

行政院國家科學委員會專題研究計畫期中報告

eHome：電子家庭雛型之設計與實作 — 子計畫二 嵌入式 Java 及 Jini 系統之設計與實作 (1/3)

計畫類別：A個別型計畫 整合型計畫

計畫編號：NSC 89-2218-E-002-074

執行期間：89年8月1日至90年7月31日

計畫主持人：陳俊良

計畫參與人員：徐國偉、楊禎泓、
劉家瑋、吳介元、劉品宜

本成果報告包括以下應繳交之附件：

A赴國外出差或研習心得報告一份

A赴大陸地區出差或研習心得報告一份

A出席國際學術會議心得報告及發表之論文各一份

A國際合作研究計畫國外研究報告書一份

執行單位：國立台灣大學資訊工程學系

中華民國九十年五月二十五日

行政院國家科學委員會專題研究計畫期中報告

eHome：電子家庭雛型之設計與實作—子計畫二

嵌入式 Java 及 Jini 系統之設計與實作 (1/3)

Design and Implementation of Embedded Java and Jini System

計畫編號：NSC 89-2218-E-002-074

執行期限：89 年 8 月 1 日至 90 年 7 月 31 日

主持人：陳俊良 國立台灣大學資訊工程學系

email: clchen@csie.ntu.edu.tw

計畫參與人員：徐國偉、楊禎泓、劉家瑋、吳介元、劉品宜

一、中文摘要

本計畫是屬於國科會 3C 整合科技群體研究計畫「eHome：電子家庭雛型之設計與實作」的一個子計畫。eHome 整合型計畫的總目標在於探討建構一個電子家庭必要的核心技術。我們預期一個電子家庭將有很多的家電，而這些家電是由嵌入式系統來控制的。這些嵌入式系統的關鍵技術至少包含作業系統、Java、WAP 以及智慧型代理人程式。本子計畫負責的是 Java 與 Jini 部分。

這是三年期計畫的第一年期中報告。

關鍵詞：Java 系統、Jini 系統、嵌入式系統、智慧型家電。

Abstract

This is a sub-project of "Design and Implementation of eHome Prototype." The goal of eHome project is to study the key technologies when building an electronic home. We expect an eHome consists of many intelligent appliances, which are controlled by embedded systems. The required key technologies of those embedded systems at least include OS, Java, WAP and agent. This sub-project focuses upon Java and Jini.

This is the report of the first year.

Keywords: Java system, Jini system, embedded system, intelligent appliances.

二、緣由與目的

eHome 計畫

本計畫是 3C 整合科技研究案「eHome：電子家庭雛型之設計與實作」之子計畫二。eHome 計畫的目的是想要探討建構一個電子家庭必要的核心技術。為了有明確的努力方向，我們勾勒的實際應用敘述如下：我們可以在離家十萬八千里遠的地方，用一個具有 WAP (Wireless Application Protocol) 能力的無線手機，利用上頭的 WML 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(可能是一個 Set-top Box)。eHome Center 對外有個 WWW 伺服器提供 WML 網頁給手機，讓我們在網頁上下命令傳回家。eHome Center 本身有夠強的 agent 能力，可以協助命令的執行。eHome Center 對內則用無線的方式與具有 Java 與 Jini 能力的家電連結。最後，這些家電聽懂了而且正確執行了我們的命令。

eHome 計畫的重點在具有 WAP 能力的手機、具有 agent 能力且可對內對外聯絡的 eHome Center、以及具有 Java 與 Jini 能力的各式 IA。整個 eHome 計畫包括了 OS, Java, WAP 以及 agent 四大部分。本子計畫所涵蓋的是 Java 部分。

Java 與 Jini 在 eHome 計畫的角色

我們打算把 Java [1-3] 與 Jini [4] 技術用在底下這些地方：

首先，我們可以讓 WAP enabling 的手機也可以是 Java enabling。WAP 手機需要附加很多的軟體，比如說 WML 瀏覽器。

假如說這些軟體可以用 Java 來發展，這些軟體將可以與平台切開來，達到跨平台的效果。要有如此效果，我們需要一個可以在手機平台上執行的 Java Virtual Machine (JVM)，以及與 WAP 有關的 Java packages。

其次，我們打算讓各式家電都內含一個嵌入式 Java 系統。跟前頭一樣，立即可以看到的好處是跨平台；在面對多樣性的家電時，這個優勢是十分值得去追求的。同理，我們要的是一個可以用在家電平台上的 JVM，以及與家電控制有關的 Java packages。

更重要的是，我們希望可以讓 Java enabling 的家電更進一步可以是 Jini enabling；以達到 universal plug and play 的效果，真正成為「智慧型家電」。要達到此一目標，除了 eHome Center 也必須要引用 Java 技術外，更重要的關鍵是如何把 Jini 技術引用到功能有限的嵌入式系統。

三、結果與討論

本子計劃的任務是提供 eHome 整體環境所需的 Java 系統，包含 eHome Center 手機以及家電。底下先敘述現有的 Java 系統，再論及我們的看法、決定以及已經完成的工作項目與成果。

現有 Java 系統

目前 Sun 公司所提供的 Java 平台由大到小有很多種。所謂的 Java 平台包含了 Java 最核心的軟體，也就是 Java 虛擬機器，和搭配的 API。

J2EE (Java 2 Enterprise Edition) [5] 是供大型伺服器用的，配置的虛擬機器引用了 HotSpot 的技術。J2SE (Java 2 Standard Edition) [6] 的適用對象是一般的個人電腦或筆記型電腦，配置的虛擬機器就是使用最廣的 JVM。這兩級產品所需的記憶體空間是由數十 Mega 起跳的，顯然不太適合 eHome 計劃內的嵌入式系統使用，即使對 eHome Center 想要建構的 Set-top Box 也可能都嫌大。

J2ME (Java 2 Micro Edition) 才是供嵌入式系統用的。J2ME 目前有兩個版本。CDC (Connected Device Configuration)

[7] 設想的應用範圍為較高階的嵌入式系統，比如影像電話、Set-top Box、較高價位的個人數位助理 (PDA) 等。配置的虛擬機器叫 CVM，原則上是在 32-bit 微處理器上執行，所需的記憶體空間大約個位數的 Mega。推出的時間是 2001 年初。

CLDC (Connected, Limited Device Configuration) [8] 名字多了 limited，意思是所需的硬體平台比 CDC 有更多的限制。設想的應用範圍為較低價位的 PDA、手機等。原則上這些裝置是由電池供電的，在 16 或 32-bit 微處理器上執行，所需的記憶體空間小於 512K。它的虛擬機器叫做 KVM，命名的理由是所要的記憶體只需以 Kilo 為單位。

比 J2ME 更小的就是 Java Card [9] 了。這是用在智慧卡 (smart card) 或是手機 SIM 卡上的 Java 系統。最好記憶體可以小到 32K 以下，64K 大概就是最上限了。利用的是 8 或 16-bit 微處理器。它的虛擬機器叫做 JCVM。

手機和家電的 Java 系統

本子計畫第一年的重點在於提供手機和家電所需的 Java 系統。手機和家電一般都歸類於嵌入式系統。一般而言嵌入式系統都會有一些限制，記憶體不夠大，CPU 不夠快等等。以 Sun 公司定義的 Java 系統而言，很明顯的，J2ME-CLDC 和 Java Card 是可能的選擇。

嵌入式系統硬體的限制將會使得它的軟體被綁手綁腳。我們必須要有能力根據特殊的應用對軟體系統做特殊的微調 (fine tune)。當然，前提是我們必須要非常了解軟體系統。

軟體系統的取得不外乎拿別人的來移植 (porting) 或是自製。我們評估目前的大環境以及自己 (指台大資訊系執行此計畫的團隊也指台灣相關的業界) 的能力，我們主張，對比較複雜的軟體移植加上改寫是比較可行的方式，比如 J2ME-CLDC 的 KVM。做這個決定事實上是有一點無奈，我們自忖我們還搶不到世界龍頭的角色。在目前科技無國界的環境下，你不是龍頭就只好跟著龍頭走。我們對自己的期許是要跟的夠快夠好。所謂的夠快夠好指的是，我們可以在取得別人新版本的軟體

後，在最短時間內移植甚至改寫的更適合我們的特殊需求。要達到此目的，基本的紮根工作不可免，比如詳細剖析相關軟體，即使是稍舊版本也無所謂。

在執行本計畫之前幾年，我們已經累積了不少 JVM 的經驗，以這些經驗為本，這個年度我們剖析了 KVM。甚至把這些成果轉移到教學上，希望讓更多沒有參與本計畫的同學也可以對 KVM 有基本的認識。以累積的經驗與人力，在往後計畫期間，我們將能勝任移植與改寫 KVM 到目標平台的工作。預計這是第二年的工作。

至於對比較小的軟體，我們主張應該要擁有獨立開發的能力，比如 JCVM。理由除了規模小的軟體自製門檻較低外，另一理由是：小軟體是對小系統，小系統可能必須斤斤計較記憶體的 byte 或是 CPU 的 clock，或者說經常會非常的平台相依。導致的結果是必須極端的微調。既然如此，不如擁有自製的能力與產品。

這個年度，我們實作了一個比標準 JCVM 還要輕量級的 JCVM。對手機或是較高檔的家電而言，這個 JCVM 或許太簡單了。但是對較低階的家電，比如電燈、烤麵包機，應該夠用了。

STK 程式的撰寫

STK (SIM Tool Kit) 是手機 SIM 卡的一套標準介面。原本的 SIM (Subscriber Identification Module) 卡只是一片晶片，它的功能，如同全名，只是用來識別手機門號的。隨著無線通訊技術的演進，越來越多的功能被加到這一片晶片上。比如，資料儲存（電話簿）、語音郵件、短訊、預付卡功能。短期間內更希望可以被賦予電子商務或是網際網路應用的功能。在添加這麼多功能的情況下，為了可以讓任意廠牌的手機都可以搭配任意卡廠出品的 SIM 卡，有必要訂定一個標準。STK 就是這樣的一套標準介面。雖然說，在標準裡定沒有規定介面下的 STK 要如何實作，但是以目前的局面來看，Java Card 很有可能是最後的贏家。

STK 程式的撰寫，事實上並沒有規劃入原計畫書中。不過基於底下兩個理由我們有必要執行這一部份的工作：

首先，我們意識到 Java Card 的程式與

一般 Java 程式，事實上是有所不同的。為了確切掌握 Java Card 程式的特性，以便開發 JCVM 時可以更掌握住關鍵點，我們有必要學習 Java Card 程式的撰寫。STK 程式就是以 Java Card 程式為本的。

另外，這一年來可以看到 STK 有點可以和 WAP 平起平坐的趨勢。eHome 計畫的手機通訊方面，原本規劃的是 WAP，WAP 子計畫 [10] 目前也順利進行中。但是，讓 eHome 計畫多提供一種手機通訊的選擇是可以讓整體計畫的成果加分的。

在這個年度裡，我們在 Bull 公司 STK 模擬器上學習 STK 程式的撰寫。程式的功能是：使用者可以把控制家電的 Java Card 程式下載到手機，利用上頭的 JCVM 執行該程式以打包要控制家電的信號，最後再送回家裡的伺服器完成家電的控制。

所謂家裡的伺服器目前只是用一個獨立的 PC 來模擬，應該儘早與 Agent 子計畫 [11] 的伺服器相結合。另外，脫離 STK 模擬器的階段而把程式灌入真正的 SIM 卡進而使用到真正的手機，也是很實際很值得去做的事。往後兩年，我們會在這上面多下工夫。

eHome Center 的 Java 系統

在規劃中，這是第二年的工作項目，不過我們已做了一些先期的準備工作。eHome Center 應該是一個 Set-top Box。對外，它是一個 WWW 或是 WML 伺服器，也需負擔 WAP Gateway 的角色。對內，它是家電的總管，利用無線的方式與具有 Java 與 Jini 能力的家電連結。它還要有夠強的 agent 能力以協助控制家電。

在計畫進行之初（2000 年下半年），對 eHome Center 所需 Java 系統的選擇，有點陷入不上不下的尷尬情境。J2SE 的 JVM 能力絕對足以負擔 eHome Center 之所需，但是數十 Mega 的記憶體需求對一個 Set-top Box 而言卻是沉重的負擔。J2ME-CLDC 的 KVM 小是夠小，但是它沒有 reflection，也沒有 serializable 的能力，意味著它沒有 RMI (Remote Method Invocation)，更進一步也將沒有 Jini 的能力。另外，網路的能力也不足。我們已做了在第二年把 JVM 瘦身到符合 eHome Center 需求的準備。

今年(2001年)初, J2ME-CDC 正式發表。原先 KVM 所沒有而預期 eHome Center 該有的, CVM 都補回來了。CVM 應該就是 eHome Center 所要的。跟 KVM 一樣, 這個年度我們也剖析了 CVM。

網路上有人說:「CVM is JVM」。這句話的意思當你剖析過 JVM 以及 CVM 你才會充分體會。CVM 的程式碼很明顯可以看出來是用 JVM 改過來的, 那麼它們的差異在哪裡? 在 CVM 程式碼裡, 你可以發現到有非常多的 compiler directive。當移植到不同硬體平台時, 藉著控制編譯時的參數, 可以產生出不同需求的 CVM。

OS 子計畫 [12] 已擇定採用 Strong Arm 加上 Linux 為 eHome Center 的平台。另一方面, 目前的 J2ME-CDC 參考版本就是在 Linux 上面發展的。好像在這邊我們的工作將變得非常簡單。事實上, 也不然。

在往後的工作中, 我們打算以我們剖析過 CVM 的經驗為本, 先嘗試活用編譯時的參數, 希望可以讓移植的工作做得非常靈活。硬體平台一改變, 我們可以在最短時間把最適合的 CVM 製造出來。我們希望也可以累積改寫 CVM 的能力, 以防萬一將來 eHome Center 平台的限制更嚴苛時(比如, 考慮到成本), 我們可以進一步對 CVM 做瘦身工作。

Jini 與家用影音網路

原本計畫的第三年, 本子計畫打算讓家電擁有 Jini 的能力, 讓他們更有智慧。本年度我們也做的一些先期的準備工作。

Jini 系統這一年來稍稍沉寂, 與它所需的軟硬體規格過高不易整合入現有嵌入式系統有關。倒是與它相類似由 Sony 等家電大廠主導的 HAVi (Home Audio/Video Interoperability) [13] 值得注意。HAVi 底下的硬體用到了 IEEE 1394, 軟體用到了 Java。1394 提供了高速傳輸影音資料的可行性, Java 讓它容易跨平台。

原計畫書中我們設想的家用網路可以由 eHome Center 來「控制」各個家電。這一年來, 我們又認知到, 在家用網路走的不止於控制信號, 應該還會有「資料」。所謂的資料主要是影音資料, 它們的特點是流量很大。一個理想的 eHome 應該也把影音資料流考慮進去。HAVi 是一個可以

選擇的標準, 而且跟 Java 有關, 所以適當引用 HAVi 到計畫裡是可能的思考方向。

四、計畫成果自評

本文完成於計畫第一年結束前兩個月, 不過第一年預計該完成的項目大致均已完成, 某些部分還稍有超前。

在第一年計畫執行期間, 主客觀環境均有所變異, 我們也不斷思考剩下兩年該如何調整才能使計畫更成功。除了原定工作項目外, 預計我們也會投注心力在 STK 以及 HAVi 的相關研究上。

五、參考文獻

- [1] Sun Micro., *Java*, <http://java.sun.com>
- [2] J. Gosling, B. Joy, G. Steele and G. Bracha, *The Java Language Specification*, 2nd Ed., Addison-Wesley, 2000.
- [3] T. Lindholm and F. Yellin, *The Java Virtual Machine Specification*, 2nd Ed., Addison-Wesley, 1999.
- [4] Sun Micro., *Jini Connection Technology*, <http://www.sun.com/jini>
- [5] Sun Micro., *Java 2 Platform, Enterprise Edition*, <http://java.sun.com/j2ee>
- [6] Sun Micro., *Java 2 Platform, Standard Edition*, <http://java.sun.com/j2se>
- [7] Sun Micro., *Java 2 Platform, Micro Edition, Connected Device Configuration*, <http://java.sun.com/j2me/cdc>
- [8] Sun Micro., *Java 2 Platform, Micro Edition, Connected Limited Device Configuration*, <http://java.sun.com/j2me/clcd>
- [9] Sun Micro., *Java Card Technology*, <http://java.sun.com/products/javacard>
- [10] 許清琦等, *無線通訊協定 (WAP) 系統之設計與實作第一年期中報告*, 台大資訊系, 90 年 5 月。
- [11] 鄭士康等, *家電網路之微伺服器及智慧型代理人軟體之設計與實作第一年期中報告*, 台大電機系, 90 年 5 月。
- [12] 王勝德等, *嵌入式即時作業系統核心程式之設計與實作第一年期中報告*, 台大電機系, 90 年 5 月。
- [13] HAVi Organization, *HAVi*, <http://www.havi.org>