

國立交通大學

電機資訊學院 資訊學程

碩士論文

異質性的應用系統程式整合研究



A Probe on Heterogeneous Applications Integration

研究生：林怡慧

指導教授：陳登吉 博士

中華民國 九十四 年 七 月

異質性的應用系統程式整合研究

A Probe on Heterogeneous Applications Integration

研究生：林怡慧

Student：Yi-Hui Lin

指導教授：陳登吉 博士

Advisor：Deng-Jyi Chen

國立交通大學
電機資訊學院 資訊學程
碩士論文

A Thesis
Submitted to Degree Program of Electrical Engineering and Computer Science
College of Electrical Engineering and Computer Science
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Master of Science
in
Computer Science
June 2005
Hsinchu, Taiwan, Republic of China

中華民國九十四年七月

異質性的應用系統程式整合研究

學生：林怡慧

指導教授：陳登吉 博士

國立交通大學電機資訊學院 資訊學程（研究所）碩士班

摘要

現今社會上，各個公司行號或單位組織內部都充斥著許多的應用程式系統。使用者常會需要輸入許多組的帳號密碼來登入各個系統，不管是相關或不相關的；若遇上公司合併，更是加深使用者的困擾。

加上目前科技日新月異，使用者已經不再只是單純使用個人電腦來使用相關應用程式。人們可以利用許許多多的裝置，例如：個人桌上型電腦、手提電腦、PDA、手機或者是其他行動裝置來連結網路或使用相關應用系統。使用者面對這樣的情況，更是需要記住更多的系統連結方式或者是更多組的帳號密碼。

因應上述的情況，對於系統管理者而言在系統的維護及開發上，更是耗神費力。除了針對異質性的應用程式的維護管理外，更要費心管理異質性的資料內容以及資料庫的認證問題。這對維護者而言是非常費時的工作。

在這篇論文中，我們提出了幾個方法來解決這一類的問題：單一登入窗口及不同內容的早期繫結(Content early binding)可減低維護人員的負擔並提供使用者一個簡單的進入點；數位式權限管理則可有效管理使用者並保護富創意的資料內容。

A Probe on Heterogeneous Applications Integration

Student: Yi –Hui Lin

Advisor: Dr. Deng-Jui Chen

Degree Program of Electrical Engineering Computer Science
National Chiao Tung University

Abstract



Nowadays, every company or organization has many kinds of different application systems. Users usually need to remember several combinations of user account and password to logon different kinds of application systems, even the related functions. When different companies or organizations are merged, users will suffer more painful work. Also, as the technology evolves daily, users may use many different kinds of devices (for example: desktop personal computer, notebook, PDA, mobile phone, or other mobility devices) to access web site or other related application systems. Users usually need to remember more IP address links or many combinations of account and password information.

For system managers, they usually need to spent more efforts on the system maintenance in order to support these different needs. System manager not only needs to maintain the heterogeneous applications, but also the heterogeneous contents and databases authentication. It is a very time-consuming activity.

In this thesis study, these problems mentioned above are addressed. Specifically, we investigate single-sign-on approach and content's early binding approach to reduce the maintenance efforts and provide an easy way for users to logon these systems. Also a DRM mechanism is provided to achieve access control and created digital contents protection.

誌 謝

本論文承蒙指導教授陳登吉老師的耐心指導與教誨之下，得以順利完成，在此對老師致上謝意。老師認真辛勤的工作態度，對學生們來說，就是最好的身教。

同時要感謝在課業上給我協助的同學。宇涵、書瑩，我們大家分工合作，在相互勉勵之下，突破技術上的種種困難，得以完成這個研究。另外，家瑜、柏昌學長以及實驗室的其他同學、學弟學妹，在平時的幫忙，在此一併致謝。

最後，感謝養育我、栽培我的父母親，以及公司內的長官同事，給我的支持與鼓勵，讓我得以完成學業，謝謝。



目錄

摘要	i
Abstract.....	ii
誌謝.....	iii
目錄	iv
表目錄.....	vi
圖目錄.....	vii
一、 緒論.....	1
1.1. 研究背景.....	1
1.2. 研究動機與目標.....	1
1.3. 研究方法與步驟.....	3
1.4. 章節概要.....	4
二、 相關研究	6
2.1. 問題範疇.....	6
2.1.1. 異質性的終端裝置.....	7
2.1.2. 異質性的多媒體內容.....	7
2.1.3. 異質性應用系統中不同使用者的存取管理.....	8
2.2. 單一登入窗口(SSO).....	9
2.2.1. JAVA 語言應用.....	9
2.2.2. LDAP 應用.....	10
2.2.3. 其他方式.....	11
2.2.4. 優缺點比較.....	11
2.3. 異質性多媒體內容早期繫結.....	12
2.3.1. Design Patterns – Adapter.....	12
2.3.2. 整合異質性內容轉譯器及屬性管理.....	12
2.4. 異質性應用系統存取管理.....	13
2.4.1. DRM.....	13
2.4.2. SET.....	15
2.4.3. CVC2.....	16
2.4.4. 3D Secure.....	16
2.4.5. 智慧財產權保護.....	18
三、 系統設計	19
3.1. 系統概述.....	19
3.2. 系統需求分析.....	21
3.3. 系統流程設計.....	22
3.3.1. 客戶端的流程設計.....	23
3.3.2. DRM 伺服器的設計.....	26

3.3.3. Content 伺服器的設計	28
3.4. 系統架構設計.....	29
3.5. MVC 架構設計	32
四、 系統實作	34
4.1. 功能對應實作.....	34
4.2. SSO 實作.....	36
4.3. 早期繫結(early binding)實作.....	38
4.4. 權限控管流程實作.....	40
五、 實作範例	43
5.1. 終端使用者範例.....	43
5.2. 新會員申請範例.....	45
5.3. 交易範例.....	46
六、 結論.....	48
6.1. 總結.....	48
6.2. 未來發展方向.....	49
參考文獻或資料.....	50



表目錄

表格 一 SSO 各家做法比較..... 11



圖目錄

圖 1 異質性系統現況.....	2
圖 2 類別的Adapter Pattern	12
圖 3 異質性的多媒體內容轉換圖.....	13
圖 4 DRM架構圖	14
圖 5 電子錢包流程圖，資料來源[12].....	16
圖 6 VISA 3D Secure流程圖，資料來源VISA[2]	18
圖 7 系統概要圖.....	19
圖 8 系統需求示意圖.....	21
圖 9 系統簡易流程圖.....	23
圖 10 單一登入窗口(Single-Sign-On)流程圖	24
圖 11 SSO的功能模組.....	25
圖 12 DRM流程示意圖.....	26
圖 13 DRM功能模組.....	27
圖 14 系統架構圖	29
圖 15 MVC設計	32
圖 16 應用MVC架構設計.....	33
圖 17 使用者案例圖(Use case).....	34
圖 18 單一登入畫面順序圖	36
圖 19 申請新會員順序圖	37
圖 20 Content Agent物件圖	38
圖 21 Content Agent實作流程圖	39
圖 22 連線存取的順序圖.....	40
圖 23 離線播放的順序圖	41
圖 24 客戶端的流程範例	43
圖 25 單一登入畫面範例	44
圖 26 新會員申請範例	45
圖 27 歡迎畫面.....	46
圖 28 試題畫面.....	47
圖 29 PDA卡片範例.....	47

一、緒論

1.1. 研究背景

科技發展迅速，人們的日常生活已經習慣科技所帶來的便利，甚至於依賴。但是『科技始終來自人性』，這句話說明了所有的系統都是來自於使用者的需求。

在每個公司行號或單位組織內部，往往會因為不同的需求開發許許多多的系統。然而，這麼多的系統也帶來了一些麻煩：使用者要登入不同的系統可能會用到不同的帳號密碼，所以使用者通常會需要過人的記憶力來記得這些帳號密碼甚至於系統的進入方式。

同樣的，針對系統維護人員來說，維護這麼多系統是很累人的。大部分的系統都需要作使用者認證及權限控管，偏偏每個系統都會自行設計一套認證及權限控管的作法。



再加上現今科技發達，人們以不再侷限於利用個人電腦來使用系統。手機和 PDA 都是目前很普遍的行動裝置，人們也因為這些行動裝置的發明更希望系統也能直接在上面使用。如此一來，對系統開發及維護人員都是一大挑戰。如果公司或單位組織合併，系統的整合更是一個大問題。

1.2. 研究動機與目標

現今工業社會，人們的生活都非常緊湊；行動裝置的發明解決了人們想隨時獲得資訊的需求。加上網路四通八達，許多的商業行為也都已普遍在網路上進行。筆記型電腦、手機、Tablet PC 或 PDA 都是目前人們最常使用的行動裝置，這些裝置也都支援上網。因此，針對一個公司行號或單位組織而言，要提供符合這些行動裝置需求的服務，勢必要準備不同的網頁內容。

如圖 1 所示，對一個終端使用者而言，他有可能會用個人電腦、手機或 PDA 來上網登入系統並要求服務。對提供服務的公司或組織而言，他們通常會設計不同的網頁或連結來回應終端使用者的需求。這麼一來，使用者除了需要記住不同的帳號密碼來登入不同的系統外，還要記得不同的行動裝置要用不同的網路連結才行。



圖 1 異質性系統現況

對於系統維護人員而言，同樣的使用者權限控管及認證機制卻要重複實作在不同的系統上；對於網頁內容提供者而言，同樣的內容卻因為要給異質性的行動裝置來使用，必須重複設計編排。這些維護工作不僅耗費人力，對使用者的使用上亦無幫助。

為了提供更好的網路服務並解決這些異質性應用系統整合的問題，在此提出幾個方法：

➤ 單一登入窗口 (SSO, Single-Sign-On)：

不管終端使用者是使用哪一種異質性的行動裝置，他們只要記得一組帳號密碼和一個連結位址即可。系統可以自行偵測使用者的裝置型態並統一作認證及授權的工作；如此一來，終端使用者和系統維護者可達到雙贏的境界。

➤ 網頁內容早期繫結(Content early binding)：

網頁內容的重複產生其實是很浪費人力的一件工作。若能在網頁內容提供者完成設計工作上傳到檔案伺服器的同時，系統可以自動轉換成不同的異質性的內容並利用早期繫結(early binding)的方式，減少動態轉換所造成終端使用者連線時間的浪費。

➤ 數位式權限管理(DRM, Digital Rights Management)：

利用數位式權限管理的方法，可以有效提供網路系統的認證及授權管理。對於網路上的交易、富創意的網頁內容及使用者的權限設計管理等，都可利用這個方法設計來達成，詳細的做法後續章節會詳加描述。

1.3. 研究方法與步驟



為了完成上述的研究目標，我們分成三大部份來進行研究：

■ Single-Sign-On：

1. 分析目前可行的 Single-Sign-On 系統設計。

2. 研究如何自動偵測異質性的行動裝置連結。

● 筆記型電腦：TablePC 也可視為同一類型裝置，所以這一類型的裝置和一般的個人電腦都比較好偵測。只要偵測瀏覽器的資訊即可得知。

● 手機：手機的作業系統有好幾種，瀏覽器的支援也有多有少，實屬較困難一致處理的部分。

● PDA：這分為兩大家族，分別為 Windows CE 和 Palm OS。

3. 商業邏輯分離設計。
4. 實作單一網頁設計並可同時服務多種異質性行動裝置。

■ 網頁內容早期繫結(Content early binding)：

1. 研究如何有效的轉換網頁內容：
 - 如何有效率的轉換單一網頁內容並可提供異質性的行動裝置方便觀看，是這一部分主要的工作。
 - 交通大學軟體工程實驗室有實作出許多的轉換樣板程式，如何應用並有效管理也是一個課題。
2. 實作商業邏輯分離：
 - 提供一個動態商業邏輯設計的介面，並有效的動態配置所有相關的網頁內容。
 - 當商業邏輯改變時，盡量做到有關網頁內容的配置程式不需更動，只需在設定檔或資料庫中更改設定即可。

■ 數位式權限管理(DRM)：

1. 研究何為 DRM 並且研究如何有效管理使用者的權限控管及授權服務品質。
2. 智慧財產權的保障是當今資訊流通的一大課題，在此提供一種簡單卻有效的方法來保護富創意的網頁內容並實作。

1.4. 章節概要

在第一章先簡單介紹本研究的背景及研究動機與目標，並且稍加描述研究的

方法及步驟。

接著在第二章，我們會列出這份研究的範圍並針對之前所描述的三大研究目標來深入研究並加以描述。

在第三章中，我們選擇一個異質性的應用系統來作分析並設計，我們會對其系統架構及主要設計理念詳細描述。

在第四章則會介紹如何實作一個異質性的應用系統，並有效管理網頁內容且減輕系統維護人員的負擔。

第五章，我們針對系統中重要功能加以呈現系統操作流程及功能範例。

第六章為總結，提出此篇論文的結論及未來發展方向。



二、相關研究

本章會先提出本研究的問題範疇並針對問題提出相對應的解法及相關研究；分為四個章節來探討。

2.1. 問題範疇

如同第一章中的圖 1 所示，本研究的問題範圍會分成三大類：

- 異質性的終端裝置：

終端使用者可以利用不同型態的終端裝置來使用相關的應用系統，這些不同型態的終端裝置，我們在此稱之為異質性的終端裝置。2.1.1 會加以詳細描述問題內容。

- 異質性的多媒體內容：

各個應用系統中為了服務終端使用者的需求，會針對不同的終端裝置來準備不同型態呈現的多媒體內容。2.1.2 會描述在此條件下系統會有哪些維護的問題並提出方法。

- 異質性應用系統中不同使用者的存取管理：

在這麼多異質性的應用系統中，使用者的權限管理及智慧財產權的保護等問題都會在 2.1.3 中探討。

2.1.1. 異質性的終端裝置

目前人們已經可以使用許多不同的終端裝置來存取應用系統，包含桌上型電腦、筆記型電腦、PDA、手機及 Tablet PC 都已經可以支援許多目前常用的系統標準。不管是一般的應用程式或連結上網，這些裝置大多都可執行。

但是對於終端使用者而言，應用系統的連結位置可能因為不同的終端裝置而有所不同；這麼一來，當使用者用不同型態的終端裝置來存取同一類型的應用程式時，使用者必須面臨要自行更換不同連結位置。甚至於還要分別輸入不同組的使用者帳號密碼資訊才可以開始使用系統。

對於系統維護者而言，則是不同的系統都要實作一些相同的功能，例如：系統認證及授權等，這對整個系統管理及維護成本控制上都是一大問題。如何才能幫助使用者及系統管理者呢？



本研究建議採用單一登入窗口 (Single-Sign-On) 的做法來統一解決以上面對異質性的終端裝置所提出的需求。如此一來，我們不僅可以提供使用者一個簡單明確的系統進入窗口，並且減低系統維護成本。這可說是一舉數得，詳細研究會在 2.2 來描述。

2.1.2. 異質性的多媒體內容

為了達到異質性的終端裝置的使用需求，系統業者或管理者常要準備異質性的多媒體內容來供使用者讀取。這些準備工作可是累壞了多媒體內容的設計者或提供者；有些做法是同一種腳本設計成多種版本來讓手機或其他不同的行動裝置存取；有些則是撰寫轉譯器的程式來達成目標。

交通大學軟體工程實驗室提出許多種的轉譯器來協助多媒體內容的服務多向化。在本研究中，我們參考[23]中所提出的互動式多媒體樣板在異質性手持行動裝置上的轉譯器的做法並提出 Adapter 整合的方式來作一更有效的管理。

除了整合轉譯器來簡化多媒體內容提供者的工作流程及減輕負擔外，我們還提出動態設定多媒體內容的商業邏輯屬性，並且利用早期繫結(early binding)的做法來改善使用者使用上的效率問題。這種做法也降低了程式的耦合度並減輕系統管理者及多媒體內容開發者的負擔。2.3 會說明其原理及架構。

2.1.3. 異質性應用系統中不同使用者的存取管理

當終端使用者進入相關的應用系統來使用時，系統的基本認證和使用者的授權處理是一定要做到的。尤其現在先進國家都倡導智慧財產權的重要，針對目前電子資訊的交流情形普遍且網路發達的大環境，如何有效率的管理及存取控管並且提供合理的智慧財產權的保障，是各個公司行號或單位組織眼前的重要課題。

目前當紅的 DRM(Digital Rights Management) 架構提出了非常完整的模組設計理念；包括權限的控管、資料的授權使用、商務交易或交易紀錄追蹤等。其中因為網路上的電子交易廣為大眾使用，各個金融機構也都提出了許多線上付款的保護機制，例如：早期的電子錢包(SET)、Master 國際組織推行的 CVC2 值驗證和 VISA 國際組織最近強力推薦的 3D Secure 線上驗證。這些都會在稍後 2.4 章節來加以探討。

除了交易及權限控管處理外，使用者再使用多媒體內容時，如何有效的控管存取情況並適當保護智慧財產權也會在 2.4 來提出。

2.2. 單一登入窗口 (SSO)

企業或單位組織內部應用系統因應使用者的需求一值增加，彼此相互不相容的狀況或浪費人力的重複開發情形，使得 SSO(Single-Sign-On)成為重要的課題。各大企業或相關的應用伺服器提供者都各自提出許多相關整合控管的方法，其核心技術大約分為幾種：利用 JAVA 提供的解法組合、輕量級名錄存取協定 (LDAP) 或應用伺服器本身提供的認證管理甚至自行開發相關模組。

2.2.1. JAVA 語言應用

Java 語言有一個最大的好處就是可以完全跨平台執行。不管是個人電腦或手機、PDA 等行動裝置，大都已經支援 Java 執行環境。所以只要是跨平台或整合系統的相關開發，企業或應用程式設計服務者都會優先採取 Java 的解法。雖然它仍有執行效率的問題，但目前硬體設施以無須擔心。

在[5]書中，我們可以看到 JAVA 提供了許多的安全元件，其中跟 SSO 相關的設計首推 JAAS(Java Authentication and Authorization Service) 架構。JAAS 架構可用來提供執行期鑑別與授權資源，並且可支援動態定義權限內容的方式。

JAAS 鑑別是透過以主體為基礎的存取控制與嵌入式鑑別的完成。JAAS 亦可以與 Java Servlets、Java Server Pages (JSP) 及 Enterprise Java Bean (EJB) 連結並提供可信賴且安全維護用戶端身分的功能。

關於 JAAS 的詳細描述，除了上述的書中介紹外，還可以參考 Sun 網站提供的規格說明[11]，其中有一段書名擷取如下：

*The JAAS infrastructure can be divided into two main components: an **authentication** component and an **authorization** component. The JAAS*

authentication component provides the ability to reliably and securely determine who is currently executing Java code, regardless of whether the code is running as an application, an applet, a bean, or a servlet. The JAAS authorization component supplements the existing Java 2 security framework by providing the means to restrict the executing Java code from performing sensitive tasks, depending on its codesource (as is done in Java 2) and depending on who was authenticated.

2.2.2.LDAP 應用

何謂 LDAP？即為輕量級名錄存取協定(Lightweight Directory Access Protocol)。在[24]書附錄 D 中列有所有跟 LDAP 相關的 RFC 文件，有興趣研究的可以到[25]來下載。另外我們也可以參考[14]這本書，書中有很詳細的說明並一步步教導如何實作相關的 LDAP 服務，在此針對 LDAP 服務的定義簡單整理如下：

It is an information model that describes what you can put in the directory or a naming model that describes how you arrange and refer to the directory data, or a functional model that describes what you can do with directory data, or a security model that describes how directory data can be protected from unauthorized access.

簡單說來，LDAP 就是將所要提供的服務內容設計成名錄的結構，利用 Profile 的設定來達到動態管理的需求。項目 (Entry) 是構成 LDAP 名錄的基本單元；可以將項目想像為名錄資訊樹 (Directory Information Tree，簡稱 DIT) 的內部或外部節點。一筆項目所包含的資訊與物件類別 (ObjectClass) 的實例 (Instance) 有關，這些物件類別會具有一些必要的或選用的屬性 (attribute)。要做好一個 LDAP 的服務，其 LDAP 綱要 (Schema) 的設計是很重要的。

2.2.3. 其他方式

除了上述的兩大類方法外，另外還有硬體設施所提供的認證及授權設定來達到 SSO 的機制。這種做法通常會根據搭配的伺服器所安裝的作業系統或分散式系統中的應用伺服器廠商所提供的模組而有些許功能上的差異。

最後還有一種彈性最佳，最能根據企業量身訂做的方式，就是自行開發該企業所需的登入模組並根據其商業邏輯設定認證及授權的相關檔案來達成客製化的需求。

哪一種方式比較好呢？基本上見仁見智，除了看系統整體規劃，還要看維護成本及系統範疇而定。下一章節會稍加比較，比較結果存屬個人研究感想，至於要使用哪一種方式，或綜合搭配，全都是系統規劃的差異。



2.2.4. 優缺點比較

表 1 SSO 各家做法比較

	JAVA 語言應用	LDAP 應用	伺服器管理	自行開發
系統規劃彈性	佳	佳	稍差	最佳
適用系統範疇	大小皆可	適用大	適用小	大小皆可
維護難易	中	難	易	中

2.3. 異質性多媒體內容早期繫結

2.3.1. Design Patterns – Adapter

在[1]和[8]中都有針對 Adapter 這個 pattern 的做法簡單描述，我們可以看圖 2 來簡單解釋 Adapter 的做法。Adapter 有兩種實作方式：一種是類別的 Adapter(繼承式)；另一種則是物件個體的 Adapter(委讓式)。在此我們用的是第一種繼承式的 Adapter 做法。

如下圖，終端使用者不需知道有 Adaptee 的存在，純粹只知道包裝後的類別並達成所需轉換的結果。

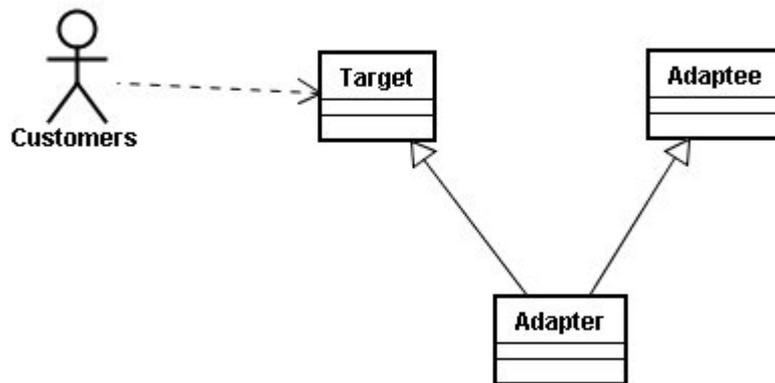


圖 2 類別的 Adapter Pattern

2.3.2. 整合異質性內容轉譯器及屬性管理

利用 Adapter 的做法將現成所有異質性的多媒體轉譯器[23]包裝起來，並增加屬性管理機制來預先準備好在 Content 伺服器上供終端使用者使用。如圖 3，該轉換模組(To Destination Process)會先轉換成每一種對應的多媒體內容，並依照系統設定檔的目錄指示安置在 Content 伺服器上。接著會根據系統設定檔中

每一個所屬目錄的權限設定來重新分配，並產生相對應的連結位置存放回資料庫中。如此一來，系統的商業邏輯只要設定在系統設定檔中即可，當商業邏輯更動時，相關模組無須更改。

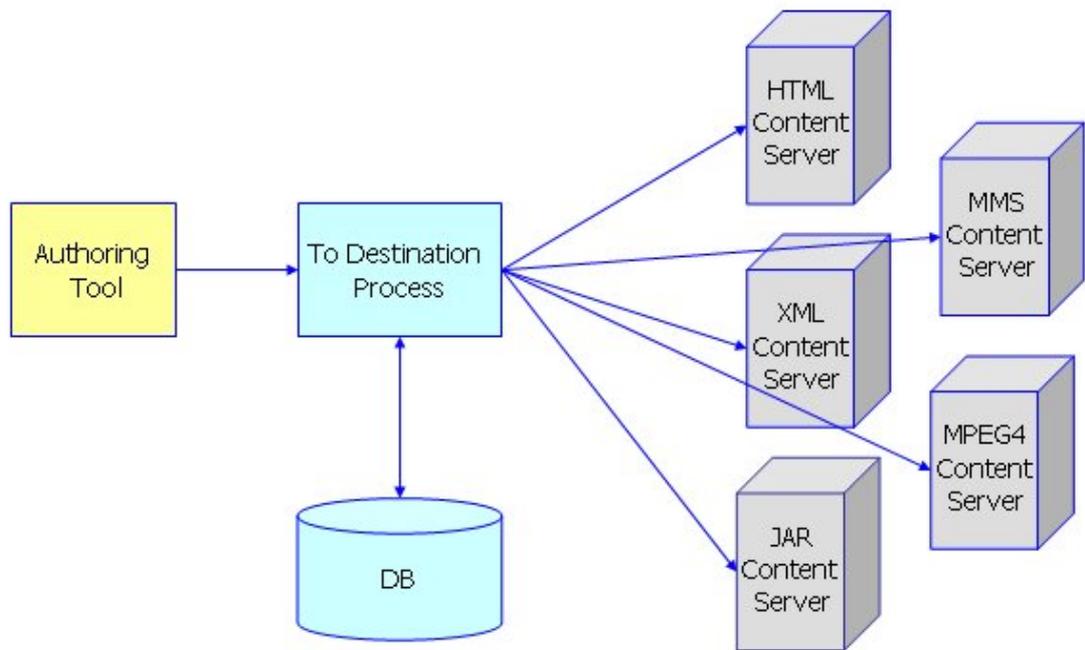


圖 3 異質性的多媒體內容轉換圖

2.4. 異質性應用系統存取管理

2.4.1. DRM

何謂 DRM？DRM 是 Digital Rights Management 的縮寫。對於它的定義，[3]文中有一小段說明很貼切，擷取如下：

Additionally, it is important to note that DRM is the “digital management of rights” and not the “management of digital rights”. That is, DRM manages all rights, not only the rights applicable to permissions over digital content.

在本研究中，我們利用 DRM 的機制來解決整個系統中的權限管理及內容保護。它不僅包括使用者權限控管設定、資源的授權管理、交易的安全及掌控，甚至也包括事後的使用紀錄追蹤及分析。

在圖 4 中，我們可以看到 DRM 整個簡單的架構。這些分類也是參考[3]得來的，其中有一段文字對於設計及實作 DRM 架構所要注意的事項，節錄於下：

In designing and implementing DRM systems, there are two critical architectures to consider. The first is the Functional Architecture, which covers the high-level modules or components of the DRM system that together provide an end-to-end management of rights. The second critical architecture is the Information Architecture, which covers the modeling of the entries within a DRM system as well as their relationships.

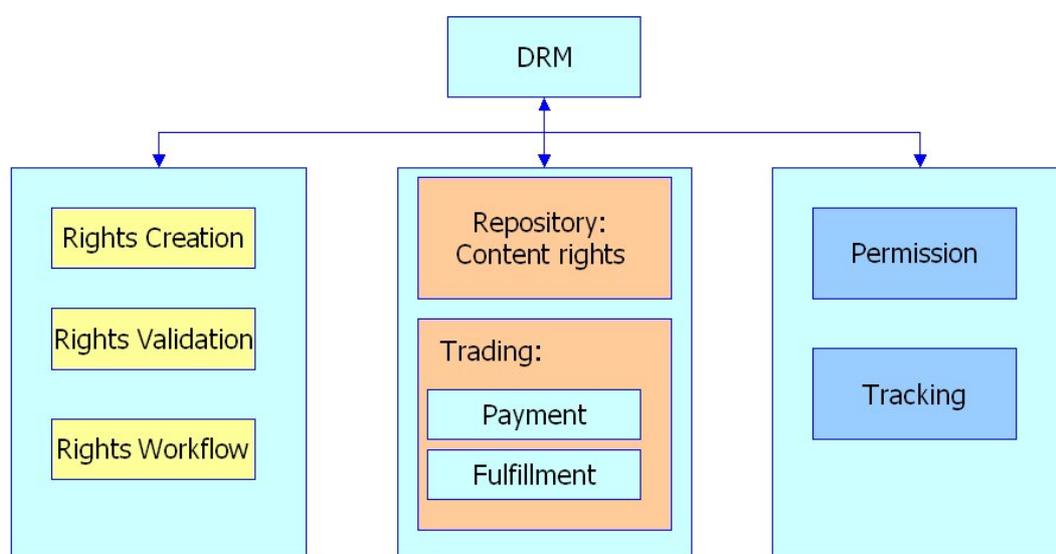


圖 4 DRM 架構圖

整個架構圖可以區分為三大區塊來看：分別是權限控管 (Rights Control)，資源控管 (Contents Control) 及 授權控管 (Permission Control)。簡單說明整個架構圖如下：

- 權限控管 (Rights Control)：

這部分包含 Rights Creation，Rights Validation，和 Rights

Workflow。

- 資源控管(Contents Control)：

這部分則是含括兩大模組：Repository 和 Trading。其中，Repository 指的是資源的權限分類工作及授權類別定義；Trading 則是只所有的交易流程，包括 Payment 和 Fulfillment。

- 認證控管(Permission Control)：

這個模組則是負責掌控實際授權的控制流程及 Tracking。Tracking 可用來追蹤使用者使用的狀況並且可以日後加以分析出使用習慣進而提供更豐富的服務內容。

2.4.2.SET



SET 是 Secure Electronic Transaction 的縮寫。這通常稱之為電子錢包的交易，圖 5 取自[12]。這個機制照理說應該是安全性相當高的線上付款機制，可是因為持卡人和商店皆要安裝 CCA 和 MCA 的認證裝置，推廣不易。現今已鮮少使用，目前 VISA 極力推廣新的 3D Secure 機制。

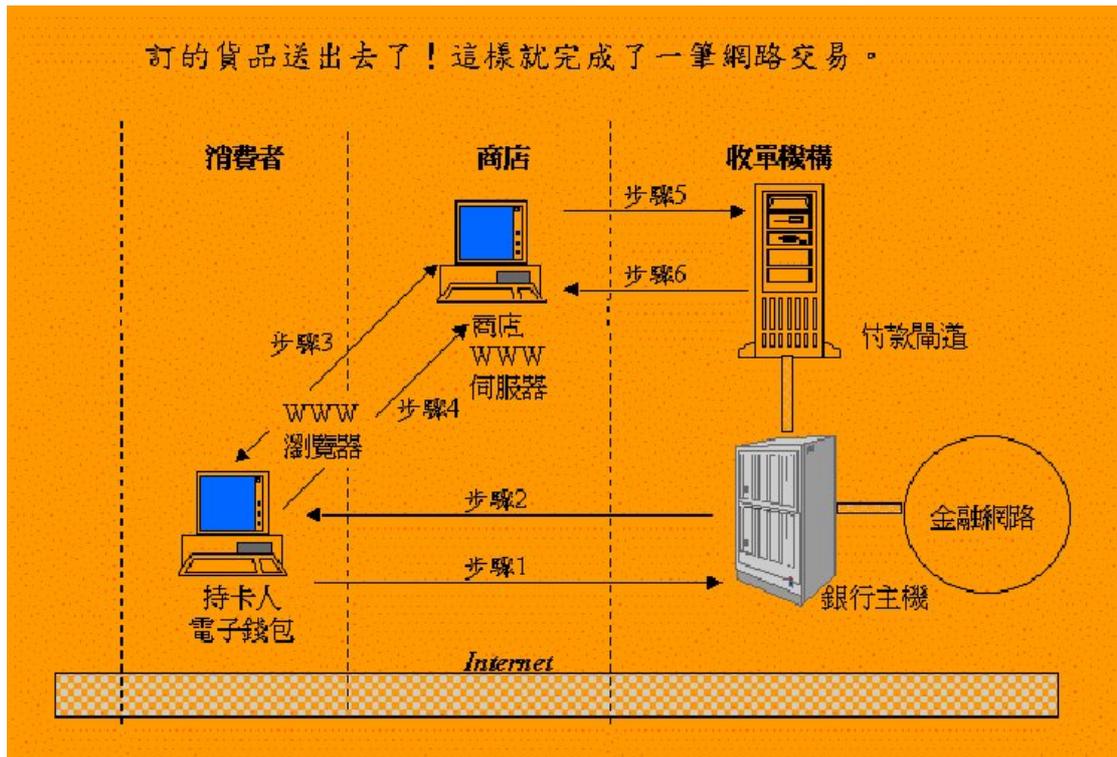


圖 5 電子錢包流程圖，資料來源[12]

2.4.3.CVC2



這是 Master 國際組織推出的安全機制，所謂的 CVC2 是指信用卡背面後三碼。這項安全認證需要搭配 SSL(Secure Socket Layer) 線上授權的做法，但是因為無法保證絕對安全，有可能偽卡也是可以取得交易授權碼，目前 Master 也積極的推動 VISA 所提出的 3D Secure 的機制。

2.4.4.3D Secure

3D Secure 機制是由 VISA 國際組織所提出的最新做法。在[2]的 3D Secure Introduction 文件中有提出解釋如下：

The 3-D secure protocol underlies a new Visa payment service designed to enhance and validate payments made through the Internet. 3-D secure is an authentication technology that uses Secure Socket Layer (SSL)

encryption and a Merchant server plug-in to:

- *Pass information and query participants to authenticate the cardholder during an on-line purchase.*
- *Protect payment card information as it is transmitted via Internet.*

所謂的 3D 指的是 3 Domains：Issuer Domain，Interoperability Domain，和 Acquirer Domain，整個運作過程及關係可從圖 6 看的出來。至於詳細的介紹，我們引用片段文章如下：

- ***Issuer Domain:** The issuer is responsible for managing the enrollment of their cardholders in the services (including verifying of each cardholder who enrolls) and authenticating cardholders during online purchase.*
- ***Acquirer Domain:** The acquirer is responsible for defining the procedures to ensure that merchants participating in the Internet transactions are operating under a merchant agreement with Acquirer and providing the transaction processing for authenticated transactions.*
- ***Interoperability Domain:** This domain facilitates the transaction exchange between the other two domains with a common protocol and shared service.*

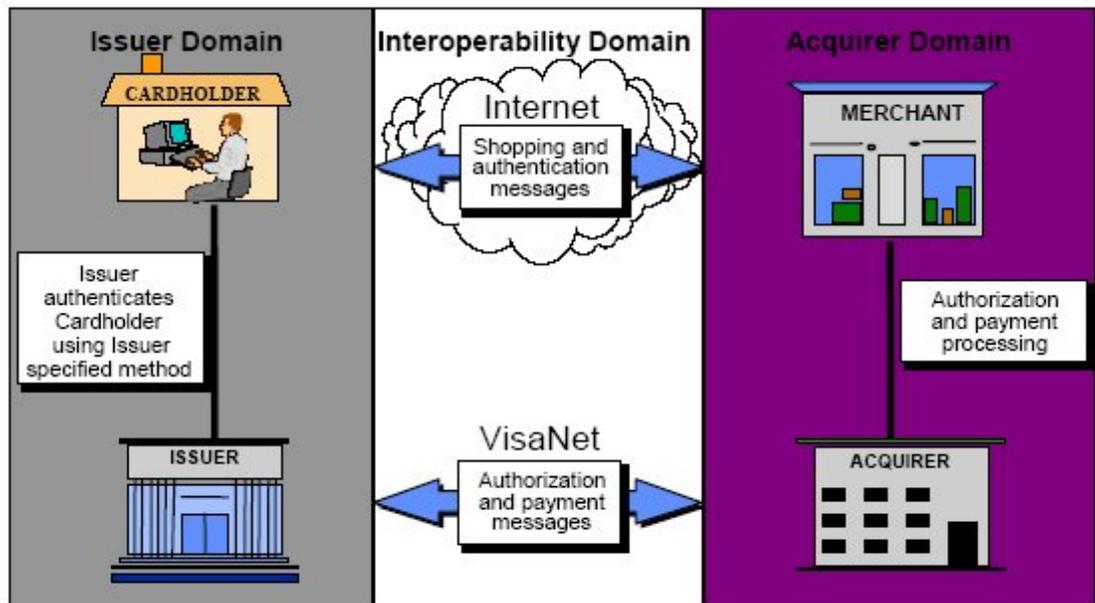


圖 6 VISA 3D Secure 流程圖，資料來源 VISA[2]

2.4.5. 智慧財產權保護

如何達到智慧財產權的保護機制？我們在此提出針對異質性的應用系統中的多媒體資源的保護措施如下：

- 自動偵測終端裝置：

系統自動偵測不同的終端裝置，並且限制終端裝置不可存取不同於該裝置格式的多媒體內容。如此可以避免所有異質性的多媒體內容被胡亂下載變賣。

- 提供自我毀滅精靈伴隨資源下載：

所謂的自我毀滅精靈，其實就是指一種代理程式(Agent)。設計一種代理程式會伴隨多媒體內容下載到任何一種終端裝置，一旦 player 播放完畢，立即銷毀所下載播放的多媒體主檔。

三、系統設計

在第二章，我們已經簡單的說明了相關研究的問題範圍及解法。這一章則挑選一種 Web-based 異質性的應用系統來作更深入的探討。

3.1. 系統概述

一個 Web-based 異質性的應用系統示意圖如下圖。

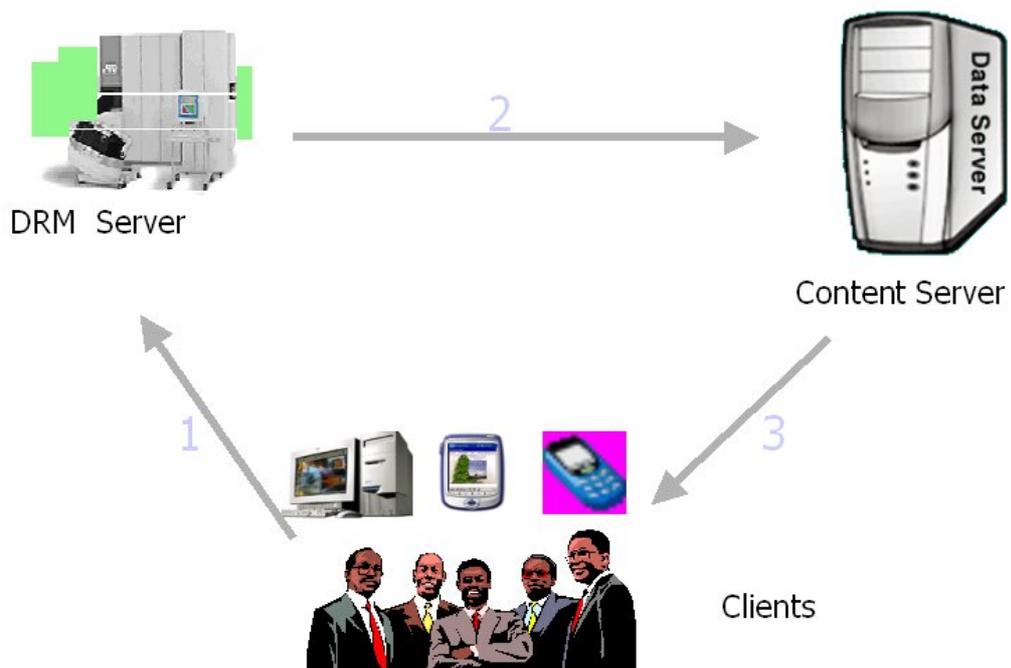


圖 7 系統概要圖

如圖所示，整個系統大致上分為三大區塊：客戶端，DRM 伺服器，和 Content 伺服器。

- 客戶端：

系統要能提供終端使用者利用個人電腦、手提電腦、手機或 PDA 來連線進入系統。對於使用者而言，不管他們用哪一種終端裝置，只要使用同一個連結網址即可登入系統取得服務。

- DRM 伺服器：

DRM 伺服器要能提供 2.4.1 所提到的數位式權限控管的架構及模組功能。也就是說，系統要能提供統一認證及授權功能，並且要能做到智慧財產權的保障。當然，所有的交易及存取紀錄都要被保留下來。

- Content 伺服器：

所有的多媒體資源內容都透過異質性內容轉換模組並依照系統設定檔中所規劃的目錄存放在此。所有的連結網頁內容也都已事先依照商業邏輯展開，只要使用者通過系統認證並授权使用，伺服器立刻依照使用者的需求提供服務。



整個流程可簡化為三步驟：

1. 客戶端透過單一連結網頁向 DRM 伺服器提出認證需求。此時，DRM 伺服器會啟動相關的認證機制來進行使用者身分認證及資源授權。
2. 當 DRM 完成認證授權工作時，伺服器會將通過認證以及獲得資源授權的使用者的需求轉到 Content 伺服器。並會指示 Content 伺服器應該配置哪一種終端裝置的內容服務。
3. 當 Content 伺服器收到 DRM 伺服器的訊息之後，會依照該終端使用者所被授權存取的內容網頁直接呈現在終端使用裝置上。

3.2. 系統需求分析

對於一個提供異質性多媒體網站系統而言，其需求大致上會類似圖 8 所顯示。主要使用者會分成三類：客戶、多媒體內容提供者和系統管理者。

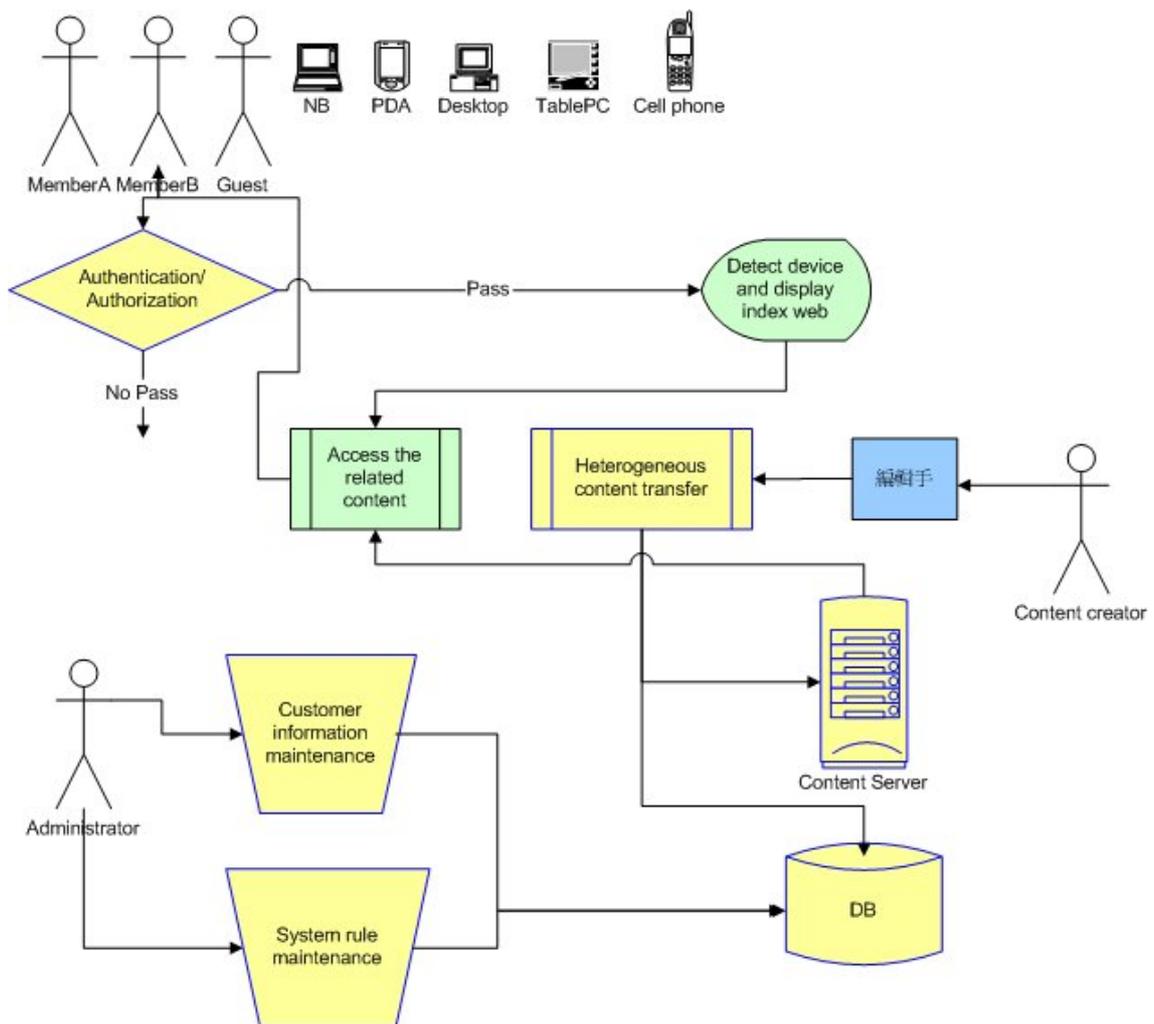


圖 8 系統需求示意圖

➤ 客戶：

- 系統提供一個統一的連結位址給客戶，即使客戶使用不同的連線裝置。
- 系統提供畫面並可接受新會員申請。
- 系統會自動偵測終端連線裝置並協助轉址到對應的目的網址。

- 系統會提供統一身分認證及資源授權的功能並紀錄所有交易或存取資訊。
- 客戶可以選擇下載到客戶端播放多媒體的內容，系統要能保障智慧財產權及會員權益。

➤ 多媒體內容提供者：

- 多媒體內容提供者採用智勝國際[26]與本校軟體工程實驗室所開發的編輯手軟體來編輯互動式多媒體內容。
- 當多媒體內容提供者完成編輯手編輯並決定要上傳至 Content 伺服器時，系統會自動轉譯成目前系統支援的異質性的終端裝置檔案格式，多媒體內容提供者不再需要多費心思來做轉換的工作。
- 系統會根據系統設定檔中所描述的目錄位置來存放已轉譯好的多媒體內容。



➤ 系統管理者：

- 系統提供一使用者維護介面給管理者維護商業邏輯和會員權限控管。
- 系統提供簡單維護介面給系統管理者來更改異質性多媒體內容的相關權限或目錄設定。

3.3. 系統流程設計

接下來這個章節，我們會設計整個系統的流程。首先，我們先假設客戶只會使用個人電腦、手機、PDA 或筆記型電腦來使用系統。整個系統只提供一個簡單的對外連結位址，所有的使用者不管是舊成員或新加入的會員甚至參觀者皆經過

同一個系統進入點。整個系統的流程圖如圖 9 所示。

系統會自動偵測連線的終端裝置並進行網頁畫面的圖素(pixel)計算，讓所有的需求相同的使用者儘管來自不同的終端裝置都看到同樣的畫面。如此會降低系統維護的人力成本。

經過系統的驗證流程，最後系統會呈現相對應的結果於終端使用者所使用的連線裝置上。

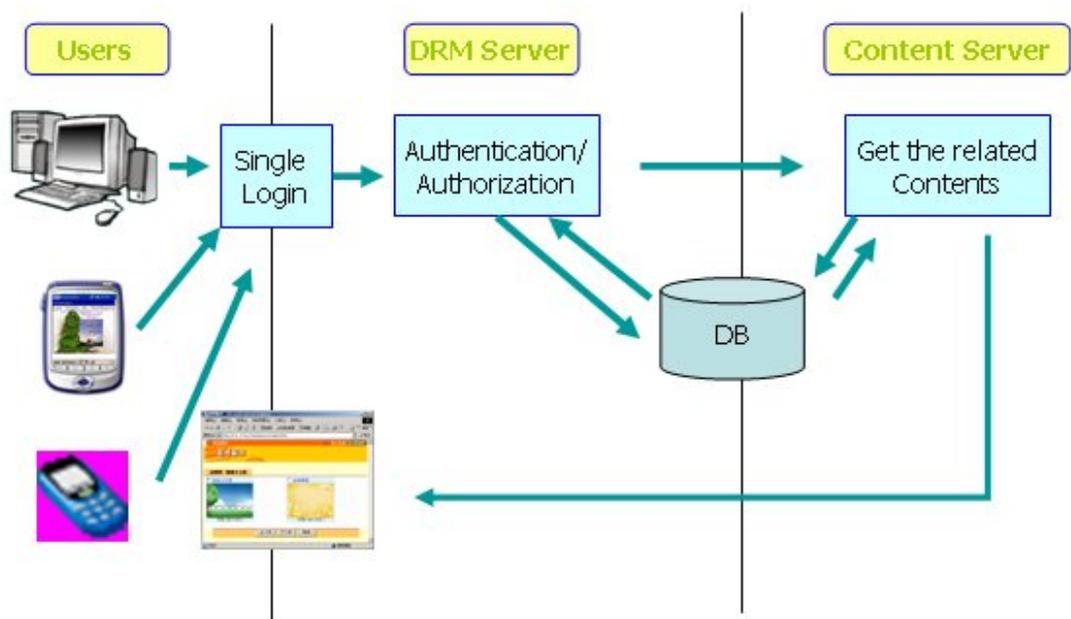


圖 9 系統簡易流程圖

3.3.1. 客戶端的流程設計

為了避免系統維護人員重複撰寫相同功能的程式設計，以及提供終端使用者一個簡單的系統連結方式，我們設計以單一登入窗口(Single-Sign-On)方式來解決上述的問題。

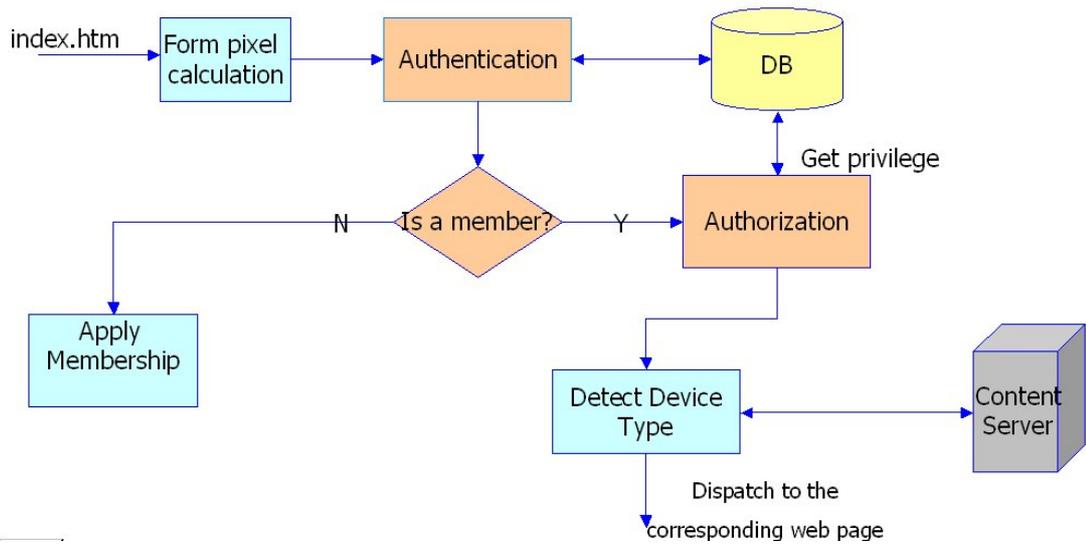


圖 10 單一登入窗口(Single-Sign-On)流程圖

如圖 10 所示，針對客戶端而言，系統流程說明如下：

- 單一連結網址(index.htm)：

系統提供單一連結網址，所有的使用者都可利用該連結進入系統並要求服務。



- 圖素運算(Form pixel calculation module)：

當用者利用異質性的終端裝置連線進入系統網頁時，系統會自動根據畫面上的圖素來自動配置網頁內容物件大小。以使用者方便查看的需求來看，最好畫面不需使用太多的捲軸。

- 身分認證(Authentication)：

系統提供統一的身分認證機制，該模組設計必須和商業邏輯分離。如此可以減輕系統維護人員的負擔並可提供重複使用的優點。除了客戶需要認證外，系統管理員以及多媒體內容提供者要上傳時，也都要透過這個模組來做身分認證。

- 資源授權(Authorization)：

當使用者通過系統認證時，系統會呼叫資源授權模組確認權限。此時系統會根據系統設定檔中不同的商業邏輯來作授權確認。這個功能模組一樣也是要設計獨立的模組並且重複使用。

- 申請會員(Apply membership)：

若是身分認證失敗，系統會自動帶出新會員申請的畫面。對於一個 Web-based 的應用系統而言，這個設計是必須的。

- 自動偵測連線裝置(Detect device type)：

當使用者通過認證並獲得系統授權時，系統會自動偵測終端連線裝置並自動分發到相關的網頁內容。

整體來看，Single-Sign-On 可用 Java 來作開發，利用 JAAS 的架構並整合上述流程，我們可以整理成一個簡單功能模組如圖 11。



圖 11 SSO 的功能模組

- Pixel calculation module：

對應流程圖中的圖素運算，負責計算終端裝置的視窗圖素並重新配置網頁內容。如此設計可以減少系統開發人員因為要開發因應異質性的終端裝置而必須重複配置網頁的工作。

➤ Login Bean :

這個物件模組基本上含括身分認證及資源授權，算是 DRM 架構中不可或缺的重要元素。

➤ Process Customer info Bean :

這一個物件模組掌管所有客戶基本資料的新增、刪除和修改的動作。當系統管理者要維護相關的基本資料或客戶申請新會員時，都會呼叫該物件模組。

3.3.2.DRM 伺服器的設計

DRM 伺服器是整個系統的權限控管重點，不僅提供身分的認證和資源的授權處理，更是含括整個相關系統資源的保護。整個運作流程，如圖 12 所示，包含當客戶端下載多媒體播放內容到用戶端時，其播放內容自我保護的機制都可看的一清二楚。

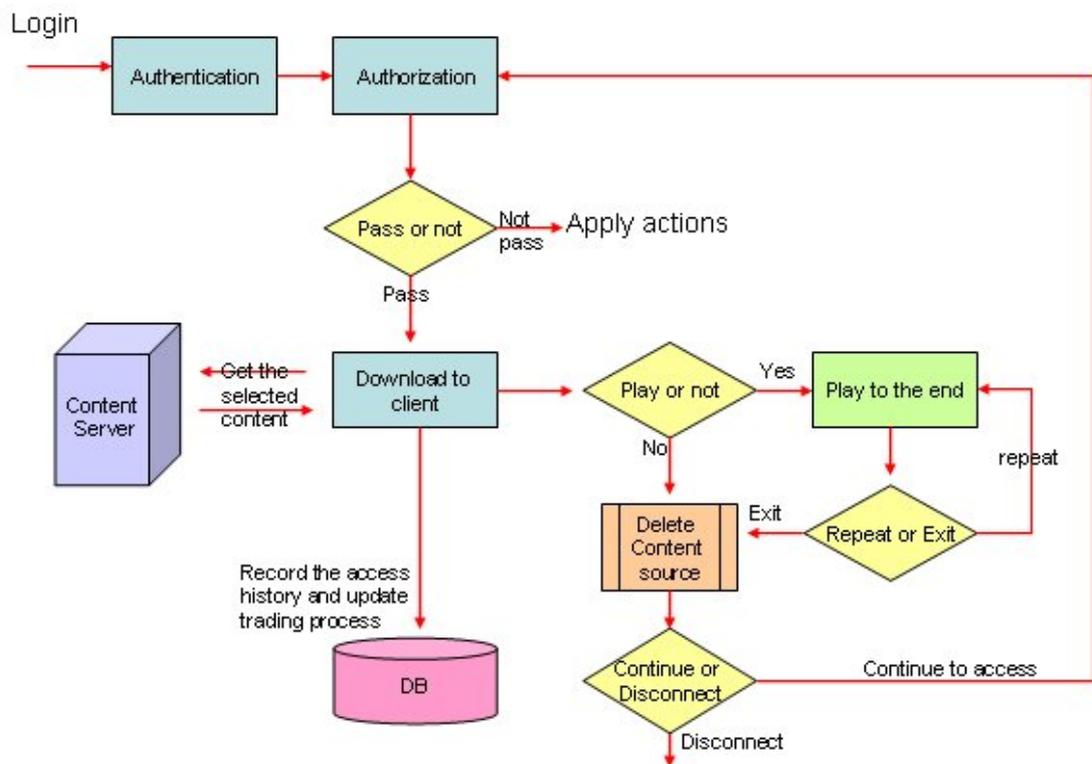


圖 12 DRM 流程示意圖

整個流程圖說明如下：

- 當使用者通過 DRM 伺服器的認證及授權之後，系統會同意終端裝置開始存取 Content 伺服器中所對應的多媒體內容。此時，系統會啟動一個監測程式，依照系統設定檔中所設定的商業邏輯以及權限群組資源分配設定，定期偵測用戶端的使用情形。
- 若用戶端要求下載到終端裝置上使用或撥放，此時系統會先判斷是否足夠授權條件可供下載。當下載成功時，隨著多媒體內容及播放器下載到用戶端的 Agent 會詢問用戶是否直接播放。
- 當播放完畢，用戶可以決定是否重播或離開播放器；一旦決定離開播放器之後，Agent 會啟動自我毀滅的功能，將下載至用戶端的多媒體播放主檔刪除。
- 假設用戶端不正常跳離播放器，或試圖中斷 Agent 的自我毀滅功能，Agent 皆會在下一次用戶端開啟播放器的第一時間完成多媒體播放主檔刪除動作。



整體來看，整個 DRM 伺服器的物件功能模組可以用圖 13 來表示。

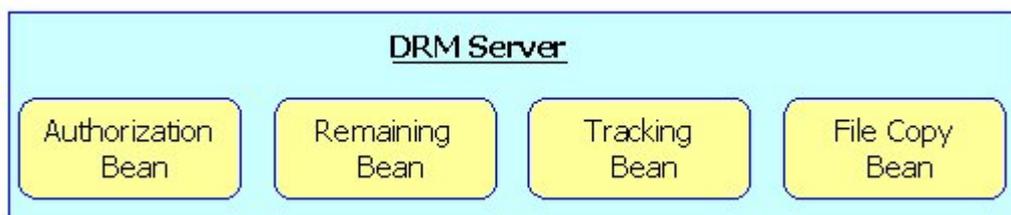


圖 13 DRM 功能模組

- Authorization Bean :

這個授權模組除了一開始的資源授權外，也會依照商業邏輯設定檔或權限設定檔的設定定期監控用戶端的變化。這一個模組會在用戶端通過身分認證之後被呼叫執行。

- Remaining Bean :

當有需要計算該用戶端商業邏輯設定中的任何運算動作，都會呼叫這一個物件模組來完成並回傳結果。這一個物件也可以算是交易過程中不可或缺的重要元素。當使用者存取任一網頁上的動作都會先作授權確認再根據設定檔決定是否呼叫 Remaining Bean 來運作。

- Tracking Bean :

任何一個有交易動作的應用系統，最好都要紀錄所有重要的交易紀錄。除了保障終端使用者的權益更可以作為日後查證依據。最近幾年強調服務導向(Service oriented)，這些交易紀錄作一有效的資料分析還可擷取出有用的資訊來提高服務品質，創造另一商業契機。



- File Copy Bean :

這一個物件則是隸屬於 Content 轉譯時，需要依照權限分組設定檔來區別所有要上傳到 Content 伺服器的多媒體資源。

3.3.3.Content 伺服器的設計

依照之前 2.3.2 章節的設計，當多媒體內容提供者上傳到 Content 伺服器時，會驅動異質性內容轉譯模組自動依照目前系統設定檔中需要作轉譯的型態一一轉換，並依照指定目錄存放。

接著，系統會依照商業邏輯的設定來作相關連結網頁的設定並轉換成 jsp 或 htm 檔案格式存放。當 Content 伺服器收到通過授權的使用者需求時，伺服器會依照需求一一回應。

3.4. 系統架構設計

整個系統架構設計，如圖 14 所示：在此我們可將異質性的終端裝置視為是用戶端，其包含了個人電腦、PDA、手機、筆記型電腦和 Table PC。所有的用戶裝置都採用 HTTP 的通信規則來存取系統服務。我們在此採用 Tomcat 作為 Web 伺服器，提供 web-based 異質性的應用系統服務。

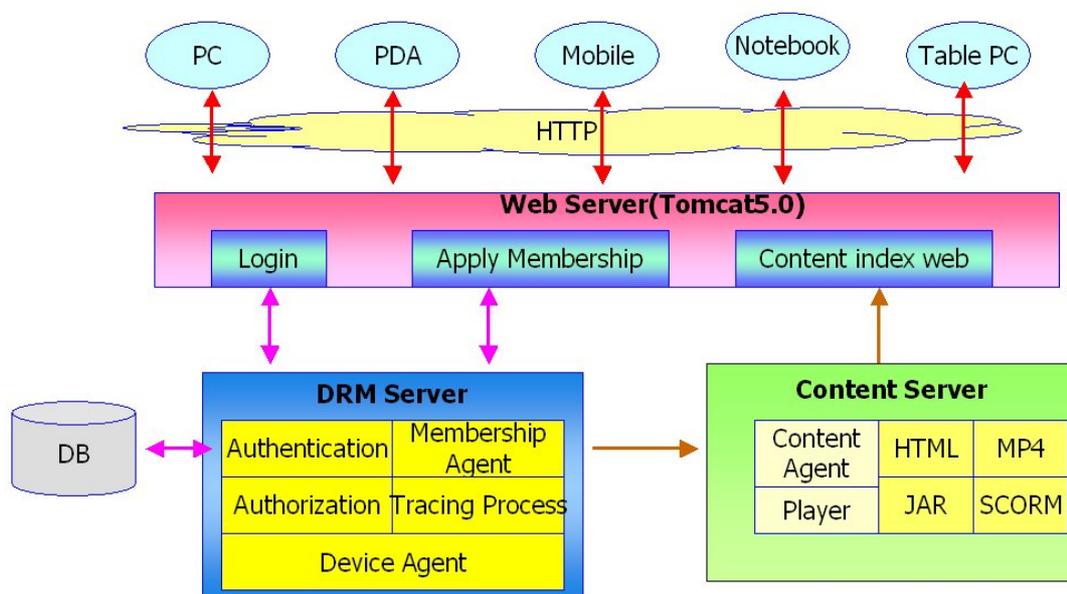


圖 14 系統架構圖

➤ Login :

提供單一系統對外登入窗口的網頁連結，也是系統的第一關卡。

➤ Apply Membership :

當有新會員邀申請入會時，系統會轉址到這個網頁並讓使用者依照步驟一步步完成會員的申請。

➤ Content index web :

這些事先根據早期繫結(early binding)機制伴隨異質性的多媒體內容轉譯而產生的網頁，Content 伺服器會依照 DRM 伺服器所傳來的訊息傳送不同的連結網址。

關於 DRM 伺服器這一部分，則是囊括 Authentication、Authorization、Membership Agent、Trading process 和 Device Agent。如同之前 2.4.1 的 DRM 架構設計，整個架構運作和資料庫是密不可分。分別解釋如下：

➤ Authentication :

這個模組負責提供身分認證並負責回傳給 Web 伺服器結論。若使用者沒有通過基本身分認證，它會告知 Web 伺服器要轉址到新會員的申請頁面；若通過認證，則會驅動 Authorization 模組來做下一步資源授權的工作。



➤ Authorization :

經過 Authentication 模組成功認證之後，接下來的一連串授權工作則會交由 Authorization 模組來運作。它會根據系統設定檔中的權限設定以及商業邏輯設定判斷授權與否。

➤ Membership Agent :

這一個模組則是在新會員申請或系統管理者維護會員相關資訊時會被呼叫。當有交易行為時，則會呼叫 Remaining Bean 來處理交易。

➤ Trading Process :

這部分則包含了之前章節所介紹的 Remaining Bean 及 Tracking module。整個運作方式其實就是進行交易時，除了依照商業邏輯的設定運算並更新使用者的相關交易資訊外，並會紀錄所有的交易動作並存放

在資料庫中。

➤ Device Agent :

當用戶端已經通過授權之後，接下來的工作就會交由 Device Agent 來處理。它會自動偵測終端連線裝置的種類並將傳送訊息給 Content 伺服器來作回應。

針對 Content 伺服器這一部分的架構，則是著重於異質性的多媒體內容的倉儲管理。為了達到智慧財產權的保護，我們也將 Content Agent 及播放器放在這裡。

➤ Content container :

這個部分就是所謂的倉儲管理。系統管理者可以依照需求並設定於系統設定檔中，包含目錄的設計或是權限的分類等皆可。至於種類，目前可轉譯的有 HTML、MPEG4、JAR、C++ 和 SCORM 的格式。要如何整合管理則依照商業邏輯來作管理。

➤ Content Agent :

這一個代理 Agent，主要的工作內容就是保護這些富有創意的異質性多媒體內容。當有下載需求時，它會隨著 Player 和多媒體播放內容檔案一起下載到用戶端，並在結束播放時執行刪除播放主檔的工作。

➤ Player :

目前多媒體播放器則仍需分為三類設計，分別是：鑲嵌在瀏覽器上的以及針對 PDA、手機設計的播放器。

3.5.MVC 架構設計

這個系統採用 Module-View-Control 架構來設計，如下圖所示：用戶端利用瀏覽器傳送給 Controller，此時 Controller 會接手處理並將要求 Module 協助做邏輯的運算，再將控制權轉交給 Viewer 來呈現。本研究採用 JSP 來開發 Viewer，用 Java Bean 來作 Module 的工作以及 Servlets 來收手 Controller 的運作。

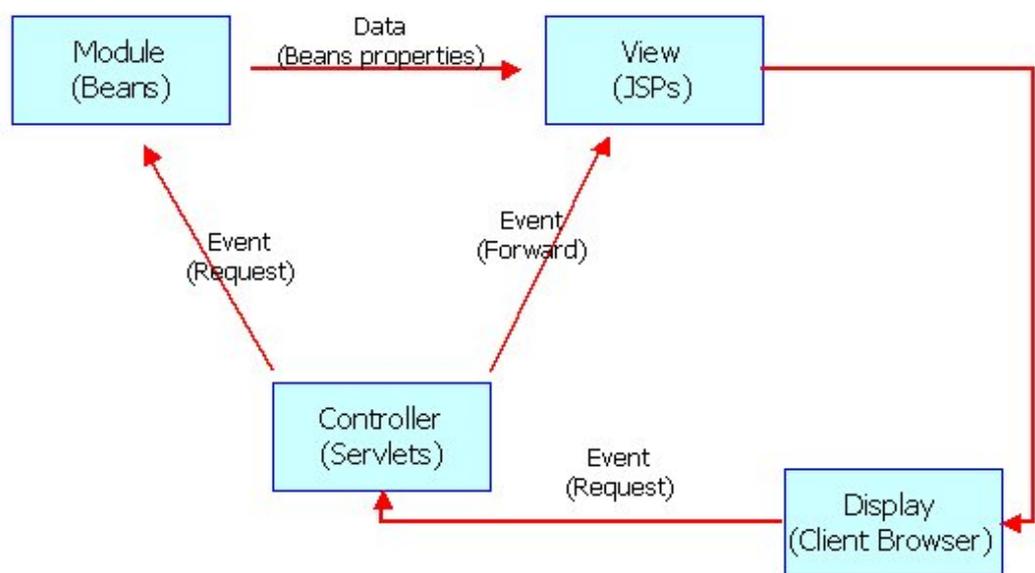


圖 15 MVC 設計

MVC 架構是一個簡單且容易實作的商業邏輯和系統邏輯切離的設計。這種獨立的设计架構可以避免當商業邏輯改變時，企業或單位組織要耗費人力多就系統作修改。有時甚至因為系統彈性的設計不佳，導致於要重新設計整個系統也是常有的事。

根據之前章節所提到的流程及功能模組，我們可將程式及套用 MVC 架構為圖 16 所示。

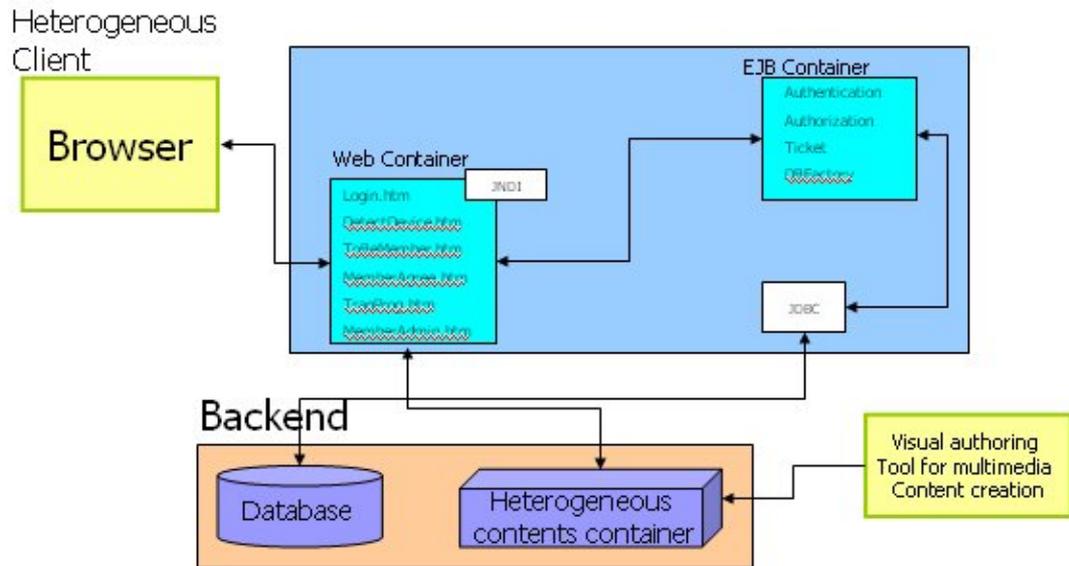


圖 16 應用 MVC 架構設計



四、系統實作

4.1. 功能對應實作

在這個小節裡，我們利用一個使用者案例圖來描述一個 Web-based 異質性的應用系統的實作。這個系統是可以提供異質性多媒體互動教學的試題網站以及提供卡片的樣板置換服務，功能如下圖所示。

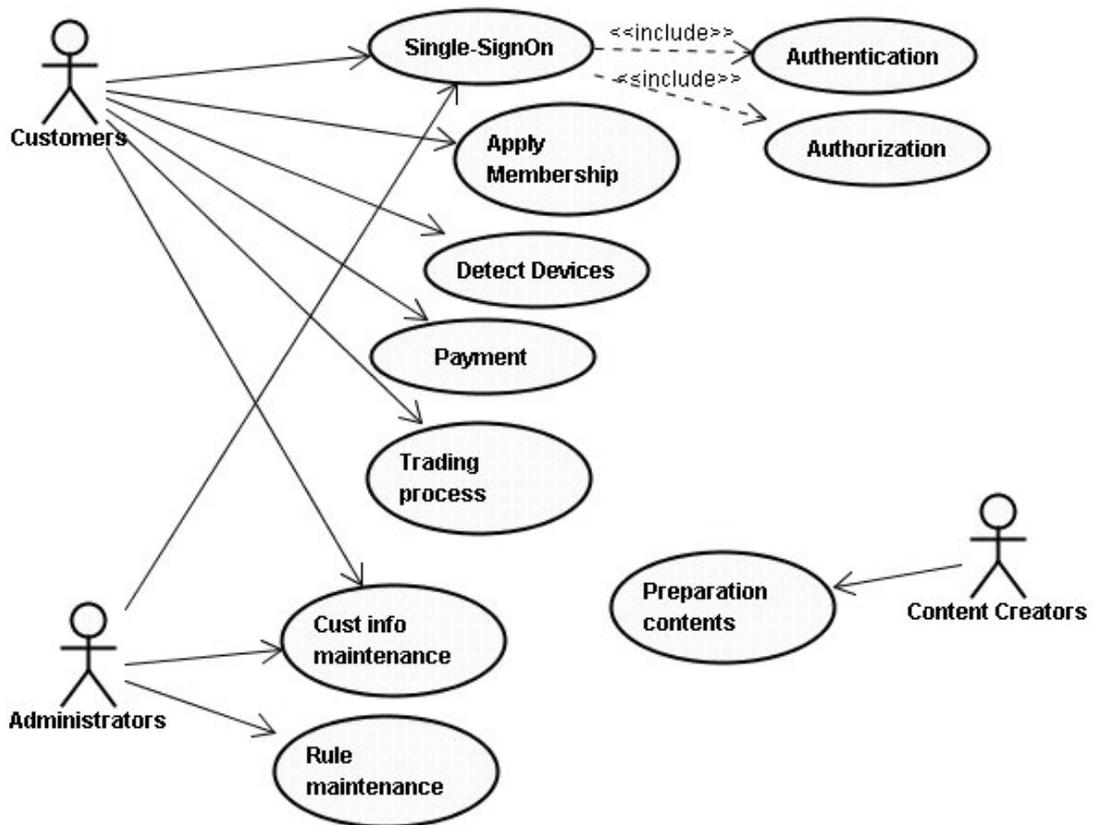


圖 17 使用者案例圖(Use case)

由圖中可以看出我們分為三種 Actors，分別是：Customers、Administrators 和 Content Creators。

- Single-Sign-On :

如同案例圖所示，此功能包含 Authentication 和 Authorization。使用者可以依照權限進入選擇不同的服務或者是點數折扣不相同等優惠。

- Apply Membership :

新客戶利用這個功能來申請加入會員並且取得會員點數。

- Detect Device :

當使用者通過系統認證授權之後，系統會依照不同的終端裝置來分派不同的網路連結位址，並通知 Content 伺服器來提供服務。

- Payment :

所有網站上的交易都靠事先的會員點數來作為存取服務的依據。當會員一開始加入時，系統會要求要繳交會費並仿照 VISA 3D Secure 的線上交易做法來取得授權，並將交易金額轉換成會員點數以方便日後存取扣繳用。

- Trading Process :

當使用者一但連線之後，只要有扣點數的交易動作，一律紀錄在資料庫中方便對帳。每當下載完成或連線播放完畢之後，系統會自動依照點數扣抵。

- Cust Info Maintenance :

提供使用介面讓使用者新增會員或系統管理者要新增、修改或停權會員資料使用。

- Rule Maintenance :

系統管理員可以依照不同的商業邏輯規則來修改系統規則設定檔。

系統可以動態依照管理者維護的資料顯示並完成扣除點數作業。系統維護填需再修改程式。

- Preparation Contents :

Content creators 在上傳編輯完成的多媒體內容之後，系統會要求需入帳號密碼來確認是否可以進行做異質性多媒體內容轉譯的工作，並依照系統設定檔中維護的待轉譯的型態、存放目錄、編號和對應點數權組的分類一一存放於 Content 伺服器中。

接下來幾個章節，我們以回應本研究的研究目標來描述實作的情況。

4.2.SSO 實作

這裡我們採用順序圖的方式來說明單一登入窗口的實作，如下圖所示，我們可以按照順序一一來做解說：

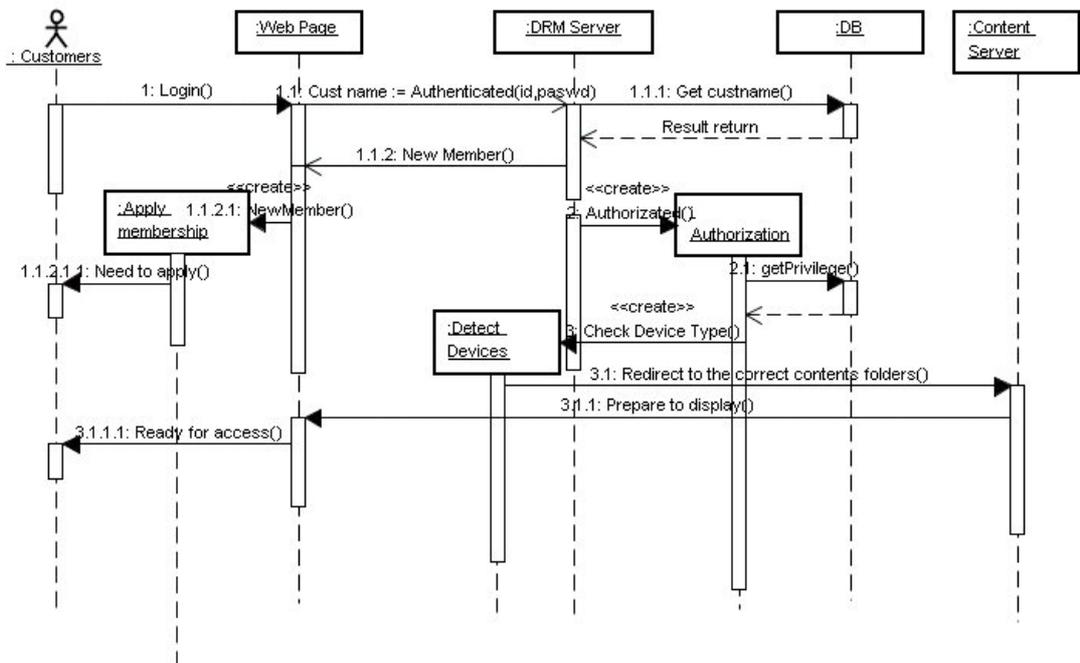


圖 18 單一登入畫面順序圖

系統可以允許客戶使用不同的終端裝置來登入系統。假設當客戶使用手機進入系統首頁時，系統會要求使用者輸入帳號及密碼。這時系統會透過 HTTP 的方式來跟 DRM 伺服器要求做身分認證。

DRM 伺服器在做認證的動作時，其實是透過 JDBC 來和關聯式資料庫查詢，若能取得相關的客戶資訊，則可確認身分；反之，則回傳訊息並告知 DRM 伺服器是 New member 的資訊。如圖中 1.1.2 所示，此時系統會轉只到會員申請網頁並開始進行新會員的申請流程。有關新會員的申請順序圖及描述，我們後半段會加以說明。

若 DRM 伺服器收到完整的客戶資訊，此時 DRM 伺服器會驅動 Authorization 物件模組並開始進行授權確認，此時會依照系統設定檔中所設定的權限群組加以確認分派。

最後會交由 Detect Device 物件模組來偵測到使用者目前使用的裝置，假設是手機連線，則會告知 Content 伺服器的 Content Agent 要轉只到手機的對應內容網址。此時手機畫面就會出現手機對應的多媒體內容網頁供使用者存取使用。

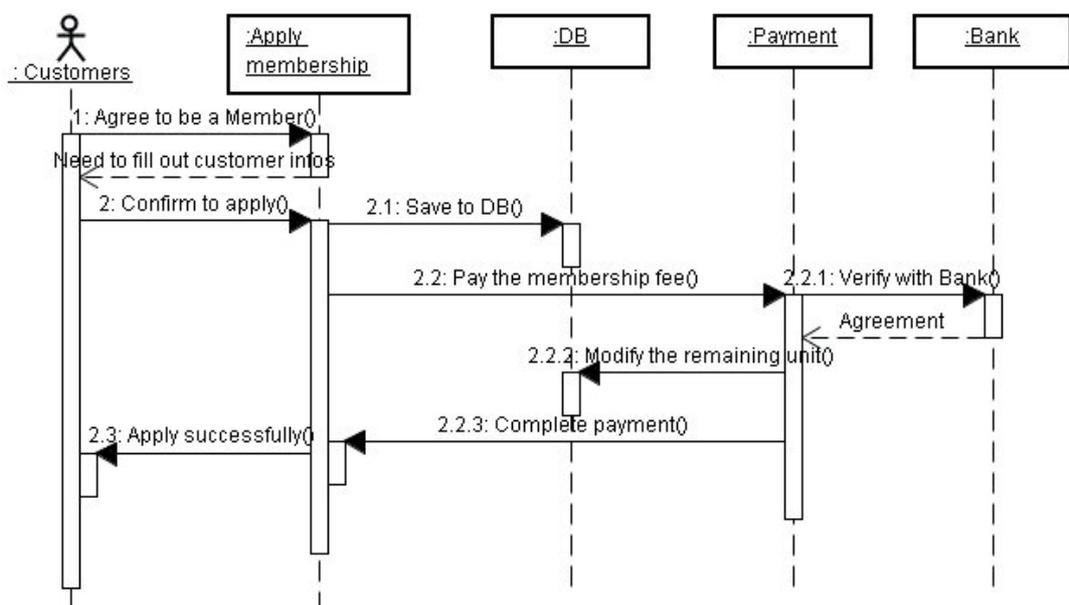


圖 19 申請新會員順序圖

至於新會員的申請狀況，我們另外以圖 19 來表示。首先，使用者會選擇是否同意成為新會員，接著進入基本資料的填寫畫面。等到填寫完畢之後，使用者會按下確認按鈕，此時 Membership agent 會去儲存資料庫的動作。

同一時間，它也會送訊息到銀行去做確認並取得授權，當取得授權之後，系統會將會員費用轉回會員點數並存回資料庫中。等到一切儲存完畢，會回復使用者並告知可以開始使用系統所提供的服務。

4.3. 早期繫結(early binding)實作

我們利用 Design pattern-Adapter 的做法來實作 Heterogeneous Content Agent。誠如 2.3 章節中描述，在此我們將交通大學軟體工程實驗室所開發的數種多媒體轉譯器重新包裝如圖 20 所示。目前實驗室有大約四種轉譯器，分別是 ToHTML、ToJar、ToScorm 和 ToMPEG4。

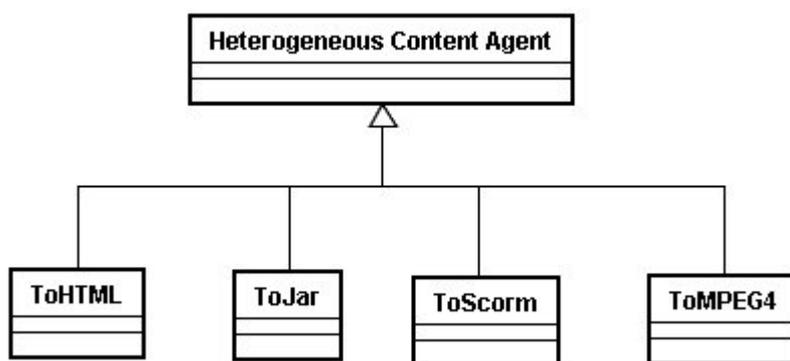


圖 20 Content Agent 物件圖

我們可以用描述檔的方式存放在資料庫中，這個 Content Agent 物件模組會在每次 Content Creators 要上傳資料時被驅動。它會依照目前系統設定好的轉譯內容及目錄對應來自行依照系統給定的編號有順序的存放。

至於詳細工作流程，我們參考圖 21 來加以講解：整個系統所使用的多媒體編輯是來自智勝國際[26]所開發的編輯手軟體，至於完整的互動式多媒體編輯及應用請參考[27]，本研究對於互動式多媒體的內容不加以探討，僅對現行系統加以整合管理。

下圖中的左半邊從 Content creator 到 Encrypt 和 Content up-loader 都屬於[27]的研究範圍。至於圖 21 中下部分用虛線框起來的部分，則是整合交通大學軟體工程實驗室已開發的轉譯器，部分研究可參考[23]。

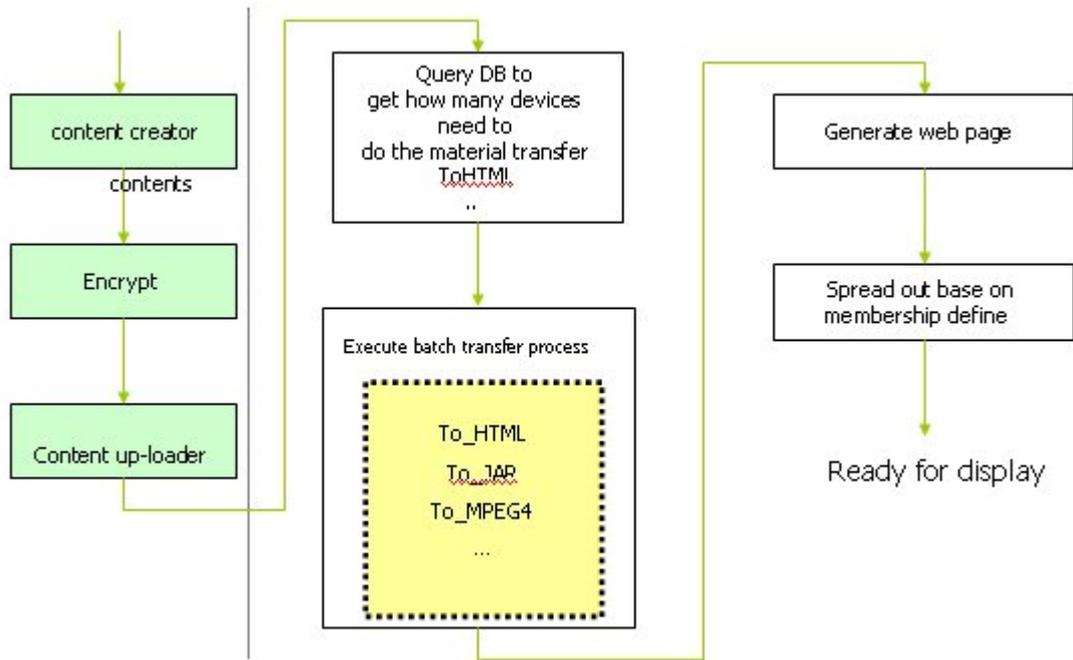


圖 21 Content Agent 實作流程圖

如上圖所示，當 Content creator 準備上傳時，系統會呼叫 Content agent 準備作轉譯的動作。首先，系統會先到資料庫去取得目前系統管理者所設定的轉譯範圍、目前系統編號及權限控管目錄存放位址等資訊。接下來，則開始進行轉譯動作；代轉譯完成之後，系統會針對權限分組重新產生網頁並將權限控管當參數存放在網頁中。若日後權限控管規則改變，此程式譯無須作更動。

4.4. 權限控管流程實作

接下來，我們來看看權限控管流程實作。有關於 DRM 的實作，由於前一章我們已經有說明過了，這裡直接使用順序圖來觀看整個使用者存取系統所提供的服務狀況。使用順序圖可以簡單易懂並可以看出物件模組間的互動。

在本章節，分成兩種情況來說明：線上存取和離線存取。

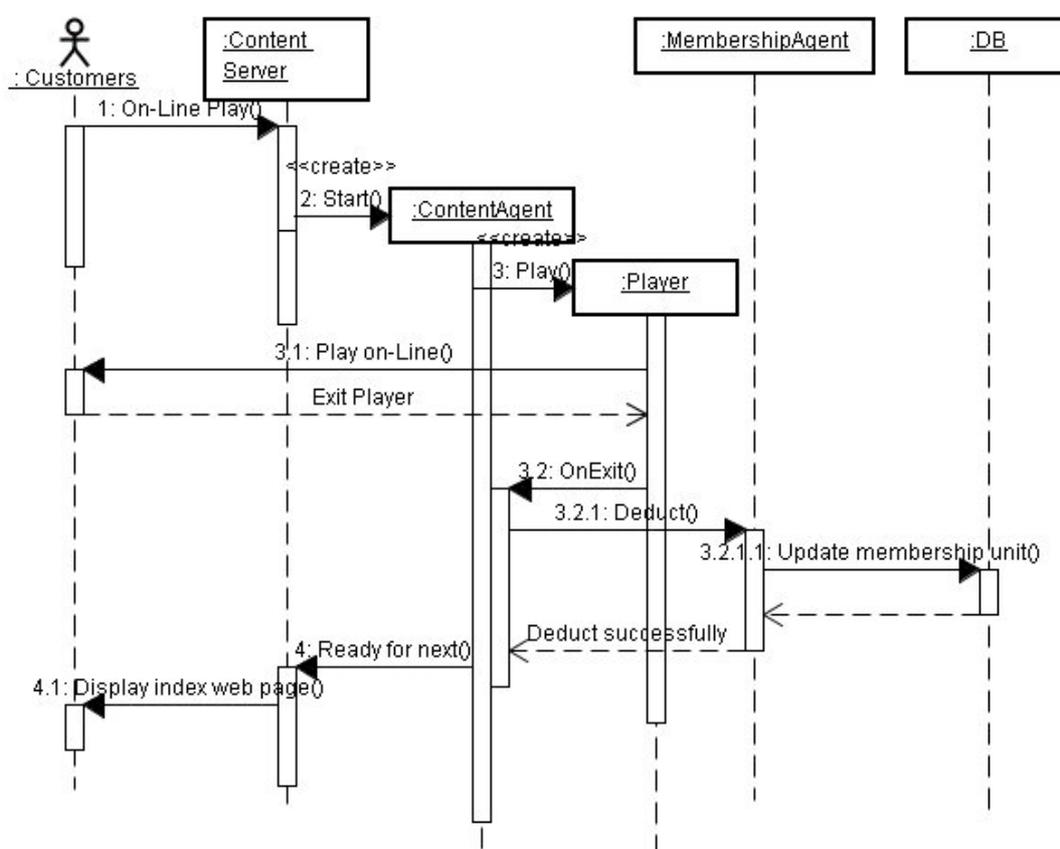


圖 22 連線存取的順序圖

圖 22 表示連線存取的順序圖，這個順序圖發生在使用者已經通過 DRM 認證且獲得授權之後。

假設使用者使用個人電腦連線進入，使用者只能選擇個人電腦用的多媒體互

動試題來進行測驗。當使用者選定一個線上是提並連線播放時，此時系統會提供鑲嵌在 IE 瀏覽器的播放器給使用者使用。當使用者播放完畢時，Content Agent 會通知 Membership Agent 負責依照該試題播放一次所需點數到資料庫扣抵。

接下來，使用者可以繼續瀏覽其他試題，當使用者選定另一是提要求播放時，Membership Agent 會先確認使否有足夠的會員點數可以播放。若有則繼續允許連線播放；反之，則回到 Payment。

至於離線播放的情況，我們假設現在使用者是用 PDA 連線進來系統要求卡片查看並下載。順序圖請看圖 23：

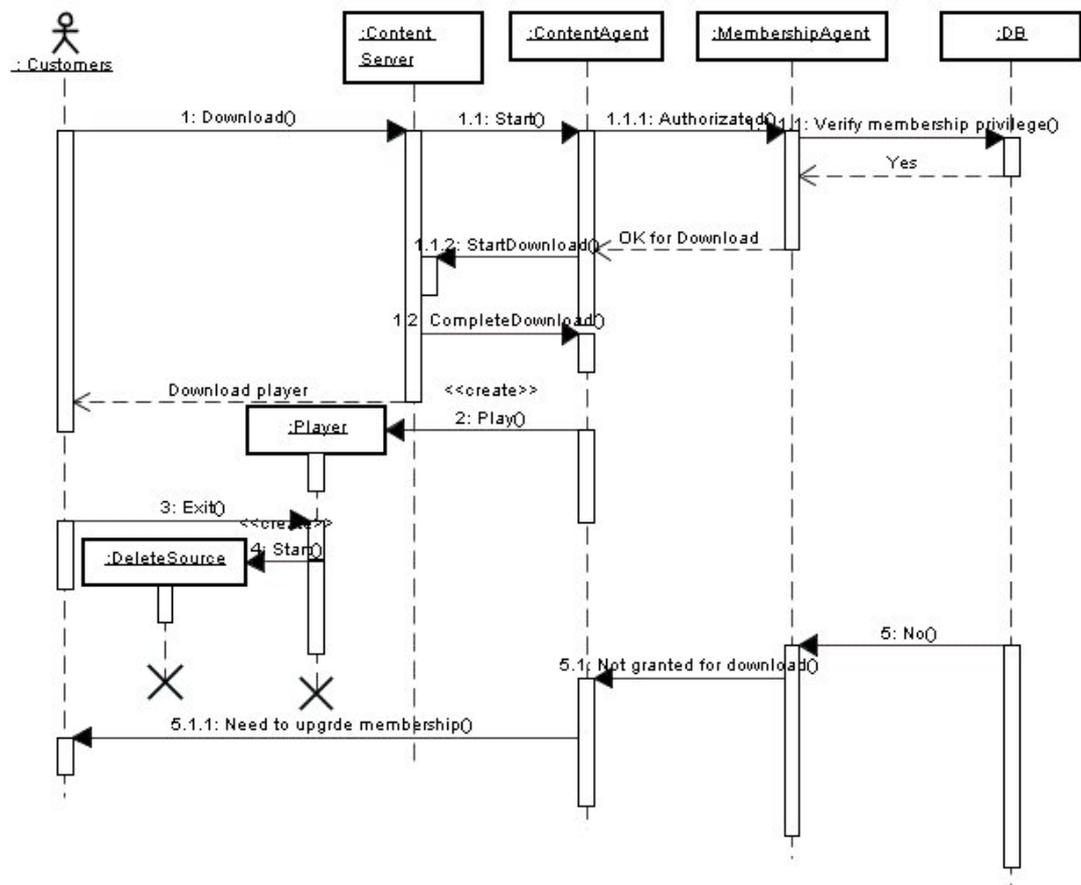


圖 23 離線播放的順序圖

當使用者要求下載到 PDA 來播放時，Membership Agent 會先確認使否有足夠點數及權限。如果回傳是"YES"，則允許下載。

此時 Content Agent 也會跟著 Player 下載到 PDA 中，一旦完成下載，系統會問使用者是否播放。若使用者確認執行，Player 便會開始執行直到使用者離開。

當使用者完成下載時，Membership Agent 一樣會依照點數扣抵並進行資料庫的修改。

等到使用者要離開 Player 的時候，Content agent 會啟動 Delete source 的模組，將下載的多媒體播放主檔刪除。倘若，系統不正常離開或使用者干擾中斷播放，等到下一次使用者開啟 Player 時，Content agent 馬上會自行偵測並刪除檔案。



五、實作範例

在本章中，我們將以呈現系統畫面的方式，介紹系統中所擁有的功能、機制及其流程。

5.1. 終端使用者範例

在展示系統畫面之前，我們先以一個範例流程圖(圖 24)開始進入狀況：

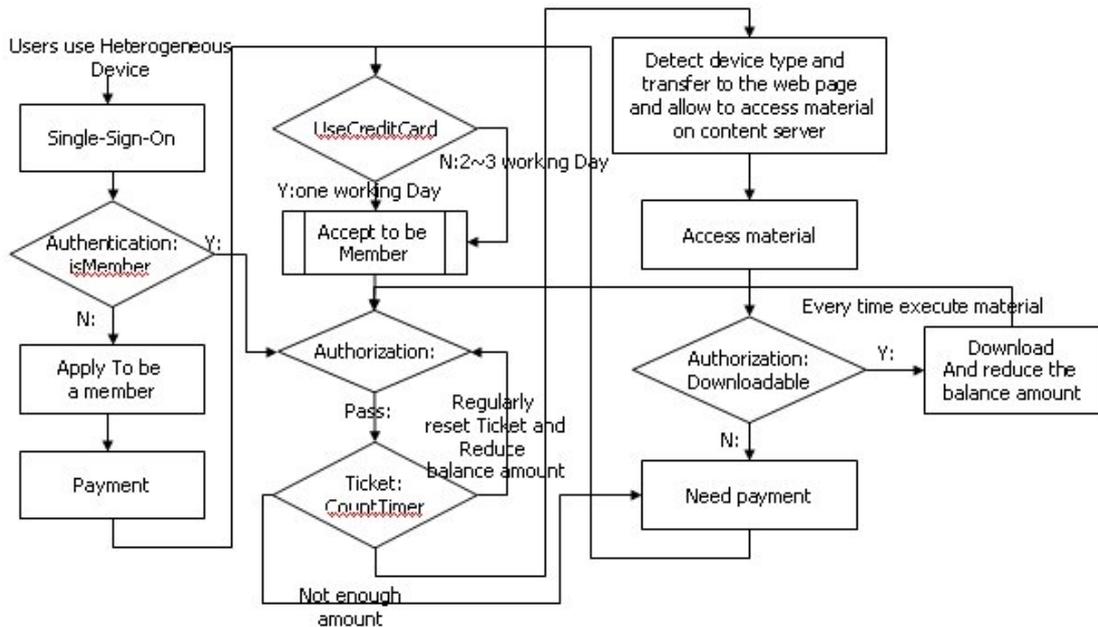


圖 24 客戶端的流程範例

假設系統規定的權限控管是分為三個等級，分別是：等級 A，可以看全部的教材內容；等級 B，能看 80% 的內容；等級 C，則只能看 60% 的內容。會員也可以依照匯款來達成繳交會費的動作，但要 2~3 個工作天。若用信用可則只要一個工作天即可開始使用系統所提供的服務。

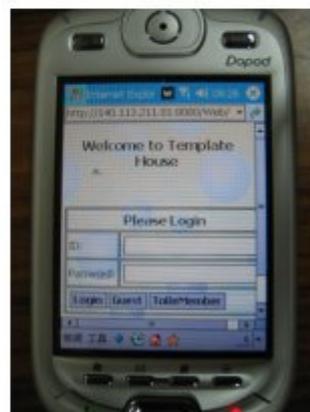
接下來我們開始看一些主要畫面在 PDA、手機和個人電腦的呈現畫面範例。下圖我們可以看到不管使用者用哪一種終端裝置連線，畫面都是同樣的。甚至連系統管理者或多媒體內容提供者都是利用這個登入畫面來連結相關維護系統或上傳並轉換相關的多媒體內容。



PC



Mobile



PDA

圖 25 單一登入畫面範例

5.2.新會員申請範例

由下圖可以看出，即使使用小螢幕的終端裝置，仍是可以共用同一網頁程式。使用者可以確認加入會員之後才開始填寫相關個人資料並確認儲存。

會員申請同意書

在你申請加入「新材之家」會員（以下簡稱會員）前，請詳細閱讀下列會員約定條款內容，如同意約定條款內容，即可使用本網站提供之服務；而不論是否新會員，均須於同意接受本約定條款後，始得使用或繼續使用本網站之各項服務。

一、接受條款
 1. 本協議同意書為本網站會員應行遵守之約定及相關權利義務關係；惟
 本網站會員不得對本同意書內容，可選擇不申請加入或為會員申請退出；
 一旦加入會員即視同本網站提供之各項服務，視為會員已閱讀、了解並同意
 本同意書之約定。

2. 本站保留隨時修改本同意書內容之權利，且不需另行通知，本
 網站會員應隨時到本網站瞭解修改內容。

3. 本網站保留隨時停止變更或刪除內容或禁止一該會員帳戶使用之
 權利，且無須事先通知或通知後再通知。但修改、更新或刪除該會員
 帳戶服務，其可能產生之證據，不獲保護。本網站對任何會員或第三人均
 不負任何責任。

同意 不同意

填寫個人資料

* 帳號：請輸入3至16個字元的小寫英文字母、數字、及「_」符號，但第一個字元須為英文字母，系統輸入方便，請鍵錄使用小寫字元

* 密碼：請輸入4個字元以上的大文字字為基本，以英文字母為限，恕不含有空白鍵及「_」，把請於輸入密碼時同時三個字元以上密碼

* 確認密碼：再次確認你所輸入的密碼

* 真實姓名：

* 性別： 男 女

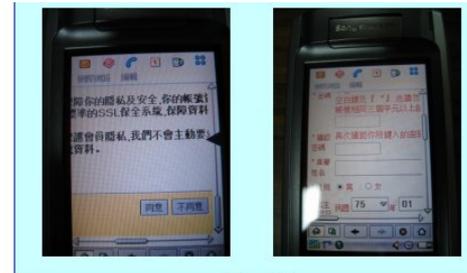
* 出生年月日：日期 / /

* 聯絡電話：

* E-mail：

* 通訊地址：郵政區號： 地址：

* 收貨地址：



Mobile

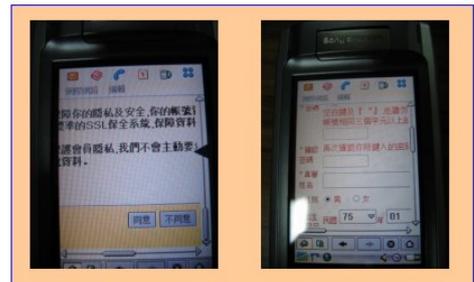


圖 26 新會員申請範例

5.3. 交易範例

整個交易流程會類似圖 27 和圖 28 所示，使用者進入系統來到了歡迎光臨的頁面。此時表示使用者已經通過 DRM 伺服器的身分驗證及授權。

接下來使用者可以選擇服務內容：卡片區、試題區或進入廣告區閒晃也可以。

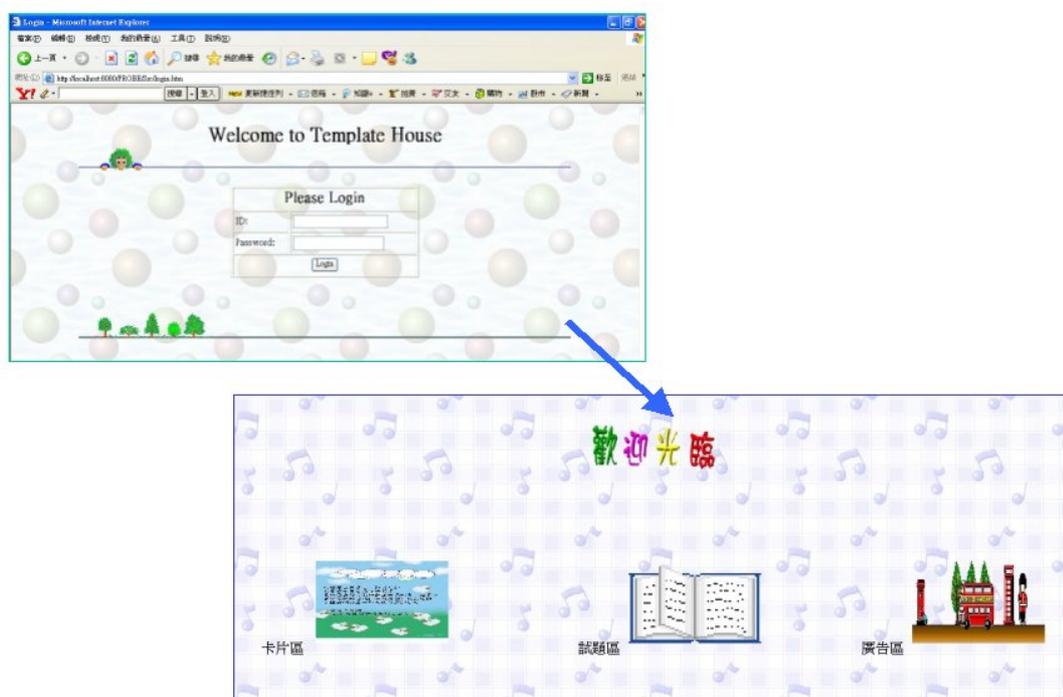


圖 27 歡迎畫面

假若是進入試題區，則如下圖會看到科目編列。所有的科目皆可動態異動，糗無須更改程式或網頁設計。

假設使用者選擇英文科，進入之後則顯示分級情況。這些分及一樣是可以動態變更。

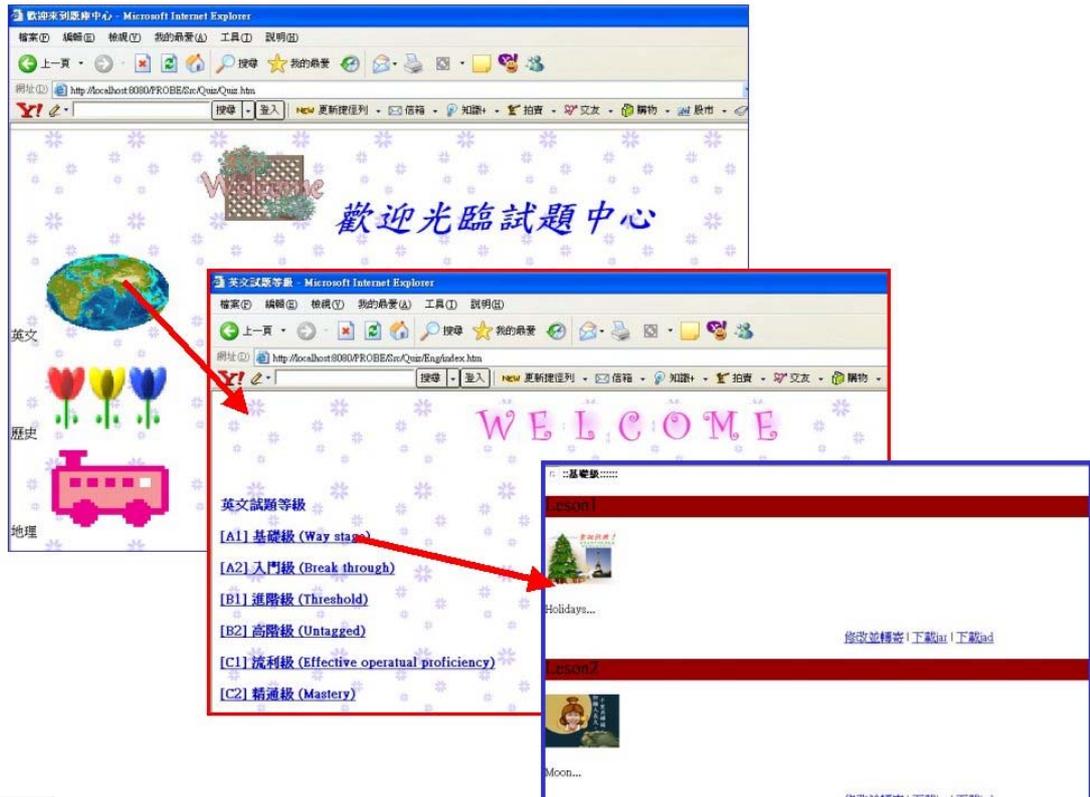


圖 28 試題畫面

若使用者選擇卡片區，則系統會自動偵測並帶到相對應的 PDA 目錄。如圖 29 所示



PDA

圖 29 PDA 卡片範例

六、結論

6.1. 總結

企業組織的成長，除了營收的增加外，往往也會帶來應用系統的擴充及整合問題。終端使用者通常會遭受到應用系統過多不同入口且重複登入的問題；而系統維護者則是需要花費許多人力來開發類似的功能模組在各種不同終端裝置的應用系統上。

有鑒於異質性的應用系統需求日益增加，更是讓上述的問題更加值得被重視。本研究論文提供了一些系統設計方法來解決這些問題，主要設計理念為：

(1) Single-Sign-On :

提供使用者一個簡單一致的系統連結位址，使用者方便使用相關系統。整合所有系統的登入畫面還可以減輕系統開發人員的維護人力並降低成本。況且還可達成重複使用的境界，方便系統管理者輕鬆管理所有人員的權限控管。

(2) 內容早期繫結(Content early binding) :

有效率的整合管理分散各地的異質性的多媒體內容。不僅可以動態的依照商業邏輯管理，還可減輕 Content creators 的負擔。從此之後，只要系統管理者設定好相關參數，所有的異質性的多媒體內容就會在上傳 Content 伺服器時自行展開。

(3) DRM 機制引用 :

引用數位式權限管理的機制，除了可以建立完整的權限管理機制外，還可以有效且適當的保護富創意的多媒體內容。

6.2. 未來發展方向

在此我們提出一些對未來研究、發展方向的建議。

(1) 目前圖素(Pixels)的計算雖可以讓手機或 PDA 的畫面顯示和 PC 同樣的畫面。但是畫面的設計便不能過於複雜，否則圖形會小到看不見。未來可以在 Context-awareness 及 Context-adaptation 兩方面更深入去研究來更符合異質性的終端裝置的需求。

(2) 網路上的交易紀錄，除了可以用來對帳外，在現今講求服務導向(Service oriented) 的社會，可以利用 Data Mining 的技術來做有效的資料分析，進而更有效率的提高服務品質。

(3) 網路上的交易是越來越頻繁，但是目前仍沒有一個可以絕對安全的交易方式。就連 VISA 和 Master 等國際組織都仍在尋尋覓覓，未來也可在網路交易安全部分作進一步研究。

參考文獻或資料

- [1]. E. Gamma, R. Helm, R. Johnson, and J. Vlissides, 物件導向設計模式, 葉秉哲譯, 中譯本二刷, 培生, 2003/5
- [2]. VISA, <http://www.visa.com>
- [3]. Digital Rights Management (DRM) ,
<http://www.dlib.org/dlib/june01/iannella/06iannella.html#Top>
- [4]. 電子商務導航, 資策會電子商務應用推廣中心, <http://www.ec.org.tw/>
- [5]. Rich Helton and Johennie Helton, Java Security 全方位解決方案, 楊松諺/上官飛鳳譯, 碁峯, 2004/5
- [6]. John O' Donahue, Java 資料庫程式設計, 陳怡和/林學文譯, 碁峯, 2004/5
- [7]. 王森, 手機.PDA 程式設計入門, 碁峯, 2003/12
- [8]. 結城浩, Design Patterns 於 Java 語言上的實習應用, 博碩文化, 2001
- [9]. 昇陽電腦教育訓練中心, 從 Visual Basic 到 Java 完全手冊, 碁峯, 2003/6
- [10]. Java World, <http://www.javaworld.com.tw/jute/index.html>
- [11]. JAAS, <http://java.sun.com/security/jaas/doc/api.html#Overview>
- [12]. 電子商務安全, http://www.cs.ccu.edu.tw/~ccc/article/ec_secure.htm
- [13]. William Stallings, 密碼學與網路安全原理與實務, 巫坤品/曾至光譯, 出版二刷, 碁峯, 2002/2
- [14]. Timothy A. Howes, Ph.D., Mark C. Smith, Gordon S. Good, Understanding and Deploying LDAP Directory Services, second edition, Addison-Wesley, 2003
- [15]. Ian Sommerville, Software Engineering, 6th Edition, Addison-Wesley, 2000
- [16]. Deng-jyi Chen, Ming-Jyh Tsai, Jia-chen Dai, and David TK Chen, "Visual Based Software Construction: Visual Requirement Authoring tool and Visual Program generator", International Conference Proceedings, ICS. 2004
- [17]. R. Greg Lavender, Douglas C. Schmidt, "Active Object: An Object Behavioral Pattern for Concurrent Programming", ISODE Consortium Inc., Austin, 1996
- [18]. Emad Barsoum, "Template Thread Library",

<http://www.codeguru.com/Cpp/W-P/system/threading/article.php/c5687>

- [19]. Mohamed Fayad, Douglas C. Schmidt. "Object-Oriented Application Frameworks".
Communications of the ACM, Special Issue on Object-Oriented Application
Frameworks, Vol. 40, No. 10, October 1997
- [20]. Douglas C. Schmidt, Ralph E. Johnson, Mohamed Fayad. "Software Patterns".
Communications of the ACM, Special Issue on Patterns and Pattern Languages, Vol. 39,
No. 10, October 1996
- [21]. 侯俊傑，物件導向的精髓: 多型與虛擬，初版，松岡，台北，1998
- [22]. Simon Brown, Robert Burdick, Jayson Falkner, Ben Galbraith, Rod Johnson, Larry Kim,
Casey Kochmer, Thor Kristmundsson，專業 JSP 程式設計 第二版，林學文/陳怡
和 譯，碁峯，2003/5
- [23]. 王宇涵，互動式多媒體樣板在異質性手持行動裝置上之呈現及其樣板轉譯器
之實作，國立交通大學，碩士論文，2005/6
- [24]. Carter，LDAP 系統管理，蔣大偉 編譯，O' REILLY，2003/10
- [25]. RFC，<http://www.rfc-editor.org/>
- [26]. 智勝國際，<http://www.caidiy.com/caidefault.htm>
- [27]. 江書瑩，互動式多媒體的視覺化劇情編輯機制應用於多媒體試題樣板套用系
統的實作，國立交通大學，碩士論文，2005/6