

國立臺灣師範大學
資訊工程研究所碩士論文

指導教授：葉耀明 博士

語言學習之影音索引技術研究

Research on Video Indexing of Language Learning

研究生：栢慶瑞 撰

中華民國 101 年 06 月

摘要

語言學習之影音索引技術研究

栢慶瑞

近年來網路的使用已成為人們生活中重要的一部份，由於網路頻寬的改善，及多媒體技術的蓬勃發展，在應用上更為多元。線上分享影音資訊已成為網路使用者之間溝通的重要橋梁，這些被分享的多媒體資訊已不僅限於娛樂、生活相關的內容。在此趨勢下，有許多教育單位將錄製的教材發布至網路上，提供網路使用者透過影片的方式學習。雖然在網路上影片取得容易，但是若想觀看影片中的特定段落，則需花費時間搜尋，因此本研究的主要探討議題為實現影音資料庫索引系統，利用索引技術增加學習影片內容的搜尋效率及準確度以及使用網頁技術製作出高互動性的瀏覽系統，提供學習者一個良好的學習環境。

本研究主要以 Silverlight、JSP、XML、eXist 等技術實現影音資料庫索引系統。我們針對教學用的多媒體材料影音呈現問題進行改善，對影片內容作分析，找出影片中的各個主題，依照主題分成數個段落，並為各主題擷取關鍵畫面作為縮圖並建立索引。接著我們以 Silverlight 網頁技術建構出簡易操作的使用者介面，搭配 XML 的結構儲存影片索引資訊，並以 eXist 資料庫有效管理 XML 文件。最後，結合 JSP 語言作影片索引搜尋的動作，不只可以同時搜尋到多部完整的影片內容，也可利用關鍵字搜尋影片片段，使查詢影片段落更快速，搭配高互動性網頁瀏覽系統呈現，增加學習效率。

(關鍵詞: 影音索引、語言學習、可延伸標記語言)

ABSTRACT

Research on Video Indexing of Language Learning

by

Ching-Ruei Bai

Internet has become an important part of people's life in recent years. Due to the improvement of the network bandwidth and the vigorous development of multimedia technology, there are more diverse applications related to online sharing audio and video contents. Recently, many educational institutions post recorded video materials to networks for users to learn through internet. However, it's easy to acquire language learning materials through internet, but one still need to spend lots of time searching in video if he wants to watch a specific video segment. Therefore, this study focuses on issues for video indexing, develops a video content indexing system which aims at providing a good learning environment that has an efficient searching for video segments (shots) and an interactive user interface.

In this thesis, we use Silverlight, JSP, XML and eXist to develop a Video Indexing System. We improve the multimedia teaching material problem, analysis the video content to identify the different themes in the film, and capture key frames for each shot as a thumbnail. Then we construct the user interface by Microsoft Silverlight , store the video segment information by XML format managed in eXist database. Finally, querying the index by JSP language, not only can search multiple video contents simultaneously, but also can provide video segments to be searched by keyword.

(keywords: Video indexing 、 Language learning 、 XML)

誌 謝

兩年的碩士生涯即將結束，經過兩年的努力和學習，終於完成了本論文，在這兩年研究所的日子裡，非常感謝我的指導教授 葉耀明老師，教授自由開放的教學風格，讓我學習到獨立思考的能力，使我成長了不少。感謝老師總會適時的指引我的研究方向、讓我參與計畫，不只學習到了專業知識，也學到了做人處事的態度及溝通的能力，使我在研究的道路上有很大的幫助。

感謝楊明正、葉慶隆教授們擔任我的口試委員，百忙中抽空審核我的論文，並提供許多寶貴的意見，指出我沒注意到的部分，使我的研究更完整。同時也感謝研究所中多位教授的教導，使我的學業更為精進。

感謝實驗室同學永倫、芝華、孝倫、明憲、冠緯、瀧濱、惠迪的陪伴與幫助，讓我的研究生活不枯燥乏味，有了大家的陪伴、一起修課、一起寫作業，使我的學習路程不孤單，感謝你們讓我的研究所生活能夠有這麼多麼美好回憶；感謝學弟們對實驗室計畫的協助，使我能夠安心地專心寫論文。

最後，感謝家人的支持，讓我能後顧之憂地完成碩士學位，感謝好朋友們在我課業繁重的時候一起玩樂，放鬆了我的心情，使我更有動力完成我的研究，在此獻上我最誠摯的感謝，謝謝。

栢慶瑞 謹誌

國立臺灣師範大學

資訊工程研究所

民國 101 年 6 月

目錄

| | |
|--|-----|
| 圖目錄..... | iv |
| 表目錄..... | vi |
| 程式碼目錄..... | vii |
| 第一章 緒論..... | 1 |
| 1.1 研究背景與動機..... | 1 |
| 1.2 研究目的..... | 2 |
| 1.3 論文架構..... | 3 |
| 第二章 文獻探討..... | 4 |
| 2.1 Microsoft Silverlight 應用程式開發平台..... | 4 |
| 2.1.1 Microsoft Silverlight 特色與介紹..... | 4 |
| 2.1.2 Microsoft Silverlight 架構..... | 5 |
| 2.1.3 Microsoft Silverlight 優點..... | 6 |
| 2.1.4 Silverlight 專案開發方式..... | 7 |
| 2.2 XAML 可延伸應用程式標記語言..... | 8 |
| 2.2.1 XAML 介紹..... | 8 |
| 2.2.2 XAML 與 Silverlight..... | 9 |
| 2.2.3 XAML 語法與物件屬性..... | 10 |
| 2.3 XML 及 eXist 原生資料庫..... | 12 |
| 2.3.1 XML 簡介..... | 12 |
| 2.3.2 eXist 原生資料庫簡介..... | 14 |
| 2.4 OpenCV(Open Source Computer Vision)..... | 16 |
| 2.4.1 OpenCV 簡介..... | 16 |
| 2.4.2 OpenCV 可應用領域..... | 16 |
| 第三章 影音資料庫之索引系統規劃..... | 19 |
| 3.1 影音資料庫之索引系統架構..... | 19 |
| 3.2 自動與手動索引模組..... | 21 |
| 3.2.1 影片組成要素..... | 21 |
| 3.2.2 索引原理與流程..... | 24 |
| 3.2.3 影像處理方法..... | 26 |
| 3.3 索引查詢..... | 30 |
| 3.3.1 XQuery 簡介..... | 30 |
| 3.3.2 XQuery 語法..... | 31 |
| 3.3.3 XQuery 查詢流程與方法..... | 33 |
| 3.4 Silverlight 影片瀏覽與同步字幕與標籤內容修改..... | 34 |
| 3.4.1 Silverlight 影片瀏覽架構..... | 35 |
| 3.4.2 Silverlight 瀏覽介面..... | 36 |

| | |
|--------------------------------------|----|
| 3.4.3 Silverlight 模組內部技術..... | 38 |
| 第四章 系統實現..... | 46 |
| 4.1 系統架構與環境建置..... | 46 |
| 4.2 系統操作..... | 48 |
| 4.2.1 使用案例圖 (Use Case Diagram) | 48 |
| 4.2.2 循序圖(Sequence Diagram) | 50 |
| 4.3 系統呈現..... | 53 |
| 4.3.1 影片上傳與自動化場景判斷..... | 53 |
| 4.3.2 影片查詢及章節呈現..... | 54 |
| 4.3.3 影片索引..... | 56 |
| 4.3.4 字幕同步..... | 57 |
| 4.3.5 章節詳細資訊及標籤修改..... | 58 |
| 4.3.6 影片索引查詢..... | 59 |
| 第五章 結論與未來發展..... | 60 |
| 5.1 結論..... | 60 |
| 5.2 未來發展..... | 61 |
| 參考文獻..... | 63 |

圖目錄

| | |
|---|----|
| 圖 2-1: Silverlight 架構圖 | 6 |
| 圖 2-2: XAML 以提示的方式設定屬性值..... | 12 |
| 圖 2-3: eXise 與 RDBMS 儲存文件意象圖 | 15 |
| 圖 3-1: 影音資料庫新舊系統功能比較圖 | 19 |
| 圖 3-2: 影音資料庫索引系統概念圖 | 20 |
| 圖 3-3: 解析度為 640*480 圖片 | 22 |
| 圖 3-4: 解析度為 160*120 圖片放大圖 | 23 |
| 圖 3-5: 選擇代表性縮圖流程圖 | 26 |
| 圖 3-6: 資料格式轉換圖 | 27 |
| 圖 3-7: 彩色影像灰階化結果 | 28 |
| 圖 3-8: 差異小的框架灰階值相減結果 | 29 |
| 圖 3-9: 差異大的框架灰階值相減結果 | 29 |
| 圖 3-10: FLWOR 運算式特定順序組合 | 32 |
| 圖 3-11: XQuery 查詢流程 | 34 |
| 圖 3-12: 影片瀏覽架構 | 35 |
| 圖 3-13: Silverlight 影片瀏覽介面 | 36 |
| 圖 3-14: Silverlight 影片章節瀏覽介面 | 37 |
| 圖 3-15: 影片關鍵畫面縮圖呈現示意圖 | 39 |
| 圖 3-16: WrapPanel 排列狀況..... | 42 |
| 圖 3-17: StackPanel 排列狀況 | 42 |
| 圖 3-18: Silverlight 存取 eXist 資料庫流程圖 | 43 |
| 圖 3-19: 手動索引介面 | 45 |
| 圖 4-1: 系統架構圖 | 46 |
| 圖 4-2: 使用案例圖 | 48 |
| 圖 4-3: 影片上傳循序圖 | 51 |
| 圖 4-4: 影片查詢循序圖 | 51 |
| 圖 4-5: 影片索引查詢循序圖 | 52 |
| 圖 4-6: 修改章節資訊標籤循序圖 | 52 |
| 圖 4-7: 字幕同步循序圖 | 53 |
| 圖 4-8: 系統影片上傳畫面 | 53 |
| 圖 4-9: 系統影片場景偵測結果示意圖 | 54 |
| 圖 4-10: 系統影片類別查詢畫面 | 55 |
| 圖 4-11: 系統影片類別查詢結果畫面 | 55 |
| 圖 4-12: 系統影片章節呈現畫面 | 56 |

| | |
|----------------------------|----|
| 圖 4-13:系統手動加入索引按鈕 | 57 |
| 圖 4-14:系統手動索引畫面 | 57 |
| 圖 4-15:系統字幕同步畫面 | 58 |
| 圖 4-16:系統章節資訊及標籤修改畫面 | 59 |
| 圖 4-17:系統影片索引查詢結果畫面 | 59 |

表目錄

| | |
|-------------------------------------|----|
| 表 2-1: Silverlight 與 Flash 比較表..... | 5 |
| 表 4-1:系統環境建置表 | 47 |
| 表 4-2:循序圖包含元素表 | 50 |

程式碼目錄

| | |
|---|----|
| 程式碼 3-1:XQuery 依照關鍵字查詢 | 34 |
| 程式碼 3-2:XQuery 依照關鍵字查詢結果 | 34 |
| 程式碼 3-3:在 XAML 中建立資料綁定(一):定義來源物件..... | 40 |
| 程式碼 3-4:在 XAML 中建立資料綁定(二):建立來源物件命名空間參考 | 40 |
| 程式碼 3-5:在 XAML 中建立資料綁定(三):在 Resource section 中建立來 源物件..... | 41 |
| 程式碼 3-6: 在 XAML 中建立資料綁定(四):綁定到物件..... | 41 |
| 程式碼 3-7:StackPanel 控制項範例 | 42 |
| 程式碼 3-8: Silverlight 讀取 XML 文件及元素的方法 | 44 |

第一章 緒論

1.1 研究背景與動機

近年來隨著資訊科技發展逐漸成熟，網際網路和電腦提供許多快速且便利多樣的服務，使得人們的日常生活跟電腦、手機等 3C 科技產品有了密不可分的關係，電腦在人們的工作、學習、休閒娛樂上已成為不可或缺的工具。在語言學習方面，傳統上都是以書本為主，但是隨著網路和多媒體資料被廣泛的應用，語言學習已經不僅限於傳統的書本瀏覽方式，在資訊科技的快速發展下，電腦輔助語言學習已成為流行，網路系統提供了語言學習更廣闊的空間，提供一個生動的多媒體應用學習環境成為一種新的學習模式，此種模式提供更活潑的內容來加強使用者對語言學習的興趣。

由於網路環境發達，只要有一台可上網的裝置，任何人都可以隨時分享及觀賞影片於網路平台上，在網路上提供教學影片給學員複習及學習已成為各個教育單位的趨勢。雖然在網路上瀏覽教學影片已變得非常方便，但若只想觀看特定的片段，則需要浪費長時間的搜尋等待時間，因此，若可以對影片內容作分析，找出影片內的各個主題段落，並擷取關鍵畫面作為縮圖，建立影片索引，即可快速瀏覽影片內容，或者使用關鍵字搜尋影片索引的段落，可大大減少搜尋等待時間，增加學習的品質。

在影音內容的瀏覽方式上，改善目前使用者觀看的方式，多年以來的觀看模

式主要是被動的，一段影片需從頭看到尾，無法指定或搜尋到影片中的部分段落，有了此系統的輔助下，觀看者可更快速準確地搜尋到影片相關的段落內容，並且提供字幕提示重點等功能，增加使用者跟影音資料之間的互動性，使用者透過此影音系統的索引和互動功能可加強學習效果。

1.2 研究目的

本研究針對影音資料學習類型影片，容易造成找不到重點片段以及影片內容過於冗長而枯燥乏味等問題進行改善，發展一套具有搜尋、索引、互動等特性的影音資料庫平台。目前有些影片本身沒有章節功能，使用者如果想要指定影片中某個段落來觀看需從頭開始慢慢觀看才能找到所需的部分，耗費時間長且下次要再觀看同一段落需重新再找一次，對於沒有章節功能的影片，本系統提供索引功能來改善這個缺點，另外增加了使用者和影片之間的互動性，使用微軟所開發的 Silverlight 網頁技術作為使用者介面，以 C#、XAML、JSP 等技術達成影片的索引和互動，建立一個管理者方便管理，使用者簡易使用的操作介面系統。

系統中除了原本提供的各類影片，廣告、影集等娛樂取向的部分之外，新增加了語言學習影片的類型，並增加索引和互動來增加使用者學習的興趣和效率，讓此系統不只在休閒娛樂方面有所發揮，也朝向教學應用發展的方向前進。

1.3 論文架構

本論文一共分成五個章節

第一章 緒論：

介紹研究背景與動機、研究目的、論文架構。

第二章 文獻探討：

介紹研究進行時，所需相關知識或參考內容。包括微軟 Silverlight 應用程式開發平台、XAML 可延伸應用程式標記語言、XML 延伸標記語言、eXist 原生資料庫、OpenCV。

第三章 影音資料庫之索引系統規劃：

系統架構規劃，包含各個模組分析與方法。

第四章 系統實現：

詳細敘述環境建置、系統架構實現、系統操作概念、系統呈現、系統使用分析。

第五章 結論與未來發展：

總結系統特色及成果，以及探討未來發展性。

第二章 文獻探討

2.1 Microsoft Silverlight 應用程式開發平台

2.1.1 Microsoft Silverlight 特色與介紹

Microsoft Silverlight 是 2007 年微軟所發展專為網頁設計的跨平台、支援多種瀏覽器的外掛互動應用程式，提供方便的程式設計模組支援 C#、VB、AJAX 等技術，基於 .NET Framework 的架構下提供新一代的網路媒體服務以及豐富互動式應用程式(RIA)，在使用者方面，以瀏覽器安裝外掛的方式提供多媒體應用程式服務，此種服務的方式跟 Adobe Flash 一樣，因此 Silverlight 與 Adobe Flash 常常被拿來做 WEB 技術的競爭，比較表為表 2-1 所示。此外 Silverlight 統合了伺服器、網頁、管理程式碼和動態語言，以及 Windows Presentation Foundation (WPF)的功能，同時也整合並相容目前所有的網頁技術，提供快速高品質的影像訊號傳輸與播放。

Silverlight 是微軟公司使用者介面技術(WPF)和可延伸應用程式標記語言(XAML)的一個子集合，提供多媒體顯示和使用向量式繪圖的動態圖像支援，程式設計師可以宣告式開發的方式(Declaring development)發展應用程式，此開發方式使他能與設計工具整合，所以微軟提供 Microsoft Expression Blend 工具給設計人員開發 Silverlight 應用程式。

Silverlight 支援多媒體串流的服務，特別是 Windows 媒體的 Streaming Media

和 MP3 音效串流，基於此項特色可以讓一般企業及開發者使用 Silverlight 發展更完整的多媒體串流應用程式，對於使用者來說就是可以透過網路以及各種裝置享受到更為精緻的使用者介面和豐富的多媒體內容。

| | 動畫處理 方式 | 影片和 音效 | 支援圖片 格式 | 多媒體 串流 | 檔案大 小 | Webcam 支援 |
|-------------|------------|-----------|------------|-----------|----------|--------------|
| Silverlight | 較優 | 較優 | | 較優 | 較優 | |
| Flash | | | 較優 | | | 較優 |

表 2-1: Silverlight 與 Flash 比較表

2.1.2 Microsoft Silverlight 架構

Silverlight 以外掛元件的方式安裝並提供服務，主要功用是提供使用者更好的多媒體體驗。Silverlight 是架構在瀏覽器上的一個執行環境，使用者可以透過這樣的環境來取得各式各樣的多媒體內容，主要目的是整合應用城市中的多媒體、圖像的功能和加強與使用者之間的互動。在沒有安裝 Silverlight 的情況下，如果要存取 Silverlight 的應用程式，瀏覽器會以安裝外掛的方式來提醒使用者安裝，這種安裝方式跟 Flash Player 一樣，而且安裝完成後平常並不會明顯感覺到它的存在，但是你可以享受到各式各樣以 Silverlight 技術製作出來的應用程式內容和多媒體服務，而且這些內容的表現品質會比 Flash 來的好。

.NET Framework 透過 CLR(Common Language Runtime)這種通用語言執行

平台來涵蓋 C#、VB.NET、C++ 等程式語言，經由 CLR Execution Engine 把網路上各種開發技術發展出來的應用程式整合，因此無論哪種技術開發出來的內容，透過 Silverlight 的架構下都可以相容和執行，所以不論使用者使用的是哪一種作業系統或是瀏覽器都一樣，也就是說有良好的相容性以及取得性。

圖 2-1 為 Silverlight 架構圖：

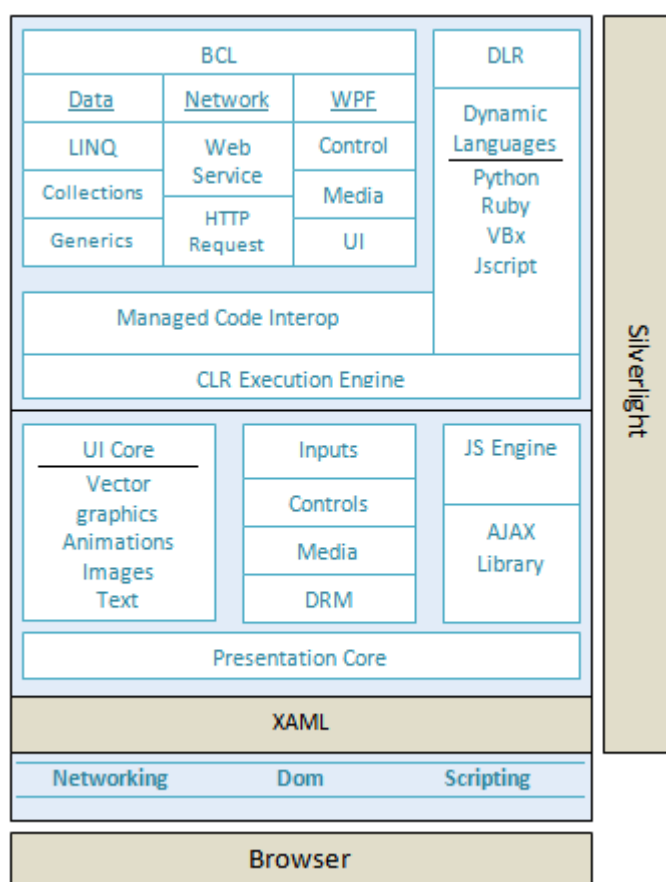


圖 2-1: Silverlight 架構圖

2.1.3 Microsoft Silverlight 優點

- 是一種支援多種瀏覽器、跨平台的開發技術。可以在所有常見的網頁瀏覽器上執行，支援的瀏覽器包括 Apple Safari、Mozilla Firefox、Microsoft

Internet Explorer 和 Google Chrome，平台包括 Microsoft Windows 和 Apple Mac OS。

- 下載快速且所占空間少。
- 以資料流的方式處理多媒體資訊。可以調整視訊品質以配合大小不同的螢幕或裝置。
- 提高互動性，使用者可以直接在瀏覽器中做拖曳、翻轉及縮放圖形的動作。
- 較高的保密性，在 Silverlight 應用程式按下滑鼠右鍵不能檢視原始程式碼，對於程式設計人員有較大的保障。
- 播放器的展現能力多樣化，製作 2D、3D 效果不需再以傳統的非線性軟體剪輯方式，Silverlight 提供更方便的製作工具，讓不會程式設計的人員也可在 Silverlight 上製作影像特效。
- 控制元件運用更彈性，Silverlight 的控制元件(Button, TextBox, TextBlock, ListBox 等)可與影像重疊，可增加跟使用者之間豐富的互動，也可以任意拖拉到任意位置或是隱藏。

2.1.4 Silverlight 專案開發方式

Silverlight 的開發工具分為兩個部份：

- 介面設計人員：使用 Microsoft Expression Blend 來進行設計。
主要用來設計介面外觀，可建立圖形、動畫、按鈕、色彩等。

- 程式開發人員：使用 Microsoft Visual Studio 來開發。

本研究以 Microsoft Visual Studio2010 做設計開發。在 Microsoft Visual Studio2010 開發的 Silverlight 專案檔也可以在 Microsoft Expression Blend 開啟，如此可讓介面設計人員和程式開發人員以不同工具合作共同完成同一個專案，達到彼此分工合作的關係。

2.2 XAML 可延伸應用程式標記語言

2.2.1 XAML 介紹

XAML(eXtensible Application Markup Language)為可延伸應用程式標記語言，是 Microsoft 針對 WPF 和 Silverlight 架構所開發的客製化語言，主要是對應用程式外觀、使用者介面、以及事件驅動做設計的標記語言，擁有跟超文件標示語言 (HTML) 一樣的網頁外觀設計的功能，以及可延伸標記語言 (XML) 的嚴謹性。

XAML 是基於 .NET Framework 的架構的標記語言，使用程式碼分離 (code-behind) 的形式讓程式設計師在撰寫程式時，不用再對使用者介面的可視物件做宣告的動作，可更專注於程式邏輯的設計，也可在撰寫程式時使用物件上定義過的屬性，也就是說透過 XAML 設計應用程式介面，可以直接表示為「建立物件」，這一個特點跟其他標記語言不同，是其他標記語言無法做到的。

因此 Silverlight 專案設計師可以透過 XAML 來設計系統外觀、使用者介面

和物件的視覺效果等，或者也可以以視覺化的方式產生 XAML 程式碼，如圖 2-2 所示。

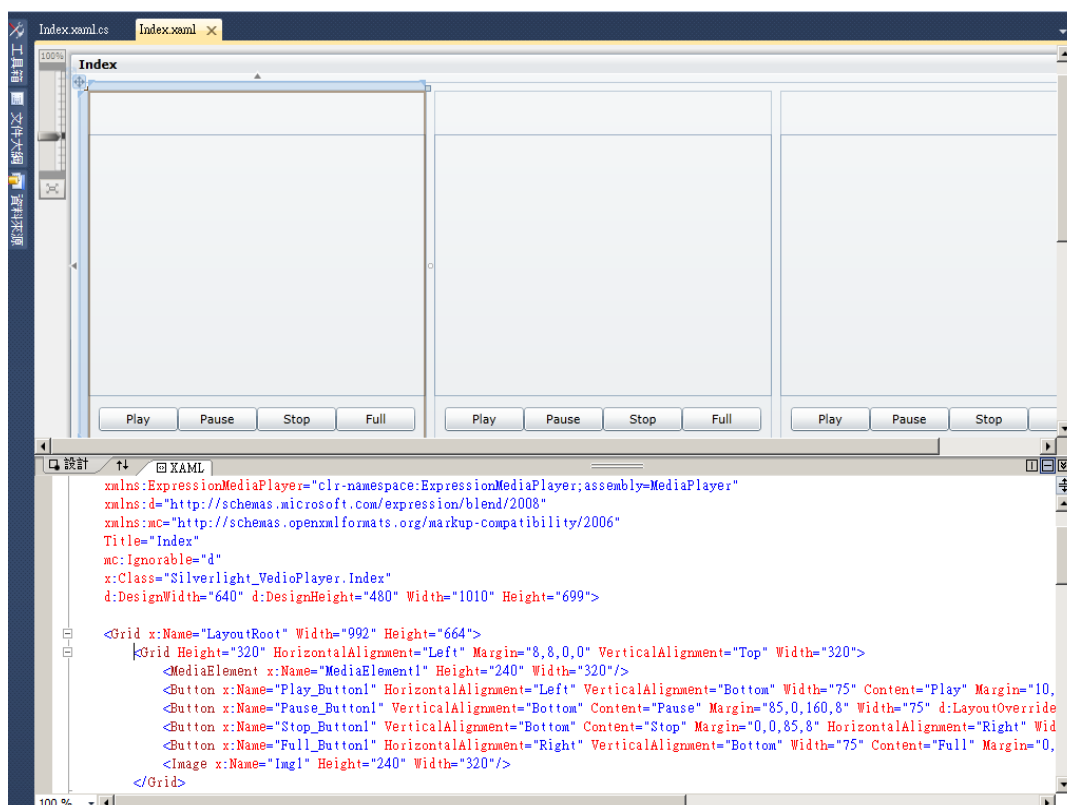


圖 2.2: XAML 以視覺化方式產生程式碼

2.2.2 XAML 與 Silverlight

在建立 Silverlight 程式時，可以使用 C#、Visual Basic 等 .NET Framework 支援的程式語言，搭配相關開發工具 Visual Studio 2010 以及 Expression Blend 來設計開發 Silverlight 應用程式，使用者可以使用 HTML 和 WPF 項目的網頁建立 Silverlight，就像 HTML 一樣，XAML 可以使用宣告式的語法來為 Web 應用程式建立使用者介面，兩個語言的差異在於 XAML 有提供更強大更完整的項目。

Silverlight 應用程式的使用者介面分成兩個部份，視覺化項目以及使用者介

面邏輯，視覺化項目存放在「.xaml」檔案中，使用者介面邏輯則存放在「.xaml.cs」檔案中。因此設計開發人員可以選擇微軟所提供的控制向來實作，自行繪製介面外觀，最後針對每個控制項撰寫背後執行動作的邏輯程式碼。

2.2.3 XAML 語法與物件屬性

XAML 是一種標記語言，其語法結構與其他標記語言差不多，例如 HTML 的結構，XAML 也具有標頭及標尾的部分，而且 XAML 是以 XML 為基底所發展的標記語言，因為有 XML 的特性，因此在物件的宣告方式上是非常嚴謹的。

XAML 的基本語法以左角括號(<)作為開頭、正斜線(/)和右角括號(>)為結尾，整個物件的宣告需在這兩個左右角括號內完成，一開始在緊接著左角括號(<)之後是類別或是結構的型別名稱，在這個型別名稱後可加上一個或多個空白，再來是宣告物件的屬性，一個物件可以包含多個屬性，但是有一個屬性為必要屬性為 Name，Name 代表物件的名稱，屬性給值的方式為「屬性名稱="值"」，若要宣告多個屬性，屬性之間以一個或多個空白來做間隔，下列為範例：

```
<Grid x:Name="LayoutRoot" Background="White"/>
```

另外，若在應用程式中有多個物件、屬性和內部文字等，也可以一併寫在開頭標記之後，這裡可以包含物件的確切內容，詳細內容通常會受到物件模型所限制。多數個物件之間的宣告方式為巢狀結構，除了有開頭標記之外，此時物件的結尾必須是「對等結尾標記」，而且並須在巢狀結構當中有適當的相對位置，也就是說，每個物件的開頭和結尾標記成對對稱。

下列為範例:

```
<Canvas>
```

```
<StackPanel>
```

```
<Image x:Name="Img1" Height="240" Width="320"/>
```

```
</StackPanel>
```

```
</Canvas>
```

在 XAML 中，要控制物件的屬性有很多種方式，但是也有些特定的物件屬性需要特定的方法才能設定，必須在該物件或是屬性的設定下設計者才能控制，經由控制物件屬性的動作，可改變應用程式物件的外觀，以及物件的位置和動作等，並可加入特定功能使物件更具變化性。

當使用者建立某物件時，物件會有個初始狀態(高、寬、內容、可視等)，開發工具會以這些屬性的值建立配置物件的外觀或位置，例如 Button 物件會預設以下屬性和其值，包括水平對齊(HorizontalAlignment) 、垂直對齊(VerticalAlignment)、寬(Width)、高(Height)、顯示內容(Content)、按下後觸發事件(Click)。下列為範例:

```
<Button x:Name="Play_Button1" HorizontalAlignment="Left"
VerticalAlignment="Bottom" Width="75" Content="Play" Height="26"
Click="PlayButton1_Click"/>
```

註：其中 Click="PlayButton1_Click"的 PlayButton1_Click 為一函式，設計者可以在後端程式碼設計按鈕按下後會做的動作。另外再設定屬性時，開發工具會以提示的方式，如圖 2-3，幫助程式設計人員選擇屬性值，可以加快設計師在撰寫時候的速度。

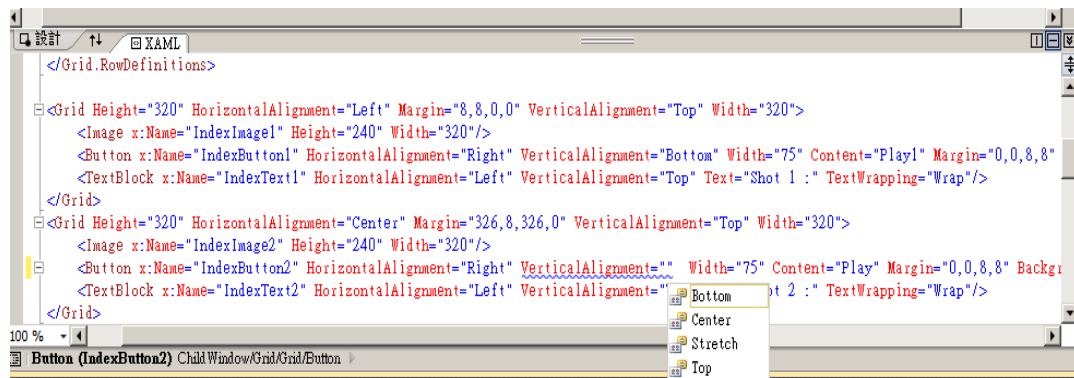


圖 2-2: XAML 以提示的方式設定屬性值

XAML 提供了一種延展的語法，可以在設計時把使用者介面和邏輯程式碼分開，簡化設計時的難度也可增加設計時的流暢度，意思就是說在設計一個應用程式專案時，可以把整個開發工作分割成介面設計人員和程式開發人員端分別進，將介面部分交給專門的介面設計師設計，程式設計師提供後端的服務部分，讓兩者之間分工更清楚，各自獨立作業發揮專長，增加開發的效率。

2.3 XML 及 eXist 原生資料庫

2.3.1 XML 簡介

XML (Extensible Markup Language) 為可擴展標記語言，在 1998 年由 W3C 所推出的標準，XML 是一種用來描述資料的標記語言，主要目的為傳送及攜帶

資料，訂定一個統一的標準有助於文件內容的宣告，並可符合跨平台的搜尋作業。

XML 是 SGML(Standard Generalized Markup Language)的子集，因為 SGML 太過於複雜、使用困難，因此 XML 是簡化的 SGML，XML 將 SGML 底層複雜的語法結構隱藏，使其更易於使用，另外 XML 比 SGML 優勢的特點在於他和 HTML 一樣都繼承了 Web 的功能，使得 XML 更適合用在網路上資訊的傳輸和處理。

在 HTML 中，不管標籤是在整個文件的哪個地方，只要有<H1></H1>標籤出現，在這兩個標籤之間的文字就會以“標題 1”的格式顯示，而在 XML 中，標籤出現的位置若不一樣，意義也會有所不同，需要考慮到此元素的父元素，例如:<課程名稱></課程名稱>這一組標籤代表的就是<課程列表><課程>裡面的<課程名稱>，而不是<課程列表>裡面的<課程名稱>，所以如果每個學校之間都用同樣的 XML 格式，那麼這些學校之間就可以交換開課資訊，所以 XML 稱為“形而上語言”(meta-language)，也就是 XML 是用來定義語言的語言。

現在有很多組織都在使用 XML 格式來定義他們領域的專屬標籤語言，像是 MathML 是用來定義數學式子的標籤，目前也成為 W3C 推薦的標準，有了這個標準之後，就不會發生在 Word 中可以正常使用，而其他文書編輯軟體無法使用的狀況，除了 MathML 之外還有影音多媒體的 SMIL 等，很多標準都被定義出來，使很多領域的資訊交換更為便利。現在很多廠商也開始採用 XML 的格式，例如 Adobe、IBM、微軟、Netscape、Oracle、Sun 等，目前穎很多新的軟體，例如 Microsoft Internet Explorer 及 RealPlayer，都已經在軟體內部使用 XML 的技術

了。

XML 的成功因素：

- 在使用電腦跟其他電腦做溝通時，常常需要互相傳遞訊息來完成工作，而 XML 可以定義傳遞訊息的標準。
- 在網路上資料的傳輸都是以串列的方式來做傳輸，所以資料必須轉成一長串的形式，稱為串列化格式 (Serialized)，而 XML 本身就是串列化的格式。
- 在接收端的電腦收到資料後必須要能解讀這些串列式資料的格式。
- 使用同種一個標準的格式，可節省系統開發人員費心在資料的交換格式的設計，可以把精神專注在更重的問題點上。

因此，XML 又稱之為運用在 WWW 上資料交換的標準格式

2.3.2 eXist 原生資料庫簡介

eXist 為 XML 原生資料庫，顧名思義此資料庫是專門用來儲存 XML 檔案的資料庫，儲存時不會破壞或是轉譯 XML 原有的階層化結構，完全依照原始文件本身的結構儲存，而關聯式資料庫儲存的方式是將資料一筆筆存在資料庫欄位中，再透過資料表之間的相互關聯性就夠出資料對應表，因此如果將 XML 文件儲存於關聯式資料庫中，必須透過特殊轉換器轉換 XML 文件結構成為關聯式資料庫的對應儲存方式，如圖 2-4，相對從關聯式資料庫讀取資料時也必須再經過一次的轉換程序，如此一來在轉換過程中不只耗費時間也消耗掉電腦效能，除此之外，

若要修改文件當中的資料，一定會影響資料之間的關聯性，則需要重新建構一張對照表，造成修改時的不方便，因此若能以原有的 XML 結構儲存，對於日後在資料庫內的修改資料動作會較有效率。

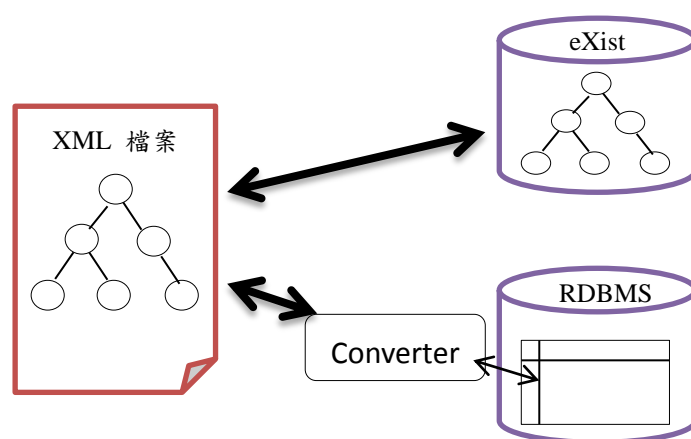


圖 2-3:eXist 與 RDBMS 儲存文件意象圖

eXist 是由 Wolfgang Meier 在 2000 年所建立出來的一套 XML 資料庫系統，eXist 為一套免費軟體，是公開原始碼的 XML 原生資料庫，除了儲存 XML 文件之外，也支援 XQuery 查詢，相容於許多網頁技術，因此成為網路應用發展時優良的應用平台。其主要特徵包含 XPath 搜尋、關鍵字搜尋，且與其他 XML 開發工具緊密整合，該資料庫軟體由 Java 所寫成，可以單獨執行，也可整合至其他應用中。

2.4 OpenCV(Open Source Computer Vision)

2.4.1 OpenCV 簡介

OpenCV 是一個基於 C/C++ 語言的開源程式碼影像處理函式庫，包含了 300 多個 C 語言函數的跨平台的中高層 API，不依賴其他外部的函式庫，為獨立的圖像處理庫，提供多種圖像處理和電腦視覺方面的 C/C++ 程序，其中大部份的函數用來處理解決電腦視覺領域中常見的問題，例如人臉辨識、動作分析、物體追蹤、傅立葉轉換、目標識別和 3D 重建等，可以整合不同圖檔格式的矩陣運算，主要應用在靜態圖片(BMP、JPG、TIF、PNG)和動態 Webcam 的影像處理。OpenCV 為 Intel® Integrated Performance Primitives (IPP) 提供了透明介面，優點包括支援 Windows 和 Linux 兩大作業系統(跨平台)、只要是選擇 OpenCV 來當作影像開發函式庫，那麼開發出來的城市都將能輕鬆地移植至其他的作業平台上(可移植性高)、使用便利、以及對於商業應用或是非商業應用都是免費的，最重要的是具備強大的圖像和矩陣運算的能力，可以減少程式設計者的編程工作量，有效提高開發時的效率，所以本系統選擇利用 OpenCV 來做影片的處理。

2.4.2 OpenCV 可應用領域

➤ 結構與矩陣(Structures and Matrices)

OpenCV 提供自定義的資料結構及矩陣，當要對影像進行操作處理時，都必須先轉換成這些一致性的資料結構進行，另外，對於圖片的存取也提供特有的線性結

構，有助於加速編碼轉換及演算法的進行。

➤ 影像處理(Image Processing)

OpenCV 包含許多影像處理的分析與功能，例如最基本的影像修改功能，文字繪製、繪製線、多邊形、曲線等，也提供更進階的功能，例如高斯計算、色彩轉換、邊緣偵測等。

➤ 結構化分析(Structural Analysis)

當影像被進行輪廓處理之後，可藉由 OpenCV 函式庫提供的分析函數來對影像物件進行處理，如幾何圖形比對、影像物件擷取、面積標定、輪廓長度測量及影像物件比對等。

➤ 動作分析與追蹤(Motion Analysis and Tracking)

OpenCV 提供了不只靜態的物件處理，也支援動態影像的處理，主要包括物件分析及運動追蹤，他能及時處理動態的連續影像，進行前後場景分離、圖像物件擷取、輪廓描繪等，並且可標示出影像中的物件運動軌道。

➤ 物體辨識(Object Recognition)

OpenCV 提供了 eigenobjects 及 hidden Markov models (HMMs) 兩大物件比對技術，可針對影像擷取物件並進行關鍵特徵搜尋及比對，可應用於人臉及指紋辨識等領域。

➤ 相機校正及三維模型建立(Camera Calibration and 3D)

OpenCV 提供的攝影機標定功能可用於三維模型建立，可將攝影機所擷取的

影像轉換成現實場景中的所對應的實際數據，可用於汽車輔助駕駛系統的開發。

第三章 影音資料庫之索引系統規劃

本研究以 Silverlight 網頁技術製作高互動的使用者介面，搭配 XML、JSP、C#、eXist 等技術實作應用於語言學習影片的影音資料庫索引系統。本章節對影音資料庫索引系統之規劃、架構以及開發技術做詳細的探討，第一節為新舊系統比較及影音資料庫的索引系統架構，透過比較圖，如圖 3-1，呈現本研究新增加的功能，在藉由架構圖，如圖 3-2 可清楚表達本系統的結構及運作行為，第二節為索引模組詳細運作流程及方法描述，包括影片組成要素分析以及利用影像處理偵測場景轉換，第三節為索引查詢模組，本小節清楚介紹查詢流程及方法，第四節為標籤及內容修改，包含 Silverlight 存取 eXist 資料庫的方法及標籤修改的技術探討，第五節為 Silverlight 網頁技術的詳細解說，利用 Silverlight 作出高互動性的網頁，以及內部模組技術說明。

3.1 影音資料庫之索引系統架構

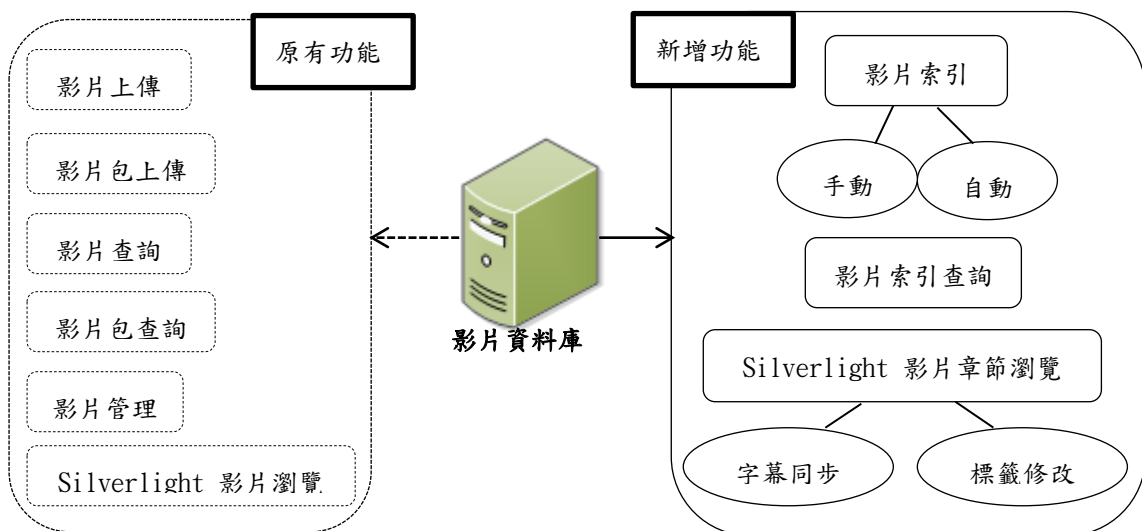


圖 3-1:影音資料庫新舊系統功能比較圖

本研究透過 JSP 網頁作為系統的入口，使用者可在 JSP 入口做影片上傳、影片管理、影片查詢和影片索引查詢的動作，當使用者上傳影片時系統會自動將影片切割成幾個章節段落，每個章節會有一組對應的標籤來描述每段章節的影片內容，這些標籤內容會儲存在 XML 文件中，在上傳至 eXist 資料庫中儲存，爾後使用者可以透過本系統提供的索引查詢模組來快速尋找到某個影片的章節，尋找到目標後系統會自動導向 Silverlight 頁面做影片的瀏覽，使用者也可在觀賞影片時手動加入影片章節索引和編輯影片章節的描述內容，增加日後使用者對於影片章節的搜尋準確度。

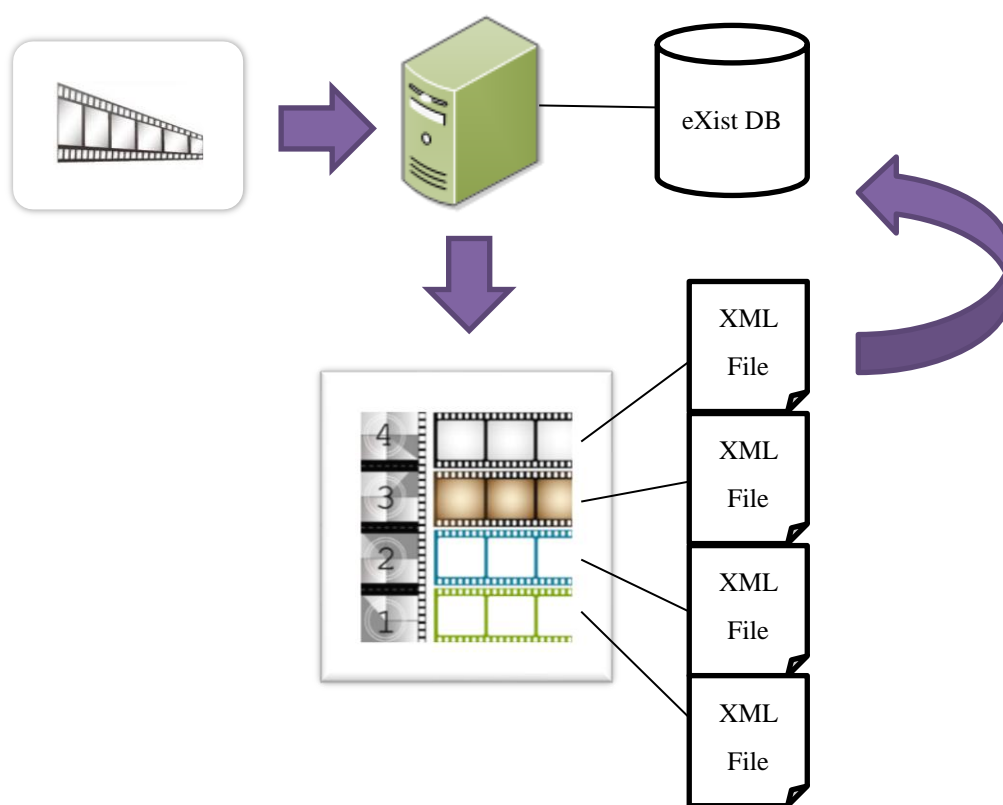


圖 3-2:影音資料庫索引系統概念圖

3.2 自動與手動索引模組

當使用者上傳整部影片至影音資料庫平台時，系統會透過自動影片索引模組將影片切割成幾個章節，此模組主要是利用自動化標示章節的步驟來降低對使用者的負擔，使用者也可在日後觀賞影片時手動加入章節。本模組利用 OpenCV 做影片的處理，判斷出影片的場景變化差異，在適當的位置加入章節索引，將整部影片標示出數個章節，標示完畢後使用者可以依照章節內容製作標籤描述或是新增標籤，並在日後可利用這些標籤內容做搜尋的動作。

3.2.1 影片組成要素

影片是由多張框架(frame)組合而成的，透過短時間內連續快速撥放框架以及利用人類視覺暫留的特性，來達到影片動態的效果。一般來說，影片的組成要素有以下幾點：

➤ 畫素(pixel):

畫素為圖像顯示的基本單位，每個畫素都有各自的顏色值，可以是三原色顯示，分成紅、綠、藍三種子像素(RGB 色域)，或者是青、品紅、黃、黑像素(CMYK 色域)，而每個畫素所能表達的顏色多寡取決於用幾個位元(bit)來表達畫素的值 (bpp, bit per pixel)，最大數可由取 2 的色彩深度次方得到，常見的值有：

- 8bpp: 由 8 個位元表示色彩，有 256(2⁸)色，稱「8 位色」。
- 16bpp: 65, 536(2¹⁶)色，稱為高色彩，也稱為「16 位色」。
- 24bpp: 1670 萬(2²⁴)色，稱為真色彩，也稱為「24 位色」。

- 32bpp:32 位色為計算機領域常見的位色，而它所代表的並不是 232 種顏色，而是在 24 位色上增加了 8 位元($2^8=256$)的灰度(也稱灰階)，32 位色也成為真彩色或全彩色。

➤ 解析度(resolution):

解析度為描述影像細節的分辨力或是影像的清晰程度，解析度越高代表影像品質越好，可表現更多細節；相對因為紀錄資訊多，所以檔案也越大。圖片是畫素的集合，而解析度是由畫素的個數所決定，即影像的長度乘上寬度，單位面積內的畫素越多代表解析度越高，所顯示的圖片越接近真實物體。比如說一張 640*480 解析度的圖片的長有 640 個畫素、寬有 480 個畫素，總共有 307,200 畫素，在同一個顯示裝置中，解析度越大，所顯示的圖片也越大，若將解析度小的圖片放大，則會呈現模糊狀態，如圖 3-3、3-4 所示。



圖 3-3:解析度為 640*480 圖片



圖 3-4:解析度為 160*120 圖片放大圖

➤ 框架率(FPS):

框架率是由每秒播放的框架數所定義，每秒播放框架數越多，影片越流暢，但是系統負荷就越大，因為要處理較多的影像。人可以看見影片，其實是因為人類眼睛有視覺暫留的特性，如果框架率為 10 以上的話，人類就會認為畫面是連貫的，因此，影片為許多圖片連續快速切換所造成的結果，而目前市面上的影片框架率大多為 24 到 30 之間不等。

➤ 影片格式(format):

- AVI: 視訊及音訊為分開儲存，可解決音訊與視訊的同步問題，具相容性好、圖像品質良等特性，缺點為檔案容量太大。
- ASF、WMV: 網路電視專用格式，框架率為 15，檔案小，但是此影片格式需要 windows media player 7.0 以上才能看，為微軟的壓縮技術。
- MOV: Apple 公司主推的視訊格式，可通用於 MAC 系統與一般 PC，視訊

檔可採壓縮或不壓縮的方式。

- MPEG4:畫質細膩、音效動感逼真，視聽效果接近 DVD 水準，且編碼後可達 DVD 影音檔的 1/3 左右，被稱為 DVD 殺手。
- MPEG-I:VCD 的編碼方式，在畫質方面表現不出色，市面上這種格式已慢慢被其他格式所取代。
- MPEG-II:DVD 的編碼方式，畫質比 MPEG-I 好，但是壓縮編碼較特殊，各種編輯軟體無法隨機使用。
- RM(Real Media):流行的網路流媒體格式之一，可自行設定編碼速率，具備動態補償，但是不適合高畫質視訊的儲存。

3.2.2 索引原理與流程

要在影片中做索引的功能，首先要將原始影片找出幾個不同主題的影片段落，為影片作章節標示，接著每一個章節各對應一份 XML 文件，文件內容為各章節的詳細資訊，日後可對於 XML 文件的元素作搜尋，將相對應的影片內容提供給使用者觀看。要將一個原始影片標示出幾個章節有兩個方式，第一種為人工的方法，影片擁有者將影片瀏覽過一遍，並藉由人的判斷將影片標示成一個一個的章節，此做法非常耗費時間，可能多於播放影片一次的時間，因此為了改善索引速度，在本研究中任何人都可以以人工的方式做影片標示，而當你標示完後，系統會記錄下來，日後其他使用者也可以根據你所提供的資訊搜尋到你所標示的地方，第二種為自動的方法，此方法不需人工觀看影片來標示章節，本研究中我們設定

影片中場景轉換的時間點作為影片章節的開始處，並以這個影片時間點做為索引的依據，日後使用者搜尋到這些影片段落時，系統將跳至這些索引時間點開始播放。

在自動索引的方法流程中，如圖 3-5，將影片標示出章節時間點後，最重要的是選出代表性圖片(Representative Images)，將影片中足以代表出重要內容的關鍵畫面選出。選出的代表性圖片可應用於影片快速瀏覽、摘要、分類等方面，或是做成小縮圖(Thumbnail Images)，提供影音資料庫之檢索及索引(Indexing)。使用者可從此代表性圖片瞭解整段影片大致內容。

此模組利用 OpenCV 自動從多個場景的影片中擷取代表性圖片的方法，經由電腦對影片中內容的剖析，將影片切割成數個章節，在系統分析影片內容後，再藉由場景變化偵測方法分割成數段的影片片段。

主要方法是判斷畫面變動量的多寡作為擷取關鍵畫面的考量，將影片的每一張圖片的特徵值(像素、顏色統計值等)抓取出來，判斷影片中每格畫面間的特徵差異來決定此處是否為影片場景轉換的地方，如果差異大於所設定之值，則此畫面為關鍵畫面，此種概念是利用影片中畫面與畫面之間應會有其相似性，若是忽然差異性過大時，表示此畫面是這段開始畫面，也可代表此影片片段之關鍵畫面，藉此來達成章節的標示做成影片的索引。

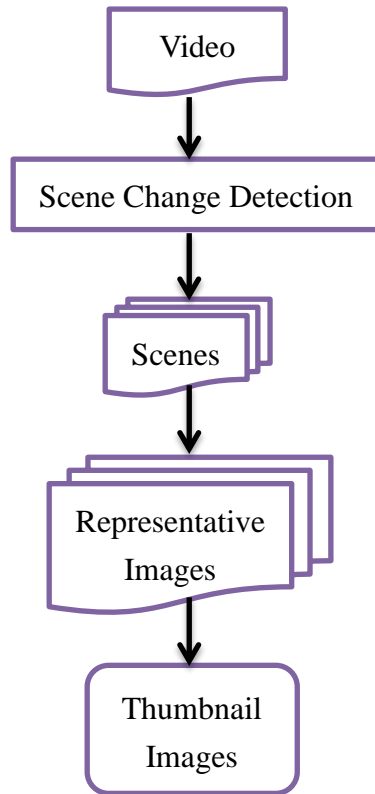


圖 3-5:選擇代表性縮圖流程圖

3.2.3 影像處理方法

為了偵測影片中的場景是否發生轉換，我們必須對影像進行分析，所以必須利用影像處理方面技術來達成，當影片中場景發生轉換時，轉換前的框架和轉換後的框架內容差異度會很大(圖)，我們遂利用此特性來判斷場景是否改變，及影像差異(image difference)。考慮到系統的效能，選擇抓取的特徵為框架的灰階值來做判斷而不是抓取框架的三原色(RGB)，在使用者將影片上傳後，影片經由自動索引模組處理程序，系統會依序從影片第一張框架開始抓取，每抓一張框架馬上就轉成灰階等待系統抓取下一張圖來做比對，只要框架的特徵值差異很大，即判斷此處為影片場景變換之處，此處為章節開始的地方。以下為詳細處理流程：

1. 讀取影片:當使用者上傳影片至伺服器中，讀取程序以檔案絕對路徑開啟資料庫中的影片檔案。
2. 擷取兩張時間相鄰的框架 I_p (前一張框架)和 I_c (目前框架):須將取出的框架的格式(可能是 bmp、jpg、png、tif)，轉換成 OpenCV 自定義的資料結構格式「IplImage」，如圖 3-6，作進一步的處理。

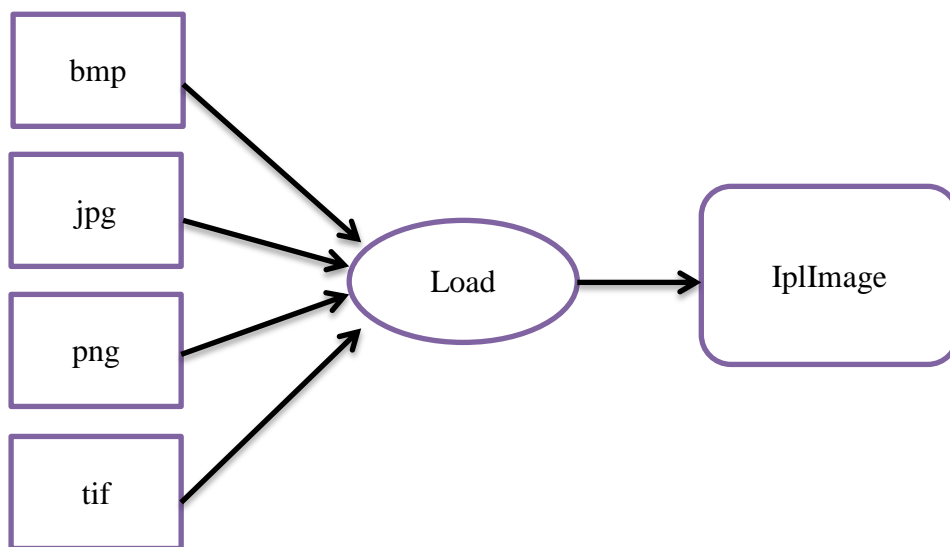


圖 3-6:資料格式轉換圖

3. 取出每個畫素:

$I_p(x,y)$: I_p 代表前一張框架在(x,y)位置畫素的灰階值

$I_c(x,y)$: I_c 代表當前框架在(x,y)位置畫素的灰階值

例如: $640*320$, $0 \leq x \leq 639$ 、 $0 \leq y \leq 319$

解析度越高的影片，畫素越多，xy 值相對越大，因此解析度越高處理時間越長。

4. 將彩色值灰階化:一般來說，影片都是以彩色呈現，由紅、藍、綠三原色線性內插所組成，我們簡化特徵，對彩色影像進行灰階化(圖)。

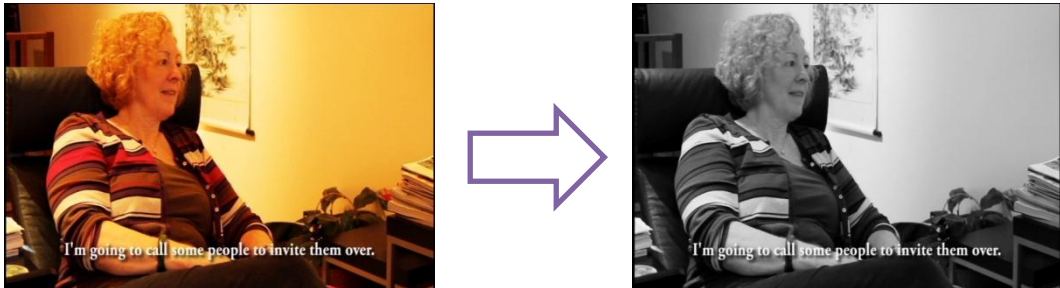


圖 3-7:彩色影像灰階化結果

5. 計算影像差異:如圖 3-8、圖 3-9，灰階化之後，我們將相同位置的畫素的灰階值相減，判斷其差異程度，若兩畫素越相似，則相減後畫素色彩值越小(越偏黑色)，式子為下列所示

$$I_p(x,y) - I_c(x,y) > T \text{ where } 0 \leq T \leq 255$$

T 即為我們自訂的門檻值(threshold)



圖 3-8: 差異小的框架灰階值相減結果



圖 3-9: 差異大的框架灰階值相減結果

6. 統計差異度大的畫素個數:

當其個數大於整個框架的一定比例時，即認為場景發生改變，並利用影片長度及框架率計算出框架時間點。

3.3 索引查詢

在影音學習平台中，查詢功能是系統中很重要的一個環節，本研究所提供的查詢功能不只能夠搜尋到整部影片，而且進而發展提供搜尋影片章節的功能，在語言學習平台中，由於教學影片可能很冗長，導致學習者對於觀看整部教學影片感到無趣，或者找不到學習重點而降低了學習效果，因此本系統提供一個完整的搜尋模組，使用者不一定要搜尋整部影片來做觀賞，而是可以藉由使用者輸入的關鍵字（一個單字、一段句子、標題等）來做影片章節的搜尋，將所有相關的影片章節內容資源呈現給使用者，使用者可以從中挑選想要觀看的部份做觀賞，達到快速學習和重點學習的目的。

資料查詢主要功能是根據使用者所輸入的關鍵字字串，在資料庫中找出包含或者相符的內容並回傳相關的 XML 文件，透過此文件所對應的影片章節可以準確的呈現給使用者。本研究中以 eXist 資料庫儲存影片章節資料的相關 XML 檔案文件，再利用資料庫中所支援的查詢語言“XQuery”作為搜尋的語法。

3.3.1 XQuery 簡介

XQuery 是由 W3C 的 XML Query 工作小組所定義開發的語言，是一種功能強大、完整和容易使用，為專門查詢 XML 檔案文件資料的查詢語言。XQuery 語言是一種路徑運算式（Path Expressions）、FLWOR 運算式、條件運算式、和 XQuery 函數的集合，它是一種強調型態的語言。XQuery 語言有以下特點：

與其他 W3C 的規格相容

- 可修改輸出資料以及新增 XML 元素和屬性
- 可支援多文件的合併查詢
- 可以輸出排序的查詢結果
- 可將 XML 文件轉換成其他 XML 文件或是輸出成 XHTML 文件
- 可查詢結構化和非結構化的資料
- 使用獨立通訊協定，以便可以在不同系統平台中取得相同的查詢結果

3.3.2 XQuery 語法

1. 路徑運算式 (Path Expressions) :XPath 為 XQuery 的子集，其 XPath 本身的標準語法已經符合 XQuery，其方法為指定 XML 文件中符合條件的某個節點 (元素) 的路徑運算式，以下為範例：

列出所有為章節名稱的元素

```
//indexname
```

列出 video 中所有的 indexname

```
Document("video.xml")//indexname
```

2. FLWOR 運算式 :XQuery 中的流程控制語法，一個 XQuery 的敘述式為一組 FLWOR (發音同 “flower”) 表示法，是由 For、Let、Where、Order by 及 Return 子句等結構而成，並以特定的順序組合而成如圖 3-10

所示：

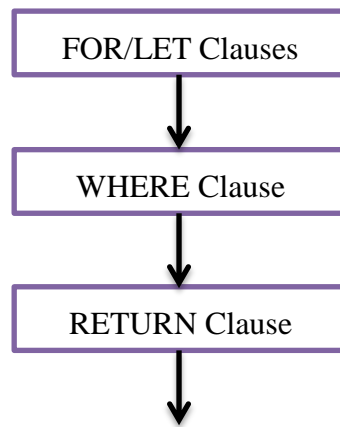


圖 3-10:FLWOR 運算式特定順序組合

- FlwrExpr：FOR 或 LET 敘述其後必伴隨一個 WHERE 敘述
- WhereClause：WHERE 敘述可在接另一串的合法的 XQuery 查詢，包含另一串 FLWR 敘述
- ReturnClause：傳回查詢結果

以下為範例：

列出所有章節

```
LET $Chapter:=//Chapter
```

```
RETURN
```

```
$Chapter
```

列出所有章節

```
FOR $ Chapter IN // Chapter
```

```
RETURN
```

```
$ Chapter
```

3. 條件運算式：支援下列 if-then-else 條件陳述式

```
if (<expression1>) then( <expression2> )else (<expression3>)
```

以下為範例：

列出 videos 中，目錄為 English learning 的元素內容，否則列出目錄為

Chinese learning 的元素內容

```
for $x in doc("videos.xml")/videodb/video
```

```
return if ($x/@category="English learning")
```

```
then <English learning>{data($x/title)}</English Learning>
```

```
else <Chinese learning>{data($x/title)}</Chinese learning>
```

3.3.3 XQuery 查詢流程與方法

查詢流程如圖 3-11 所示，使用者透過介面輸入查詢關鍵字，字串經過 eXist 資料庫所提供的 API 轉換成 XQuery 語法，透過 XQuery 語法於資料庫中搜尋，最後顯示結果於 Silverlight 網頁。

:

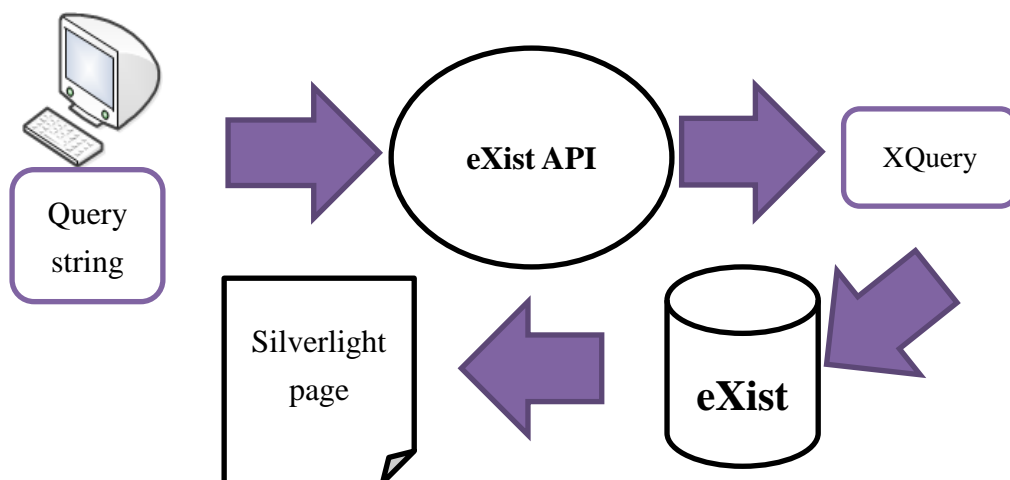


圖 3-11:XQuery 查詢流程

程式碼 3-1 說明依照關鍵字查詢的程式碼片段，尋找 indexname 為”自動語音系統”，並回傳 id、clip、indexname。

```

for $x in collection(/db/video)//video
where $x/indexname='自動語音系統'
return (data($x/id),data($x/clip),data($x/indexname))
  
```

程式碼 3-1:XQuery 依照關鍵字查詢

```

1326033790343
2
自動語音系統
  
```

程式碼 3-2:XQuery 依照關鍵字查詢結果

3.4 Silverlight 影片瀏覽與同步字幕與標籤內容修改

要增加語言學習整體品質，除了使用影片索引系統之外，重要的是提供使用者一個良好互動的影片瀏覽環境，透過生動的瀏覽系統呈現最完整的學習內容給使用者學習。本系統透過微軟公司所推出的 Silverlight 網頁技術實作影片瀏覽系

統，經由 Silverlight 所設計出的應用程式除了有跨平台的優點之外，也提供了高互動性的影音功能使用介面，Silverlight 強大的影音功能及互動性剛好符合本系統所要呈現的理念，這些優點使得使用者可藉由本系統達到良好的學習效果。

3.4.1 Silverlight 影片瀏覽架構

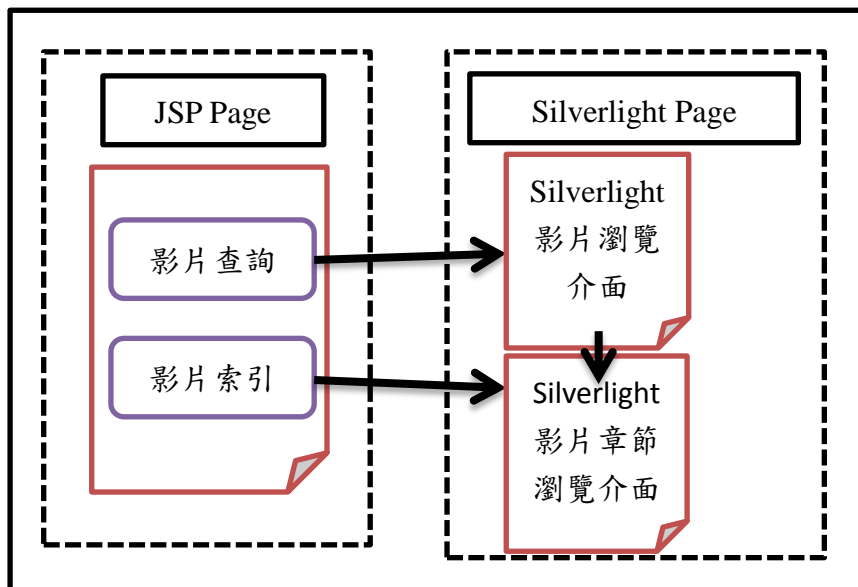


圖 3-12:影片瀏覽架構

如圖 3-12，Silverlight 瀏覽介面主要分為兩個，一個為影片查詢結果的呈現介面，也是系統中主要瀏覽影片的地方，另外一個為影片索引查詢結果的呈現介面，主要提供影片章節的瀏覽。以下小節針對兩個瀏覽介面所擁有功能模組及各模組內使用到的技術作詳細的介紹及討論。

3.4.2 Silverlight 瀏覽介面

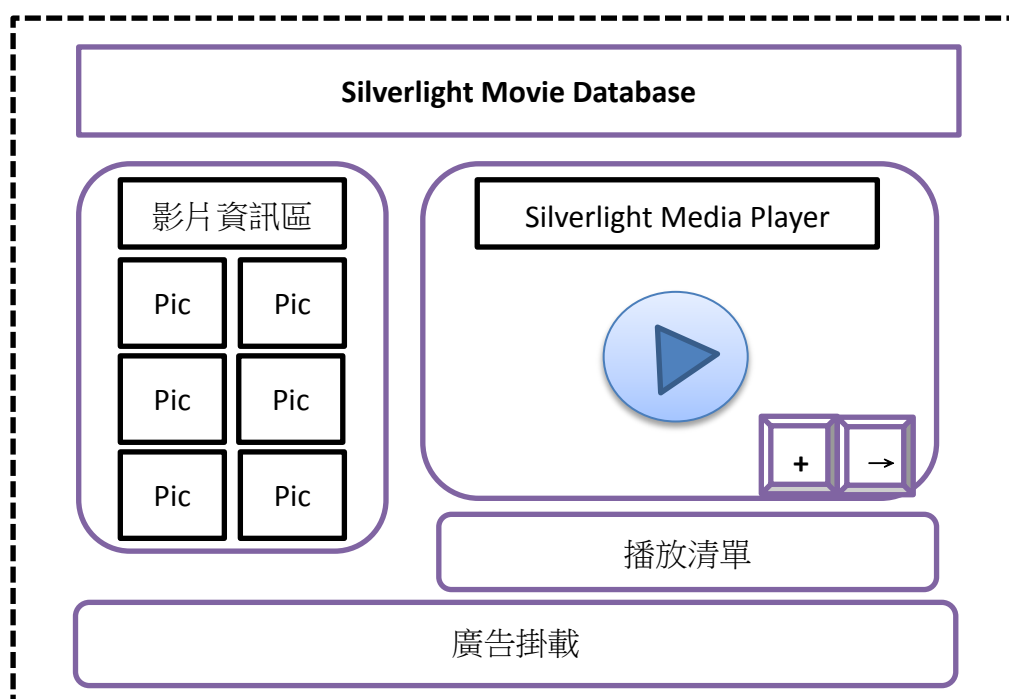


圖 3-13:Silverlight 影片瀏覽介面

如圖 3-13，Silverlight 影片瀏覽介面分成幾個區塊及功能：影片資訊區、Silverlight Media Player、廣告掛載、播放清單、手動加入索引按鈕、章節呈現按鈕等，系統使用流程大致可分為下列步驟：

1. 影片資訊區提供影片縮圖做為跟使用者之間的第一互動，使用者可點選有興趣的影片縮圖，系統將提供影片名稱及長度資訊，作為第一階段篩選，使用者可在此決定是否觀看影片。
2. 若使用者對於所提供之影片名稱及長度仍不足以判斷是否為有興趣觀賞之影片，可在第一步驟中選取「影片詳細資訊按鈕」按鈕，將會彈出子視窗描述更詳細的內容。

- 彈出子視窗有描述完整的影片資訊及播放按鍵，按下播放按鍵，影片立即在 Silverlight Media Player 上播放。
- 使用者可利用播放清單加入想觀賞的幾部影片，系統將依照播放清單輪播影片內容。
- 廣告掛載區一般用來作為網站商標、站台辨識以及列出內容贊助商，可在影片開始前及結束後插入廣告，減少使用者因為影片下載緩衝時間而放棄播放。
- 在播放影片的同時，若使用者想要手動加入索引，可按「+」按鈕，彈出一小視窗後，使用者可以拖曳時間軸設定開始及結束時間、並為此段影片內容輸入一個標題，系統會將此片段標示索引，供日後使用者尋找及觀賞。
- 使用者可透過「→」按鈕導向章節瀏覽介面，觀看目前影片的所有影片章節內容。

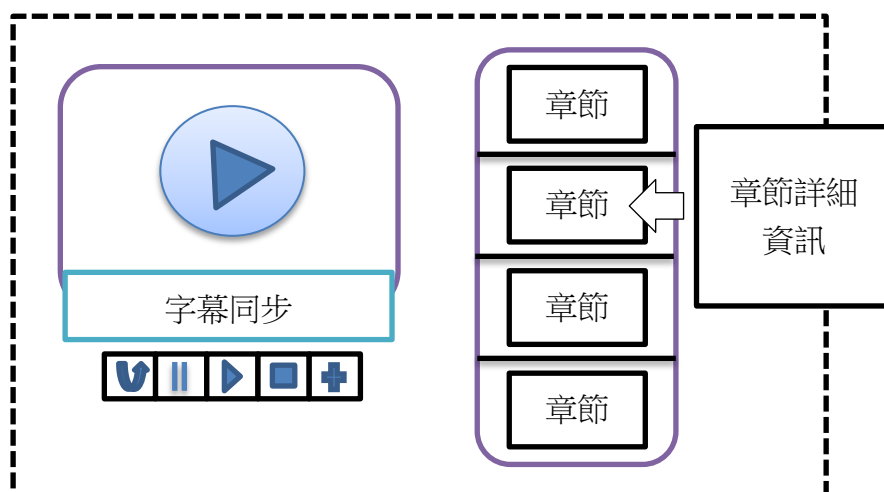


圖 3-14:Silverlight 影片章節瀏覽介面

如圖 3-14，Silverlight 影片章節瀏覽介面分成幾個區塊及功能：影片章節排列區、章節詳細資訊區、MediaElement 播放器、字幕同步區等，系統使用流程大致可分為下列步驟：

1. 章節排列區提供各章節影片內容的關鍵畫面，並列有開始時間以及章節編號，使用者透過關鍵畫面的呈現可大略推知章節內容，作為第一階段的快速篩選。
2. 點選章節縮圖可觀看章節詳細資訊，配合關鍵畫面可完全了解章節內容，詳細資訊包含章節主題、開始時間、結束時間、語言、以及章節描述等，並可在這個步驟對於詳細資訊內容的不完整處做修改，以便日後查詢此章節的準確性。
3. 點選章節縮圖可直接於播放器中播放章節內容，並在播放器下方提供字幕同步的功能，可於特定時間點顯示字幕，字幕內容可以是影片內容的重點、單字或是整段對話等，已主動的方式呈現此段學習影片的重點，加強使用者學習的效率。

3.4.3 Silverlight 模組內部技術

➤ Silverlight 影片資訊及章節排列區

利用影片畫面作為使用者篩選影片內容的依據為最有效率的方法，相對於純文字對於影片內容的描述更為有用，圖片為刺激視覺感官最直接的方式，若能以

圖片為主要呈現、搭配詳盡的文字說明，則使用者可以很快速的了解影片內容，在短時間內做出選擇。舊系統中，在影片上傳階段自動化為整部影片擷取影像，不過所擷取的影像並非影片關鍵畫面，造成使用者看到的影片縮圖可能不是重點，而影響選擇的結果，雖然有搭配文字說明，不過對於圖片篩選的功能性就降低了一點。為了增加完整性，我們分析整部影片的內容，找出每個片段的關鍵畫面作為縮圖，並利用縮排技術呈現於資訊區，如圖 3-15。

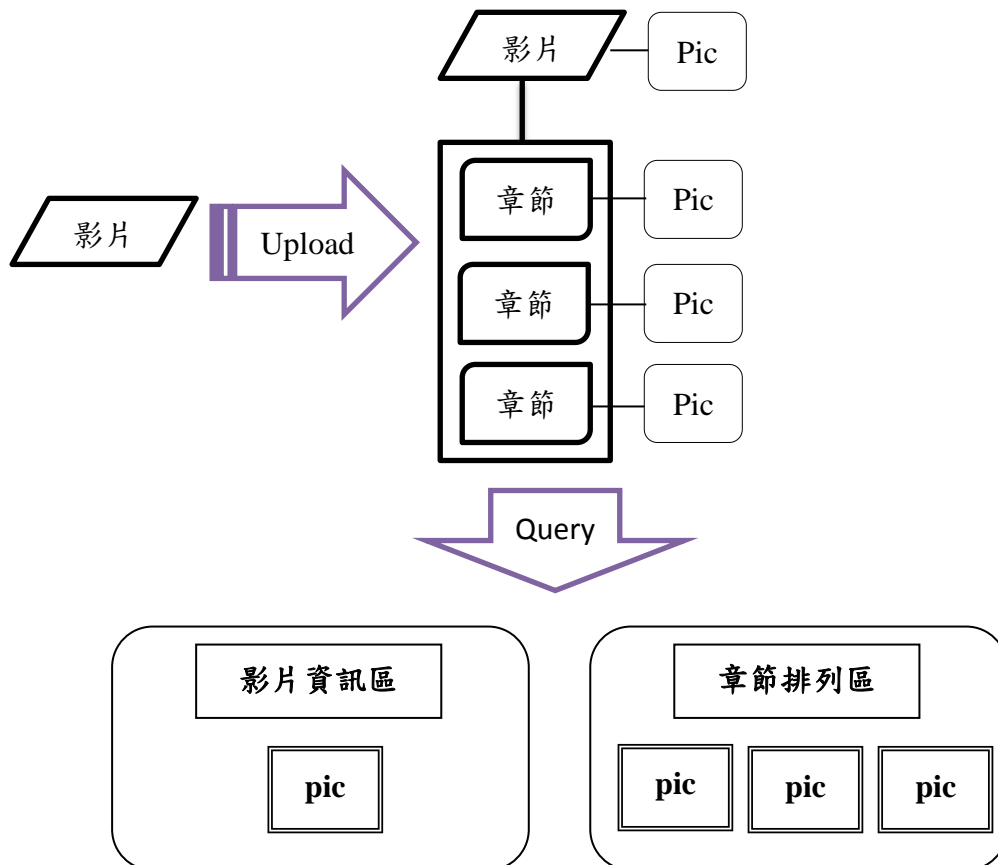


圖 3-15:影片關鍵畫面縮圖呈現示意圖

- 資料綁定(DataBinding)

DataBinding 為一種資料連結技術，提供網站開發者及使用者許多優點，包括製作網頁更為容易、更新及下載網頁更快速，並且較不依賴遠端伺服器，會有

這些優點是因為它允許網頁資料可以保存在本地端且與設定格式的原始碼離。

在本系統中 Silverlight 使用者介面常常要與資料庫作聯繫，一般來說，任何使用者額外的要求需要很多次傳送到伺服器端的處理程序，當使用者提出另外要求且整個龐大的處理程序又要重來一次，造成伺服器的負擔且耗時又沒效率。在影片資訊或是章節排列區需要動態顯示查詢結果，使用資料綁定技術可降低負擔且有效率。

本系統中使用 Binding 技術將資料綁定到 listbox 控制項中，由於 Silverlight 的 DataBinding 技術，在頁面讀取時就先將查詢資料的結果綁定到控制項中，讀取結束查詢結果隨即顯示出來，並在 listbox 當中每一個被點選的縮圖，可以透過底下的方式抓取到，透過程式碼可以清楚了解綁定的方式。

```
public class ReadIndexXML
{
    public string IndexImageUrl { get; set; }
    public string IndexImageID { get; set; }
}
```

程式碼 3-3:在 XAML 中建立資料綁定(一):定義來源物件

```
<UserControl
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:ExpressionMediaPlayer="clr-namespace:ExpressionMediaPlayer;assembly=MediaP
layer"
xmlns:My="clr-namespace:BindingXAML"
x:Class="Silverlight_VedioPlayer.ShowSearchIndex"
>
```

程式碼 3-4:在 XAML 中建立資料綁定(二):建立來源物件命名空間參考

```

<UserControl.Resources>
  <DataTemplate x:Key="DataTemplatel">
    <Image x:Name="BookImage2" Width="120" Height="120" Source="{Binding
      IndexImageUrl}" HorizontalAlignment="Center"
      VerticalAlignment="Center"/>
  </DataTemplate>
  <ItemsPanelTemplate x:Key="ItemsPanelTemplatel">
    <controlsToolkit.WrapPanel Orientation="Horizontal" ItemHeight="125"
      ItemWidth="125"/>
  </ItemsPanelTemplate>
</UserControl.Resources>

```

程式碼 3-5:在 XAML 中建立資料綁定(三):在 Resource section 中建立來源物件

```

<ListBox x:Name="IndexList" BorderThickness="8" Margin="8"
  Cursor="Hand" ItemTemplate="{StaticResource DataTemplatel}"
  ScrollViewer.HorizontalScrollBarVisibility="Disabled"
  SelectionChanged="BooksList_SelectionChanged"
  ItemsPanel="{StaticResource ItemsPanelTemplatel}"
  Background="#FF0E0E0E" BorderBrush="#FF847A7A" Grid.Row="1"
  d:LayoutOverrides="VerticalAlignment" Height="150"/>

```

程式碼 3-6: 在 XAML 中建立資料綁定(四):綁定到物件

- Silverlight 版面配置面板

本系統使用 Silverlight 所提供的控制項 WrapPanel 以及 StackPanel 自動排列呈現資訊區的縮圖，WrapPanel 控制項跟 StackPanel 控制項類似，但是可以將元素分配放置於多行，如圖 3-16 可依垂直方向(由上而下由左而右)或是水平方向(由左而右由上而下)排列容器內的項目，且不會因為內部元素超過呈現區邊緣而遭到裁切，隨著子物件越來越多，WrapPanel 控制項會繼續使物件換行，直到空間用完，才會裁剪物件，而 StackPanel 控制項會隨著新增的子物件愈來愈多，將裁

剪 (或隱藏) 超出面板大小的物件，如圖 3-17。

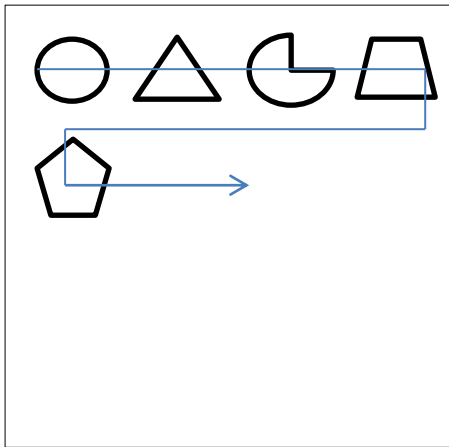


圖 3-16:WrapPanel 排列狀況

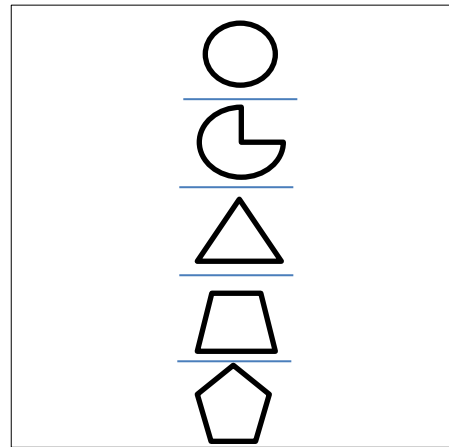


圖 3-17:StackPanel 排列狀況

Silverlight 提供許多控制項，對於開發人員在設計時非常方便，例如只要設置控制面板的 Orientation 屬性值，就可輕鬆的決定圖片排列的方向，或者配合使用者螢幕大小作縮放調整，不須再多花時間在編寫控制項外觀的部分，更可專心在後端行為的撰寫，內部程式碼如以下程式碼 3-7 所示。

```
<StackPanel Orientation="Vertical">
  <StackPanel Orientation="Horizontal">
    <TextBlock Height="20" Text="" FontSize="14"/>
    <TextBlock Height="20" Text="" FontSize="12"/>
  </StackPanel>
  <StackPanel Orientation="Horizontal">
    <TextBlock Height="20" Text="" FontSize="14" />
    <TextBlock Height="20" Text="" FontSize="12" />
  </StackPanel>
</StackPanel>
```

程式碼 3-7:StackPanel 控制項範例

➤ 章節詳細資訊及標籤修改

當使用者透過圖片不能完全了解影片內容時，可點選縮圖，系統將顯示出詳細資訊給使用者，當作下一階段的篩選條件，圖片配合文字可以完整表達出影片內容。另外，若使用者覺得詳細資訊的描述並不完全符合影片內容，可在查看資訊的同時對內容作修改。

● Silverlight 查詢 eXist 資料庫

本研究以 XML 文件技術為核心，提供使用者定義描述標籤的功能，可於每段影片附屬一組標籤描述，而這些標籤內容不只是顯示給使用者觀看的資訊，也可應用於系統內部的應用程式設定。系統運作過程中，應用程式需常常讀取 XML 檔案，由於 XML 具有結構以及可選擇讀取指定標籤的特性，可增加資料應用上的靈活性。

圖 3-18 為 Silverlight 查詢 eXist 資料庫流程圖，每部影片都有唯一的識別碼，當使用者點選其中一縮圖時，系統依照此識別碼讀取對應的 XML 檔案，程式碼 3-8 為 Silverlight 讀取 XML 文件及元素的方法。

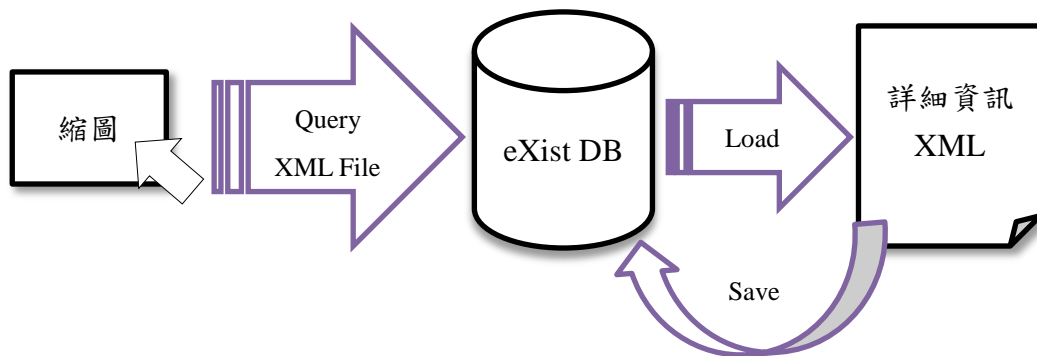


圖 3-18: Silverlight 存取 eXist 資料庫流程圖

```

private List<ReadIndexVedioXml> listReadIndexVedioXml = new List<ReadIndexVedioXml>();
Uri Vedioxmlfile = new
Uri("http://localhost:8081/eXist/Index_snapshot/"+ID+"/"+selectedclip+".xml");
WebClient client = new WebClient();
client.DownloadStringCompleted += new
DownloadStringCompletedEventHandler(SearchVedioXmlFileElement);
clientB.DownloadStringAsync(Vedioxmlfile);
//=====
XDocument document = XDocument.Parse(e.Result);
XElement rssFeed = XElement.Load(document.CreateReader());
var items = from item in rssFeed.Elements("videoindex")
select new ReadIndexVedioXml
{
IndexName = ((string)item.Element("indexname")).Trim(),
IndexClip = ((string)item.Element("clip")).Trim(),
IndexBegin = ((string)item.Element("begin")).Trim(),
IndexEnd = ((string)item.Element("end")).Trim(),
IndexLanguage = ((string)item.Element("language")).Trim(),
IndexDescription = ((string)item.Element("indexdescription")).Trim(),
};

```

程式碼 3-8: Silverlight 讀取 XML 文件及元素的方法

➤ 播放區 (Silverlight Media Player & Media Element)

在 Silverlight 專案中播放視訊和音訊有兩種選擇，可以使用 MediaPlayer 和 MediaElement，前者比較輕巧，表現上優於 MediaElement，而後者是比較重量級的 Framework Element，可以在 XAML 中宣告使用，支援 Loaded 和 Unloaded 事件，並可設置開始及結束播放時間、或者在某時間點顯示文字、製作功能按鈕控制視訊等多種客製化應用，設計人員可以依照自己及客戶需求開發相關的功能及外觀，在表現上比較多元。

➤ 字幕同步

MediaElement 支援許多觸發事件，例如當媒體被成功打開時觸發

MediaOpened 事件、當媒體未能成功打開時觸發 MediaFailed 事件、播放狀態發

生改變時觸發 CurrentStateChanged 事件、下載進度發生改變時觸發

DownloadProgressChanged 事件等，開發者可充分利用這些 MediaElement 提供的

觸發事件設計應用，在某個狀態應發生什麼效果，透過這些方便利用的事件函式，

造成不一樣的互動，本研究利用這些事件在正確的時間點對於 TextBlock 物件做

顯示改變，達到字幕同步的效果。

➤ 手動索引

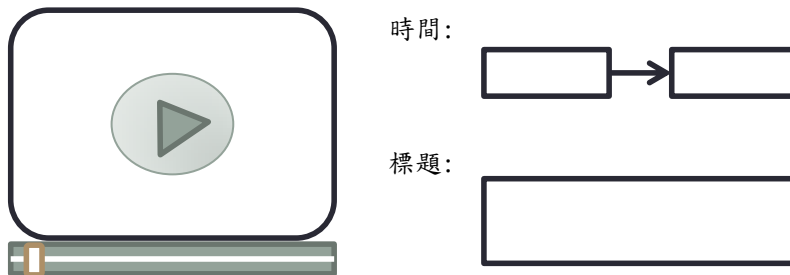


圖 3-19:手動索引介面

如圖 3-19，手動索引介面提供三個輸入資訊的地方，使用者可透過拖曳播放器下方的時間軸對時間做設定，並在標題的地方輸入關於這段時間內的主題，系統將儲存這些資訊當作索引的依據，使用者可透過搜尋關鍵字找到標題符合的這個標示片段。

第四章 系統實現

本章第一節內容為說明系統概觀、系統架構、環境建置。第二節以使用者案例圖及循序圖說明使用情況及系統內部的溝通流程。第三節為系統實際操作。

4.1 系統架構與環境建置

圖 4-1 為系統架構圖，進入平台前先透過帳密認證，認證通過之後進入首頁，本系統以 JSP 頁面作為首頁，提供影片查詢、影片上傳、影片索引、廣告掛載等功能，系統再依使用需求相向資料庫及多媒體伺服器進行連線，導向 Silverlight 頁面做影片瀏覽的動作。

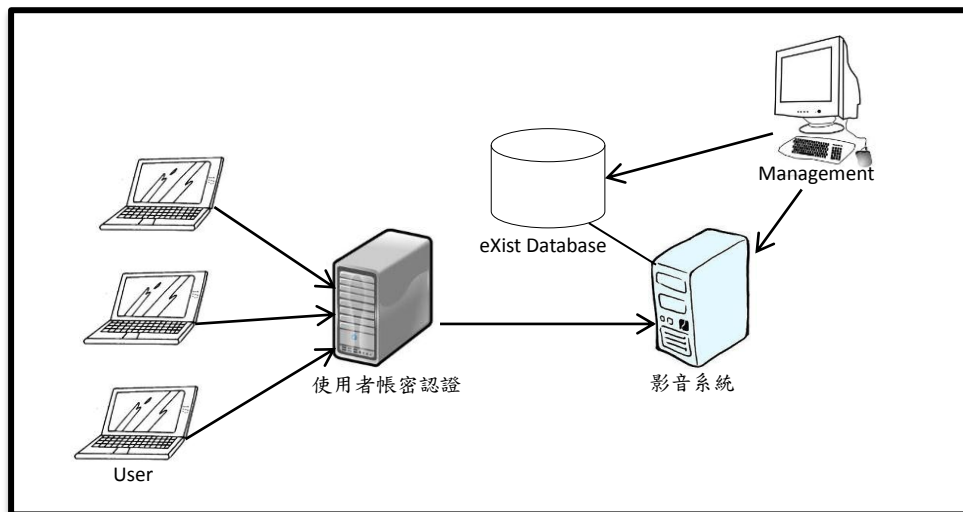


圖 4-1:系統架構圖

本系統環境建置如表，以 eXist 原生資料庫開放原始碼技術為基礎，管理 XML 檔案的編輯和輸出，以整份 XML 文件為單位儲存至資料庫內，保有 XML 檔案原始的結構，對於管理及存取文件的功能更有效率。由於儲存資料日漸增加，在 XML 內部管理的部份，以 Protégé-OWL 套件所產生的 RDF/OWL 來實現，可

將彼此互相關連的 XML 文件依類別做連結。對於 XML 文件處理的部份則是使用 JDOM 套件，利用 XML Parse 讀取 DOM 資料，將 XML 資料轉為 Java 物件方便修改及新增內部元素，最後再以 XML 的格式輸出。由於 eXist 及 Protégé-OWL 皆以 Java 語言開發，因此需安裝 Java JDK 作為執行環境，而網頁管理部分以 Tomcat 網頁伺服器作為網頁容器。

系統以 .NET 架構為主，在影片場景判斷方面需要使用到影像處理技術，因此需要建置 OpenCV .NET 作為影像處理環境。

網頁呈現以 JSP 與 Silverlight 技術為主，以 JSP 網頁作為系統入口介面，在導向 Silverlight 網頁瀏覽影片，Silverlight 有跨平台的特性，大部分平台皆能執行，大大增加網頁可用性，且網頁具有高度互動性，可提供更生動的視覺感受。

| 名稱 | 版本 |
|-------------------------|---|
| 作業系統 | Windows Server 2008 r2 |
| Java 環境 | Java JDK 6.0 |
| 網頁容器 | Tomcat5.5.28、IIS7 |
| 網頁技術 | JSP/Servlet、JavaScript、Silverlight、HTML、CSS、Ajax、XAML |
| 資料庫 | eXist1.2.6 |
| 開發環境 | Visual Studio 2010、Expression Blend |
| 程式語言 | C#、JSP、Java、C++ |
| 其他工具 | Protégé、Windows Media Services 9.5、OpenCV |
| Java OpenSource package | JDOM1.0、eXist、protégé-owl |

表 4-1:系統環境建置表

4.2 系統操作

4.2.1 使用案例圖 (Use Case Diagram)

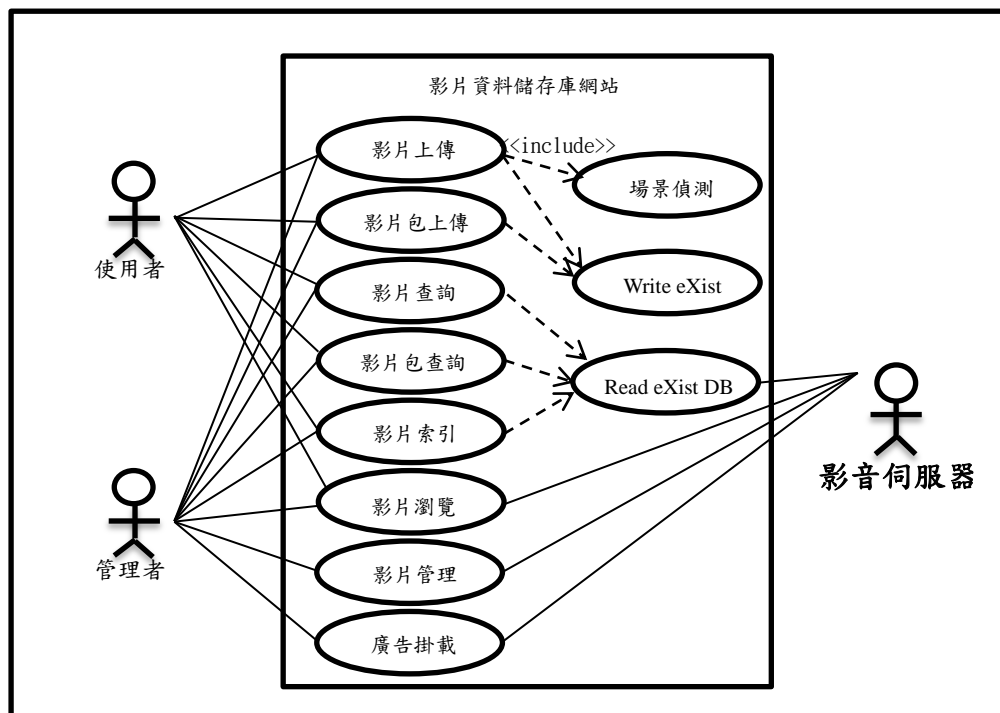


圖 4-2:使用案例圖

本小節利用 UML(United Modeling Language)的概念，說明從使用者的觀點來描述行為者為使用案例之間互動與關係，包含行為者和使用案例兩個元件，行為者就是與系統相關的人事物，在本系統中的行為者為使用者、管理者、影音伺服器，使用案例是完成某一特定工作的集合，本系統使用案例為影片上傳、場景偵測、讀寫資料庫、影片查詢等動作。行為者用人形符號表示，使用案例用橢圓形表示。行為者與使用案例以實線作連接，使用案例之間以虛線單向箭頭作連接。

如圖 4-2，本系統功能有影片上傳、影片包上傳、影片查詢、影片包查詢、影片索引、影片瀏覽，上述六個功能為使用者與管理者皆有權限使用，除此之外，還包含影片管理及廣告掛載兩個只有管理者才有權限使用的功能。以下為各個功能的簡易描述：

- **影片瀏覽:**瀏覽者可在此功能下觀看排呈廣播模式的影片內容。
- **影片上傳:**使用者或管理者透過影片上傳模組將所擁有的影片上傳至系統中，上傳過程中影片會先經過場景偵測處理，自動將影片索引成幾個章節，另外也會有將影片資訊寫入資料庫的動作。
- **影片查詢:**使用者可選擇查詢條件或手動輸入查詢資訊，透過查詢模組比對資料庫中的資料，將符合條件的影片結果列於 Silverlight 網頁中。
- **影片包上傳:**提供上傳一系列相關影片的功能，需先將系列影片壓縮成一個檔案才可上傳。
- **影片包查詢:**使用者輸入欲查詢的影片包資訊，查詢結果以影片包為單位呈現。
- **影片索引:**使用者可藉由此功能查詢影片片段，而不是整部影片，以章節為單位呈現結果，可同時找出相關索引條件的章節內容。
- **影片管理:**管理者才可使用的管理模組，可新增或刪除影片類別。
- **廣告掛載:**管理者才可使用的功能，可在影片前或後增加廣告影片。

4.2.2 循序圖(Sequence Diagram)

在使用案例圖做完後，可發現系統提供了那些功能，可是系統應該有哪些動作來完成這些功能，透過循序圖分析可以清楚了解每一個使用動作所觸發的事件及函式，並透過沿著時間軸一步一步將某個使用案例中，系統元件之間訊息交換表現出來。循序圖所包含的元素如表 4-2 所示：

| 元素 | 說明 |
|---------------------------|--|
| 角色 (Class role) | 每個物件在系統中扮演的角色，使用者、管理者、資料庫、JSP 網頁、Silverlight 網頁。 |
| 生命線 (Lifeline) | 每個角色物件存活的時間，以角色下延伸的虛線表示。 |
| 活動 (Activity) | 角色處理一項工作所需時間，以覆蓋虛線的方框表示。 |
| 訊息 (Message) | 角色之間溝通的訊息，以方框間的箭頭符號表示。 |

表 4-2:循序圖包含元素表

在本小節中，針對幾個重要流程以循序圖帶出其細節，包括影片上傳、影片查詢、影片索引、觀看影片章節、修改章節資訊標籤。

➤ 影片上傳

使用者選擇影片路徑、填寫影片資訊表單後，按下上傳按鈕，影片經過場景偵測程序回傳關鍵畫面及章節開始結束時間，系統依照填入及回傳資料製作成 XML 文件檔案，接著使用 eXist 提供的 API 將 XML 檔案存入資料庫，判斷使用者所輸入的影片類別資訊，依照 RDF/OWL 階層結構將影片 ID 記錄在所屬類別

中，並上傳影片檔案及關鍵畫面至儲存庫中。

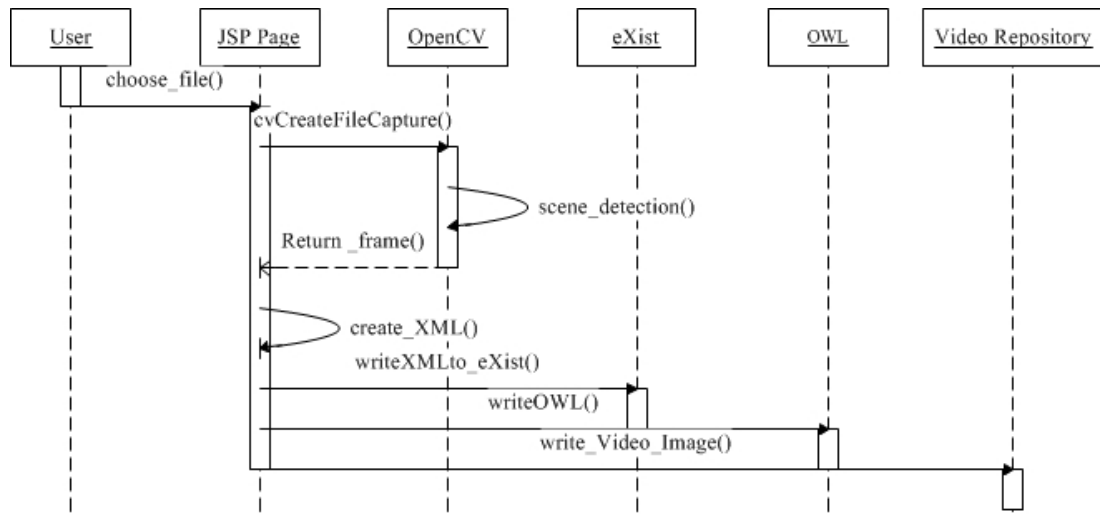


圖 4-3:影片上傳循序圖

➤ 影片查詢

當使用者選擇影片類別，查詢模組依照類別剖析 RDF/OWL 樹狀結構，回傳符合類別條件下的所有影片 ID，藉由此 ID 搜尋資料庫中的 XML 文件及儲存庫中的影片，最後結果顯示於 Silverlight 影片瀏覽系統，使用者即可於此處觀賞影片及影片資訊。

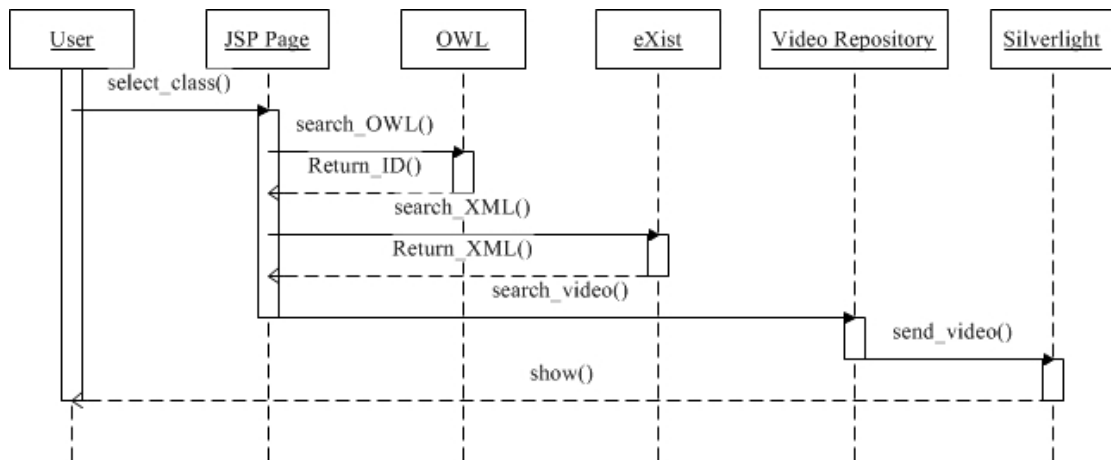


圖 4-4:影片查詢循序圖

➤ 影片索引查詢

使用者輸入關鍵字當作索引條件，系統依照關鍵字綜合搜尋，讀取資料庫中的符合資訊的 XML 檔案並回傳 ID 及 Clip 編號，依這兩項條件搜尋儲存庫中的影片，將結果顯示於 Silverlight 瀏覽系統中，使用者可於此頁面中選擇章節觀賞。

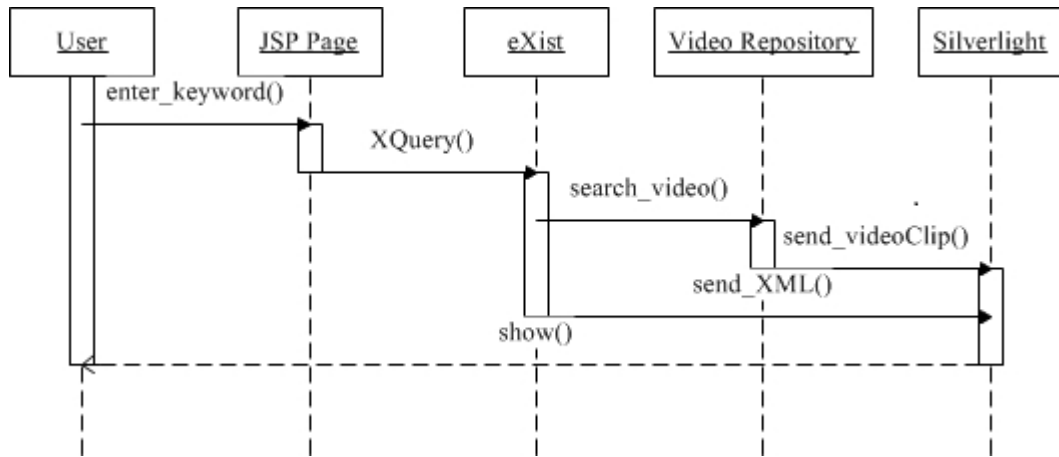


圖 4-5:影片索引查詢循序圖

➤ 修改章節資訊標籤

當系統顯示章節資訊於畫面中，使用者可以修改資訊內容並按下儲存按鍵，Silverlight 將章節內容傳送到 JSP 頁面，透過 JSP 新增一份 XML 文件，並寫入資料庫中，覆蓋舊的 XML 文件。

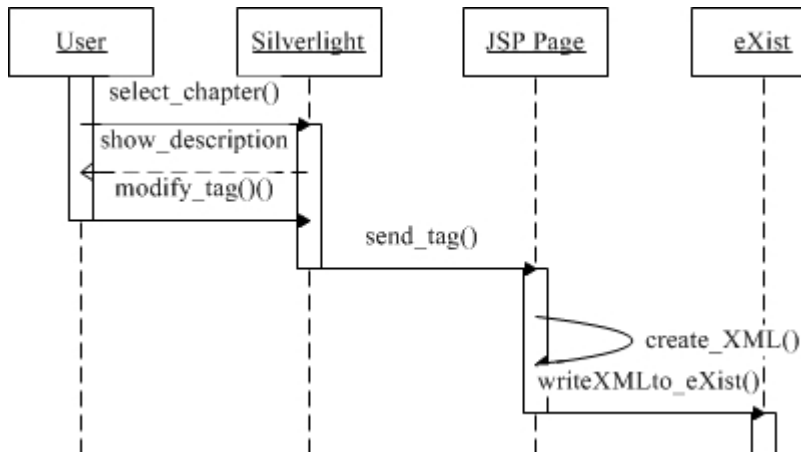


圖 4-6:修改章節資訊標籤循序圖

➤ 字幕同步

使用者在觀賞章節影片時，系統依照正在播放的章節，從資料庫中抓取相對應的字幕，隨著所設定的時間點顯示於畫面中。

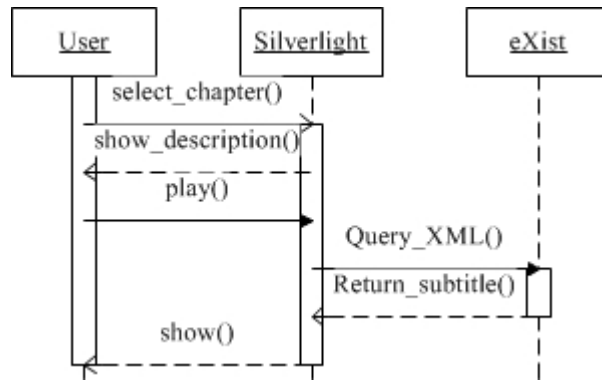


圖 4-7:字幕同步循序圖

4.3 系統呈現

4.3.1 影片上傳與自動化場景判斷

The screenshot shows a web application interface for video uploads. The title is "影片資料儲存庫" (Video Data Storage). The interface includes a navigation bar with options: "影片瀏覽", "影片上傳", "影片查詢", "影片包上傳", and "影片包查詢". The "影片上傳" (Video Upload) section is active. It contains the following fields and controls:

- 上傳檔案: C:\Users\Administrator\Desktop Browse...
- 影片名稱: 卡通英語教學1
- 影片長度: 00 時 01 分 31 秒
- 影片大小: 30.7 MB
- 影片來源: XML Lab
- 影片版權: XML Lab
- 影片格式: wmv
- 影片類別: 英語教學, 英文檢定, 公共電視
- 影片描述: 卡通版英語教學1-自動語音系統

At the bottom, there are "上傳" (Upload) and "清除" (Clear) buttons.

圖 4-8:系統影片上傳畫面

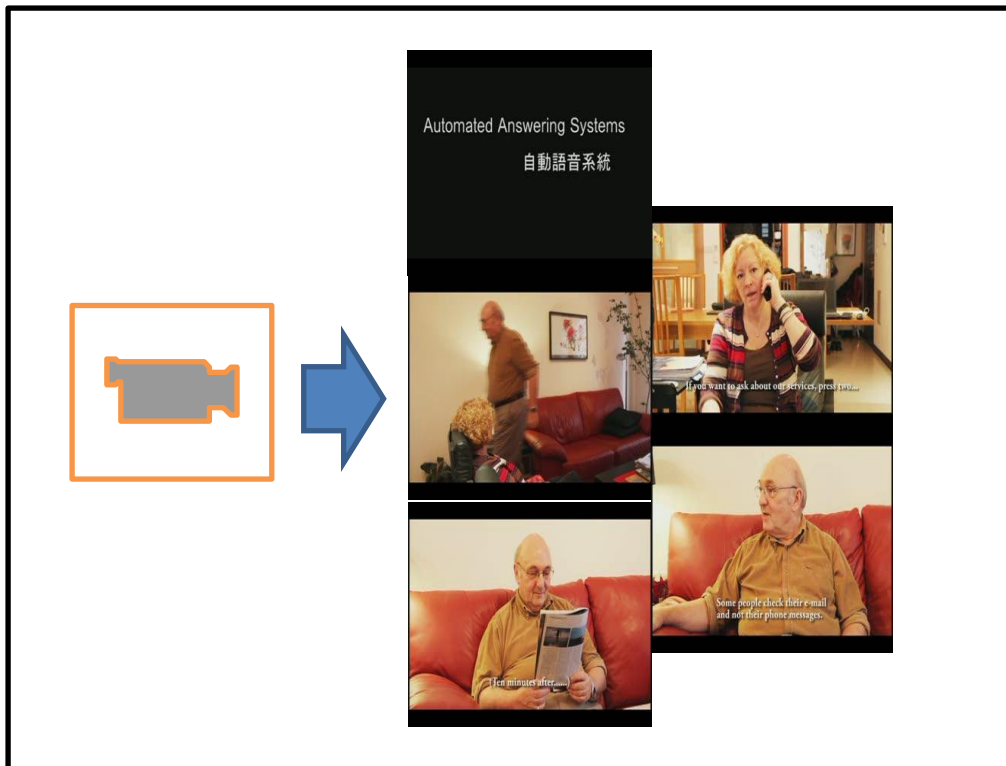


圖 4-9:系統影片場景偵測結果示意圖

1. **上傳檔案:**選擇一影片檔案，系統以檔案路徑作為來源。
2. **影片資訊:**上傳者填入影片的相關資訊，如圖 4-8，提供做為顯示於使用者參考的影片資料及搜尋影片時的條件，所填入資訊日後無法做修改。
3. **上傳:**確認無誤後按下上傳按鍵，系統依照表格產生 XML 文件並存入 eXist 資料庫，影片進入場景偵測程序，偵測結束後影片檔案也自動上傳至伺服器如圖 4-9。

4.3.2 影片查詢及章節呈現

影片查詢方式有三種，如圖 4-10 選擇一種查詢方式做查詢，輸入條件後按下查詢按鍵，並顯示結果於 Silverlight 主要頁面中，如圖 4-11。



圖 4-10:系統影片類別查詢畫面



圖 4-11:系統影片類別查詢結果畫面

左邊為查詢結果的影片排列，選擇其中一影片開始播放後，可透過檢視章節的按鈕(右箭頭按鈕)檢視目前播放影片的所有章節，如圖 4-12，檢視章節的頁面也提供一個播放器，右邊為所有章節的排列，選擇其中一個章節後播放即開始播放影片章節內容，若使用者想看整段影片則可馬上關掉此頁，系統立即回到主頁面，並保有剛剛使用者使用的狀態。

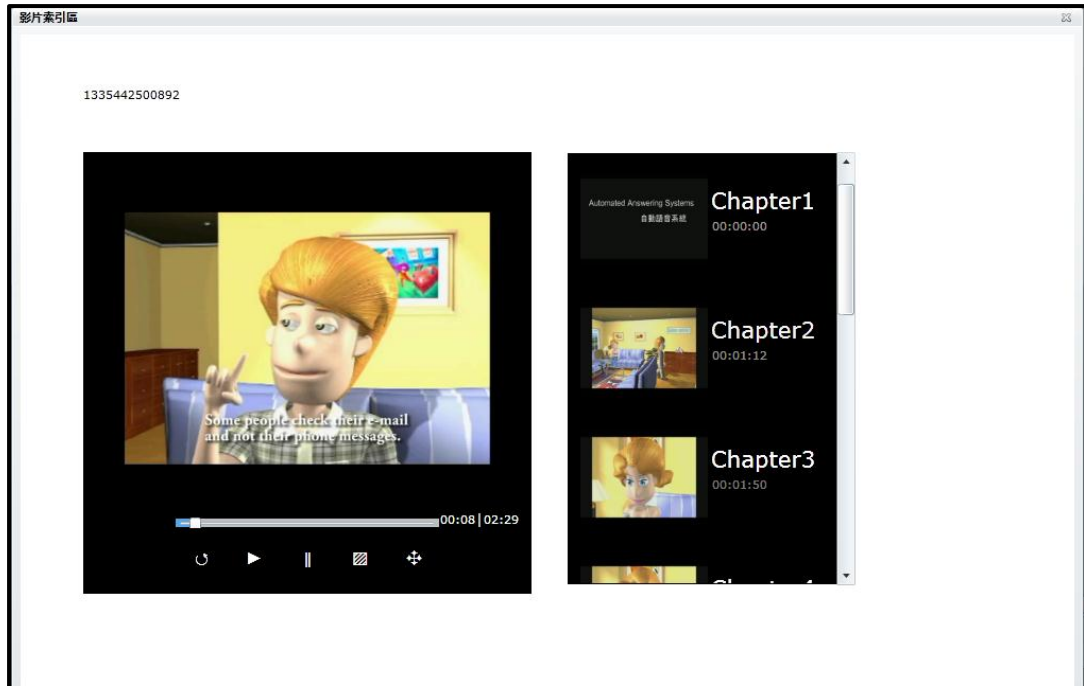


圖 4-12:系統影片章節呈現畫面

4.3.3 影片索引

使用者在主要頁面觀賞影片時，若覺得影片某段可作為一個章節，則使用者可以對影片做增加章節的動作，如圖 4-13，按下加號按鈕，則跳出章節編輯畫面，如圖 4-14，以拖曳的方式設定章節開始及結束時間。



圖 4-13:系統手動加入索引按鈕

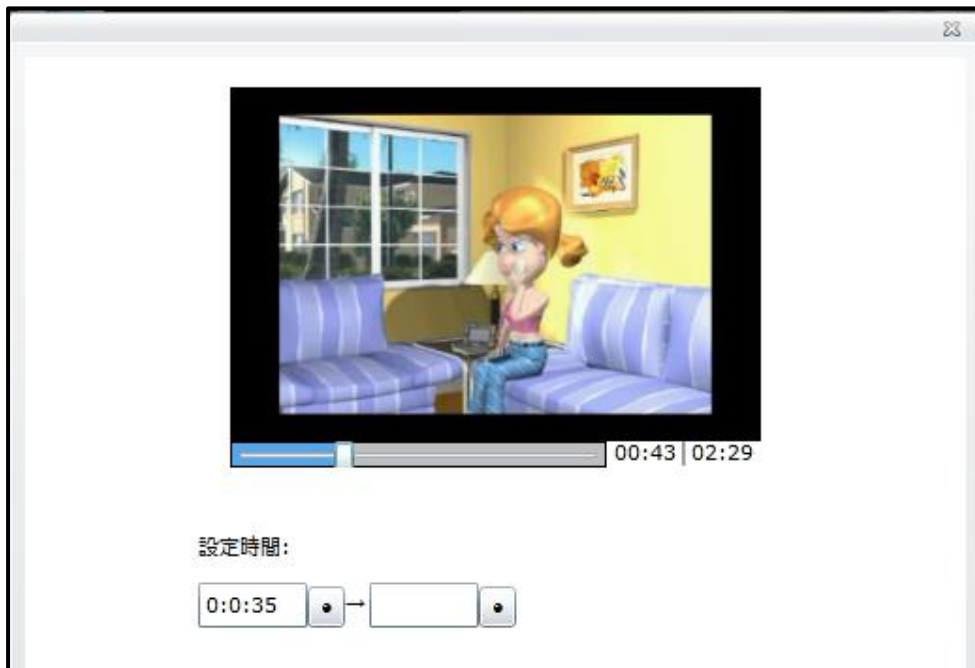


圖 4-14:系統手動索引畫面

4.3.4 字幕同步

字幕同步為使用者透過編輯，在特定時間點顯示於畫面中，如圖 4-15，而非

影片中的字幕，作用在於顯示單字或句子，使用者可以更快速了解影片教學重點，加強學習印象。



圖 4-15:系統字幕同步畫面

4.3.5 章節詳細資訊及標籤修改

點選章節排列區的其中一個章節，彈出小視窗呈現章節詳細資訊，並可修改資訊內容，此內容為日後索引此章節的重要查詢條件，所以當使用者對資訊描述得越詳細，日後查詢越準確。

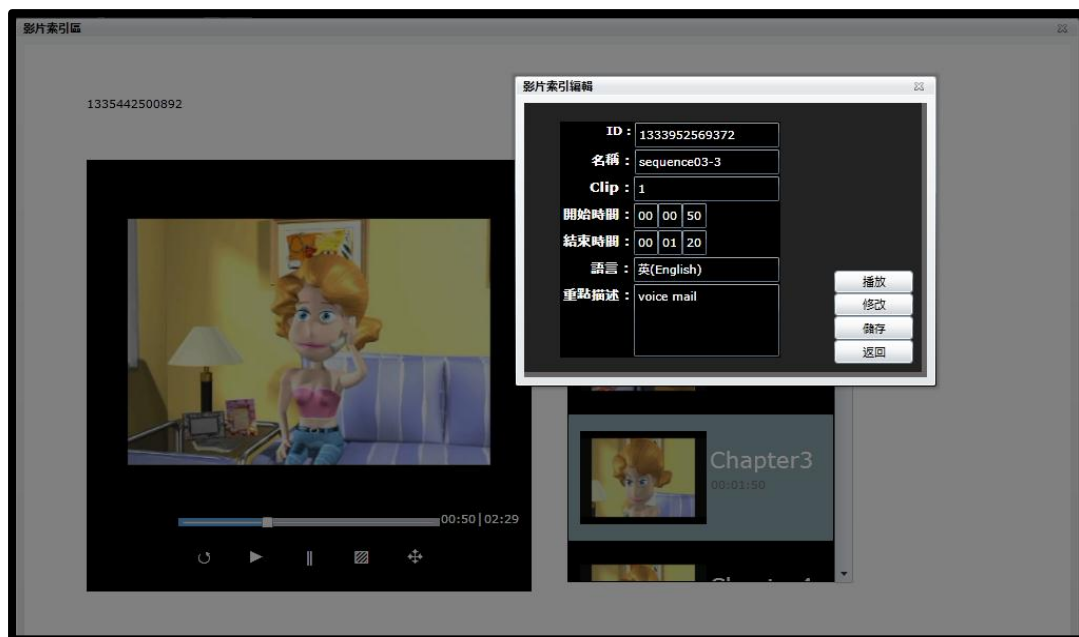


圖 4-16:系統章節資訊及標籤修改畫面

4.3.6 影片索引查詢

影片索引查詢為另一種查詢的方式，如圖 4-17 搜尋的結果以影片章節為單位呈現，輸入關鍵字後，按下查詢按鈕，系統導入 Silverlight 呈現結果，同樣的，此頁面也提供一個播放器，當使用者選擇位於頁面下方的章節縮圖，章節內容立即播放於播放器中。

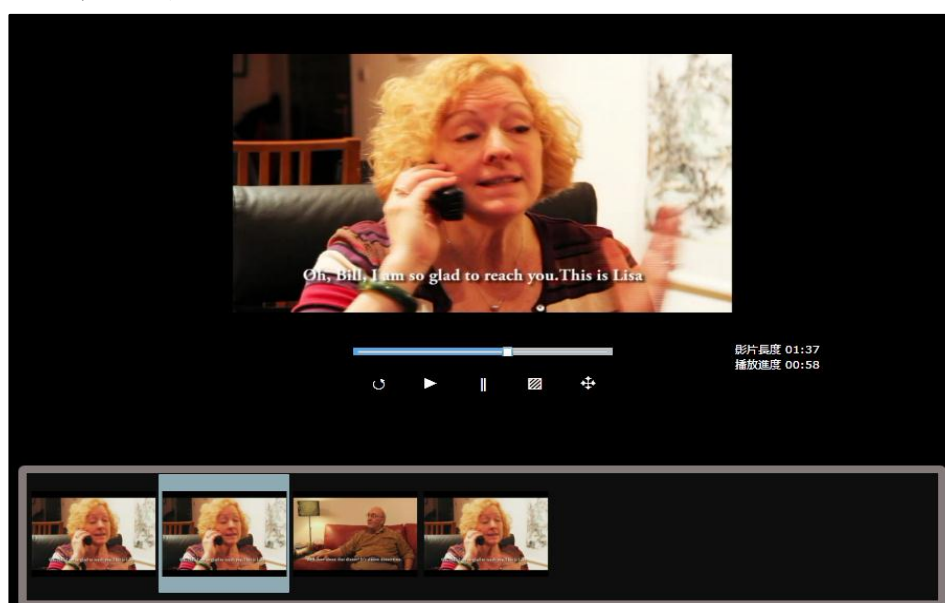


圖 4-17:系統影片索引查詢結果畫面

第五章 結論與未來發展

5.1 結論

在多媒體資源豐富以及網際網路開放的特性下，任何人都可以透過一個平台，分享各式各樣的資源，網路上的資源有千百種，其中以多媒體訊息最為豐富生動，也是人與人之間傳遞接收訊息最為直接的方式。本系統提供一個良好的影音平台及空間，透過簡單的使用者介面，影片擁有者可以上傳影片至影音系統中與他人共享，而使用者也可透過直覺化的介面搜尋及觀賞到資料庫中的多媒體影片。影片擁有者上傳的影片不分種類，在舊平台中對於大部分影片已提供良好的觀賞環境，本研究延續舊平台中所提供的瀏覽系統，透過改良，在教學影片的呈現做更深入的應用，提供了影片章節化、章節索引及搜尋、字幕同步等功能。授課者可藉由此平台上傳教學資源，學員透過生動的多媒體教材學習知識，影片章節化可以使原本冗長的教學影片有更清楚的呈現，學員可以直接搜尋重要的教學片段來觀看，而字幕同步可以提示學習時的重點，以互動及生動的方式呈現優良的學習環境，可加強學習效果及縮短學習的時間。因此，本研究不只提供了完整的管理規範，可有效管理日趨龐大的影片檔案，也提供了多樣化的功能，在影片的娛樂及教學表現更出色、更完整。

5.2 未來發展

➤ 行動版

目前智慧型手機發展快速，配合行動網路 3G、WiFi 等應用越來越多，若能發展手機版本，對於平台使用率會大大增加，且使用者可隨時隨地取得多媒體服務，達到隨時學習及各種娛樂等效果。也可搭配最近流行的 NFC 技術，透過 Tag 感應快速進入系統，或者搭配校園情境應用取得不同的學習影片內容，例如藉由感應 Tag 或者情境感知得知使用者目前位置，系統依照地理位置提供相關資訊或教學內容撥放，免去搜尋影片的時間，提高使用方便性。

➤ 查詢效率及準確度

隨著儲存的影片檔案越來越多，使用者輸入關鍵字所搜尋到的影片內容準確度也會下降，或者搜尋結果過多，使用者往往不能快速找到想要看的內容，未來可利用綜合搜尋的方式，使用者透過多重查詢條件的輸入，縮小搜尋範圍增加搜尋準確度。

➤ 個人化

提供個人化專屬服務，例如：專屬頻道、最喜愛影片、個人視頻直播等。

➤ 推薦系統

提供每日瀏覽次數前幾名影片、總瀏覽數排名影片、各類使用者最愛影片等，或者使用者可對於影片作評分及等第，增加影片人氣等功能，增加使用者之間的交流。

➤ 分散式系統及雲端

運用分散式系統的概念，資料儲存於不同的資料庫中，且資料庫之間可以做交流及互補，減少資源浪費，並且利用雲端服務，類似備份資料，不會因為伺服器的故障而影響使用者。

參考文獻

- [1]. Matthew MacDonald, "Pro Silverlight 4 in C#" (2010)
- [2]. eXist-db Open Source Native XML Database. from:
<http://exist.sourceforge.net/>
- [3]. Chueh-Wei Chang, Kuan-Ying Li. A Representative Images Selection Method for Videos of Multiple Scenes
- [4]. W3C Recommendation (2007, Jan). XQuery 1.0: An XML Query Language. Retrieved from <http://www.w3.org/TR/xquery/>
- [5]. W3C Recommendation. (2007, Jan). XML Path Language XPath 2.0. Retrieved from <http://www.w3.org/TR/xpath20/>
- [6]. Silverlight. Microsoft. from: <http://www.microsoft.com/silverlight/>
- [7]. W3C Candidate Recommendation (2009, Jan). XQuery Update Facility 1.0. Retrieved from <http://www.w3.org/TR/xquery-update-10/>
- [8]. Efficient XML Interchange for Rich Internet Applications © 2009 IEEE
- [9]. Content-based Image Indexing and Retrieval with XML Representations. Proceedings of 2004 International Symposium on Intelligent Multimedia. Video and Speech Processing October 20-22.2004 Hong Kong
- [10]. Tim Lammarsch, Wolfgang Aigner, Alessio Bertone, Silvia Miksch, Thomas Turic, Johannes Gartner. A Comparison of Programming Platforms for Interactive Visualization in Web Browser Based Applications. 12th International Conference Information Visualisation
- [11]. Xiaochun Zhou, Chenlong Shao. The Design of Virtual Meeting System Based on Silverlight © 2010 IEEE
- [12]. M. J. Pickering and S. Ruger, "Evaluation of Key Frame-based Retrieval Techniques for Video," Computer Vision and Image Understanding, Vol. 92, No. 2-3, pp. 217-235, Nov.-Dec. 2003
- [13]. Engin Mendi, Coskun Bayrak. A Web-based Medical Video Indexing Environment. 2010 IEEE Fourth International Conference on Semantic Computing
- [14]. Li-Wen Chen. An Event-based Video Retrieval System by Combining Broadcasting Baseball Video and Web-casting Text. (2010)
- [15]. Chien-Hsing Lee. Video Indexing and Retrieval Using MPEG-7 Motion Activity Descriptors. (2005)
- [16]. 涂長助 以 XML 原生資料庫實現影音管理儲存庫之研究. 國立臺灣師範大學資訊工程研究所碩士論文.(2009)
- [17]. 章立民 Silverlight 3.0 全面精通手冊 (初版 ed.). 台北市:基峯資訊股份有

- 限公司. (2009, Sep)
- [18]. 陳會安 JSP 2.0 網頁設計範例教本 (初版 ed.). 台北市:學貫行銷股份有限公司.(2005, Feb)
- [19]. 簡孟軍 以 XML 原生資料庫及 Silverlight 網頁技術實現影音管理儲存庫之研究. 國立臺灣師範大學資訊工程研究所碩士論文.(2010)
- [20]. 井民全 簡單的數位影像處理 (C# 篇) Image-Processing Using C-Sharp