私立東海大學資訊工程與科學研究所

碩士論文

指導教授：周忠信

智慧型網路名稱服務

Web Intelligent Naming System

研究生：姜天戩

中華民國　九十一　年 七 月 十八　日

摘要

本論文旨在提出一個網際網路上的智慧型名稱服務。由於現存此類型的服務，皆不允許人們以日常生活中認知名稱的習慣來使用網路，必須遵守某些命名規則或完全符合於特定的關鍵詞才能獲得。因此，本研究將探討表達方式不同但概念相同之名稱如何對應至單一網址。換言之，現行的名稱服務皆為生硬的「一對一關係」，也就是一個名稱僅能代表一個網址，若使用者查詢時輸入的名稱與之未能完全符合，便無法獲得解答；而本研究的目標則是改善這個狀況，替名稱與網址之間建立有彈性的「多對一關係」，意即使用者能以個人的詞彙來指稱網址，名稱系統自會找出語意上相同的解答。為此，本系統的實作採用了泛稱為 "yellow-pages" 的目錄服務式架構，並搭配具有同義詞集的辭典，設計了一個智慧型的比對方式。使用者輸入的字串，在這個比對中將先後經過詞性標定、metadata 查詢及條件過濾，以判別出最適切的網址。在實作完成後，本研究還將此名稱服務公開於網際網路上運作將近八個月，以其間收集的紀錄資料，進行使用行為的統計，而由其結果得知本系統已達到 80% 的命中率，驗證了本研究效果良好；同時更進一步分析這些數據，發現一般人對於網址名稱的認知方式，以藉此找出本系統可以繼續改進之處。此外，目前開放在網際網路上的智慧型名稱服務，全由 Java 開發而成。

Abstract

This thesis introduces a human-friendly Internet naming service called Intelligent Naming System (INS). It provides a mechanism to map a name to its corresponding Internet address or a web URL. The name used in INS can be any human understandable string in natural language. It is not for computers or networks to understand, and therefore it is not necessary to follow the same syntax similar to Internet address, domain name or URL. The goal of this thesis is to improve the usage of Internet naming, or web browsing, as easy as possible by automatically binding conceptually equivalent names to the target address or URL. This solution is especially useful for non-English language users since the names can be in any natural language. The architecture of INS is a directory-enable solution. There are three major components in INS. The first component is the directory that is used to store dictionary and the metadata of the naming knowledge. The second component is a quick matching service which is responsible for resolving the mapping between names and their addresses managed in the cache. The third component is called intelligent matching service which is used to resolve the mapping required the support of additional knowledge. After the mapping result set is obtained, a rule-based filter is applied to generate the final mapping. The Intelligent Naming System has been implemented and freely accessed on Internet for about eight months. After analyzing the logged information collected in this period, it shows that the mapping resolved by INS is over 80% correct. Also, from this study, we can identify some valuable user behaviors when using human-friendly naming service instead of using traditional Internet addressing or a web URL. INS is totally developed in Java and its performance is quite good especially when using in the Internet world.

目次

摘要 i

Abstract ii

目次 iii

圖索引 iv

第一章　緒論 1

1.1 **研究動機及目的** 1

1.2 **章節內容** 7

第二章　現存名稱服務相關技術探討 8

2.1 **網域名稱系統** (Domain Names System, DNS) 8

2.2 **國際化域名** (Internationalized Domain Names, IDN) 9

2.3 Uniform Resource Identifier (URI) 11

2.4 Keywords System (KS) 14

2.5 Common Name Resolution Protocol (CNRP) 15

2.6 **總結** 17

第三章　智慧型名稱服務架構 18

3.1 **系統架構概觀** 18

3.2 **查詢機制** 27

3.3 **總結** 33

第四章　名稱服務使用行為分析 34

4.1 **範例劇情** 34

4.2 **使用行為分析** 37

4.3 **總結** 45

第五章　結論及未來展望 46

5.1 **結論** 46

5.2 **未來展望** 47

參考文獻 48

圖索引

圖 1.1 名稱與網址的關聯在不同名稱服務及搜尋引擎上的行為 5

圖 3.1 系統架構圖 19

圖 3.2 Directory 的 schema 20

圖 3.3 Quick matching service sequence diagram 28

圖 3.4 Intelligent matching service sequence diagram 28

圖 4.1 瀏覽器外掛程式 34

圖 4.2 多筆網址選擇頁 36

圖 4.3 網址無法辨認頁 37

圖 4.4 原始命中率 38

圖 4.5 無對應結果中非網址名稱與網址名稱的相對比例 39

圖 4.6 網址名稱命中率 39

圖 4.7 各類型網址名稱與總查詢人次的相對比例 40

圖 4.8 回覆多筆網址結果的佔總查詢人次的相對比例 41

圖 4.9 加上地名為修飾詞後可獲得單一正確網址的比例 42

圖 4.10 詞長之於所有查詢中相對比例和命中率的關係 43

第一章 緒論

1.1 研究動機及目的

近幾年來,「網際網路」(Internet) 迅速地普及至各個族群，不再為資訊領域的專業人士所獨享。例如在「全球資訊網」(World Wide Web, WWW) [1]，為了指出網路資源所在位置而設計的 "Uniform Resource Locator" (URL) [2]，已不僅使用於「網路瀏覽器」(Web Browser) 對電子媒體如「超文件」(Hypertext) 的存取交換上，更進一步地出現在傳統媒體如電視廣播或報章雜誌的新聞廣告中。

然而，即使在日常生活中已經隨處可見這些網路資訊出現在所謂「離線」(off-line) 的環境中，實際使用網路「上線」(on-line) 時，仍有不少讓非專業人士望之卻步的問題存在。以 URL 為例，最顯著的障礙發生於其格式只能使用英文來表達的特徵，無論在傳播上、記憶上乃至於理解上，都較使用母語來得困難。事實上，URL 經由「網域名稱系統」(Domain Name System, DNS) [3] 建構出的「階層式名稱結構圖」(hierarchical naming graph)，來轉換對應的「網際網路協定位址」(Internet Protocol Addresses, IP Addresses)，其初衷即較為偏向利於電腦程式的處理，而非針對人們使用時的「可讀性」(readability) 所設計；因此即使是一位深諳英文的使用者，若不是對 DNS 的命名規則稍具了解，面對 "www.thu.edu.tw" 這樣的網址，仍可能對其略詞如 "thu" 或其格式中的 "." 所代表的意義不甚明白，進而間接地造成記憶上的困難。

有鑑於此，負責網際網路標準審訂的 "Internet Engineering Task Force" (IETF)

[4]、各地區的「網路資訊中心」(Network Information Center, NIC) 及數個民間企業皆紛紛提出了各自的解決之道。而這些解決方案的實作方式大致可以分為兩種，一類是仍然採用現有 DNS 命名規則的架構，主要為 IETF 研議中的「國際化域名」(Internationalized Domain Name, IDN) [5]，其中幾個使用非拼音語系之地區性 NIC 如 CNNIC (China NIC) [6]、TWNIC (Taiwan NIC) [7]、KRNIC (Korea NIC) [8] 等，已各自提出修改過的 DNS 伺服器，以使其母語能用於域名的表示；另一類則以關鍵字 (Keyword) 或一般名稱 (Common Name) 對應至 URL 的解析 (Resolution) 為主，通稱為「關鍵字系統」(Keywords System, KS) [9]。

上述的狀況，可說是為了對更具「親和力」(Human-friendly) 之網址名稱的需求而生的因應之道。基本上，一個新的名稱服務若要滿足這項需求，由已知的解決方案中，可歸納出以下兩種特性：

一、讓網址命名能在「多國語言」(multilingual) 式的環境下進行；在瀏覽器的網址列上將不再只能輸入英文。

二、可直接使用「自然語言」(natural language) 的方式來表示網址；網址不再需要遵守某些命名規則，如 DNS 以 "." 來區隔階層的語法。

第一項是每種現存的服務皆已觸及的領域，換言之，單純地讓各種語言能夠直接使用於網址名稱上，已是最低限度內必須提供的功能，至於其中的技術或標準各有巧妙不同之處，將留待第二章再加以說明；但在第二項特性上，不同的實作方式就有了明顯的差異：採取相容於 DNS 命名規則的作法，因為依然保留著

階層式結構，使得“www.thu.edu.tw”變成「東海大學.教育.tw」之類的形式，僅可說是類似直譯自英文的橫向移植，往往不如 KS 的關鍵字來得更直觀、更容易讓人們記憶及理解；另一方面，若要真正地將網址的名稱提昇至自然語言的層次，光靠 KS 提供單一的關鍵字對應，也存在著力有未迨之處。試想平日人們對於同樣的一件事物，因著個人認知及習慣的差異，就可以有許多不同的命名方式，好比“www.thu.edu.tw”可稱為「東海大學」，也有可能簡稱「東大」，或是加上某些修飾詞形成「台中東海大學」或「私立東海大學」等。若按照現行 KS 的作法，為了對應這些概念上相同但表達方式不同的名稱，就必須針對同一個網址註冊數量可觀的關鍵字，否則難免掛一漏萬。因此這些「一對一關聯」(1-to-1 relation)的對應方式，已隱含了增加管理者、註冊者與使用者三方面皆有所困擾的風險。

從另一個角度來看，若要透過關鍵字來取得網址，對網路有一定熟悉程度的使用者，多半會經由搜尋引擎來達到目的。隨著搜尋引擎技術的提昇，大多數情況下已能準確地提供使用者預期的結果，甚至能夠直接「導向」(redirect) 至最佳的網址，如 Google 的「好手氣」功能 [10]。此時是否還需要名稱服務，便成了一個有待釐清的疑問。事實上，由使用者輸入的查詢字串與系統輸出的網址結果之間的關係，便能替名稱服務與搜尋引擎作出一個明確的分野。一般來說，搜尋所展現的行為是種「一對多關聯」(1-to-many relation)。使用者輸入關鍵字後，將得到一個包含許多 URLs 的結果集合，而在大多數搜尋引擎的作法中，這個一

對多的關聯取決於該關鍵字與結果集合中各個 URL 所指向的文件內容 (content) 間的「相似度」(similarity) [11]。此時若使用者輸入的關鍵字稍有不同，即可能出現相去甚遠的結果，例如在 Google 搜尋「東海大學」，得到約 27,100 筆結果，其中第一筆結果正是 “www.thu.edu.tw”；但搜尋「台中東海大學」時，得到的結果只有約 401 筆，而且第一筆結果並未正確地指向 “www.thu.edu.tw”，而是一個內含此字串的網頁。於是，當使用者輸入一個名稱時，若期待的結果為對應之「網址」，便屬於名稱服務的工作；而欲獲得「內容」相關的網頁，則應交給搜尋引擎處理。這個責任分配，將影響到本論文第四章之中的分析結果。

本研究的看法是，一個理想的名稱服務，應該允許使用者以各自的習慣用語來查詢。即使用字遣詞上有所差異，只要概念上都指向相同的目標，便應能順利地連結至唯一的網址。此時，無論是「東海大學」、「東大」、「台中東海大學」或「私立東海大學」等多種不同用法，都將透過該服務，連結至同一個網址－ “www.thu.edu.tw”。換言之，本研究企圖將名稱與網址之間的關聯建立在概念 (concept) 的「等同」(equivalent) 上，與前文提到的現存名稱服務或搜尋引擎相較，則可稱為一種「多對一關聯」(many-to-1 relation)。在圖 1.1 中，表達了本章所提到的這三種關聯之間，以網址 ”www.thu.edu.tw” 為例時呈現的行為差異：IDN 方面以 TWNIC 的作法為例，其服務僅能處理完全符合格式如「東海大學.教育.tw」的查詢，為「網域名稱」(Domain Names) 與網址的一對一關聯；Keywords System 雖不必如 IDN 使用以網域為基礎的語法，仍只能處理完全符合的關鍵字。

假設該服務中僅有「東海大學」與 "www.thu.edu.tw" 的對應，則「台中東海大學」便無法直接獲得解答，也形成了關鍵字與網址的一對一關聯；若使用搜尋引

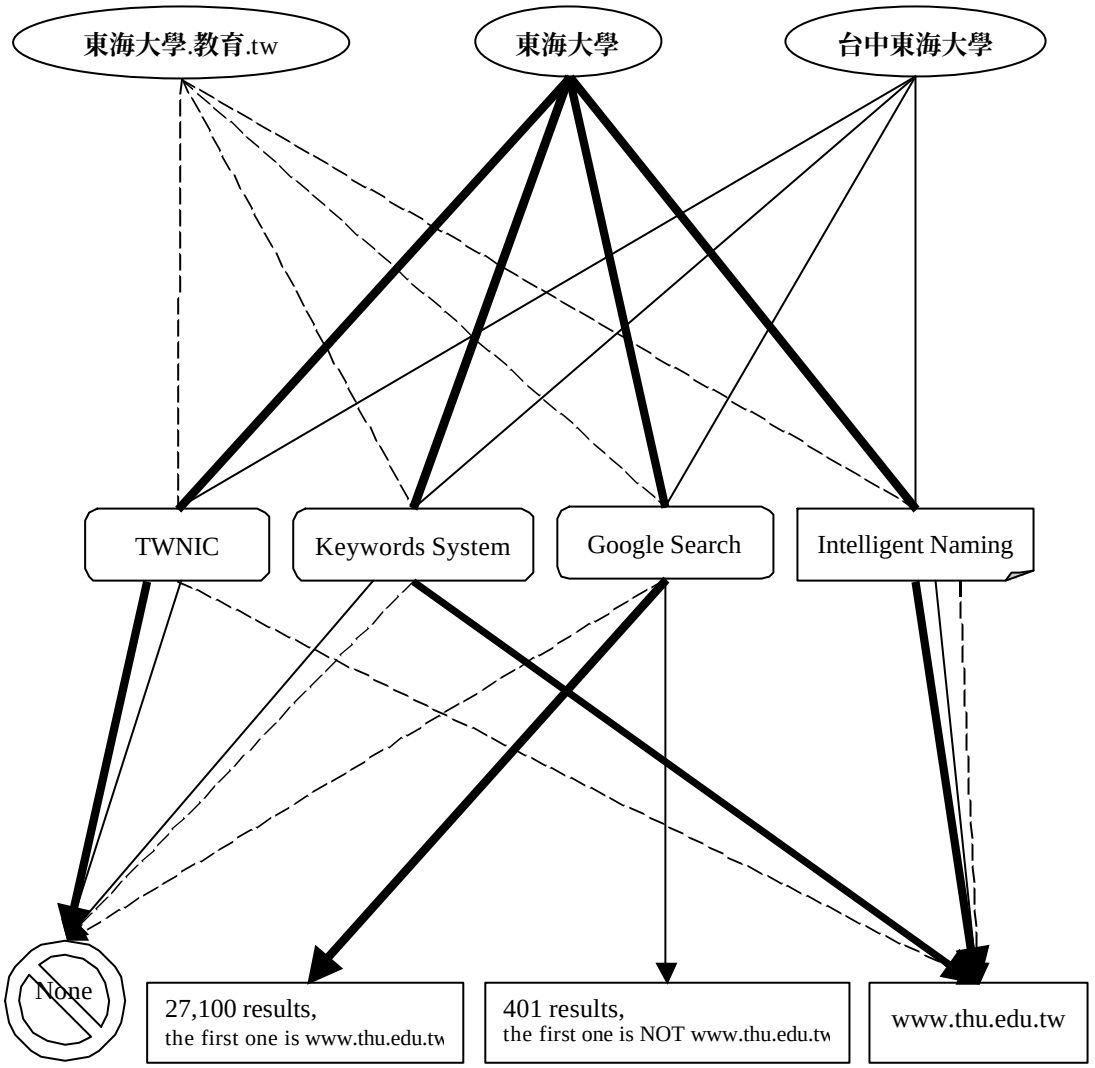

圖 1.1. 名稱與網址的關聯在不同名稱服務及搜尋引擎上的行為

擎如 Google，「東海大學」及「台中東海大學」的搜尋結果，就像前文中已提及的，不但都是傳回多筆網址，明顯為關鍵字與網址的一對多關聯，其結果數量往往不同，且第一筆解答也不一致；然而無論是「東海大學」或「台中東海大學」，都能透過本研究提出的名稱服務得到唯一的解答，至於「東海大學.教育.tw」格

式的名稱，因為屬於 DNS 的層次，本服務也會直接略過，轉交使用此語法的 TWNIC 處理，展現了不同表達方式之名稱與網址間的多對一關係。

本研究之所以稱為「智慧型」的名稱服務，其著眼點即在於更接近人性的、更友善的使用方式。好比人與人溝通時，不必刻意統一詞彙使用上的習慣，便能自然而然地意會對方命名指稱的對象。而本研究在實作上將透過一個瀏覽器專用的「外掛程式」(plug-in)，由 HTTP 統一將客戶端的查詢傳送至伺服端，藉此運行於現行系統之上，向下相容於目前的通訊協定。而伺服端則以所謂 “yellow-pages” [12] 的概念，即「目錄服務」(directory service) 的原理，配合一個「同義辭典」(Thesaurus) [13, 14]，也就是整理了每個已知詞彙之「同義詞集」(Synonym Set) [13, 14] 的辭典，針對使用者輸入的名稱進行比對，以因應不同網址名稱使用習慣上的需求；其後並配合過濾規則的套用，檢閱比對的結果以找出最適切的解答。必須特別界定的是，本研究的範圍定於「名詞片語」式之網址名稱上，因為更複雜的自然語言句型處理已超出名稱服務的責任；而一般所謂多國語言式的名稱服務，指的是架構上對各地編碼系統之相容，並非各地語言的翻譯或慣用詞彙的對照，因此本研究實作上將只針對台灣地區通行的網址名稱為目標。另外，為了驗證本研究的效果，該名稱服務已公開於網際網路上實際運作，在將近八個月內收集了約十萬人次的查詢紀錄，以作為網址名稱使用行為的分析依據。根據該分析結果，可知本研究實作之名稱服務，其直接命中率已達 80%，

並能對網址名稱的使用習慣提出建議。同時，目前本研究成果已實際技術轉移供業界使用 [15]。

1.2 章節內容

本論文的第二章將簡介並檢討網址名稱服務現存的相關技術，包括網域名稱系統 (DNS)、國際化域名 (IDN)、Uniform Resource Identifier (URI)、Keywords System (KS) 及 Common Name Resolution Protocol (CNRP)；第三章則說明本研究提出的智慧型名稱服務其架構中 Directory 的 schema 與負責服務的各項元件，以及查詢機制的互動過程；第四章分析並探討了本研究於網際網路上實際運行時收集的使用行為紀錄，藉以驗證實作結果並提出對網址名稱使用上的建議；最後乃於第五章提出本論文的結論及未來發展方向。

第二章 現存名稱服務相關技術探討

2.1 網域名稱系統 (Domain Name System, DNS)

「網域名稱」(Domain Names) 建構在階層式的名稱結構圖上，簡稱為「域名」，事實上就是一個涵蓋全世界網路位址與名稱對應的樹狀結構。每一個網際網路上的位址，都應擁有一個由英文縮寫表示的，以其行政區域為「根節點」(root node)、組織類型為「子節點」(child node)、組織名稱為「次子節點」(sub-child node)，並以句點 "." 分隔所構成的域名，而「葉節點」(leaf node) 則是每個已註冊的「主機名稱」(host name)。以台灣地區為例，根節點為 "tw"，子節點計有 "gov"、"org"、"net"、"com"、"edu" 及 "idv" 幾類。使用者於瀏覽器網址列輸入的形式，其實可以視為英文在表達地址時依照每個單位階層高低而遞增排列的習慣。因此，在 DNS 中一個表達了「東海大學」概念的名稱，由左至右將展現為網域階層由下而上的順序，與使用漢字的中、日、韓 [16, 17, 18] 等地區在語言習慣上剛好相反。同時，因為一般 DNS 伺服器採用的是 UNIX 平台上的 Bind 程式，仍有許多早期版本僅支援 7 位元的 ASCII 碼，使得非英文字元無法直接相容於域名架構。

由於域名是樹狀的架構，因此每個已註冊的主機都有固定的位置，當以名稱查詢網址時，使用者所預設的 DNS 伺服器會依照其輸入的域名結構，在域名樹中「追蹤」(traversal) 每個「路徑」(path) 上是否有符合的節點；換言之，這個查詢在沒有其他技術輔助的前提下，必須要「完全符合」(exact match) 代表每個

節點的英文縮寫才能獲得對應。一般來說，DNS 在使用上所提供最基本的彈性是，除了其紀錄型別為 “IN” 的主機名稱之外，尚可設定 “CNAME” 作為「別名」(alias)。然而在組織類型與組織名稱的層次上，卻充斥著相當程度的混淆。例如早先發燒過一陣子的 “.com”，在美國或台灣即代表了商業組織，但在日本與韓國，相同意義的節點卻是以 “.co” 標示；而且在 “.com” 風潮流行時期，由於該組織類型的節點使用率爆滿，Internet Corporation for Assigned Names and Numbers (ICANN) 被迫新增了 “.biz” 組織類型，暴露出域名架構一方面必須維持一定的命名規則，無法直接使用自然語言查詢，一方面又容許相同概念的組織類型被不同的英文縮寫表示，產生了近似自然語言中「多詞同義」(synonymy) 的問題而不主動解決，徒增使用者與註冊者的困擾。

2.2 國際化域名 (Internationalized Domain Name, IDN)

在 IETF 與 ICANN 的主導下，IDN 是目前域名為支援多國語言而制定中的標準。由於多國語言已牽涉到各種編碼技術標準的取捨問題，以及在中日韓表意字元上發生的異體字對應問題，使得 IDN 尚未能有一個最終且最佳的解決方案出現。這個部分有著兩種恰恰相反的實作方式，將於下文中簡介之。

- **ASCII Compatible Encoding (ACE)**

  多國語言的支援問題，目前最為流行的方案是仰仗 ISO 10646 所制定的 Universal Character Set (UCS)。無論是採用 UCS 中的那一種實作方法，皆

需透過兩個以上的位元組來表達其字元。然而原始的 DNS 機制僅能辨認以一個位元組儲存的 ASCII 編碼，故針對 IDN 所提出的草案，有不少的考量集中在相容於 ASCII 的策略上，也就是所謂的 ASCII Compatible Encoding (ACE)。事實上，這個多位元組字元轉換成單位元組字元的過程，原理上就是一種壓縮法的應用。目前關於 ACE 有相當多的實作方式，如 Simple ACE (SACE) [19] 或 Row-based ACE (RACE) [20] 等，皆在 IETF 之下的 IDN 工作小組 [21] 中發表其草案，並偶有 IEEE 論文 [22] 針對其中某一種作法來發展，就不再此一一詳加說明。

- **Native Encoding**

  與 ACE 相反的思考方式，則是放棄直接使用傳統的 DNS，而透過修改 DNS 伺服器程式，來增補其對多國語言各式字元集的支援能力。以漢字的問題為例，目前在東亞地區的幾個網路資訊中心所提出的架構裡，除了日本的 JPNIC 採用 ACE 式的解決方案之外，其餘如台灣的 TWNIC、中國大陸的 CNNIC 及韓國的 KRNIC [16, 17, 18]，都傾向於此類型的實作方式，各自提出了以適應該地區母語 (native language) 為主的「本土化」(localized) DNS 服務。表面上它們只需要修改 DNS 伺服器的程式，使其能夠正確解譯其母語慣用的編碼系統即可，然而實際運作時卻常因為網際網路的瀏覽環境及通訊協定等造成了種種例外狀況，讓使用起來難免顯得綁手綁腳。譬如在台灣推行中文網域的 TWNIC，其作法就常常碰上無法

穿透代理伺服器或防火牆的窘境。

2.3 Uniform Resource Identifier (URI)

根據 World Wide Web Consortium (W3C) 的定義 [2]，Uniform Resource Identifier 是用來識別 (identify) 網路資源 (web resources) 的短字串，使得各種名稱定址方式及通訊協定能在統一 (uniform) 的規格下操作，讓原先可能需要經過「登入某伺服器、執行某指令」等多項手續的流程，簡化為滑鼠點擊一次即可完成所有工作的方法。URI 包含了通行已久的 URL 及尚在研議中的 URN [24] 兩個子集合，URL 與 URN 之間又有所交集，將分別於後文簡介。

- **Uniform Resource Locator (URL)**

  URL 透過簡單的格式，讓使用者能以相同的語法，存取數量龐大且類型繁多的網路資源。一般情況下，其字串中冒號 ":" 左方表示了通訊協定類型，也就是該網路資源提供服務的方式；而 ":" 或 "://" 右方至分號 "/" 左方之間的部分，就是前面已經介紹過的 DNS 語法；最後在 "/" 右方的子字串，則表示了該資源在其位址上的主機之中，檔案系統存放方式的「相對」(relative) 路徑。例如 "http://www.thu.edu.tw/index.html" 指出的就是在 "thu.edu.tw" 網域的 "www" 主機上，透過 HTTP 協定提供 web 服務的根目錄下之的檔案 "index.html"。URL 使用上雖然方便，但本質上仍是以技術層面為出發點所設計的語法，就一般使用者來說還不夠友善，其原

因主要在於 URL 無法靈活應付動態的需求。所謂動態需求，常見的情況有二：一是使用 URL 時，碰上該 URL 指出的網路資源實際上已經不存在於該位置的情況；另一是某網路資源在不同的位址擁有許多「複本」(replica)，但這些相同的物件卻各自有著截然不同的 URL，使用者往往無從得悉，或是面對著這些 URLs 不知應如何選擇。事實上，這個部分已牽涉到名稱服務能否提供近乎「永久有效」(persistent) 之網址名稱的問題。有些人傾向於將此問題留給 URI 的另一個成員－URN 來解決，但也有不全然贊同這個方式的看法，像是 W3C Hypertext Style 中的 "Cool URIs don't change" [23]。

- **Uniform Resource Name (URN)**

  URN 作為一種 URI，其訴求正如前面所提及的，著眼於一個網路名稱的「永久性」(persistence) 及「有效性」(availability)，在一般情況下，也就是一種「與位置無關」(location-independent) 的識別方式。事實上，URN 不但與位置無關，也不一定要是 "on-line" 的資源，例如某本書的 URN 可能以其 ISBN 表示為 "urn:ISBN:0-123-456789"，而該書不見得能在網際網路上找到電子版，也許只有紙本存在於某個圖書館中，甚至也有可能是已亡佚的古籍。這個類型的 URN，除了隸屬於 URI，也將和 Digital Object Identifier (DOI) [25] 有所關聯。IETF 針對 URN 的需求，已經作出了 RFC 2141 [26] 來規範其語法。但因著 URN 之於「不變名稱」的部分，需要新

的方式來進行解析對應，實作方法各異。其中有透過現行 URL 格式來運作的 Persistent URL (PURL) [27]，形成了與 URL 的交集；也有由 The Corporation for National Research Initiatives (CNRI) 主導，與 DOI 相關的 Handle System [28]。

- **Human-Friendly Resource Names (HFN)**

  Ballintijn、Steen 及 Tanenbaum 於 2001 年提出的 "Scalable Human-Friendly Resource Names" [12]，其目的與 URN 相似，希望能作到 "Naming Replicated Resources"；但與 URN 不同的是，HFN 的訴求在於提出一種比 URN 更可讀 (human readable) 的名稱服務，企圖讓使用者能夠更容易由名稱望文生義，而不像 URN 在 RFC 1737 [24] 中僅被要求能直接抄寫 (transcribe)。就前文提及透過 URN 識別的書本資源為例，採用數字編號如 ISBN 以求永久有效性，但使用者便不易直接了解該 URN 的意義以幫助記憶；HFN 則希望能以名稱來表示該資源。接著 Ballintijn 等人提出了 "yellow-pages approach and white-pages approach" [12] 的看法。所謂 yellow-pages，指的就是以電話簿分類廣告「黃頁」為譬喻象徵的目錄服務式作法，通常是透過 LDAP [29] 存取其中的 attributes 以作為命名的依據；而 white-pages，則是以「名稱結構圖」(naming graph) 來運作，也就是 DNS 所採用的方式。HFN 所選擇的是 white-pages 架構，利用修改過的 DNS 來儲存其名稱。於是使用 HFN 時，將先經過這個特殊化的 DNS 提

供名稱服務來解析，再對應到其系統中負責解決複本問題的「定位服務」(location service)，取得該名稱的 URN，以及該 URN 可能進一步對應的 URLs。於是，HFN 可說是一種新的 URI，與 URN 和 URL 的目標皆有所不同。它包含了名稱服務的實作，但解析完成之後仍會回到現行的 DNS 架構上，與前文中簡介 IDN 時曾提及某些修改 DNS 作法以改變名稱服務的方式，也有著根本上的差異。

2.4 Keywords System (KS)

所謂「關鍵字系統」(Keywords System)，語出 Y. Arrouye、T. W. Tan 及 X. Lee 於 IETF 提出的一項草案 [9]，總稱所有以關鍵字對應至網址的名稱服務。而「關鍵字」(Keyword) [9]、「實際名稱」(Real Name) [30, 31, 32] 或「一般名稱」(Common Name) [33]，常常指的是同一個概念，也就是直接使用人們在日常生活中對每個網路資源的稱呼，作為命名及識別的方式。一般來說，這樣的方式都會有一個不同於 IDN 架構的伺服器，以解析使用者輸入的關鍵字與 URI 之間的對應，而必須搶在現行 DNS 之前先作處理。其目的在於使用關鍵字來建構一個支援多國語言式查詢的，且對使用者更友善 (user-friendly) 的名稱服務。而 KS 的實作方式，常是以儲存了關鍵字和 URI 的對照表來進行比對，也就是第一章所提到的「一對一關聯」。但是當使用者輸入的網址名稱與其對照表中關鍵字不符時，又常出現轉而導向至搜尋引擎的作法，搖身一變為「一對多關聯」的情形。

由於這個狀況在實際應用上發生率太過頻繁，常給人名稱服務與搜尋引擎間分工混淆不清的印象。

現今世界上各個角落都有不同的組織提供了此類型的服務，最著名的是 Arrouye 所屬公司 RealNames Corporation (RealNames.com)，透過與 Microsoft 和 VeriSign Corporation (VeriSign.com) 的合作，號稱在 Internet Explorer 的網址列上直接支援全球各地的關鍵字解析；另外有幾個公司如美國的 AOL、netword.com 及英國的 CommonName.com 也支援自然語言式的英文名稱對應，但 AOL 只在其所屬的網路中提供這種服務；而表現出對 KS 需求最為強烈的，則是亞太地區使用非拼音語言的國家，計有中國大陸的北京因特國風公司 (3721.com)、韓國的 NetPia.com 及泰國的 nipa.co.th 等，其中 NetPia.com 支援中日韓三種語系，並與 nipa.co.th 有技術合作關係。

附代一提，Arrouye 等人在 IETF 發表的草案中，把 CNNIC 及 TWNIC 等單位所作的本土化 DNS 服務也歸類成一種 KS。然而以本論文的觀點看來，這些單位依舊與 DNS 糾纏不清的作法，與該草案中提出「在 DNS 及 IDN 之上的名稱服務層」有些矛盾，故不將其納入 KS 的範圍來闡述。

2.5 Common Name Resolution Protocol (CNRP)

與 KS 相似的是，CNRP 也是發表於 IETF 上的一個草案 [33]，且三位作者都有 KS 的背景：N. Popp 是 RealNames.com 的一員，M. Moseley 隸屬於

Netword.com，而 M. Mealling 則來自 VeriSign.com。事實上，CNRP 可視為增進 KS 能力的一種方式，以補足 KS 單靠關鍵字對應無法進一步取得更多資訊的缺憾，而在其實作上最積極的就是 VeriSign.com。

一個 Common Name 就是一種 Keyword，代表著人們對世界上各式各樣實體之名稱的認知，如商標、品牌等。而 CNRP 的作法，乃是利用 XML 定義了許多「屬性」(attribute)，如某公司的「地理位置」(geography)、某網站使用的語言，或是該網路資源屬於那個「分類」(category) 等資訊，作為額外的查詢條件，進一步收斂 KS 實際運作時可能由名稱服務轉至搜尋引擎而得到的多筆網址結果。於是，CNRP 的名稱服務，主要透過 XML 文件來進行，下面所展示就是以 ”Benz” 為網路名稱，並僅限位於美國之網站的查詢：

```
<query>
<commonname>Benz</commonname>
        <property name="language" type="rfc1766">
                US
        </property>
</query>
```

這個文件可以用 MIME 格式，由 HTTP 或 SMTP 來傳送。若要直接以 URI 的方式進行查詢，IETF 上已經有一個名為 ”go” 的 URI 架構草案 [34] 被提出來討論，實際上就是向提供 CNRP 解析服務的主機上埠號 1096 處傳送參數組合來進行查詢。以下是假設某支援 ”go” 架構的主機為 ”cnrp.foo.org” 時，與上例等效的查詢語法：go://cnrp.foo.org?Benz;geography=US

事實上，當這個協定以 Common Name 為名時，不禁讓人聯想到目錄服務中

的 CN，以及目錄服務透過每個節點的屬性來進行比對的方式。更進一步來說，如果有一個目錄中存放節點都是專為名稱服務而設計，CNRP 的效果將與 LDAP 十分類似，甚至可以說是名稱服務為導向，且使用 XML 或 “go” URI 來進行類似 LDAP 查詢語法的特殊目錄服務，意即 CNRP 可以選擇 LDAP 為實作方式。也因為這個特色，使得 CNRP 具備了與同樣在架構上類似 LDAP 之 Universal Discovery, Description and Integration (UDDI) [35] 相容的可能性，但這已超出一般性網際網路名稱服務的範圍，故不在此詳加討論。

2.6 總結

本章所簡介與名稱服務相關的幾個技術，皆有其缺漏之處。DNS 使用上因難懂難記造成的不便已毋需多提，而 IDN 除了尚未成熟外，又因其作法為更動 DNS 架構，不但在名稱語法上依然不夠直觀，還常與代理伺服器或防火牆衝突；URI 裡發展中的 URN 及 HFN，雖較 URL 具有更高的彈性，但其目標較偏向於處理複本問題，對名稱服務沒有太多直接的助益；而 KS 固然已提供了自然語言式的用法，卻受限於僅能作一對一關聯的設計；CNRP 雖能輔助 KS，但仍是一對一乃至於一對多式的查詢，且使用者必須額外了解其查詢語法，不易直接以日常生活中對名稱的認知來操作。

第三章 智慧型命名系統架構

3.1 系統架構概觀

套用 Ballintijn 等人 [12] 的分類觀點，本論文提出的架構比較接近 yellow-pages 機制所採用的目錄服務式作法。但本研究與一般透過目錄服務來完成之名稱服務的顯著不同處，在於採用「辭典」(Dictionary) 來回歸一般人使用習慣的方式。考慮到人們對於網址名稱的認知與運用，實際上是先有各種表現上的差異，才有後來的歸納整理乃至於分類管理。因此一般使用者不見得能夠直接理解目錄服務中各節點具備「屬性」的概念，但至少人人都可以在日常用語之於網址名稱上有共通的認知。於是本研究先利用辭典來應對使用者的詞彙，再透過辭典中為了名稱服務而預先分類的詞性與目錄式「知識庫」(Knowledge Base) 的屬性接軌。因此，本研究的實作在巨觀上可區別為提供服務的軟體系統與提供資訊的目錄系統兩大部分，其內部各自包含的元件如圖 3.1 所示，茲分述於下：

- **Knowledge Base**

  本系統為網址命名用途專門規劃了一個「知識庫」(Knowledge Base)，每一筆「記錄」 (record) 中都包含了網址 (URI)、註冊名稱、及描述該網址各項 metadata 的屬性，如圖 3.2 所示。基於管理便利性及快速比對的需求，註冊名稱在系統中將視為唯一 (unique) 的索引鍵 (key)；metadata 的數量視網址類型而定，通常可能的屬性有網址的「地域」(region) 、「組織類型」(organization)、「修飾詞」(qualifier) 及「稱號」(label) 等。本研究實

作上雖可視為一種目錄服務，但與一般目錄伺服器稍有不同的是，不必刻意地將其記錄以每個節點建構為樹狀結構，因為不見得每種網址名稱皆有階層的概念，必要時再以 metadata 描述即可；而在網址方面，現階段的系統中每一筆記錄只會擁有一個多半為 URL 形式的 URI。待日後 URN 標準發展成熟後，URI 本身就有能力自行處理複本 (replica) 的狀況時，再考慮這方面的實作問題。

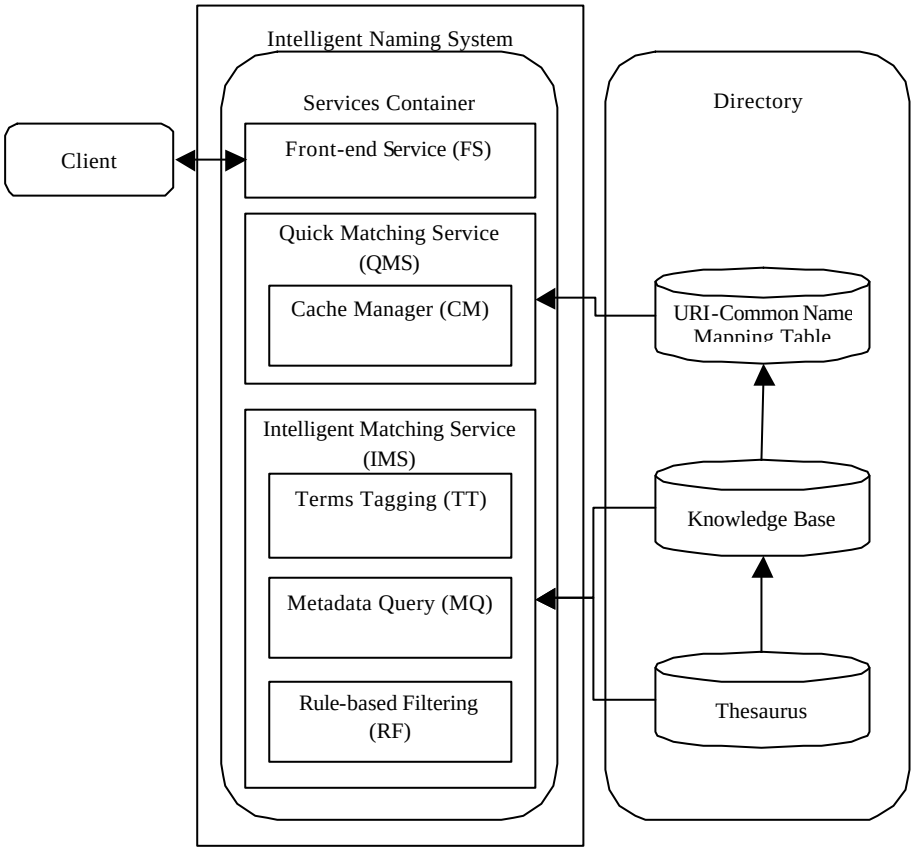

圖 3.1.系統架構圖

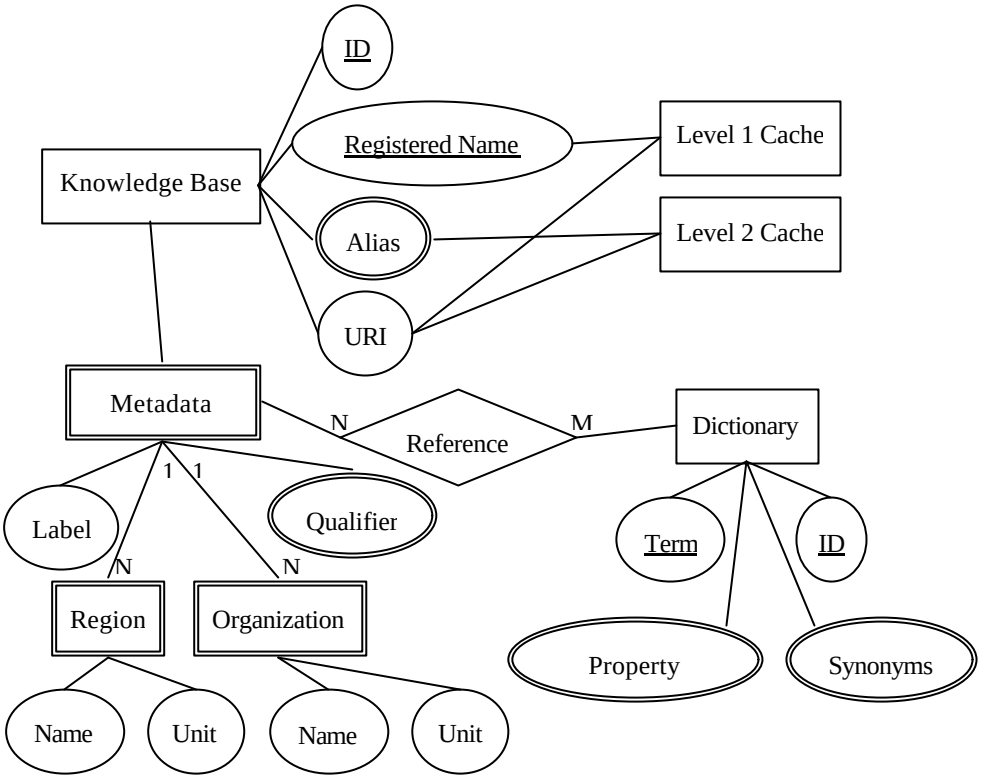

圖 3.2. Directory **的** schema

- **URI-Common Name Mapping Table**

  這個表在一開始僅有與 KS **的作法類似的，由知識庫中每筆記錄裡名稱與網址組成之「靜態」對應，儲存於** Level 1 Cache **中。之所以稱為** URI-Common Name **而非** URI-"Unique Name" **的** Mapping**，在於本名稱服務另有「動態」對應機制，能基於前次網址名稱的查詢結果及使用頻率分析出適當的「別名」**(alias)**，修改這個對照表的內容，放入** Level 2 Cache**。因此如圖** 3.2 **所示，本表涵括了兩個層次的快取，分別用在知識庫記錄中的「註冊名稱」**(Registered Name)**，及由使用紀錄取得的「別名」**(Alias) **上，於是將兩種用法總稱為** Common Name**。事實上，在一個對照表中常有「索**

引鍵」(Key) 的概念，用於這個兩層架構中，即可將註冊名稱可視為一種「主要索引鍵」(Primary Key)，而別名則身處「次要索引鍵」(Secondary Key)的地位。如此一來，這個對照表中將可能有許多不同表達方式的名稱，都指向同一個網址，直接支援了「多對一關聯」的特性。而此動態對應機制將靠著 Cache Manager 及 Intelligent Matching Service 來完成，後文另有說明。

- **Dictionary: a thesaurus**

  另一個與知識庫相關的元件是「辭典」(Dictionary)，而且是一個「同義辭庫」(Thesaurus) 形式的辭典。其中詞彙與詞性的關聯應與知識庫中 metadata 的屬性一致，亦可見於圖 3.2 中，換言之該辭典會隨著 metadata 的新增而可能有所成長；至於同義詞之間的關聯，則是以「同義詞集合」(Synonym Set, Synset) 的結構存在，必須由管理者人工維護。

- **Front-end Service (FS)**

  負責接收來自客戶端 (client) 的網址名稱，通常為瀏覽器網址列的輸入字串。由於各作業系統平台及瀏覽器的版本差異，造成網址名稱往往不只一種編碼格式，FS 便必須於此時針對接收到的字串加以標準化 (Normalize)，使其統一成系統內部運作時的格式。此刻這個標準化的過程不但具有統一編碼格式的功能，還得過濾掉常用於 HTTP 的「保留字元」(reserved character)。因此這個元件中對輸入字串真正進行名稱服務前的預

先處理，必須依照各種已知的狀況配套適應，將目前網路環境中各種命名系統及編碼方式混雜並行的情形區隔開來，使本系統能在後續的比對過程中作最有效的運用。另一方面，服務流程全部結束後，所有的傳回值依舊將統一歸向 FS，經由此元件針對不同類型之結果來作輸出方式的選擇。若系統決定出唯一的網址為最佳解答，則 FS 會自動使瀏覽器導向至該處；若系統傳回的結果為無解或多筆網址解答，則 FS 就輸出相應的網頁以向使用者說明其狀況，並能進一步提供建議，如重新其輸入字串導向至搜尋引擎，或是將此一狀況回報給系統紀錄，讓管理者能夠隨時查核並嘗試修正。

- **Quick Matching Service (QMS)**

  在名稱服務的實際工作上，首先會嘗試由 QMS 取得立即的結果。Quick Matching Service 顧名思義，強調的是快速地由名稱對應至網址的服務，事實上是與 KS 相當類似的概念，透過對 URI-Common Name Mapping Table 中名稱的比對，直接找出已註冊的網址。然而本系統在此一部分仍有著與一般靜態對應的 KS 不同之處，即透過 Cache Manager 進行的，依照使用率而改變的動態對應。

- **Cache Manager**

  Cache Manager 所「快取」的對象，除了原本已向系統註冊的名稱外，尚包括使用率較高但未直接註冊為名稱的用法。這些用法在初次查詢時，無

法立即由 QMS 得到解答，但在轉由下文將介紹的 Intelligent Matching Service 處理後，若能夠順利得到網址對應，即可「回饋」(feedback) 給 Cache Manager，由其決定是否將此一用法視為別名，加入知識庫與 Level 2 Cache 中，使該名稱下一次能夠進入 QMS 快速回應的範圍。

- **Intelligent Matching Service (IMS)**

  此子系統正如其名，為智慧型命名系統的核心功能之所在。凡是語義上相同的名稱，皆應在 IMS 處理後得到建立彼此關聯的可能性。而幫助 IMS 作到這一點的，正是以下將介紹的三項元件：

  - ✓ **Terms Tagging (TT)**

    所謂「詞彙標註」(Terms Tagging) 的過程，是將一字串先經過「斷詞」(Terms Segmenting) 處理，得到有意義的字元排列後，再將可能的屬性透過查詢「辭典」(Dictionary) 的記錄而「標註」(Tagging) 於每個詞之上。如此一來，原先的字串將轉變成每個詞彙與其詞性關聯的集合，換言之，可視為辭典本身的一個子集合，該子集合的元素來自與原字串中排列方式相同的詞彙。

    使用表意符號如漢字的語系在自然語言處理上，一向較使用拼音符號如羅馬字母的語系有著更高的難度。基本上最大的不同之處，在於表意語系通常不似拼音語系有利用空白來作為「分隔符號」(delimiter) 的習慣，因而使得斷詞過程往往無法精確地求得唯一解

答。以中文來說，常在不同的認知前提下，即有可能產生不同的斷詞方式，而且每一種皆言之成理，無法斷然決定誰對誰錯。例如「台中市政府」這樣不過短短五個字的名稱，便因為「市」一字既可作為「台中」的後置字元，也可作為「政府」的前置字元，產生了所謂「搶詞」的狀況，而出現｛「台中市」,「政府」｝及 ｛「台中」,「市政府」｝等至少兩種不同的斷詞結果。這樣的結果在人看來是顯而易見且可以並行不悖的，但在電腦上以程式處理時，就無法單純地靠字串比對來完成。

而一般在無分隔符號的字串斷詞方法上，最常見的兩大策略分別是詞頻統計法與辭典查詢法。由於本系統的其他元件也將需要辭庫的輔助來達到目的，因此未採用詞頻統計式的架構進行斷詞，其間的差異因不在本論文討論的範圍內，故於此點到為止。

事實上正因為斷詞問題有其模糊之處，也說明了人們對於網址名稱自有其認知的差異，而詞彙標註便成了之後進一步判斷的重要依據。考量到一般使用習慣中，網址名稱幾乎皆以名詞片語的形式展現，故此處的工作並不是單純地標註文法上的詞性，而是就其意義作屬性上的分類。

另一方面，為求更為全面地涵括一般人的使用習慣，本研究特別採用「同義辭庫」(Thesaurus) 作為詞彙標註時所用的辭典。同義辭庫

除了一般辭典結構上必定提供的詞彙與詞性間之對照關係外，尚維護了許多稱為「同義詞集合」(Synonym Set, Synset) 的關聯。藉著這些關聯，斷詞後得到的各個詞彙可被進一步地「擴展」(expand)，使得斷詞結果集合中將包含不屬於原字串的排列，但在語意上有一定程度關聯甚至相同的詞彙。藉著這個擴展過的詞集，將使得原先單純的字串比對式查詢開始有了「查詢擴展」(query expansion) 的特性。例如前面舉出的例子「台中市政府」，即有可能因為「市政府」在同義辭庫中可找到「市府」這樣的一個同義詞，使得將作為比對之用的詞集將由｛「台中」，「市政府」，「台中市」，「政府」｝擴展為｛「台中」，「市政府」，「台中市」，「政府」，「市府」｝。

- ✓ **Metadata Query (MQ)**

  有了經過詞彙標註的詞集，便可透過詞彙本身字串及其標記的屬性作一雙重的比對。換言之，必須是字串及屬性皆符合的才可被視為相等。而這個比對的範圍，則是儲存於本系統目錄結構之知識庫中的 metadata。所謂 metadata，最常見的定義是 ”data about data”，是一種以屬性為基礎 (attribute-based) 的輔助資訊。通常屬性呈現的方法不外乎「索引鍵－鍵值」的邏輯，也就是一組以既定的「鍵」(key) 為「索引」(index) 與任意字串為「值」(value) 的成對關係。

  回顧一個辭典的架構，將可發現其中詞彙與詞性則是以特徵為基礎

(property-based) 的結構。換言之，是由任意字串為索引鍵與既定詞類為鍵值的(property)「特」所組成。其思路方向恰好與由屬性組成的 metadata 相反。因此本研究在解析網址名稱時，便利用了這個特性作為比對條件。事實上，將這個概念反向操作，也可視為經由管理知識庫中 metadata 裡每一個屬性而間接維護了辭典中各個特徵，也就是各詞彙與其詞性關係的過程，而使得知識庫與辭典裡用於比對的資訊形式能夠一致。

- ✓ **Rule-based Filtering (RF)**

光靠著標記過的詞集向 metadata 進行比對所得的結果，與適切地處理自然語言表現在網址名稱上的行為仍有差距。一般來說，主要有兩種問題尚待解決，分別是網址名稱在「佈署」(Deploying) 性質及「語意」(Semantic) 性質上的判斷。為了解決不同性質的問題，就必須有因應的配套規則，於是本研究透過這個以規則為基礎 (rule-based) 的過濾器元件，來設法收斂出適當的結果。

佈署性質的問題常發生於使用者輸入的網址名稱缺乏地域性的資訊，而造成本服務面臨一個名稱可能有多筆網址候選時。其中尤以學校類型的網址名稱最容易產生這樣的狀況。例如在台灣地區有不只一個以「忠信」為名的小學，若使用者未能於輸入時即包含更多的詞彙，本服務便將需要一個規則來取捨過多的網址結果。

語意性質的問題則是一般目錄服務式服務之通病。假設目前「東海大學」除了校方主網址之外，尚註冊了所有校內學院學系等各單位的網址，而這每一筆網址的 metadata 都將包含了與「東海大學」相同的屬性，於是便形成了一個所有與東海大學相關的網址集合。但是實際上當使用者以「東海大學」為網址名稱時，期望中應只獲得校方主網址，畢竟若使用者需要東海大學校內其他單位的網址，應會在其輸入字串中包含這些單位名稱的詞彙。

3.2 服務機制

智慧型名稱服務之機制的互動過程，顯示於圖 3.3 與 3.4 的 sequence diagram 中。首先 FS 作為對外的窗口，處理輸入和輸出字串時，將面臨編碼方式及比對策略的抉擇。在實際進入名稱服務前，FS 會先執行各項字串的前置處理，然後自 QMS 起跑作網址名稱的比對。如圖 3.3 所示，這個部分乃是單純地進入 CM 所管理的對照表中尋求名稱完全符合的網址。若順利地取得解答，隨即傳回 FS 作輸出，否則便轉而交由 IMS 作更進一步的嘗試。

CM 作為 QMS 子系統的主要元件，接收到來自 FS 的資料後，將先查核搜尋關鍵詞列表，判斷是否為應進行服務的網址名稱，或是直接導向至搜尋引擎即可。確認之後，隨即進入第一層快取中進行比對。若順利取得相應的 URI，便直接轉向 FS 作輸出；要是無法馬上得到結果，還會嘗試使用異體字對照表，

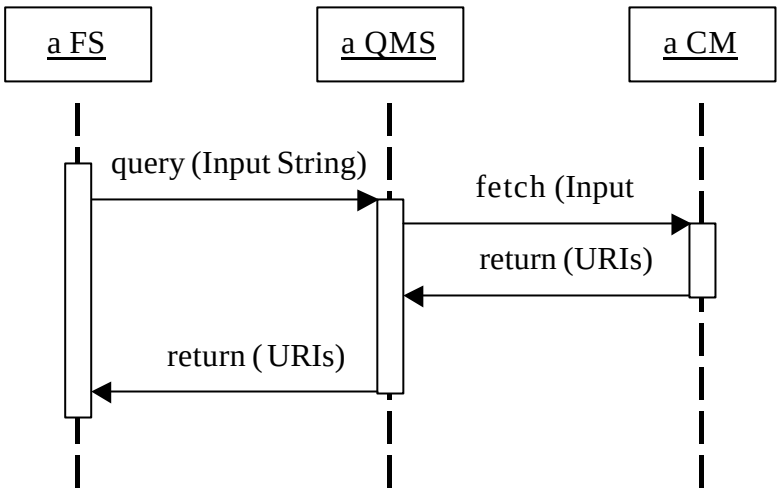

圖 3.3. Quick matching service sequence diagram

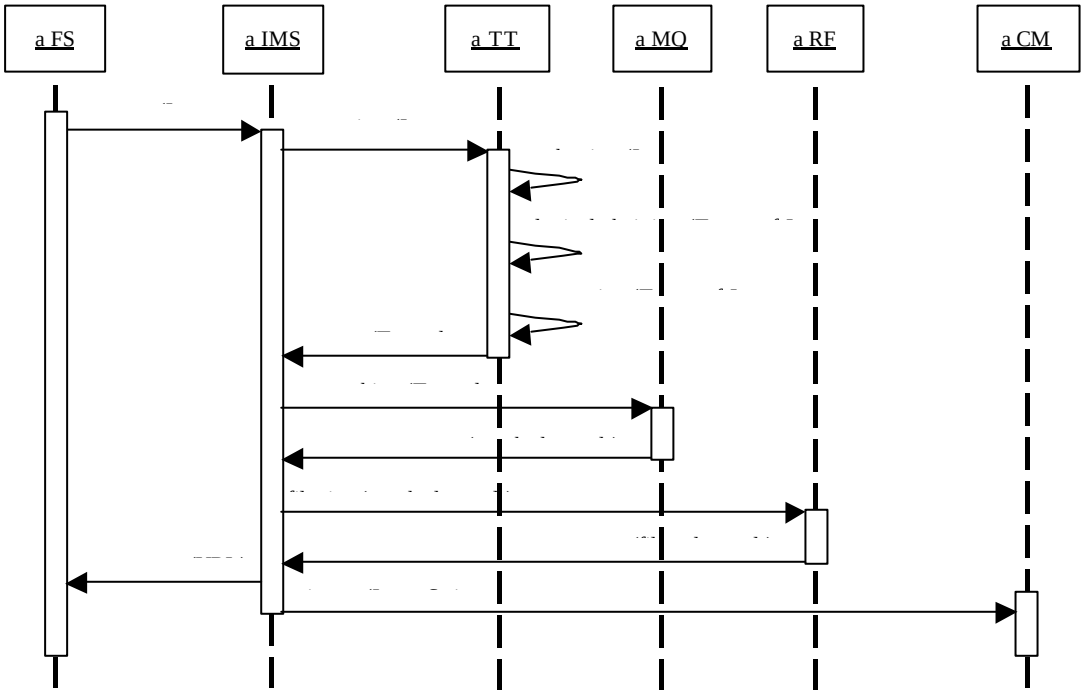

圖 3.4 Intelligent matching service sequence diagram

進行字元之間的代換。例如當使用者以「臺中市政府」為網址名稱時，將於此處代換為「台中市政府」。由於代換發生的機會不多，且不具語意檢查的性質，故不考慮超過一個字元的組合。例如「台証證券」這樣的名稱，「台」、「臺」及「證」、「証」都可互換，但實際上不見得每一種組合都有人使用，還不如將此問題留待 IMS 解決來得合理且有效。若異體字代換後仍未獲得結果，則進入第二層快取中，分別以原輸入字串及異體字代換過的字串尋求是否有相同的別名，視結果有無來選擇返回 FS 輸出或轉由 IMS 進行更深入的比對。第二層快取中的別名，來自於先前 IMS 成功比對出的，不同於第一層快取中註冊名稱用法的使用紀錄。以上的所有動作，都是由 Cache Manager 向 URI-Common Name Mapping Table 所「取得」(fetch)，但快取機制尚須與本系統中其他的部分互動，尤其第二層快取具備了透過 IMS 使用紀錄動態更新的特性，因此尚應提供基本的新增、修改及刪除等操作方法。

當進入 IMS 之後，已由 FS 預先處理過的字串會由 TT 按順序作「字元組排列」(tokenizing)、「語意鏈結」(lexical chaining) [13, 14, 38] 及「詞彙重組」(recognizing) 等動作，以完成詞彙標註的工作，整個流程如圖 3.4 所示，現分別說明如下:

- **Tokenizing**

  本研究預期使用者傾向於以名詞片語作為網址名稱，而且目前並未在自然語言的處理上深入至其他文法及容錯的層次。換言之斷詞時將只需要處理

「子字串」(sub-string) 即可，不必進一步檢查所謂的「子序列」(sub-sequence)，於是在字串處理演算法上就未加以深究。Tokenizing 的方式採用了最直觀的窮舉法，或者以其單位切割的行為而稱作 n-gram 方法，列出所有的子字串集合，便不在此特別詳細說明其演算法。

- **Lexical Chaining**

  語意鏈結的過程，首先即是將子字串集合的元素一一代入辭典，篩選出已知的詞彙，接著挑出每一個詞彙的同義詞集合，形成原字串內含詞彙與其同義詞的關聯。這個部分參考了 S. J. Green 等人 [38] 的研究，但並未直接採用其鏈結方式，而僅萃取了同義詞集合的概念，應用在詞彙的擴展上。

- **Recognizing**

  直接由詞彙及其同義詞的關聯所組成的詞集，不該直接全體用於接下來的 metadata 比對，而應依照其原始的詞彙及詞性建立代換關係，重組出適當的排列。通常這個代換關係除了原字串原位置上的詞彙可直接套用外，還有可能依照詞性之間特定的前後置詞關係來進行。針對詞性中組織與地名兩類，內部又可分為名稱及單位兩種，而這些屬性的詞彙之間必須維持相互依存、成對出現的狀況。例如前文中曾舉出「台中市政府」擴展過後的詞集為 {「台中」,「市政府」,「台中市」,「政府」,「市府」}，雖然「市府」一詞是來自「市政府」的同義詞，但在這個詞集中,「市府」與「台中市」各自隸屬的詞性並不相互排斥，故最後可作為 metadata 比對用途

的詞集排列方式將是｛「台中」,「市政府」｝、｛「台中市」,「政府」｝、｛「台中」,「市府」｝及｛「台中市」,「市府」｝等四種。

- **Matching**

  有了能進行查詢用途的詞集排列之後，便能開始利用其中的詞彙及標註的詞性，成對地作為一種特徵集合，與知識庫中 metadata 的屬性集合作比較，找出兩相符合的記錄。這個動作實際上為先比較詞彙的異同，再比對詞性與屬性是否符合。之所以有這樣的順序，是考量到詞彙相同但屬性全然不符的特殊狀況，以維持之後仍能作判斷的機會。如網址名稱若為「台中大學」，則經 Recognizing 得出的詞彙與詞性排列應為｛「台中」：組織名稱；「大學」：組織單位｝，假設知識庫中沒有一筆記錄的 metadata 有「台中」作為組織名稱而「大學」視為組織單位的屬性，將無法找出解答；但知識庫中可能有許多組織單位屬性為「大學」同時地域名稱屬性為「台中」的記錄，則最低限度仍可猜測出使用者需要網址的應為位於「台中」的大學而非名叫「台中」的大學。

- **Filtering**

  這個流程裡進行的是佈署規則或語意規則兩種不同策略的選擇，在過濾完畢之後，才將結果傳回 FS，由 FS 統整輸出。實際上佈署規則套用的方式，須藉由客戶端提供的地域資訊作輔助。這個動作與詞彙標註中詞彙重組後產生的結果相似，但僅限於地域屬性的詞彙，且只作用於 MQ 取得的結

果過多時，例如當使用者要求名稱為「忠信國小」的網址時，將取得不只一筆的結果，則此時應加上某些地域詞彙形成如「台中忠信國小」來加以修正。實作上除了被動地期望使用者能充分提供此類輔助資訊外，也可主動地依照客戶端的來源或設定等方式猜測；語意規則靠的是回頭再檢視已符合記錄的 metadata，若其中尚未用於比對的資料仍有組織屬性，則表示此筆紀錄的意義不止使用者輸入之詞彙所能提供的部分，便應於此處篩去，一如前文中已舉出的東海大學校方主網址與其下單位網址的關係。另一方面，當此類狀況造成解答數目在嘗試過濾後仍多於三十筆時，系統將進一步認定此輸入缺乏作為名稱查詢的必要資訊，而轉為增加 Cache Manager 中搜尋關鍵詞列表的紀錄，並讓該輸入字串由本次流程開始即變為導向至搜尋引擎的類型。

IMS 工作完成回到 FS 中後，尚需選擇輸出時的策略，由查詢結果的數量，來決定使用單筆輸出或多筆輸出的方式。雖然結果數量無論為零、一或多筆，在邏輯上可共用一種輸出方式，但透過 HTTP 傳回資料的實作方法不只一種。基於應用上的考量，各別處理不單能針對傳輸效能作最佳化，尚可以「範本」(template) 方便輸出多筆網址頁面的套用。FS 選擇了輸出方式後，會順便將此次服務流程留下紀錄，並結束整個名稱服務的比對機制。

3.3 總結

本研究的實作以 FS 為出入口，處理過輸入資料後，在內部先經 QMS 查詢，若未能立即取得 Cache 中的直接對應，則進一步至 IMS 中，以類似 yellow-pages 架構的知識庫中之 metadata 和同義辭典為基礎，和經過詞彙標註的名稱作比對，必要時並套用佈署規則或語意規則，判斷是否有適合的網址，再回到 FS 選擇輸出方式並留下紀錄，完成智慧型名稱服務的流程。

第四章 名稱服務使用行為分析

本章的第一部分將由客戶端透過瀏覽器外掛程式使用網址名稱服務的角度，展示本系統使用上的範例劇情。該劇情中將分別介紹常用網址名稱的類型，及其各自對應的結果輸出狀況。第二部分則是依據本研究實際於網際網路上公開運作八個月後所收集的「紀錄」(log)，分析使用者對於名稱服務的使用行為。同時根據此分析結果，嘗試驗證本研究的優劣，以進一步找出可能改善的空間。

4.1 範例劇情

本研究搭配了一個為 Internet Explorer 設計的外掛程式。當此外掛程式安裝完畢後，Internet Explorer 上除了網址列外，將出現另一個「名稱列」。該名稱列最一般的用法，就是直接輸入使用者對某個網址名稱的認知，以知名入口網站「奇摩」為例，效果如圖 4.1 所示：

圖 4.1. 瀏覽器外掛程式

本系統除了上述用法之外，尚提供了幾種便利的設計，讓使用者在進行網

路瀏覽時，只需透過此名稱列作為單一窗口即可。使用者可以「英文簡稱」如 ”kimo” 來查詢，節省打字的時間；或者當需要某單位或個人之電子郵件時，則可在名稱前加上前置字元 “@”，如「@奇摩」就會傳回奇摩站的客服信箱位址；如果使用者不想進入名稱服務，而想對某關鍵字進行搜尋，也可在詞彙前加上 ”?” 表示，換言之於名稱列輸入「?奇摩」時，會自動導向至搜尋引擎，列出文件內容與「奇摩」一詞相似的網址結果集，而非名為「奇摩」的網址。同時，名稱列依然保有與原網址列相容的行為，因此使用者仍可在名稱列上輸入一般的 URI 如 ”http://tw.yahoo.com/” 或 TWNIC 支援的中文域名「奇摩.商業.tw」，毋需為了牽就原本的使用習慣而回到網址列上輸入。

本研究成果使用上最大的特色，即在於容許使用者對同一網路資源有概念相同但表達方式不同的名稱。因此針對 ”http://tw.yahoo.com/”，除了可以「奇摩」為名之外，尚可以「雅虎奇摩」稱之，甚至是「奇摩站」或「台灣雅虎」皆可。再加上其英文簡稱 ”kimo”，就有了數量相當可觀的多對一關係，然而實際註冊的記錄只有一筆，其名稱可能僅為「奇摩」而已。

當使用者輸入了某些可能產生混淆的名稱時，系統將帶出一個多筆網址選擇的頁面，提示使用者進一步挑撿出其心目中真正期待的結果。舉例來說，若有使用者以 “yahoo” 或「雅虎」為名，期望能找到 ”http://tw.yahoo.com/”，系統卻因為比對得知，在中文繁體網站中還有一個 ”http://chinese.yahoo.com/” 與此名稱有所關聯，需更進一步的資訊來加以釐清，便將此兩筆結果皆列示於同一頁

圖 4.2. 多筆網址選擇頁

面，以其註冊名稱為提示，供使用者點選，如圖 4.2 所示。另外，本系統認定一個詞彙若擁有與其關聯的網址超過三十筆，則將該詞彙視為搜尋關鍵詞而導向至搜尋引擎，於是這個多筆網址選擇頁面便不會用於顯示超過三十筆紀錄的情況。

若使用者輸入的名稱尚無法為系統所辨識，其可能性主要有二：一是本系統知識庫中無該筆記錄，通常是未能發現並註冊，或是該名稱雖然存在於現實世界，但於網路環境中尚無對應；一則為該詞彙尚不在本系統辭典已知範圍內，通常是過於一般且簡短的，實際上應視為搜尋關鍵詞的用法。因此本系統在無法提供使用者解答時，將有進入錯誤回報流程及轉交搜尋引擎兩種策略，由使用者自行判斷。但其預設行為以視為搜尋關鍵詞為主，讓使用者在無法一次取得解答的狀況下，仍能夠快速地進入搜尋引擎確認其他的可能性。如圖 5.3 所示。

圖 4.3. 網址無法辨認頁

4.2 使用行為分析

本研究的實作系統已公開於網路上運作了將近八個月，累積了約十萬的使用人次，每一次都經由本系統紀錄了輸入及輸出的對應狀況。這一節將以此數據為樣本作歸納分析，藉此驗證本研究的成果及提出對網址名稱使用的建議。

首先，圖 4.4 說明了名稱服務的命中率。最初的數據顯示其效果不盡理想，在進一步分析回覆結果為無對應的狀況後，發現其中有超過一半的使用方式，是針對「非網址名稱」—即第一章所述之針對「網頁內容」，而非針對「網址」所進行的查詢，如圖 4.5 所示。換言之，該名稱在 Internet 上並沒有與之對應的網址存在，僅可能出現於某些網頁的內容之中。而本論文於第一章釐清名稱服務與搜尋引擎的分野時，即已言明凡是與網頁內容相關的查詢，應屬於搜尋引擎的工作範圍。因此，本研究於系統實作上，會將此類問題直接導向給搜尋引擎解決。

而在過濾掉這類「非網址名稱」的使用人次後，本系統的命中率便可超過八成，堪稱效果良好，見圖 4.6。同時在接下來的分析之中，也將以預先過濾掉非網址名稱的查詢人次為樣本。除此之外，將非網址名稱的查詢導向至搜尋引擎，還有降低區域網路流量的附加價值。假設每一次輸入查詢或輸出回應的網路流量負載值為 n，則每次使用搜尋引擎取得網址至點選後網頁顯示完畢，需要輸入及輸出各兩次，等於 4n。而使用本系統若能直接取得網址，只需要 2n；若經過導向至搜尋引擎，便需要 3n。因此總合起來約為 $2n \times 80\% + 3n \times 20\% = 2.2n$，換言之將減少 $2.2/4 = 55\%$ 的區域網路流量負載。

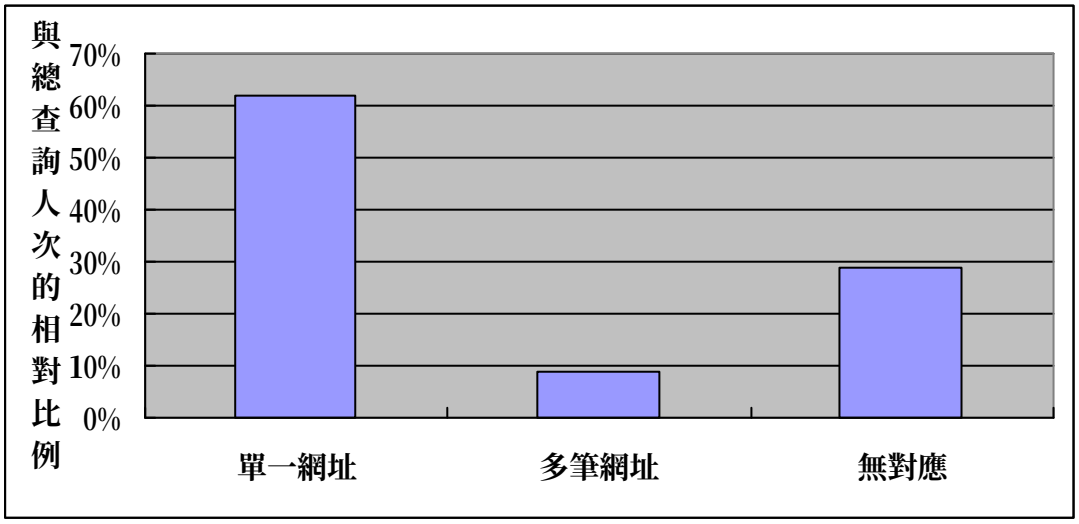

圖 4.4. 原始命中率

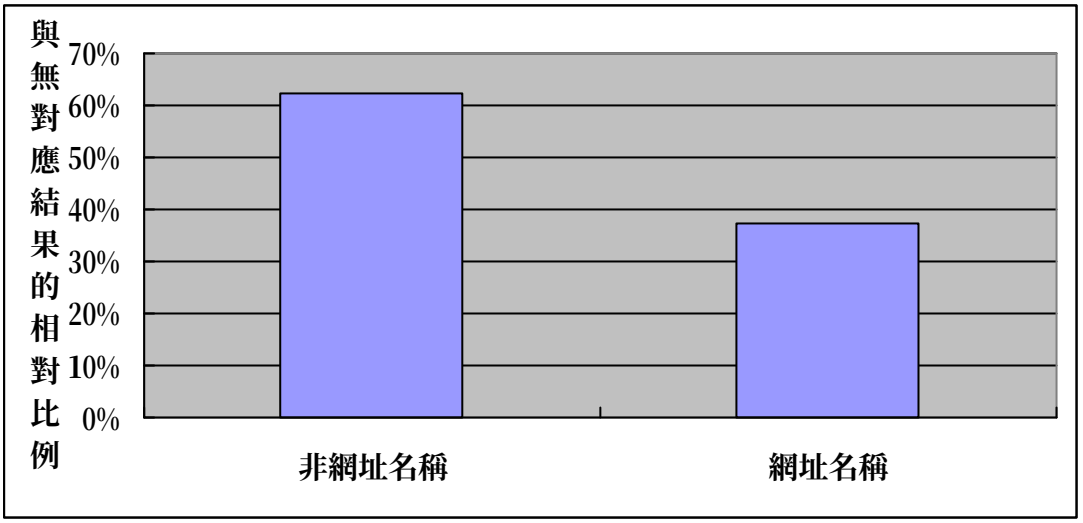

圖 4.5. 無對應結果中非網址名稱與網址名稱的相對比例

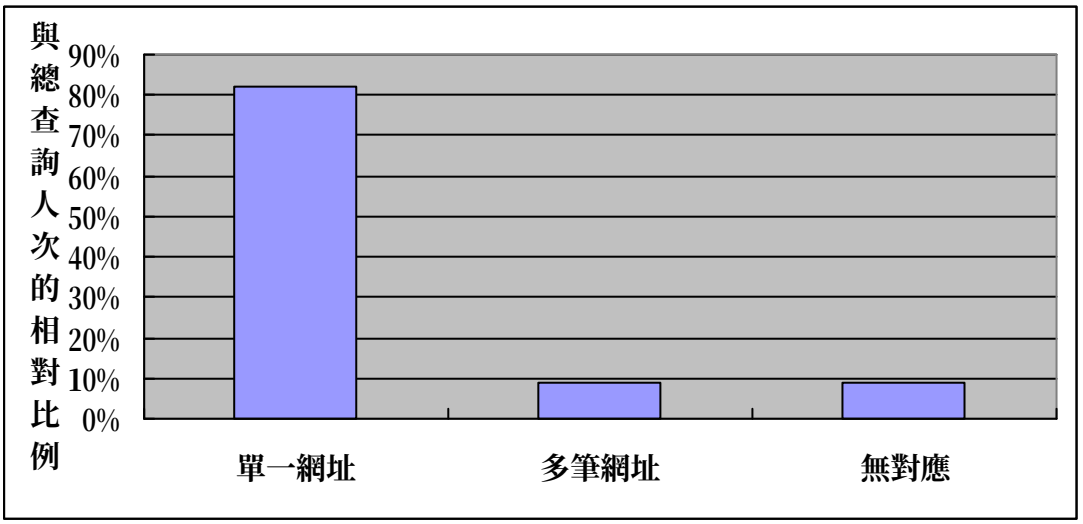

圖 4.6. 網址名稱命中率

接著是不同類型之網址名稱的與總查詢人次的相對比例之分析。由圖 4.7 可以看出，一般人最感興趣者常是公司之類商業組織的網址；值得注意的是，查詢人名的次數佔了將近五分之一，顯示人們在使用名稱服務時，直接以人名來指稱的需求也相當強烈。其他類型則代表了公司、學校或政府單位以外的組織名稱，

以及人名之外的專有名詞，例如前者可能為醫院，後者可能為書名等。從這個佔有率看來，使用者對於公司、人名、學校及政府單位的需求總和起來已超過 90%，因此若要提昇本服務的品質，便應針對這幾類網址的名稱使用行為作進一步的探討，尤其是在公司及人名兩類上，以找出改善的方式。

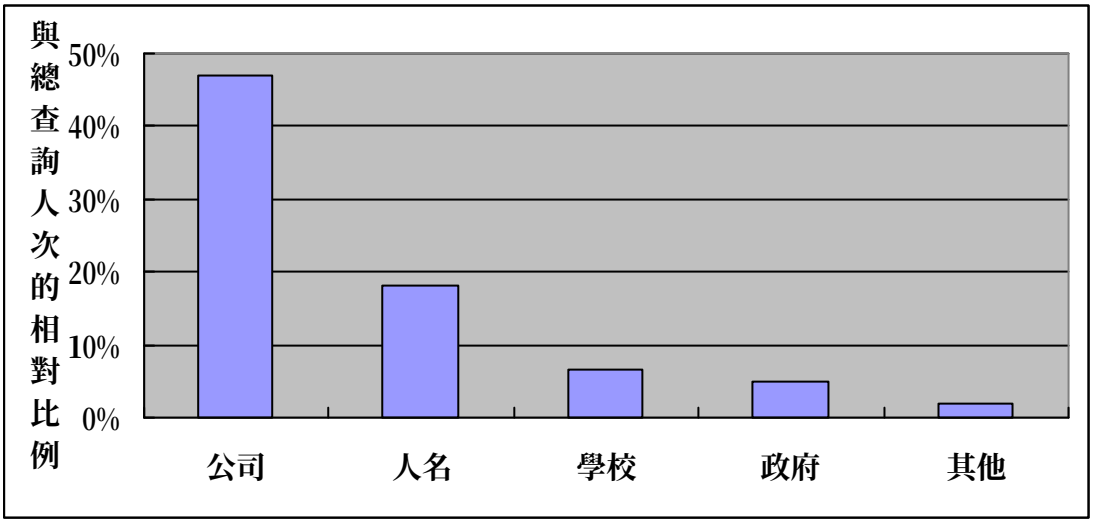

圖 4.7. 各類型網址名稱與總查詢人次的相對比例

首先從回覆多筆網址結果的情形切入。參照圖 4.8，最容易發生此狀況的是學校類型的網址名稱，例如不同地區的中小學，其名稱便常有重覆的現象。若配合圖 4.7 的使用率來看，因同名而造成混淆的情況中，應優先考慮解決方案的除了學校之外，還應包括人名類型的網址名稱，畢竟個人姓名也無可避免地有相同的可能。

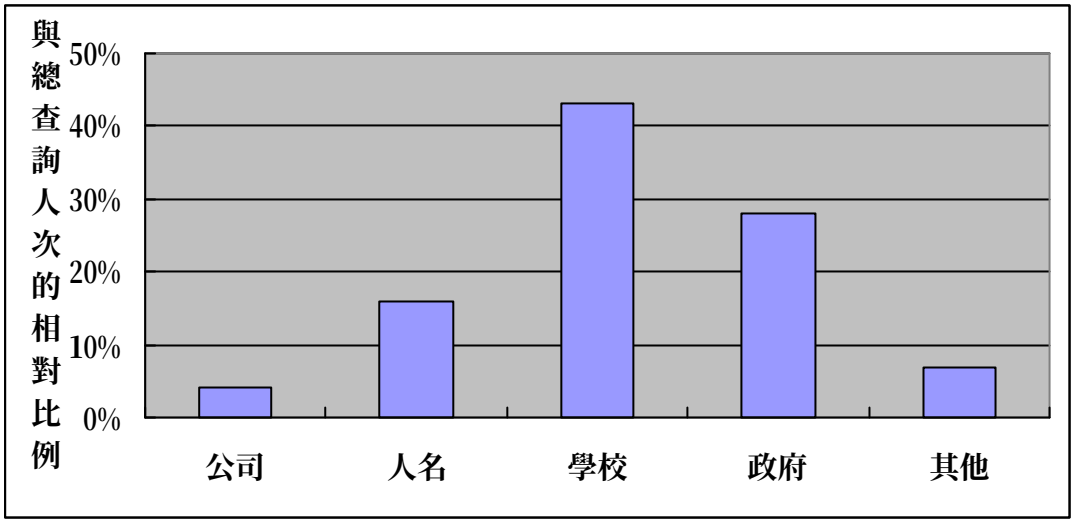

圖 4.8. 回覆多筆網址結果的佔總查詢人次的相對比例

接著針對上述狀況，本研究進一步作了一些統計。圖 4.9 列出了原本可能發生混淆，但加上地名作為修飾詞後，順利得到正確單一網址的比例。這個統計的作法，是以圖 4.8 樣本中出現的名稱為主，先取出所有子字串符合的查詢，再比較其中有地名為前置詞的查詢直接命中的比例。舉例來說，圖 4.8 的資料中曾出現「中正國小」這樣的名稱，則樣本就為所有「中正國小」加上前置詞的查詢，如「台中市中正國小」、「台北市中正國小」等，再計算這些樣本中能夠取得單一網址的名稱所佔之比例。由這個統計的結果看來，對學校及政府類型名稱而言，以地名釐清其對象相當有用；其他類型中包含的醫院等組織名稱多半有地域性，於是地名修飾詞也有不錯的效果；公司的名稱，除非分公司或連鎖店有獨立網址，地名常是非必要的資訊，故其獲得正確單一網址的相對比例最低；至於同樣也很需要修飾詞以降低同名機率的人名，改善則相當有限。因為無論以出生地或

居住地作為人名的修飾詞，仍可能出現同名者，或是經常變動而不適合作為固定名稱的一部分。於是，搭配此類名稱的修飾詞，應為地名以外的詞彙，如個人的暱稱、職稱等相關資訊。

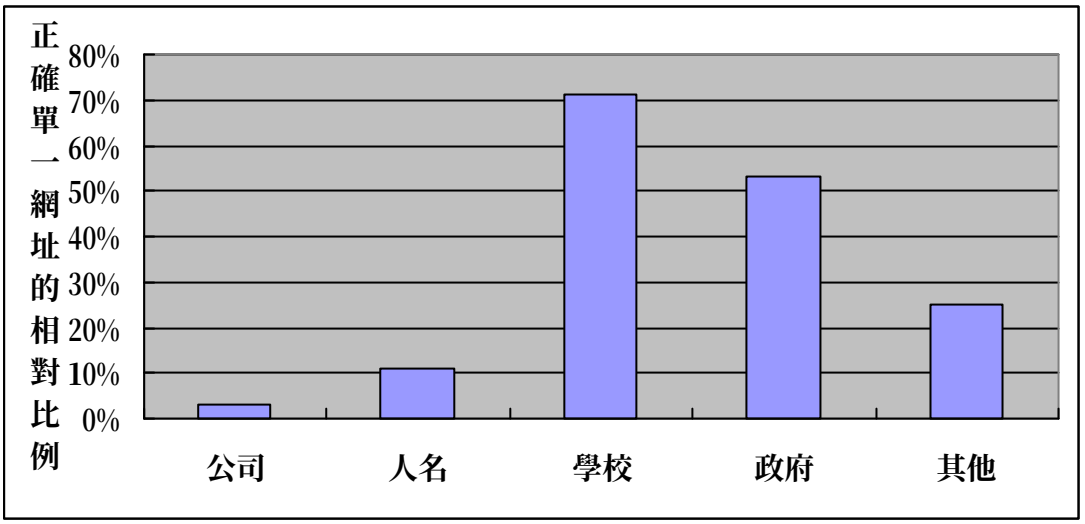

圖 4.9. 加上地名為修飾詞後可獲得單一正確網址的比例

基於以上的分析結果，為了讓網址名稱的使用更有效，本研究傾向於建議使用者增加修飾詞來進一步界定名稱的意義。以高中以下的各級學校為例，往往是在不同地區，其名稱才較有相同的可能，故使用者應可增加地域類型的詞彙作為修飾；同樣地，針對個人姓名，也應考慮使用可能的修飾詞來作為完整的網址名稱。因此，當使用者對名稱服務註冊或查詢時，以修飾詞和名詞成對的組合，會比單獨使用名詞來得精確。於是，接下來得面對的，將是如何找出最適當的修飾詞與名詞之組合形式。

為探究上述問題，可參考圖 4.10 中兩組數據的互動。長條代表的是名稱所

用詞彙長度與總查詢人次的相對比例；折線顯示的為詞長與命中率的關係。由長條圖可知，一般人在應用網址名稱時，不願花太多時間輸入較長的詞彙，可接受的範圍約為五字以下。而由折線圖走向來說，太短的名稱命中率不高，通常以四字的詞彙效果最佳。於是整體看來，本研究建議在使用網址名稱時，無論是註冊或查詢，皆應考慮以長度為二到五字之間的詞彙來命名。將這個結果與上一段的分析齊觀之便可發現，基於使用中文的前提，詞彙長度限於二到五字以內時，等於要求網址名稱應為一個修飾詞搭配一個名詞的片語，才能在使用上的便利性與正確性之間取得最佳平衡，進而充分發揮名稱服務的效益。

不過這個建議有一例外狀況，即針對某特定時間內某特定事件而產生的網址名稱。以近幾日來 (2002 年 7 月下旬) 電視廣告「菲玲」遭禁播而轉戰網路的例子來說，該詞彙原本不存在於生活中，自然不會有人用於名稱服務，同時也不符

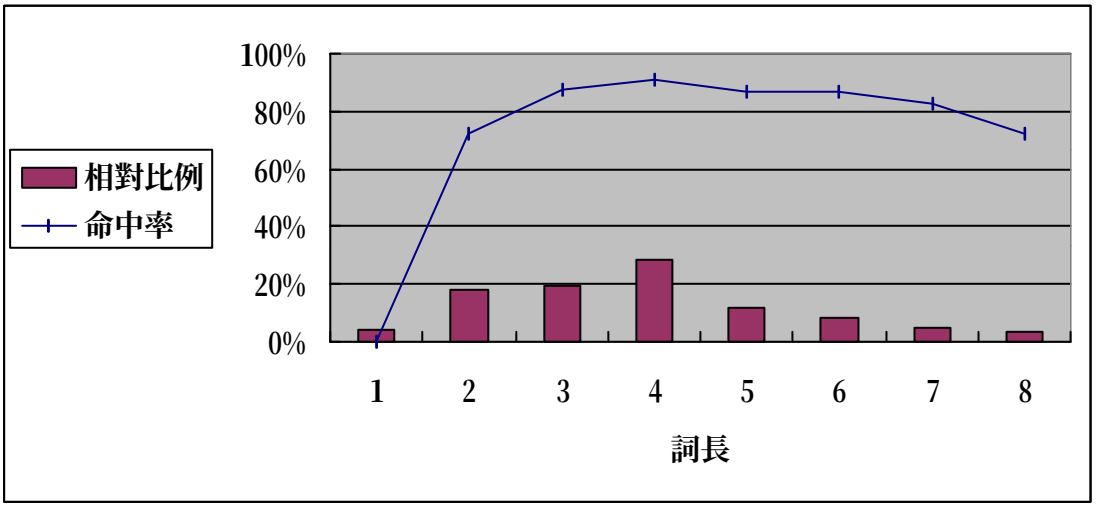

圖 4.10. 詞長之於所有查詢中相對比例和命中率的關係

合前面建議的網址名稱形式。但在這個「標語」(slogan) 快速傳播而廣為人知後，

其對應的網址變產生了大量的需求，迫使名稱服務必須針對這個詞彙有正確的回覆結果。不過從本系統的紀錄看來，這個網址名稱的使用頻率將有於短期內大起大落的特質，換言之在一段時日後，該網址可能已不存在，而名稱也將乏人問津，則名稱服務便可不再處理該詞彙的對應關係。於是，針對這個例外狀況，本研究將之歸納為「事件」(event) 類型的網址名稱，以與一般網址名稱之建議形式作區別。其特在於有時效性，且沒有固定的表達方式，動加以管理。

另一方面，本系統紀錄中顯示，由修飾詞與名詞組合成的網址名稱，當字數為五字以上時，亦出現了包含介系詞「的」、「之」等用法。而目前除了直接註冊為網址名稱時可運作正常外，若要進一步透過 Intelligent Matching 的機制來查詢，便因為系統尚未能處理這種形式而導致回覆無對應網址的結果。然而就中文的習慣來說，修飾詞常是作「限制」或「形容」用途的詞彙，以介系詞「的」為例，意義上就可能是以所有格形式作限制像「某人『的』網址」之用法，或是作為形容詞如「網址『的』名稱」之一部分。目前這類型的網址名稱在本研究提出的架構中尚未能有效地處理，僅以例外狀況對待而個別處理之。當然，若依照前面提出的建議來使用網址名稱，則鮮少有機會碰上這個問題。

4.3 總結

有鑑於上述分析結果，在自動導向了非網址名稱的查詢至搜尋引擎後，已知本研究在實作上有超過 80% 的命中率，成效良好；而一般人在網址名稱的使用上，最感興趣的前一二名分別是公司及人名；至於學校或政府的網址名稱，則常出現多筆網址結果的情況，若再考量使用比例，最需要解決此問題的是學校與人名類型；為避免因網址名稱過短而回覆多筆網址結果，使用者應於輸入時加上修飾詞如地名或暱稱等，則多數的同名混淆問題皆可獲得良好的解決；另外基於詞長對命中率和佔有率的關係，提示了應盡量使用二至五字內的詞彙作為網址名稱的建議。綜合以上兩個現象，本研究建議使用者無論是對網址名稱進行註冊或查詢，皆應以一個修飾詞加上一個名詞組合而成的片語為原則。但透過某些廣告標語的使用行為可知，若網址名稱屬於事件類型則不在此限。另外，五字以上的名稱常內含介系詞，該語法目前將需要個別處理。

第五章 結論及未來展望

5.1 結論

本論文針對現存的及發展中的各種名稱服務及相關網際網路標準作了一番比較，並針對其在於使用者親和力方面的缺失之處，提出一智慧型的名稱服務以求改進。本研究讓使用者能以符合自然語言習慣的名稱來對應網址，透過辭典和知識庫的配合，形成一個智慧型比對機制，允許較其他已知的名稱服務更具彈性的名稱表達方式。使用者僅需以其對某網路資源的認知進行命名，在概念上及語意上相同的名稱即可取得唯一的對應網址，而毋需與註冊的名稱完全符合，這個設計使得名稱與網址之間獲致更為理想的「多對一」式關聯對應。

而本研究的實作完成後，由其紀錄中的使用行為分析得知，已有 80% 以上的直接命中率，驗證了本研究成果的確有效；同時也進一步歸納出了最適當的網址名稱形式，應是一個修飾詞搭配一個名詞的組合，其例外則為事件類型的網址名稱。

此外，由於系統內部也判斷了以網頁內容為目標之搜尋關鍵詞，與以網址為目標之名稱的不同，適時地自動轉向重導使用者的查詢至搜尋引擎，減少了使用者在反覆尋求解答時所花費的點閱時間及區域網路對外的流量。這個將網路瀏覽行為整合至單一窗口服務的特色，成為本研究的一項附加價值。

5.2 未來展望

本論文提出的系統架構已內含了一般目錄服務的作法，因此應能進一步與未來可能成為標準之一的 CNRP [31, 33, 34] 接軌，同時待 URN 及 IDN 技術成熟之後，也可直接朝該方向相容，更進一步來看，本系統也應有潛力輔助 UDDI [35] 的應用。

此外，也可考慮提供更多不以瀏覽器為介面的服務，如現在相當流行的 Instant Message 軟體 ICQ、MSN Messenger 等，讓這些軟體的使用者能夠免於記憶數字帳號之苦。更有甚者，除了一般「桌上型電腦」(desktop) 所連結的網際網路環境之外，命名服務的需求也存在於「行動式網路」(mobile web) [39] 相關設備如行動電話、PDA 的使用上，及更為生活化的應用如語音及 set-top box 的領域中。這些類型的服務將原先本論文致力於「自然語言」式查詢輸入的目標，進而提昇至「自然的使用者介面」(natural user interface) 的層次。另外，由第四章的使用分析中得知，本研究尚應進一步探討名詞片語以外的名稱用法，尤其是當字數超過五字時最易出現的介系詞問題，以提昇系統處理不同於本研究建議之網址名稱形式時的能力。

參考文獻

[1] “About The World Wide Web”, World Wide Web Consortium, http://www.w3.org/WWW/.

[2] “Web Naming and Addressing Overview, URIs, URLs, …”, World Wide Web Consortium, http://www.w3.org/Addressing/.

[3] P. Mockapetris, “Domain Names – Concepts and Facilities”, RFC 1034, IETF, November 1983, http://www.ietf.org/rfc/rfc1034.txt.

[4] Internet Engineering Task Force, http://www.ietf.org/.

[5] E. Z. Wenzel and J. Seng, “Requirements of Internationalized Domain Names”, IETF IDN Working Group, 2001, http://www.i-d-n.net/draft/draft-ietf-idn-requirements-08.txt.

[6] “**中文通用域名系統**”, **中國互聯網絡信息中心**, http://www.cnnic.net.cn/.

[7] “**網域名稱及技術推廣服務專區**”, **台灣網路資訊中心**, http://cdns.twnic.net.tw/.

[8] Korea Network Information Center, http://www.nic.or.kr/.

[9] “Keywords Systems - Definition and Requirements”, http://search.ietf.org/internet-drafts/draft-arrouye-keywords-reqs-01.txt.

[10] “Google **搜尋建議：好手氣鍵**“, http://www.google.com/intl/zh-TW/help.html#C

[11] J. Allan, “Building Hypertext Using Information Retrieval” Information Processing and Management, Volume: 33, No. 2, 1997, Pages: 145-159

[12] G. Ballintijn, M. Steen, and A. S. Tanenbaum, “Scalable Human-Friendly Resource Names”, IEEE Internet Computing, September-October 2001, Pages: 20-27

[13] J. Morris and G. Hirst, “Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text”, Computational Linguistics, Volume: 17,

No. 1, 1991, Pages: 21-48, 1991

[14] G. Hirst and D. St-Onge, “Lexical Chains as Presentations of Context for the Detection and Correction of Malapropisms”, WordNet, The MIT Press, 1995

[15] **點名簿**, http://www.cardpen.com/

[16] J. Y. Lee, K. C. Hwang, and K. H. Lee, “The Reconstruct of DNS for Korean Domain”, Proceedings of the IEEE Region 10 Conference, Volume: 1, 1999, Pages: 206 -209

[17] J. Y. Lee, K. C. Hwang, and K. H. Lee, “A Dynamic Hangul E-Mail Address Management Protocol”, Proceedings of the IEEE Region 10 Conference, Volume: 1, 1999, Pages: 210 -213

[18] J. Y. Lee, K. C. Hwang, and K. H. Lee, “The Extension of Internet Domain Name System f or Korean Domain”, IEEE Parallel Processing, 1999, Pages: 214 -219

[19] “Simple ASCII Compatible Encoding (SACE)”, http://www.i-d-n.net/draft/draft-ietf-idn-sace-00.txt

[20] “RACE: Row-based ASCII Compatible Encoding for IDN”, http://www.i-d-n.net/draft/draft-ietf-idn-race-03.txt

[21] IETF IDN Working Group, http://www.i-d-n.net/

[22] P. C. Wu, “Using Plain Base32 ASCII-Compatible Encoding in the Local Part of E-mail Addresses”, IEEE Proceedings of the 2002 Symposium on Applications and the Internet (SAINT.02), Pages: 214-219

[23] “Cool URIs don’ t change”, http://www.w3.org/Provider/Style/URI.html

[24] “Functional Requirements for Uniform Resource Names”, http://www.ietf.org/rfc/rfc1737.txt

[25] Digital Object Identifier, http://www.doi.org/

[26] “URN Syntax”, http://www.ietf.org/rfc/rfc2141.txt

[27] Persistent URL, http://purl.oclc.org/

[28] Handle System, http://www.handle.net/

[29] W. Yeong, T. Howes, and S. Kille, “Lightweight Directory Access Protocol”, RFC 1777, IETF, March 1995, http://www.ietf.org/rfc/rfc1777.txt

[30] K. Teare, “A Briefing Paper on Next Generation Naming Services“, Briefing Paper for ICANN Meeting/June 1-4, 2001

[31] Y. Arrouye, V. Parikh, N. Popp, and K. Teare, “Internet Naming for the Next Generation”, RealNames Corporation, September 6, 2001

[32] Y. Arrouye, “The RealNames System —An International Human-Friendly Web Navigation System”, 16th International Unicode Conference, March 2000

[33] “Common Name Resolution Protocol (CNRP)”, http://search.ietf.org/internet-drafts/draft-ietf-cnrp-12.txt

[34] “The 'go' URI Scheme for the Common Name Resolution Protocol”, http://search.ietf.org/internet-drafts/draft-ietf-cnrp-uri-07.txt

[35] “UDDI and RealNames Keyword Technology”, Microsoft and RealNames Corporation

[36] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, “Design Patterns”, Addison-Wesley, 1995

[37] Mark Grand, “Patterns in Java”, Wiley Computer Publishing, Volume: 1 and 2, 1999

[38] S. J. Green, “Building Hypertext Links By Computing Semantic Similarity”, IEEE Transactions on Knowledge and Data Engineering, Volume: 11, No. 5, September-October 1999, Pages: 713-730

[39] N. Popp, “Using Keyword Technology to Redefine Addressing and Navigation on the Mobile Web”, RealNames Corporation, 2001