

近場通訊之萬用服務框架

張玉山(Yue-Shan Chang)

台北大學通訊工程學系

台北縣三峽鎮大學路 151 號

ysc@mail.ntpu.edu.tw

陳宏志(Honh-Jih Chen)

中央大學通訊工程學系

桃園縣中壢市五權里 2 鄰中大路 300 號

955203022@cc.ncu.edu.tw

摘要

近場通訊技術(Near Field Communication, NFC)，一種以 RFID 為基礎的短距離無線通訊技術，使用者在操作時只要將擁有 NFC 功能的裝置互相靠近感應，再視情況加上使用者確認，即可完成服務。其能涵蓋的應用範圍相當廣泛，舉凡小額付費、P2P 傳送、驗證服務等皆可為其應用。但相對的當 NFC 裝置收到 NFC 應用服務訊息時該如何處理，將成為一個問題。故規範一個良好的 NFC 應用服務互動流程、建置一個泛用的應用服務平台，便成為 NFC 目前的重要議題之一。於是本論文中，我們首先訂定共通的 NFC 應用服務訊息格式，接下來規範應用服務訊息的溝通流程，最後依據此溝通流程與訊息格式建立行動裝置軟體框架。

關鍵字：近場通訊(NFC)、射頻識別(RFID)、非接觸智慧卡(Contactless Smart Card)、行動商務、驗證識別系統、點對點傳輸

一、簡介

近場通訊(Near Field Communication, NFC)技術是一種短距離無線通訊技術，通訊距離約 10~20 公分，常見之傳輸速率為 106、212 及 424kbps。NFC 技術以射頻識別技術(Radio Frequency Identification, RFID)技術為基礎，結合現存的互連技術演變而來，並且相容於非接觸智慧卡技術(Contactless Smart Card)。自國際標準訂定以來，已被世界多個國家接受，並進行

試驗推廣或實際應用。

NFC 行動裝置的操作非常簡單，如同 RFID 一樣，啟動 NFC 功能後，將 NFC 裝置互相靠近接觸，即可建立連線且自動完成應用服務。行動裝置結合 NFC 技術能讓應用服務的操作與互動更加的便捷，更能加速應用服務的推廣以及提高其接受度。目前已有相當多 NFC 應用服務，如非接觸式付費、門禁系統、智慧型看板等，皆為其現今或未來之可行應用。

然而 NFC 技術在標準訂定的時候，對於應用服務的部分卻未有完善的規範。如七[2]中，NFC 目前尚無一個整合的互動模式框架。在七[3]、[4]中，分別提出 Virtual Tag 及 Profiles 的概念，希望能定義自己的 NFC 應用服務訊息包裝。至此我們可知目前 NFC 在應用服務的互動與訊息包裝方面仍未完善。如此，NFC 的用戶們便很可能因沒有一個共通的訊息格式，而無法使用應用服務。

於是在本論文中，將會評估近場通訊之應用服務，訂定一個通用的 NFC 應用訊息格式，並且規範 NFC 應用服務溝通、解析之流程，最後建立整合的 NFC 應用服務軟體框架。如此方能讓 NFC 之應用服務在操作時符合當初設計之理想。

接下來，會依序從 NFC 的技術背景開始介紹，而後帶到本論文的主體，規範通用的訊息格式與訊息解析流程，將 UNFCS 框架建構出來，最後實際的部署在軟、硬體的環境之上。文末，會帶上結論以及未來研究方向的探討。

二、技術背景

2.1 Radio Frequency Identification

RFID 技術的基本運作原理是將欲傳送的訊號載在電磁波上，而接收端在接收訊號的時候可以識別訊號源及讀取訊號中資訊。目前訂定射頻識別技術標準的組織主要有兩個，ISO 及 Auto-ID Centre(由 EPC Global 管理)。

一般的 RFID 系統中，通常包含以下三個部份，射頻識別標籤 (RFID Tag)、射頻識別讀寫器 (RFID reader) 以及訊息服務系統 (Data processing subsystem)。而工作頻率大致上可以分成三個區域，低頻(125 ~ 134.2 kHz 或 140 ~ 148.5 kHz)、高頻(13.56 MHz)及超高頻(868 ~ 928 MHz)。

由於其非接觸式、唯一識別及被動標籤無須內建電源等特性，使得射頻識別技術其可應用的領域十分廣泛，舉凡物流管理、電子收費系統、通行證等，皆為其常見的應用服務。

2.2 Near Field Communication

近場通訊(Near Field Communication, NFC)是一種新的無線連接技術，由 Philips、Sony 及 Nokia 所共同開發的 X，並於 2004 年成立近場通訊論壇(NFC Forum)，負責制定近場通訊的協定及技術推廣等事務 X。目前 NFC 主要的技術標準，如 NFCIP-1(IEC 340、ISO/IEC 18092)、NFCIP-2(IEC 352、ISO/IEC 21481)、NFC-WI(IEC 373、ISO/IEC 28361)等，皆已通過國際標準組織 ISO/IEC 及 IEC 的認證[1]。

最主要的標準為 ISO 18092 (NFCIP-1)，其中制定 NFC 技術的介面與協定等相關細節。NFC 裝置工作的載波射頻頻率域 (RF field) 為 13.56 MHz。訊息傳送的位元編碼使用曼徹斯特碼。通訊模式分為被動模式(Passive Mode)與主動模式(Active Mode)兩種。傳送的位元速率(bit rate)由通訊模式來劃分，在被動模式下可為

106~424 Kbps，在主動模式下可為 106~6780 Kbps。雖然位元速率最高可達 6780 Kbps，但目前市面上主要仍為 106、212、424Kbps 這三種。

NFC 裝置分成發起者與目標設備，而主、被動模式主要的差別也在於，目標設備在主動模式下會以自身的電力產生 RF 場並依照發起者選定的速率回應(如圖 1)，而在被動模式下則是由感應發起者的 RF 場產生能量後，回應發起者的命令(如圖 2)。



圖 1、NFC 通訊—主動模式

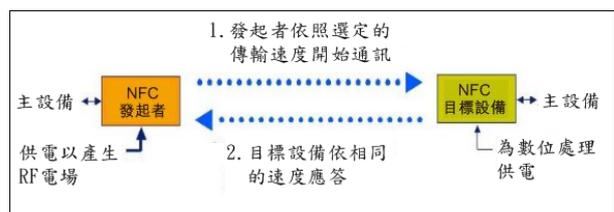


圖 2、NFC 通訊—被動模式

2.3 NFC Data Exchange Format

當資料訊息要使用 NFC 技術傳送時，需要將資料訊息轉為 NDEF Message 的形式傳送(如圖 3)，關於這部份的規範 NFC Forum 制定了 NFC Data Exchange Format (NDEF 1.0)。其中 NDEF Record 即為 NFC 技術的資料封包。關於 NDEF Record 之欄位、格式、長度等的相關定義。



圖 3、由數個 Records 組成的 NDEF Message

NDEF Message 是由許多 Records 所組成，Record 即為 NFC 通訊的資料封包，其封包的格式有長、短兩種格式，其最主要的差別即如名

稱所言，在於資料封包的長度大小。Record 的結構大致上可以分成三個部份：

- Record 的狀態：Record 的第一個 Byte。紀錄 Record 在 NDEF Message 的位置(MB、ME)、型態(TNF)、長或短格式(SR)等資訊
- Record 的長度：記錄此 Record 中所夾帶的負載的長度。隨著 Record 格式的長、短有所不同。短格式有一個 Byte，亦及負載大小最多為 28 等於 256 個 Bytes。而長格式則有四個 Bytes，負載長度可以高達 232 Bytes
- Record 的負載：擺放欲傳送的資料或指令

三、訂定通用訊息與溝通流程

在理想上，一般 NFC 應用的操作情境，使用者在操作的時候只需要將 NFC 功能打開，然後把 NFC 裝置靠近在一起，便會自動完成應用服務，此動作亦有一個名稱“接觸(Touch)”。某些較複雜的應用服務，則需要使用者確認。

若要使得 NFC 應用服務在 NFC 裝置上，能夠如理想中自動化、人性化的應用情境，那麼在 NFC 應用服務方面就需要有一個通用的應用服務訊息格式以及其溝通的協定。

經過綜觀 NFC 的應用服務後，可以觀察出大致上分成五種類型(如圖 5)，而其中所需要傳送的必要資料有：

- 應用程式名稱(Application Name)：在近場通訊裝置中，可以安裝不只一種的應用程式，來提供不同的應用服

務。故訊息中必須指明該訊息應由何應用程式來處理，抑或此訊息為何應用程式所有。例如，電子票卷中，要有發行公司之名稱

- 執行動作(Action Value)：訊息中會帶有欲執行之動作的指令，如儲存、開啟、檢視、傳送、刪除、修改、查詢等動作。通常都是以傳值的方式代表要執行的動作，然後在應用程式中會有值與動作指令的對照表。例如，小額付費服務的訊息中可藉由“2”的值，代表“扣款”的動作
- 資料內容本體(Data Content)：應用服務程式所需要或欲傳送的資料。例如，電子票卷的驗證碼、點對點傳輸的照片檔案等
- 其他(Others)：視不同應用服務所提供的功能，其訊息亦需要夾帶相對應的資料。例如，用來描述優惠卷的文字說明、電影票的有效時間、欲傳送檔案的大小等

NDEF Message				
Service Type	Applicaton Name	Decision Code	Content	Description

圖 4、近場通訊應用服務通用訊息格式

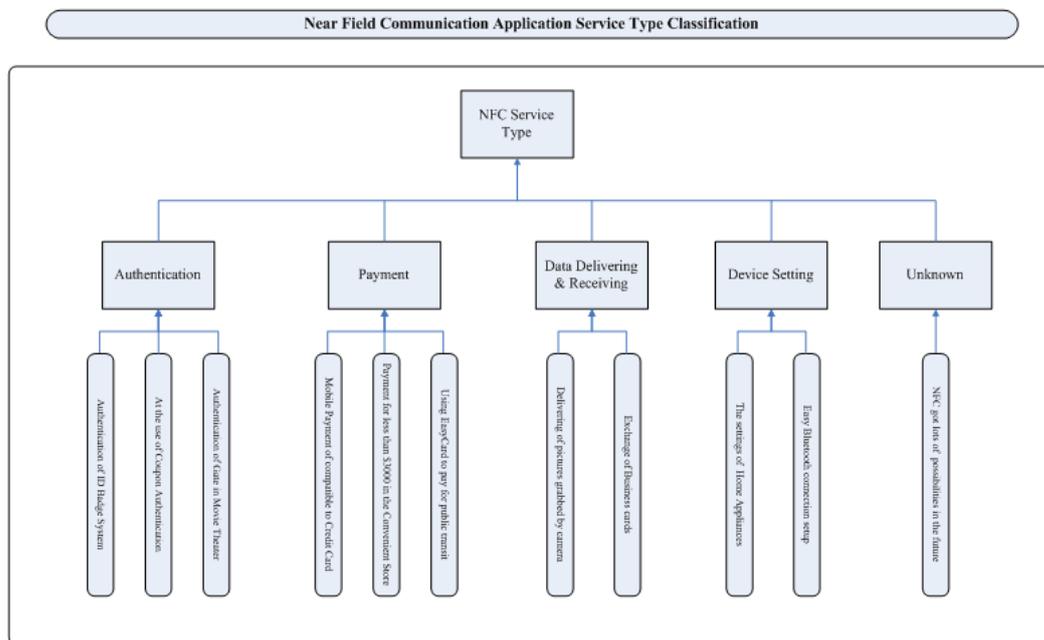


圖 5、近場通訊應用服務 - 運作模式分類圖

於是，我們可以整理出一個近場通訊應用服務的通用格式(如圖 4)，其內容欄位依序為：

- Service Type：此欄位中是紀錄應用服務訊息屬於五種運作模式中的哪一種模式(如表 1)

表 1、應用服務模式與欄位中代號對照表

應用服務模式	相對應之代號
驗證模式 (Authentication)	"AU"
消費模式 (Payment)	"PA"
資料交換模式 (Data Exchange)	"DE"
裝置連接模式 (Device Setting)	"DS"
未來模式 (Unknown)	與以上不相同者

- Application Name：此欄位中存放的資訊，代表該訊息由哪家公司所發行，又或該訊息將交由哪家公司所開發的程式所執行
- Decision Code：依照 Application 的不同，其代表的意義也會跟著不同。可為以發行之 Record 之序號，亦可為判斷該以何種動作執行運算 Content 之 Action Code，其代表的意義完全取決

於 Application 在設計時的定義

- Content：應用服務進行訊息交換時，存放訊息資料的地方。比方說，門禁驗證的金鑰、電子票卷的驗證碼、小額消費的金額、裝置連接設定時的設定參數等
- Description：應用服務對於這個訊息 Record 的描述，其描述內容是要展示給用戶端觀看的。比方說，描述門禁驗證的金鑰是屬於哪一道門禁系統的、描述電子電影票卷所屬電影的內容簡介、描述小額付費的消費資訊等

定義了 NFC 應用服務的通用訊息格式，當應用服務有共通的訊息溝通格式之後，NFC 裝置在接收到訊息時便能夠明確的理解且作出正確的反應。

NFC 應用服務的操作上有基本兩種狀態，傳送訊息與接收訊息。關於傳送訊息方面十分簡單，只要將欲傳送的資訊依照通用訊息格式編排，再編寫成 NDEF Message 的格式送出。而重點即在於接收訊息方面，當 NFC 裝置收到訊息

時，將如何去解讀，且作出反應。接收訊息及解析的流程如圖 6。

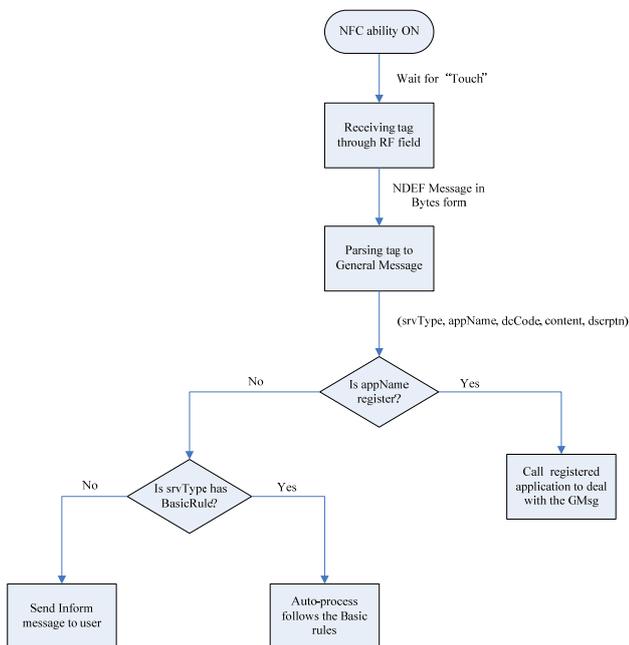


圖 6、訊息解析流程圖

首先檢查其 Application name，查詢該訊息是否以有相對應的應用程式註冊且負責處理。若有的話，便呼叫其應用程式，而後將該訊息交由應用程式作後續的判斷處理。若無相對應的應用程式註冊，則比對 Service Type，判斷其是否為五種基本運作模式其一。如果是的話，接下來依照其訊息內容，以及所屬的基本運作自動處理，執行 NFC 應用服務。若其運作模式非基本模式中的任何一種，那麼便需要通知使用者，並將接收到的訊息顯示出來，再交由使用者作後續的判斷。

四、UNFCS Framework 之建構

本章我們開始著手建立遵循且能夠解析、處理 NFC 應用服務的通用訊息格式之應用服務框架，NFC 通用應用服務框架如圖 7。主要分成三個階層：

- 應用層(Application Layer):所有應用服

務程式皆放在這一個 Layer 中。該層中的應用服務設計，皆以此應用框架為基底，然後再針對不同的應用服務需求來撰寫其所需要的功能、控制指令及訊息內容

- UNFCS 核心層(Universal Near Field Communication Service Core Layer):此 Layer 主要的目的在於管理、解析及儲存近場通訊相關的資料及 Records
- NFC 介面層(NFC Interface Layer):該層屬於近場通訊實體層的部份，為近場通訊裝置的最前端，實際與其他裝置連結(非接觸式的)的部份

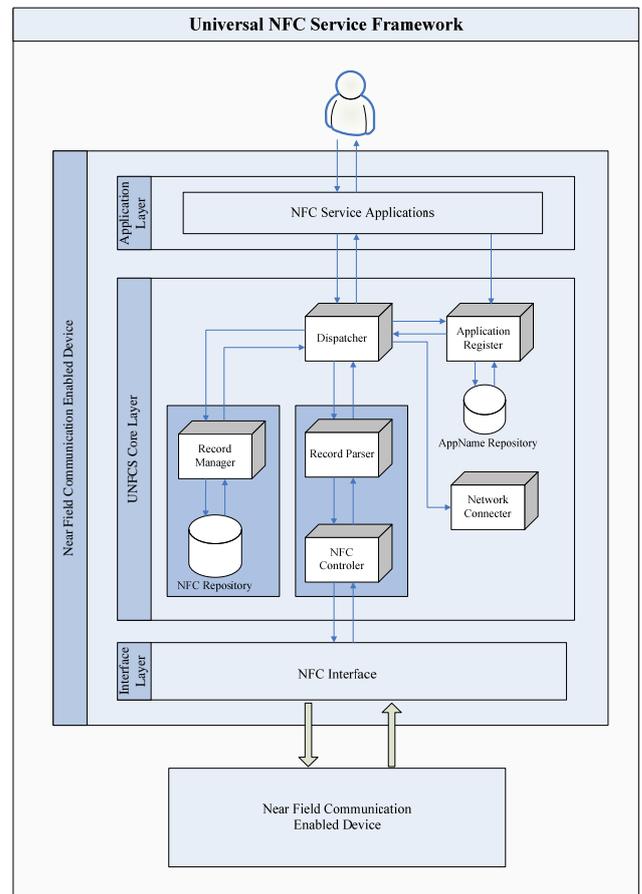


圖 7、NFC 通用應用服務框架示意圖

UNFCS 核心層則是 NFC 技術實體層的部份。UNFCS 核心層即為本章的主軸，其中包含八個部件，分別是 Dispatcher、Application

Register、Record Manager、Record Parser、Network Connector、NFC Controller、AppName Repository 及 NFC Repository。分別提供近場通訊應用程式所需要的各種能力，亦可以獨自執行簡單的近場通訊服務應用。

- Dispatcher (DP)：DP負責通用訊息之解析與判斷且為NFC工作平台與應用服務程式之橋樑(圖 8)。Dispatcher中分成三個部份，Dispatching Component (DCom)、Application Interface (AppIf) 與Basic Rule。當Dispatcher收到指令與訊息時，會由DCom做動作的判斷處理，若為基本應用服務的話，則會讀取Basic Rule中相對應的規則協助執行應用服務。AppIf則提供一個應用服務框架與應用服務程式溝通的介面。

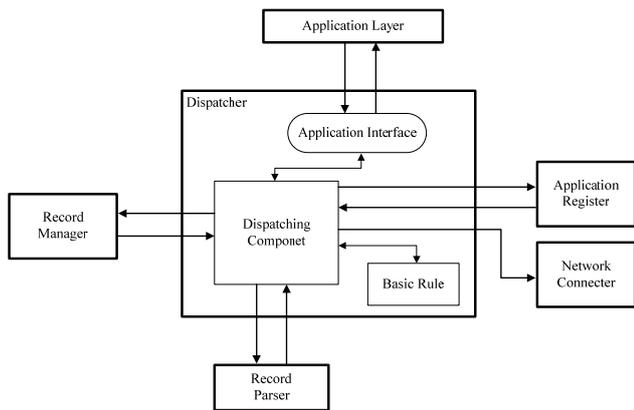


圖 8、Dispatcher 示意圖

- Application Register (AR)：AR主要協助串接應用服務框架與NFC應用服務程式，建構在此框架上之應用程式可向AR註冊。當DP收到屬於已註冊應用程式的訊息時，即會呼叫相對應的程式並將訊息轉交給程式執行(圖 9)。Register Component(RCom)提供DP查詢註冊資訊、Application Programs進行註冊等功能。Match Filter中存有註冊資訊比對之規則。

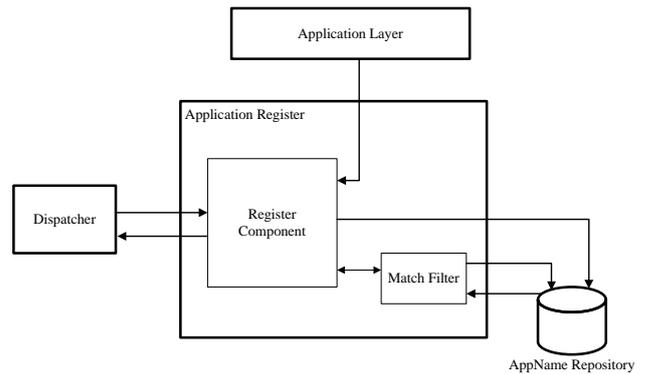


圖 9、Application Register 示意圖

- Record Manager (RM)：RM擁有操作 NFCR 中資料之能力。依照DP之請求，可以將訊息存入NFCR中，亦可執行讀取、修改、刪除、比對驗證NFCR中資料等動作(圖 10)。主要由RMCom提供上述之能力，Match Filter輔助資料的讀取與比對驗證。

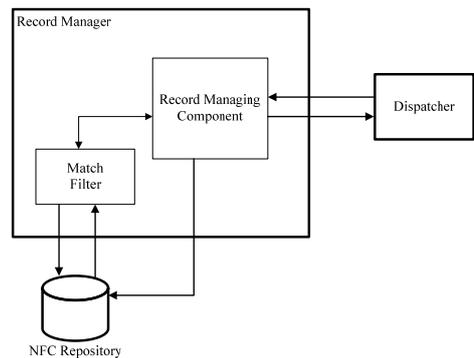


圖 10、Record Manager 示意圖

- Record Parser (RP)：RP為通用訊息格式與 NDEF Message 的中繼點，負責將訊息制作成 NFC 的 Record，再將 Record 串成 NDEF Message 交由 NFC Controller 傳送出去；反之，收到 NDEF Message 時，便將訊息從封包中讀出來，交給 DP 解析處理()。由 Record Packaging Rule 提供 NFC 定義的基本 Record 封包格式，如 Text、URI、Smart Poster 等。

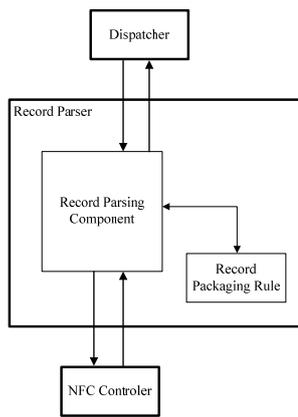


圖 11、Record Parser 示意圖

- NFC Repository (NFCR)及 AppName Repository(ANR)：此二部件為儲存單元。NFC Repository 用於存放 NFC 應用服務通用訊息，其中的資料僅開放給 RM 使用，如要對其中的資料執行儲存、讀取、刪除等動作，皆需透過 RM。AppName Repository 用於存放 NFC 應用服務程式之註冊資訊，應用

程式申請註冊時，AR 會將其註冊資訊存在其中，而 DP 收到訊息時，則會透過 AR 查詢應用程式是否已註冊。

接著簡單以「門禁驗證-設定鑰匙」與「門禁驗證-驗證鑰匙」展示兩張流程圖做為範例。圖 12是「門禁驗證-設定鑰匙」的工作流程，門禁系統(ID Badge System)建立鑰匙，並將鑰匙傳送給用戶儲存。工作初始時，鑰匙建立的部分由門禁系統之後端程式完成，接著門禁系統為初始者開啟NFC通訊並等待NFC裝置靠近接觸。NFC裝置端部分，則由用戶操作開啟NFC功能，靠近接觸門禁系統。圖 13為「門禁驗證-驗證鑰匙」之工作流程圖，當NFC行動裝置中存有鑰匙的時候，只要開啟NFC功能，靠近接觸門禁系統即會自動感應回傳鑰匙，由門禁系統驗證執行後續驗證鑰匙的動作。

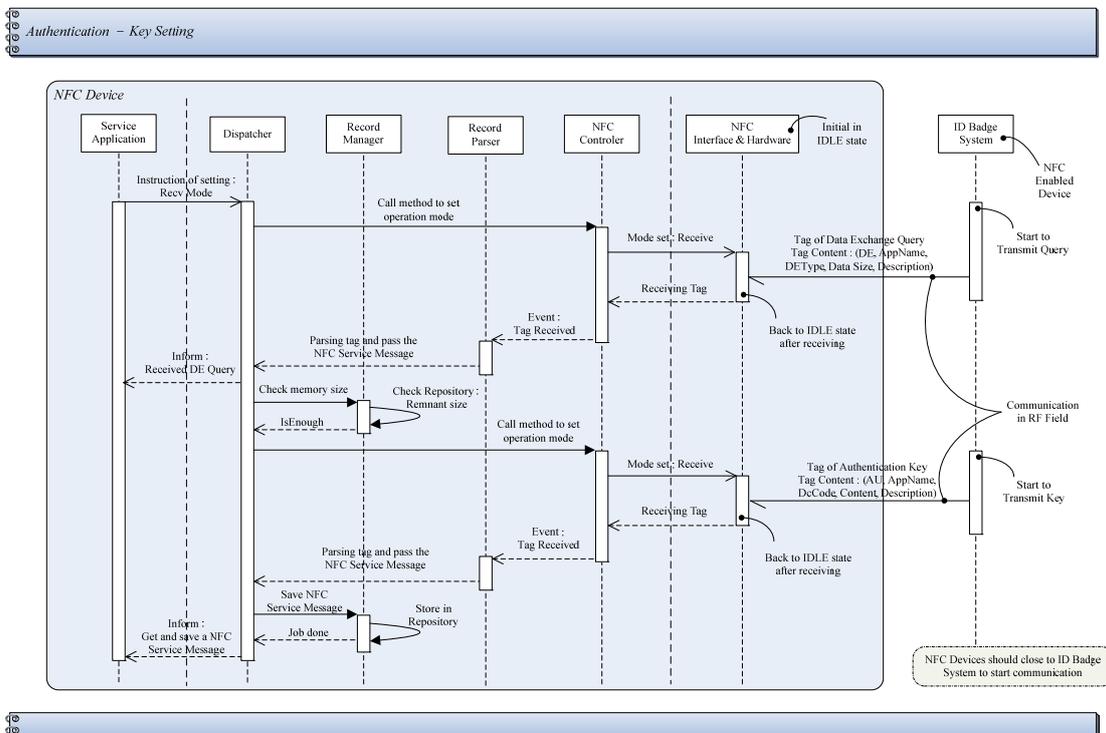


圖 12、「門禁驗證-設定鑰匙」之工作流程

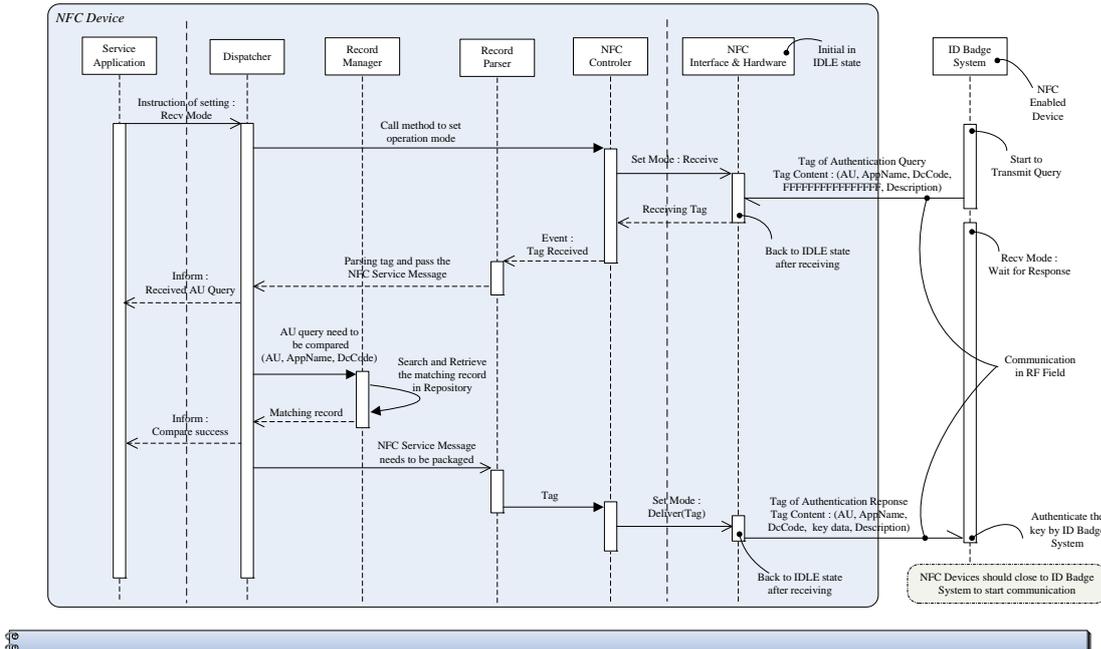


圖 13、「門禁驗證-驗證鑰匙」之工作流程圖

五、UNFCS 框架與應用服務之實作

關於NFC行動裝置部份，我們使用筆記型電腦加上具有NFC通訊能力之Reader，然後在筆記型電腦上運行Sun Java (TM) Wireless Toolkit 2.5.2 for CLDC模擬手機環境，藉此取代NFC手機(圖 14)。NFC Reader採用維識科技(WAVE-TEK) WR100 RFID/NFC HF Reader。

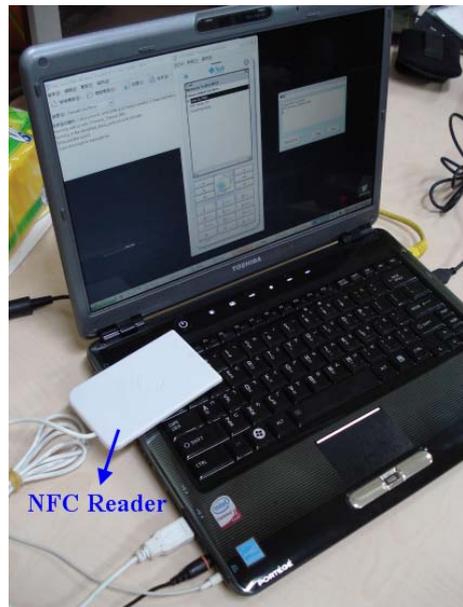


圖 14、筆電與 NFC Reader 示意圖

關於 UNFCS 框架之實作，我們選用 J2ME 將其編寫成 J2ME MIDlet，並直接部署上樹枝環境中。UNFCS 框架 MIDlet 分成三個部份：

- 用戶使用之 UNFCS Framework MIDlet 介面撰寫，使用者實際感受到與操作的介面，架構在 UNFCS 框架之各部件類別上

- UNFCS Framework 內部各部件之類別撰寫
- WAVE-TEK WR100 SDK 之串聯與修改，使用 Borland C++ Builder 6.0 撰寫，提供 NFC 通訊技術相關的實體功能，其間與 J2ME MIDlet 透過 Socket Connection 連接傳送指令，操控 NFC Reader 之狀態

實作完成的 UNFCS Framework MIDlet 之運行圖。其主要有兩個部份，分別是 Look up Files 頁面與 NFC Mode ON/OFF 頁面，用戶透過操作這兩頁面所提供的選項，即可以進行 NFC 應用服務(圖 15)。Look up Files 頁面即為 UNFCS 訊息資料庫之操作頁面。當頁面開啟的時候，會讀取 NFCR 中的資料並將目前建立的 NFC 運作模式顯示出來。NFC Mode ON/OFF 頁面即為開啟 NFC 功能之操作頁面，NFC 應用服務之運作即從此開始。

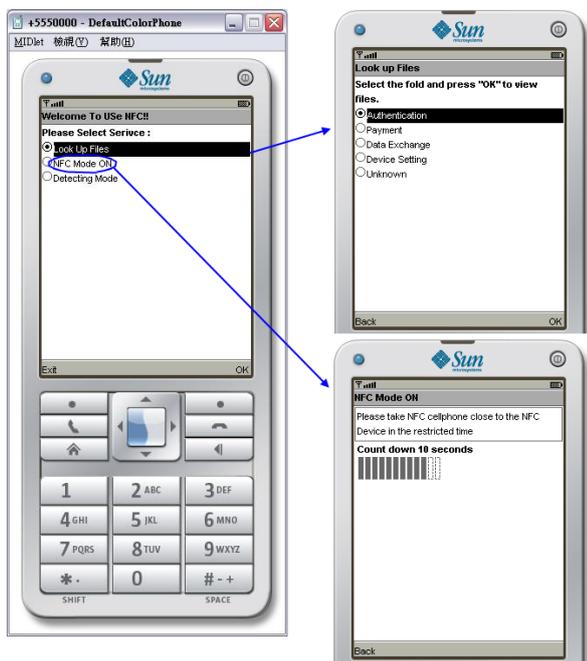


圖 15、UNFCS Framework MIDlet 運行圖

應用服務伺服端的部份，我們使用 Borland C++ Builder 6.0 撰寫 NFC 應用服務後端之訊息處理系統，並將其架設在 Intel Core 2 Due 桌上型電腦上。在此我們撰寫了一個綜合的 NFC 應用服務伺服端程式，包含有三種 NFC 應用服務系統，分別為社區門禁系統、

智慧型海報-優惠卷下載與商家小額消費系統。



圖 16、「門禁驗證-設定鑰匙」之手機介面流程

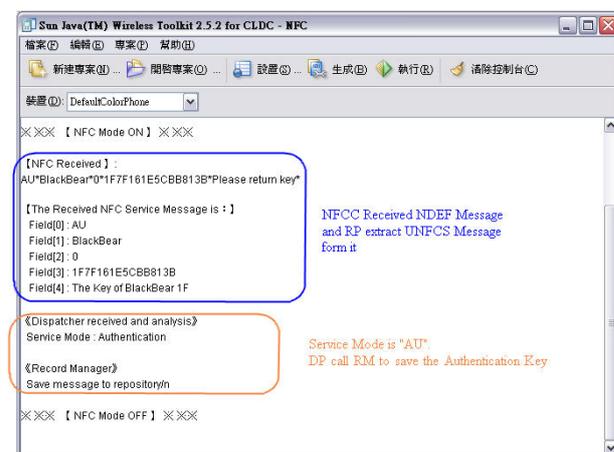


圖 17、「門禁驗證-設定鑰匙」之工作流程

六、效能分析

以下根據第五章中所舉的例子，我們測量「驗證模式-設定鑰匙」與「驗證模式-鑰匙驗證」兩個部份在 UNFCS 框架中的工作時間。其後顯示之數據皆為實測 5 次之後，再取其平均值而得的。

圖 18 為「驗證模式-設定鑰匙」之 UNFCS 框架運作時間。其中傳送的資料有請求檔案傳送之訊息 (43 Bytes) 與鑰匙資料本體 (57 Bytes)。由圖中可以觀察出，在於 Record 封包之拆解與包裝上，需要較長的時間，這是由於當收到 NFC 封包之後，還需要將訊息從封包中抽取出來，再轉換成通用訊息格式。接著，當訊息以通用訊息的形式在 UNFCS 框架下運行時，即可進行快速的判斷與處理。

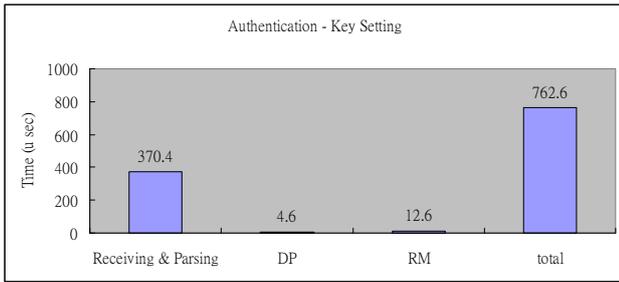


圖 18、「驗證模式-設定鑰匙」之 UNFCS 框架運作時間

圖 19~圖 21 為鑰匙驗證在 UNFCS 框架下之工作時間數據，其資料庫中分別儲存著 1 隻、5 隻與 10 隻的驗證鑰匙。從數據中可以看出當資料庫中的鑰匙從 1 隻增加到 5~10 隻的時候，Record Manager 在比對的時候，時間明顯跟著增加。而 Dispatcher 的部份，在此僅需要做單次的訊息要求的判斷，其工作時間小於 10^{-6} 秒。

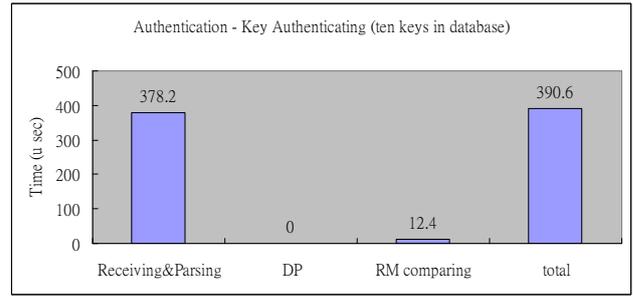


圖 21、鑰匙驗證 (資料庫中：十隻鑰匙)

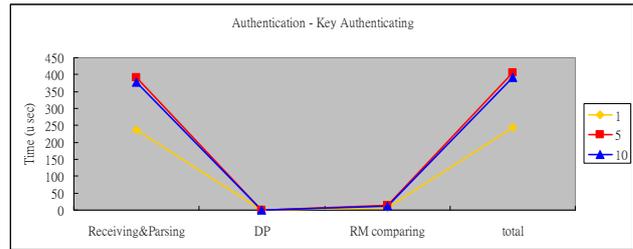


圖 22、鑰匙驗證時間比較圖

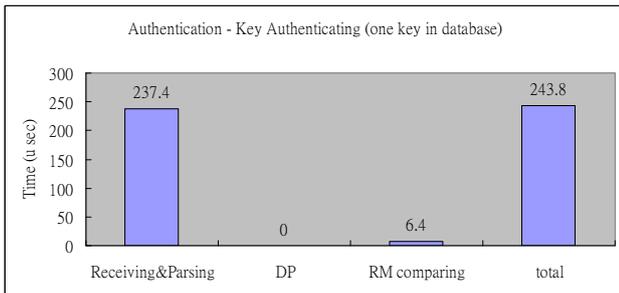


圖 19、鑰匙驗證 (資料庫中：一隻鑰匙)

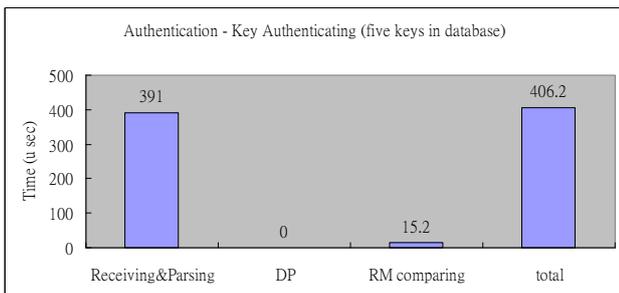


圖 20、鑰匙驗證 (資料庫中：五隻鑰匙)

七 結論與未來研究方向

本論文在緒論中，由現今 NFC 之應用服務以及論文文獻等方面進行探討，發現現今的 NFC 應用服務，尚無一個良好的訊息溝通協定與整合的應用服務互動框架。於是在本論文中，先從整合 NFC 應用服務訊息開始，訂定一個泛用於各種 NFC 應用服務的訊息格式，接著規範其溝通與解析的流程，最後再建立整合的應用服務互動框架。

經過本論文的研究之後，我們成功的規範出一個良好的 NFC 應用服務溝通協定，且建立一個名為 UNFCS 近場通訊應用服務整合框架確保訊息的溝通能夠流暢。此外，我們尚藉由撰寫幾個 NFC 應用服務程式，模擬本論文之框架在 NFC 應用服務中之使用情境。透過與 NFC 應用服務的互動可以明顯看出，應用 UNFCS 框架建構出之 NFC 應用服務，能讓使用者在操作時，感到更加的自動化以及人性化。

在未來延伸研究的部份，我們希望能進一步將本架構實作到 Nokia 6131i 等 NFC 手機上，如此便能夠直接跨足實際生活之應用。除

此之外，也希望將檔案傳輸、消費以及驗證部分的安全性考量納入討論

參考文獻

- [1] 王宏文，“淺談近距離無線通訊技術”，*網路通訊雜誌*，2005.7
- [2] Yaw Anokwa, Gaetano Borriello, Trevor Pering, Roy Want, “ A User Interaction Model for NFC Enabled Applications”, *in Proc. PerComW'07*, 2007, pp. 357-361.
- [3] Zoe Antoniou, Srikant Varadan, “Intuitive mobile user interaction in smart spaces via NFC-enhanced devices”, *in Proc. ICWMC'07*, 2007, pp 86-86.
- [4] Márton Csapodi, András Nagy, “New Application for NFC Devices”, *Mobile and Wireless Communication Summit, 16th IST*, 2007, pp 1-5.
- [5] NFC Forum, “[NFC Interface and Protocol \(NFCIP-1\)](#)”, *ISO/IEC 18092 、 ECMA340*, 2004