

應用於數位家庭網路之智慧型萬用遙控器控制技術

王威眾

工業技術研究院 電腦與通訊工業研究所

wilson_wang@itri.org.tw

摘要

近年來數位家庭這個題目相當的火熱，各產官學界對於此一產業亦大力推動，目前各類傳統家電正面臨汰舊換新，伴隨著是各類數位化的家電裝置的到來，在數位家庭中，如何將所有的數位家電串聯起來並提供一個整合的控制平台系統可同時存取控制所有的家電，將可更加提升數位家電的便利性，提供一個真正未來化的”數位家庭”。

智慧型萬用遙控器控制技術目的在於制定一個整合性的數位控制平台，能夠針對各種不同類型的數位家電裝置，提供一個共通的存取控制平台，任何一種類的家電裝置加入，無須經由任何的驅動程式安裝與設定動作，智慧型萬用遙控系統即能新增一組控制模組，並且此控制平台將可移植在不同的系統上，可真正達到隨時隨地利用隨手的控制裝置來控制數位家庭裝置。

本文將就數位家庭的網路環境、智慧型萬用遙控系統架構、技術及開發工具做簡單之介紹。

關鍵詞：數位家庭、遙控器、UPnP、URC、XML。

Abstract

The goal of universal remote console technology is to build an integrated control framework. While adding any kind of home device, there is no need to have any extra driver setup and configuration. URC framework will create a new control module automatically. In this way we can control easily in a digital home.

Key word : remote control、UPnP、URC、XML

1. 前言

當今使用者對於資訊家電的依賴性日趨加強，當各類的資訊家電透過特定的中介軟體互相連接，家中的客廳擺上連接內外網路的家用開道器，廚房裡有了自動提供食譜的網路冰箱，陽台的洗衣機會連網通知經銷商它的馬達出了問題，數位家庭生活於焉展開，家庭中最具代表性的場所便是客廳，「決戰客廳」的口號於是喊出。在家庭網路的應用中，各項數位家電裝置該如何串接而彼此溝通扮演了一個關鍵性的角色，如 WLAN、

802.15.4/Zigbee、ECHONET、LonWorks、UPnP、1394、X10... 等，又若能集合所有數位家電裝置的控制於一手，更是便利且重要的議題。

2. 數位家庭相關技術介紹

2.1 中介軟體之介紹

數位家庭的各類裝置可能採用各式各樣的網路環境，因此要提供一個中介軟體以供兩種或兩種以上的應用軟體轉換及傳遞資訊，主要介紹 UPnP 及 Jini，如下所述：

2.1.1 UPnP

2.1.1.1 什麼是 UPnP

UPnP(Universal Plug and Play)是 UPnP Forum 在 1999 年提出的通訊協定。隨著越來越多數位裝置的出現，如何能夠以簡單設定、安裝方式，有效提供裝置對作業系統的隨插即用(Plug and Play)能力便顯得更加重要。UPnP 採用了此重要的概念，在網路環境內，能夠對數位裝置與服務項目做搜尋(discovery)與控制(control)的動作。UPnP 並不只是單純隨插即用技術的延伸，它可真正達到免設定(zero-configuration)與自動搜尋來自各廠商所製造的裝置系統，這一切的動作對於使用者來說，都會是一個自動運作而不須手動設定的程序。

2.1.1.2 UPnP 元件

UPnP[9]主要包含了兩個元件分別為 Device 與 CP(Control Point)，分別描述於下：

- (1) Device：Device 是在 UPnP 環境內可受控制的服務或裝置，在 Device 上我們可定義各類可提供的服務內容，又或者在裝置的配置方面可以是一個巢狀的裝置結構，舉類說明，在錄放影機裝置上可提供快倒轉功能服務、時間顯示服務，而在 TV/VCR 的裝置上，我們將可看到電視與錄放影機的組合裝置。
- (2) Control Point：CP 在 UPnP 環境內扮演控制端的角色，具備有 discovery 與 control 的功能，在進行完 discovery 程序後，CP 將可以進行以下動作：

- 取得 device description 資訊並得到相關的服務內容。
- 針對服務項目做出呼叫控制(Action)的動作。
- 針對服務項目做註冊通知的動作，一旦服務內容的狀態改變，device 將會傳遞事件通知至 CP。

2.1.1.3 UPnP Phase

如上所述，目前我們知道在一個網路環境中，CP 可以發現裝置，對裝置的服務做出呼叫控制的動作，並且對事件通知做註冊的動作，相對的，Device 會對呼叫控制與事件通知註冊做回覆確認的動作，為達到此功能目的，UPnP 環境定義了下列各種狀態階段如下

- Addressing：Device 加入網域中，取得唯一的 IP 位址，使得其他的裝置可與之聯繫。
- Description：Device 會將本身所含有的服務項目與功能描述於 XML 語言中。
- Discovery：CP 將會搜尋到 Device 裝置並取得裝置描述文件。
- Control：Device 處理經由 CP 所發送出的 action 動作。
- Eventing：在狀態改變時，Device 通知已註冊的 CP。
- Presentation：device 可選擇提供一個 HTML 格式介面來接受控制與管理。

2.1.1.4 UPnP Protocol

UPnP 運用了許多現存的網路標準協定，這些協定廣泛存在於現有的網際網路與區域網路中，運用這些廣泛使用的通訊協定，省去了開發及遵守新協定的困難，其通訊協定架構如圖 1 示：

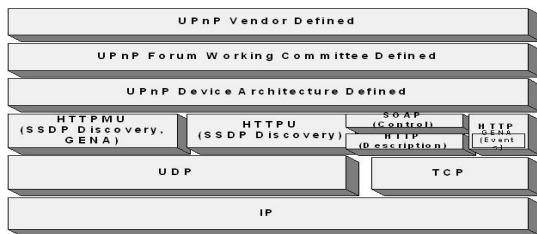


圖 1 UPnP Protocol Stack

資料來源：UPnP Forum

經由 UPnP 的協助，我們將可建構一個整合性的數位家庭網路如圖 2 示：



圖 2 UPnP 環境的數位家庭網路

2.1.2 Jini

2.1.2.1 Jini 的架構與技術

Jini[7]技術在 1999 年由 Sun Microsystems 公司所提出，主要運用 Java 技術所發展的 Service Discovery Protocol 建立出一開放性的分散式計算環境，並能夠隨時隨地簡單存取網路資源。Jini 技術主要的功能為裝置或設備不需安裝及設定，即具備隨插即用的連接功能，意指 Jini 主要在提供自發性的網路連結，允許用戶將幾乎任何支援 Java 的設備可直接連接到 Jini Community，並且網路的所有元件可以立即識別這個新的設備及其能力。

Jini 是一種分散式結構體系，主要由三個部分所組成：基礎結構(Infrastructure)、程式模型(Programming model)和服務(Service)。這三個部分雖有區別但又密不可分，基礎結構支援並利用程式模型，服務則內嵌於基礎結構中，服務物件包含並利用程式模型的介面進行服務之間的存取。

如圖 3 示，Jini 架構共分 Jini 服務提供者(Jini Service Provider)、Jini 客戶端(Jini Client)和 Jini 搜尋服務(Jini Lookup Service)三個部分：

- Jini Lookup Service：本部分是為 Jini 整體架構中的重要核心，主要提供 Jini Client 搜尋服務以及提供讓 Service Provider 註冊服務等功能。
- Jini Client：向 Lookup Service 搜尋所需的服務，而該服務是為 Service Provider 所提供。
- Jini Service Provider：提供 Client 所需的服務，並將該服務註冊至 Lookup Service。

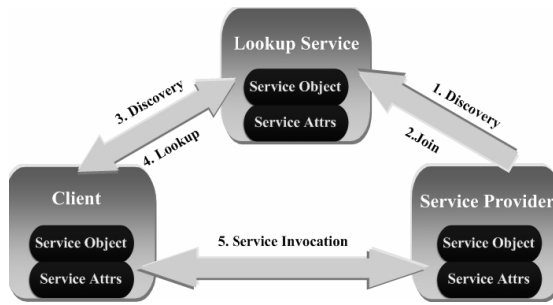


圖 3 Jini 執行架構

資料來源：Sun Microsystems

當 Jini Service Provider 提供服務給 Client 時，Jini Service Provider 會以廣播方式去 discovery Jini Lookup Service 的存在，並將自己的服務物件(包含 Java 介面，其中包括 Jini Client 和應用程式所要調用來執行服務的方法和服務屬性註冊到 Jini Lookup Service 中，該過程採用的是 Join 的動作。當 Jini Client 需要服務時，它會根據服務的類型或屬性向 Jini Lookup Service 查詢合適的服務，然後 Jini Lookup Service 會將查詢的結果回傳給 Jini Client。當 Jini Client 決定使用某一服務時，Jini Lookup Service 則將把所需服務的服務物件拷貝發送給 Jini Client，最後，Jini Client 將會通由該服務物件與 Jini Service Provider 進行直接聯繫。

2.2 控制網路相關傳輸技術之介紹

電力線上網(Power Line Communication, PLC)是指利用電力線傳輸資料和語音訊號的一種通信方式。使用 PLC 主要的原因是全球電力線多已建置完成，無須再建設新的網路。在每個家庭的環境中，幾乎都能夠找到電源插座，普及率更高達於乙太網路、電話線等。X-10 和 LonWorks 為目前公認的傳輸技術標準，本研究將針對這兩種技術做介紹與比較。

2.2.1 X-10

X-10[10]發展於1970年，由蘇格蘭的 Pico Electronics 公司開發出來的通訊協定，為全球最早發展 Power Line 的標準，主要是透過家中現有電源線(AC power line)，將控制命令傳送到被控制端上，並利用電源線較少干擾時段來傳送訊號，避免其他電器的訊號干擾。目前在美國家庭自動化裝置中(如燈控開關、保全系統、及電話答錄機等相關產品)佔約7成市場已經採用X-10規格產品或是介面。在AC 60Hz/120V的電力線中，X-10是以120Hz/5V的脈波訊號進行傳輸，封包長度為22個Bit，指令格式為(Header + Home Code + Unit/Function Code)，X-10的封包標題固定為1110，其餘18個Bit以互補的方式編碼(如:1101 編碼後變10 10 01 10)，編碼後由AC 120V的零相位角開始傳送，且每筆傳送三次。

至於X-10是如何運作，以最簡單的方式來說，整個系統包含4個部分，整理如下：

- 傳送器(Transceiver)：如桌上控制盒及掛在牆上的模組等，主要是用來對燈具或家電產品傳送控制訊號。
- 燈具及家電上的模組：主要是用來接收控制訊號，並將相對應之燈具及家電產品開啟或關閉(ON/OFF)。
- 可移動式的輸入裝置：類似像遙控器一樣的裝置，透過如紅外線(IR)的方式來遙控(將控制命令送至傳送器上)。
- 交流電線：主要是將訊號從傳送器透過電線送至燈具或是家電處。

2.2.2 LonWorks

LonWorks[4]發展於1990年，由美國Echelon公司所制定之「家庭自動化」控制規格，依循LonTalk通訊協定所發展的全分散式智慧控制網路技術。LonWorks是為一種點對點(peer to peer)的網路控制技術，主要由四個部分所組成：LonWorks通訊協定、Neuron Chip、控制及連接裝置、LonWorks網路服務(LonWorks Network Services, LNS)。LonWorks技術是唯一包含OSI Model中全部七層，該通訊協定是獨立運作的(media-independent)，能夠在電源線、雙絞線、無線(RF)、同軸線、紅外線、光纖等多種介質上傳輸。

完整的LonWorks平台包括LonTalk協定互通標準(interoperability standards)與軟體應用程式介面(API)。LonWorks網路服務中具有支援IP功能，因此能夠讓Lonworks網路與Internet相當容易連結，當然也方便進行遠端監控動作，透過電源線，LonWorks傳輸速率約為5.4kbps。另外，LonWorks通訊協定及LonWorks技術在電力線及雙絞線上的傳輸規格，可區分為ANSI/EIA 709.1、709.2及709.3等標準。

就技術而言，為了避免發生資料碰撞等問題，LonWorks使用一獨特之媒體存取控制(media access control, MAC)演算法來決定資料封包何時要傳送，以提高品質穩定度，基本上是以隨機方式將某一特定裝置連接上傳送媒介。除此之外，也使用位址演算法來決定資料封包如何從起點傳送到終點。每個封包內包含了起點位址與終點位址，可以是實體位址(physical address)、裝置位址(device address)、群組位址(group address)或廣播位址(broadcast address)。

3. 使用者介面產生

3.1 UIML

UIML[8]是一種基於 XML 格式的描述語言，可用來定義使用者介面。當初定義 UIML 時主要希望能減少在多種裝置上呈現使用者介面的開發時間，因此設計 UIML 的最主要目的即在於提供一個標準的使用者介面描述語言來供給各類型的裝置使用，圖 4 為一個 UIML 的格式範例。

```
<?xml version="1.0" ?>
<!DOCTYPE uiml PUBLIC "-//Harmonia//DTD UIML 2.0 Draft//EN"
"UIML2_0g.dtd">
<uiml>
  <head>...</head>
  <interface>
    <structure>...</structure>
    <content>...</content>
    <behavior>...</behavior>
    <style>...</style>
  </interface>
  <peers>...</peers>
  <template>...</template>
</uiml>
```

圖 4 UIML 格式

資料來源：www.uiml.org

因為 UIML 採用 XML 格式，因此可將不同使用者介面元件以不同的標籤來區分，此 UIML 內容不包含任何平台的定義說明，舉例來說明，此元件中並不包含<windows>如此的標籤，對於可適用的平台，設計人員只需要能在其工作平台上解譯處理 UIML 文件，便可以取得相關聯的使用者介面資訊。

3.2 CC/PP

CC/PP(Composite Capability and Preference Profile)[2]主要提供了 Web 應用程式作到可描述 Capability 與 Preference 的能力，CC/PP 可用來描述任何可存取 Web 服務的 User Agent，而此 User Agent 將包含了有硬體平台、軟體平台及應用程式的組合，針對這些 User Agent 的 Capability 與 Preference 描述內容是必要的，但是因為某些特定的手持裝置可能在網路傳輸率與資料容量大小方面，都會受到相當程度的限制，因此為了解決此問題，CC/PP 的存在便可以較有效率的方式來提供所必須要的文件資料。

RDF(Resource Description Framework)是由 W3C 提出的一個 XML-style 的語言，而 CC/PP 便採用了 RDF 的格式，資料的組成可採用檔案的預設內容，或是加入新增要修改的部分內容，便可取代原本所預設的內容。

在 CC/PP 中的編碼方式主要區分為兩種，分別為"indirect"與"inline"，在 inline 模式中，client 裝置將會傳送整份描述 Capability 與 Preference 的 RDF 敘述檔案，如圖 5 所示：

```
Profile:
Profile-Diff: <?xml version="1.0"?>
<RDF xmlns="http://www.w3.org/TR/1999/PR-rdf-syntax-19990105#"
xmlns:PRF="http://www.w3.org/TR/WD-profile-vocabulary#">
  <Bag>
    <Description about="HardwarePlatform">
      <Defaults>
        <Description PRF:Vendor="Nokia"
          PRF:Model="2160"
          PRF:Type="PDA"
          PRF:ScreenSize="800x600x24"
          PRF:CPU="PPC"
          PRF:Memory="16mB"
          PRF:Bluetooth="YES" />
      </Defaults>
    </Description>
    <Modifications>
      <Description PRF:Memory="32mB" />
    </Modifications>
  </Bag>
  <Description ID="SoftwarePlatform" PRF:Sound="On"/>
</RDF>
```

圖 5 Encoding in inline mode

資料來源：www.w3.org

而在 indirect 模式當中，client 裝置則只會將需要修改的內容放在 Profile-Diff 當中，其餘的 Capability 與 Preference 內容將放在 URL 參考位址，而非將完整的 RDF 描述內容都放進來，如圖 6 所示：

```
Profile:"http://www.profile1.com", "http://www.profile2.com"
Profile-Diff:<?xml version="1.0"?>
<RDF xmlns="http://www.w3.org/TR/PR-rdf-syntax-19990105#"
xmlns:PRF="http://www.w3.org/TR/WD-Profile-vocabulary">
  <Description ID="SoftwarePlatform" PRF:Sound="On"/>
</RDF>
```

圖 6 Encoding in indirect mode

資料來源：www.w3.org

4. 智慧型萬用遙控器控制技術

4.1 架構說明

未來是一個數位家電充斥的環境，會搭配控制器以方便使用者使用，但如果每個家電都有各自的控制器，如此的情況將會顯的相當的複雜且缺乏便利性，而萬用控制器的想法也因此而生，一機在手，就能控制所有的東西，享受真正數位家庭的好處，而不是讓大量的未來家電充斥又無法簡單易懂的使用它們。此為本篇論文的目標。

圖 7 為我們所建構的數位家庭系統的概觀圖，藉著使用者操作 URC (Universal Remote Console) [1][3][5][6]裝置，如手機、個人數位助理 (PDA) 或電腦，便可利用文字，圖示等介面，透過有線或無線網路來操作數位家電，如電燈、電視、咖啡機、電梯或其他的服務如網路銀行、醫療照護...等。

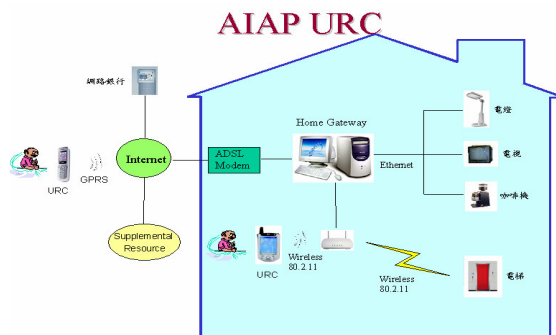


圖 7 URC 數位家庭示意圖

智慧型萬用遙控系統使用 AIAP-URC 此一標準，此一標準為 INCITS/V2 協會於 2005 年 2 月提出的，目前已進行到最新的 draft2 版本，為達到 AIAP-URC 的功能目的，我們的系統架構圖如下，主要依據 INCITS/V2, The Information Technology Access Interfaces Technical Committee of the InterNational Committee for Information Technology Standards 所提出的 INCITS 389、390、391、392、393 等文件。系統架構分為三個主要的元件和兩個網路。如圖 8 所示：

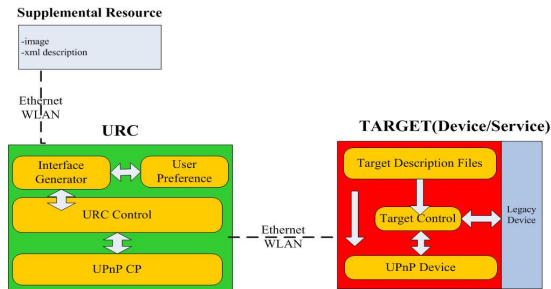


圖 8 智慧型萬用遙控系統架構圖

1. URC(Universal Remote Console)為主要的控制器端，用來作為接收裝置訊息及發送控制命令的終端機，裝置可為 PC、Laptop、PDA、mobile。內部包含有 UPnP CP、URC Control、Interface Generator、User Preference 等模組。
2. Target 為主要被控制的裝置端，可主動廣播其裝置資訊並進入等待被控制狀態，設備可為電燈、電視、音響、咖啡機...等等。Legacy Device 為表示傳統的非 UPnP or AIAP-URC 裝置。Target 為達成 AIAP-URC 功能包含 UPnP Device、Target Control 模組與 Target Description。
3. Supplemental Resource 提供了外部資源參考，可採用 Web Server 技術來存取。
4. 兩個網路介面主要以 Ethernet 和 WLAN 作為溝通之網路介面。

4.2 設計內容

URC 標準定義了 XML 描述語言來定義各項裝置控制命令與使用者介面內容，此一資料將會由 Target 端傳送到 URC 端，接收並處理資料後，將可得到 Target 的各項控制資訊與顯示資訊，藉由 URC 端將使用者介面資訊解譯處理後，建構出對應的使用者介面，再串接關聯的控制命令動作，便可以在 URC 端取得完整的 Target 控制權。

URC 內定義了各類的抽象使用者介面元件，稱之為 abstract interactors，代表了各類型輸出的使用者介面元件，每一個 abstract interactor 將對應到 URC 實際存在平台上的 concrete interactor，在 URC 端接收到 Target 的 XML 資訊後，經由解譯的過程再產生相對應的使用者介面，而 URC 的最主要設

計目的即在於針對各類型的 Target 裝置，可以在同一 URC 控制端上來同時做存取控制的目的。

URC 定義了四個重要的 XML 標準描述語言檔案來描述其 Target 的本質，以下將分別介紹其功能：

1. Target Description：可用來記錄在 Target 端上可接受的 Socket 控制動作，Target 會在裝置搜尋的階段中提供給控制端。
2. User Interface Socket：以抽象化的概念來描述 Target 端上可運作的功能，在 Socket 內容中我們可定義 statics、variables、commands、notifications 等不同種類的 socket。
3. Presentation Template：此即代表之前所提之 User Interface Implementation Descriptions，可作為產生使用者介面的參考資訊，內容描述了每個 socket 相對應的使用者介面元件。
4. Resource Sheet：內容包含了 Resource Description，可紀錄各項 resource 的種類、文字內容、角色。

4.3 重要程式模組說明

在 UPnP Device 模組中，主要工作為處理 URC 與 Target 之間的溝通協定，為了達到我們計畫目標，我們採用 UPnP Device 的架構為核心來建構，其中 Device 為此模組為最重要的 Class，其中內含 DeviceData Class，此 Class 包含三種 Threads，描述如表 2：

表 2 DeviceData class

Thread 名稱	Thread 功能
HTTPServer Thread	處理 HTTP Packet 的接收
SSDPSearchSocket Thread	處理 Control Point 詢問網路上有哪些 UPnP Device 的 SSDPPacket
Advertiser Thread	定時於發送 SSDP Announce，主動通知 Control Point 此 Device 存於網路上。

相對應於 Target 端的 UPnP Device，我們在 URC 端亦將採用 UPnP Control Point 的架構核心來做為與 Target 端 UPnP Device 相通的通訊協定橋樑，其中幾項重要的 class 說明如表 3：

表 3 URC 端重要的 class

Class 名稱	Class 功能
CtrlPoint	依據 UPnP Control Point 所建立的主要 Thread，處理 UPnP 的各項動作
CtrlPointPane	URC 將呈現的程式主畫面
LightPane	依據各項裝置所呈現的控制畫面
UIElement	儲存畫面元件資料
SResource	Supplemental Resource 的存取
UserPref	可作為使用者喜好的設定與讀取

4.4 prototype 實現

為了達到 URC 所規範的標準，目前我們正在進行 URC prototype 的實現工作，以驗證 URC 的可行性。

4.4.1 開發環境

在本計劃的開發環境中，技術層面上的構想可主要分為下列幾點：

- 作業系統：因考量使用者的使用環境與熟悉性，本計劃初期將採用 Windows 作業平台，並以 Java 作為系統發展的環境。
- 程式語言：取決 Java 具有物件導向、簡單和 Unicode 等特性，加上由於 Java 在不同平台上的移植較為方便，因此本研究採用 Java 作為主要的程式開發語言。
- 硬體平台：初期將使用 x86 系統的硬體平台，採用的主要目的在於容易建構開發環境並擁有較充足的資源空間，減少開發的時程，後期將再深入 PDA 與 mobile 的平台。

開發工具將採用 JBuilder 2005，並採用 Cyberlink UPnP 作為底層的通訊協定。為了驗證 URC 的可行性，我們實作了一個 URC 的 prototype，如圖 9 所示，左邊的應用程式為 URC 的控制畫面，URC 可針對不同的 Target 產生不同的控制畫面，而右邊即存在兩個 Target 裝置分別為模擬的電燈與時鐘裝置，使用者可透過 URC 來同時控制此兩項家電裝置。

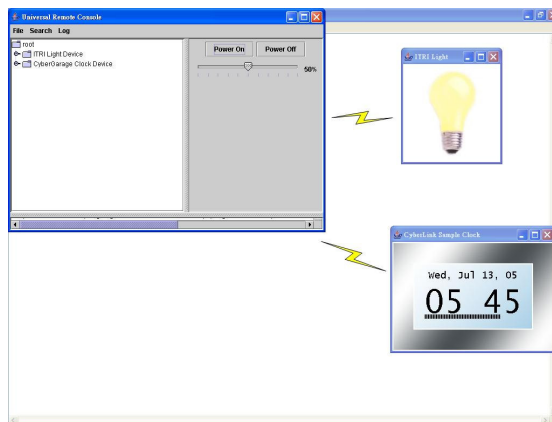


圖 9 URC 控制畫面

5. 結論

5.1 結論

URC 的標準提供了使用者個人環境的 URC 萬用遙控器介面，可以一機在手，操控各類型的數位家電裝置，為未來的數位家庭帶來更多的便利性。

5.2 未來工作

以遙控器的角色而言，URC 將不可避免的存於短小輕便的裝置上，並且極有可能以隨身攜帶或是伸手可及的範圍存在於使用者的環境當中，如可搭配應用資料探勘的技術，藉由分析各類資料庫數據，提供更多因人因地的服務，將可更加提升 URC 的附加價值，未來本計劃將亦會朝此方向來努力。

參考文獻

- [1] Bill LaPlant, Shari Trewin, Gottfried Zimmermann, and Gregg Vanderheiden. The Universal Remote Console: A Universal Access Bus for Pervasive Computing
- [2] CC/PP, <http://www.w3.org/Mobile/CCPP>
- [3] Gottfried Zimmermann, Gregg Vanderheiden, Al Gilman. Prototype Implementations for a Universal Remote Console Specification
- [4] LonWorks, <http://www.lonworks.com>
- [5] NCITS V2 – Standards Development Committee on Information Technology Access Interfaces. http://www.ncits.org/tc_home/v2htm/V2docs/v201011.htm
- [6] Nichols, J., & Myers, B.A. (2001). Studying The Use of Handhelds To Control Everyday Appliances. http://www-2.cs.cmu.edu/~pebbles/papers/pebbles_puc.pdf
- [7] Sun Microsystems, Jini Network Technology, <http://www.sun.com/jini>
- [8] UIML, <http://uiml.org>
- [9] UPnP, <http://www.upnp.org>
- [10] X-10, <http://www.x10ideas.com>