1 電影 1 英 1
男女之間 1 普 2 梅格萊恩 湯姆漢克斯 1 導演六 1 美國公司二

60 1

2/18/2003 23:00:00 史帝芬史匹柏呈獻幽浮入侵第七集 戲劇 1 單元
劇 1 電影 1 英 1 科幻 1 普 1 演員六 1 導演七 1 史帝芬
史匹柏

61 1

2/18/2003 23:00:00 回到未來2 戲劇 1 影片 1 電影 1 英 1

科幻	1	普	1	演員七	1	導演八	1	美國公司三	
63	1								
2/18/2003	21:00:00	教父續集	戲劇	1	影片	1	電影	1	英 1
社會寫實	1	護	1	演員八	1	導演九	1	美國公司四	
64	2								
2/18/2003	18:00:00	晶兵總動員	戲劇	1	影片	1	電影	1	英 1
科幻	1	普	1	演員九	1	導演十	1	美國公司五	
2/18/2003	23:00:00	晶兵總動員	戲劇	1	影片	1	電影	1	英 1
科幻	1	普	1	演員九	1	導演十	1	美國公司五	

圖 4.2. 電視節目表檔一例

我們也完成了簡化的電視節目資訊伺服器(server)及客戶端程式(client)，可分別裝設於電視節目提供廠商的網站及使用者的電腦或機上盒之中。客戶端程式由使用者的智慧代理人程式控制，藉由 HTTP 網路通信協定，每週於固定時間向伺服器程式索取電視節目表檔。因此伺服器程式和客戶端程式基本上就是一對功能簡化的 HTTP 伺服器與客戶端程式。由於兩端程式均由電腦控制，因此不需實作圖形使用介面。

伺服器程式假定其所聆聽之連接埠(port)為 8080，以一個無窮迴圈不斷聆聽伺服器通訊插槽(server socket)是否有客戶端程式連結，偵測到有客戶端程式連結時，即利用由 accept() 方法獲得之客戶端通訊插槽(socket)，建立一個內有執行緒(thread)之 HTTPServer 物件，處理客戶端的請求。伺服器主程式如圖 4.3。

```
public class TVProgramInformationServer {
    static final int PORT = 8080; //default port
    public static void main(String[] args) {
        ServerSocket serverConnect = null;
        try {
            serverConnect = new ServerSocket(PORT); //listen on port
            System.out.println(
                "\nListening for connections on port " + PORT +
                "... \n");
            while (true) {
                HttpServer server = new
                    HttpServer(serverConnect.accept());
            }
        } catch (IOException e) {
```

```

        System.out.println( "Connection error :" + e );
    }
}
}

```

圖 4.3. 電視節目資訊伺服器端主程式

HTTPServer 類別實作 Runnable 介面，在其執行緒執行的 run() 之中處理客戶端請求。此處假定客戶端只會傳來取得檔案的 HTTP 指令 GET，並呼叫 doGet() 方法，讀取檔案後傳回。此處假定此一伺服器固定傳回最新之電視節目表檔 TVProgram.tbl，故忽略客戶端傳來之檔案名稱。圖 4.4 為 HTTPServer 的原始程式碼。

```

class HttpServer implements Runnable
{
    static final File WEB_ROOT = new File(".");
    Socket connect;
    public HttpServer(Socket connect)
    {
        this.connect = connect;
        Thread threadRunner = new Thread(this);
        threadRunner.start(); //start thread
    }
    public void run()
    {
        try {
            BufferedReader in = new BufferedReader( new
                InputStreamReader(connect.getInputStream()));
            PrintWriter out = new PrintWriter(
                connect.getOutputStream());
            String input = in.readLine();
            StringTokenizer parse = new StringTokenizer(input);
            String method = parse.nextToken().toUpperCase();
            if (!method.equals("GET")) {
                sendStatus( out, 501, "Not Implemented", method );
            }
            else {
                doGet( out );
            }
        }
    }
}

```

```

        connect.close();
    }
    catch (IOException e)
    {
        System.err.println("Server Error: " + e);
    }
}

private void doGet( PrintWriter out ) {
    try {
        File file = new File(WEB_ROOT, "TVProgram.tbl");
        int fileLength = (int)file.length();
        FileInputStream fileIn = new FileInputStream(file);
        byte[] fileData = new byte[fileLength];
        fileIn.read(fileData);
        fileIn.close();
        BufferedOutputStream dataOut = new
            BufferedOutputStream(connect.getOutputStream());
        dataOut.write(fileData, 0, fileLength);
        dataOut.flush();
        dataOut.close();
    }
    catch (IOException e) {
        sendStatus(out, 404, "File not found",
            "TV Program Info missing");
    }
}

private void sendStatus( PrintWriter out, int code,
    String status, String msg ) {
    out.println("HTTP/1.0 " + code + " " + status );
    out.println("Server: HttpServer 1.0");
    out.println("Date: " + new Date());
    out.println("Content-Type: text/html");
    out.println();
    out.flush();
}
}

```

圖 4.4. HTTPServer 類別程式碼

客戶端程式則首先利用通訊插槽建立與伺服端的連結，再傳出 GET 的指令，再將收到的資料輸出。圖 4.5 為其主程式型式之原始程式碼。在我們往後的討論中，此一程式將成為智慧型代理程式所呼叫的一個方法，其連結之網址將由此處之 localhost 改為電視節目資訊伺服網站之網址，輸出也將改成由程式指定的資料檔。

```
public class TVProgramInformationClient {
    public static void main(String[] args) {
        try {
            InetAddress address = InetAddress.getLocalHost();
            Socket s = new Socket( address, 8080 );
            int c;
            OutputStream out = s.getOutputStream();
            String str = "GET " + "TVProgram.tbl" + " HTTP/1.0 \n";
            byte buf[] = str.getBytes();
            out.write(buf);
            InputStreamReader in = new
                InputStreamReader( s.getInputStream() );
            while( (c=in.read())!= -1 ) {
                System.out.print((char) c);
            }
            s.close();
        } catch (Exception e) {
            System.out.println(
                "TVProgramInformationClient error: " + e );
        }
    }
}
```

圖 4.5. 電視節目資訊客戶端主程式

4-4 使用者收看時間紀錄與分析

在 Impeccable 的系統中，使用者必須填寫問卷，對每一個所收看的節目，就偏好程度從 1 到 5 加以評分，以便系統利用決策樹演算法學習使用者的偏好。但是要使用者對每個節目評分，對使用者來說實在是相當麻煩的工作。所以我們改成由電腦或 Havi 系統監看使用者收看節目的狀況，統計節目收看時間的比例，進而推斷使用者偏好的程度。

本系統的智慧型代理人每週讀入電視節目表檔，配合電腦監看程式或 Havi 系統傳來之收看時間紀錄檔，處理之後產生電視節目評分檔。收看時間紀錄檔格式為

```
{ <date> + <time> + <channel> }
```

而電視節目評分檔格式為

```
{ <programName> + <rating> + <category> +  
  { <numberOfValues> + {<attributeValue>} } }
```

圖 4.6 及圖 4.7 分別為收看時間紀錄檔及配合圖 4.2 之電視節目表所得之電視節目評分檔的例子。收看時間紀錄檔中，以頻道號碼 -1 代表關機。

2/18/2003	18:00:00	64
2/18/2003	18:03:05	53
2/18/2003	18:56:35	12
2/18/2003	19:38:40	47
2/18/2003	19:58:21	53
2/18/2003	20:48:17	-1
2/18/2003	23:04:15	48
2/18/2003	23:15:18	47
2/18/2003	23:25:56	48
2/18/2003	23:57:41	56
2/19/2003	00:05:12	57
2/19/2003	00:08:25	56
2/19/2003	00:08:55	59
2/19/2003	00:15:12	60
2/19/2003	00:16:13	61
2/19/2003	00:17:18	64
2/19/2003	00:18:15	63
2/19/2003	00:20:03	48
2/19/2003	00:21:33	-1

圖 4.6. 收看時間紀錄檔一例

名偵探柯南 5 戲劇 1 影片 1 動畫 2 日 中 1 警匪偵探 1 普
 1 演員一 1 導演一 1 日本公司一
 晶兵總動員 1 戲劇 1 影片 1 電影 1 英 1 科幻 1 普 1 演
 員九 1 導演十 1 美國公司五
 我們的島 5 知識 2 自然生物 文化藝術民俗 1 主持人五 1 製作單
 位二
 國家地理饗宴:非洲毒王 1 知識 1 自然生物 1 主持人二 1 美國國
 家地理雜誌
 我的這一班 5 戲劇 1 單元劇 1 電視劇 1 中 1 教育 1 普 1
 演員二 1 導演二 1 公共電視

圖 4.7. 電視節目評分檔一例

程式首先讀入收看時間紀錄檔及電視節目表檔，建立節目與收看時間資料的對應表。對應表中的收看時間資料是一個資料錄(record)，包含節目是否正在收看之旗標、節目開始時間、節目結束時間、及由各次收看的時間長短與比例所構成的陣列列串(array list)。同一個節目的收看時間可能因使用者轉台再轉回而分割成幾個片段(圖 4.8)，因此在計算某次收看的總時間時，必須照各片段收看時間的起點與終點，和節目的起始時間比較，求出此一片段收看時間與節目播放時間重疊的時間長度，再將他們加總。所得的總收看時間要除以節目收看起訖間距得到一個節目收看時間比例。節目收看起訖間距定義為開始收看與結束收看之間的時間(包含中間轉台到其他節目的時間)，因此當節目收看起訖間距超過某一門檻(我們設定為 60 秒)，所得節目收看時間比例愈高，就表示使用者愈專注，愈有興趣；而節目收看時間比例愈低，表示使用者真正收看的時間比例愈少，顯示使用者愈沒有興趣。最後評分時，把評分分為 1 到 5 五等，節目收看時間比例在 0.2 以下者，評為 1 分；節目收看時間比例在 0.2 到 0.4 之間時，則評 2 分；依此類推，當節目收看時間比例在 0.8 以上時評分設為 5 分。如果節目收看起訖間距低於門檻無法評分時，則予以跳過不處理。

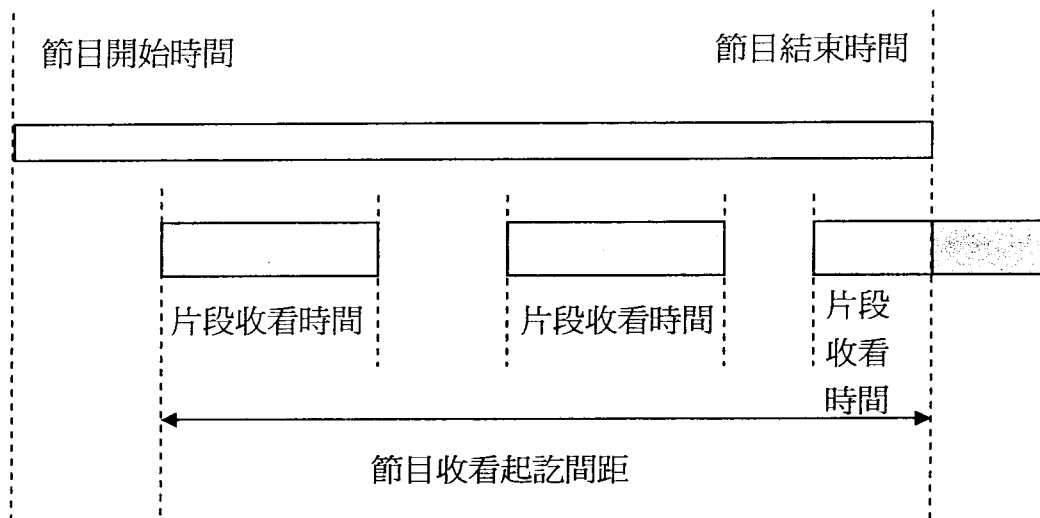


圖 4.8. 電視節目收看時間紀錄及處理示意圖

4-5 使用者偏好之學習

Impeccable 系統因為使用四個屬性描述所有電視節目，因此只需要以一個決策樹來學習使用者的偏好。我們的系統將電視分為九類，每類各有多寡不等的屬性，因此使用九個決策樹學習單元來分別學習各分類的偏好，這與 A. D. Shapiro 提出的結構化歸納技術(structured induction)[30]有相類似的觀念。從另一個角度看，如果將節目的大分類也看成一個屬性，則我們的屬性架構看起來亦與 J. R. Quinlan 中所說的結構化屬性(structured attributes) [21]相似，但是 Quinlan 所提到的僅是一個複雜屬性的處理，與我們有多種屬性系統的情況並不相同，因此 Quinlan 建議的方法如將屬性資料值化至最低層，再用群組(grouping)技術整理為多個屬性，或者將每一層定義一個新屬性，均不適用於我們的系統。

學習使用者偏好大致可以分成範例檔準備及決策樹學習兩步驟。準備範例檔時，先讀入各分類的原始屬性資料，其次處理電視節目評分檔，將之改寫為各個分類的範例檔。範例檔的格式如下：

```
{ <programName> + {<attributeValue>} + <decisionValue> }
```

圖 4.9 為圖 4.7 之電視節目評分檔產生之戲劇節目範例檔，另外還會產生知識節目範例檔、新聞節目範例檔等等，如果電視節目評分檔內沒有某一分類的節目，則產生的該分類範例檔即為一個空檔案。同時注意如果節目的某幾個屬性都有多個資料值，則該節目將產生多個範例，每個範例包含一種屬性資料值組合。例如電影「電子情書」的「主要演員」有梅格萊恩及湯姆漢克斯兩人，則產生兩個範

例，一個的「主要演員」為梅格萊恩，另一個的「主要演員」為湯姆漢克斯。圖 4.9 檔案的第一個節目「名偵探柯南」有兩個範例，一個的「語言」為日文，另一個的「語言」為中文，也是一樣的道理。這種做法可以使智慧型代理人程式在學習時不會遺漏節目的某些屬性值，但也有可能加強了該節目的其他屬性資料值在學習時的地位，造成偏頗；不過假如節目夠多，這種偏頗可能沒有那麼嚴重。

名偵探柯南	影片	動畫	日	警匪偵探	普	演員一	導演一	日本公司一	5
名偵探柯南	影片	動畫	中	警匪偵探	普	演員一	導演一	日本公司一	5
晶兵總動員	影片	電影	英	科幻	普	演員九	導演十	美國公司五	1
我的這一班	單元劇	電視劇	中	教育	普	演員二	導演二	公共電視	5

圖 4.9. 由圖 4.7 之電視節目評分檔產生之戲劇範例檔

處理電視節目評分檔之同時，也由電視節目表檔收集各分類不定值屬性的資料值，將其定為該屬性的資料值集合，產生與該分類範例檔對應之屬性檔。例如，戲劇分類之屬性「主要演員」原為不定資料值之屬性，但讀完圖 4.2 的電視節目表檔後，共收集有 11 個資料值，因此在我們對應的戲劇範例檔中，「主要演員」不再是不定資料值的屬性，而是可能有 11 種資料值的確定屬性。同樣的方法，圖 4.2 之電視節目表檔可決定「導演」有 10 種資料值，「製作單位」有 9 種資料值，由此可以寫出戲劇分類的屬性資料檔如圖 4.10。屬性資料檔格式如下：

```
{ <decisionAttribute> + <numberOfDecisionValues> +
  {<decisionValue>} +
  { <attribute> + <numberOfAttributeValues> +
    {<attributeValue>} } }
```

第一行為決策值即評分範圍，第二行起為各屬性資料值範圍，後面加上“NotOrdered”表示該屬性並非有序連續變數，因此不需用上一章 3-3 節的方法處理。同時注意「主要演員」、「導演」、「製作單位」的屬性資料值為收集電視節目表檔對應資料而得。

```
Decision 5 1 2 3 4 5
種類 3 影片 單元劇 連續劇 NotOrdered
呈現方式 5 電影 動畫 電視劇 偶戲 歌仔戲 NotOrdered
語言 6 中 英 日 閩 客 韓 NotOrdered
```

型態 19 動作 冒險 警匪偵探 社會寫實 戰爭 西部 男女之間 同性
 戀 喜劇 歌舞 歷史 傳記 恐怖 親情倫理 科幻 懸疑 教育 勵志
 情色 NotOrdered
 觀眾設定 4 普 護 輔 限 NotOrdered
 主要演員 11 演員一 演員二 演員三 演員四 演員五 梅格萊恩 湯姆
 漢克斯 演員六 演員七 演員八 演員九 NotOrdered
 導演 10 導演一 導演二 導演三 導演四 導演五 導演六 導演七 導
 演八 導演九 導演十 NotOrdered
 製作單位 9 日本公司一 公共電視 美國公司六 美國公司一 美國公司二
 史帝芬史匹柏 美國公司三 美國公司四 美

圖 4.10. 戲劇屬性資料檔

產生各分類之屬性資料檔及範例檔後，即可分別交付各分類對應之決策樹學習單元，產生使用者各分類偏好之決策樹。如果某分類之範例檔為空檔案，亦即使用者沒有收看該類節目之紀錄，則對應之決策樹亦為空決策樹(null)。各分類決策樹學習單元的學習機制與上一章介紹的單一決策樹學習單元學習演算法相同，此處不再贅述。

4-6 電視節目之推薦

智慧型代理人程式完成使用者偏好學習之後，每週取得最新節目表之後，即可依照所學得的使用者偏好及節目屬性資料推薦新電視節目給使用者。推薦過程可分為三步驟，即待評分節目檔案準備、節目評分、節目排序及整理。系統首先讀入新的電視節目表檔，建立節目資料列串，並將每一節目給予一個代碼，並存入節目的頻道與播放時間等資料。其次由節目資料列串，依照節目分類產生各分類的待評分節目檔案，格式如下：

```
{ <programID> + {<attributeValue>} }
```

與準備範例檔時相同，此處同一個節目有可能產生幾個待評分的例子。以圖 4.2 的電視節目表檔為例，產生的戲劇節目待評分檔如圖 4.11。

```

0 影片 動畫 日 警匪偵探 普 演員一 導演一 日本公司一
0 影片 動畫 中 警匪偵探 普 演員一 導演一 日本公司一
6 單元劇 電視劇 中 教育 普 演員二 導演二 公共電視
8 連續劇 電視劇 中 親情倫理 普 演員三 導演三 公共電視
9 單元劇 電視劇 英 男女之間 輔 演員四 導演四 美國公司六
  
```

10	影片	電影	英	喜劇	普	演員五	導演五	美國公司一
11	影片	電影	英	男女之間	普	梅格萊恩	導演六	美國公司二
11	影片	電影	英	男女之間	普	湯姆漢克斯	導演六	美國公司二
12	單元劇	電影	英	科幻	普	演員六	導演七	史帝芬史匹柏
13	影片	電影	英	科幻	普	演員七	導演八	美國公司三
14	影片	電影	英	社會寫實	護	演員八	導演九	美國公司四
15	影片	電影	英	科幻	普	演員九	導演十	美國公司五

圖 4.11. 由圖 4.2 電視節目表檔產生的戲劇節目待評分檔

每一分類的待評分節目檔接著送交各分類的決策樹學習單元進行評分，其方法與上一章的討論相同。評分完畢會產生各分類節目的已評分檔，以圖 4.11 的檔案為例，學習單元會在每一個節目後加上評分，產生的已評分檔包含節目代碼及評分，如圖 4.12。如果某一節目的某一屬性出現學習單元先前未曾學過的資料值，如電影魔鬼終結者之「主要演員」為阿諾史瓦辛格，先前沒出現過，則此節目之評分定為“CanNotDecide”。

0	5
0	5
6	5
8	5
9	5
10	1
11	1
11	1
12	1
13	1
14	1
15	1

圖 4.12. 對圖 4.11 之檔案的評分結果

接著將各分類的已評分檔讀出，併入一個推薦節目列串，如果節目之評分為“CanNotDecide”，則列入另一個無法判斷的節目列串。推薦節目列串的各项節目接著進行 **insertion sort**，依照評分由高而低排列，再照節目的代碼，消去同一代碼而排名在後的節目，以除去重複的情形。最後得到的列串便可配合節目資料列串，將節目名稱、評分、頻道、播放時間等顯示給使用者選擇。無法判斷的節目列串，同樣除去重複後，也可以提供使用者參考選看。

4-7 操作實例

上一章提到的代理人基地程式 eHomePlace1，在此略加改變，成為 eHomePlace2 程式。為偵錯方便，eHomePlace2 只容納本章的電視節目推薦代理人程式。

系統啟動時，出現與圖 3.19 相同的起始畫面。主視窗顯示尚未有代理人程式啟動。此時可於選單 Agent 中選擇產生(Create)新代理人，或啟動(Activate)先前存檔之代理人。代理人產生或啟動後，主視窗顯示目前的代理人程式為 TVAgent0，如圖 4.13，其 Sensor 及 Effectuator 空白，表示 TVAgent0 不需要感應器及致動器。

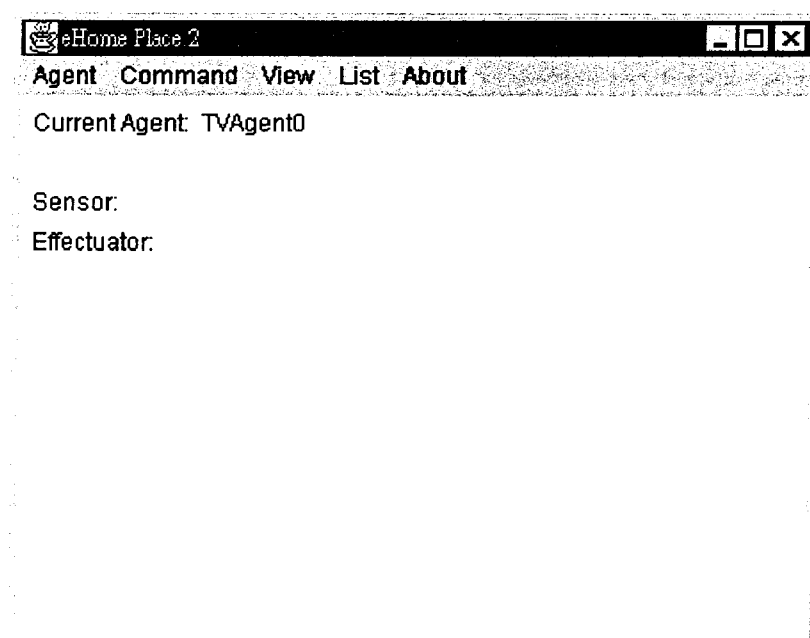


圖 4.13. 產生 TVAgent 後之主視窗畫面

電視節目推薦代理人產生或啟動後，由選單 Command 選擇選項 Run，即可於每週固定時間(此處定為星期一上午九點)至電視節目資訊伺服網站取得節目表，並依使用者偏好推薦節目。若系統在此固定時間恰好關機，則開機時系統會檢查本週是否已取得節目表，若尚未取得本週節目表，則會自動連線擷取節目表，推薦節目。取得節目表後，若尚無使用者偏好資料，或距離上次學習已有四週，則代理人程式會由累積的使用者收看資料，重新學習使用者偏好，再進行節目推薦。推薦節目時，代理人程式會顯示依照評分高低排列之推薦節目表如圖 4.14。使用者可由節目表中選擇想要收看的節目，代理人程式會在節目播放時間到時，顯示訊息提醒使用者收看。此項提醒功能技術上並不困難，但由於時程有

限，目前尚未完成。

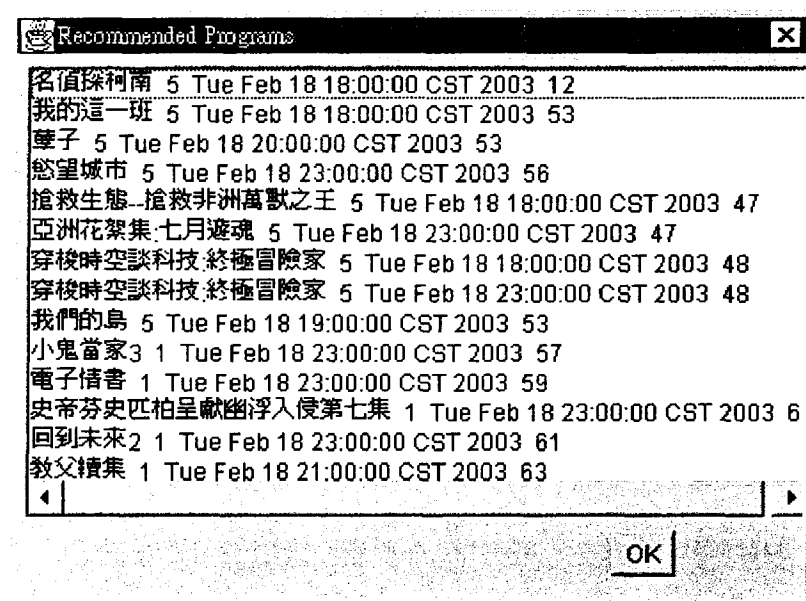


圖 4.14. 推薦節目列表視窗

4-8 系統主要結構設計

系統結構主要仍與圖 3.1 相同，但此時之代理人基地程式(agent place)不必與伺服器相接，直接與使用者互動；另一方面，Havi 家電系統紀錄使用者收看之時間與頻道，寫入 user profile，由代理人程式讀入，作為學習使用者偏好之用。同時，代理人程式也可連線至電視節目資訊伺服網站，讀取節目資訊表。以下介紹代理人基地程式 eHomePlace2 及電視節目推薦代理人程式 TVAgent 之結構與演算法。

eHomePlace2 程式同樣採取 Model-View 架構，其模型類別程式 EHomePlaceModel 與上一章 3-5 節介紹的同名程式大致相同，僅增加了對話盒顯示功能，以便顯示所推薦的電視節目表。電視節目推薦代理人類別程式 TVAgent 同樣實作圖 3.23 的介面 EHomeAgent 及介面 Runnable，另外定義的方法成員如表 4.2，各方法成員的工作概念已於 4-4、4-5、4-6 三節介紹。TVAgent 類別內有九個學習單元，使用第 3-2 節的決策樹學習法，分別學習使用者對九大分類節目的偏好。

方法成員	功能
prepareExamplesFile	準備範例檔
buildProgramList	建立節目資料列串
prepareTestFile	準備待評分節目檔
recommendPrograms	推薦節目

表 4.2. 電視節目推薦代理人程式 TVAgent 的特殊方法成員

TVAgent 執行工作時，其執行緒方法成員 run() 所進行之工作流程大致如下：

1. 每四週定時學習使用者偏好
2. 每週定時至電視節目資訊伺服網站取得節目表
3. 取得節目表即依所學使用者偏好，對各節目評分，並依評分高低推薦
4. 使用者可由推薦節目表選擇欲收看節目，代理人程式屆時會提醒使用者 (未完成)

評分工作由 RatingSetter 物件完成。RatingSetter 的主要功能為依照 4-4 節說明，讀入電視節目表檔，配合電腦監看程式或 Havi 系統傳來之收看時間紀錄檔，處理之後產生電視節目評分檔。成員變數主要包括節目總表 gProgTbl、收看時間紀錄列串 gLogTbl、節目收看紀錄表 progWatchingTbl。節目總表 gProgTbl 為一個 TreeMap，其鍵(key)為頻道號碼，值(value)為頻道的節目表 ChannelProgramTable。頻道節目表 ChannelProgramTable 類別含有一個 ArrayList，由該頻道的節目項 ProgramItem 合成，節目項 ProgramItem 包含播放時間及節目資料 Program。收看時間紀錄列串 gLogTbl 是一個 ArrayList，由收看時間紀錄項 LogEntry 構成，LogEntry 包含收看時間及收看頻道號碼。節目收看紀錄表 progWatchingTbl 是一個 HashMap，以節目 Program 為鍵，收看時間紀錄 WatchingRecord 為值。收看時間紀錄 WatchingRecord 包括節目開始時間、節目結束時間、收看開始時間、收看結束時間、及一個收看時間片段紀錄 TimeRecord 的列串。TimeRecord 包含收看時間片段之時間長度，及其所占節目播出時間的比例。RatingSetter 類別的主要方法成員如表 4.3 所示。

方法成員	功能
readInGrandProgramTable	讀入電視節目表檔，建立 gProgTbl
readLogData	讀入收看時間紀錄檔，建立 gLogTbl

buildProgramWatchingTable	建立 progWatchingTbl
finishWatchingRecord	完成 WtchingRecord
determineWatchingTimesAndDuration	決定收看時間長度
finishProgramWatchingTable	完成 progWatchingTbl
writeOutExamplesFile	寫出評分完成之節目範例檔

表 4.3. 類別 RatingSetter 的主要方法成員

第五章 結論

本研究計劃已完成 eHome Center 之伺服器及智慧型代理人軟體之設計與實作。第一年利用 PC 環境發展完成接受 HTTP，送出 WML 網頁之伺服器，並藉 RS-232 有線介面控制電燈與冷氣機，驗證 eHome 以無線電手機 WML 瀏覽器控制家電之觀念。第二年以無線遙控電燈及冷氣機，加強伺服器功能，使其能接受 WAP，並開發完成具有學習使用者操作習慣能力之簡單智慧型代理人軟體，以便代替使用者進行日常家電操作。第三年延伸第二年完成之具有決策樹學習能力之智慧代理人軟體，開發出依使用者收視偏好推薦電視節目之智慧型代理人系統。我們認為智慧型家電以及互動電視即將進入一般家庭，而家電使用的個人化也是未來的趨勢，因此本研究開發的技術及經驗，將有相當大的實用價值。其中的使用者偏好學習及推薦技術也可用於為使用者推薦過濾電影、音樂、書籍、電腦遊戲等。

參考文獻

- [1] <http://www.wapforum.org/what/technical.htm>
- [2] <http://ailp12.csie.ntu.edu.tw/WAP>
- [3] <http://agents.www.media.mit.edu/group/agents/>
- [4] <http://www.cs.cmu.edu/~softagents/>
- [5] <http://www.cs.umbc.edu/agents/>
- [6] <http://www-cdr.stanford.edu/ABE/>
- [7] <http://research.ibm.com/iagents/>
- [8] <http://agents.hpl.hp.com/overview.htm>
- [9] <http://www.cyc.com/>
- [10] <http://www.generalmagic.com/>
- [11] P. Maes, "Agents that reduce work and information overload," *Communications of the ACM*, vol. 37, no. 7, pp. 31 – 40, 1994.
- [12] A. Dutta-Roy, "Networks for homes," *IEEE Spectrum*, vol. 36, no. 12, pp. 26-33, December 1999.
- [13] <http://www.homeautomation.org/>
- [14] <http://www.cebus.com/>
- [15] <http://www.x10.com/>
- [16] <http://www.lonworks.com/>
- [17] 祥泉, *WAP & WML 手機網站開發手冊*, 知城, 民國 89 年.

- [18] <http://www.forum.nokia.com/main.html/>
- [19] D. R. Callaway, *Inside Servlets*, Reading: Addison Wesley, 1999.
- [20] T. Mitchell, *Machine Learning*, McGraw-Hill, 1997.
- [21] J. R. Quinlan, *C4.5: Programs for Machine Learning*, San Mateo: Morgan-Kaufmann, 1993.
- [22] 戴顯權, *資料壓縮*, 2.1 節, 紳藍 2001.
- [23] A. M. Mood and F. A. Graybill, *Introduction to the Theory of Statistics*, 2nd ed., New York: McGraw-Hill, 1963, Section 11.6.
- [24] M. Abramowitz and I. Stegun ed., *Handbook of Mathematical Functions*, New York: Dover, 1964, Sections 26.5 and 26.2.
- [25] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1994.
- [26] S. G. Aki, *The Design and Analysis of Parallel Algorithms*, Prentice-Hall.
- [27] 何宛倫, 吳賀翔, 黃仁暉, 王松豐, 葉彌妍, *智慧型代理人電視節目推薦系統—個人化的資訊服務*, 國家高速電腦中心第一屆全國軟體設計比賽參賽報告, 台灣大學電機系, 2001。
- [28] 蘇燕輝, *電視節目企劃與類型認知*, 2002 年 6 月, <http://www.ccpct.org.tw/share/mroom/>。
- [29] 黃葳葳, *衛星暨有線電視節目管理*, 1988 傳播管理新思潮—傳播管理研討會, 民國 87 年 4 月 25 日, 高雄。
- [30] A. D. Shapiro, *Structured Induction in Expert Systems*, Turing Institute Press, 1987.