私立東海大學資訊工程與科學研究所

碩士論文

指導教授：羅文聰 博士 (Dr. Winston Lo)

以目錄服務為基礎的企業資訊系統架構

Enterprise Information System Architecture Based on LDAP Directory Services

研究生：黃一民 (Yimin Huang)

中 華 民 國 九 十 三 年 七 月

摘　要

由於目前企業所建置的各項網路服務與資訊應用系統未能有效整合使用者帳號密碼，形成許多各自為政的資訊孤島，本論文提供一個以目錄服務為基礎的企業資訊系統架構，透過目錄服務技術與 XML 標準，提供一個多種目錄、應用程式資料交換與同步的方法，達到使用者身分識別資料整合的服務要求，以建構易於使用、具備安全且管理方便的企業資訊運作環境。

目錄服務發展至今，已是一個相當成熟、穩定的技術，它可用來作為使用者身份驗證、系統使用授權及建置企業部門、人員與所有網路資源關係目錄樹等相關應用。因此，如何建置企業完整目錄並透過 LDAP 與組織過去所建置或是未來規劃發展的資訊系統結合，以有效發揮目錄服務的功能，即是目前企業欲執行系統整合計畫的一個關鍵點。

本論文以東海大學作為實際導入此解決方案之對象，藉由連結其人力資源與學生學籍管理系統資料庫，透過 XML 標準與超目錄服務進行使用者身分識別資料交換，並提供資料異動自動化供應的能力與使用者自我服務的功能，以建立一個動態且完整的東海校園目錄，達成使用者身分識別資料統整，同時結合個人校園資訊入口網站的建置，以期後續能逐步整合各種校園資訊系統，提供使用者更簡便使用的資訊環境。

致　謝

首先要感謝指導教授羅文聰博士，讓我在這四年一邊工作一邊求學的過程中，在課業上甚至是工作領域裡對我的指導及各種資訊技術、管理實務觀念的啟發，我相信求學的目的在於學得解決問題的能力並應用在實際生活中，慶幸這段日子裡，我除了學到新的資訊技術與觀念，更增添一份解決日常工作所面臨問題的自信與勇氣。

接著要感謝我的工作夥伴，在這篇論文的系統實作上對我的支援與協助，他們分別是東海大學電算中心系統發展組的鐘子杰先生、李維貞小姐、江維信先生、鄭慧燕小姐、薛家羿先生、何琇瑩小姐、巫碧嫦小姐、巫文杰先生與陳天佑先生，以及兩位資工系學妹蔡欣儒同學與游莉敏同學，能和您們這群技術優良、工作認真、熱忱盡責與團隊合作的好夥伴一起工作，夫復何求！

最後感謝我最愛的家人，在這段期間所持續給予的關懷與鼓勵，使我能專心求學。對於所有關心我的朋友，也在此致上我最誠摯的謝意，並衷心地祝福你們。

目　次

圖 示 目 次

第一章 導論

本章說明現今企業所面臨資訊系統使用者身分識別資料管理(Identity Management)與整合的困境，並具體提出一個以目錄服務(Directory Services)[1]為基礎，利用 XML[5,9,10]標準達到資料交換與自動供應服務(Provisioning Services)[3,11]的企業資訊系統架構(Enterprise Information System Architecture)，同時針對目錄服務、XML 與超目錄服務(Meta Directory)[8,13]等技術和標準分別描述。

1.1 研究動機

過去以來，企業為了強化競爭力而持續不斷的導入各種資訊軟硬體解決方案，形成眾多且各自獨立的網路服務與資訊應用系統遍佈於公司各部門中，這些系統可能建立在各種異質平台，如 Netware, Windows NT, Windows 2000, Linux, Solaris 等作業系統上運作，且各系統皆有其存放使用者身分識別資料的專屬目錄，這些目錄彼此無法進行資料的交換，所以系統管理員不得不管理每一個使用者的數個帳號資料，當使用者個人資料異動時，系統管理員就必須對所有不同的目錄系統進行資料的維護；使用者則為了存取其數個被授權使用的系統，不得不記住數組系統帳號與密碼，危險的是，為了方便記憶，他們多半將帳號、密碼寫在桌面上任何地方，而破壞了原來使用密碼的安全目的；另外，當資訊部門未能對系統帳號使用生命週期妥善規劃與管理時，還可能出現員工離職仍可使用原有帳號密碼存取系統資訊的可怕情況。

以上種種不便與潛在風險，在企業內資訊系統持續增加的情況下，除了困擾使用者，更增加資訊人員日常工作負擔。企業現在終於發現，資訊系統的使用者帳號管理與整合一直是過去所忽視的，如何建置一個

單一、集中化的目錄服務以整合企業資訊系統使用者身分識別資訊、增進帳號管理效率顯然是當務之急；因此，透過目錄服務以提供使用者身份管理及存取權限控管(Access Control)，並讓資料在異質系統間進行自動交換與同步，就是提供企業跨平台、異質系統資源整合與資訊安全的一個解決方案。

基於以上的考量，本論文將建議一個以 LDAP[1,2,4](Lightweight Directory Access Protocol)目錄服務為基礎設計的企業資訊系統架構，並以東海大學為實作對象，建置植基於 LDAP 的校園使用者身分管理中心，同時以自動供應服務整合校務資訊系統與個人資訊入口網站(Portal Services)之使用者身分識別資料，並規劃使用者自我服務(User Self-Services)中心，以提供個人校園帳號註冊與查詢、密碼查詢與重設、個人基本資料如通訊地址、聯絡電話或者辦公室分機號碼維護等自助功能、透過目錄服務可迅速查詢校園部門組織與人員資訊，甚至可以使用即時訊息(Instant Message)工具進行業務聯繫、使用 email 進行訊息發送等多項服務功能，圖 1-1 為東海大學未來資訊整合與發展架構示意圖。

圖1-1 東海大學未來資訊整合與發展架構示意圖

本論文系統架構發展完成後，將會形成一個 LDAP-based Internet Service Framework，企業應用程式可透過目錄服務及資料交換與同步機制快速整合，同時由於使用者帳號管理的工作簡化，一個易於使用、具備安全且管理方便的企業資訊運作環境已經形成。

1.2 論文章節架構

本論文的第一章介紹提出本系統的研究動機及與本系統相關的研究項目和標準；第二章介紹本系統的整體架構，說明系統中各個模組的功能及實作所需用到的相關技術，並解釋各個模組間的互動關係；第三章說明系統實作設計與相關實作畫面；第四章則提出整個系統的結論並說明本系統尚待改良與建議新增之功能。

1.3 相關研究

以下將介紹和本論文有關的國內外相關研究，相關的部份有：目錄服務簡介、XML 簡介，以及可將目錄、資料庫與應用程式儲存庫(Repository)等許多不同資源資訊整合的超目錄服務架構介紹。

1.3.1 目錄服務簡介

目錄服務是一種網路服務，能讓網路上所有資源，包括使用者個人資料、圖像、電子郵件帳號、個人網頁、應用程式網址與存取設備如印表機等，依照一種階層化(Hierarchical)的組織架構來管理，這種組織方式能使得所有資源，依照較符合實際需要的方式排列，而不必讓使用者記住資源所在的實際位置。幾乎所有的目錄服務都是依據 X.500[1,2]規格發展，但由於該規格過於複雜，沒有廠商能夠完全實作 X.500 目錄服務，目前被廣泛採用的標準則為 LDAP 目錄服務。

通常企業利用目錄服務來儲存人員、設備及其他網路資源等資料，

然後提供員工一個簡便易用的線上通訊錄，利用員工姓名或編號等關鍵字來查詢目錄伺服器所儲存且授權公開的資訊如員工電話或分機號碼、電子郵件帳號、辦公室編號、組織部門關係圖等，所以目錄可視為一種特殊的儲存庫，專門針對讀取、瀏覽和搜尋操作而設計，擁有精細複雜的過濾能力，不同於新增、修改資料頻繁的關聯式資料庫，並沒有複雜的 Transactions 和 Rollback 機制，但在存取權限的許可下，仍可對目錄的內容進行修改。

以下針對目錄發展的兩個協定：X.500 與 LDAP 分別說明：

一、X.500

透過網路來存取目錄服務，需要一種統一的通訊方式(或稱協定)來規範各個系統對目錄服務的存取。ITU(International Telecommunications Union)提出定義架構在 ISO(International Standards Organization)/OSI(Open System Interface)環境的 X.500 標準，提供一個存取目錄的協定稱為 DAP(Directory Access Protocol)，可以讓各種系統以 DAP 來存取目錄服務。X.500 也提供可擴展式的資訊架構，可以很自由地儲存各種不同類型的資料，並可擴充至數百萬筆資料的儲存。但是 X.500 最大的問題是其通訊規範並不是建立在 TCP/IP 上，而且其命名規則太過複雜，使用與維護管理不易，執行效率不佳。為了要建立單一化、統一的目錄服務，反而要付出很高的管理成本。

二、LDAP

1990 年初，美國密西根大學根據 X.500 規格重新改良，提出一套可以在現行的 TCP/IP 環境下實作的目錄服務版本，稱為 LDAP。LDAP 最早被設計出來的目的是讓前端的應用系統只需要少許的資源，就可以透過 LDAP 來存取 X.500 的目錄資訊，也就

是說，LDAP 可以讓各個系統在 TCP/IP 網路上存取 X.500 目錄服務，因此可保留 X.500 資料模式的優點，將目錄系統擴充至可容納數百萬筆資料的規模，且不需要投入太大的建置成本，如圖 1-2[2]。

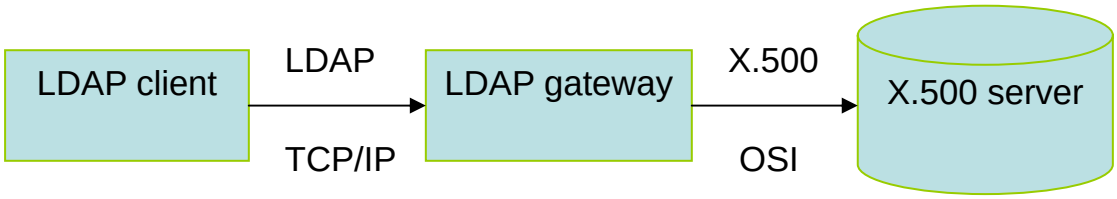

圖1-2 LDAP as a gateway to X.500示意圖

隨著 LDAP 版本演進及市場需求的影響，原本擔任 X.500 目錄系統中間角色的 LDAP 也可以開始獨立運作，成為一獨立的目錄系統，因此，企業用戶不必再建置 X.500 目錄系統，可直接建置以 LDAP 為基礎的 LDAP 目錄系統，將企業的資訊直接儲存在 LDAP 目錄之中。專為 LDAP 存取所設計的 LDAP Directory Server，其背後所支援的也是 X.500 所規範的資料結構與類別，可以與一般的 X.500 目錄服務相容以及進行資料的交換整合，資料的維護容易，且執行效率甚佳。

LDAP 規格詳細定義於 RFC2251 “Lightweight Directory Access Protocol (v3)”[6]，其中的 Information Model[1]與 Naming Model[1] 分述如下：

1. Information Model

LDAP server 所提供的目錄服務需要有一個儲存庫來儲存資料，為了提供 LDAP client 端對目錄快速查詢與回應的要求，LDAP 的儲存庫是以階層性樹狀結構(Hierarchical tree-like structure)來存放資料，稱為 DIT(Directory Information Tree)，如圖 1-3 所示。DIT 中的每一個節點都是一個物件，稱為 Entry，Entry 相當於關聯式資料庫中的 Record，Entry 可以用來儲存一個使用者的資料，也可以儲

存組織的資料，或是硬體設備的資料等。

一個 Entry 可以包含多種屬性(Attribute)，一個屬性包含一個屬性型態(Attribute Type)及一個或多個屬性值(Attribute Value)，而每個 Entry 至少要定義一種物件類別(object class)才能知道該 Entry 是屬於哪一種物件類別，例如有關人的物件類別就有 person、organizationalPerson 和 inetorgperson，如圖 1-4 所示，這些物件類別具有繼承的關係，例如 person 除了繼承 top[2]所定義的屬性外，還會定義其 Requires(必須的)及 Allows(可有的)屬性，其 Requires 屬性有 sn 和 cn，Allows 屬性則有 description、seeAlso、telephoneNumber 和 userPassword，而 organizationalPerson 則是繼承 person 中所定義的 Requires 和 Allows 屬性後，再定義出其他屬於本身物件的屬性。物件類別在 Entry 中是必要的屬性，其屬性型態為「objectClass」，表示方式如「objectClass: inetorgperson」。定義這些物件類別以及每個物件類別中所含的屬性即為目錄的 schema。

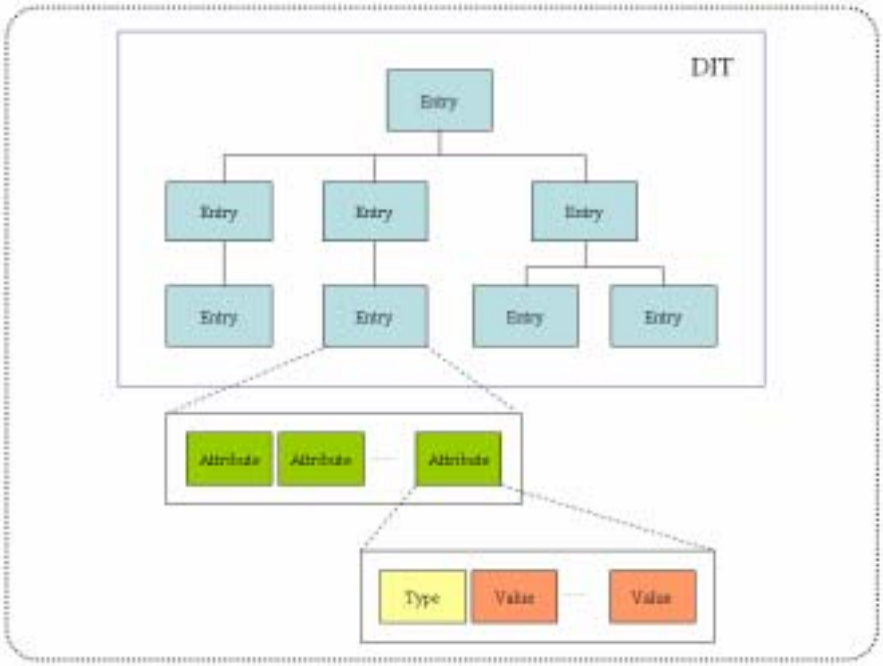

圖1-3 Directory Information Tree示意圖

每個 Entry 一定有一個辨識名稱 DN(Distinguished Name)，用以辨識其在 DIT 中的唯一性，也就是在 DIT 中不可能找到有相同 DN 的兩個 Entries。以圖 1-5 的 Entry 為例，該 Entry 所儲存的是使用者黃一民於東海大學目錄中的個人資料，其 objectClass 為 inetorgperson，而 top、person 與 organizationalPerson 則為被 inetorgperson 所繼承的物件類別，此 Entry 所使用到的屬性類別包含 cn、sn、givenName、uid、title、mail、telephoneNumber 等，黃一民即為 cn 的屬性值，而 mail 的屬性值則為 g892901@student.thu.edu.tw；另外，根據 DN 由右而左的路徑即代表樹狀結構由上而下的 Entry，我們可以得知其在 DIT 中的位置，而 DN 中的 uid=g892901 即為該 Entry 的 RDN(Relative Distinguished Name)。

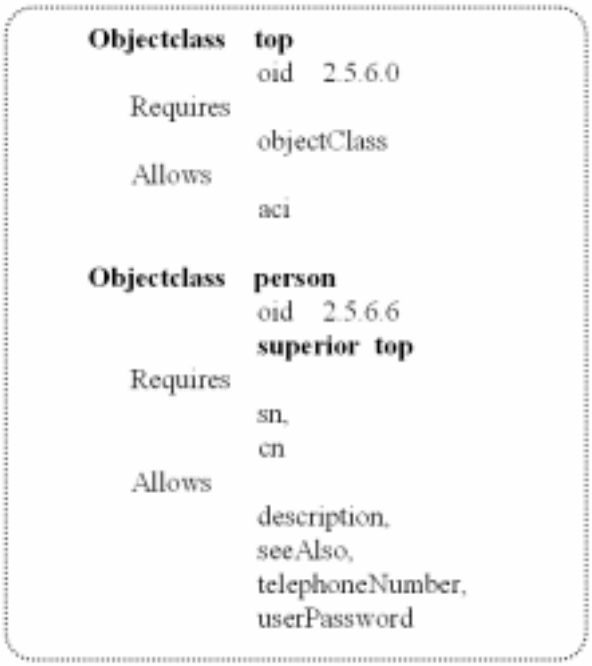

圖1-4 top和person物件類別及其所定義的屬性

2. Naming Model

由於 LDAP 中的目錄為階層式樹狀結構，一般可依地理位置和

組織關係來設計。DN 的命名原則有兩種：

dn:uid=g892901,ou=Students89,ou=people,dc=thu,dc=edu,dc=tw
cn:黃一民
sn:黃
givenName:黃一民
objectClass:top
objectClass:person
objectClass:organizationalPerson
objectClass:inetorgperson
uid:g892901
svuserclass:G29040
enterday:8909
title:學生
mail:g892901@student.thu.edu.tw
telephoneNumber:04-23590121#3035

圖1-5 一個屬於inetorgperson物件類別的Entry

(1) 傳統命名(Traditional Naming)[1]

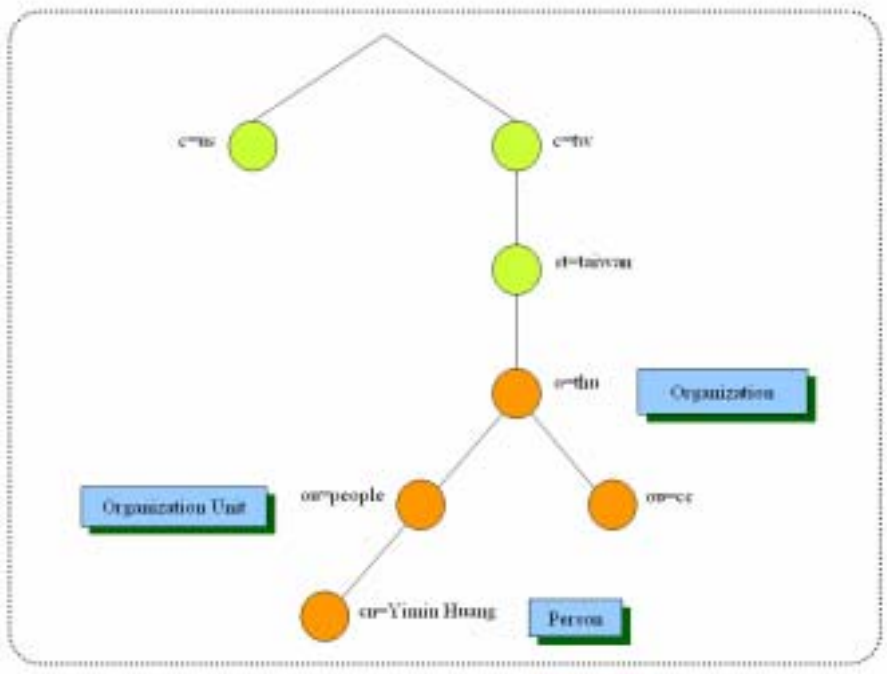

圖1-6 Traditional Naming of DN示意圖

傳統命名原則是根據 X.500 對 DN 命名原則修訂，如圖

1-6 所示，是依照國家組織的從屬關係來建立目錄中的 Entry，最上層的 Entry 為國家—台灣(c=tw)，下一層為省份—台灣省(st=taiwan)，再下一層為組織—東海大學(o=thu)，再下一層為組織內的單位—電算中心(ou=cc)，組織內的單位下有人(cn=Yimin Huang)。

(2) 網際網路命名(Internet Naming)[1]

以目前 LDAP 應用在網際網路的情況看來，傳統的命名原則無法提供較適宜的 Entry 命名方式，為了實際的需求及方便性，所以根據 TCP/IP 網域命名為基礎而訂定了新的命名原則，即網際網路命名，其使用新的屬性 dc(domain country)來代表 TCP/IP 域名。以圖 1-7 東海大學 DIT 的表示方式為例，Entry 由上而下分別為 dc=tw，dc=edu 與 dc=thu，DN 即為 dc=thu,dc=edu,dc=tw，對應域名為 thu.edu.tw，而其下層命名原則仍與傳統命名原則相同。

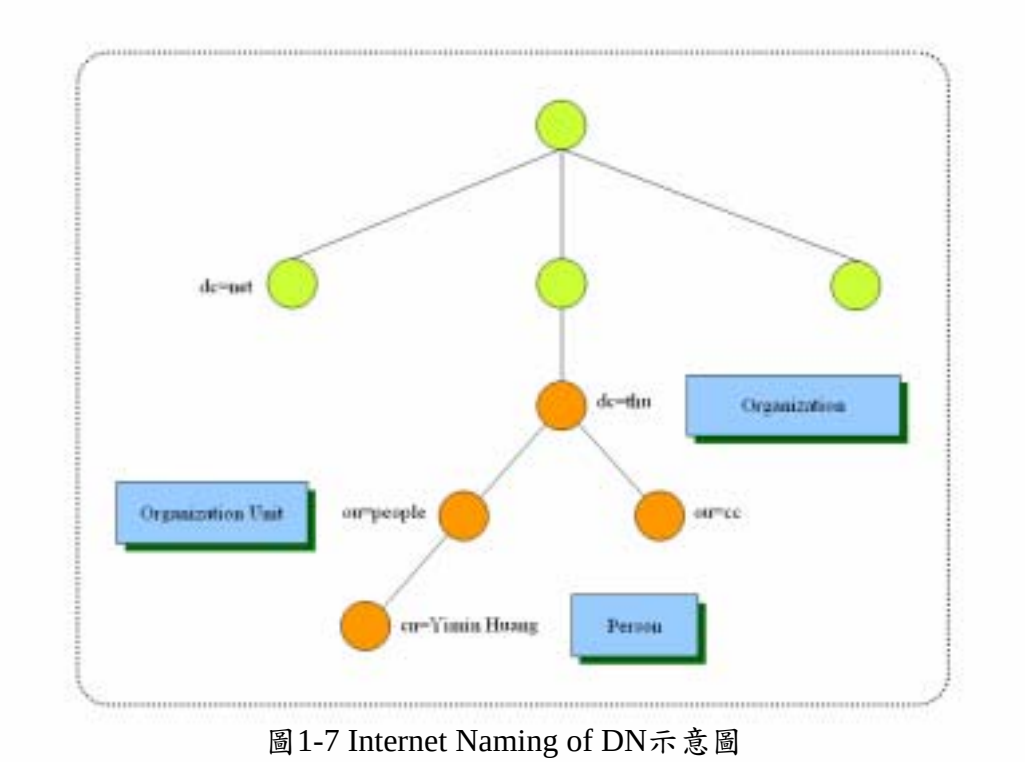

圖1-7 Internet Naming of DN示意圖

1.3.2 XML 簡介

XML 的全名是 eXtensible Markup Language，中文譯為可擴展標記語言，制定的目的是為了能在網際網路上傳送或處理資料，著重在對文件資料的結構性描述，其前身是標準通用標記語言 SGML (Standard Generalized Markup Language)[9]。經 ISO 批准的 SGML 標準(ISO 8879：1986)，其成熟度與穩定性相當高，主要的目的是在提供描述電子文件的規範，也就是對文件進行結構化的法則，而文件經過這種標準通用的結構化處理後，即可被廣泛的應用，並透過電腦來作最有效的處理。SGML 的應用通常以大型企業、政府機關、需大量資料維護或特殊複雜資料處理的行業為主，如美國的國稅局(IRS)便以 SGML 來設計稅表等文件[10]、出版工商業及醫療系統，都有使用 SGML 的應用，如文件全文檢索、製作手冊、製作目錄、大量文件的版本變更管理。

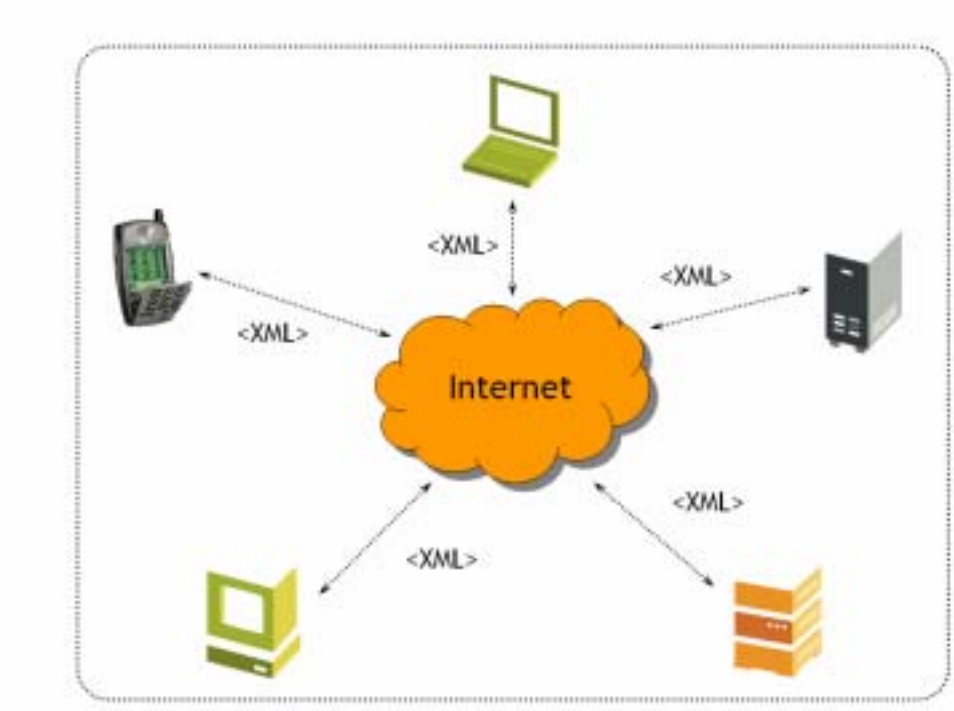

圖1-8 異質系統藉由XML進行資訊互通示意圖

由於 SGML 的複雜度高，無法普及並適合在 WEB 上的應用，而超文件標記語言 HTML(HyperText Markup Language)又過度的簡單，無法

自訂控制標籤，只能應用在資料的顯示上，無法擴展與處理大量的資料，XML 就是為了解決這些問題而制定產生的。SGML 與 XML 都是一種 Meta Language，可以用來制定產生其他的標記語言(Markup Language)，透過 DTD(Document Type Definition)的使用來加強文件結構上的要求，並可結合不同的排版樣本(Style Sheet)，顯示出不同的瀏覽效果。

從商業角度來看，XML 最大的貢獻在於協助異質系統間的資訊互通。過去，企業間或是企業內部因為建置了許多不同的系統平台、應用程式與資料庫系統，造成資訊流通的困難，這些異質系統之間欲進行資訊交流，往往需要使用特殊的軟體，才能順利跨越彼此的疆界予以整合。現在，企業異質系統之間可以很方便的透過 XML 來作資料交換媒介，XML 簡單易讀，對於各類型的資料，舉凡物件、文章、資料庫裡的資料、目錄所存的資料、圖形等等，都能標示。企業只要於執行資料交流的機器安裝 XML 解析器，便可以解讀由其他機器傳來的資訊，更方便的是，XML 解析器取得容易，有許多相關軟體可供人免費下載，且大多以 JAVA 寫成，使得只要在安裝有 JVM(Java Virtual Machine)的設備，如 Server、PDA、Mobile Phone 等，都立即成為可解析 XML 資訊的平台，如圖 1-8 所示，如此，異質系統間不用擔心看不懂對方的資料格式，也不用擔心對方內部是採用何種格式儲存資料，運用 XML 作為中介格式即可互相交換或讀取目的資料。

標記語言是由自訂的標籤(tags)所組成，這些標籤若單獨存在是無意義的，必須結合文件資料才能成為一份有用的電子文件，而有用的電子文件是指應用軟體能解讀電子文件中的標記語言，並藉由標記語言的意義來對電子文件做特定的處理。所以標記語言可使電子文件變得更有結構性，而這種結構性使得應用軟體能加以解讀並彈性的應用。

圖 1-9 為一個 XML 的使用範例，此範例將圖 1-5 的 inetorgperson Entry 透過 XML 的格式來表示，以<people>、<inetorgperson>、<cn>等目錄內 Entry 的類別與屬性名稱作為 XML 的控制標籤，控制標籤都是成對的，在成對的控制標籤中所包含的就是 XML 文件中的資料，如「<cn>黃一民</cn>」中的黃一民就是 cn 控制標籤所包含的資料；在控制標籤內還可以允許設定屬性，如「<inetorgperson dn= "uid=g892901,ou=Students89,ou=people,dc=thu,dc=edu,dc=tw">」中的 dn 即為 inetorgperson 控制標籤的屬性。由此範例可以得知 XML 文件也是以階層性樹狀結構來表示，如同目錄服務的 DIT，所以無論資料是儲存於關聯式資料庫或是目錄中，都可以透過 XML 的轉換，讓異質系統間可以進行資料的交換。

```
<People>
    <inetorgperson dn="uid=g892901,ou=Students89,ou=people,
     dc=thu,dc=edu,dc=tw" >
        <cn>黃一民</cn>
        <sn>黃</sn>
        <givenName>黃一民</givenName>
        <objectClass>top</objectClass>
        <objectClass>person</objectClass>
        <objectClass>organizationalPerson</objectClass>
        <objectClass>inetorgperson</objectClass>
        <uid>g892901</uid>
        <svuserclass>G29040</svuserclass>
        <enterday>8909</enterday>
        <title>學生</title>
        <mail>g892901@student.thu.edu.tw</mail>
        <telephoneNumber>04-3590121#3035</telephoneNumber>
    </inetorgperson>
</People>
```

圖1-9 XML文件範例

可擴展樣規語言 XSL(Extensible Stylesheet Language)[10]是專門為 XML 設計的樣板語言。XSL 共分為兩部分，第一部分是負責將 XML 原始碼轉換成另一種格式，第二部分稱作樣板物件 FO(Formatting

Object)，提供大量的樣板指令，可用來配合印刷或螢幕顯示，精確的設定外觀式樣，譬如字的大小、擺放的位置等。1994 年 W3C 正式將 XSL 關於轉換語法的敘述，從 XSL 分出來，另稱為 XSLT(Extensible Stylesheet Language Transformation)[7]，T 代表 Transformation，也就是轉換的意思，此後，XSL 僅剩下 FO 的部分，FO 的發展已經停滯，和日新又新的 XSLT 形成強烈的對比。XSLT 本身即是 XML 的應用，且是直接架構在 XML 的語法之上，可以輸出任何格式正確的 XML 文件，它使用 XML 路徑語言 XPath(XML Path Language)在 XML 文件找尋資料，其語法可以用來指出文件架構或資料的位置。XSLT 1.0 版本於 1999 年首次對外公佈，是目前 XML 技術人員在轉換各種不同 XML 標籤所定義的文件時，最為重要的工具之一。

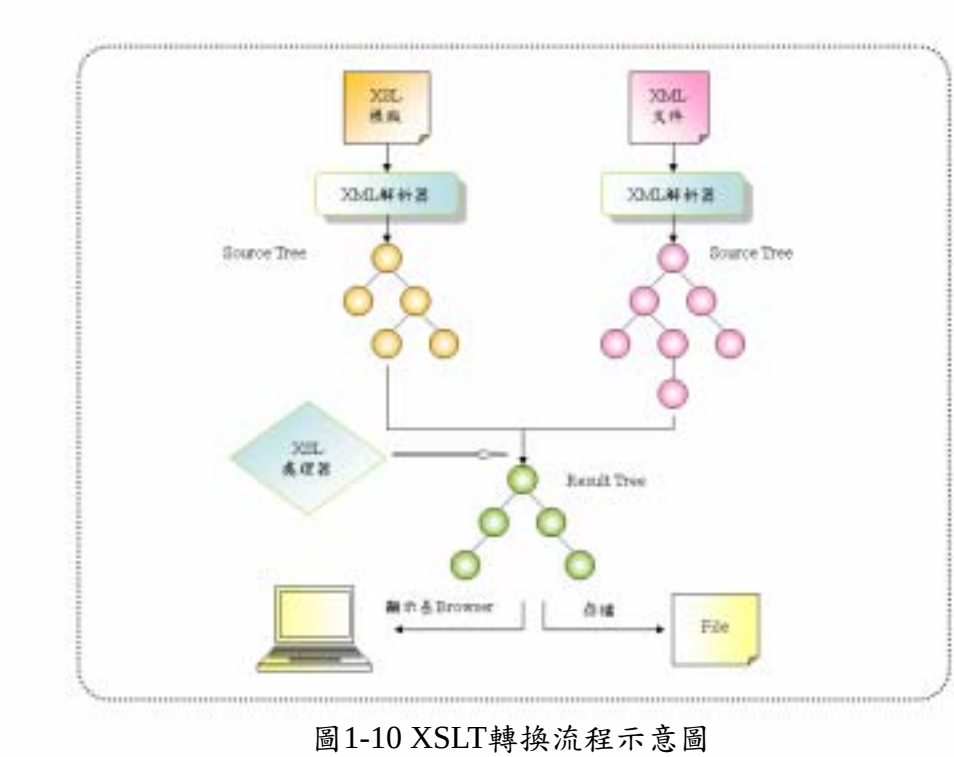

圖1-10 XSLT轉換流程示意圖

如圖 1-10[10]所示，任何 XML 文件，都必須經過 XML 解析器將 XML 文件中的物件和結構分析出來，成為來源樹(Source Tree)，才能再進一步利用。另外，XSLT 轉換必須由特殊的軟體來擔任，這種軟體稱

為 XSL 處理器(XSL Processor)，在 XSL 處理器啟動前，XSL 樣板也必須先經由 XML 解析器分析成來源樹，因為 XSL 樣板也可視為一份 XML 文件，掌握了樣板中的指令後，XSL 處理器便開始依照樣板中的指示，在來源樹上搜尋，若找到合適的物件，即按照樣板的規定(Template Rules)，產生一個 XML 片段，一直到整個 Tree 都尋遍為止，即由 XSL 處理器產生一個新的結果樹(Result Tree)，XSLT 轉換後的輸出碼，可儲存至另一個新的檔案，或以 HTML 格式經由瀏覽器顯示出來，而原來 XML 檔案則保持不變，所以樣板是用來設定 XML 文件外觀的，並不會影響原來的 XML 原始碼。

1.3.3 超目錄服務簡介

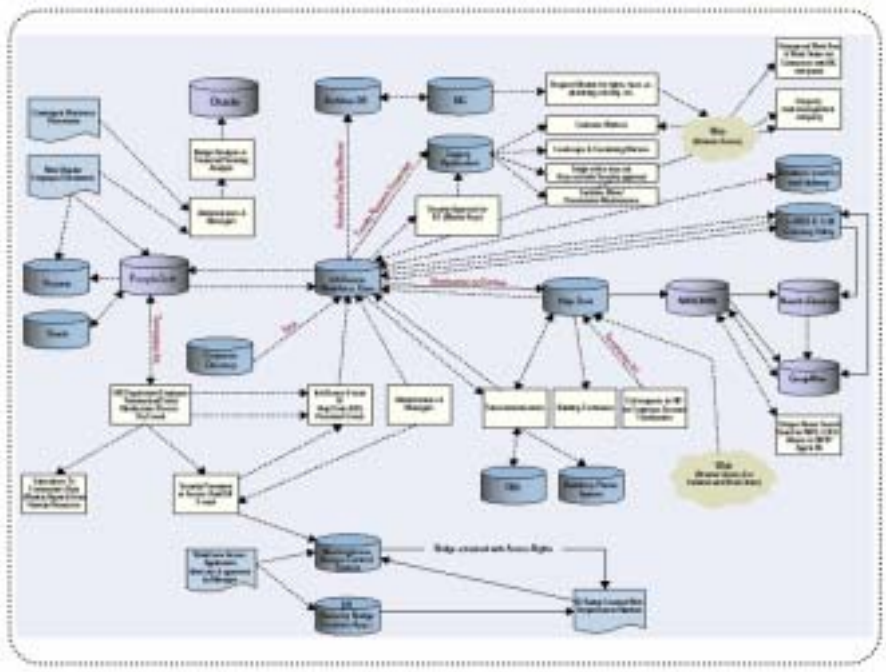

圖1-11 System Architecture without Meta Directory示意圖

多數的企業，有關系統用戶資訊大都分散在不同的目錄、網路作業系統與資料庫中，如圖 1-11[12]所示。為了將用戶資訊整合，並且可跨越各應用程式和系統，達到資料共享的管理，導入超目錄服務是一個有效的解決方案。簡單來說，超目錄服務能將目錄、資料庫、企業人力資

源管理系統、電子郵件以及網路作業系統等許多不同的使用者身份識別資訊整合，協助企業處理員工身份管理等工作，避免過去許多企業為了維持眾多資訊系統應用程式運作正常，資訊人員常需維護多個不同的目錄或儲存庫而疲於奔命。例如過去企業新進員工至人資部門報到後，他必須向 IT 部門提出資訊系統使用申請，結果該名員工可能在幾天後才完全獲得其所有資訊系統使用帳號密碼及登入權限，這種情況，無疑是企業人力資源的浪費，同樣的情況也發生在員工離職，資訊系統管理員未能即時關閉該員所有系統使用帳號與存取權限，造成企業資訊安全漏洞。

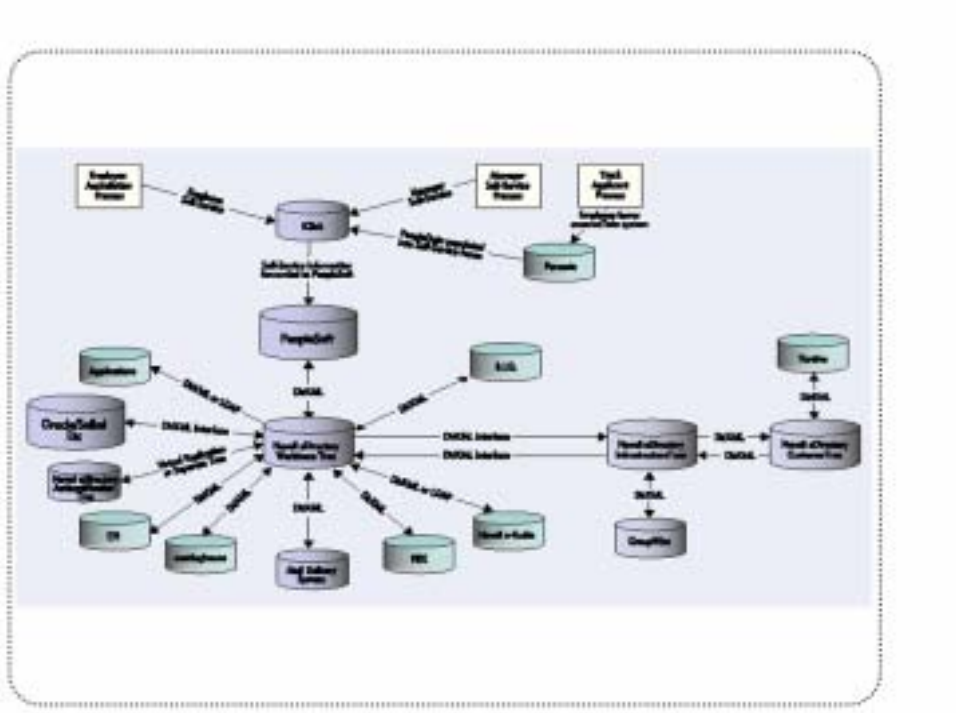

圖1-12 System Architecture with Meta Directory示意圖

使用超目錄服務就是要解決上述問題，當企業員工離職、報到或者調整部門時，身份識別資料的異動隨即供應至所有需要此識別資料的系統，這樣一來，新進員工馬上可以使用電子郵件服務，或者員工辦理離職手續後，其企業資訊系統帳號也隨之關閉。因為超目錄服務需要將企業識別資訊從非常分散的資源，建成一個儲存在目錄下單一且統合的狀

態，所以這個目錄可視為超目錄服務的核心元件，用來存放整個企業內相關儲存庫中的身份識別資料。另外，目前資訊廠商所發表的超目錄服務產品如 Novell 公司的 Novell Identity Manager(或稱 DirXML) [12,14,17] 與 Sun Microsystems 公司的 Sun Java System Identity Manager 都包含服務供應與同步化功能(Synchronization)，以協助在不同應用程式中資訊的雙向同步整合，同時提供多種資源轉接器(Connector/Driver)，可針對目錄伺服器、資料庫、訊息平台、以及企業應用程式如 PeopleSoft、SAP 等，如圖 1-12 所示，支援服務供應與身份識別的同步，完成整個身份識別管理生命週期，從初始帳號建立與後續資料變更，一直到與公司的關係到期時註銷帳號等等，完全地予以自動化。

在現今企業資訊系統分散且使用者身份識別資料難以整合與管理的狀況下，透過建置超目錄服務於企業資訊系統的核心，可將之視為企業現行與未來資訊系統架構的中樞神經，因為超目錄服務中的目錄不僅是企業所有身分識別資料的集合，亦可作為變更資訊的來源與查詢企業內身份識別屬性的標的，而企業後續所建置的資訊系統只要依循 LDAP 協定，可直接以此目錄進行使用者身分驗證與存取權限設定。

第二章 系統架構

本章進一步介紹以目錄服務為基礎的企業資訊系統架構(Enterprise Information System Architecture Based on LDAP Directory Services)，建議以目錄服務建置企業使用者身分識別資料中心，並且在不影響原有資訊系統的情況下，藉由自動供應服務同步各系統之身分認證資料；同時詳細說明供應服務與使用者自我服務組成元件及其運作機制。

2.1 以目錄服務為基礎的企業資訊系統架構概述

隨著資訊科技的發展，企業不斷地將各種符合營運所需的應用系統導入其資訊運作環境中，當企業在他們的資訊架構新增一個系統時，總會建置一套此系統所需要的用戶帳號資料，可是，通常在該企業資訊系統架構中的一個或多個系統中，早已存在以某種形式包含了這些資訊，於是，這些分佈在各系統且類似的資訊加深了企業在整個公司系統內維持資料準確性和連續性的難度，而且，更新資料的過程不僅要耗費系統管理員驚人的時間和辛勞，還有可能埋藏代價巨大的人工錯誤危險。

為了克服以上的缺點，本論文建議一個以目錄服務為基礎的企業資訊系統架構，藉由允許企業內的應用程式集中檢索和儲存共用資訊，以目錄作為企業各種身份識別資料的儲存庫，同時透過超目錄服務的機制協助「非目錄化」系統以及「目錄化」系統進行資料分享與自動供應的功能，使其與目錄無縫結合，並進一步利用目錄的強大功能，建立一個基於目錄服務的企業通訊錄，提供企業員工線上查詢公司內人員、組織資訊甚至直接利用電子郵件與即時傳訊工具進行聯繫溝通，同時包含線上更改個人基本資料與密碼的服務，當然，任何的個人資料修改將根據所訂定的政策(Policy)自動同步至需要此資料的所有應用程式儲存庫。

圖 2-1 為本論文建議以目錄服務為基礎的企業資訊系統架構示意圖，除了包含核心元件目錄服務作為 Identity Vault(企業用戶身分庫)外，主要還有四個模組，分別是供應服務、使用者自我服務、應用程式服務(Application Services)以及 IT 服務管理(IT Services Management)，其中 Application Services 又包含企業資訊應用系統、網路服務(Network Services)、個人資訊入口網站服務與商業智慧服務(Business Intelligence Services)等元件，而導入 IT Services Management 主要目的在將 IT 系統的管理及服務運作流程標準化並且融入 IT 部門的日常營運中，希望藉以提高 IT 需求回應時間與服務水準。以下各節將分別針對幾個重要的關鍵元件做詳細說明。

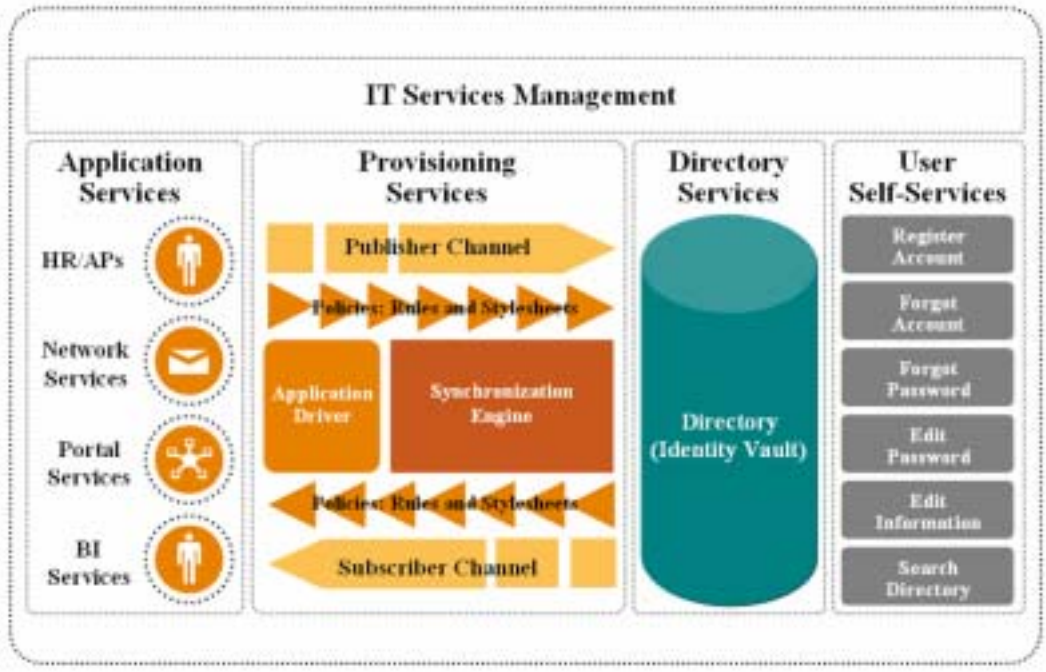

圖2-1 以目錄服務為基礎的企業資訊系統架構示意圖

2.1.1 Synchronization Engine

供應服務主要是用來處理目錄或應用程式發生的資料改變事件而進行一系列的資料轉換與派送。由於各種儲存庫 Schema 的不同，Synchronization Engine[16,18]透過使用 XML 跨平台的強大功能來進行

Repository 間資料異動的規則與格式轉換，按照 Policy 的定義處理一個資料變動事件，其結果就是一個包含資料變化事件的 XML 檔案被轉換成一個或多個特定用途的 XML 檔案，這些檔案被應用於 Target Repository 來進行必要的變化，使相關的資料能同步異動。這種轉換使得不同的資訊系統間實現了資料交換的可能，另外，資料改變事件也可以被轉換成另一種改變事件，例如當我們在電子郵件系統刪除一個使用者帳號時，這個刪除事件會被 Synchronization Engine 轉換為從目錄中將該使用者 Entry 內 email 屬性所對應的電子郵件帳號屬性值刪除，而非把該使用者在目錄中的 Entry 刪除。

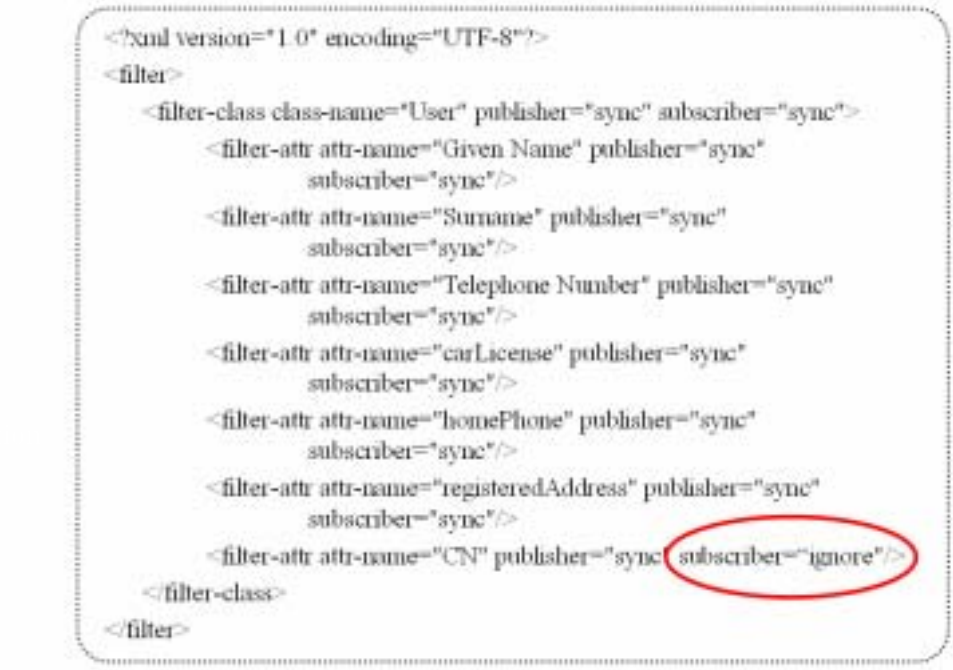

圖2-2 Filter範例

在 Synchronization Engine 中，資料的傳送過程分別受到 Filter[18](過濾器)、Rules[16,17,18] (規則)和 Stylesheets[17,18] (資料格式表)所控制與影響。Filter 是用來對欲進行同步的物件類別和屬性的限制定義，也就是說，系統管理員必須在 Filter 設定哪些類別和屬性允許或接受同步作業，如圖 2-2 所示，CN 屬性僅接受 Publisher Channel[16,17](發佈通

道)的資料同步(sync)，而不接受 Subscriber Channel[16,17](訂閱通道)的資料同步(ignore)，所以藉由 Filter 的處理，也同時保證了資料同步來源的可信賴性(authoritative) [17,18]。Rules 和 Stylesheets (統稱為 Policies)同樣是由系統管理員所設定，Synchronization Engine 即是根據 Policies 來進行一系列的規則處理(包含映射模式、對應、建立及放置規則)、指令轉換與資料格式的處理，如圖 2-3[18]所示。

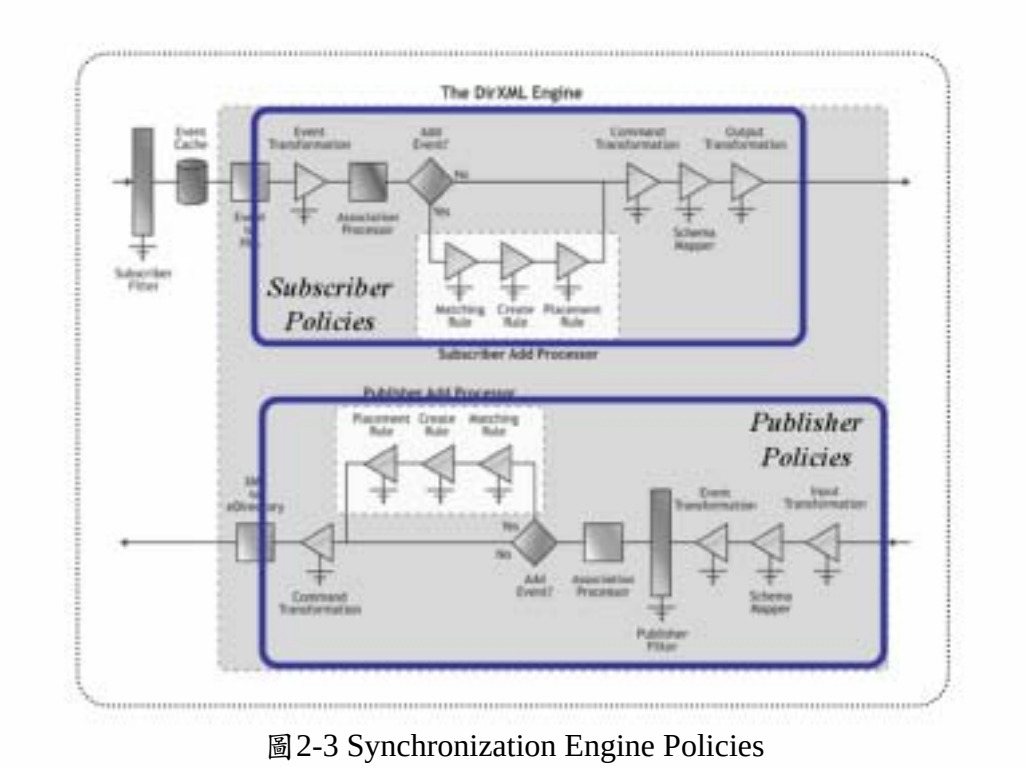

圖2-3 Synchronization Engine Policies

以下針對圖 2-3 中三角形箭頭所代表的 Policies 項目作詳細的功能說明：

一、 Event Transformation(事件轉換)：

Event Transformation 是一種 XSLT stylesheet，負責將 Event[14]由一種型態轉換成另一種型態，且一個 Event 可以依規則轉換成多個 Event，如圖 2-4 所示，一個刪除事件被轉換成一個修改事件。

二、 Matching Rule(對應物件)：

Matching Rule 是一種基於 XML 的規則，負責確認目錄和應

用程式物件之間是否擁有唯一的對應關係，以供判斷是否新增物件時所用，如圖 2-5 所示。

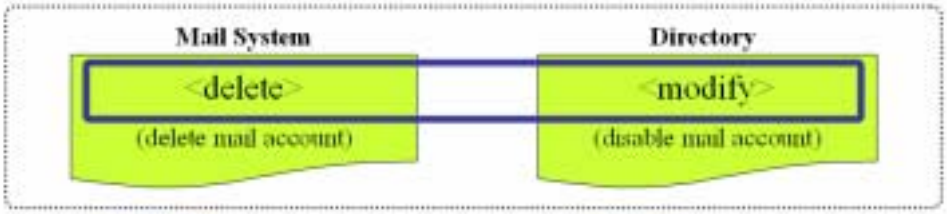

圖2-4 Event Transformation

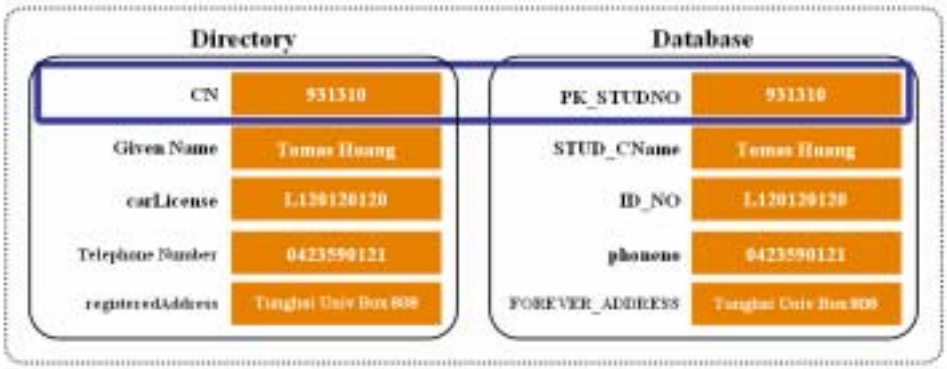

圖2-5 Matching Rule

三、 Create Rule(建立物件)：

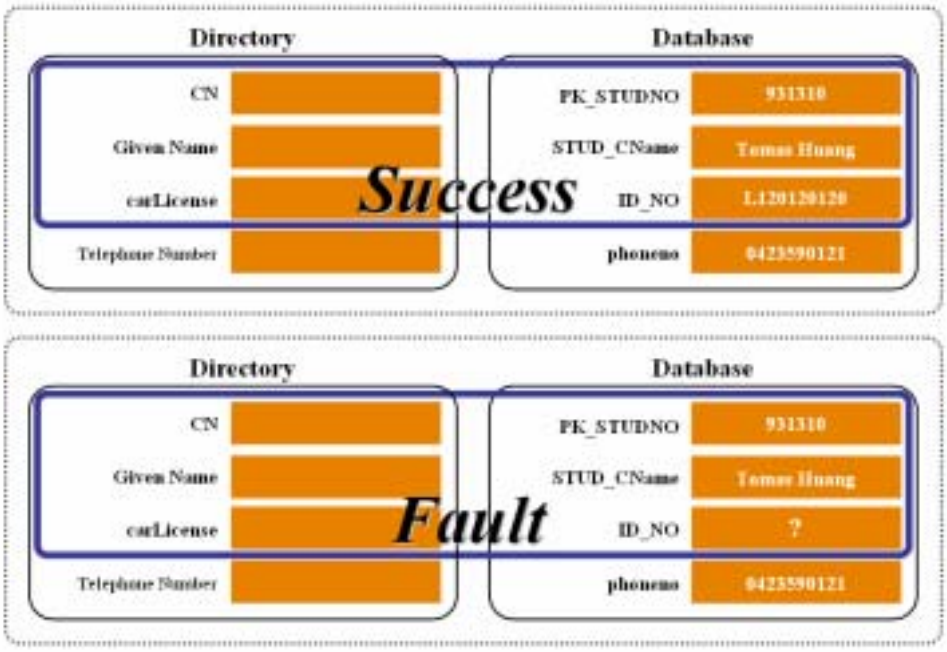

圖2-6 Create Rule

Create Rule 是一種基於 XML 的規則，負責確認在目錄或應用程式中建立物件之前，是否所有的要求屬性已經得到滿足，如圖 2-6 所示，若 Create Rule 設定在目錄中建立使用者物件時，需要學生學號、姓名與身分證字號等欄位資料，則任何缺少上述欄位資料的新增動作皆會被拒絕(Vetoed)。

四、 Placement Rule(放置物件)：

Placement Rule 是一種基於 XML 的規則，負責確認在目錄或應用程式中所新增的物件應該置於何處，如圖 2-7 所示，若 Placement Rule 設定欲在目錄中新增的物件類別為 User 時，該物件將會被系統擺放至 dn: ou=Students93,ou=Users,o=THU 下。

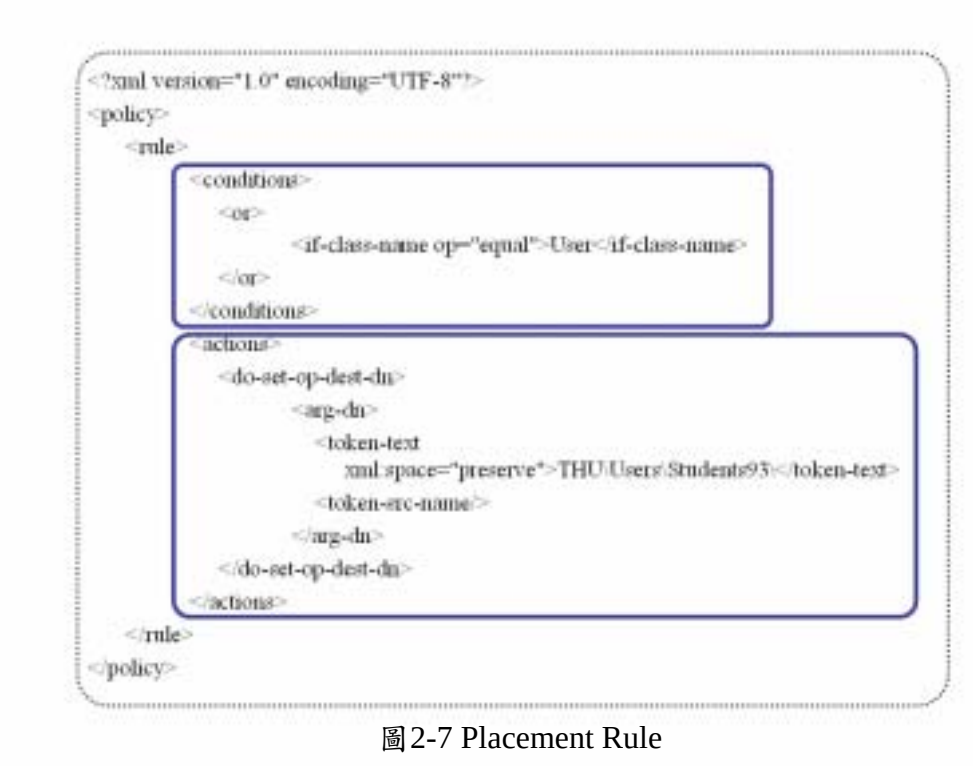

圖2-7 Placement Rule

五、 Command Transformation(命令轉換)：

Command Transformation 是一種 XSLT stylesheet，負責轉換命令模式，例如當 Synchronization Engine 從應用程式中找到一個未設定關聯的對應時，它除了依原政策對此物件的某些屬性進行異

動，同時也發出另一道命令於目錄內對應物件新增此一關聯屬性。

六、 Schema Mapping(映射模式)：

Schema Mapping 是一種基於 XML 的規則，負責確認目錄和應用程式之間所有類別與屬性一對一的映射，如圖 2-8 所示，目錄類別 User 與資料庫表格 view_thaastu 映射，而屬性 Telephone Number 則與欄位 phoneno 映射，以下依此類推。

圖2-8 Schema Mapping

七、 Output/Input Transformation(資料轉換)：

Data Transformation 是一種 XSLT stylesheet，負責將資料從一種格式轉換成另一種格式。例如若資料庫表格的生日欄位資料格式為年月日(如 690505)，而其映射的目錄生日屬性格式為月日年(如 050569)，當進行資料庫至目錄的資料異動同步時，該筆「690505」的 data 將被轉換為「050569」。

2.1.2 Application Driver

Application Driver[17,19]負責在 Application(應用程式、目錄、資料

庫)與 Synchronization Engine 間傳遞資訊，同時也協助以 XSLT 進行應用程式原生格式(Application Native Format, ANF)與 XML-Format 之轉換。

Application Driver 是依據所連結的應用程式特性來實作，例如應用程式若是資料庫，則可以實作基於 JDBC(Java Database Connectivity)的 Driver 用以和大部分的資料庫系統如 Oracle、Sybase Database 進行溝通；而若應用程式是目錄，則可以實作基於 LDAP 的 Driver 與之連結，也就是說，Identity Vault 欲與各種不同的應用程式進行資訊分享與同步時，必須針對個別應用程式採用適當的 Application Driver 來連線。

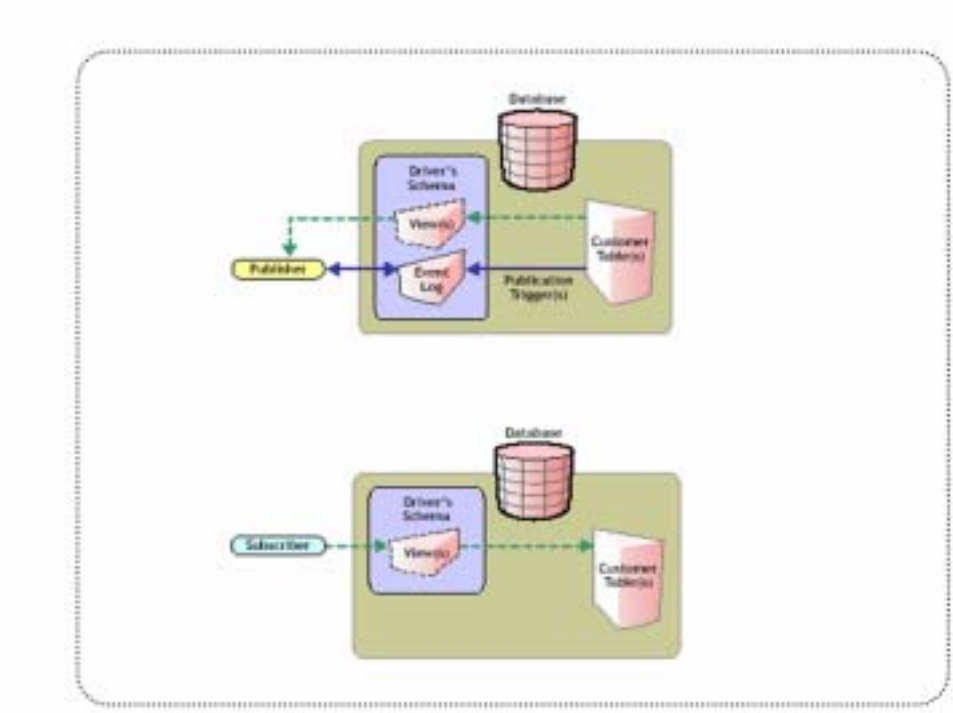

圖2-9 資料庫與目錄間資料異動同步流程示意圖

圖 2-9[19]分別表示資料庫應用程式與目錄間的資料異動同步事件，透過 Publisher 和 Subscriber Channels 並藉由 Application Driver for JDBC 與 Synchronization Engine 傳遞相關訊息以進行同步動作。以 Publisher Channel 說明，當資料庫的 Customer Table 發生資料異動，隨即 Trigger 一筆記錄至 EventLog，而 Publisher 讀取該筆 Log 後即從

Customer View 找到該筆異動記錄，並傳送至 Synchronization Engine 進行一系列 Policies 處理，以同步更改目錄內的對應物件屬性或執行新增、刪除物件動作，最後當資料同步成功後，Publisher 將會把該筆 Log 刪除或標記表示已完成同步動作。

2.1.3 User Self-Services

User Self-Services 主要在提供系統使用者對個人帳號密碼或基本資料的管理機制，讓使用者能以自助的方式解決個人系統帳號密碼忘記、重設或修改等情況，同時開放部分的個人基本資料項目由使用者自行管理，如通訊地址、聯絡電話等，以及線上企業通訊錄等服務。以上功能除了讓使用者能即時處理個人帳號密碼問題以及修改基本資料外，也相對降低了企業相關部門或資訊服務台處理使用者身分識別資料的頻率，直接節省企業成本。

然而，企業是否能提供 User Self-Services 這項服務，必須取決於該企業能否有效導入身分管理機制，這也是本論文前幾節所提到，企業可藉由超目錄服務整合所有相關的資料儲存庫，並建置完整目錄服務以作為身分管理中心，所以 User Self-Services 僅需連結身分管理中心(Directory)，即能以 LDAP 為基礎提供各項使用者自助服務功能，而不需要另外花費時間在後端各系統與資料的整合，因為這個部分早已透過供應服務解決了。圖 2-10 為東海大學目前規劃的 User Self-Services 服務項目，各項目功能分述如下：

一、 Register Account(帳號註冊)：

Register Account 提供使用者登入系統之前所必須完成的相關設定，包含帳號與其他身分資料的驗證，要求使用者自行設定密碼與密碼提示語、答案，並允許輸入另一組電子郵件帳號供後續作為帳號查詢之用。使用者在完成上述所有設定後，企業即根據其身分

開放所有基本必須的企業資訊系統(如收發電子郵件、登入企業個人入口網站)使用權限供使用者開始著手進行個人工作。

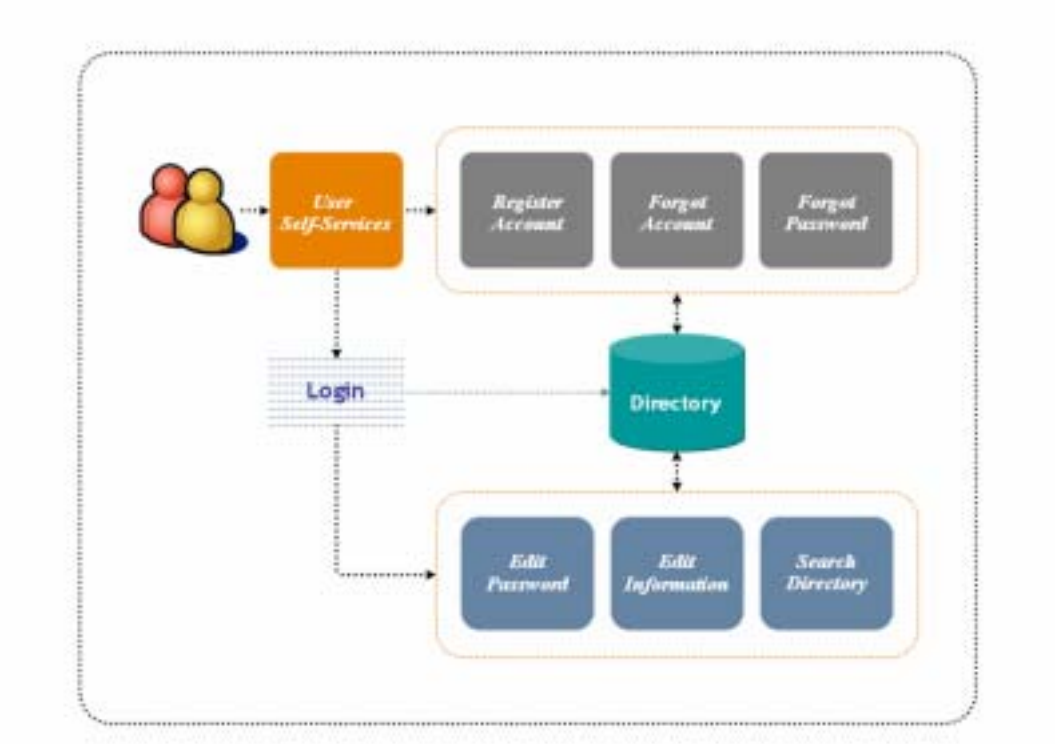

圖2-10 東海大學USSC服務項目

二、 Forgot Account(忘記帳號)：

Forgot Account 提供使用者忘記個人帳號時使用，系統將發送其帳號至使用者於「Register Account」時所輸入的電子郵件信箱，由使用者自行查詢個人帳號。這個功能將降低使用者向資訊服務台尋求協助查詢帳號的頻率。

三、 Forgot Password(忘記密碼)：

Forgot Password 提供使用者忘記個人密碼時使用，藉由詢問與驗證使用者帳號及其於「Register Account」時所輸入的密碼提示語、答案，來達到讓使用者重設密碼的目的。這個功能將有效避免傳統使用者向資訊服務台尋求協助處理其密碼問題。

四、 Edit Password(變更密碼)：

Edit Password 提供使用者可以隨時進行個人密碼的修改。

五、 Edit Information(修改個人資料)：

Edit Information 提供使用者自行管理如通訊地址、行動電話等基本資料，而任何資料的修改都可以透過供應服務同步至相關系統，如人力資源管理系統資料庫或企業其他目錄中。這個功能將取代使用者過去必須連絡人資部門處理資料變更的傳統方式。

六、 Search Directory(線上通訊錄)：

當企業身分管理中心建立後，藉由建置一個線上通訊錄的功能來服務企業內員工，甚至對企業外使用者適度開放查詢屬性是非常便利的，而且所開放查詢的資料絕對是最新最即時。透過 Search Directory 機制，讓我們尋找一個企業內員工資料如辦公室位置、電話分機號碼、電子郵件帳號、個人網站甚至是部門主管都非常迅速與方便，尤其是若能再提供以點選方式就可以直接透過電子郵件收發軟體發送 email 或與某位員工以即時傳訊工具作即時溝通聯繫，這個 Search Directory 功能將更為完善、吸引使用者並有效達到企業內員工協同作業的需求。

2.2 模組互動關係

2.2.1 Building Associations

為了讓 Synchronization Engine 能夠在目錄與各種不同的應用程式間傳送與同步異動資料，明確的記錄目錄物件與應用程式物件的對應關係，以達到物件或屬性資料異動同步的正確性，建立關聯(Associations)[16,17,18]是必須的，這個關聯不僅要能夠有效指出物件之間的對應關係，當我們新增或找到未設定關聯的物件時，Synchronization Engine 也必須能自動建立該關聯。利用身分管理中心的物件屬性來存放關聯是最簡便的解決方式，換句話說，這個關聯屬性負責讓 Synchronization

Engine 知道有哪些應用程式物件是和此企業目錄物件有關，同時也明確定義哪個關鍵值是唯一可進行關聯物件的對應。

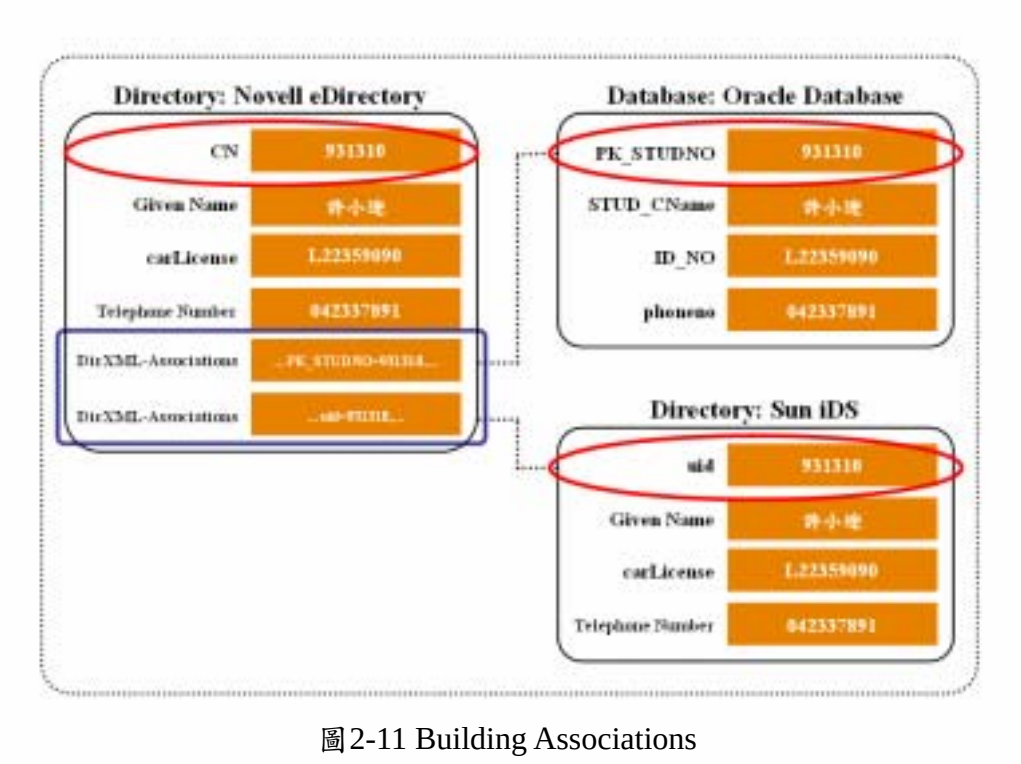

圖2-11 Building Associations

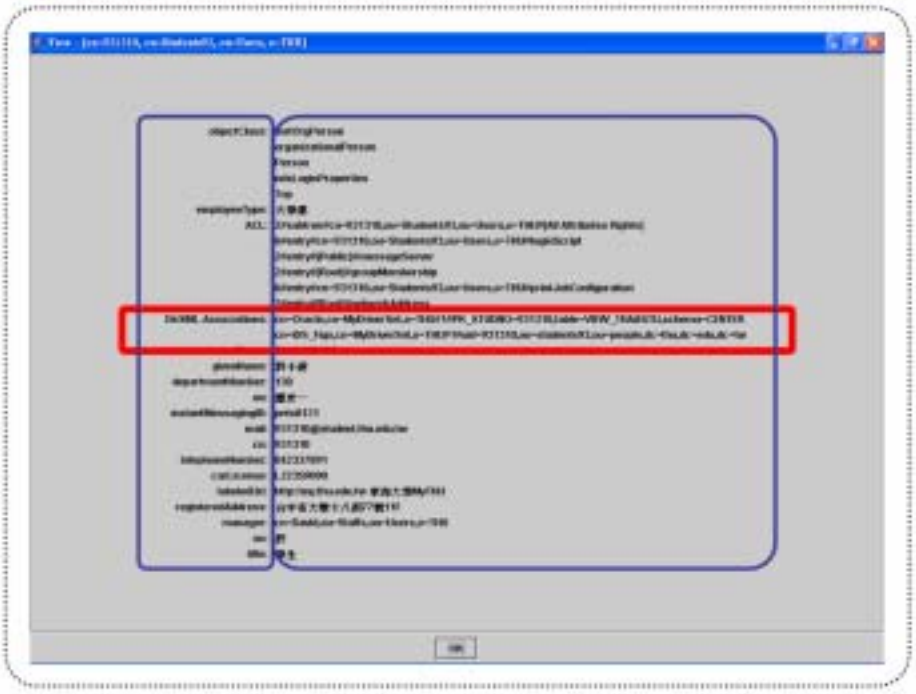

圖2-12 具有兩個Associations屬性的Entry範例

因為目錄可能和多個應用程式連結以共享資訊，所以 Entry 內的關

聯屬性是可以多值的，也就是當某個應用程式加入供應服務機制時，它的關聯資訊即被新增至關聯屬性中，如圖 2-11 所示，當 Oracle Database、Sun One Directory 兩個系統與 Novell eDirectory[15](身分管理中心)連結時，在 eDirectory CN=931310 的 Entry 內，其關聯屬性即有兩筆資料，以供作 Synchronization Engine 進行供應服務的依據。圖 2-12 為一個具有兩個關聯屬性的 Entry 範例。

2.2.2 Publisher/Subscriber Processing

從應用程式到目錄的資料流稱為 Publisher Channel，也就是從應用程式資料異動事件發生，經 Synchronization Engine 轉換命令至目錄進行資料異動同步的所有步驟處理流程，可視為應用程式發佈變化到目錄中；而從目錄到應用程式的資料流稱為 Subscriber Channel，是用來將目錄物件異動事件，經 Synchronization Engine 轉換命令至應用程式進行資料異動同步的所有步驟處理流程，亦可視為應用程式訂閱從目錄中發生的變化(參考圖 2-3)。

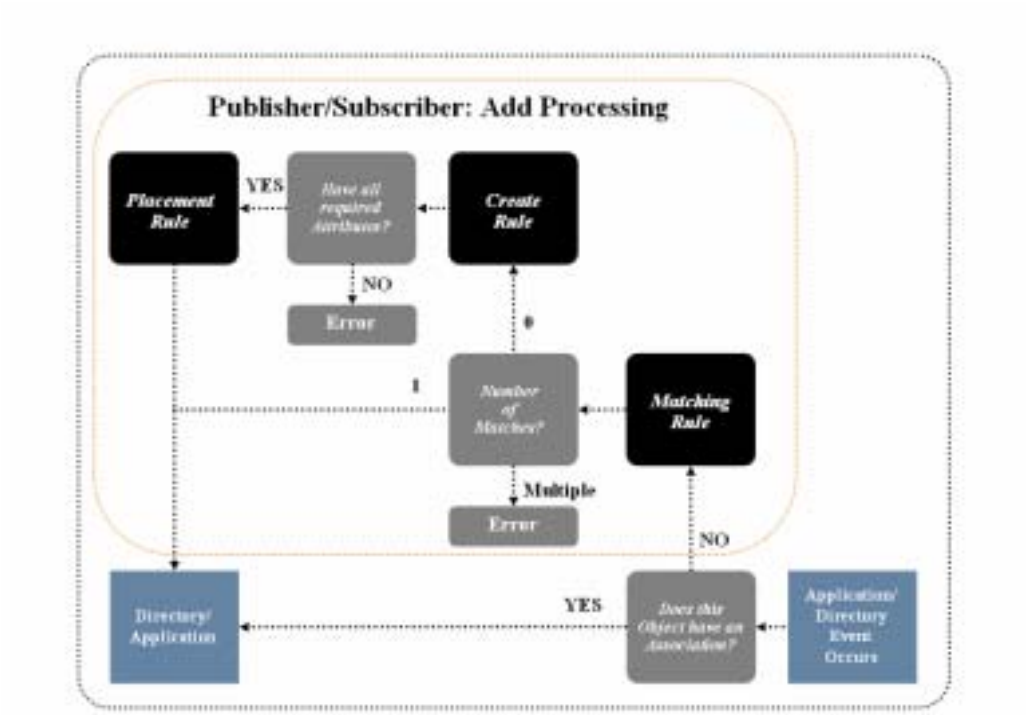

圖2-13 Publisher/Subscriber: Add Processing示意圖

圖 2-13 Add Processing 表示 Publisher/Subscriber 依據關聯屬性存在與否(參考圖 2-11 與圖 2-12)，來判斷應用程式物件與目錄物件的關聯性，以進行一系列的事件處理，其步驟說明如下：

一、 應用程式(或目錄)發生資料變更事件，Publisher/Subscriber 即尋找目錄是否存在與此變更物件相關的關聯屬性，若在目錄中找到關聯屬性，即對目錄(或應用程式)內的對應物件進行資料同步動作。

二、 若目錄內並未存在任何相關的關聯屬性，即以 Matching Rule 比對目錄(或應用程式)內物件，以確認目錄(或應用程式)內是否存在未設定關聯的對應物件。

三、 若在目錄(或應用程式)內找到一個對應物件，即進行該物件的資料同步，並新增關聯屬性至目錄物件中，作為後續物件對應的關聯。

四、 若在目錄(或應用程式)內未找到任何對應物件，則表示此為新增物件事件，需透過 Create Rule 檢查欲提供給新增物件所需的屬性是否完整。

五、 若所有提供給新增物件所需的屬性完整，即使用 Placement Rule 確認新增物件在目錄(或應用程式)內的正確存放位置，進而建立新物件，並在目錄物件內加入此對應關聯屬性，作為後續物件對應的關聯。

Subscriber Channel 在處理目錄物件的變化事件時，是透過讀取 Event Cache[14,17,18](目錄物件變化事件暫存器)中的事件，來進行後續依照 Policies 執行的一系列動作(參考圖 2-3)。使用 Event Cache 的目的在存放已經 Filter(參考第 2.1.1 節)過濾的目錄變化事件，所有 Event Cache 中的事件都將一直存放在 Cache 中，直到該事件已經

Synchronization Engine 成功完成應用程式的同步作業為止，如此可確保任何的事件不會因為供應服務的中斷而遺失，造成無法執行應用程式同步作業，使企業身份識別資料不一致。

Publisher Channel 在處理應用程式的變化事件時，則是透過讀取 Log(可視為應用程式變化事件暫存器)的方式(參考第 2.1.2 節)，來進行後續依照 Policies 執行的一系列動作(參考圖 2-3)，其目的則如同 Subscriber Channel 的 Event Cache 一樣，都是在確保任何 Publisher Channel 的變化事件不會因為供應服務的中斷而遺失，造成無法執行目錄物件同步作業，使企業身份識別資料不一致。而上述 Log 的資料格式與型態則和應用程式有關，例如應用程式若為資料庫系統，則其 Log 稱為 EventLog，是以 Table 的方式來儲存事件，而應用程式若是 LDAP，則其 Log 稱為 ChangeLog。

第三章 系統架構實作

本章以東海大學為對象，實作導入以目錄服務為基礎的資訊系統架構解決方案，並說明系統建置所使用的資訊平台特性與優點，最後詳細說明本論文架構的導入過程與方法。

3.1 實作環境

本論文將以 Oracle Database(東海大學校務行政系統資料庫)及 Sun One Directory(提供東海大學個人資訊入口網站 MyTHU 與師生資訊系統使用者身份驗證之目錄服務伺服器)連結 Novell 公司的安全身分管理解決方案(Novell Secure Identity Management Solution)產品：Novell eDirectory 與 Novell Identity Manager 來建置一個以目錄服務為基礎的資訊系統架構。

3.1.1 Novell eDirectory

eDirectory 是 Novell 公司所設計並支援 LDAP 的目錄服務伺服器，這項產品源自 Novell Directory Services(NDS)，而 NDS 則是 Novell 於 1994 年根據目錄服務 X.500 規格所發展出來並植基於 Novell Netware 網路作業平台上的產品。Novell eDirectory 有以下幾個特性：

一、 跨平台支援(Cross-Platform)：

eDirectory 可以安裝於 Solaris、Linux、AIX、Netware、Windows NT 與 Windows 2000 等作業平台執行。

二、 可擴展性(Highly Scalable)：

eDirectory 可以建置 10 億個物件。

三、 開放的標準(Open Standards-Based)：

eDirectory 支援 LDAP，所以任何支援 LDAP 的應用程式，皆

可透過 LDAP 來與 eDirectory 溝通。

四、 可靠性(Reliable)：

eDirectory 具有分散式容錯體系結構，可以確保即使一個或多個目錄伺服器因網路中斷，仍可提供目錄資料的存取服務。

五、 安全性(Secure)：

eDirectory 具備安全保護措施，以對目錄資料提供安全存取與管理。

以 eDirectory 作為企業使用者身分識別資料的儲存目錄，可結合 Novell Identity Manager(參考第 3.1.2 節)來實現企業眾多應用程式的身分識別資料分享與異動同步，而有關物件對應的關聯屬性，也同時存放於 eDirectory 物件中，作為供應服務執行應用程式資料異動同步的依據。

3.1.2 Novell Identity Manager

Identity Manager 是 Novell 公司所提出的超目錄服務解決方案，是以 eDirectory 目錄服務為基礎，並於其中建立關聯屬性，然後透過 XML 的應用，來實作為一個資料共享服務機制。

DirXML 包含三個主要關鍵元件，說明如下：

一、 DirXML Engine：

DirXML Engine 控制著 eDirectory 和某一應用程式之間的資訊同步；或者說，DirXML Engine 所控制的是所有與其連結的應用程式和 eDirectory 間的資訊同步，其中 eDirectory 可視為企業使用者網路資料交換中心的角色。另外，DirXML Engine 可運作在 eDirectory 目錄樹中的一個或更多伺服器上，而每個 DirXML Engine 都可能包含一個或多個 DirXML Driver 分別與各應用程式溝通，以進行資料交換，如圖 3-1 所示。

DirXML Engine 在 Subscriber/Publisher Channels(參考圖 2-3 與

第 2.1.1 節說明)扮演關鍵角色，主要針對資訊(轉換為 XML 文件)在 Channels 的傳送過程，以 Filter 過濾同步所需的類別屬性，並根據一系列 Rules、Stylesheets 作 XML、XSLT 轉換，以進行任何應用程式物件的相對應變化事件處理。

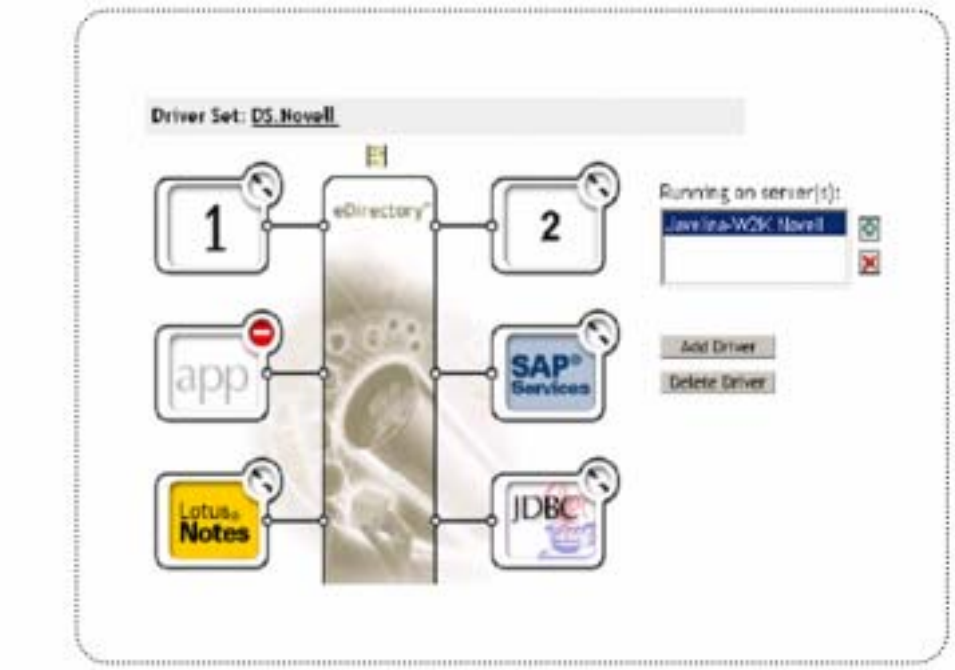

圖3-1 透過DirXML與eDirectory連結的應用程式示意圖

二、 DirXML Driver：

DirXML Engine 藉由 DirXML Driver(參考圖 2-9 與第 2.1.2 節說明)來與應用程式(包括 SAP、PeopleSoft、Exchange、Lotus Notes、eDirectory、Active Directory、LDAP 與 Database 等系統)溝通。儘管 DirXML 已包含多種應用程式的 Driver，但 DirXML 開發工具仍可提供系統管理員自行建立客製化的 DirXML Driver 以連接其他的應用程式。

本論文於東海大學的導入實例中，分別使用 DirXML Driver for JDBC、DirXML Driver for LDAP 與 Oracle Database、Sun One Directory 溝通。

三、 eDirectory Interface：

eDirectory Interface[14]是用來偵測(detect)eDirectory 所產生的物件變化事件，而為了確保每個事件都能經 DirXML Engine 處理且成功完成應用程式的資料同步，每個變化事件都會被儲存於 Event Cache(參考第 2.2.2 節)直到應用程式的同步作業完成為止，這樣將不會因網路中斷或者任何驅動程式異常而造成資料的遺失，影響企業資料的一致性。

3.2 實作導入

以下兩小節將介紹本系統於東海大學實作導入時，關於人員報到至離校期間，校園資訊系統帳號使用生命週期管理的建議作法，同時配合校務行政系統資料庫與目錄服務 Schema 規劃、供應服務規則定義與 User Self-Services 建置，構成一個完全以目錄服務為基礎的東海校園資訊系統架構，如圖 3-2 所示。

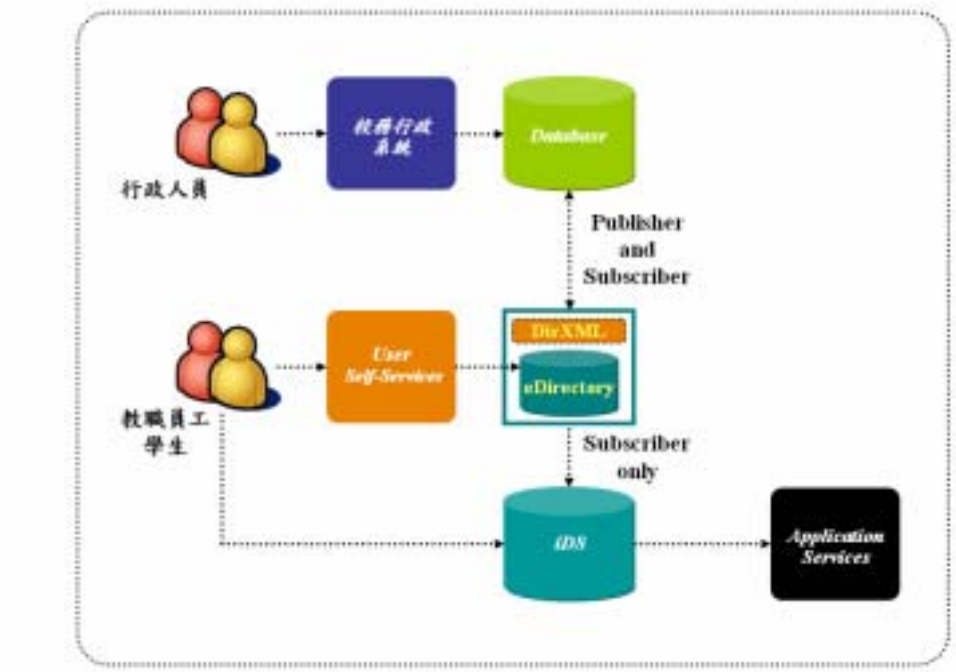

圖3-2以目錄服務為基礎的東海資訊系統架構示意圖

3.2.1 系統設計

本論文的系統架構是以 eDirectory 為基礎，作為使用者身分識別資料的中央儲存機制，並透過 DirXML 實現與資料庫及其他目錄服務系統的資料同步，在實作時，需先完成「User Account Lifecycle Management Plan(使用者帳號生命週期管理計畫)」及「Database and Directory Schema Design(資料庫與目錄 Schema 規劃)」，主要目的在確認使用者帳號從建立到刪除的完整作業與資料流程，同時對資料庫表格欄位至目錄結構的設計做妥善的規劃，最後才真正導入 DirXML 來定義一系列 Policies，讓供應服務正確的運作。以下分別說明：

一、 User Account Lifecycle Management Plan：

現階段的東海大學人事室與註冊組雖然分別使用人力資源管理系統及學生學籍管理系統來管理全校教職員工學生的個人基本資料，但由於這些資料一直無法即時準確的與其他如 email、撥接(Radius)、目錄服務等系統整合，以及欠缺全校使用者帳號管理的完整規劃，所以各系統長久以來即分別獨立運作，且使用者帳號資料仍持續重複的建置於各系統中，而各系統帳號密碼的管理對電算中心系統管理員來說，絕對是最煩雜、無趣、沒有效率且浪費時間的工作，不僅如此，一個帳號處理的疏忽都可能造成資安災難；另一方面，使用者也因為必須記住眾多的系統帳號密碼而傷透腦筋，甚至還必須時常向電算中心請求協助處理其帳號密碼忘記的問題。

如圖 3-3 所示，新進人員到校時，必須先至人事室(學生須至註冊組)報到並領取其校園身分證件，然後他必須帶著該證件至電算中心申請電子郵件帳號，若該員負責業務需使用校務系統，則必須再向電算中心申請校務系統帳號，後續該員若忘記電子郵件帳號或密碼，則須再度前往電算中心進行帳號查詢和密碼重設，若他要

修改個人通訊資料則須至人事室(或註冊組)辦理，最後當他至人事室(或註冊組)辦理離校手續時，電算中心若未能即時獲得該員離職訊息，以刪除其各系統使用帳號與權限，可能發生嚴重的資安漏洞。

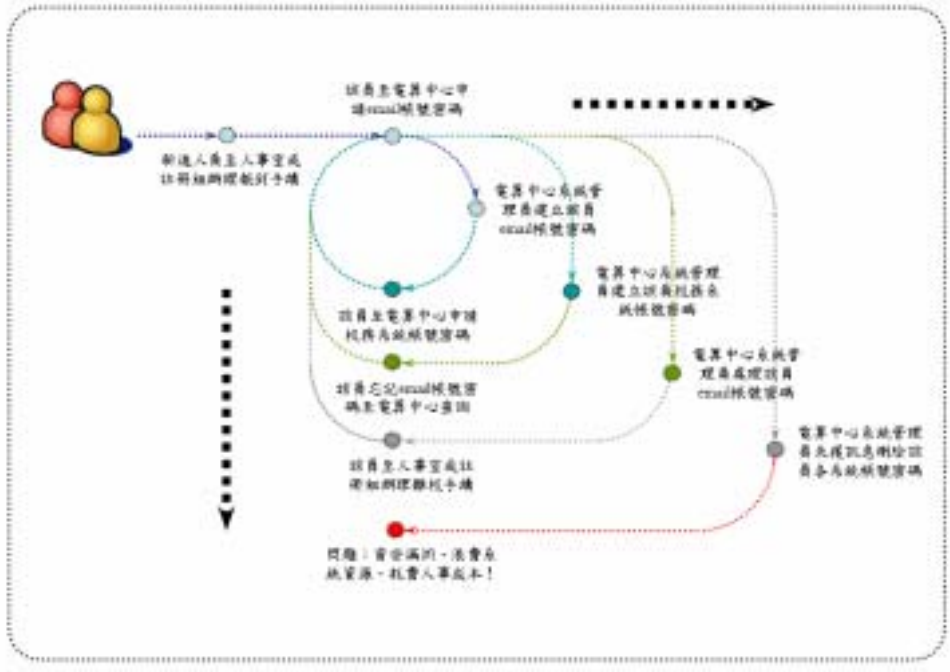

圖3-3 東海校園帳號密碼原有處理流程示意圖

圖 3-3 說明，當企業所有系統使用者身分識別資料未能互相分享，以及帳號使用週期未能謹慎管理時，即形成企業潛在資訊安全風險與資源浪費，為了解決此一問題，建立一個東海大學使用者帳號生命週期管理方案是必須的，也就是強化現行人力資源管理與學生學籍管理系統功能，確立校務行政系統資料庫人員基本資料為未來系統使用者帳號來源，然後導入 DirXML 來連結使用者身分識別資料的中央儲存機制(eDirectory)，以提供供應服務與其他應用程式進行所有相關身分識別資料的分享與異動同步，另外建置的 User Self-Services 則提供使用者於線上進行帳號註冊、密碼設定、帳號密碼查詢、修改個人資料與通訊錄查詢，可有效減少各相關業務服務台之人力作業成本，並方便使用者自我管理個人身分識別資料。

如圖 3-4 所示，過去由電算中心所負責的使用者帳號建立、維護與刪除等管理工作，已移往人事室與註冊組配合教職員工學生報到或離校手續一併辦理，由於包含帳號的所有使用者身分識別資料的維護功能早已整合於人力資源管理與學生學籍管理系統，所以對行政人員來說，帳號的處理並未增加其現有工作的負擔，而後續所有相關系統使用者帳號資料的同步則由 DirXML 自動進行；另外，因為建置了 User Self-Services Center(USSC，個人資訊服務中心) ，如圖 3-5 所示，對所有教職員工學生來說，自行解決個人帳號、密碼或基本資料修改等問題也遠比過去必須尋求電算中心或相關單位服務台處理要來的快速且方便(參考第 2.1.3 節)。

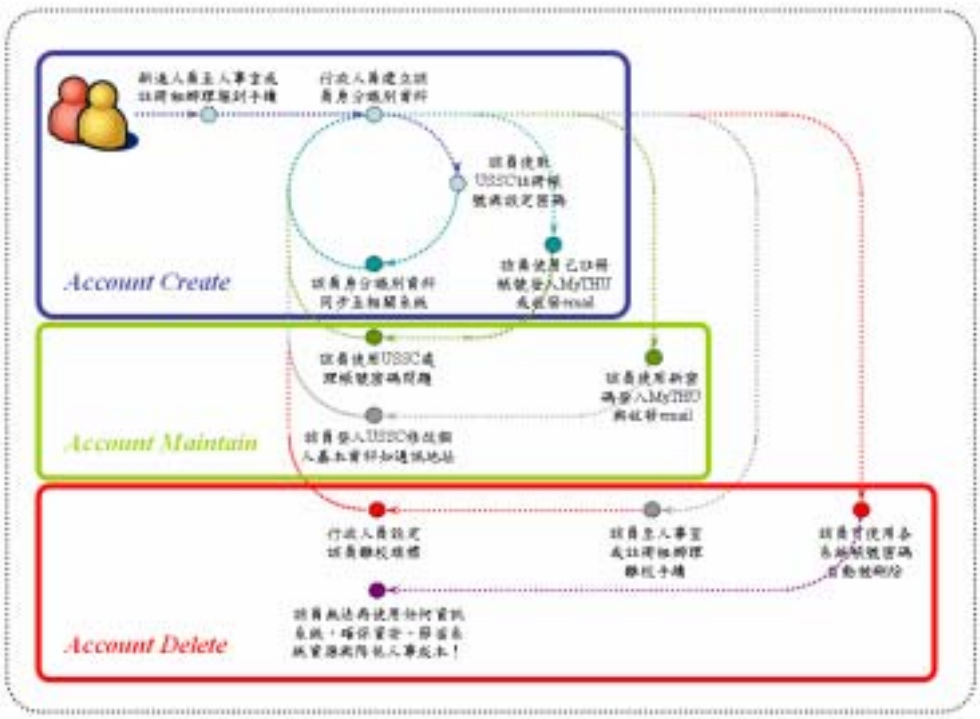

圖3-4 東海校園帳號密碼處理流程改善作法示意圖

以下說明圖 3-4 系統帳號密碼處理流程步驟：

一、 新進教職員工學生至人事室或註冊組辦理報到手續。

二、 行政人員使用人力資源管理系統或學生學籍管理系統輸入該員所有個人身分識別資料。

三、 該員身分識別資料經 DirXML 從校務行政系統資料庫同步至 eDirectory，再同步至其他應用程式如 Sun One Directory，此時使用者尚未至 USSC 註冊帳號與設定密碼，故仍無法登入使用各系統。

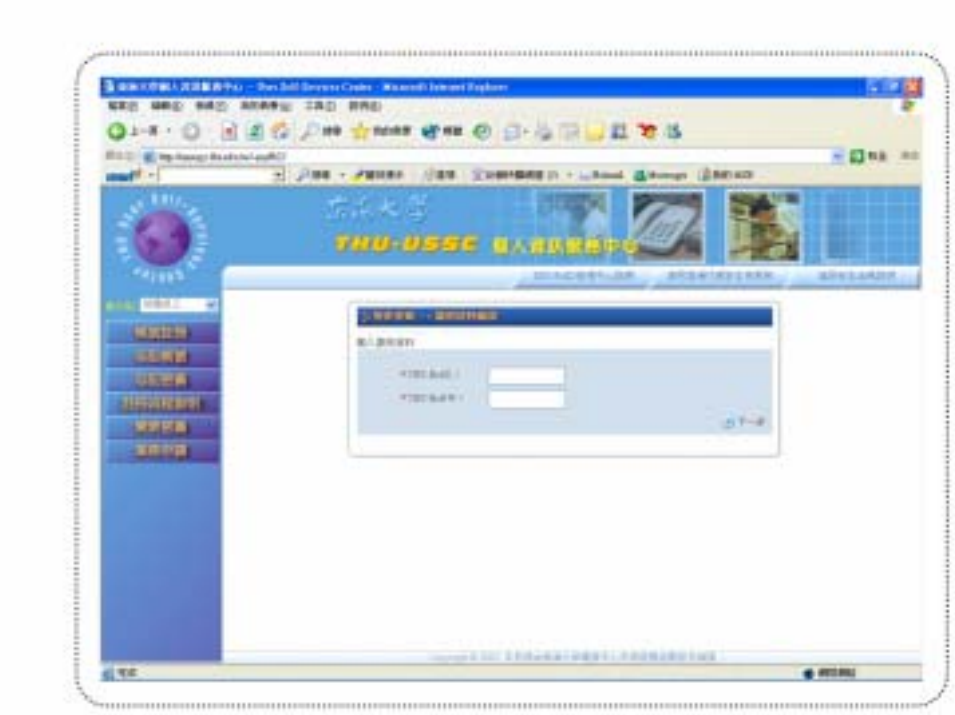

圖3-5 東海大學USSC

四、 該員使用 USSC「Register Account」功能完成密碼與密碼提示語及答案之設定，所設定的密碼隨即透過 DirXML 同步至其他應用程式如 Sun One Directory。

五、 教職員工學生以帳號及自行設定之密碼登入使用學校各資訊系統如 MyTHU、師生資訊系統、email 與 Radius 等應用程式服務。

六、 後續教職員工學生個人資料可由行政人員或個人自行修改，異動資料同樣透過 DirXML Publisher/Subscriber Channels 自動同步到相關應用程式如資料庫、LDAP。

七、 教職員工學生至人事室或註冊組辦理離校手續時，行政人員

使用人力資源管理系統或學生學籍管理系統對該員個人資料記錄設定離校旗標，此資料變化事件經 DirXML 同步至 eDirectory 及其他應用程式進行該員帳號之刪除，從此該員不再擁有任何學校資訊系統帳號及使用權限。

二、 Database and Directory Schema Design：

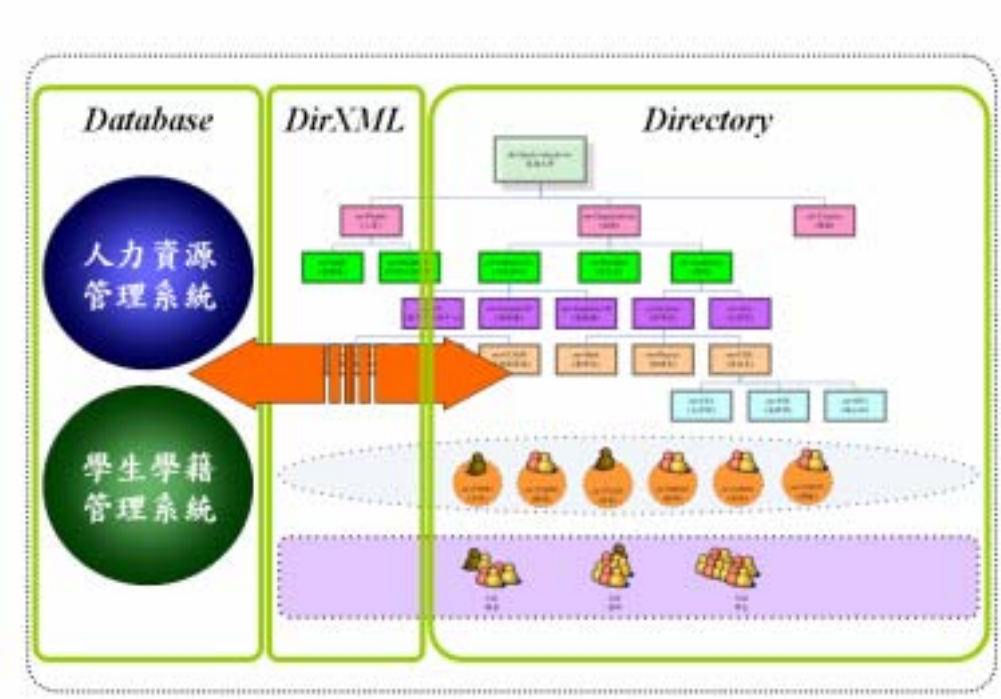

圖3-6 校務行政系統資料庫與目錄服務關係示意圖

欲將校務行政系統資料庫內有關人員、組織資料藉由 DirXML 同步至 eDirectory 及其他應用程式前，必須先確認資料庫與目錄 Schema 的對應關係，如圖 3-6 所示。圖 3-7 是本論文系統架構規劃時，目錄內組織 Etnry(人員 Entry 另以同樣方式處理，省略說明)與資料庫 View Record 的 Schema Mapping，首先是分析校務行政系統資料庫內有關組織的所有 Table 及相關欄位資料，並建立一個 View(ds_view_organize)，以供後續目錄組織 Entry、Attribute 建立或異動之對應，參考圖 3-8，接著將 View 內的所有 Records 分別對應建立 eDirectory 內組織 Entry 於正確位置，如系統發展組(CCSDS)

所在 DIT 位置的父節點與父父節點分別是 CC 及 AdminUnits，參考圖 3-9，最後則是 eDirectory 與 Sun One Directory 組織 Entry 之對應，參考圖 3-10。

1 Oracle Database – *View (ds_view_organize)*

dep_cd	Description	OU	POU	PPOU	Business Category
2700	電子計算機中心	CC	AdminUnits	Organizations	Computer Center
2701	系統發展組	CCSDS	CC	AdminUnits	System Development
2702	網路技術組	CCNTS	CC	AdminUnits	Network Technology
2703	教學及研究支援組	CCIRS	CC	AdminUnits	Instruction Research

2 Novell eDirectory – *OU*

OU	CC	CCSDS	CCNTS	CCIRS
Business Category	Computer Center	System Development	Network Technology	Instruction Research
Description	電子計算機中心	系統發展組	網路技術組	教學及研究支援組

3 Sun One Directory – *ou*

ou	CC	CCSDS	CCNTS	CCIRS
businessCategory	Computer Center	System Development	Network Technology	Instruction Research
description	電子計算機中心	系統發展組	網路技術組	教學及研究支援組

圖3-7 Database and Directory Schema Mapping

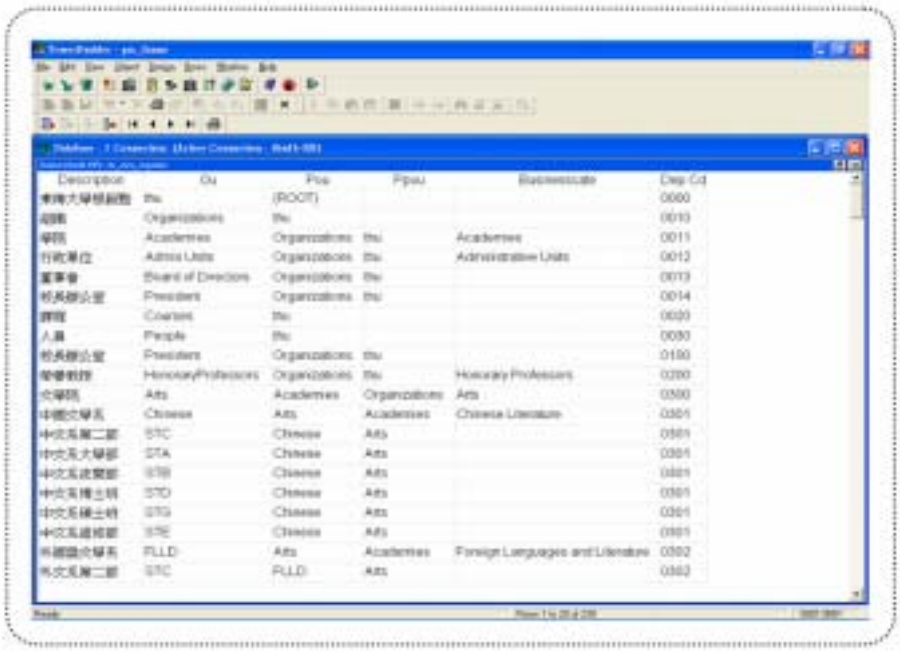

圖3-8 View(ds_view_organize)

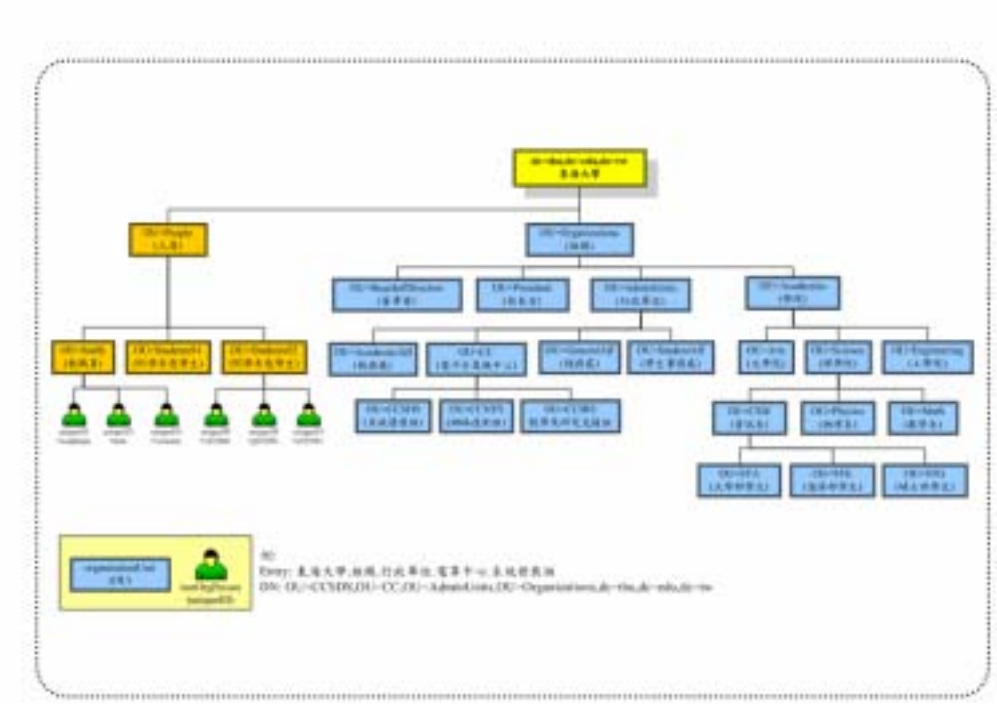

圖3-9 Novell eDirectory DIT Architecture示意圖

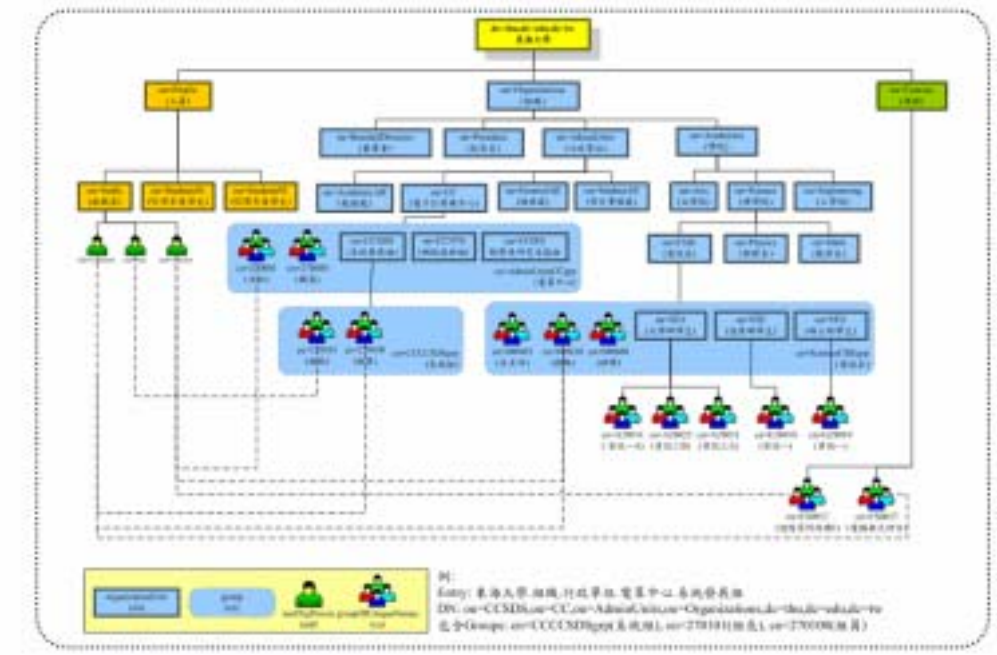

圖3-10 Sun One Directory DIT Architecture示意圖

圖 3-11 表示透過 DirXML 的設定，eDirectory 已與 Oracle Database(JDBC)、Sun One Directory(LDAP)連結並執行物件變化事件 Publisher 及 Subscriber 之異動同步作業，有關 Policies 之定義將

於下節說明。

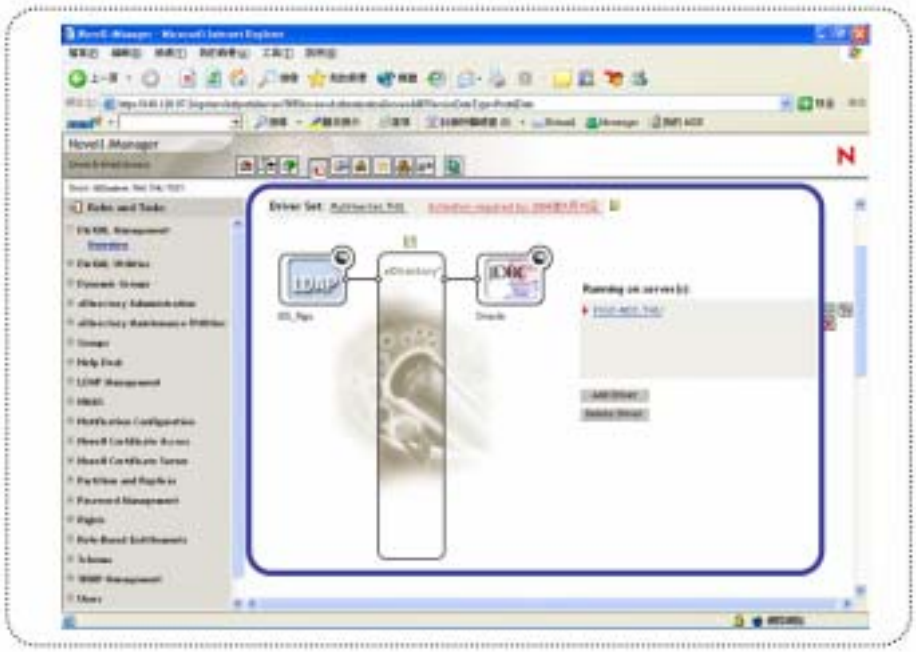

圖3-11 以DirXML連結eDirectory、DB與LDAP示意圖

3.2.2 執行畫面

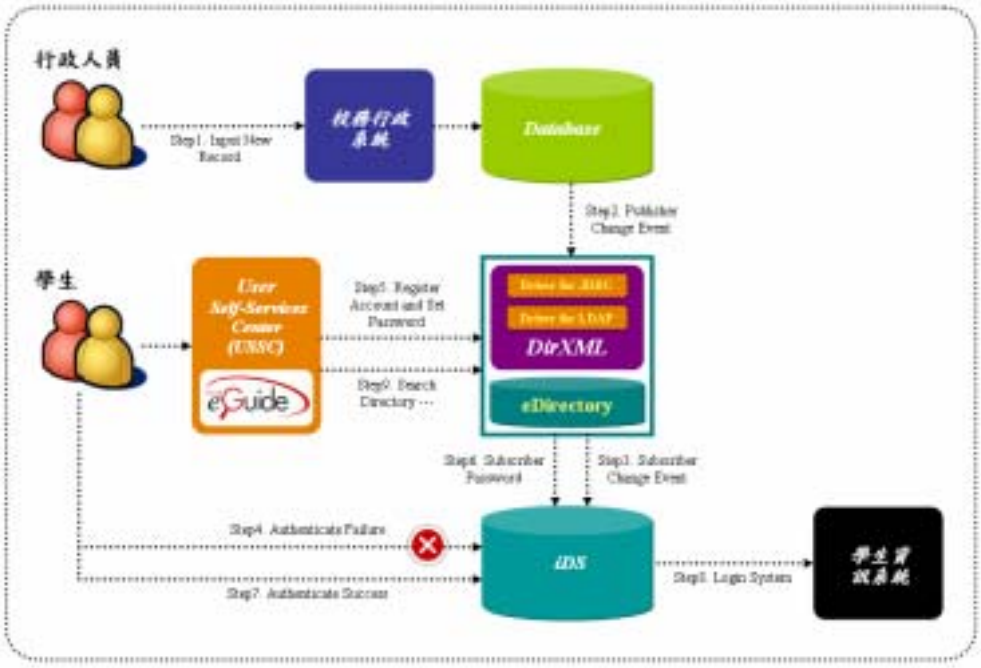

圖3-12 實作系統供應服務劇情說明示意圖

以下將從行政人員於學生學籍管理系統加入一筆新生基本資料的

劇情(Scenario)開始，如圖 3-12 所示，說明本系統架構的同步機制與相關系統之執行畫面。

圖3-13 於學生學籍管理系統新增一筆學生基本資料

某位新生報到入學時,註冊組行政人員透過學生學籍管理系統輸入該生的學籍資料，學號為 931319，如圖 3-13，這筆資料會存入校務行政系統資料庫的學生學籍資料主檔，其 Table 名稱為 THAASTU。為了使 DirXML 能偵測 THAASTU 有否執行資料新增、修改或刪除動作，當 THAASTU 的資料有異動時，會觸發一個名為 T_THAASTU 的 Trigger。

透過 Trigger 將該筆學號為 931319 的 Record 記錄到另一個稱為 EVENTLOG 的 Table。EVENTLOG 記錄了資料被異動過的 Table Name、Primary Key、Event Type、Status 等資訊，所以在此例中，被異動過的 Table Name 為 VIEW_THAASTU(View 的名稱)，Primary Key 為學號 PK_STUDNO=931319，欄位 Event Type 為 5 代表該筆資料為新增(Event Type 為 6 代表修改，4 代表刪除)。

DirXML 會根據 JDBC Driver 中 Publisher Polling Interval 參數設定的時間(已設定為 5 秒)，每隔 5 秒檢查一次 EVENTLOG，當發現 EVENTLOG 目前有一筆新增的記錄 VIEW_THAASTU.PK_STUDNO= 931319，即至 VIEW_THAASTU 取得需同步到 eDirectory 的資料。EVENTLOG 中已經 DirXML 處理過的記錄若成功同步至 eDirectory，則其 Status 欄位會由 N(代表 New)改為 S(代表 Success)，並將此筆記錄從 EVENTLOG 中刪除。

由於 THAASTU 包含許多欄位，但並不是每個欄位資料都需要與 eDirectory 同步，所以 Create 一個與 THAASTU 相對應，稱為 VIEW_THAASTU 的 View 是一個方便的作法，VIEW_THAASTU 是把 THAASTU 中需要同步到 eDirectory 的欄位 select 出來，包含學號(STUD_NO)、姓(LNAME)、聯絡電話(TEL_NO)、帳號(USERNAME)、姓名(STUD_CNAME)、身分證號碼(ID_NO)、戶籍電話(TEL_NO1)、戶籍地址(FOREVER_ADDRESS)等欄位。

DirXML 透過 JDBC Driver 取得學生學籍資料主檔 THAASTU 新增的一筆資料後，即根據所設定的 DirXML Driver for JDBC Publisher Policies，Create 相對應的 USER Object 至 eDirectory。Publisher Policies 的設定流程如圖 2-3，步驟分述如下：

一、 Schema Mapping：如圖 3-14 所示，首先必須設定 VIEW_THAASTU 對應到 eDirectory 的 USER Object，也就是學生的學籍資料在 eDirectory 是要以人的屬性來儲存。然後設定 VIEW_THAASTU 同步到 eDirectory 的欄位所對應 USER Object 的屬性，例如學號(PK_STUDNO)對應到屬性 CN、聯絡電話(PHONENO)對應到屬性 Telephone Number，姓(LNAME)對應到屬性 Surname 等。在 DirXML 中所有的 Policies 設定皆

以 XML 的格式來儲存(可參考圖 2-8)，每個 Policy 也是 DIT 中的一個 Entry，並以屬性 XmlData 儲存該 Policy 的 XML 格式內容，如圖 3-15，Schema Mapping 的儲存位置為 cn=MappingRule, cn=Oracle,cn=MyDriverSet,o=THU。

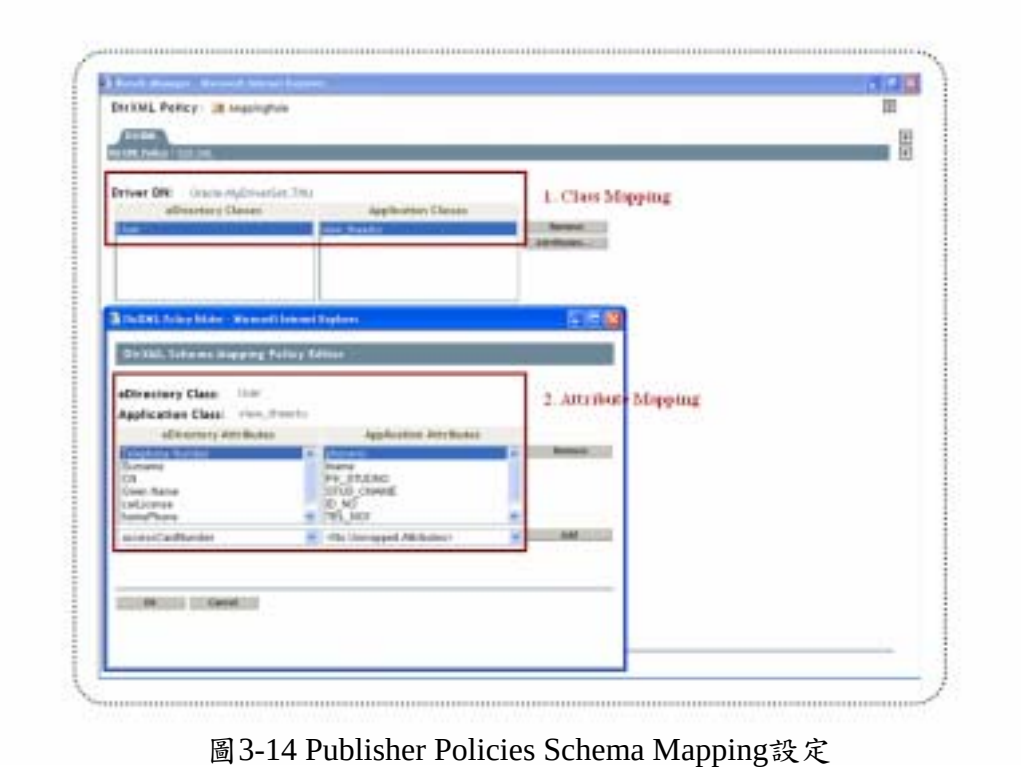

圖3-14 Publisher Policies Schema Mapping設定

二、 Filter：Filter 用來設定與 VIEW_THAASTU 欄位相對應的每個屬性是否允許 Publisher 或 Subscriber 同步。如圖 3-16，屬性 Telephone Number 的 Publish 設為 Synchronize，表示學生的聯絡電話 Telephone Number 是由註冊組行政人員根據大考中心的該生資料所輸入，可透過 DirXML 同步到該生 eDirectory 的 USER Object 中；另外若於 Filter 將 Telephone Number 屬性的 Subscribe 設為 Synchronize，則表示學生可以透過使用 USSC 修改自己的聯絡電話，讓在 eDirectory 中被修改的聯絡電話資料同步至資料庫的學生學籍資料主檔 THAASTU。有關 Filter 設定的 XML 格式文件可參考圖 2-2。

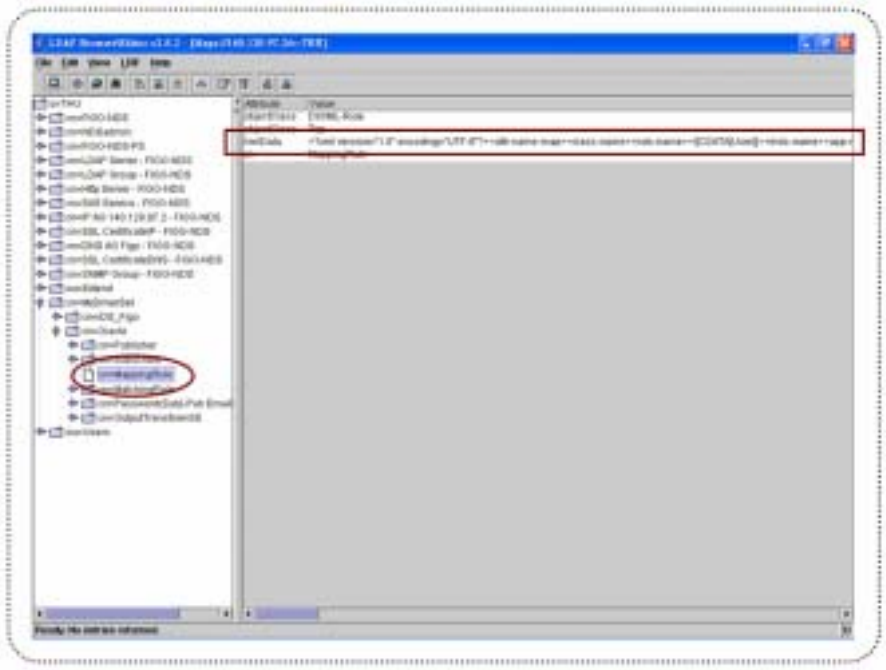

圖3-15 Schema Mapping Policy以XML格式存於DIT XmlData屬性

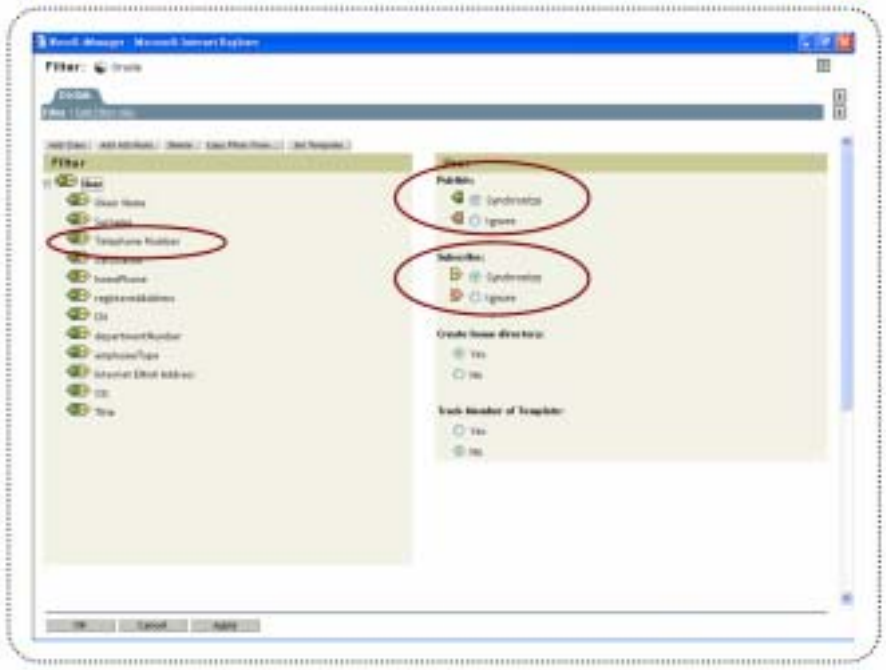

圖3-16 Publisher Filter設定

三、 Matching Rule：Create eDirectory cn=931319 USER Object 之前，必須先檢查 eDirectory ou=Students93,ou=Users,o=THU 的 subtree 下是否已經有該筆 cn 存在，其 Policy 以 XML 格式

設定如圖 3-17，若該筆 cn 已存在，則 Publisher 無法同步成功，否則繼續進行下一個 Create Rule 步驟。

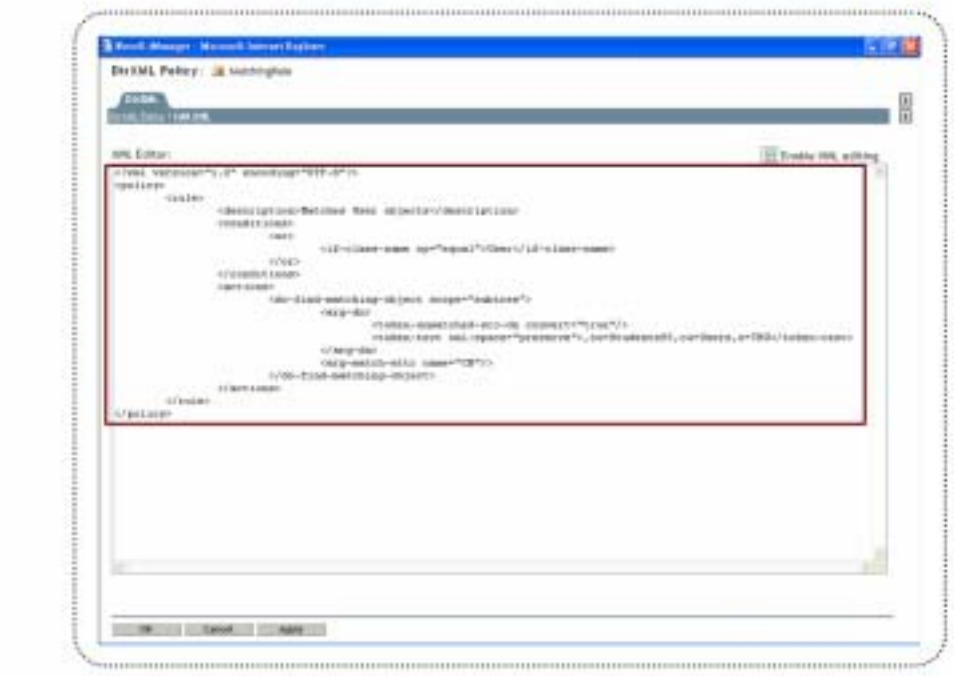

圖3-17 Publisher Policies Matching Rule設定

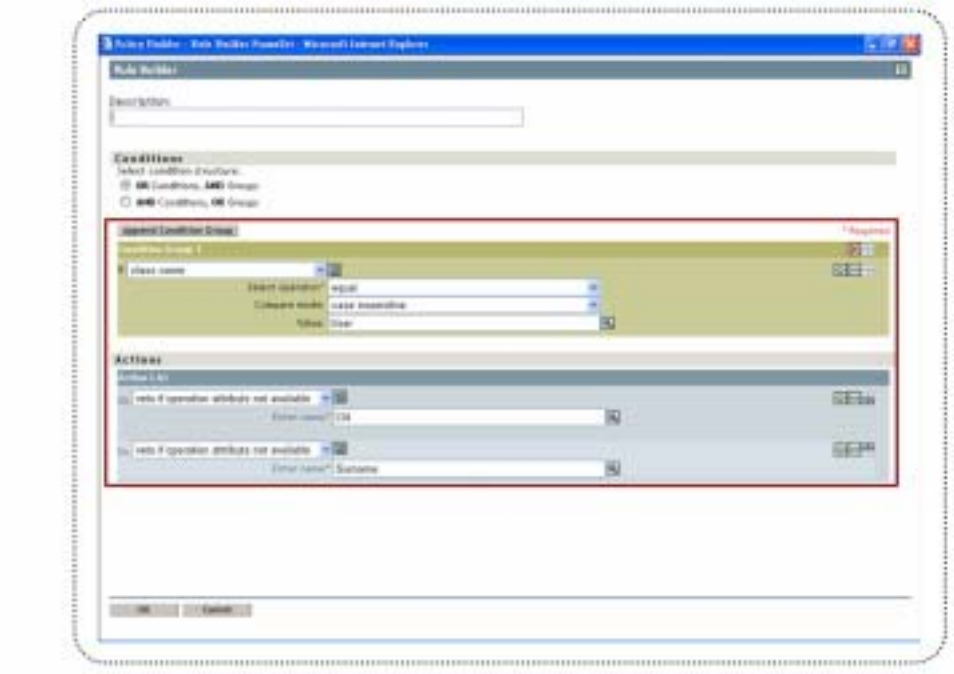

圖3-18 Publisher Policies Create Rule設定

四、 Create Rule：代表 Create USER Object 時所必須滿足的條

件，如圖 3-18 所示，若欲 Create 的 USER Object CN 與 Surname 屬性沒有任何值，則無法 Create 該 USER Object；否則繼續進行下一個 Placement Rule 步驟。

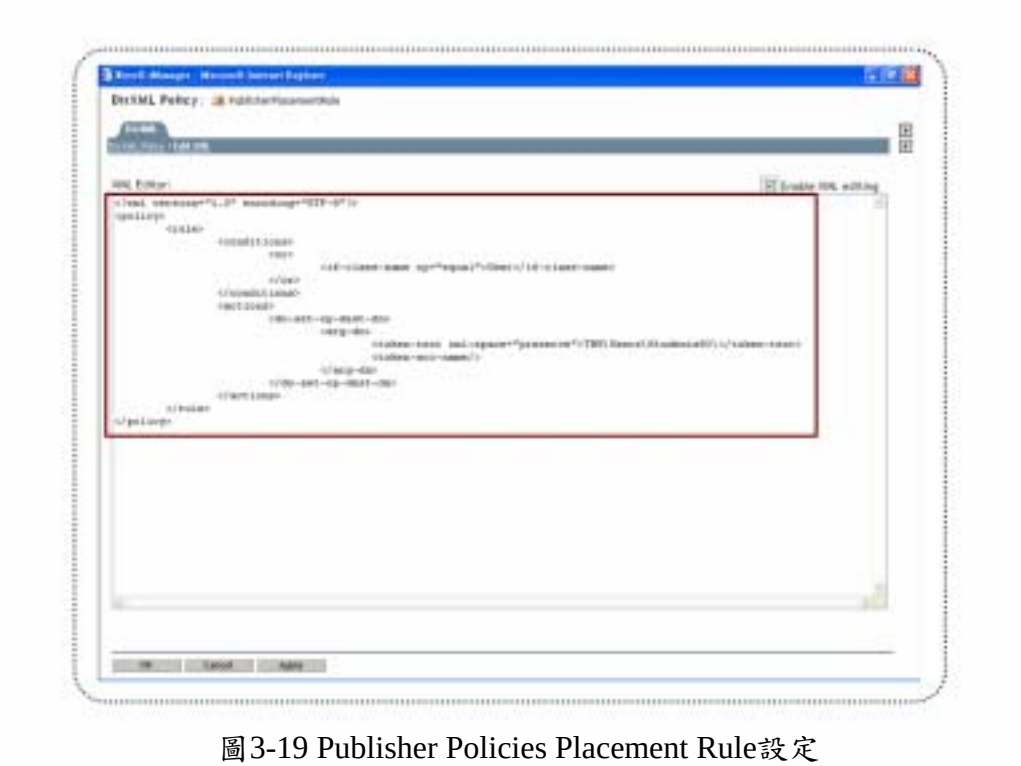

圖3-19 Publisher Policies Placement Rule設定

五、 Placement Rule：用來設定前一步驟所欲 Create 的 USER Object 應置於 eDirectory DIT 中的哪個路徑下，如圖 3-19 所示，學號 931319 將 Create 於 ou=Students93,ou=Users,o=THU 下。

經過一系列 DirXML Publisher Policies 處理後，透過 Ldapbrowser 工具連接 eDirectory，可於 ou=Students93,ou=Users,o=THU 路徑下找到 cn=931319 的 User Obect，如圖 3-20 所示，圖中方框內資料即為該生從學生學籍資料主檔 THAASTU 同步過來的屬性內容，如姓、姓名、聯絡電話、戶籍電話、戶籍地址、身分證號碼等；另外，此物件還包含兩個關聯屬性，屬性名稱為 DirXML-Associations，屬性值為 cn=Oracle,cn=MyDriverSet,o=THU#1#PK_STUDNO=931319, table=VIEW_THAASTU,schema=CENTER 與 cn=iDS_Figo,

cn=MyDriverSet,o=THU#1#uid=931319,ou=students93,ou=people,dc=thu,dc=edu,dc=tw，分別記錄此 USER Object 與 Oracle 資料庫學生學籍資料主檔 THAASTU 的 STUD_NO=931319 的關聯性，以及與 iDS 目錄的物件 uid=931319 的關聯性。

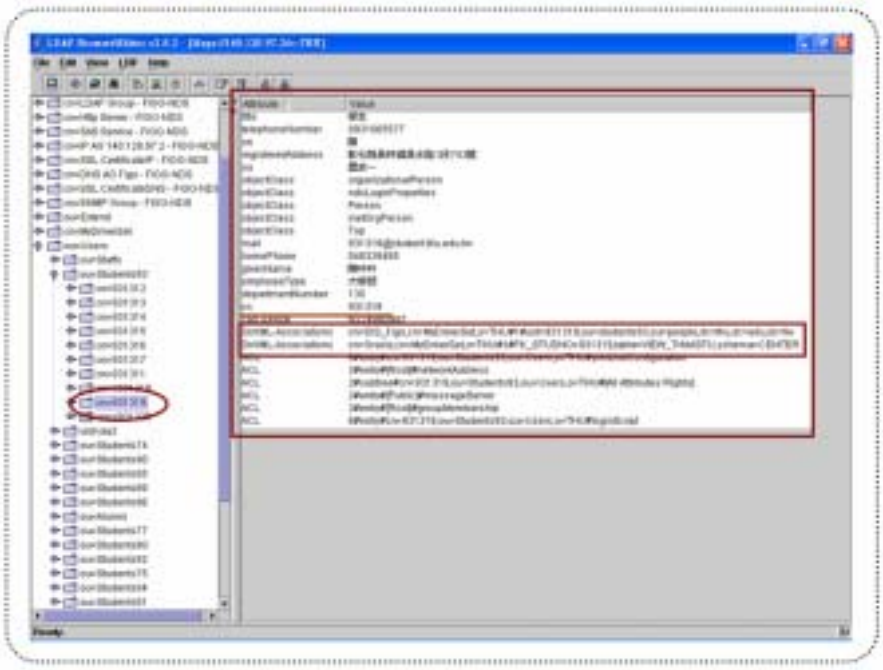

圖3-20 在eDirectory找到cn=931319之Entry

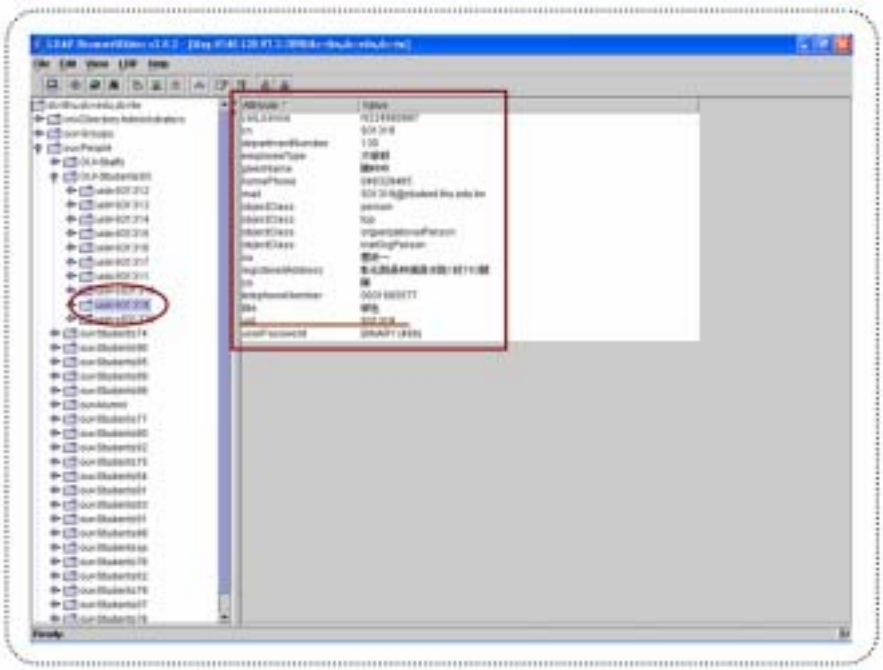

圖3-21 iDS uid=931319之Entry

iDS uid=931319 之 Entry 屬性內容可參考圖 3-21，其內容是經由 DirXML Subscriber Channel 從 eDirectory 資料同步而來。後續該名學號 931319 的新生透過 USSC 進行帳號註冊(包含密碼設定)後，即可使用所有以 eDirectory 為身分識別中心的相關系統，在此論文實作中，則包含 USSC(Based on eDirectory)與學生資訊系統(Based on iDS)。

第四章 結論及未來工作

本論文提出的系統架構，其貢獻在於：

一、 建立一個以目錄服務為基礎的企業資訊系統架構,而此架構的核心即是作為企業使用者身分識別資料中心的目錄服務，任何支援 LDAP 協定的應用程式都可與之整合,或者透過 DirXML Driver 來連接應用程式，是一個具有高度擴充與整合性的系統發展架構，有別於目前企業內使用者身份資料重複、分散且無法分享的儲存於各應用程式儲存庫中，可有效節省系統帳號管理成本與降低資安風險。

二、 建立使用者自我服務中心,提供使用者自行管理帳號密碼、修改個人基本資料與線上通訊錄查詢等服務機制，可提高使用者工作時效並降低企業資訊服務台處理前述問題的作業成本。

三、 將此架構實作於 Novell eDirectory 目錄伺服器與 DirXML 資料共享服務平台上，並和校務行政系統資料庫、MyTHU 個人資訊入口網站及師生資訊系統 Sun iDS 目錄伺服器連結，完成使用者身分識別資料異動的同步，後續並將整合其他應用程式如電子郵件、Radius 等網路服務(參考圖 2-1)，以進一步建置東海大學未來資訊整合與發展架構(參考圖 1-1)。

本系統架構未來發展：

一、 本系統架構可以加入包含 Automated Approval Process、Delegated Administration、Identity Audit、Role-Based Access Control 與 Single Sign On 等機制來實現企業員工零時差且能夠以一組帳號密碼做為身分驗證，並取得適當的系統執行權限來完成工作。

二、　本系統架構的企業員工身分識別資料是以目錄服務伺服器來存放，而企業員工入口網站的使用者身分驗證與存取權限也可透過目錄服務達成，換句話說，企業應該充分運用入口網站作為跨部門資訊整合與作業流程管理的前端 IT 基礎架構，而以目錄服務作為後端使用者身分如帳號、密碼、授權群組、系統使用權限等資訊的管理中心。

三、　建置以目錄服務為基礎的各種應用程式如軟體派送、資產管理、系統使用授權管理等 IT 資源自動化管理系統，以解決當前企業面對各種 IT 資源不斷增加、更新但資訊人員疲於奔命仍無法有效處理的問題。另外，隨著企業逐步建立目錄服務來統一管理員工身分資料，IT 部門也必須同時轉型為服務中心，以提供使用者自動化 IT 服務為目標，展現資訊部門在企業中的真正營運價值。

參考文獻

[1] http://www.openldap.org/doc/admin22/, OpenLDAP Foundation, OpenLDAP 2.2 Administrator's Guide.

[2] R. Weltman and T. Dahbura, LDAP Programming With Java, Addison-Wesley, January 2000.

[3] C. Walton, “eProvisioNing: Get Your Business in Hand,” Novell Connection, October 2001, pp.6-24.

[4] 蔣大偉編譯，Gerald Carter 著，LDAP 系統管理，美商歐萊禮股份有限公司台灣分公司，民 92 年 10 月。

[5] http://www.w3.org/TR/2004/REC-xml-20040204/, World Wide Web Consortium, Extensible Markup Language(XML) 1.0 (Third Edition) W3C Recommendation, February 2004.

[6] http://www.rfc-editor.org/rfc/rfc2251.txt/, Network Working Group, Lightweight Directory Access Protocol (v3), December 1997.

[7] http://www.w3.org/TR/2003/WD-xslt20-20031112/, World Wide Web Consortium, XSL Transformations (XSLT) Version 2.0 W3C Working Draft, Novenmber 2003.

[8] L. Kennard, “Too Many Diretories? Synch ‘Em With DirXML,” NetWare Connection, May 2000, pp.8-12.

[9] 陳建勳譯，E. T. Ray 著，XML 學習手冊，美商歐萊禮股份有限公司台灣分公司，民 90 年 5 月。

[10] 勞虎，無廢話 XML，兩隻老虎工作室 www.2tigers.net，民 88 年。

[11] L. Kennard, “Provisioning Access to Network Assets,” Novell Connection,

November/December 2002, pp.6-19.

[12]L. Kennard, “i-Login: One Net for Novell Employees,” Novell Connection, February 2002, pp.6-18.

[13]L. Kennard, “i-login: It’s One Net Live From Novell,” NetWare Connection, December 2000, pp.6-20.

[14]http://www.novell.com/, Novell, Inc., Novell Nsure Identity Manager 2 Administration Guide, January, 2004.

[15]http://www.novell.com/, Novell, Inc., Novell eDirectory Technical White Paper.

[16]L. Kennard, “Check Out That DirXML Engine,” NetWare Connection, May 2000, pp.16-20.

[17]M. E. McKell, “An Introduction to Novell’s DirXML,” Novell Appnotes, July 2000, pp.4-18.

[18]http://www.novell.com/, Novell, Inc., Novell DirXML Technical White Paper.

[19]http://www.novell.com/, Novell, Inc., Novell DirXML Driver for JDBC 1.6.2 Implementation Guide, January, 2004.