

國立交通大學

資訊科學與工程研究所

碩 士 論 文

樣板式多媒體內容在手持行動裝置之情境
及內容協調與呈現-以樣板式的多媒體英文
試題為例子

The Context Aware and Content Adaptation of Template Based
Multimedia Presentation on Handset Device - Using Template
Based English Test Questions with Multimedia Contents as
Examples

研 究 生：洪啟彰

指 導 教 授：陳登吉 教授

中 華 民 國 九 十 五 年 八 月

樣板式多媒體內容在手持行動裝置之情境及內容協調與

呈現-以樣板式的多媒體英文試題為例子

The Context Aware and Content Adaptation of Template Based
Multimedia Presentation on Handset Device - Using Template Based
English Test Questions with Multimedia Contents as Examples

研究生：洪啟彰

Student：Chi-Chung Hung

指導教授：陳登吉

Advisor：Dr. Deng-Jyi Chen



A Thesis
Submitted to Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in
Computer Science
August 2006
Hsinchu, Taiwan, Republic of China

中華民國九十五年八月

樣板式多媒體內容在手持行動裝置之情境及內容協調與呈現- 以樣板式的多媒體英文試題為例子

學生：洪啟彰

指導教授：陳登吉 博士

國立交通大學資訊工程學系碩士班

摘要

由於行動裝置的使用普及率越來越高，更多的行動裝置必須支援各式各樣的多媒體內容才能滿足使用者需求，然而大部分的多媒體內容僅能在桌上型電腦上呈現播放，業者通常必須重新設計和在桌上型電腦上的同樣內容，才能使同樣的內容在行動裝置上順利瀏覽播放，但這樣的重新設計內容勢必對設計內容的業者造成一大負擔。

本研究探討多媒體內容的改編及轉換機制，以本實驗室開發的樣板式多媒體英文試題製作軟體為例子，利用情境感知(Context aware)方式設計演算法進行分析協調版面的動作，並期許轉換協調後的內容清楚、不失原語意及被使用者容易瀏覽。最後並對一般傳統作內容改編協調(Content adaptation)的兩個方法靜態、動態改編轉換分析其好壞，提出靜態及動態混合方法來達到空間及時間的較佳化。

本研究實作了內容改編協調伺服器(Adaptation server)，可將原本在桌上型呈現的多媒體內容依照參數的給定改編協調到一般傳統行動裝置可以支援的格式，並搭配 Web-based 的協調管理介面，提供簡便自動化的改編協調。此外設計客戶端的播放器除了驗證協調後畫面的可行性，並用以模擬內容轉換存取的三種模式-靜態、動態、靜態及動態混合方式，本播放器可播放格式為 XML-Based 的 XMG 格式，純為模擬而自訂。

The Context Aware and Content Adaptation of Template Based Multimedia Presentation on Handset Device- Using Template Based English Test Questions with Multimedia Contents as Examples

Student: *Chi-Chung Hung*

Advisor: *Dr. Deng-Jyi Chen*

**Department of Computer Science and Information Engineering
National Chiao Tung University**

Abstract

The growing popularity of mobile devices demands various multimedia presentation contents on these devices. However, most multimedia presentation contents can only be supported on desktop computer which has more computing power and less constraint than mobile devices. For content providers, they often have to replicate design and implementation those multimedia contents which originally were supported on desktop computer to fit on different kinds of Mobile devices. Eventually, more efforts and cost must be spent for content providers.

In this thesis research, we discuss the technology of multimedia content adaptation and content negotiation under mobile devices. We use the Template Based Multimedia English Test Questions generated by the template based multimedia contents creation tool as example to design a multimedia content adaptation and content negotiation algorithm for the presentation under different kinds of mobile devices. Also, a static and dynamic mixed adaptation method is implemented to balance the space and speed requirement during the multimedia content adaptation for mobile devices. The pros and cons based on their adaptation time and disk usage are analyzed to compare with the traditional static and dynamic adaptation approach.

Based on the proposed algorithms, we constructed an adaptation server which can automatically analyze, adapt, trans-code, and covert the original multimedia contents to a specified target mobile device presentation format. We also designed a web-based client application program in the target mobile device (such as PDA) to verify the presentation of those adapted multimedia contents.

誌謝

本論文承蒙指導教授陳登吉老師的耐心指導與教誨下，得以順利完成，在此致上對老師無限的感謝。陳老師是個很熱心負責的一位教授，除了在我們研究領域上的指引及點醒，更重要是讓我們學習到作研究的觀念及態度，此外陳老師也很關心我們生活上的問題。

另外在此我必須感謝在交大和我同甘共苦、互相砥礪的同學及朋友，尤其是研究室的成員，我覺得我可以認識他們是我一生的榮幸，和他們相處很愉悅，在課業上、生活上得以互相指導、勉勵，也讓我學到很多書本上學不到的東西。

最後我必須感謝我的家人，尤其是我父母，雖然我自高中就開始在外求學至今，但背後如果沒有他們的支柱、沒有他們的栽培，也就沒有今天的我。今天有高學歷的我，我將歸功於我父母，也期望我自己能在將來有好的成就、好的作為，不枉費他們的栽培。



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 研究動機與目標.....	1
1.2 研究方法.....	1
1.3 研究範圍.....	2
1.4 章節概要.....	3
二、相關研究、文獻探討.....	4
2.1 情境感知改編協調.....	4
2.1.1 情境資訊(context information).....	4
2.1.2 情境感知(Context aware).....	5
2.1.3 內容改編協調(Content Adaptation).....	7
2.2 樣板式多媒體內容.....	8
2.3 多媒體文件格式之腳本語言.....	9
2.3.1 HTML.....	9
2.3.2 SMIL.....	10
2.3.3 SVG.....	10
2.3.4 EBS (編輯手劇情描述語言).....	11
2.4 多媒體轉換相關工具.....	11
2.4.1 XSLT.....	11
2.4.2 Lame.....	12
2.4.3 Devil.....	13
2.4.4 MS Window Media Encoding Script.....	13
三、系統功能需求分析.....	14
3.1 內容伺服器系統評估.....	14
3.2 改編協調伺服器功能分析.....	15
3.3 Web-based協調介面功能需求.....	17
3.4 行動裝置應用程式開發功能需求.....	18
四、系統架構.....	20
4.1 系統設計說明.....	20
4.1.1 系統架構圖.....	20
4.2 子系統架構說明.....	21

4.2.1	內容伺服器資料庫系統之維護.....	21
4.2.2	改編協調伺服器之開發.....	21
4.2.3	應用端程式開發.....	25
五、	系統設計與實作	27
5.1	已改編協調的內容伺服器之維護實作.....	27
5.1.1	試卷資料庫規格.....	27
5.1.2	後端程式之開發.....	28
5.2	改編協調伺服器之設計與實作.....	29
5.2.1	描述檔剖析器製作.....	29
5.2.2	協調引擎設計製作.....	32
5.2.3	目標模組(Target module)製作.....	42
5.3	應用端之設計與實作.....	43
5.3.1	Web-based改編協調開發製作.....	43
5.3.2	PDA Application 客戶端播放器製作.....	47
六、	應用範例	53
6.1	管理改編協調者的操作畫面.....	53
6.2	PDA客戶端瀏覽裝置操作畫面.....	58
6.3	改編協調前後內容呈現差別.....	60
6.4	靜態和動態轉換及混合方式轉換數據分析.....	66
七、	結論	68
7.1	總結.....	68
7.2	未來發展方向.....	69
	參考文獻或資料	70

表目錄

表 1：靜態和動態的好壞處.....	8
表 2：兩種內容伺服器架設環境(content server)	14
表 3：各種多媒體轉換所花費時間.....	23
表 4：多媒體試題試卷資料庫資料表.....	27



圖目錄

圖 1:一般決策引擎模組	6
圖 2 : adaptation server運作過程	15
圖 3 : web-based介面協調管理過程	17
圖 4 : web-based介面批次協調過程	18
圖 5 : 行動裝置上應用程式操作過程.....	19
圖 6 : 系統架構圖.....	20
圖 7 : 已改編協調的內容伺服器後端應用程式架構運作模式.....	21
圖 8 : 多媒體英文試題在PC及PDA上的畫面.....	22
圖 9 : 多媒體英文試題協調後在PDA上的畫面.....	23
圖 10 : Adaptation server詳細內部架構圖.....	24
圖 11: 應用端程式粗略架構圖.....	25
圖 12 : 管理協調者運作圖.....	25
圖 13 : PDA application 三種模式瀏覽端運作圖.....	26
圖 14 : EBS描述檔範例圖	29
圖 15 : parsing script運作過程圖	30
圖 16 : Parsing Tree	30
圖 17 : 英文試題版型分配可能.....	32
圖 18 : 英文試題段落分配範例.....	33
圖 19 : 頁數分配演算法.....	34
圖 20 : 頁數分配演算法範例.....	35
圖 21 : Temporal translation整體運作過程	35
圖 22 : 版型分配比例估計演算法.....	36
圖 23 : 計算英文字長度範例.....	36
圖 24 : 英文句子斷行演算法流程圖.....	37
圖 25 : 圖片縮放比例演算法範例說明.....	38
圖 26 : 不同圖縮放長寬比例差別.....	39
圖 27 : 推薦選擇版型演算法例子.....	39
圖 28 : 圖 27 例子的數據分析.....	40
圖 29 : 運用XSLT轉換相對應的多媒體腳本XML文件	42
圖 30 : Temporal XML範例	42
圖 31 : 一份根據temporal xml 轉換到 HTML的XSLT style-sheet	43
圖 32 : 協調者之web-based運作圖	44
圖 33 : 出題使用者從登入到選題出試卷的第一階段流程.....	45
圖 34 : 使用者輸入參數即時回傳確認動作流程圖.....	45
圖 35 : 使用者進行批次轉換試題的過程圖.....	46

圖 36：批次轉換至上傳過程.....	47
圖 37: PDA端應用程式運作圖	47
圖 38: PDA應用程式之下載試卷的子模組流程圖	48
圖 39: PDA使用者對於下載試題的過程圖	49
圖 40: 一份XMG描述格式範例	50
圖 41: XMG轉換成樹狀結構圖	50
圖 42: 管理協調者登入畫面	53
圖 43: 進入編輯一份試卷畫面	54
圖 44：篩選資料庫試題並挑選試題成試卷	54
圖 45: 改編協調的第一步驟-參數設定	55
圖 46: 改編協調的第二步驟-自動批次轉換.....	55
圖 47：改編協調的第三步驟-手動調整選取	56
圖 48: 改編協調第四步驟-轉換並上傳至試題平台	56
圖 49: 轉換上傳成功畫面	57
圖 50: PDA試題播放器-登入畫面及偵測device profile	58
圖 51：PDA試題播放器-列出試卷及下載試卷	58
圖 52：PDA試題播放器-下載試卷完成及播放試卷試題	59
圖 53：PDA試題播放器-播放試題及選擇題目	59
圖 54：範例一 640x480 呈現畫面.....	60
圖 55：範例一 320x240 呈現畫面.....	60
圖 56：範例一 240x320 呈現畫面.....	61
圖 57：範例一 240x240 呈現畫面.....	61
圖 58：範例一 240x240 呈現畫面.....	61
圖 59：範例二 640x480 呈現畫面.....	62
圖 60：範例二 320x240 呈現畫面.....	62
圖 61：範例二 240x320 呈現畫面.....	63
圖 62：範例二 240x240 呈現畫面.....	63
圖 63：範例二 160x180 呈現畫面.....	63
圖 64：範例三 640x480 呈現畫面.....	64
圖 65：範例三 320x240 呈現畫面.....	64
圖 66：範例三 240x320 呈現畫面.....	65
圖 67：範例三 240x240 呈現畫面.....	65
圖 68：範例三 160x180 呈現畫面.....	65
圖 69：三種模式時間差異圖.....	66
圖 70：三種模式磁碟空間使用量(同一種環境).....	67
圖 71：三種模式磁碟空間使用量(不同環境).....	67

一、緒論

1.1 研究動機與目標

由於行動裝置的普及，越來越多的行動裝置必須支援各式各樣的多媒體內容才能滿足使用者需求，然而大部分的多媒體內容僅能在桌上型電腦上呈現播放，業者通常必須重新設計和在桌上型電腦上的同樣內容，才能使同樣的內容在行動裝置上瀏覽播放，但這樣的重新設計內容勢必對設計內容的業者造成一大負擔。

本研究探討多媒體內容的轉換機制，藉由情境感知(Context Awareness) [1] [2] [6] [9] 和內容改編協調(Content Adaptation) [6] [9] 的技術來解決在異質環境下的多媒體內容呈現，減少內容提供製作者的設計內容所花費的時間及開發成本。我們以本實驗室開發的樣板式多媒體英文試題製作軟體為實驗的例子，設計協調演算法以自動化快速轉換內容及調整版型，並對一般傳統作內容改編協調的兩個方法靜態(static)及動態(dynamic) [2] [4] [5] [12] [14] 分析其好壞，提出靜態及動態混用方法來達到空間及時間的較佳化。

改編協調的目標並不在乎轉出來的多媒體內容格式，而是能夠正確掌握內容的資訊以及每個多媒體物件的座標位置及大小，使得轉出來的不失原內容創作者所想表達的意涵，並且能夠在異質環境下順利播放。

本研究實作內容改編協調伺服器(Content Adaptation Server)，可將原本在桌上型呈現的多媒體內容依照參數的給定而重新安排版面並轉譯到一般傳統行動裝置可以支援的格式，例如：HTML [15]、SVG [16]、SMIL [17] 等等。此外除了協調轉換之外，更探討使用靜態及動態的轉換過程混合方式，希望在內容改編協調的過程中分析多媒體內容，將轉換比較花時間的動作以靜態轉換方式完成(例如: audio、video轉換)，而對於版型調整或比較不花費時間的轉換以動態來操作，而本研究將以實驗來量測三種不同形式的改編協調(靜態、動態、靜態及動態混合方式)的磁碟空間使用量及轉換時間數據分析。

1.2 研究方法

本研究必須整合目前的現存英文內容試題平台，因此必須先了解原來的系統結構及存放在平台裡的多媒體內容描述檔格式，進而規劃出整個系統架構及需求，而後評估及實作，初步可以分為下列方法步驟：

- 了解現存系統、及文獻探討：
包括試題平台、多媒體試題描述格式語言、傳統情境感知(Context aware)、內容改編協調(Content Adaptation)技術探討、傳統多媒體腳本描述語言、多媒體轉換。
- 系統功能分析：
除了主要分析改編協調伺服器(Adaptation Server)功能之外，還必須為現存的多媒體內容伺服器(Content Server)規劃新介面以及兩個應用層面之功能需求-管理協調介面功能、行動裝置播放器設計。
- 系統設計規劃：
清楚目前的試題平台後，必須獨立規劃出新系統，並和舊系統整合。依需求設計演算法，找出文獻探討可用之處來設計我們主要的內容改編協調伺服器並訂定子系統之間的溝通模式及介面。
- 系統實作：
依各個子系統，選定開發工具，進程式撰寫並一邊進行單元測試工作。
- 實際運作評估及數據量測：
除了確定改編協調正常之外，並依照靜態協調轉換、動態協調轉換、靜態及動態混合方式來量測存取時間差異及磁碟空間使用量差異，分析評估他們彼此之間的好壞。



1.3 研究範圍

本研究相關名詞及參考範圍，敘述如下：

- PC上的英文試題編輯工具—樣板式多媒體試題套用系統〔18〕〔27〕

以樣板方式〔8〕〔27〕來快速編輯試題，試題內容具豐富的多媒體動畫劇情呈現。

- 樣板式多媒體英文試題
以樣板式〔8〕〔27〕多媒體試題套用系統編出來的多媒體試題為研究主題，因其具有樣版性質。
- 多媒體文件腳本描述格式
研究一般 XML based 劇情描述的文件，例如:HTML、SMIL、SVG、EBS 等。
- 情境感知改編協調(Context Aware Adaptation)
情境感知方式來轉換改編多媒體內容文件。本研究將以文獻的方法來探討傳統內容改編協調(Content Adaptation)機制，及演算法設計為主。

- 手持行動裝置
PDA：Pocket PC 2003 為主的應用程式開發。
- 多媒體轉換
多媒體文件轉換、版面協調轉換、格式轉換等等。

1.4 章節概要

在本章提出本論文研究之動機與目標，期許多媒體內容在行動裝置上的呈現能夠盡善盡美，並跨越異質裝置的瀏覽限制。最後簡略地介紹研究方法及研究範圍。

在第二章節裡，將介紹相關的研究議題，包括情境感知改編協調(Context Aware Adaptation)文獻之詳細探討、多媒體內容介紹。

第三章中，會進行現存系統的評估，以及內容改編協調伺服器(Content Adaptation Server)之系統功能規劃，應用層介面功能需求。

第四章中，先介紹系統的整體架構圖以及各個子系統架構。

第五章裡，細節的介紹內容改編協調演算法以及整個系統的實作過程。

第六章，系統操作流程，實例展示。

第七章，未來發展及展望。



二、相關研究、文獻探討

本研究主要探討多媒體內容情境感知的改編協調(Context aware adaptation) [1] [2] [6] [9] 技術，而多媒體內容轉換技術一直在電腦科學領域被研究及推廣，常見的應用是著手於異質行動裝置多媒體內容呈現及設計。而本研究也以多媒體英文試題為例子，嘗試將試題作改編協調到異質行動裝置上可以播放的格式，並求得有好的呈現品質，因此在開發本系統之前，我們會先瞭解一般傳統多媒體內容轉換的文獻。首先會介紹什麼是內容感知為基礎的轉換，其中會提到一般情境感知(Context aware) 的技術架構，以及內容改編協調(Content adaptation) [6] [9] 的種類，並嘗試分析他們的優缺點。

另外本研究的來源多媒體內容以樣板式多媒體試題為例子，所以會簡單介紹一下何謂樣板式多媒體內容，此外要作多媒體內容轉換，當然必須要考慮到多媒體的種類，因此這裡會介紹一些多媒體內容格式，以及相關的多媒體物件轉換工具介紹。

2.1 情境感知改編協調

2.1.1 情境資訊(context information)

情境(Context) [1] [2] 是一可以用來描述任何抽象或具體東西的表徵資訊。這些抽象或具體的東西可以是一個人、一個地方、或是任何的物件，只要這些東西彼此之間都和使用者或是應用程式互相牽引著。

舉例來說以下都可以稱是情境資訊(Context information) [1] [2] [9] :

- Identity
- Spatial information
- Temporal information
- Environment information
- Social situation
- Activity
- Resources
- Schedules and agendas

我們可以舉簡單的例子來說，對同一份多媒體內容，他可能被在不同環境的客戶端透過網路下載觀看，然而它對客戶端使用者卻有不一樣的感受。比如有些客戶端所在的網路環境頻寬品質很差，他必需等待很久的下載存取時間。或是兩個客戶端他們所使用的瀏覽裝置支援程度不同，螢幕大小有顯著差別，有些可能在一

個螢幕畫面可以看到全部的多媒體內容，有些則限於螢幕太小則需藉由其他方式看到全部(例如：縮小全部畫面、藉由 Scrollbar 拖拉)。這些影響多媒體內容品質及使用者觀看操作等因素，都可以稱為是情境資訊。

2.1.2 情境感知(Context aware)

情境感知(Context aware)意指有能力去使用或擷取情境資訊(Context information)。最近在行動異質計算環境下，使用者有可能每天在不定時刻的透過跨平台或跨時間的存取計算服務能力。到最後使用者的注意力可能會被分散到幾個不同的 activities。再者計算能力的載具(Device)越來越小，而導致裝置被使用能力的論點探討。情境感知是一個能夠顯示關於使用者、載具、環境以及那些可以增加使用者在異質行動裝置上互動的資訊觀念。

目前情境感知計算(Context Aware Computing)仍是一個新興而且被眾多人士討論的議題。他主要的意涵在於提供一個運作模式和使用它來設計給開發情境感知系統的工具。對於強調人機介面的系統來說，情境感知系統越來越重要及普及。因為人機介面的系統必須將人的使用經驗、情況(user experience、situation)納入整合到計算機的計算過程中。情況感知(situation awareness)更可被用來減少人為輸入的繁雜動作。藉由分析情境資訊有可能減少額外的計算工作、提供使用者更友善的介面服務。

一般情境感知的協調系統需要透過核心的原件—決策引擎(decision engine) [6] 來進行處理，而大致決策的流程如下圖 1 所示：

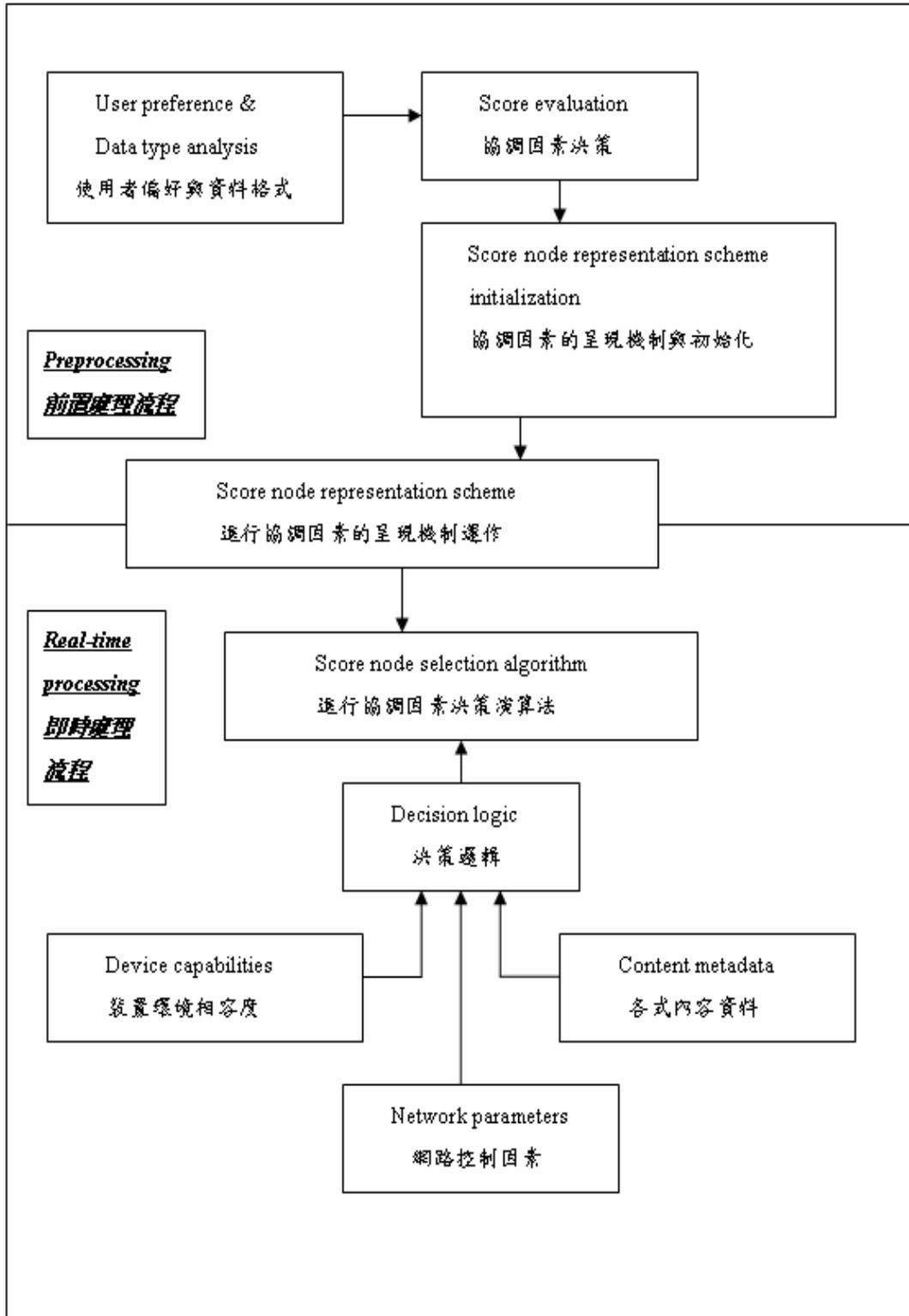


圖 1:一般決策引擎模組

由圖 1 得知一個決策引擎的行為有兩種階段簡述如下：

1. 前置處理流程(這步驟發生在使用者要求之前)所負責的定量品質分析(quantitative analysis of quality)是以品質數值(quality value)來衡量的一個基準，而分數評估(score evaluation)和表徵(representation)則是一個總計分數(aggregate scores)所成的集合當作主要的輸入資訊給決策引擎，然後決策引擎再決定最佳化的輸出版本。

2. 及時處理流程(這步驟處理使用者要求)所負責的決策邏輯(decision logic)和分數節點選定(score node selection)、次序關係(ordered relation)是進行決策引擎的演算法，定義一個分數節點(score node)的意思是指使用者喜好特性的權重分數，用以捕捉所有可能的偏好值(preference values)的組合，而決策邏輯(decision logic)幫助找出對應到一個內容呈現的最好分數節點(scoring node)，而這個節點是依據一些特性參數所決定出的。一般而言，決策此流程是否可獲得正確的協調畫面深受原始內容及所使用的轉譯法影響。

2.1.3 內容改編協調(Content Adaptation)

多媒體內容的轉換包括了內容形式的變化、格式的轉換、多媒體物件的轉換，這些過程為的是能夠符合在目前行動裝置快速發展及無線網路頻寬演變下的異質環境。基於使用目的的不同，內容改編協調(Content Adaptation)的設計會有所不同，有些會以Qos(Quality of Service) [3] 來考慮，有些則以支援程度來作為轉換的依據。而通常大部分的內容改編協調伺服器(Content Adaptation Server)會搭配情境感知(Context awareness)機制來做智慧型轉換。

對於一般多媒體內容的轉換存取方法分成下列兩種類型 [2] [14]：

- 靜態改編協調(Static adaptation)

在編輯內容的時候，內容將經過處理而轉出各式各樣的版本並儲存在特定的資料庫裡。而在呈現的時候，會選擇相對應適合播放的版本。

- 動態改編協調(Dynamic adaptation)

所有的分析轉換動作皆在當客戶端有需求時才適時反應。也就是動態的產生多媒體內容，相對於靜態改編協調來講它不需要額外的儲存空間。

下表 1 是靜態改編協調和動態改編協調的優缺點分析：

表 1：靜態和動態的好壞處

	好處	壞處
retrieve static data	<ol style="list-style-type: none"> 1. 可即時deliver給使用者 2. 展示品質可固定 	<ol style="list-style-type: none"> 1. 對於不同的行動裝置都必須預轉出相對應的版本，因此佔空間
retrieve dynamic data	<ol style="list-style-type: none"> 1. 由於所有多媒體內容將由動態產生所以不會浪費磁碟空間 	<ol style="list-style-type: none"> 1. 線上轉換data將花高代價的CPU time，使用者必須等待轉譯時間 2. 展示流程品質無法固定

本研究另外一個議題是希望結合靜態以及動態的好壞處，來較佳化轉譯時間以及磁碟空間使用量。

而一般作改編協調的架構有下列幾種〔14〕：

- 伺服器為基礎的改編協調(Server-based adaptation)

媒體伺服器(Media Server)負責處理分析所有情境資訊(Context information)，選擇適當的轉換方式，而其轉換方式可以是靜態或是動態。通常有管理介面提供管理者來改編協調，以期許轉出來的結果是合理的。

- 客戶端為基礎的改編協調(Client-based adaptation)

客戶端基礎的轉換是考慮到裝置的支援能力，利用客戶端的轉換而選擇最佳的呈現畫面，而選擇的可能是所有的樣板形式集合。客戶端為基礎的好處是伺服器端不需要知道客戶端的裝置支援程度，壞處是客戶端需要強力的處理能力(Computing power)，不然對於複雜的多媒體內容也許呈現會有延遲狀況。

- 代理伺服器為基礎的改編協調(Proxy-based adaptation)

類似於伺服器為基礎的改編協調，只不過比伺服器為基礎的改編協調較少的控制，也沒有人為預視動作，所以沒辦法保證轉出來的是最佳結果。

2.2 樣板式多媒體內容

多媒體以定義來說是在電腦系統中組合兩種或兩種以上媒體的一種人機互動式訊息交流和傳播媒體。使用的媒體包括文字、圖形、圖像、聲音、動畫和電視圖像〔28〕。

多媒體是超媒體系統(hyper-multimedia)中的一個子集，而超媒體系統是使用超連結(hyper-link)構成的全球信息系統，全球信息系統是網際網路上使用

TCP/IP 協定和 UDP/IP 協定的應用系統。二維的多媒體網頁使用 HTML、XML 等語言編寫，三維的多媒體網頁使用 VRML 等語言編寫。目前許多多媒體作品使用光碟發行，以後將更多地使用網路發行。

目前的多媒體應用層面範疇很多，主要用於廣告、教學或遊戲用途。利用多媒體網頁，廠商可以將廣告變成有聲有色甚或是互動形式來呈現，除了吸引顧客之餘，更能在第一時間提供顧客即時的消費資訊、產品資訊，唯一缺點是如果多媒體內容檔案太大以致於下載時間較長，顧客容易失去耐心，這是多媒體的一大缺憾。而利用多媒體作為教學用途，可以增加自學過程的互動性，更可以吸引學生學習、提及學習興趣，以及利用視覺、聽覺及觸覺三方面的回饋(Feedback)來增強學生對知識的吸收。

樣板式多媒體內容〔8〕〔27〕的定義是具有腳本的多媒體內容，其內容組成可由多種多媒體物件組成。而基於他們的設計用途以及有些同類型的多媒體內容編排方式的固定性，在設計內容時可以用樣板形式來設計。可以節省除了設計語意內容額外要考慮的版面設計代價。

而本研究探討的多媒體英文試題是本實驗室開發的多媒體內容，就有如上面的特性，將試題分成三個段落-“Stimulus”、“Question”、“Answer”，並強調試題的互動性質。然而這樣的英文試題要直接呈現在各式各樣的行動裝置上播放，可能造成對使用者的另一項觀看負擔(起因於畫面不協調或是瀏覽器需作額外的協調動作)，因此本研究主要針對樣板形式的多媒體內容作協調分析，並提出解決方案。



2.3 多媒體文件格式之腳本語言

2.3.1 HTML

HTML〔15〕是lingua franca用來公布在World Wide Web的超文件(Hypertext)。他是一個基於SGML(Standard Generalized Markup Language)的非專利格式，可以用任何工具去編輯創造這樣的文件。HTML使用標籤像是<h></h>來結構化文章的段落、標頭、排列、超連結等等。詳細的HTML的文法定義於W3C組織文件下。而大部分瀏覽器都支援JavaScript，他是一種基於對象的腳本語言(script language)，原名是LiveScript，目前已經在WWW上廣泛用於動態web頁面的編成語言。而一般在桌上型電腦上HTML瀏覽器包括了微軟的Internet Explorer〔30〕、Mozilla Firefox〔29〕、Netscape〔31〕等等，Pocket PC的HTML瀏覽器為Internet Explorer。

2.3.2 SMIL

原名為Synchronized Multimedia Integration Language，SMIL [17] 能夠將一組獨立的多媒體對象合成為同步的多媒體演出。SMIL也是XML Based的描述語言。他比單純的HTML更能闡述劇情的時序性。大致可以列出他的特徵如下：

1. 可以創建 slide-show 的呈現，如同 power-point 一樣
2. 支援多媒體物件的播放(text、video、audio、...)
3. 支援同時呈現多個多媒體檔案播放
4. 支援播放來自多個 web server 的多媒體檔案
5. 包含超連結、暫停、開始、下一步的按鈕
6. 可以定義連續性的元素播放或定義播放時間限制
7. 可以定義元素的位置、及隱藏能力。

目前標準 SMIL2.0 事由 W3C 在 2001 年八月制訂而成。一個標準的 SMIL 檔案包含下列：

- 呈現的版型(layout)
- 呈現的時序性
- 多媒體元素的來源

PC上的播放器很多，而PDA上的播放器有Pocketsmil2.0 [32]，支援的多媒體有文字、聲音(WAV,MP3),圖片(BMP, JPG, PNG),影片(MPEG)。並支援使用者互動及超鏈節，唯一的缺點是文字處理能力較差及排版較差。目前手機上的多媒體簡訊所用的描述語言為SMIL的subset。

2.3.3 SVG

原名是Scalable vector Graphics [16] 可縮放向量圖形，是基於可擴展標記語言(XML)，用於描述二維向量圖形的一種圖形格式。SVG由W3制定，是一個開放標準。SVG嚴格遵從XML語法，並用Text-based的描述性語言來描述圖像內容，因此是一種和圖像解析度無關的向量圖形格式。SVG圖格式具有以下優點：

1. 圖像文件可讀，易於修改編輯。

2. 與現有技術可以互動融合。例如 SVG 技術本身的動態部分(包括時序控制和動畫)就是基於 SMIL 標準
3. SVG 圖形格式可以方便的建立文字索引，從而實現基於內容的圖像搜尋。
4. SVG 圖形格式可以用來動態生成圖形。

SVG 面臨的主要問題一個是如何和已經佔有重要市場份額的向量圖形格式 Flash 如何競爭的問題，另一個問題就是 SVG 的本地運行環境的下廠家支持程度。而除了標準的 SVG 之外，又提供了兩個 SVG 子集-SVG Basic、SVG Tiny，主要的目標是為手持行動裝置等設備提供向量圖形顯示格式。S 主要支持以下幾種顯示對象：

1. 向量顯示對象，基本向量顯示對象包括矩形、圓、橢圓、多邊形、直線、任意曲線等
2. 嵌入式外部圖像，包括PNG、JPEG、SVG等
3. 文字對象

而目前正在研究制定的為SVG1.2 更能提供多媒體的支援，如：audio、video 等等。而最常用的SVG播放器來自Adobe的Adobe SVG Viewer。行動裝置則有java 的TinyLine及PDA的eSVG viewer [33] 。

2.3.4 EBS (編輯手劇情描述語言)

是由本實驗室開發的一個多媒體劇情描述語言 [18] 。他主要目的在於提供由實驗室開發的一套多媒體編輯軟體-編輯手 [18] 進行編輯模式時，儲存各個多媒體元素的呈現方式，及動畫過程。而編輯手也將利用此描述格式轉譯成其他標準的多媒體文件格式的描述語言，如上面介紹的HTML、SVG、SMIL、或是MP4、Flash等壓縮格式。而本實驗室也開發一套針對樣板式的多媒體試題編輯工具，他的內部劇情描述語言亦是採用EBS格式。而本描述語言並非基於標準XML為基礎格式，原因在於他使用的目的不是給終端的觀看者用，而是給編輯手或是命題手的程式來使用，進而加快程式的效率。

2.4 多媒體轉換相關工具

2.4.1 XSLT

XSLT [24] 全名為 Extensible Style-sheet Language Transformations 的簡稱，他是一種對 XML 文件作轉換動作的語言。他是 XSL (Extensible style-sheet language) 規範的一部份。XSL 規範的另外一部份是 XSLF (F 代表格式化對象 Formatting Objects)。

XSLT 是把 XML 文件轉換成另一種 XML 文件的 XML 轉換語言。即原文檔的所有數據或者部分數據生成另外的 XML 文件。在這個轉換的過程當中涉及到下面幾項：

- 加上一些像 HTML 的固定標籤
- 移動本文
- 對本文排序

被轉換的原來 XML 文件本身具有樹狀結構。XSLT 語言是聲明性的語言，即 XSLT 文件本身只是包含了一些轉換規則的文檔。而這些規則可以被應用到轉換過程中。XSLT 本身也是一份 XML 文件檔，所以它也必須遵守嚴格的 XML 規範。

XSLT 處理器會首先確定使用 XSLT 中的哪些規則，然後根據優先權作相對應的轉換動作。下面是一個簡單的 XSLT 文件範例架構：

```
<?xml version="1.0" ?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
...
</xsl:stylesheet>
```

目前處理 XSLT 的處理器 (XSLT Processor) 有 Xalan C++ / Java 版本。

2.4.2 Lame

Lame [21] 是一個 Open Source 的 MP3 音頻壓縮軟體。「LAME」這個字眼是個英文縮寫，原本來自 LAME Ain't an MP3 Encoder。它自 1998 年以來由一個 Open Source 組織開發，現在已經成為一個在許多方面都達到最高品質的 MP3 編碼器。他包含了下面幾項特點：

- 高優化度的預置模式
- MPEG1,2 和 2.5 layer 3 的編碼
- 編碼引擎可以被編譯成共享 Library 或是 DLL
- 高速編碼

- 支持 CBR ABR 和 VBR 編碼方式
- 支援 Exact Audio Copy 及 CDex

2.4.3 DevIL

DevIL [22] 也是開放原始碼(Open Source)的影像處理軟體，全名為 Developer's Image Library。主要提供給程式開發員一個簡便的Library，此外它具有強大的圖形處理能力以及讓程式開發員容易使用。DevIL主要提供下列對於影像的處理能力：

- 載入圖檔
- 儲存圖檔
- 轉換圖檔
- 影像過濾
- 其他的一般影像處理

目前 DevIL 可以載入.bmp .cut .doom .gif .ico .jpg .lbm .mdl .mng .pal .pbm .pcd .pcx .pgm .pic .png .ppm .psd .psp .raw .sgi .tga .tif .hdr 等。而可以儲存的格式包括.bmp .dds .h .jpg .pal .pbm .pcx .hdr .pgm .png .ppm .raw .sgi .tga .tif。

此外 DevIL 提供下列的 API 以供呈現: OpenGL , Windows GDI , SDL , DirectX 和 Allegro。而且開放的原始碼可以用 DevIL, Djgpp, MSCV++ , Linux gcc, Delphi, Visual Basic , Power Basic 和 Dev-C++ 等編譯器重新編譯。

2.4.4 MS Window Media Encoding Script

它是微軟公司 [30] 出的一個影片編碼命令列程式(Command-line)，利用他可以簡便的提供下面多種功能：

- 轉換 .wma, .wmv, .asf, .wav, .mpg, .mp3, .bmp, .jpg 的多媒體影像檔。
- 捕捉從裝置上的即時影像
- 作多媒體影像廣播用途
- 另外可將一些影片或聲音作優化，例如改變影片大小品質、音量音質調整

三、系統功能需求分析

本系統的主要目的是希望能將存放在目前原來未改編協調的內容伺服器 (Server for Content without adaptation) 的多媒體內容透過內容改編協調伺服器 (Adaptation Server) 轉換協調，並將轉換結果上傳至另外一個獨立的內容平台，我們稱此平台為已改編協調的內容伺服器 (Server for adapted content)，此平台將存放著轉換協調完成的各式各樣多媒體內容，並且能依照使用者設定或有用的情境資訊 (Context information) 作排放及紀錄。另外異質行動裝置可以透過無線網路以三種行為模式-靜態 (static)、動態 (dynamic)、動態加靜態混合方式 (static + dynamic) 從中擷取特定可以播放的多媒體內容。以靜態來說，對於那些已經轉換協調好的多媒體內容，是透過管理協調者的預覽轉換而後上傳到已改編協調的內容伺服器。因此本系統大致可以分成四個功能架構去探討：

- 內容伺服器系統功能評估
- 改編協調伺服器功能分析
- 管理協調者 web 介面功能需求
- 行動裝置應用程式功能需求

以下會針對這四個功能詳細介紹。

3.1 內容伺服器系統評估

內容伺服器 (Content Server) 分成兩種，一為是未改編協調的內容伺服器，本內容伺服器由實驗室學長姐開發而成，目前有多達千個的數位多媒體內容。另一是可以用來放置各式各樣多媒體內容的伺服器，我們稱為已改編協調的內容伺服器。而他們兩個伺服器的詳細架設環境規格如下表 2：

表 2：兩種內容伺服器架設環境 (content server)

未改編協調的內容伺服器	
作業系統	FreeBSD 5.0 / apache2.0
資料庫	MySQL
程式開發語言	php
已改編協調的內容伺服器	
作業系統	Win2000 advanced server / Tomcat5.0
資料庫	MsSQL
程式開發語言	Java 、JSP

未改編協調的內容伺服器的資料庫存放利由實驗室開發的一套英文試題編

輯軟體編輯而成的樣板式多媒體試題內容(包含了描述檔、資源檔)。我們可以利用 HTTP 通訊協定來存取下載描述檔及其他的多媒體資源檔,例如:聲音、圖片、影片。基本上本伺服器資料庫比較不需要去作擴充維護,其主要是因為本伺服器是用來資料庫查詢下載資源用途。

對於已改編協調的內容伺服器的用途來說,他是儲存異質性的多媒體內容,這些內容是經由未改編協調的內容伺服器裡的多媒體內容轉換過去的。本伺服器必須維護一份資料庫來記錄多媒體試題,以供後來協調完後上傳更新,或是下載查詢的動作。更新的動作有刪除、新增、修改等等。查詢的動作以過濾篩選查詢為主,篩選項目包括腳本格式、圖片格式、字形大小、影片格式、螢幕大小,其主要目的用來提供客戶端行動裝置的多媒體支援程度及多媒體品質的篩選能力。

3.2 改編協調伺服器功能分析

我們可以嘗試去瞭解改編協調伺服器(Adaptation Server)的運作過程,進而去分析他需要哪些功能。從下面的運作流程來看,內容改編協調伺服器首先必須要先擷取必備的資源檔,再來接受來自應用端的參數值,第三步會進行分析文件的工作,第四步會搭配參數的設定以進行改編協調的能力,最後將轉換協調後的結果發佈至已改編協調的內容伺服器。



圖 2 : adaptation server 運作過程

底下我們介紹這幾個功能的細節部分：

3.2.1 擷取必備資源檔

改編協調伺服器(Adaptation Server)和未改編協調的內容伺服器是架構於不同平台的兩個伺服器,因此改編協調伺服器必須要跟內容伺服器有個通訊協定,進而才能擷取儲存在原來試題平台資料庫上的多媒體描述檔,因此我們必須先下載必要的多媒體文件描述檔以進行分析,進而下載相對應的其他多媒體物件,而內容伺服器的溝通介面是透過 HTTP 通訊協定及 MySQL 資料庫查詢來完成。

3.2.2 擷取必備參數值

改編協調伺服器(Adaptation Server)必須能夠接受來自應用端的參數,應

用端可能為管理協調者的管理程式或是客戶端的行動裝置播放器程式，這些參數即是情境感知(Context Aware)的部分情境資訊(Context Information)，對於在我們協調版型具有重大的意義。本論文研究主要著重於樣板式多媒體內容在行動裝置上的版型協調呈現，因此參數包括了下列幾種，這些參數是作協調及轉換的重要資訊：

1. 裝置螢幕大小：格式包含了 160x180、240x240、240x320、320x240、320x320、320x480、480x320、480x480 的解析度，目前大部分的 Pocket PC 為 240x320。這裡考慮到分析螢幕大小的原因是基於未來要重新調整多媒體內容-包括了 Layout 重新安排，及其他圖片的縮放等等。
2. 字形大小：英文字形主要以 Arial 為主，大小則 9px~12px 為主，我們必須考慮到重新設定英文字形的大小。
3. 圖片格式：較常見的一般圖檔格式有 BMP、JPG、GIF、PNG，這裡設置圖片格式參數是要考量到行動裝置裡的應用程式支援能力，之後會作相對應格式的轉換。
4. 輸出描述檔案格式：以一般的多媒體腳本描述語言為輸出格式，較常見的有 HTML、SVG、SMIL、XMG(本論文自訂的 XML Based 格式)

3.2.3 解譯分析文件

下載回來的描述格式為加密的 EBS 檔，此加密是為了保護內容之考量，因此必需要經過一道解密手續才能進行描述檔的分析動作。當然在設計解譯程式之前，要先瞭解原本的 EBS 格式描述語意文法。由於這個步驟可以知道此腳本內容是否有其他多媒體物件來組成（例如：聲音、影片、圖片等等），因此在此步驟必須能夠同時下載其他的多媒體物件。此步驟歸納的功能如下：

1. 解密描述檔案格式
2. 分析描述檔案內容裡的多媒體資訊
3. 下載其他多媒體物件

3.2.4 改編協調

當我們解譯分析完描述檔，所有的多媒體描述資訊都已經儲存在一個資料結構中，我們必須能依照參數的設定，進行這些數據的分析並設計協調演算法，協調演算法的目的是提供最佳協調後版型的呈現結果。

系統必須能提供多媒體物件轉換及描述檔格式的轉換功能，多媒體物件包括了 Image、Audio、Video。對於 Image 和 Video 要能夠提供格式的轉換、縮放的轉換。Audio 要能提供格式的轉換。而描述檔格式的轉換，應該按照參數的

給定轉換成相對應的格式，例如 HTML、SVG、SMIL 等等。這裡的部分功能總結如下：

1. 分析描述檔案內容
2. 協調演算法
3. 多媒體物件轉換

3.3 Web-based 協調介面功能需求

這裡的設計介面是以試題協調管理者來考量，除了帳號登入畫面外，提出了下面幾個基本的功能，一樣是按照運作過程來分析：



圖 3：web-based 介面協調管理過程

3.3.1 挑選欲轉換的多媒體內容清單

系統要能夠從未改編協調的內容伺服器(Server for Content without adaptation)擷取多媒體內容目錄清單，而且能提供篩選多媒體內容的介面，例如以本研究以樣板式多媒體英文試題為例，便可以篩選試題等級難易度(入門級、基礎級、初級、中級、高級)、試題形式(聽、說、讀、寫)等等，使用者可以從篩選的試題列表當中挑選試題以編列到試卷列表裡。

3.3.2 設置必備參數值

在上面使用者編完一組待轉換多媒體內容清單後，這裡設計的是個 Web 頁面以提供協調者能夠從中選擇他欲轉換出去的各项參數，參數內容如 3.2.2 小節提到的，我們簡述如下：

1. 裝置螢幕大小：格式包含了 180x160、240x240、240x320、320x240、320x320、320x480、480x320、480x480 的解析度
2. 字形大小：英文字形主要以 Arial 為主，大小則 9px~12px 為主。
3. 圖片格式：較常見的一般圖檔格式有 BMP、JPG、GIF、PNG
4. 輸出腳本格式：HTML、SVG、SMIL、XMG(本論文自訂的 XML Based 格式)

3.3.3 批次分析各個內容

由於管理者不可能一次單點選一個多媒體內容作改編協調，因此這裡的設計模式是希望藉由批次(Batch)的方式來一次轉換協調完成。例如本研究以一份多媒體英文試卷有相當多的多媒體英文試題，使用者如果一題一題做改編協調的動作將相當繁瑣，所以與其這些反覆的動作，不如讓程式自動批次轉換所有試卷裡的題目，並提供協調後的預覽結果。簡略一下這裡應該提供給管理協調者的功能如下：

1. 列出第一步驟選取的多媒體內容清單
2. 使用者進行勾選單題作協調分析或是全選作批次協調分析
3. 系統應能顯示分析進度
4. 協調後預覽呈現



圖 4：web-based 介面批次協調過程

3.3.4 手動篩選

批次自動協調完的動作完成後，這裡額外給予管理協調者其他版型選擇考量，每一多媒體內容除了自動選取的最佳結果外，並提供其他所有版型的參考畫面，以達到管理協調者想要的轉換協調結果。

3.3.5 發佈內容

這裡事實上的作用為觸發改編協調伺服器(Adaptation Server)實際去作批次轉換的動作，而轉換的依據為剛剛手動選取後的最終結果樣式，並在轉換完畢後壓縮成一份壓縮檔，透過改編協調伺服器上傳至已改編協調的內容伺服器(Server for adapted content)，最後能夠在畫面顯示上傳完成的頁面，或是提供一些額外的資訊給使用者知道(例如：檔案大小)。

3.4 行動裝置應用程式開發功能需求

這裡的設計功能是針對給行動裝置使用者來操作使用，改編協調伺服器 (Adaptation Server) 可以轉出的內容包括了 HTML、SMIL、SVG、XMG (本研究自己定義)，至於我們不用現成的播放器來撥放 HTML、SMIL、SVG 而要自己設計播放器是因為現存的播放器都有部分功能的缺憾，此外自己設計的播放器得以驗證改編協調伺服器轉換出來的多媒體內容樣式結果是否合理，並且可以跟改編協調伺服器作溝通，以利將來模擬三種模式的存取方法-靜態、動態、靜態及動態的混合方式。

我們一樣可以分析行動裝置上的應用程式運作過程來勾勒出他的功能面

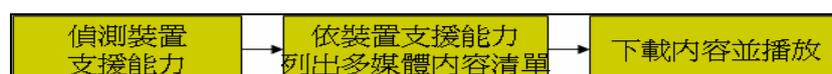


圖 5：行動裝置上應用程式操作過程

3.4.1 偵測裝置支援之能力

本應用程式能夠主動自動偵測裝置上的螢幕設定大小、字形設定大小，並將這些及本應用程式可以播放的多媒體格式當作參數，透過 HTTP 通訊協定傳遞給改編協調伺服器 (Adaptation Server) 知道以進行動態轉換，或者是直接傳給已改編協調的內容伺服器 (Server for adapted content) 去篩選多媒體內容。

3.4.2 下載內容之能力

應用程式應能依照參數顯示存放在已改編協調的內容伺服器 (Server for adapted content) 的多媒體內容相對應支援的內容列表。並依照使用者點選的某份試卷進行下載的動作，如果點選的試卷不存在需要由動態產生的話，這時就得跟改編協調伺服器有溝通的請求，並等待轉譯完成而後再次下載。而我們下載回來是個壓縮格式，因此要能自動解壓縮，使用者不需要再點選解壓縮程式造成額外的操作負擔。

1. 依裝置支援能力列出多媒體內容清單
2. 下載選定的多媒體內容並顯示下載進度
3. 自動解壓縮多媒體內容

3.4.3 多媒體內容劇情播放能力

下載完解壓縮一份多媒體內容，及應顯示可以播放內容的訊息，使用者可以按播放按鈕進行內容播放，按下一題或上一題以播放相對應的內容，此外更可以從試卷裡的題目列表挑選特定試題來播放。而用來描述多媒體內容的描述檔案語言為本研究自訂的 XML Based 的格式-XMG，所以本播放器要能夠解析 XMG 格式。

四、系統架構

4.1 系統設計說明

4.1.1 系統架構圖

本系統大致架構圖如圖 6 所示：

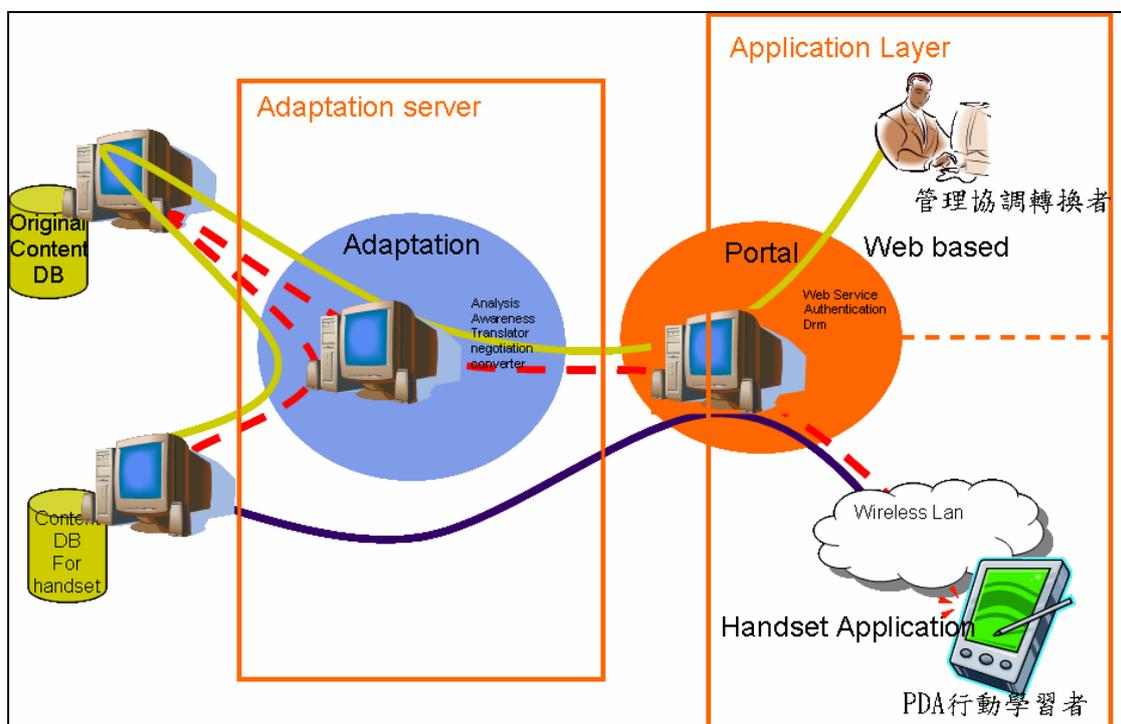


圖 6:系統架構圖

主要可分成三個子系統去做開發：

- 改編協調後的內容伺服器(Content server)資料庫系統之維護
- 改編協調伺服器(Adaptation server)之開發
- 應用端程式(Application Layer)開發

從架構圖可以清楚的看出他們的關係，Application Layer 的應用程式會透過入口伺服器(Portal server)和改編協調伺服器(Adaptation server)去對現存系統資料庫裡多媒體內容做改編協調。Application Layer 當然可以直接存取現存的資料庫系統。而詳細的子系統架構說明將於下節說明。

4.2 子系統架構說明

4.2.1 內容伺服器資料庫系統之維護

在 3.1 節的分析下，我們可以發現對於未改編協調的內容伺服器(Server for Content without adaptation)可以不需要任何更改動作，因為我們僅需做查詢下載內容動作而已。而對已改編協調的內容伺服器(Server for adapted content)是我們必須要更改維護的地方，因為已改編協調的內容伺服器需要提供平台以供由改編協調伺服器(Adaptation server)改編協調完成的多媒體內容上傳用，而且也需要被行動裝置查詢下載使用，因此資料庫的定義是必要的。已改編協調的內容伺服器的後端應用程式架構圖如下：

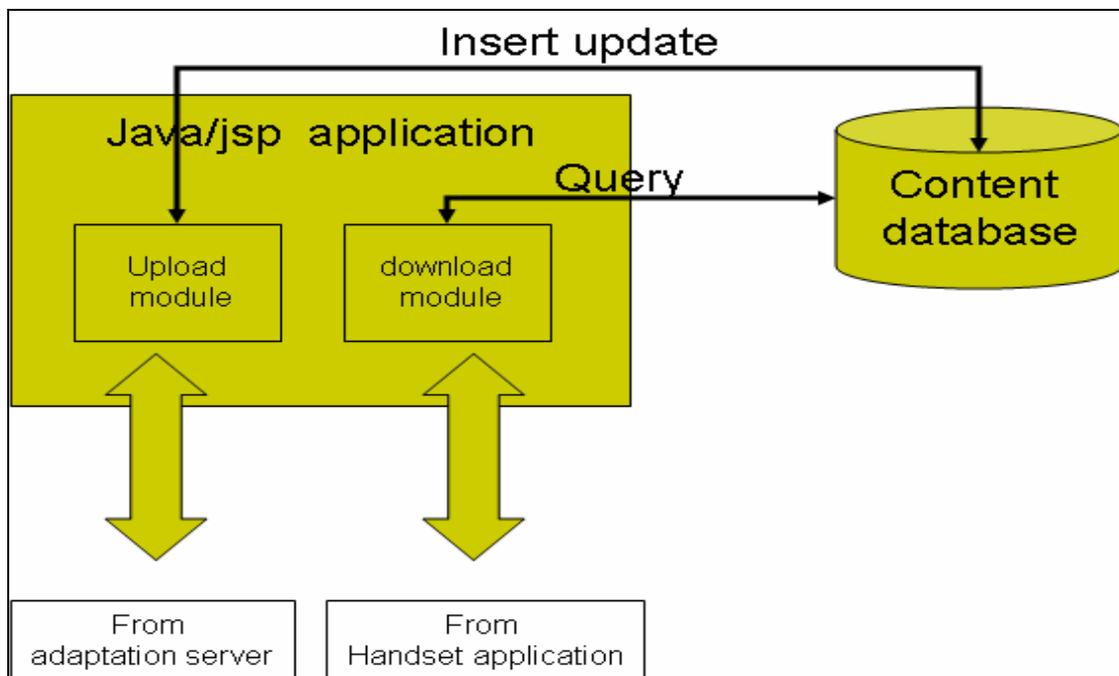


圖 7: 已改編協調的內容伺服器後端應用程式架構運作模式

4.2.2 改編協調伺服器之開發

本設計將強調三個主情境感知資訊(Context Aware Information)來幫改編協調伺服器(Adaptation Server)裡的主要轉譯協調模組提供必備的資訊：

- 內容版型資訊(Content layout information)

這裡會研究探討 Layout 的影響因素如下：

1. 字形大小
2. 文字數目
3. 文具段落
4. 圖片大小
5. 影片大小
6. 以及各個多媒體物件位置座標
7. 裝置螢幕大小

為了使用者瀏覽方便，我們必須將 Layout 作協調，我們底下以樣板式多媒體英文試題為例：

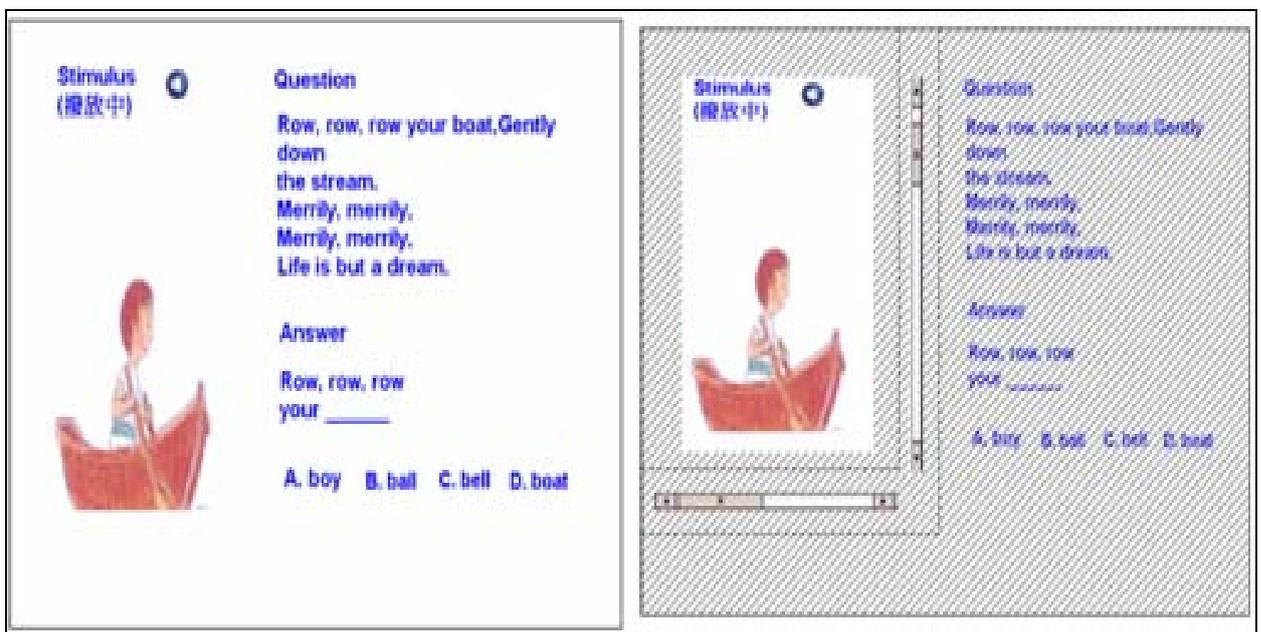


圖 8：多媒體英文試題在 PC 及 PDA 上的畫面

圖 8 的左邊是一個在 PC 環境瀏覽的樣板式多媒體英文試題，如果透過 PDA 原封不動的下載此內容，呈現結果將如右邊所示。可以看到陰影部分是使用者目前沒辦法看到的部分，陰影部分必須藉由 PDA 上的 Scrollbar 來拖曳才能使內容呈現在 PDA 螢幕上。但是如果我們將版型經過協調後，呈現的模式可以如下圖 9 所示，我們可以看到原本一個頁面的內容被我們分成三個頁面來顯示，而每個頁面是由這份多媒體文件裡的相關多媒體物件來組成，也就是根據內容語意 (Content Semantics) 來劃分。如本例來說是分成三個段落-“Stimulus”、 “Question”、 “Answer”。而各個頁面是以超鏈結 (hyper-link) 來作切換。

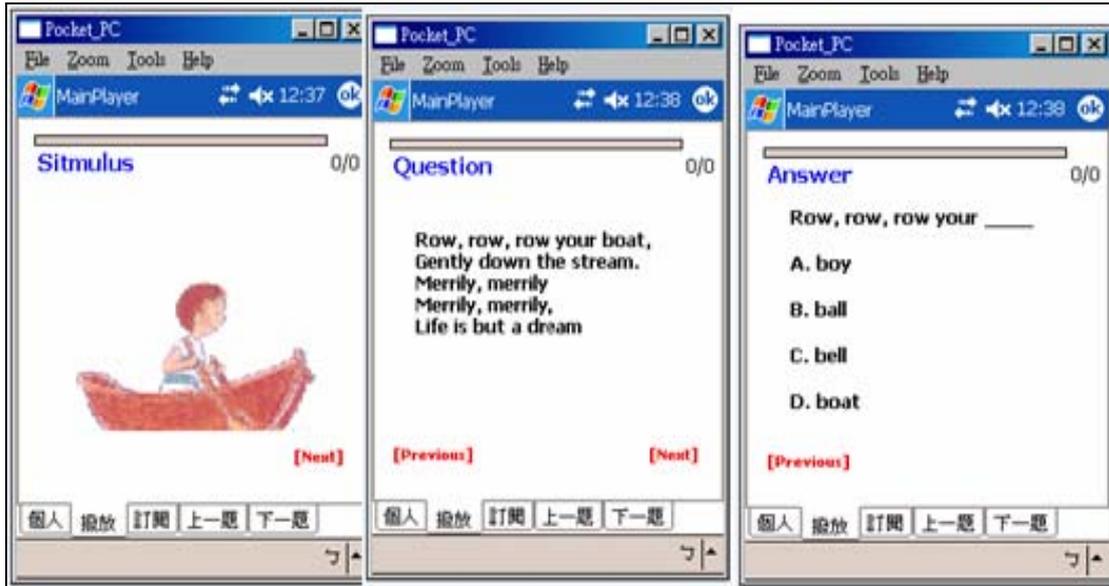


圖 9：多媒體英文試題協調後在 PDA 上的畫面

- 裝置能力資訊(Device capability information)

依據裝置支援程度，多媒體物件必須轉換，轉換內容包括下列幾種：

1. 圖片大小縮放、格式轉換
2. 聲音格式轉換
3. 影片大小縮放、格式轉換
4. 腳本劇情描述格式轉換

- 空間和時間資訊(Spatial and Temporal information)

這裡會考慮到時間跟空間的衡量，而延伸出來不同轉換方式-靜態(static)和動態(dynamic)轉換。我們可以先分析一個例子的數據如下表 3：

表 3:各種多媒體轉換所花費時間

	原來規格	輸出規格	所花時間
板型轉換	xml	xml	小於1秒
圖片轉換	10KBBMP	5KB JPG	小於1秒
聲音轉換	1.76MB wav	164KB Mp3	約3~4秒
影片轉換	320x240 1.29MB WMV	176X144 1.1MB WMV	約12秒

從上表可以看出花最多的時間在於聲音及影片的轉換，因此我們可以將轉換時間

比較長的多媒體物件(Sound、Video)以靜態方式事先轉換完成，而其他多媒體物件(Image、Text、Layout)則以動態方式轉換完成。這樣一來可以降低傳統用動態轉換過程時間，並可減少一般傳統用靜態方式的空間成本。這是本研究將提出的靜態加動態的混和轉換存取方式。

改編協調伺服器(Adaptation Server)主系統詳細架構可以如下圖 10 所示，主要由三個子模組所構成：

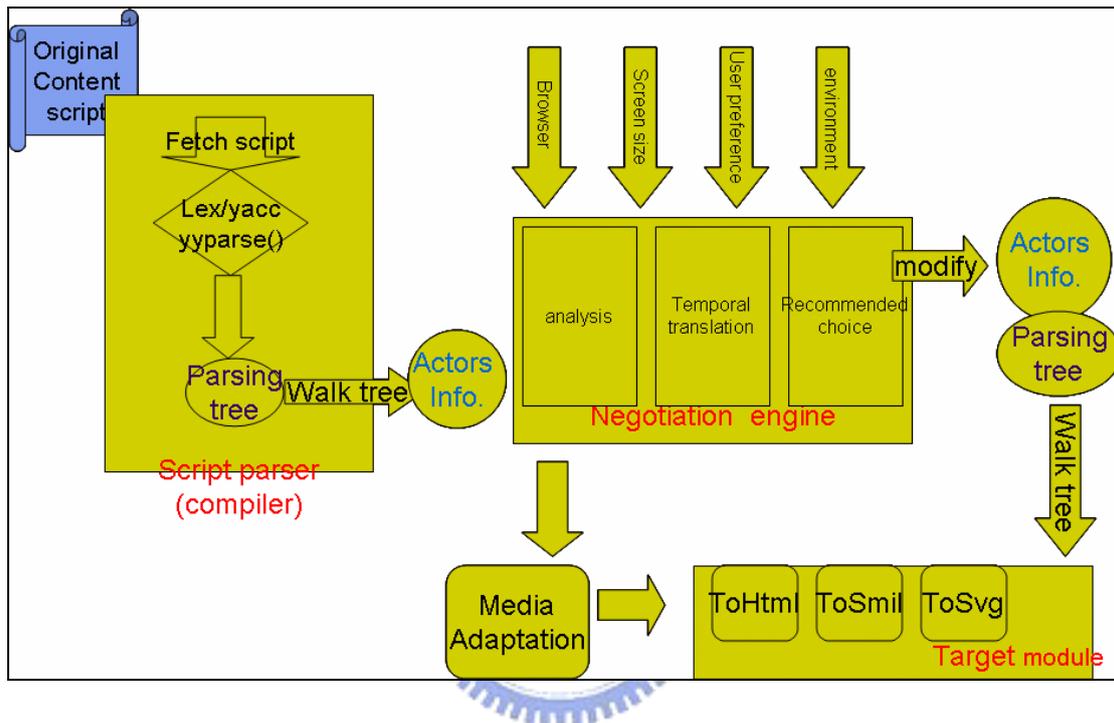


圖 10: Adaptation server 詳細內部架構圖

- 描述檔剖析器 (Script parser):

此模組解譯來自未改編協調的內容伺服器 (Server for Content without adaptation) 資料庫裡的多媒體內容描述檔，描述檔由本實驗室開發的一套多媒體編輯工具產生。解譯完的資訊我們會把他存在一個資料結構裡，以提供轉換引擎 (Conversion engine) 作分析。

- 協調引擎 (Negotiation engine):

此模組主要的目的是接受來自描述檔剖析器 (Script parser) 的劇情描述資訊，並接受由應用端 (Application Layer) 傳來的各項參數，進行分析協調的動作。過程會先經由分析 (Analysis)、暫時轉譯 (Temporal translation)、推薦選擇 (Recommended choice)，然後將協調後的資訊寫回劇情描述資料結構裡。過程中會依照參數的指定將多媒體物件

(Audio、Video、Image)轉換成相對的格式或改變大小。

- 目標模組(Target module):

最後我們要將已經協調完的劇情資訊寫回相對應的多媒體內容呈現描述檔格式，這裡的腳本描述格式以 XML Based 的為主。

而為了滿足應用端程式的需求，改編協調伺服器(Adaptation Server)可以額外做包裝的工作。以多媒體英文試題試卷來說，還必須提供將試卷壓縮後，並上傳至已改編協調的內容伺服器(Server for adapted content)的功能，當然這裡必須搭配應用端的互動協定。

4.2.3 應用端程式開發

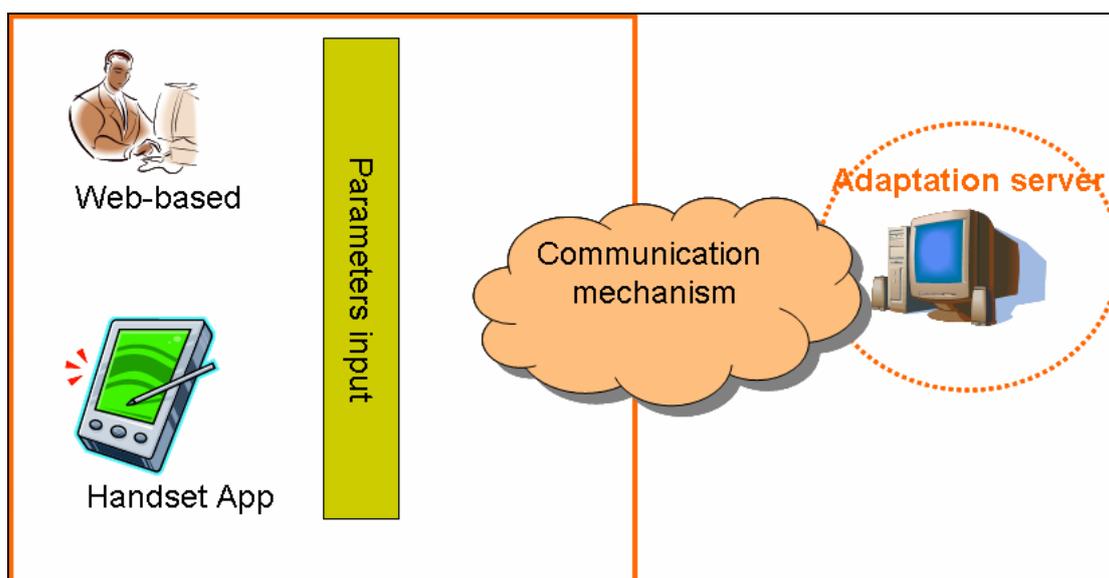


圖 11: 應用端程式粗略架構圖

不管 Web-based 管理改編協調介面或行動裝置應用瀏覽程式(Handset application) 開發，重要的是他與改編協調伺服器(Adaptation Server)之溝通的訊息傳遞、參數輸入。我們以下面兩個應用運作圖形來解釋他們的運作過程：

- Web-based 管理協調者運作圖



圖 12：管理協調者運作圖

管理者第一步驟必須輸入相對應的參數以供改編協調伺服器參考用，參數包含了裝置的螢幕大小、文字大小、圖片格式及描述檔輸出格式。皆下來第二步等待改編協調伺服器的協調分析，而分析完後將推薦樣式結果呈現給管理者知道，管理者可以在第三步驟選取其他可能的樣式呈現結果，否則預設會以系統在第二步驟分析的推薦樣式結果為最後選定結果。在第四步驟管理者就必須等待改編協調伺服器依據最後選定的結果作最後的轉換並上傳到試題平台，此平台也就是已改編協調的內容伺服器(Server for adapted content)。

● PDA application 三種模式瀏覽端運作圖

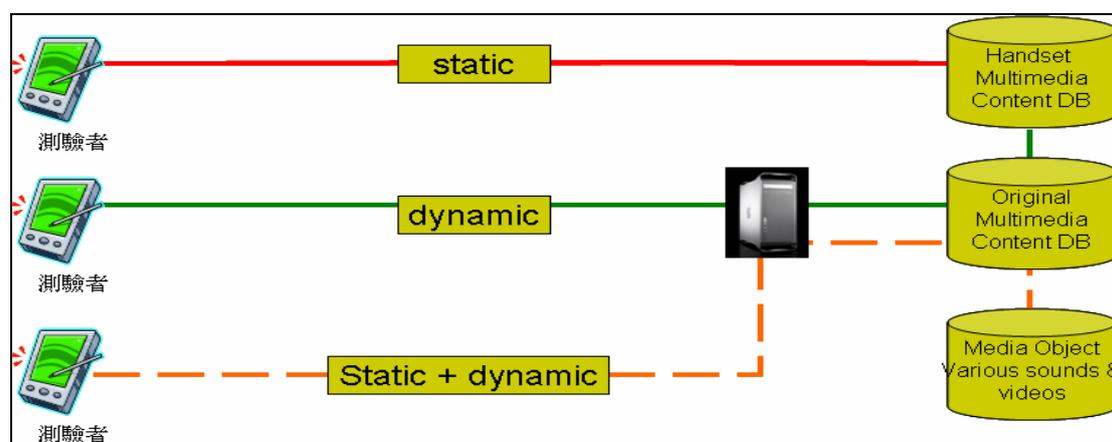


圖 13：PDA application 三種模式瀏覽端運作圖

對於客戶端使用者來說(如圖 13 的測驗者)，他有三種行為操作模式-靜態、動態、靜態及動態混合，這樣的三種模式設計是為了分析他們存取時間差異及磁碟空間用量。以靜態來說，PDA 上的應用程式直接會去資料庫擷取已經轉譯好的多媒體內容，這些內容也就是存在由上面「管理協調者」已事先將試題上傳至已改編協調的內容伺服器(Server for adapted content)的資料庫裡。而動態的方式是多媒體內容全由改編協調伺服器(Adaptation Server)來自動產生。靜態及動態混合方法則是利用將事先已經轉好的多媒體物件格式(例如：MP3,WAV,AVI)再搭配動態的分析協調版型及動態產生描述檔案，之後再經由包裝傳達給客戶端瀏覽。

五、系統設計與實作

5.1 已改編協調的內容伺服器之維護實作

5.1.1 試卷資料庫規格

欄位	型態	屬性	Null	預設值
<u>paper_id</u>	int(5)		否	0
tcoolnum	int(5)		否	0
screensize	varchar(200)		否	
format	varchar(200)		否	
img_format	varchar(200)		是	NULL
snd_format	varchar(200)		是	NULL
veo_format	varchar(200)		是	NULL
font_size	int(3)		否	0
lastModified	datetime		否	0000-00-00 00:00:00

表 4：多媒體試題試卷資料庫資料表

上表為多媒體試題試卷資料庫資料表，此資料表用來紀錄每一份多媒體的詳細規格，目的在於提供給應用端程式查詢或更新用途。詳細欄位內容說明如下：

Paper_id：為主鍵，紀錄每份試卷的 identity

Tcoolnum：紀錄這份試卷裡有多少多媒體英文試題

Screensize：是目標裝置的 screensize 大小

format：是多媒體內容腳本語言格式。

Img_format：是圖檔的格式

Snd_format：是音檔的格式

Veoforamt：是影片檔的格式

Font_size：是字形的大小

LastModified：是最後更新日期

5.1.2 後端程式之開發

從 4.2.1 小節的架構圖可以看出後端程式可以分成兩個模組：

- 上傳模組(Upload module)：接收改編協調伺服器(Adaptation Server)將試題試卷打包好的上傳資料內容。而資料內容除了多媒體內容的壓縮檔案外，還包含了上述資料表的重要參數。上傳完畢，將更新資料庫的資料表。由於改編協調伺服器上傳方式採用 Http Post 方式，因此這邊的設計可以搭配 JSP 來撰寫，如下面四個步驟：
 1. Parse HTTP request with Query String
 2. Write uploading File to the Specified Disk
 3. Update Database
 4. Response HTTP request with handled status
- 下載模組(Download module)：接收來自行動裝置裡的多媒體英文試題播放器發出的靜態請求，以進行資料庫查詢動作，請求內容分成試卷列表請求、和試卷檔案請求：
 - ◆ 試卷目錄請求
 1. Parse HTTP request with Query String
 2. Query Database
 3. Response HTTP request with the content's entry lists
 - ◆ 試卷檔案請求
 1. Parse HTTP request with Query String
 2. Query Database
 3. Write the specified file's content as a HTTP response

5.2 改編協調伺服器之設計與實作

本節將針對第四章提到的改編協調伺服器(Adaptation Server)架構的三個子模組去設計及實作，包含描述檔解析器(Script parser)、轉換引擎(Conversion engine)、目標模組(Target module)，我們將這些內容詳述於下面幾小節。

5.2.1 描述檔剖析器製作

描述檔剖析器(Script parser)主要功能是擷取描述檔內容資訊，在製作剖析器之前我們要先瞭解原始多媒體描述格式結構。下面是一個典型的文字多媒體描述方式：

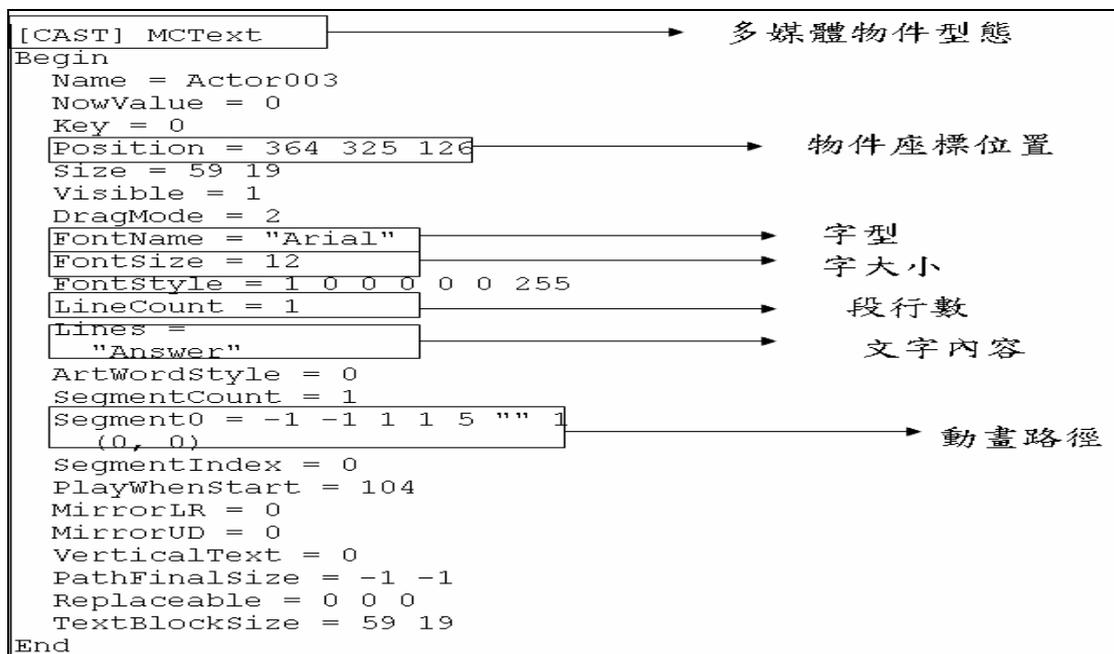


圖 14: EBS 描述檔範例圖

我們可以由上面的例子看到文字多媒體的所有屬性，包括了有多媒體物件的座標、文字的字形、文字的大小、以及斷行數另外還有動畫的路徑呈現。

在這裡我們會運用到複雜的編譯器設計的過程(compiler process)及技巧，不過如果只是單純的要擷取多媒體物件屬性值的話，我們可以採用暴力法直接掃描檔案的關鍵字即可。但是我們選擇利用 compiler 的過程技術考量是為了原本的描述檔不僅僅記錄著多媒體物件屬性值，還記錄著劇情的呈現及它們之間的互動甚或是動畫呈現。此外如果我們利用暴力法來解譯描述內容的話，這對未來的描述格式維護相當不易。因此我們會利用到編譯器設計的過程間接產生的剖析樹(parsing tree)來記錄整個多媒體屬性值及劇情描述過程等等，此外 parsing tree 也比較好維護內容描述檔的結構及未來描述檔的維護工程，甚至在目標模

組(Target module)寫出相對應的 XML 文件也來得比較方便。

下圖 15 是運用製作編譯器過程來擷取多媒體物件屬性值的過程：

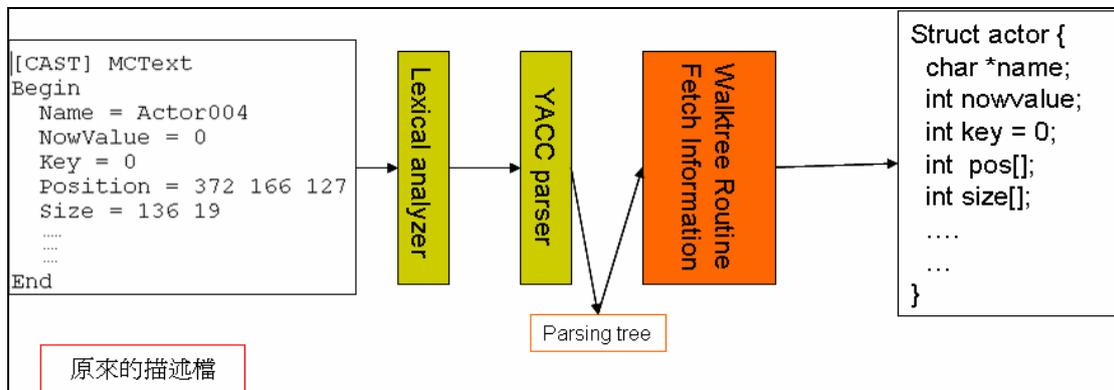


圖 15：parsing script 運作過程圖

我們看到上圖 15 中所所示的，原本的描述檔內容會經由詞彙分析器(Lexical Analyzer)產生字元流(tokens stream)，然後在交由 parser generator 的 yacc 產生的文法剖析器(Grammar Parser)進行文法比對，之後會產生一個 parsing tree 結構出來。我們以樣板式多媒體試題的內容來說，這裡的 parsing tree 結構會如下圖 16 所示：

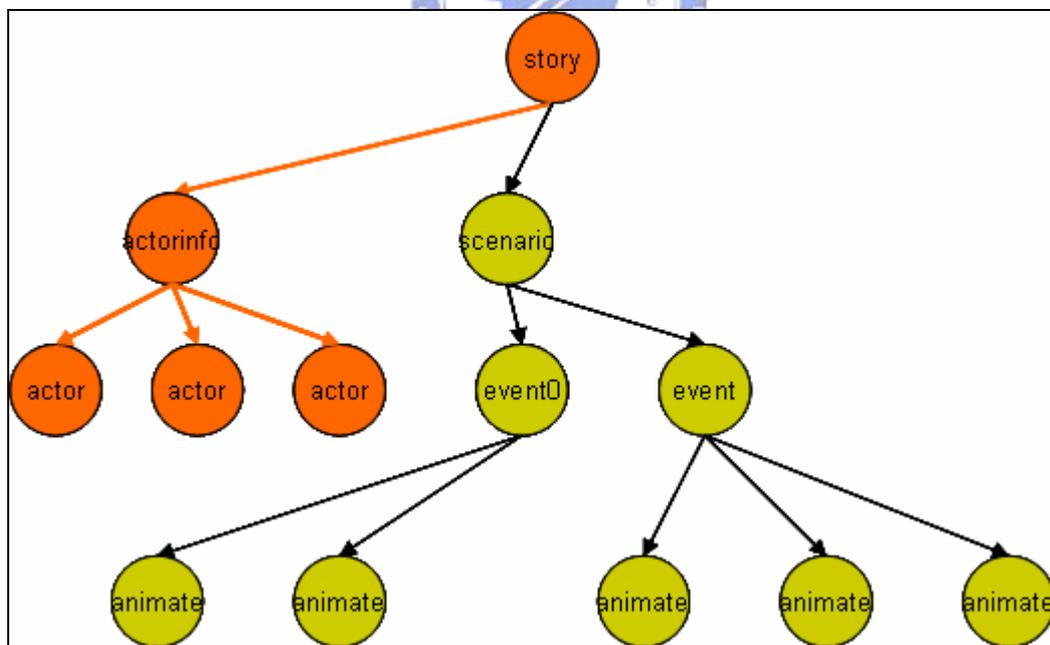


圖 16: Parsing Tree

而在描述檔裡的多媒體所有資訊都儲存在這棵樹狀結構裡，例如多媒體物件的屬性值就記錄在上圖 16 中的[actorinfo]節點下的子節點之下，描述劇情的部分則記錄在[scenario]節點下的子節點之下。要取得他們的資訊可以利用 Tree

Traversal 也就是圖 15 的 walktree routine。

接下來底下我會介紹比較跟compiler相關的細節部分。我們必須借助UNIX上的工具lexical analyzer generator-lex [25] 來完成我們的詞彙分析器(Lexical Analyzer)，以及利用UNIX上的工具parser generator-yacc [25] 來完成我們的文法剖析器(Grammar Parser)製作。

- 詞彙分析器(Lexical Analyzer):

首先我們必須先撰寫一份 lex 原始碼用來定義我們的描述檔所用的字詞，本描述檔所用的詞彙比較簡單，只能要夠辨認 Identity、Integer number、Keyword、String 及常見的 operator (+ - =) 即可，lex 的寫法是以 Regular expression 來辨認字是屬於哪種型態之 token。四種型態的 Regular expression 如下：

```
Integer number : [0-9]+  
Identity : [a-zA-Z_][a-zA-Z_0-9]*  
String : \"([^\n]|\\[\"\\n])*\"  
Keyword : \"CAST\" | \"MCText\" | \"Begin\" | \"End\"...
```

經由 lex 會將描述檔的文句產字元流(tokens stream)。

- 文法剖析器(Grammar Parser):

詞彙分析器(Lexical Analyzer)將原來的描述檔內容拆成字元流(tokens stream)，交由文法剖析器(Grammar Parser)以驗證描述檔裡的描述文法合理之外，並會同時產生 parsing tree 的資料結構。在製作文法剖析器必須撰寫一份 yacc 程式碼以定義描述檔描述文法，以下是以文字演員為例子：

```
CastInfo : LBRACK CAST RBRACK MCTEXT _BEGIN  
Cast_Statement_List _END  
  
Cast_Statement_List : Cast_Statement  
| Cast_Statement Cast_Statement_List  
  
Cast_Statement : Cast_Ident ASGN Expression_Right  
  
Expression_Right : Cast_Right_List  
| Cast_Ident
```

```

Cast_Right_List : Cast_Right
| Cast_Right_List Cast_Right
Cast_Right : Cast_Integer
| STRING
| LPAREN Cast_Integer COMMA Cast_Integer RPAREN

```

在比對文法的過程中，除了錯誤處理之外，再來必須在過程中把 Parsing Tree 以 bottom up 方式建出來，也就是在有用的文法裡撰寫 action code，通常是創建樹狀結構的節點(Tree Node)以記錄資訊，另外所謂有用的文法是在有需要的資訊必須在此文法比對將之保留住。

5.2.2 協調引擎設計製作

協調引擎(Negotiation engine)會將剛剛從描述檔剖析器(Script parser)解析的多媒體文件內容資訊，以及根據參數的輸入，進行改編協調。整個過程會經由四個步驟來完成：分析(Analysis)、暫時轉譯(Temporal translation)、推薦選擇(Recommended choice)、媒體轉換(Media Conversion)。

- 分析(Analysis)

這裡分析的主要目的是能夠區分原來的多媒體內容是什麼樣式的呈現，並將原來的內容作切割的動作，然後再進行相對應的行動裝置模擬呈現。由於英文的試題版型固定分成“stimulus”、“question”及“answer”三個段落，我們可以以這三部份作為段落區分(Paragraph)，再依據字數、字形大小、圖片大小、螢幕大小，進行頁數分配(pages)。我們將欲轉換輸出的模擬版型規劃成如下圖 17 之右邊四種 Case 的可能。

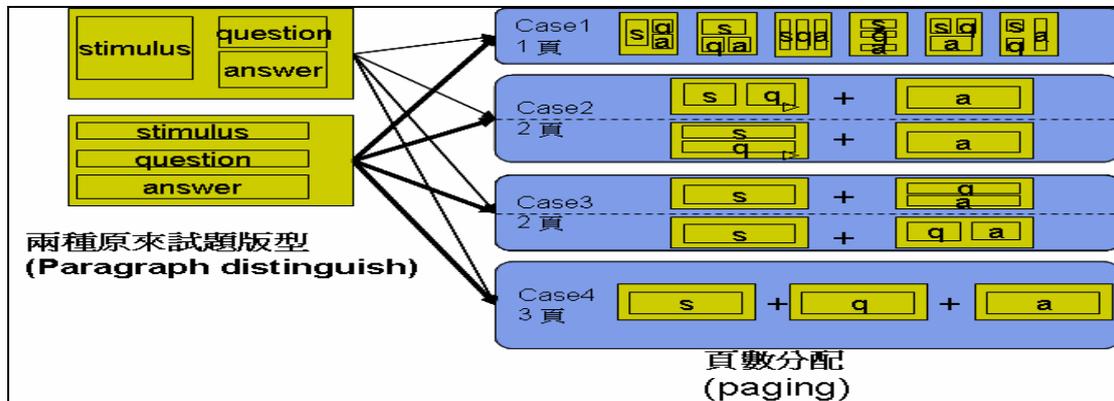


圖 17: 英文試題版型分配可能

■ 段落區分的演算法：

由於單純從文件資訊並沒有直接辦法來決定哪一個多媒體物件是屬於“stimulus”、“question”或“answer”段落，唯一可以決定的便是用座標來判定。

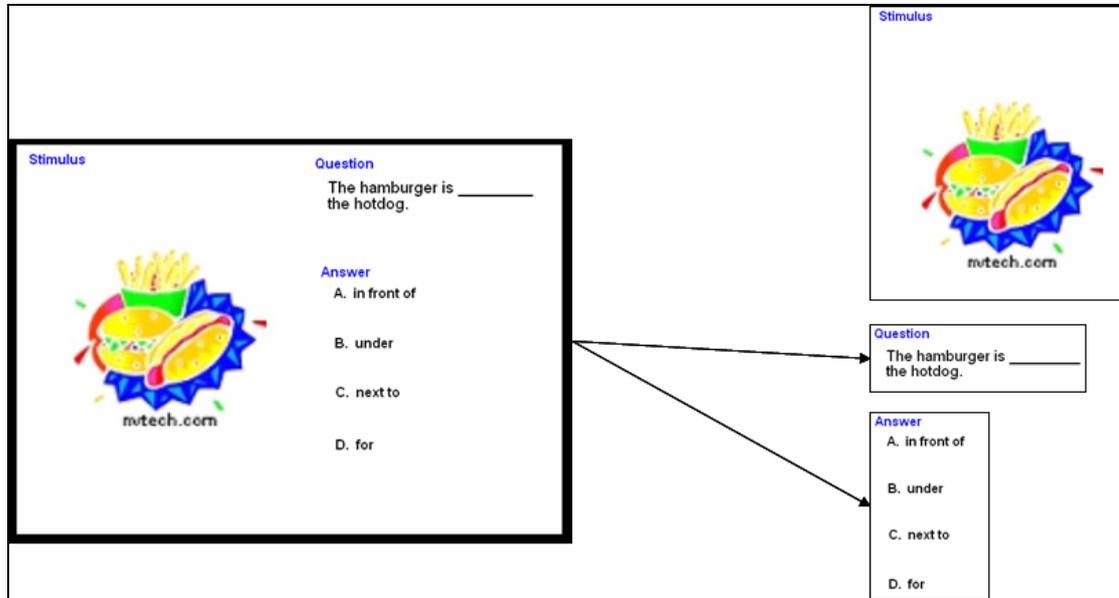


圖 18: 英文試題段落分配範例

以上面圖片的例子來說的話，演算法會先搜尋三個關鍵字[Stimulus]、[Question]、[Answer]，也就是本試題將會有三個段落，接下來會將其他的多媒體物件安排至這三個段落裡，安排方式依照位置的比較，虛擬碼如下：

```
Foreach Mediaobj in Story {
```

```
    If Mediaobj.x < Question.x && Mediaobj.x < Answer.x →  
    Mediaobj belongs to [Stimulus]
```

```
    If Mediaobj.x >= Question.x && Mediaobj.y < Answer.y →  
    Mediaobj belongs to [Question]
```

```
    If Mediaobj.x >= Answer.x && Mediaobj.y > Answer.y →  
    Mediaobj belongs to [Answer]
```

```
}
```

■ 頁數分配演算法：

對於頁數分配的話，以英文試題來解釋的話，我們要先界定四種轉換出去的case。

- ◆ Case 1: 是“stimulus”、“question”及“answer”這三個段落裡頭縮放後的內容可以在相對應的行動裝置一個頁面容納得下。
- ◆ Case 2: 是將原本的多媒體內容以兩頁方式來呈現。也就是“stimulus”及“question”這兩個段落裡縮放後的內容可以在相對應的行動裝置一個頁面容納得下，但“answer”這段落裡的内容無法在剩餘的空間容納，於是便將他獨立成一頁。
- ◆ Case 3: 是將原本的多媒體內容以兩頁方式來呈現。也就是“question”及“answer”這兩個段落裡縮放後的內容可以在相對應的行動裝置一個頁面容納得下，但“stimulus”這段落裡的内容無法在剩餘的空間容納，於是便將他獨立成一頁。
- ◆ Case 4: 這裡的情況必須將“stimulus”、“question”及“answer”各分成三頁來呈現，原因可能是“question”段落內容太多必須用一個頁面來呈現，或者另外一個情況是“stimulus”和“question”或“question”和“answer”兩個段落內容加起來會超過一個頁面。

以簡單的判斷式來描述上面這四種情況，如下圖 19：

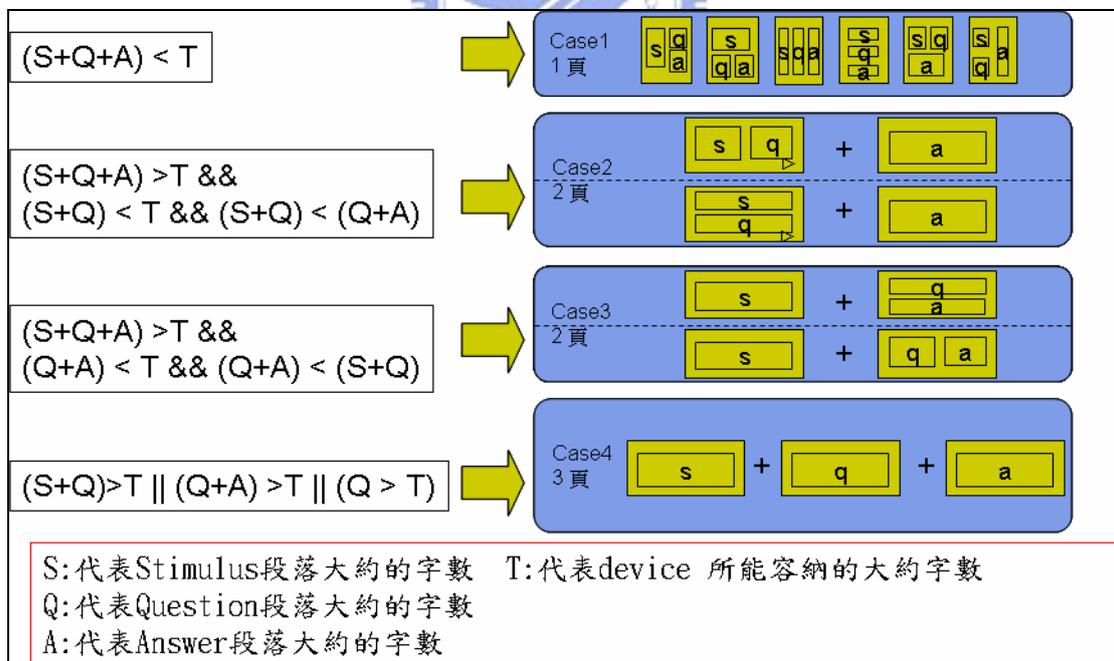


圖 19: 頁數分配演算法

我們以一個簡單的例子來詳細說明整個頁數分配過程，下圖 20 是三個段落的字數加總結果小於 PDA 最大可容納的字數，所以我們可以選定 case1 的情況，也就是以一個頁面來呈現即可，我們以一般 PDA 常見的螢幕大小 240x320 像素，

以及字形大小 10px，這樣的螢幕大小大約可以容納 400 個英文字，皆下來必須算出原來的每個段落大約的英文字數，在下面的例子來說“Stimulus”為包含一張圖片段落，對於包含圖片的內容，必須將他等比例縮小到相對的 size 然後再以 10px 的英文字大小來估測可以容納多少英文字。

以PDA screen size = 240x320,
font size = 10px
平均大概可容納T = 400words


+

+


171 + 61 + 88 = 320 < 400 (case 1)



若段落裡面含有圖片，我們將以圖片等比例縮小時的面積可能容納的字數來計算

圖 20: 頁數分配演算法範例

- 暫時轉譯(Temporal translation)

上一步驟的分析結果，我們可以得到欲轉換輸出的結果應該要分成多少頁數才能呈現；在這個步驟裡，我們必須暫時的轉換已經選定 case 裡的所有可能版型。當然每種版型必須搭配不同的文字斷行(Line breaking)，還有圖片縮放(Image scaling)，不過到目前只知道每個版型的樣式，並不知道每個樣式裡的區分大小，因此在這之前必須要先進行版型比例分配。暫時轉譯(Temporal translation)的整體運作過程可以如下圖 21 所示：

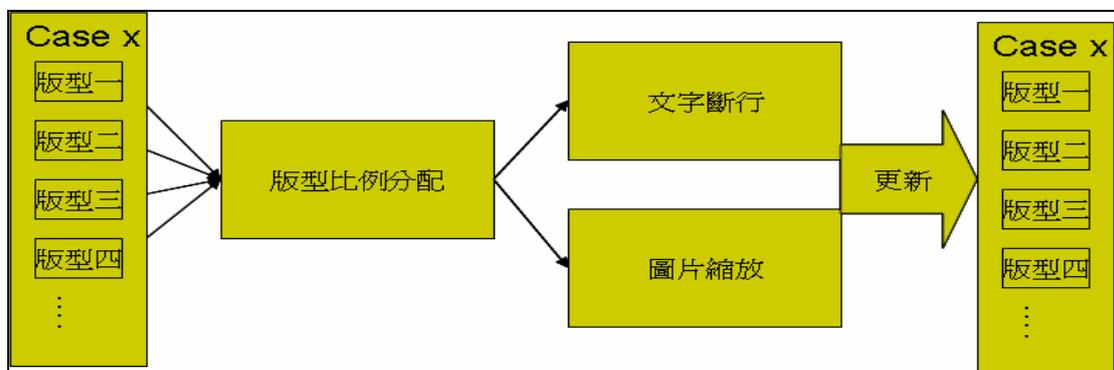


圖 21: Temporal translation 整體運作過程

皆下來會一一介紹版型比例分配演算法、文字斷行演算法以及圖片縮放演算法。

■ 版型比例分配演算法

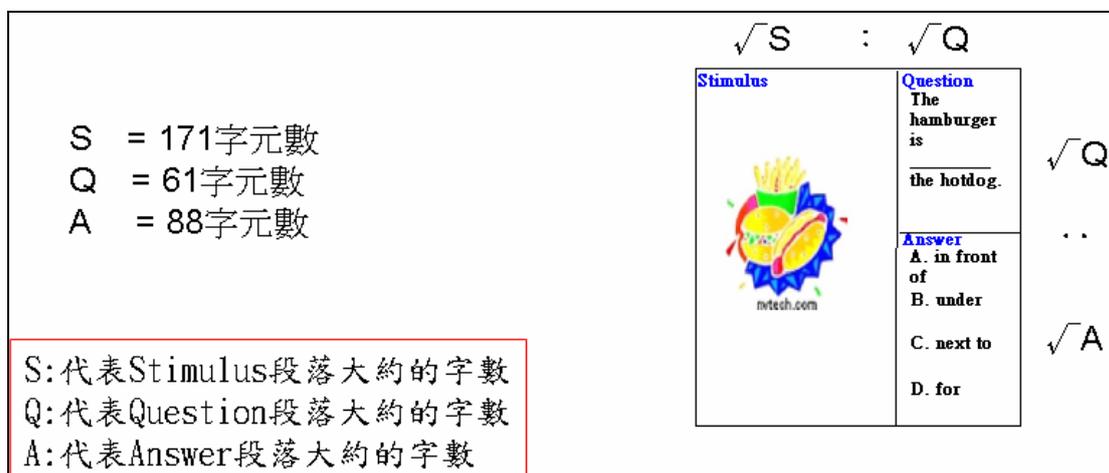


圖 22: 版型分配比例估計演算法

以上圖 22 的例子來說的話，我們選定 case1 的其中一版型來轉換，而他們個各段落的長寬分配比例我們會以他們的段落所佔的字數開跟號之比值來計算。這只是粗略的估計方式，並不能保證完全符合每個段落所需的大小。

分配完段落每個區塊的比例之後，接下來會進行所屬多媒體物件安排至相對的段落區塊裡，每個安排的動作很簡單，除了多媒體位置的設定外，如果是文字多媒體物件的話，會進行重新斷行的動作，如果是影像多媒體物件的話，會進行重新縮放(scaling)的動作。

■ 文字斷行演算法

精準的文字斷行必須考慮到每個英文字的長寬，在我們的系統裡會儲存著多種文字大小的英文字表[font table]，藉由字表的查詢來計算每個英文單字的長度。為了加數計算，字表以 hash 方式來儲存。下面圖 23 是以計算英文字“Stimulus”為例子。

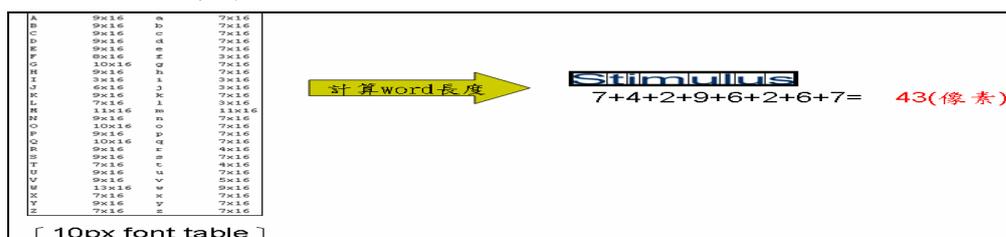


圖 23: 計算英文字長度範例

而字表[font table]的製作方式，我們利用 Microsoft visual c++6.0 的開發工具來創建 9~12pt 的字表，下面是 vc++ 的虛擬碼：

```

CFont cfont;

cfont.CreatePointFont(90,"Arial"); // 9 point 的 arial 字形

dc.SelectObject(&cfont);

for(int i=33 ; i <=126; i++) { // ascii code 33~126

    c = (char)i;

    GetTextExtentPoint(dc.m_hDC,c ,1,&csize);

//取得一英文字的大小

// csize.x  即是一英文字的寬

// csize.y  即是一英文字的長

}
    
```

要進行斷行的話，在填入段落區塊的過程中，必須一直比對區塊的寬度，只要下一個要填入的英文單字超過可用剩餘空間的寬度的話，資料結構裡會記錄斷行的記號。整體斷行機制的運行過程如下圖 24：

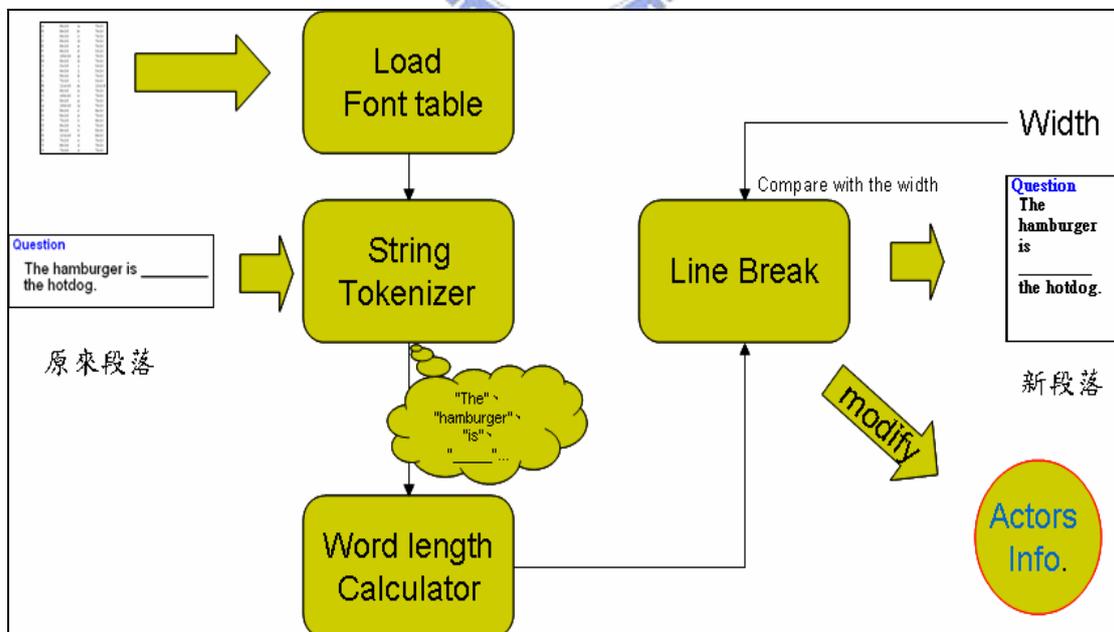


圖 24: 英文句子斷行演算法流程圖

■ 圖片縮放演算法

上面是處理文字多媒體物件的規劃，而遇到影像多媒體物件的，即要重新計算他的長寬，依據分配到的空間進行比例縮放。以下例子可以闡述影像縮放演算法：

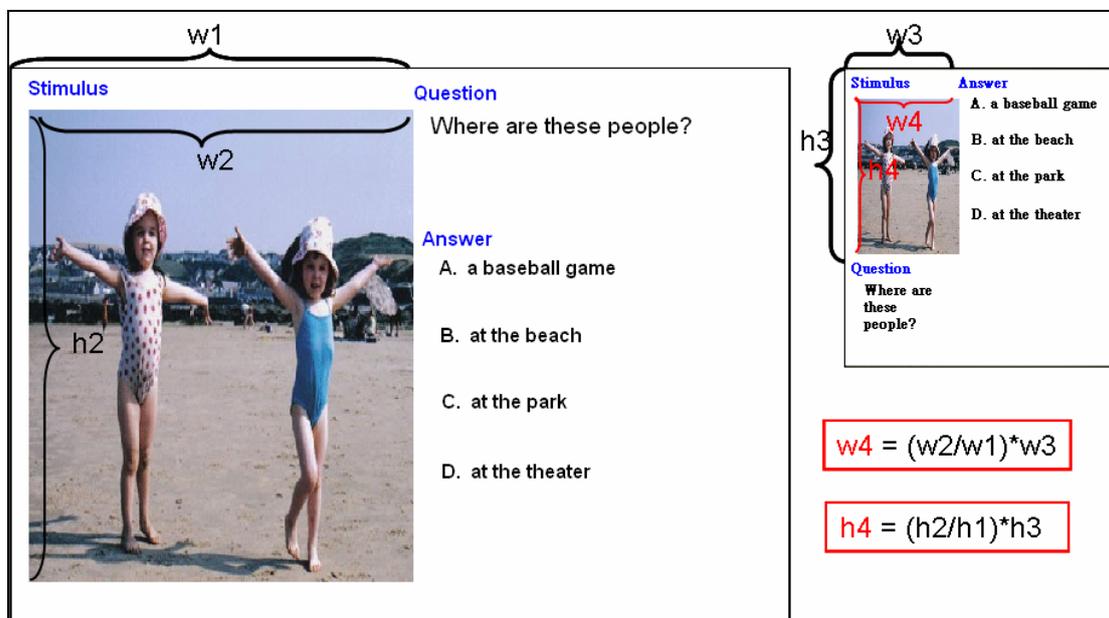


圖 25: 圖片縮放比例演算法範例說明

圖 25 左邊是原來的多媒體試題，原來“Stimulus”段落區塊長寬是 h_1, w_1 ，原圖片長寬是 h_2, w_2 ，經過分配比例得到的新區塊長寬是 h_3, w_3 。新圖片長寬 h_4, w_4 是我們要計算的。以右邊的比例公式可以很簡單的算出 h_4, w_4 結果。

● 推薦選擇(Recommended choice)

從上面的步驟下來，我們已經選定四種 case 之一種，並暫時的轉換到此 case 裡的所有版型。在這步驟中，我們要分析這些版型的好壞，以決定哪一個可以得到推薦的最佳轉換結果。選定的演算法會從三個分析項目去考量-斷行數、影像縮放比、重疊覆蓋與否：

- 1、刪除重疊或覆蓋的-希望所有內容能清楚及一覽無遺。
- 2、刪除斷行數多的-斷行數多的文句對於閱讀者會顯得吃力。
- 3、選擇圖縮放長寬比最接近 1.0。

這裡要闡述的影像縮放長寬比是以原來影像長寬比和新的影像長寬比，兩個相除的比值，而其原因是為了達到影像最不失原影像所傳達的意涵，我們以下圖 26 說明影像縮放長寬比：

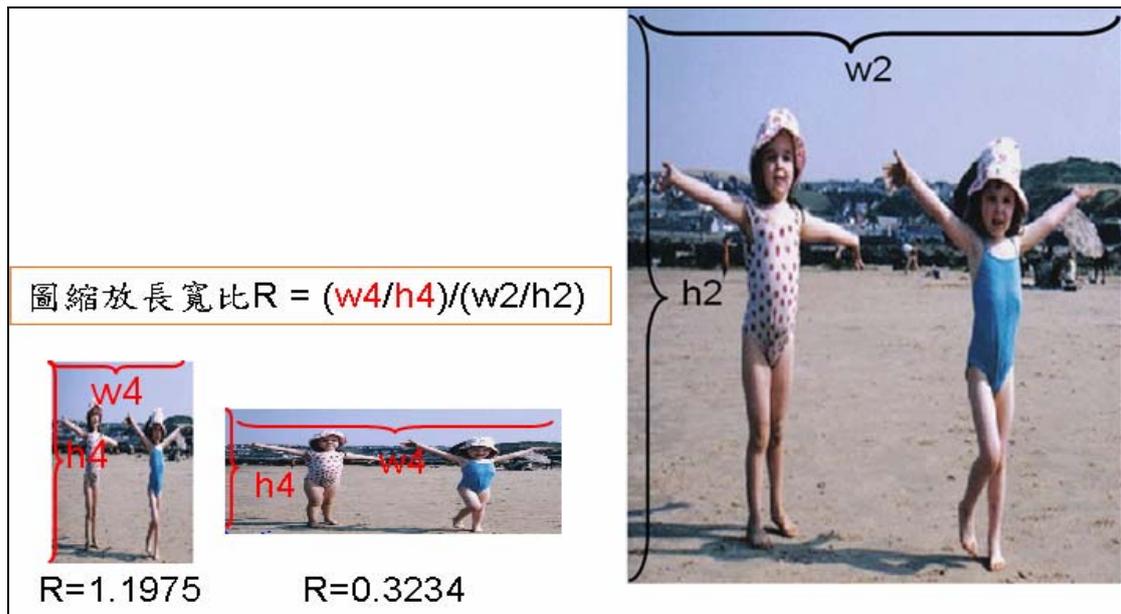


圖 26：不同圖縮放長寬比例差別

我們可以看到圖 26 右邊為原始在 PC 上呈現的圖片，左邊兩小圖片其縮放長寬比分別為 1.1975 及 0.3234，很明顯的可以看出兩張小圖片的好壞，也就是其縮放長寬比最近 1.0 的為 1.1975，其圖片內容的比例最接近原始圖片內容的比率。

再來我們將以下圖 27、28 的例子來說明整個推薦選定過程：

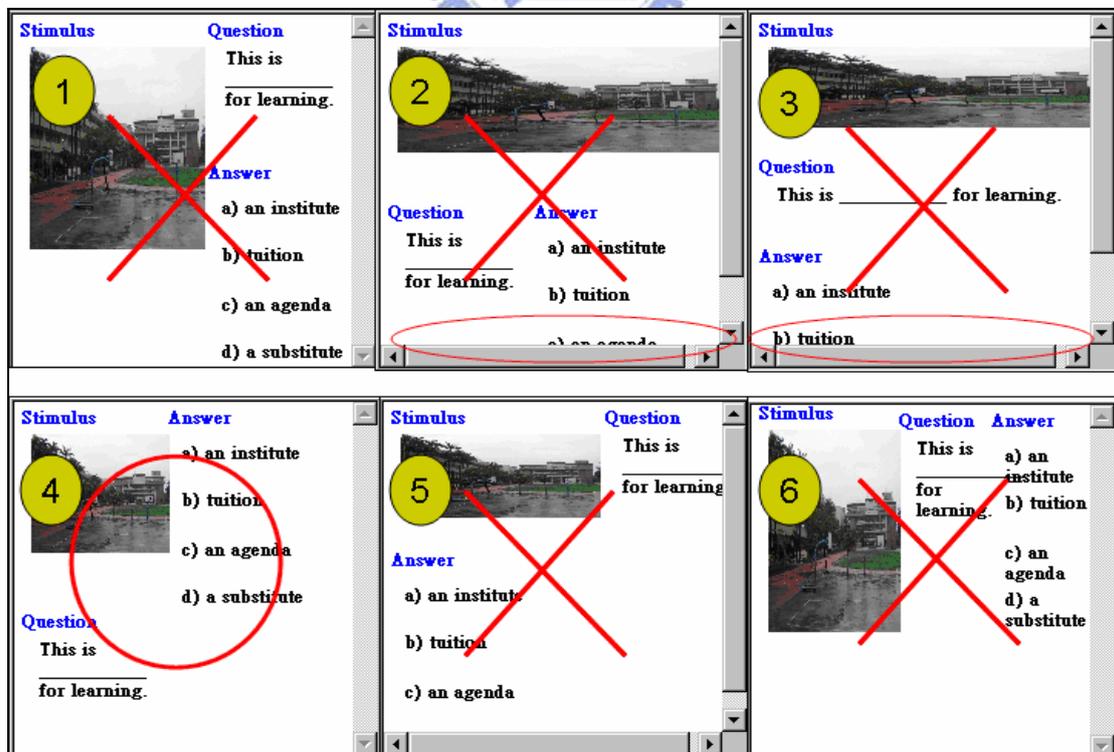


圖 27：推薦選擇版型演算法例子

可能版型	斷行數	圖長寬	overlap
	2	0.708448	0
	2	2.572785	1
	0	3.368009	1
	2	0.952883	0
	2	1.955918	0
	6	0.534443	0

圖 28: 圖 27 例子的數據分析

以上面這例子來說的話，我們會先運行演算法的第一個步驟，也就是先將有覆蓋到的版型刪除(圖 27 的 2、3)，再來刪除斷行數多的(圖 27 的 6)，如果有相同的斷行數版型(圖 27 的 1、4、5)，再選取圖縮放長寬比的比值最接近 1 的(圖 27 的 4)，所以我們得到可以推薦的第四版型給使用者。

● 媒體轉換(Media Conversion)

我們除了文字外會以開放函式庫(open library)或現有的函示庫來組織我們這裡的多媒體物件轉換，主要的多媒體物件轉換包括了下列：

■ 圖片轉換：

函式庫: DevIL on unix version

轉換內容: 圖檔格式轉換(png、bmp、jpg、gif)、縮放轉換(scaling)

方法:

1. 下載 DevIL 並安裝

bmp 轉 jpg 300x200

C language like syntax:

```
illnit();
```

```
ilulnit();
```

```
ilLoadImage("temp.bmp");
```

```
iluScale(300, 200, 1);
```

```
ilSaveImage("temp.jpg");
```

■ 聲音轉換：

函式庫: Lame Mp3 encoder, Microsoft media Encoder8.0 on win32

轉換內容： 聲音格式轉換(wav、mp3)

方法：

1. 下載 lame library windows version 運用 command line 完成

Wav 轉 mp3 : /lame.exe temp.wav temp.mp3

2. 下載 media encoder 運用 command line 完成

Wav 轉 mp3 : cscript.exe wmcmd.vbs -input

temp.wav -output temp.mp3 -option value

■ 影片轉換：

函式庫: Microsoft media Encoder8.0

轉換內容： 影片格式轉換(mpg、wmv)、縮放轉換

方法：

1. 下載 media encoder 運用 command line 完成

mpg 轉 wmv : cscript.exe wmcmd.vbs -input

temp.mpg -output temp.wmv -v_height height -v_width
width

這個指令會將原本 temp.mpg 影片檔轉成另一種格式 temp.wmv 並且能夠將 Dimension 縮放。

5.2.3 目標模組(Target module)製作

上面小節我們已經選取最佳的版型等待轉換，只要我們能夠掌握協調完的多媒體物件資訊是對，這個小節將介紹實際轉換已選定的版型到特定多媒體內容描述格式，此格式來自使用者參數輸入。當然我們必須先知道這些格式的語法，事實上大部分的多媒體內容描述格式都是 XML 格式的描述語言，因此我們可以利用 XSLT 進行他們的轉換動作。而轉換過程如下圖 29 所示：

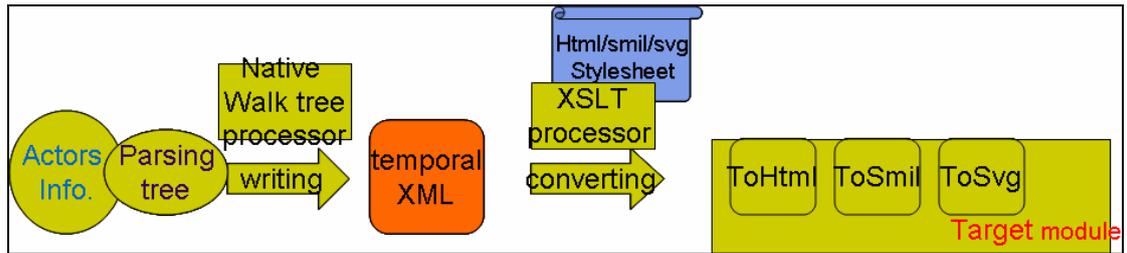


圖 29: 運用 XSLT 轉換相對應的多媒體腳本 XML 文件

可以看到我們更改協調後的所有多媒體屬性將存在上圖所示的 ActorsInfo. 這個資料結構裡，我們搭配利用之前在 5.2.1 節提到產生的 Parsing Tree，寫回 Temporal XML 格式，再交由 XSLT processor 搭配已經寫好特定轉出格式的 XSL style-sheet 來輸出我們預定的描述檔。下圖 30 是一個 Temporal XML 簡單的例子：



圖 30: Temporal XML 範例

XSLT 是把 XML 文件轉換成另一 XML 文件的 XML 語言。從上圖 29 轉換過程中，Temporal Xml 即可以是 Target Xml(HTML、SMIL、SVG)，而不需要經由 XSLT processor 轉換，不過這樣一來對於目標模組(Target module)的開發者來說都必須為特定的轉換格式去修改撰寫圖 29 中的 Walktree 程式。

但如果經過 XSLT 轉換，目標模組(Target module)只要專注於 XSLT style-sheet 的撰寫即可，將來擴充或修改成其他格式也較為便捷。

如果要用 XSLT 技術來轉成 HTML 必先要準備一份 HTML 的 XSLT style-sheet，如下圖所示：

```
<xsl:stylesheet version = '1.0'
  xmlns:xsl = 'http://www.w3.org/1999/XSL/Transform' >
<xsl:template match = "/" >
<html>
<body>
<xsl:for-each select = "/Story/ActorInfo/mctext" >
  <nobr >
    <xsl:attribute name = "id" >
      <xsl:value-of select = "Sid" />
    </xsl:attribute >
    <xsl:attribute name = "style" >
      <xsl:text >position:absolute</xsl:text >
      <xsl:text >;top:</xsl:text >
      <xsl:value-of select = "posx/." />
      <xsl:text >;left:</xsl:text >
      <xsl:value-of select = "posy/." />
      <xsl:text >;font-size:</xsl:text >
      <xsl:value-of select = "fontsize/." />
      <xsl:text >;color:</xsl:text >
      <xsl:value-of select = "color/." />
    </xsl:attribute >
    <xsl:for-each select = "lb" >
      <xsl:value-of select = "." />
      <br />
    </xsl:for-each >
  </nobr >
</xsl:for-each >
</body >
</html >
</xsl:template >
</xsl:stylesheet >
```

圖 31: 一份根據 temporal xml 轉換到 HTML 的 XSLT style-sheet

XSLT style-sheet 定義了他將 Temporal XML 轉換到另一種 XML 的規則。而這裡我們定義轉換成 HTML 的規則很簡單，對於轉換多媒體文字而言，我們可以先 match 到 /Story/ActorInfo/mctext 的標籤並擷取他的屬性值，並將這些數值轉成 HTML 搭配標準屬性方式(CSS)來幫多媒體物件訂定大小等其他屬性值。此外文字多媒體比較要注意的是斷行表示，Temporal Xml 描述斷行是以 <lb> 標籤來表示，轉至 HTML 則必須要用
 標籤來替換。其他多媒體格式 SVG、SMIL 則分別利用 <tspan> 和外部檔案描述來幫多媒體文件做斷行的描述。

5.3 應用端之設計與實作

5.3.1 Web-based 改編協調開發製作

本小節介紹的應用端程式是針對給管理協調者來使用，會依前述功能面的需求來設計，以及按照架構介紹的訊息傳遞模式來實作。

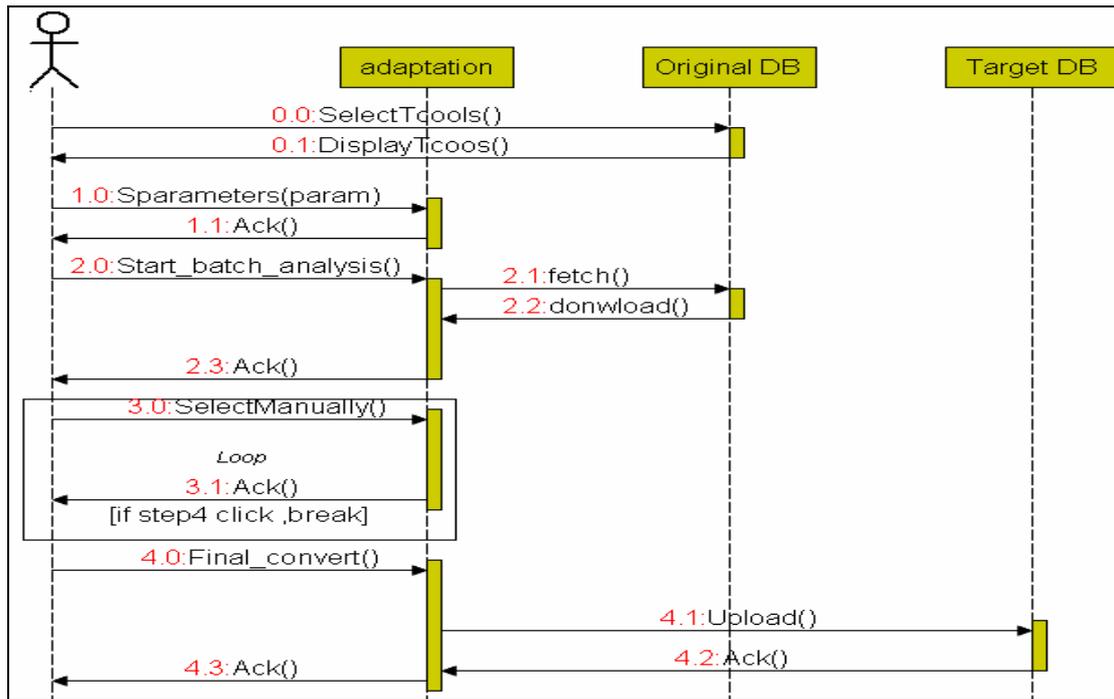


圖 32: 協調者之 web-based 運作圖

而下面幾個步驟是按照圖 32 的訊息運作圖來描述：

- 0: 從未改編協調的內容伺服器 (Server for Content without adaptation) 資料庫選擇多媒體內容
- 1: 輸入參數給予改編協調伺服器 (Adaptation Server)
- 2: 改編協調伺服器開始批次從資料庫擷取媒體資源分析
- 3: 操作員可以手動選取其他的轉換結果
- 4: 最後批次轉換、並上傳至已改編協調的內容伺服器 (Server for adapted content) 之資料庫

● 選題畫面之流程設計

我們要轉換協調的是多媒體英文試題，對使用者會以管理帳號登入，並可以建立一份新試卷，試題資料庫的來源是架構於 FreeBSD5.0 MySQL 平台之上。而本應用程式會架設於 FreeBSD4.8 的另外一台 apache 網頁伺服器，因此使用者會經由 PHP 程式的資料庫查詢，並可讓使用者過濾篩選試題，當使用者選好題目出完一份待轉譯的試卷，按下確定扭，會經過一道儲存的手續，整個處理過程如下：

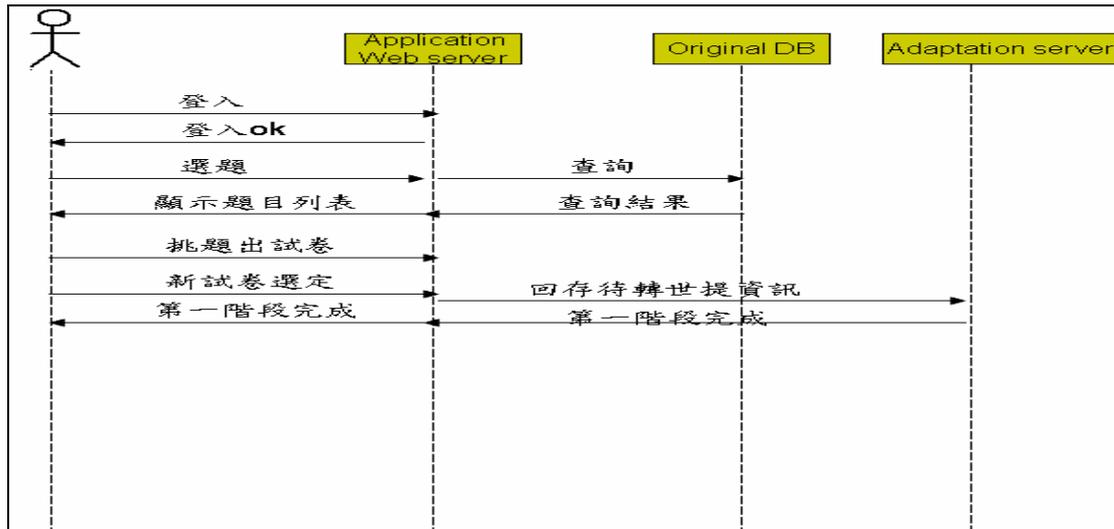


圖 33: 出題使用者從登入到選題出試卷的第一階段流程

● 輸入參數之流程設計

這裡設計必須讓出題協調者能手動輸入，他欲轉換的目標裝置的重要參數，包括了-裝置的螢幕大小、字體大小、圖片支援格式、以及最後欲轉換出的多媒體內容文件格式。當選定完畢後，這時會再跟改編協調伺服器(Adaptation Server)回報以儲存資訊。

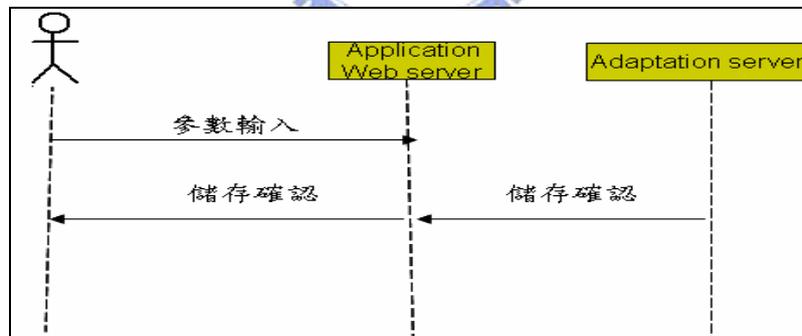


圖 34: 使用者輸入參數即時回傳確認動作流程圖

● 批次協調分析

一份試卷可能有眾多的題目，為了讓使用者不需要一個一個去協調，這裡設計會一次批次協調全部。由於改編協調伺服器(Adaptation Server)的設計一次要求只能協調單一試題，而使用端預設是 Internet Explorer，我們會發送多次的以 Http Request 來達成批次協調，而在單一網頁程式要發 Http Request 的話，我們必須要用到 ActiveX Object control 夾於網頁設計中來完成，協調完的結

果更可以讓使用者知道。下圖 35 面是單一網頁作批次協調必須跟伺服器溝通的運作過程，以及 Http Request 運用 ActiveX Object control 的 JavaScript 程式寫法。

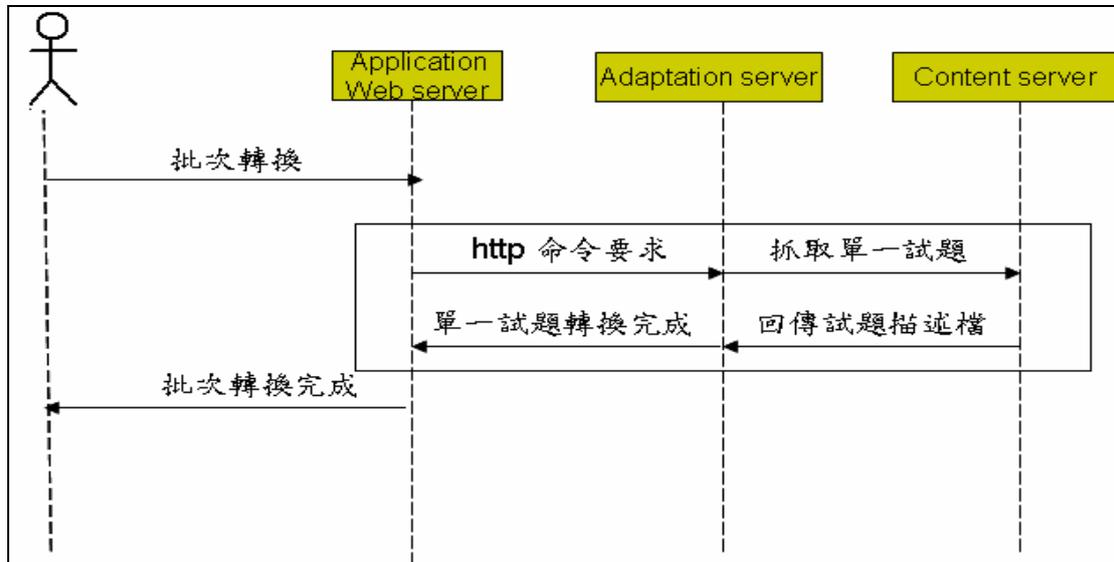


圖 35: 使用者進行批次轉換試題的過程圖

```

function sendrequest(url) {
  if (window.ActiveXObject)
  {
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    if (xmlhttp) {
      xmlhttp.onreadystatechange=disptcools;
      xmlhttp.open("GET",url,true);
      xmlhttp.send();
    }
  }
}

```

● 手動選取協調後之版型

在上一步驟協調完的結果並不一定能夠滿足使用者的喜愛，所以這裡設計方式是讓使用者可以用手動方式選擇其他額外的版型。選定後的結果將再一次回傳給改編協調伺服器(Adaptation Server)知道。

- 上傳至已改編協調的內容伺服器

前面的批次協調，只是作單純的分析協調。這裡會做最終的轉換動作，包括了欲轉換的格式寫入、以及多媒體物件轉換、最後會壓縮成一個 Zip 檔案然後上傳至試題平台，此試題平台專存放適合行動裝置多媒體內容的資料庫。這些序列性的動作亦是用批次來完成，對管理者來說，這裡要花比較多的時間等待。從英文試題管理協調介面來看，這裡同樣必須發出多個 Http Request，如同上面步驟的批次協調一樣來完成批次轉換動作到最後的上傳內容，我們可由下圖 36 來闡釋：

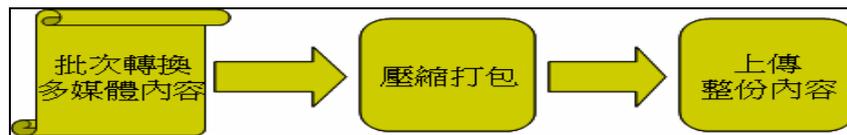


圖 36：批次轉換至上傳過程

5.3.2 PDA Application 客戶端播放器製作

本小節基於目前的行動裝置對於常見的多媒體內容格式(HTML、SMIL、SVG)的播放器有部分功能的缺憾，所以這裡介紹的應用端程式是針對給行動裝置的使用者來使用，主要功用在於提供使用者能播放多媒體試題之能力，並具有上網和改編協調伺服器(Adaptation Server)溝通、動態下載試卷之能力。下圖 37 是客戶端的 PDA 應用程式的 sequence diagram 描述如何跟改編協調伺服器溝通，以及如何透過改編協調伺服器動態產生多媒體內容進而下載回來觀看。

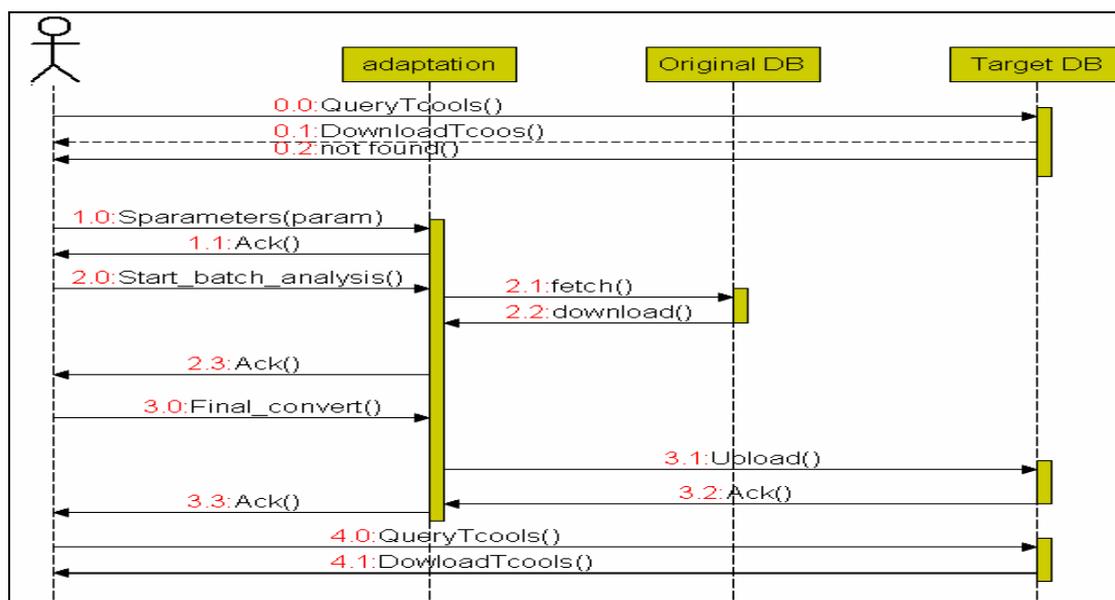


圖 37: PDA 端應用程式運作圖

- 0: 從已改編協調的內容伺服器(Server for adapted content) 資料庫查詢下載試題，如果資料庫不存在此試題(0.2)，將進行動態轉換產生試題(1-4)。
- 1: 傳入 device 支援程度之參數給予改編協調伺服器 (Adaptation Server)
- 2: 改編協調伺服器開始批次從資料庫擷取試題資源分析題目
- 3: 最後批次轉換、並上傳至已改編協調的內容伺服器資料庫
- 4: 從已改編協調的內容伺服器資料庫下載剛剛動態轉好的

- 偵測 Device Profile 之能力

此功用在於能夠偵測行動裝置的支援程度，以利在和改編協調伺服器 (Adaptation Server) 溝通時作為參數之參考用途。目前可以偵測螢幕大小、字形大小設定。

- 從特定目標內容伺服器下載內容之能力

本下載方式透過 Http 通訊協定方式下載，而已改編協調的內容伺服器(Server for adapted content)也需搭載相對應的動態網頁程式，此網頁程式以 JSP 來撰寫，以進行和行動裝置的應用程式溝通。行動裝置應用程式會夾帶裝置支援程度參數發出 Http Request 取回試卷清單，使用者點選清單上的某份試卷進行下載動作，而下載回來是個壓縮檔，因此本程式會搭載一個解壓縮程式，所以使用者即需等待下載跟解壓縮時間而後進行試題播放。下圖 38 顯示這步驟會牽扯到個幾個 routine。

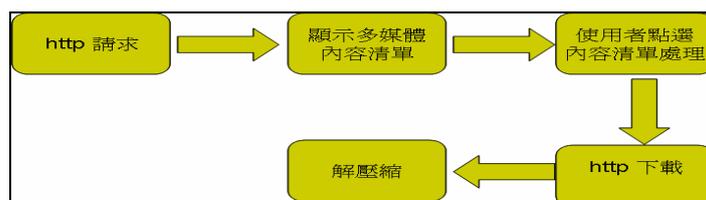


圖 38: PDA 應用程式之下載試卷的子模組流程圖

- 和改編協調伺服器溝通之需求

這裡主要針對使用者進行動態轉換內容的功能需求，也就是當試卷資料庫的某份試卷不存在的情況下，本使用端應用程式要進行和改編協調伺服器(Adaptation Server)溝通，進而轉換試卷，然後會經由上個步驟再次下載試卷回來觀看。

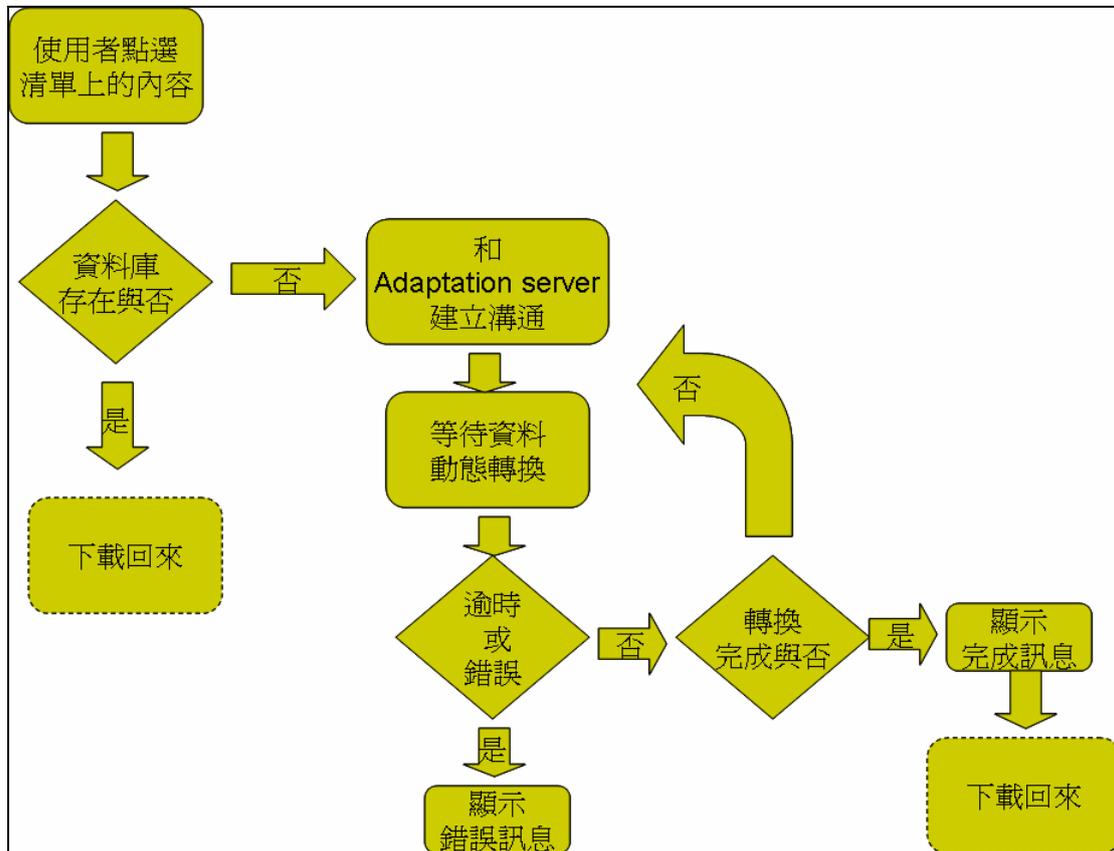


圖 39: PDA 使用者對於下載試題的過程圖

使用者一開始會點選清單上的內容，這時 PDA 端應用程式會去資料庫查詢此份多媒體是否存在於實體資料庫裡，如果是的話 PDA 應用程式會直接下載這份內容回來觀看。反之如果不存在於實體資料庫，PDA 應用程式會和改編協調伺服器(Adaptation Server)建立連線並準備嘗試作動態轉換的動作，這時對 PDA 使用者必須等待轉換結果，轉換過程中如果逾時或是有錯誤訊息會提供給 PDA 使用者知道，否則使用者必須持續等待直到完成為止而後再下載回來。

- 多媒體內容劇情播放能力

本部分主要實作出能夠播放經由本研究內容改編協調(Content adaptation)轉換出來的自訂行動裝置多媒體內容劇情描述格式-XMG。XMG 本身亦是 XML Based 的描述語法。而多媒體物件支援程度包含圖片的 BMP 格式、聲音的 MP3 WAV 格式、影片的 WMV 格式、文字 9-11pt Arial 格式。本播

放劇情模組採用 WINCE4.0 MFC 架構撰寫而成。在瞭解播放劇情實作之前先瞭解一下 XMG 的描述方式。另外為方便起見，劇情描述內容由多媒體物件來組成，凡構成多媒體劇情描述的多媒體物件，我們將他簡稱「演員」。

```

- <Story>
- <ActorInfo>
  <actor type="TEXT" name="Actor000" file="Stimulus" font="size(13);style(artist);color(0,0,255)" path="5,5" />
  <actor type="IMAGE" name="Actor" file="@Res002.BMP" frames="1" path="10,24" />
  <actor type="TEXT" name="Actor001" file="Question" font="size(13);style(artist);color(0,0,255)" path="5,5" />
  <actor type="TEXT" name="Actor002" file="The man and the woman are \npreparing to get married. Who is \non the left?" font="size(13);style(artist);color(0,0,0)" path="15,23" />
  <actor type="TEXT" name="Actor003" file="Answer" font="size(13);style(artist);color(0,0,255)" path="5,117" />
  <actor type="TEXT" name="Actor004" file="A. The bridegroom." font="size(13);style(artist);color(0,0,0)" path="15,134" />
  <actor type="TEXT" name="Actor010" file="B. The bride." font="size(13);style(artist);color(0,0,0)" path="16,164" />
  <actor type="TEXT" name="Actor011" file="C. The groom." font="size(13);style(artist);color(0,0,0)" path="16,194" />
  <actor type="TEXT" name="Actor012" file="D. The wife." font="size(13);style(artist);color(0,0,0)" path="16,226" />
  <actor type="TEXT" name="next" file="[Next]" font="size(12);style(artist);color(255,0,0)" path="200,220" />
  <actor type="TEXT" name="pre" file="[Previous]" font="size(12);style(artist);color(255,0,0)" path="180,220" />
</ActorInfo>
- <Scenerio>
- <event name="prelude" vscrollbar="false" hscrollbar="false">
  <animate name="Actor000" action="SHOW" />
  <animate name="Actor" action="SHOW" />
  <animate name="next" action="SHOW" />
</event>
- <event name="next" vscrollbar="false" hscrollbar="false">
  <animate name="Actor001" action="SHOW" />
  <animate name="Actor002" action="SHOW" />
  <animate name="Actor003" action="SHOW" />
  <animate name="Actor004" action="SHOW" />
  <animate name="Actor010" action="SHOW" />
  <animate name="Actor011" action="SHOW" />
  <animate name="Actor012" action="SHOW" />
  <animate name="pre" action="SHOW" />
</event>
- <event name="pre" vscrollbar="false" hscrollbar="false">
  <goto name="prelude" />
</event>
</Scenerio>
</Story>

```

圖 40: 一份 XMG 描述格式範例

上面是一份 XMG 多媒體英文試題描述例子，多媒體物件屬性值我們利用“actorInfo”標籤節點下的子標籤“actor”來定義，對於陳述各個演員如何呈現播放，我們定義在“Scenerio”的標籤下。我們將這份 XML 文件轉成樹狀結構解釋比較來得清楚。

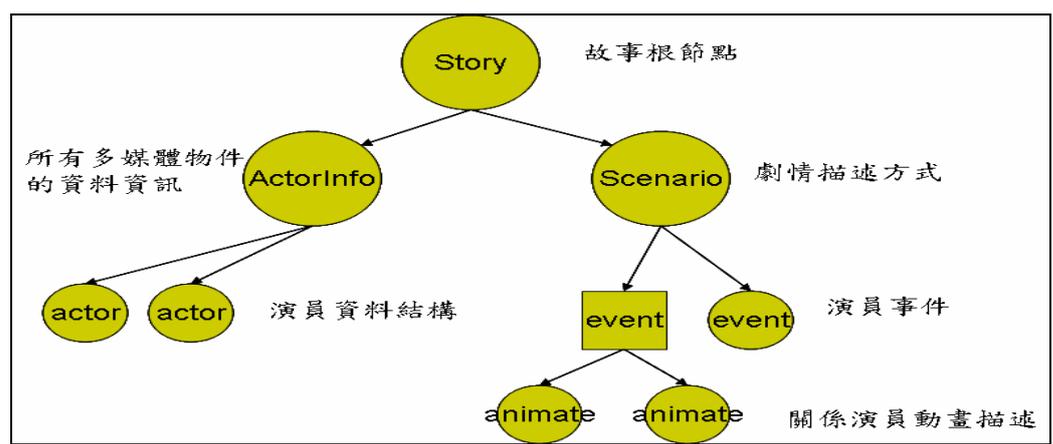


圖 41: XMG 轉換成樹狀結構圖

很明確的，播放器一定要有個 XML parser。其實 XML parser 可以用現成的 MSXML 來用，但基於本 XML 文件簡單，因此本研究實際實作 XML parser，以減少記憶體使用量，及提升效率。Parsing 完多媒體文件後，我們將所有演員的資訊，也就是上圖的“actor”節點資訊儲存到一個資料結構如下：

```

Class Actor {
    CString name;
    CString type;
    CString resource_name;
    Int path[];
    Bool play();
    Bool stop();}

```

程式在 run-time 裡，記憶體會一直記錄著上圖的樹狀結構。當劇情描述事件發生時(包括開場劇情、使用者互動事件劇情)，程式會把“scenario”這個節點找出來，並延伸下去找到相對的事件處理節點(上圖的“event”節點)，然後進行動畫劇情的呈現(“animate”節點)。“Animate”節點描述了演員播放聲音、路徑演出、影片播放、圖片呈現等等。

而在處理音樂部分，程式裡會用到開放函示庫(open source library)-[fmodce library]。他支援音樂大檔案的串流(stream)播放及小檔案的完全載入(full load)，在這裡應考量多媒體英文試題的音樂檔案大小大多很大。因此採用串流(stream)方式播放，如果以完全載入來播放也將造成記憶體的限制及播放時延遲(delay)情況。而用串流(stream)播放的語法如下：

```

FSOUND_Init(44100, 32, 0) //初始化聲音系統，僅需程式開始呼叫一次
sstrm = FSOUND_Stream_Open(L"/TEST.MP3", FSOUND_NORMAL ,
0,0);
//這裡打開文件是準備在 real-time 裡播放，並未解碼整份文件。
FSOUND_Stream_Play(FSOUND_FREE,sstrm); //開始播放音樂
FSOUND_Stream_Stop(sstrm); //停止音樂
FSOUND_Close(); //僅需程式結束時呼叫

```

處理影片的技巧比較複雜些，雖然有提供開放原始碼(open source)供人參考，但對於嵌入在應用程式上，比較難以控制。因此本影片處理會以 activex control 方式呼叫 pocketpc 內建 mediaplayer 來播放檔案，並鑲嵌於我們設定的視窗裡。首先必須先安裝 microsoft 提供的 pocketIE 控制物件，再來必須對提供的 mediaplayer control 介面有一定的瞭解，對 mediaplayer control 的初始化如下步驟：

```

CComPtr<IWMP> m_spWMPPlayer;
QueryInterface(__uuidof(IWMP),(void**)&m_spWMPPlayer);
m_spWMPPlayer->QueryInterface( IID_IConnectionPointContainer,
(void**)&spConnectionContainer );
spConnectionContainer->FindConnectionPoint( __uuidof(_IWMPEve
nts), &m_spConnectionPoint );

```

之後你就可用 CComPtr<IWMP> m_spWMPPlayer; 此物件來播放影片檔。
我們需要的物件 method 有下列：

```
m_spWMPPlayer->put_FileName();  
m_spWMPPlayer->Stop();  
m_spWMPPlayer->put_AutoStart();
```

此外要隱藏原本 mediaplayer 的操作使用介面可以用如下的 method:

```
m_spWMPPlayer->put_ShowControls();  
m_spWMPPlayer->put_ShowAudioControls();  
m_spWMPPlayer->put_ShowPositionControls();  
m_spWMPPlayer->put_ShowTracker();
```

在處理動畫劇情的時候，對於程式播放時的掌控相當的重要，面對處理眾多動畫物件，比較好的方式是運用多個行程或多執行序來管理，當然程式要能避免死當情況。在 evc++ 底下我們可以運用 win32 api 的 CreateThread() 來創建執行序使用。使用方法如下說明：

```
CreateThread(  
    NULL,  
    0, //初始化 stack size  
    (LPTHREAD_START_ROUTINE)ThreadProc, //處理 Thread 的函式名  
    (LPVOID)this, //新 Thread 的處理參數  
    0,  
    &thread  
);  
Void ThreadProc() {  
    While(1) {  
        //處理動畫動作  
        Sleep(1);  
    }  
}
```

六、應用範例

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，主要有下面四項。

1. 管理改編協調者的操作畫面
2. PDA 客戶端瀏覽裝置操作畫面
3. 改編協調前後內容呈現差別
4. 靜態和動態轉換及混合方式轉換數據分析

6.1 管理改編協調者的操作畫面

(1) 管理者以帳號密碼登入

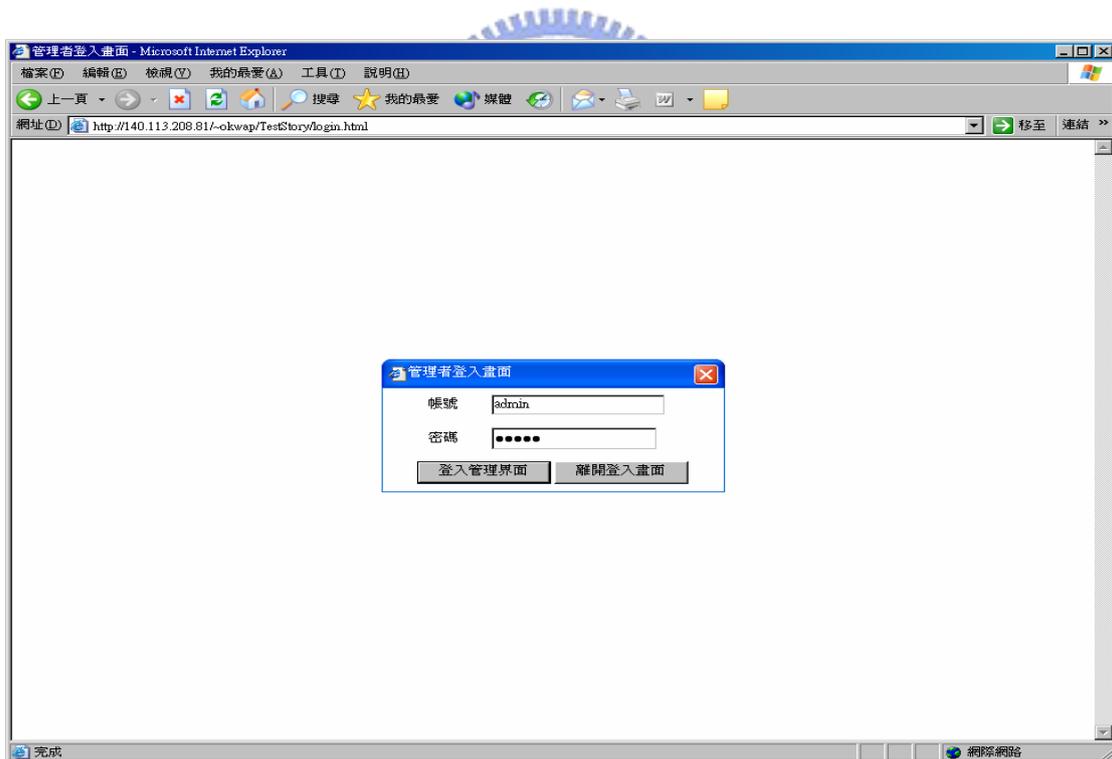


圖 42: 管理協調者登入畫面

(2)點選左邊的“新增試卷”，將出現如下圖右邊的選題畫面，出題方式可篩選試題的等級難易度或是類型而後挑題。

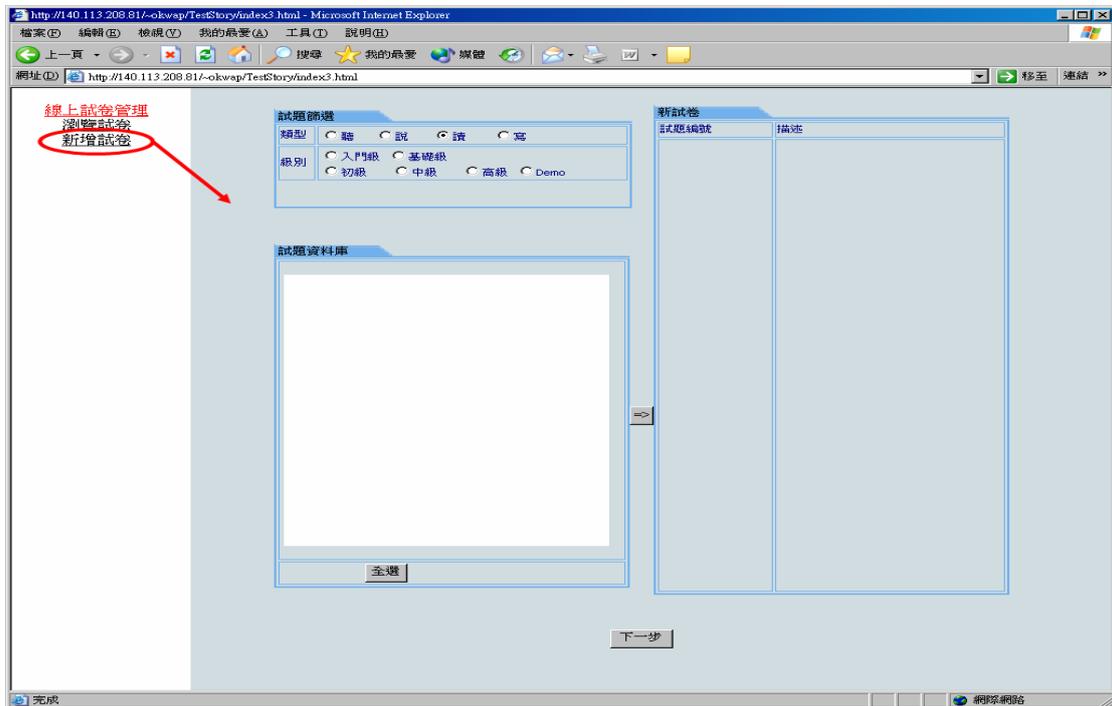


圖 43: 進入編輯一份試卷畫面

(3)可以從篩選的試題列表挑出試題放入右方的新試卷框裡

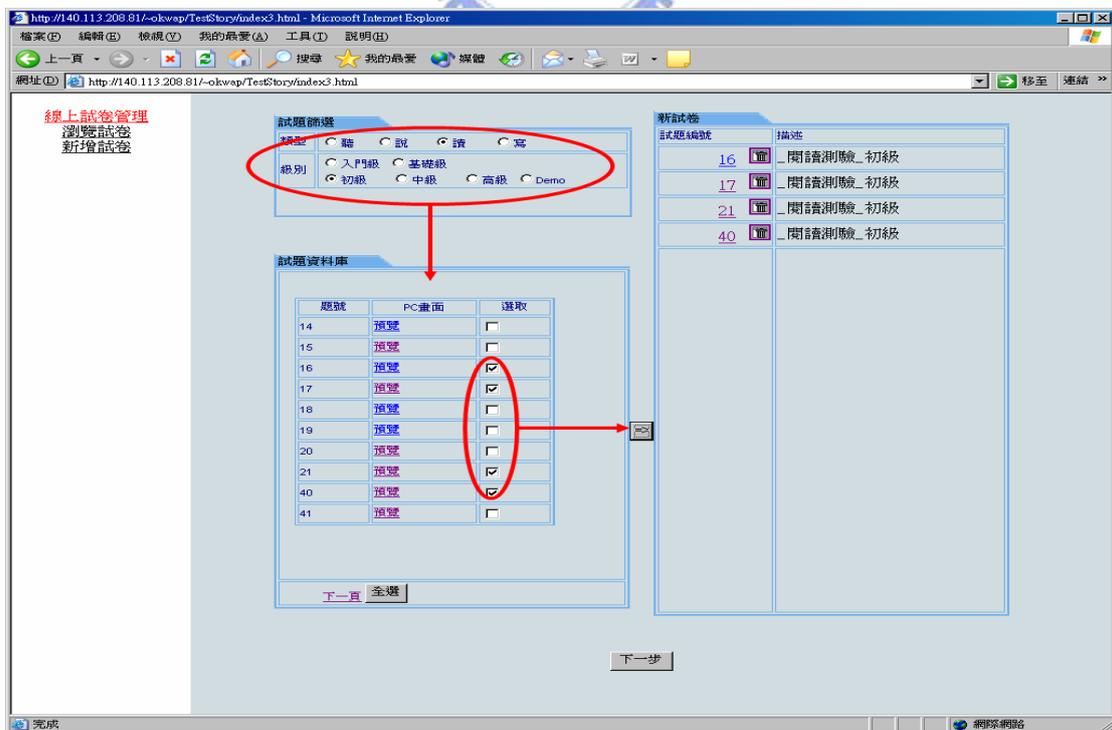


圖 44: 篩選資料庫試題並挑選試題成試卷

(4)在上一步編完一份試卷後，管理者可以按“下一步”按鈕就會進入轉換協調程序的四個步驟畫面，如圖 45。首先第一步是使用者必須設定一些重要轉換參數。

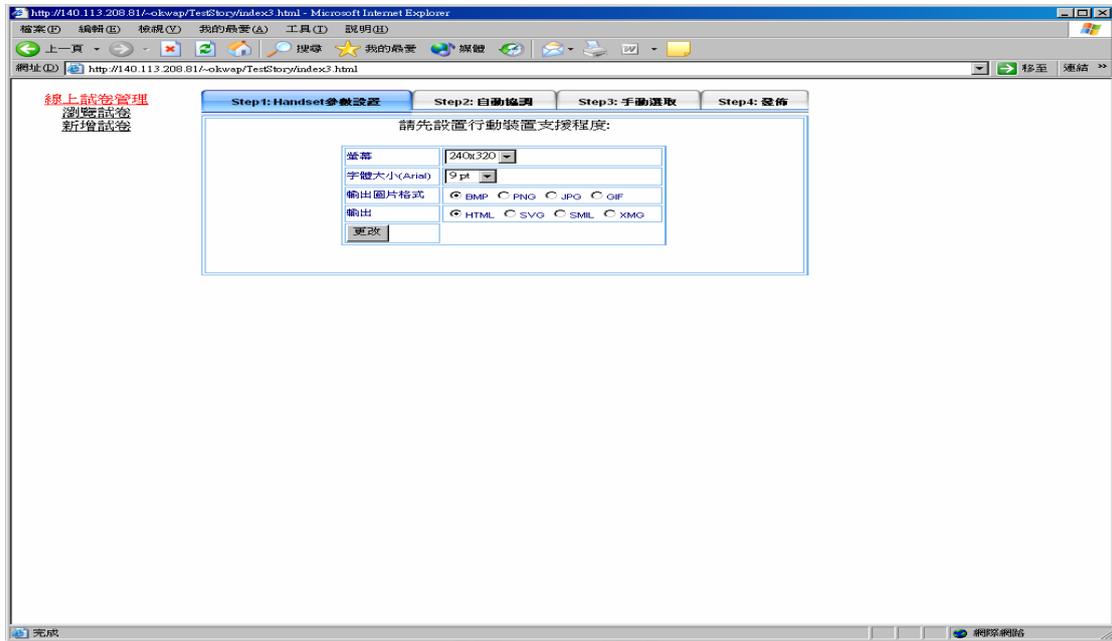


圖 45: 改編協調的第一步驟-參數設定

(5)進入批次改編協調畫面，使用者可以按選擇剛挑出來的試題並做批次協調的動作，如圖 46。

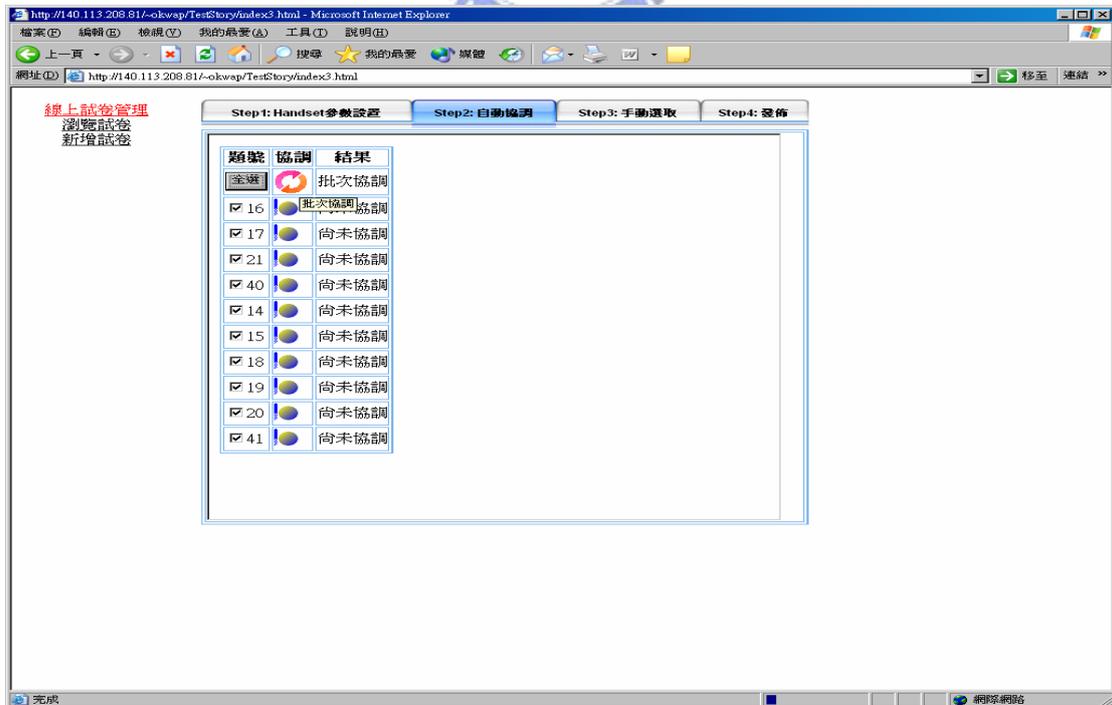


圖 46: 改編協調的第二步驟-自動批次轉換

(6) 進入手動選取板型畫面，並提供呈現預覽結果給使用者觀看，如圖 47。

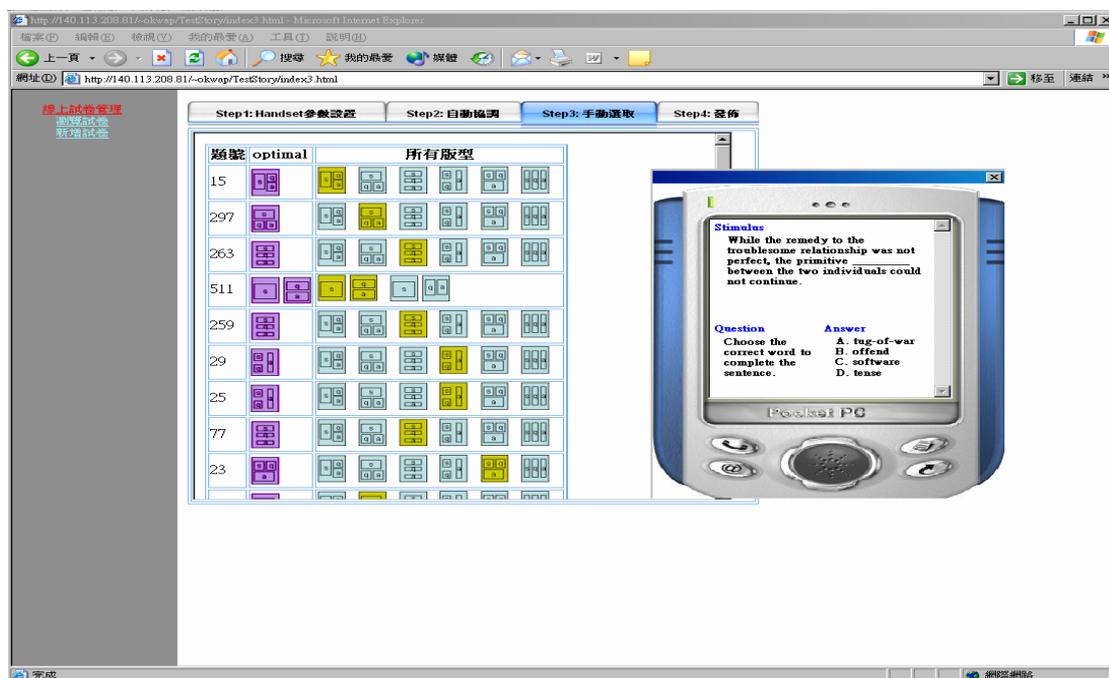


圖 47: 改編協調的第三步驟-手動調整選取

(7) 以上一步驟選完的最佳結果為基礎，將整份試卷的內容作轉換，轉換完畢並壓縮上傳至試題平台

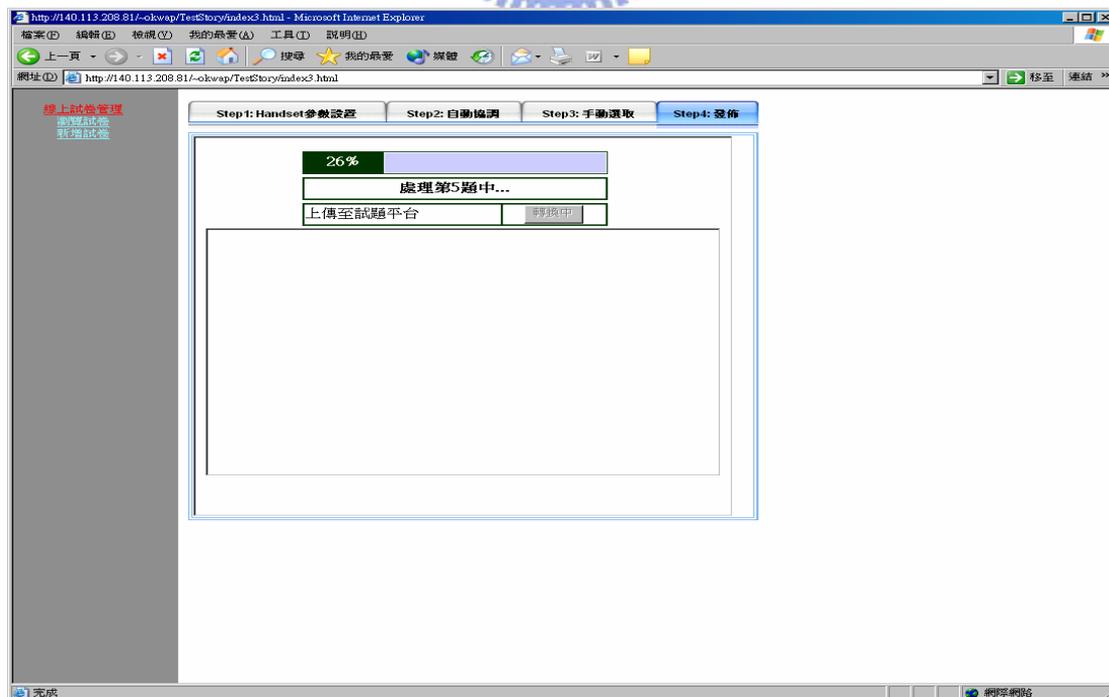


圖 48: 改編協調第四步驟-轉換並上傳至試題平台

(8)最後顯示上傳完成的訊息，必顯示上傳至平台的下載試卷畫面。

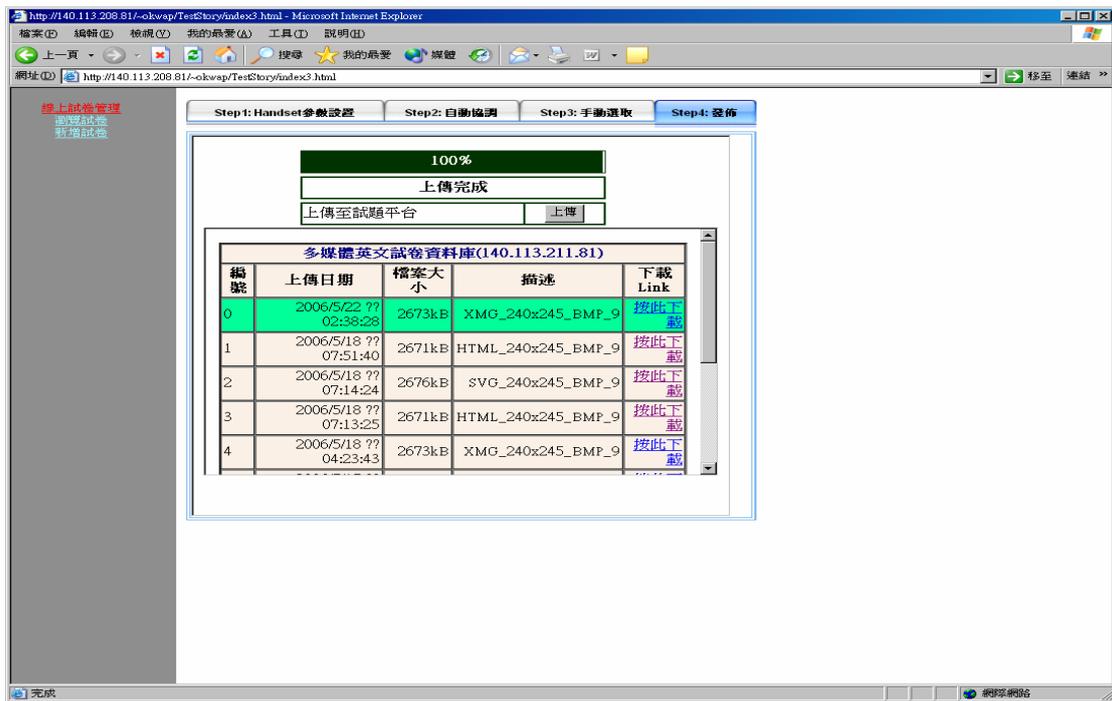


圖 49: 轉換上傳成功畫面



6.2 PDA 客戶端瀏覽裝置操作畫面

(1)使用者將以有效的帳號密碼登入，進入之後會成右圖所示

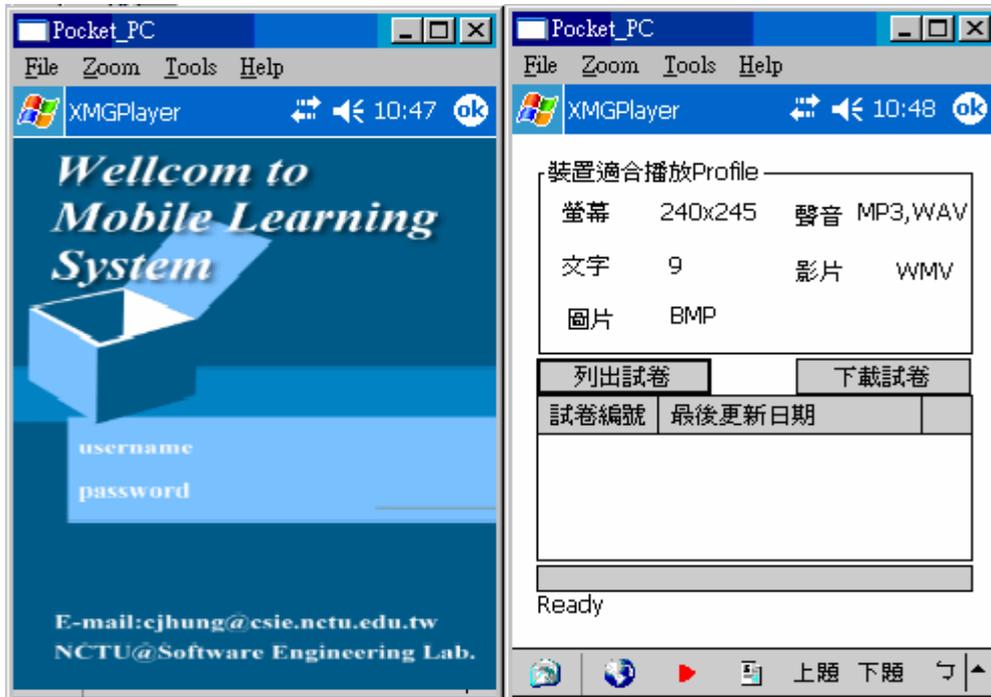


圖 50: PDA 試題播放器-登入畫面及偵測 device profile

(2)使用者要按「列出試卷」按鈕，而從 server 端回傳試卷列表。之後選擇某個試卷後，再按「下載試卷」按鈕，進行下載動作。

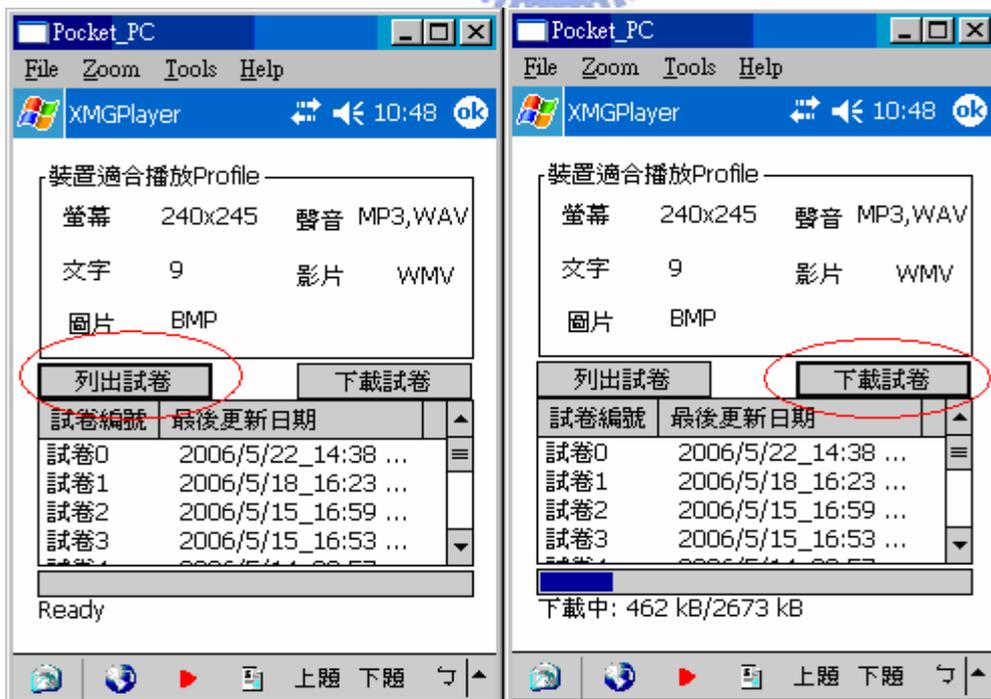


圖 51: PDA 試題播放器-列出試卷及下載試卷

(3) 下載完畢並經由程式自動解壓縮後，點選下方工具列「紅色播放按鈕」，便依照試卷的編排方式一題一題播放。

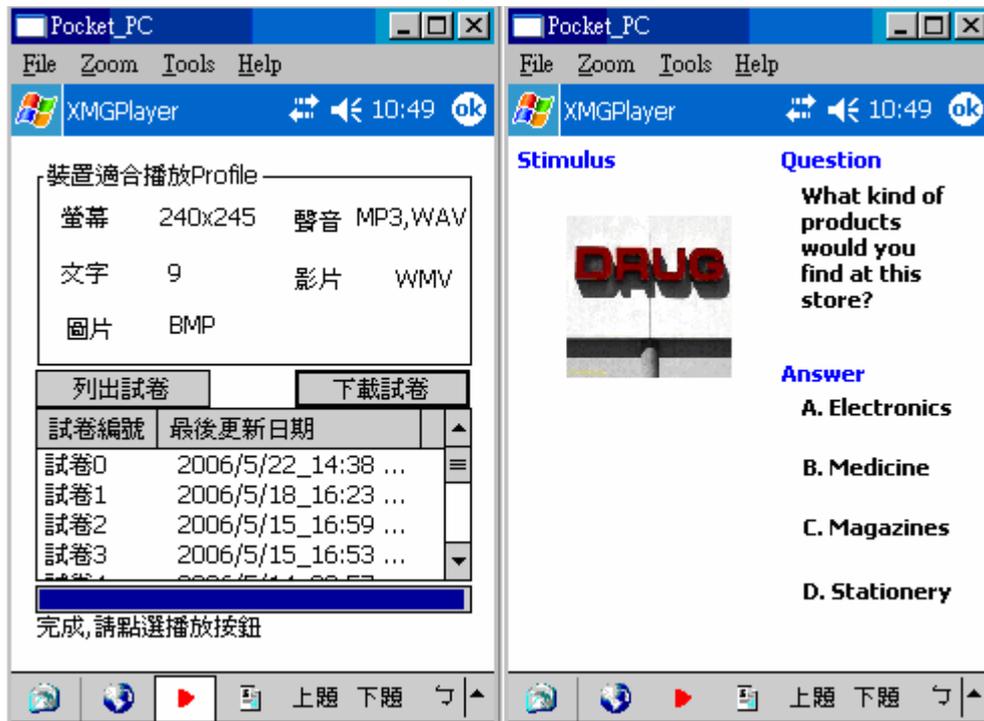


圖 52：PDA 試題播放器-下載試卷完成及播放試卷試題

(4) 點選下一題按鈕會進行下一題的播放。此外播放器也提供跳躍的選擇，使用者可以選則跳到其它題去觀看內容。

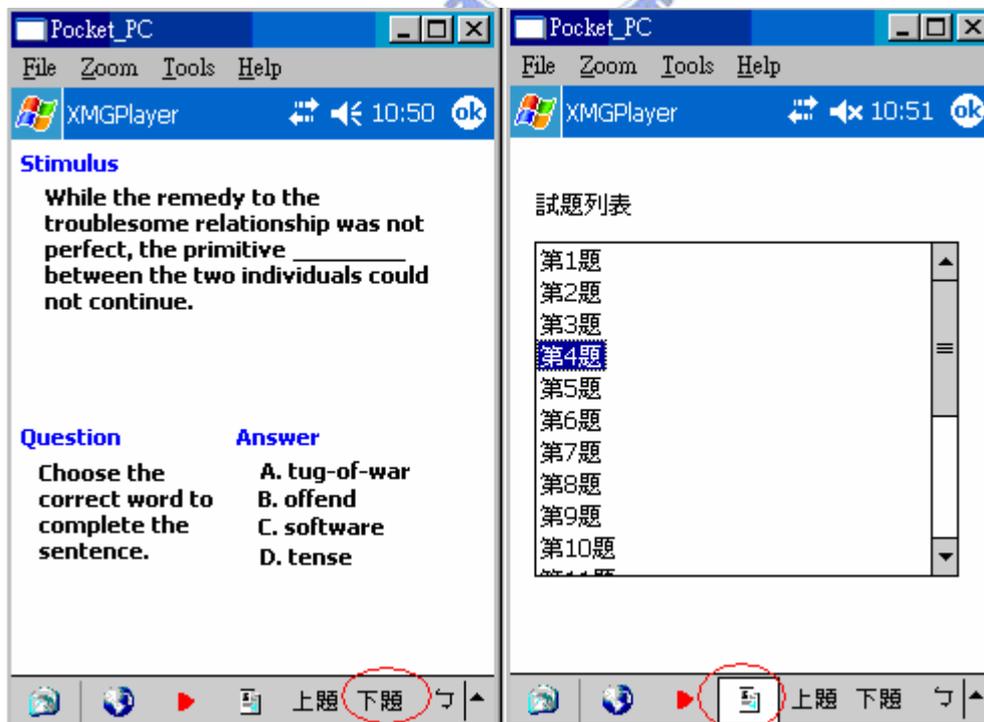


圖 53：PDA 試題播放器-播放試題及選擇題目

6.3 改編協調前後內容呈現差別

這裡我們以三個多媒體英文試題為範例，而轉換出去的對象以四種不同螢幕大小(320x240、240x320、240x240、160x180)的行動裝置來作比較，此外字形大小設定為 Arial 10px。

● 範例一：

1. 原來在 PC 呈現畫面(640x480)

Stimulus	Question
	Where is this taking place?
	Answer
	A. A chorus
	B. A court
	C. A charity
	D. A church

圖 54：範例一 640x480 呈現畫面

2. 320x240 螢幕大小呈現畫面

Stimulus	Question
	Where is this taking place?
	Answer
	A. A chorus
	B. A court
	C. A charity
	D. A church

圖 55：範例一 320x240 呈現畫面

3. 240x320 螢幕大小呈現畫面

<p>Stimulus</p> 	<p>Answer</p> <p>A. A chorus</p> <p>B. A court</p> <p>C. A charity</p> <p>D. A church</p>
<p>Question</p> <p>Where is this taking place?</p>	

圖 56：範例一 240x320 呈現畫面

4. 240x240 螢幕大小呈現畫面

<p>Stimulus</p>  <p style="text-align: right;">Next</p>	<p>Question</p> <p>Where is this taking place?</p> <p>Answer</p> <p>A. A chorus</p> <p>B. A court</p> <p>C. A charity</p> <p>D. A church</p> <p style="text-align: right;">Previous</p>
---	---

圖 57：範例一 240x240 呈現畫面

5. 160x180 螢幕大小呈現畫面

<p>Stimulus</p>  <p>Question</p> <p>Where is this taking place?</p> <p style="text-align: right;">Next</p>	<p>Answer</p> <p>A. A chorus</p> <p>B. A court</p> <p>C. A charity</p> <p>D. A church</p> <p style="text-align: right;">Previous</p>
---	---

圖 58：範例一 240x240 呈現畫面

● 範例二：

1. 原來在 PC 呈現畫面(640x480)

<p>Stimulus</p> 	<p>Question</p> <p>What can you do at this machine?</p> <p>Answer</p> <ul style="list-style-type: none">A. Make a phone call.B. Withdraw cash.C. Earn money.D. Send a postcard.
--	--

圖 59：範例二 640x480 呈現畫面

2. 320x240 螢幕大小呈現畫面

<p>Stimulus</p>  <p style="text-align: right;">Next</p>
<p>Question</p> <p>What can you do at this machine?</p> <p>Answer</p> <ul style="list-style-type: none">A. Make a phone call.B. Withdraw cash.C. Earn money.D. Send a postcard. <p style="text-align: right;">Previous</p>

圖 60：範例二 320x240 呈現畫面

3. 240x320 螢幕大小呈現畫面

<p>Stimulus</p> 	<p>Question</p> <p>What can you do at this machine?</p>
<p>Answer</p> <p>A. Make a phone call.</p> <p>B. Withdraw cash.</p> <p>C. Earn money.</p> <p>D. Send a postcard.</p>	

圖 61：範例二 240x320 呈現畫面

4. 240x240 螢幕大小呈現畫面

<p>Stimulus</p>  <p style="text-align: right;">Next</p>	<p>Question</p> <p>What can you do at this machine?</p> <p>Answer</p> <p>A. Make a phone call.</p> <p>B. Withdraw cash.</p> <p>C. Earn money.</p> <p>D. Send a postcard.</p> <p style="text-align: right;">Previous</p>
--	--

圖 62：範例二 240x240 呈現畫面

5. 160x180 螢幕大小呈現畫面

<p>Stimulus</p>  <p>Question</p> <p>What can you do at this machine?</p> <p style="text-align: right;">Next</p>	<p>Answer</p> <p>A. Make a phone call.</p> <p>B. Withdraw cash.</p> <p>C. Earn money.</p> <p>D. Send a postcard.</p> <p style="text-align: right;">Previous</p>
---	---

圖 63：範例二 160x180 呈現畫面

● 範例三：

1. 原來在 PC 呈現畫面(640x480)

Stimulus

Mountain climbing is a difficult and dangerous sport. A mountain climber needs to have knowledge and skills and must be in good shape. They must also know how to read maps, because they must find their way where there are no paths to follow. This type of activity can be dangerous for an untrained person. In fact, even many skilled climbers have lost their lives.

Question

Which of the following is NOT something a mountain climber needs?

Answer

- A. knowledge and skills
- B. ability to read maps
- C. paths to follow
- D. be in good shape

圖 64：範例三 640x480 呈現畫面

2. 320x240 螢幕大小呈現畫面

Stimulus

Mountain climbing is a difficult and dangerous sport. A mountain climber needs to have knowledge and skills and must be in good shape. They must also know how to read maps, because they must find their way where there are no paths to follow. This type of activity can be dangerous for an untrained person. In fact, even many skilled climbers have lost their lives.

[Next](#)

Question

Which of the following is NOT something a mountain climber needs?

Answer

- A. knowledge and skills
- B. ability to read maps
- C. paths to follow
- D. be in good shape

[Previous](#)

圖 65：範例三 320x240 呈現畫面

3. 240x320 螢幕大小呈現畫面

<p>Stimulus</p> <p>Mountain climbing is a difficult and dangerous sport. A mountain climber needs to have knowledge and skills and must be in good shape. They must also know how to read maps, because they must find their way where there are no paths to follow. This type of activity can be dangerous for an untrained person. In fact, even many skilled climbers have lost their lives.</p> <p style="text-align: right;">Next</p>	<p>Question</p> <p>Which of the following is NOT something a mountain climber needs?</p> <p>Answer</p> <p>A. knowledge and skills B. ability to read maps C. paths to follow D. be in good shape</p> <p style="text-align: right;">Previous</p>
---	---

圖 66：範例三 240x320 呈現畫面

4. 240x240 螢幕大小呈現畫面

<p>Stimulus</p> <p>Mountain climbing is a difficult and dangerous sport. A mountain climber needs to have knowledge and skills and must be in good shape. They must also know how to read maps, because they must find their way where there are no paths to follow. This type of activity can be dangerous for an</p>	<p>Question</p> <p>Which of the following is NOT something a mountain climber needs?</p> <p>Answer</p> <p>A. knowledge and skills B. ability to read maps C. paths to follow D. be in good shape</p> <p style="text-align: right;">Previous</p>
---	---

圖 67：範例三 240x240 呈現畫面

5. 160x180 螢幕大小呈現畫面

<p>Stimulus</p> <p>Mountain climbing is a difficult and dangerous sport. A mountain climber needs to have knowledge and skills and must be in good shape. They must also know</p>	<p>Question</p> <p>Which of the following is NOT something a mountain climber needs?</p> <p style="text-align: right;">Previous Next</p>	<p>Answer</p> <p>A. knowledge and skills B. ability to read map C. paths to follow D. be in good shape</p> <p style="text-align: right;">Prev</p>
--	--	--

圖 68：範例三 160x180 呈現畫面

6.4 靜態和動態轉換及混合方式轉換數據分析

本研究依據客戶端使用者來量測一般用靜態和動態所花費擷取多媒體內容的等待時間差異，及 server 端所花費磁碟空間差異。並和本研究提出的靜態和動態的方法作比較。

實驗數據使以本實驗室開發的樣板式多媒體試題為多媒體內容，我們以一份試卷 18 題的各樣的可能性組合來量測，也就是一份試卷裡面包含了多少題的聲音試題及不包含聲音的試題來組合。如下圖的時間比較所示，橫座標為一份試卷的組成成分，例如 4:14 代表這份試卷有 4 題包含聲音，14 題不包含聲音。下面是我們的分析結果：

- 花費時間差異：

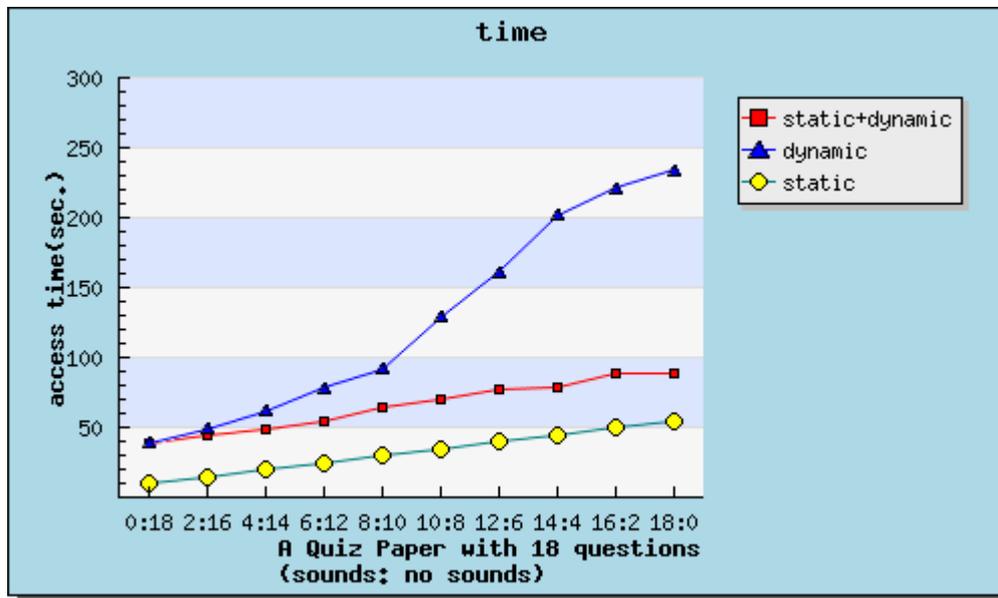


圖 69：三種模式時間差異圖

我們可以看到靜態動作僅需要花費下載時間，而對於動態來說則需要長時間的轉譯過程加上下載時間。而靜態及動態混合方式所花費的時間介於他們兩個之間，而且可以看到試卷如果由更多的聲音試題比例組成，靜態及動態混合方式和動態方式的時間差異越大。

- 磁碟空間使用量

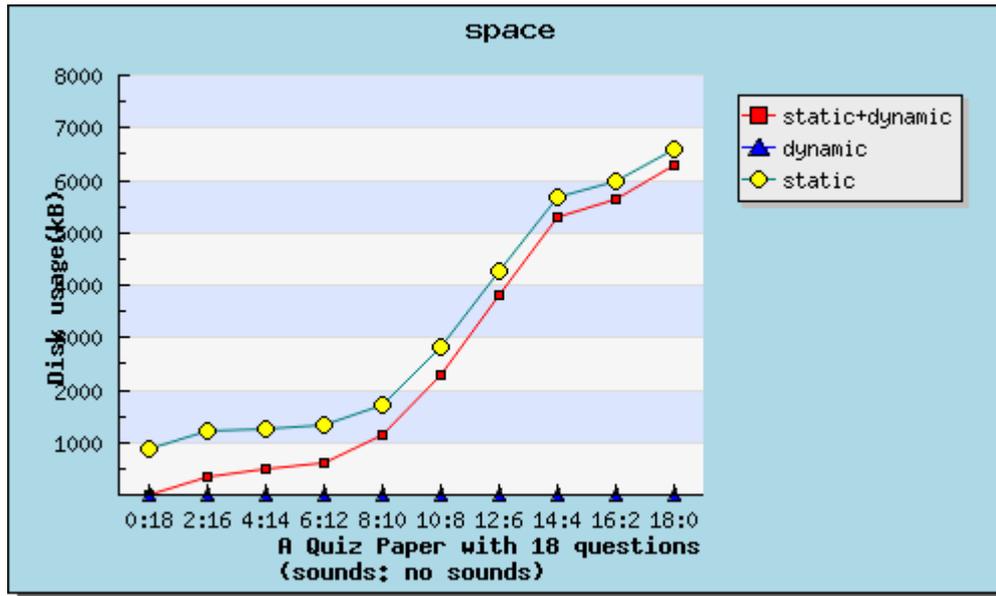


圖 70：三種模式磁碟空間使用量(同一種環境)

對於動態來說，由於所有的資料都會由動態來產生，因此它完全不需要花費任何額外磁碟的空間。而靜態的方式，因為它必須預先將轉譯好的多媒體內容存放在資料庫裡，所以須花費磁碟空間。而我們提出的靜態及動態混合方式花費的磁碟空間卻介於這兩者之間。不過由上圖所示，試卷由聲音試題組成成分越多，靜態及動態混合方式和靜態方式所花費的磁碟空間相差不大，這是因為聲音多媒體物件佔了整份多媒體內容的大小。不過這張圖是針對同一個行動裝置支援一種格式而言，如果有兩個異質的行動裝置(例如：螢幕大小不一樣、聲音支援程度相同)，靜態必須轉譯成兩份多媒體內容，然而對於靜態及動態混合方式來說，它儲存的僅需各式各樣的多媒體物件一份即可，剩下的描述檔、layout 協調、包裝皆在動態完成產生。由下圖可以明顯的看出靜態和靜態及動態混合方式的磁碟空間差異程度。

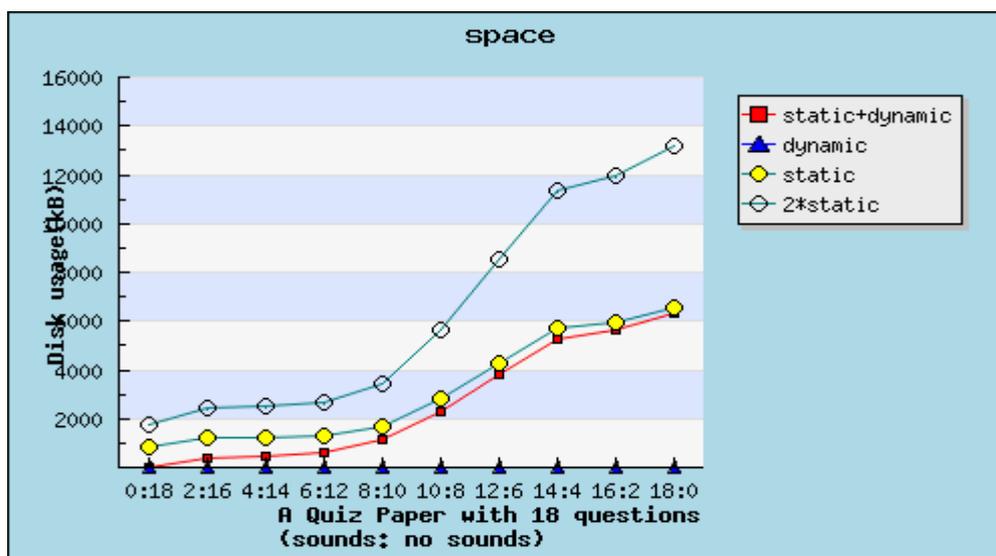


圖 71：三種模式磁碟空間使用量(不同環境)

七、結論

7.1 總結

本系統提出對於樣版式的多媒體內容在異質的行動裝置播放機制及呈現技巧。以樣板式多媒體英文試題為例子來開發改編協調伺服器(Adaptation server)，並提供兩個應用程式-Web Based 管理介面給內容改編管理者使用、以及 PDA 上的多媒體英文試題播放器，除了播放內容外並用以模擬靜態、動態、靜態及動態混合方式存取多媒體內容。

主要的特點：

(1)由於目前行動裝置上播放多媒體內容的需求量日以遽增，本研究設計的改編協調伺服器搭配 web-based 管理協調介面能幫助內容提供者(Content Provider)節省因異質行動裝置需額外的重新設計內容成本，除了提供自動化的協調轉換及預覽功能，並能將轉換好的內容發佈至特定平台。

(2)本研究對於樣板式多媒體內容在行動裝置上的呈現方式提供改編協調的機制，提出較佳化的分析演算過程。依據內容樣板式段落語意、異質行動裝置螢幕大小、字形大小、文字數目及各項的參數分析，並以內容清楚、不失原內容的語意及使用者容易閱覽的前提下將版面重新調整。

(3)基於大多數多媒體內容描述都是以 XML 格式來呈現，例如:HTML、SVG、SMIL，因此本改編協調伺服器設計的轉換出去的格式以 XML 為基準，並以 XSLT 設計轉換方式讓程式設計師容易擴充成其他 XML Based 的多媒體內容格式。

(4)此外從 6.4 節的數據分析，觀察到一般傳統的靜態和動態轉換存取多媒體內容的優缺點，我們綜合了他們的優缺點提供靜態及動態混合的轉換存取模式。混合方式相對於靜態方式來講節省了伺服器的磁碟空間使用量，混合方式相對於動態來講縮短了客戶端使用者的等待轉換時間。

7.2 未來發展方向

在此我們將提出未來可能發展的方向及尚未解決的問題。

(1) 本研究對於樣板式的多媒體內容的版面協調仍有其他可以考慮的因素當作參數而且演算法並非相當準確，例如以字數來決定頁數、以字數的開根號來分配比例等等，相信未來將有更精確的方法來分析。

(2) 目前轉譯方式是為特定的版型來作協調，也就是如果來源的版型有不一樣的話，將要改寫改編協調伺服器(Adaptation server)，未來可能會討論樣板型的改編協調器的產生器可能性，或是更模組化的設計改編協調伺服器，以節省改編協調伺服器的開發維護。

(3) 本論文提到的動態轉換存取內容，對於客戶端使用者來說仍需要等待長的轉換時間，除了多媒體物件的轉換外，原因在於每個試題轉換動作是以批次在同一台改編協調伺服器轉換，因此可以利用分散式架構來將轉換的過程時間降低。

(4) 限於目前 PDA 上的 Internet Explorer 僅支援部分的 CSS 及 JavaScript，所以部分的呈現方式無法如預期，例如動畫、媒體物件的位置尚未支援。因此本研究轉出 HTML 格式僅能在桌上型電腦上模擬呈現。此外 SVG、SMIL 此類的標準格式在 PDA 上的播放器，並沒有 Full Set 支援，因此呈現效果也不如預期，僅能期待未來的開發。

(5) 目前轉換的多媒體內容來源是樣板式多媒體英文試題，以後更可以往不一樣的內容來做延伸，例如樣板式多媒體教材的改編協調。

參考文獻或資料

- [1] Anind K. Dey , Gregory D. Abowd ,“Towards a Better Understanding of Context and Context-Awareness”, Technical Report GIT-GVU-99-22,1999
- [2] Augusto Celentano , Ombretta Gaggi ,“Context-aware design of adaptable multimodal documents”,2004
- [3] Narendra Shaha, Ashish Desai & Manish Parashar , “Multimedia Content Adaptation for QoS Management over Heterogeneous Networks”, proc of International Conference on Internet Computing 2001,USA,2001
- [4] Raimund Brandt, Christian Horstnagl ,Helmut Reiser ,“Dynamically Adaptable Mobile Agents for Scaleable Software and Service Management”, Journal of Communications and Networks,2001
- [5] Raimund Brandt ,Helmut Reiser, “Dynamic Adaptation of Mobile Agents in Heterogeneous Environments”, In Picco, G. P., editor, Mobile Agents: Fifth International Conference, MA 2001, Proceedings, (2230), Lecture Notes in Computer Science (LNCS), 70–87, Springer, Atlanta, Georgia, ,USA, Dezember, 2001.
- [6] Wai Yip Lum , Francis C.M. Lau ,“A Context-Aware Decision Engine for Content Adaptation”, IEEE Pervasive Computing,2002
- [7] Jérôme Euzenat, Nabil Layaïda, Victor Dias ,“A semantic framework for multimedia document adaptation”, In Proc. 18th International Joint Conference on Artificial Intelligence (IJCAI), Acapulco (MX), pages 31--36,2003
- [8] AUGUSTO CELENTANO ,OMBRETTA GAGGI, “TEMPLATE-BASED GENERATION OF MULTIMEDIA PRESENTATIONS”, International Journal of Software Engineering and Knowledge Engineering ,2003
- [9] Tayeb Lemlouma , Nabil Layaïda ,“Context-Aware Adaptation for Mobile Devices”, In IEEE Int. Conf. on Mobile Data Management, Berkeley, CA,USA,2004

- [10] David Gotz Ketan Mayer-Patel , "A General Framework for Multidimensional Adaptation", ACM Multimedia, USA, 2004
- [11] Kang Zhang, Jun Kong, Meikang Qiu, Guang-Lei Song, "Multimedia layout adaptation through grammatical specifications", ACM/Springer Multimedia Systems, 2005
- [12] Robson Eisinger , Rudinei Goularte , "Towards Context based Dynamic Video Adaptation", ACM International Conference Proceeding Series; Vol. 125, USA, 2005
- [13] Wai Yip Lum , Francis C.M. Lau , "On Balancing Between Transcoding Overhead and Spatial Consumption in Content", In Proc. of ACM Mobicom, 2002
- [14] Zhijun Lei , Nicolas D. Georganas , "Context-based Media Adaptation in Pervasive Computing", in Proc. of Canadian Conference on Electrical and Computer Engineering, 2001
- [15] HyperText Markup Language(HTML),
<http://www.w3.org/MarkUp>
- [16] Scalable Vector Graphics(SVG),
<http://www.w3.org/Graphics/SVG/>
- [17] Synchronized Multimedia Integration Language(SMIL),
<http://www.w3.org/AudioVideo/>
- [18] 智勝國際 , <http://www.caidiy.com/caidefault.htm>
- [19] FLASH, <http://www.adobe.com/>
- [20] MPEG4,
<http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm>
- [21] Lame, <http://lame.sourceforge.net/>
- [22] DevIL, <http://openil.sourceforge.net/>
- [23] Microsoft® Windows Media™ Encoder,
<http://www.microsoft.com/taiwan/windows/windowsmedia/wm7/encoder.htm>
- [24] XSLT, <http://www.w3.org/TR/xslt>
- [25] Compiler TOOL-LEX, YACC, <http://dinosaur.compilertools.net/>

- [26] Audio Library - Fmod, <http://www.fmod.org/>
- [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”,交大碩士論文,2005
- [28] 維基百科, <http://zh.wikipedia.org/>
- [29] Mozilla, <http://www.moztw.org/>
- [30] Internet Explorer, <http://www.microsoft.com>
- [31] Netscape, <http://www.netscape.com>
- [32] Pocketsmil2.0, <http://wam.inrialpes.fr/software/pocketsmil/>
- [33] eSvg, <http://esvg.ultimodule.com>

