

國立交通大學

資訊學院 資訊學程

碩士論文

多媒體教材內容製作之驗證方法實作 —
以 CMMI 驗證流程領域為基礎

**A Verification Method for Multimedia Content Creation
Based on Verification Process Area of CMMI**



研究生：陳信江

指導教授：陳登吉 教授

中華民國九十八年六月

多媒體教材內容製作之驗證方法實作 —
以 CMMI 驗證流程領域為基礎
**A Verification Method for Multimedia Content Creation
Based on Verification Process Area of CMMI**

研究生：陳信江

Student：Chen-Hsin Chiang

指導教授：陳登吉

Advisor：Deng-Jyi Chen

國立交通大學
資訊學院 資訊學程
碩士論文



Submitted to College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Computer Science

June 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年六月

多媒體教材內容製作之驗證方法實作 — 以CMMI驗證流程領域為基礎

學生：陳信江

指導教授：陳登吉 教授

國立交通大學 資訊學院 資訊學程碩士班

摘 要

多媒體教材開發的過程中，如何確保最終教材成品與需求規格的一致性關係著整個教材品質的重要關鍵。多媒體教材的教材品質主要來自於產品內容和製作程序。其中產品內容主要源自於設計者的創意與巧思。而良好製作程序則須經由系統化的驗證方法在製作過程中加以控管，以保證教材品質符合需求。

由於多媒體教材同時具有數位教材與電腦軟體的雙重特性，非常適合以軟體工程的理論基礎來管理實作教材軟體開發流程，如此可以確保開發出的教材符合產品需求規格。而 CMMI 的驗證流程領域則定義了標準的流程、制度以及量化資訊，作為軟體開發的流程標準，進而提升軟體品質。因此本研究以 CMMI 的驗證流程領域為基礎結合軟體工程的驗證理論，提出解決產品與需求規格不一致的問題的驗證方法。並且實作出符合 CMMI 規範的多媒體教材驗證管理系統。

在本論文中，我們將分析多媒體教材的製作生命週期，從教材開發階段開始到產品製作完成的過程中結合驗證管理機制，協助教材製作者經由標準化的開發作業流程製作教材，並透過適當的驗證審核步驟來控管產品品質，以達到產品和需求規格相符合的一致性目標。

關鍵字：驗證、確認、能力成熟度

A Verification Method for Multimedia Content Creation Based on Verification Process Area of CMMI

Student : Chen-Hsin Chiang

Advisor : Dr.Deng-Jyi Chen

Degree Program of Computer Science
National Chiao Tung University

ABSTRACT

In the process of developing multimedia courseware, the key factor is to ensure the consistency between the final courseware and the requested standard. The quality of the multimedia courseware comes from the content of the product and the producing process. In which the content of the product is originated from the designer's creativity and ingenuity. And the good producing process must be controlled by using systematically verification method in the process of producing, to ensure the courseware reaches the requested standard.

Since the multimedia courseware contains the dual quality of both the digital courseware and computer software. It is very suitable to apply the software engineering theory as base to control the development for the standard operating procedure (SOP) of the software. In order to ensure the developed courseware reaches the requested standard. CMMI Verification process area defines the standard procedure, regulation and quantification data. It can be used as standard procedure for software development and hence improved the quality of the software. Thus this study applies verification procedure of CMMI as base together with software engineering verification theory. The study finds out the solution to solve the problem of inconsistency of the product and the requested standard and work out the verification management system for multimedia courseware which is qualified to CMMI regulation.

In this study, we will analyze life cycle of the multimedia courseware, combine verification management system through beginning stage of development to the final process of the finishing product. We will also use the standard operating procedure to help the engineer in the process of developing courseware. Furthermore we will control the quality of the product through suitable verification method in order to reach the consistency between the product and requested standard.

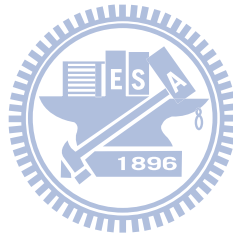
Keywords : Verification, Validation, Capability Maturity

誌謝

感謝指導教授 陳登吉老師的耐心指導與教誨，老師不但在研究上給予我們方向與指引，也提供我們許多的研究資源，本論文得以完成，老師給予我許多寶貴建議，在此致上對老師無限的感謝。

另外也感謝本論文的口試委員們對本論文提出的建議，有了各位老師的指導，才能使得本論文與研究更加完善。

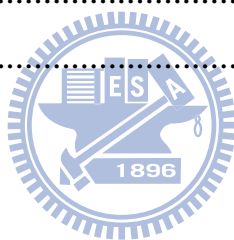
最後，感謝軟體工程實驗室的同學、學長姊、學弟妹們，在研究期間給予我許多課業和生活上的協助以及共同成長的機會，謝謝你們。



目錄

摘要.....	iii
ABSTRACT.....	iv
誌謝.....	v
目錄.....	vi
圖目錄.....	viii
表目錄.....	i
一、緒論.....	1
1.1 研究背景與動機.....	1
1.2 研究目的.....	2
1.3 研究方法.....	3
1.4 章節概要.....	4
二、文獻探討.....	5
2.1 共享內容物件參考模型(SCORM).....	5
2.1.1 教學素材(Assets).....	6
2.1.2 教材包裹(Content Package).....	7
2.1.3 內容組織(Content Organization).....	8
2.1.4 共享內容物件(Sharable Content Object).....	9
2.1.5 執行環境(Run Time Environment).....	10
2.2 軟體品質.....	11
2.3 能力成熟度整合模式(CMMI).....	12
2.3.1 CMMI 歷史沿革.....	13
2.3.2 CMMI 表述模式.....	14
2.3.3 能力度與成熟度.....	15
2.3.4 流程領域.....	17
2.3.5 驗證流程領域(Verification process area).....	19
2.3.6 預期效益.....	20
三、系統分析與流程設計.....	22
3.1 CMMI 驗證流程領域分析.....	22
3.2 功能性需求與非功能性需求.....	25
3.3 多媒體教材製作生命週期.....	27
3.4 多媒體教材單元的相依性關係.....	30

3.4.1 多媒體教材製作與非循環有向圖.....	31
3.4.2 多媒體教材單元資料結構與演算法.....	34
四、系統實作與導入.....	42
4.1 系統平台與開發工具.....	42
4.2 MVMS 系統架構.....	45
4.3 系統製作.....	47
五、實例應用與分析.....	57
5.1 驗證準備.....	57
5.2 同仁審查.....	62
5.3 驗證執行.....	64
5.4 矯正措施界定.....	65
5.5 系統導入成效.....	66
5.6 系統導入問題探討.....	67
六 結論與未來研究方向.....	74
6.1 結論.....	74
6.2 未來研究方向.....	75
參考文獻.....	76



圖目錄

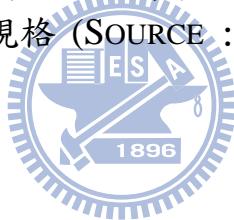
圖 1： FILE TYPES OF ASSETS	7
圖 2： CONTENT PACKAGE	8
圖 3： CONTENT ORGANIZATION	9
圖 4： SHARABLE CONTENT OBJECT	10
圖 5： RUN-TIME ENVIRONMENT 概念示意圖	11
圖 6： CMMI DERIVATION	13
圖 7： 連續式與階段式表述的結構	14
圖 8： 流程領域與相關的模式組件	18
圖 9： 靜態和動態的驗證與確認	20
圖 10： 驗證流程四大模組示意圖	25
圖 11： 多媒體教材開發生命週期流程圖	29
圖 12： 多媒體教材開發週期非循環有向圖	33
圖 13： 多媒體教材節點建構子虛擬碼	34
圖 14： 節點搜尋算法虛擬碼	35
圖 15： 複製節點功能演算法程式碼	36
圖 16： 加入新節點到 GRAPH 流程圖	37
圖 17： GRAPH 的初始狀態示意圖	38
圖 18： 加入 SNE1 到 GRAPH 的狀態示意圖	38
圖 19： 加入 NLS1 到 GRAPH 的狀態示意圖	38
圖 20： 加入 NLS2 到 GRAPH 的狀態示意圖	39
圖 21： 加入 NLS3 到 GRAPH 的狀態示意圖	40
圖 22： 加入新節點到 GRAPH 演算法程式碼	41
圖 23： MVMS 系統環境平台	42
圖 24： MVMS 系統資料傳遞示意圖	43
圖 25： 典型多媒體教材品質管理系統	44
圖 26： MVMS 系統架構	45
圖 27： MVMS 類別圖	46
圖 28： 多媒體教材靜態檢查程序	48
圖 29： 多媒體教材靜態檢查使用案例圖	48
圖 30： 多媒體教材動態驗證單元	49
圖 31： 同仁審查使用案例圖	50
圖 32： 同仁審查序列圖	52
圖 33： 驗證執行活動圖	54
圖 34： 矯正措施作業活動圖	56

圖 35：多媒體教材 “WHAT IS ON THE MAT?”	58
圖 36：劇情三素材對應圖	58
圖 37：CONTENT ORGANIZATION 對應關係.....	59
圖 38：建立教材主要單元結構畫面	60
圖 39：建立素材清單畫面	60
圖 40：教材 “WHAT IS ON THE MAT?” 非循環有向圖	61
圖 41：建立驗證程序執行畫面	62
圖 42：同仁審核確認執行畫面	63
圖 43：編續教學審核畫面	63
圖 44：劇情單元驗證進行	64
圖 45：單元驗證清單執行畫面	65
圖 46：矯正措施執行畫面	66
圖 47：教材 LEC001 錯誤項目統計柱狀圖 (BAR CHART)	67
圖 48：多媒體教材範例	68
圖 49：多媒體教材範例不一致性項目盒圖	69
圖 50：教材 LEC001 不一致性項目柏拉圖	70
圖 51：演員屬性與演員數目關係曲線	71



表目錄

表 1：連續式與階段式表述的差異 (SOURCE：CMMI-DEV V1.2)	15
表 2：能力度與成熟度等級的比較 (SOURCE：CMMI-DEV V1.2)	17
表 3：驗證流程領域連續式表述之特定目標及執行方法	19
表 4：MVMS 系統功能性需求表	26
表 5：MVMS 系統非功能性需求表	27
表 6：分鏡腳本範例	31
表 7：多媒體教材驗證準則	47
表 8：同仁審查原則	50
表 9：多媒體教材審查時程	51
表 10：常見錯誤型態	53
表 11：缺失改善分析步驟	53
表 12：多媒體教材範例驗證數據表	68
表 13：動畫演員屬性規格 (SOURCE：編輯手提供)	72



一、緒論

1.1 研究背景與動機

由於資訊科技的進步，使得人們學習知識的方式也跟著被資訊化，而個人電腦的普及化加上網際網路興起，更突破了時間與空間的限制，使得學習不再是單純的學生和老師之間人的組合，而變成是學習者和學習教材之間的互動，在這當中多媒體教材無疑的扮演了最重要的角色。當越來越多的多媒體教材取代了傳統學習中老師的地位之餘，多媒體教材的製作品質問題也就再也無法被忽略。

多媒體教材的品質問題中，不一致性是影響教材品質的最關鍵因素，歸納多媒體教材的不一致性主要有下列三種：

一、產品與需求規格不符的問題

在製作多媒體教材的過程中，經常發生製作出來的最終成品與原始需求規格不符的問題，教材製作者不是為此修改規格遷就教材成品，就是反覆修正教材直到符合規格，倘若不一致情況嚴重，甚至必須重作教材，造成時間、成本上的浪費。

二、教材模組之間整合的問題

在多人協同作業的多媒體教材開發環境下，因為缺乏標準化的開發作業流程及良好的驗證規範，所造成不同的開發成員所設計的教材模組相互間產生不一致的衝突或互斥的現象以至於無法順利整合，造成開發時程的延誤以及開發成員士氣的低落。

三、教材模組本身品質不一致問題

目前所提出的各種多媒體教材標準中，以 ADL 組織制定的 SCORM (Sharable Content Object Reference Model) 標準，最受到重視。SCORM 規範了數位教材的內容與格式，實現了教材資源共享的理念，但卻無法控管數位教材的品質，使得即使製作的教材符合了 SCORM 的規範，卻依然可能有教材模組間品質不一致的情形發生。

多媒體教材的教材品質主要來自於產品內容和製作程序。其中產品內容主要源自於設計者的創意與巧思。而良好製作程序則須經由系統化的驗證方法在製作過程中加以控管，以保證教材品質符合需求。

由於多媒體教材除了具備學習教材的特性之外，本身也是屬於電腦軟體產品的一種，而 CMMI 的驗證管理流程領域 (verification process area) 在特定目標中提出了特定方法可以用來達成此一致性目標。因此本研究將以 CMMI 的規範為基礎，並整合軟體工程中的驗證與確認理論提出一套多媒體教材管理驗證方法並實作出來以確保所開發出來的教材產品與需求規格的一致性。

1.2 研究目的

本研究的主要目的在提出一套以 CMMI 驗證流程領域規範為基礎並整合軟體工程的驗證理論的多媒體教材驗證管理方法，設計並實作出多媒體教材驗證系統。其主要目的分為四部份：

一、符合 CMMI 軟體品質規範

依據 CMMI 驗證流程領域 (Verification Process Area) 所規範的特定目標 (Specific Goal, SG) 與特定執行方法 (Specific Practice, SP) 為主要依據，設計多媒體教材驗證管理系統，協助達到產品與規格的一致性。

二、可以整合到多媒體教材的需求管理系統

本系統可以整合到多媒體教材的需求管理系統中配合審查機制，根據多媒體教材開發生命週期，在製作過程中透過階段驗證的方式，驗證每個階段是否符合系統的需求規格。藉此方法來確保產品的正確性，提高多媒體教材的品質。

三、與多媒體教材編輯軟體整合

由於多媒體教材製作者通常為教育工作人員，其設計教材重點在於教材的內容知識性、學習性及創意上，對於需求規格制定、文件格式等並不擅長。這種情況下容易因為開發人員的認知差異，造成開發出來的多媒體教材產品無法完全符合原始規格的需求。本系統提供從分鏡腳本、劇情、演員、場景等相關參數的規格檢測及回溯驗證機制。最終透過輸出 XML 格式文件可以作為符合 SCORM 規範的不同多媒體教材製作系統之間一個一致性的內容格式轉換平台。

四、結合統計學方法協助找出不一致性原因

本系統結合統計學理論方法，針對已知教材樣本數提供錯誤項目分佈情形以及關鍵錯誤項目等統計圖表，協助找出造成不一致最主要項目原因，作為系統未來改善參考依據。

1.3 研究方法

為了達到多媒體教材製作的一致性要求，本研究首先以 CMMI 的規範為基礎，有效的控管因為變更原始設計所造成的不一致性（Inconsistency）品質問題，設計品質驗證與檢測機制。使系統能在原始設計變更發生時，適時的提供教材審核委員與製作人員，相關受到該項設計變更對於整體多媒體教材所造成影響範圍的資訊，協助教材製作人員採取適當矯正措施，使得多媒體教材最終成品能符合產品需求規格。因此本研究方法可分為下列步驟來進行：

一、相關研究與文獻探討：

由於多媒體教材的最終成品是以電腦軟體的型態呈現。因此，若要從多媒體教材開發流程中依階段逐步進行驗證程序，就必須先依照軟體工程的開發方法論中，分析軟體開發生命週期各階段，因為設計錯誤或變更所導致對整個軟體專案開發所造成的影響。從中找出最適合改進的步驟方法，並結合符合 SCORM 標準的媒體數位教材的架構與開發流程，作為系統設計模型基礎。

二、驗證系統模型設計：

針對 CMMI 驗證流程領域所提出的特定目標與特定方法要求步驟，評估並設計符合 SCORM 數位教材架構規範的驗證系統模型，分析其所需要定義與規劃的多媒體教材開發流程品質驗證機制以及架構。並依據此系統模型設計在一個多人同時線上作業的網路平台執行環境的多媒體教材驗證管理系統（Multimedia Verification Management System ,MVMS）。

三、系統設計與製作：

根據 MVMS 模型以及 CMMI 的驗證流程規範，以 Server-Client 的架構來設計 MVMS。MVMS 的功能主要分為四大系統模組與十項系統功能，依照 CMMI 的驗證流程規範的功能要求，詳細定義 MVMS 的系統整體架構與各項功能模組的運作機制與規格。同時定義 MVMS 進行驗證作業的作業程序，並以一個數位教材的實例，來展示 MVMS 系統各項功能與多媒體教材開發流程驗證步驟標準作業程序。

四、實例應用分析：

根據所定義的多媒體教材開發流程驗證管理作業程序，以一個多媒體教材開發範例，展示在多媒體教材驗證管理系統（MVMS）進行數位教材品質驗證的作業程序與 MVMS 的各項模組功能，以驗證 MVMS 模型的可行性與適用性。

五、結論與建議：

在此歸納研究心得並提出一些對後續研究、發展方向的建議。本研究是依據國際性軟體開發品質標準 CMMI 以及數位教材開發規範 SCORM 為基礎，設計 SCO 層次的多媒體教材開發流程的品質驗證方法，對於驗證後的教材內容具有符合需求規格的高品質特性。本驗證機制可應用至 SCORM 課程的多媒體教材開發上。相關建議將詳述於第六章。

1.4 章節概要

本論文共分五個章節，其內容簡述如下：

第一章「緒論」，說明本論文研究機動的描述與希望達成之研究目的，並對論文章節做簡單的介紹。

第二章「相關文獻探討」，探討與本研究相關的背景知識與文獻，如 CMMI 規範與軟體工程標準。

第三章「系統分析與流程設計」，說明依據 CMMI 規範為基礎，以驗證管理流程設計多媒體教材驗證管理系統。

第四章「系統製作與導入」，依據第三章所分析出來的需求與流程，提出系統的架構並作細部的功能設計。規劃在網路環境下，以主從架構 (Server-Client) 開發多媒體教材驗證管理系統。


第五章「實例應用與分析」，以一個實際多媒體教材案例，展示本系統執行成效。

第六章「結論與未來展望」，將對本論文做完整之總結，並提出可以繼續改進之方向與建議以及未來之發展趨勢。

二、文獻探討

2.1 共享內容物件參考模型(SCORM)

目前國際上已有許多組織提出電子媒體教材的相關標準，例如：歐洲的 Alliance of Remote Instructional Authoring and Distribution Networks for Europe project (ARIADNE)，IEEE 的 Learning Technologies Standardization Committee (LTSC)，美國國防部的 Advanced Distributed Learning initiative (ADL)。其中又以美國國防部 ADL 的 SCORM 最受重視並被大多數國家教育單位及多媒體教材製作廠商引用為遵循規範[1] [2] [3]。實際上 SCORM 採用了許多 IEEE LTSC，IMS，AICC 及 ARIADNE 等的規格。



美國國防部所推動的先進分散式學習(ADL) 先導計畫所制定的 SCORM (Sharable Content Object Reference Model) 標準整合了以上的標準，希望透過「教材再用與共享機制」的建立，來縮短教材開發時程、減少教材開發成本、促成教材能在各學習平台間共通，並能紀錄歷程檔案，因應隨時更新教學之需求。其主要優點如下[3]：

1. 可重用性 (Reusability) — 在不同應用環境下，學習內容或學習物件可以重複使用。
2. 取得容易性 (Accessibility) — 學習者可以在任何時間、任何地點獲取適當的學習內容。
3. 耐久性 (Durability) — 學習物件不會因為科技或標準提升或改變而無法使用。
4. 可互通性 (Interoperability) — 教育資源可以在不同平台上呈現，或利用其他工具重新編輯。
5. 可負擔性 (Affordability) — 增加效益及產出，減少教材開發的時間與成本。
6. 適應性 (Adaptability) — 可以依據學習者的學習情況，適時地調整學習內容。

SCORM 規範的組成包含以下幾個部分

- (1) 總覽 (SCORM Overview)：包含 ADL 計畫的完整介紹，以及 SCORM 的技術規範概要。
- (2) 內容聚合模型 (Content Aggregation Model, CAM) [4][5]：CAM 的目的在於定義如何識別、描述各類學習內容，並將相關的學習內容彙集成一個課程，且能在不同 LMS(Learning Management System)[6] 之間使用。
內容聚合模型是由以下組成的：
 1. 內容模型 (Content Model)：定義了一次學習體驗的內容元件的命名。
 2. 內容包裝 (Content Packaging)：定義了一次學習體驗的固定動作 (內容結構) 以及如何在不同的環境中組合學習資源的活動 (內容打包)。
 3. 詮釋資料 (Meta-data)：描述 SCORM 組成部分的說明和要求
 4. 編序和導覽 (Sequencing and Navigation)：定義編序和導覽資訊的說明和要求，描述了活動的固定次序。
- (3) 執行環境 (Run-Time Environment, RTE)：主要引用自 AICC (Aviation Industry CBT Committee) 對於互通性 (interoperability) 的規範，其內容說明如何利用 API (Application Program Interface) adaptor 建構學習者與 LMS 之間的執行環境。利用 RTE 所定義的 API 與 data model 可以使 LMS 記錄學習者的學習狀態，並根據這些學習狀態決定後續需要提供給學習者的學習內容，或是作為改善教學內容的依據。

2.1.1 教學素材(Assets)

Asset 在 SCORM 中被定義為最小且基本的學習資源，中文稱做教學素材。Asset 是一種數位化的媒體的統稱，包含聲音、圖形、影像、文件檔等可藉由全球資訊網 (WWW) 呈現給學習者，也就是說 Asset 必須是能在使用者的網際網路瀏覽器(web browser) 端運行和展示的電子媒體型態文件，如圖1 所示。一個 Asset 可以是由其他多個 Assets 所構成，並且每一個 Asset 皆有所屬的 Asset Meta-data 輔助說明其內容，以利於搜尋，並達到重複使用目的。Asset 和其他 Asset 的聯繫是依靠教材包裹 (Content Package) 來達成。Asset 和共享內容物件 (Sharable Content Object ,SCO) 最大的不同處在於 SCO 是與學管理平台(Learning Management System ,LMS) 進行溝通的,並且由 LMS 的觀點來看 SCO 是一個最基本不可分割的學習資源。

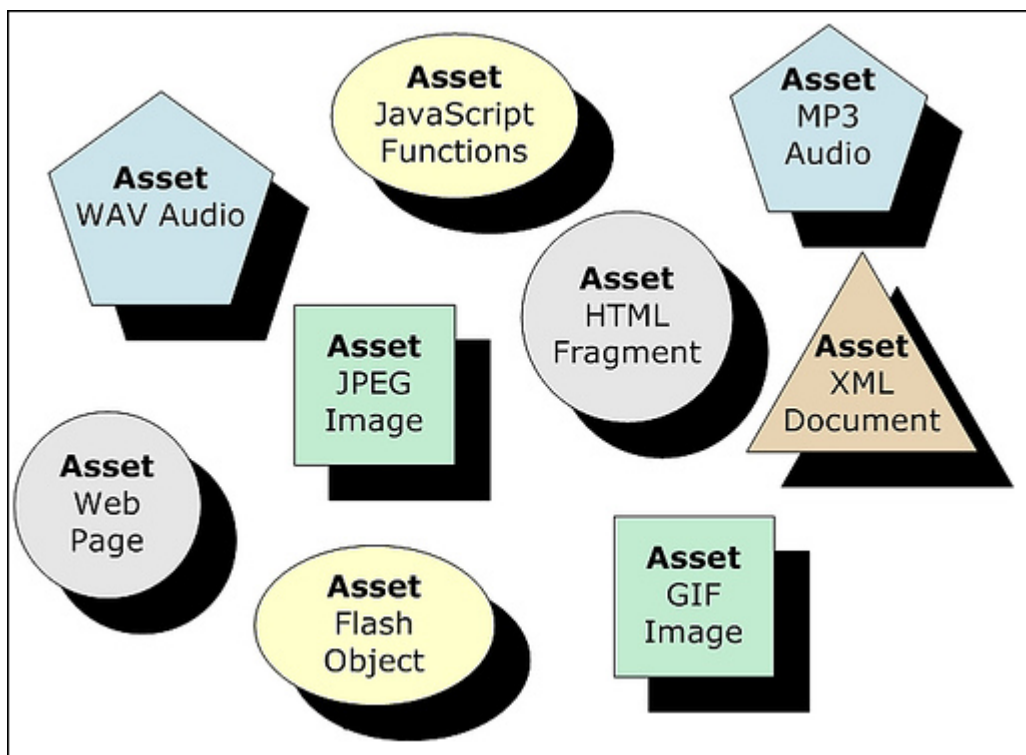


圖 1 : File types of Assets (Source: ADL, 2004)

2.1.2 教材包裹(Content Package)

Content Package (CP) [6] 的設計是爲了在不同的數位學習平台系統之間交換彼此的學習資源。Content Package 所構成的要素，如圖2 所示。其中包含了「Manifest」與「Physical Files」這兩個要素。

- **Manifest**：是一個 XML-Based Manifest 檔紀錄，檔名 imsmanifest.xml，存在 zip 檔內用於表示組織與資源，以描述課程結構與包裹的學習資源，在 Organization 區段，定義課程教材架構和行為。在 Resource 區段，在教材包裹中所使用到的資源列表。
- **Physical Files**：所有相關的實體檔案，包括課程內容、多媒體素材等等。內容套件除提供標準化規格外，也管理物件額外資訊與使用者界面的控制，並提供 LMS 如何處理內容包裹的使用說明。

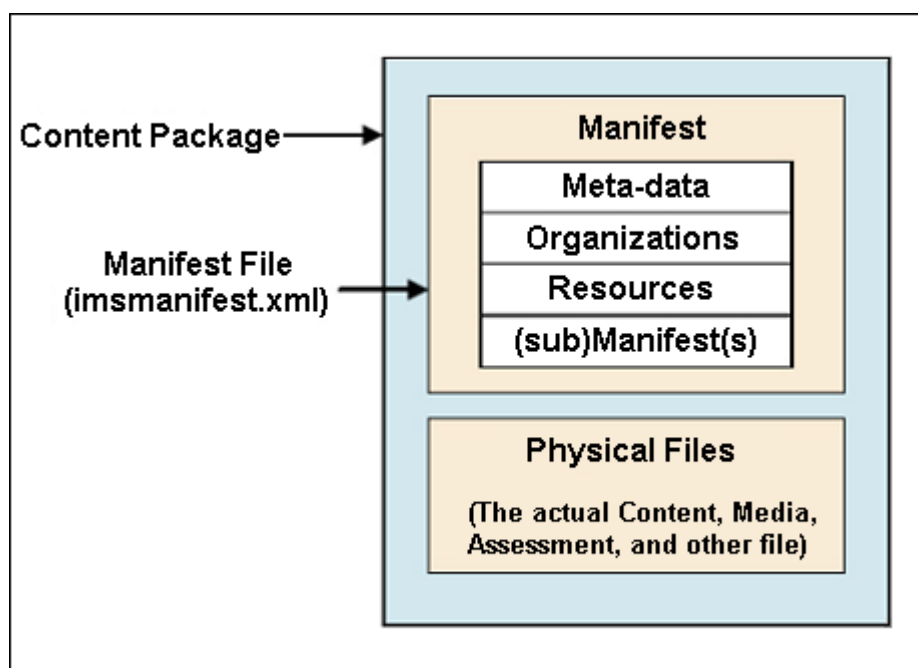


圖 2 : Content Package (Source: ADL, 2004)

2.1.3 內容組織(Content Organization)



SCORM SN 規格書中描述內容組織 (Content Organization) 主要是將教學內容以巢狀的階層性結構圖描繪出各單元，以顯示預期使用的教材及一個活動與其他活動之間的相關聯性，如圖3 所示。內容組織結構中呈現的活動可以由其他學習活動(學習子活動)組成，而它們自身又可以包括其他學習活動。Content Organization 是由 Asset 或 SCO 組成項目 (Item)，每個 Asset 或 SCO 可以稱為一個學習資源 (Resource) ，而每個項目代表一個學習活動，可能是一個課程、或一個模組等。因為 Content Organization 是提供教材設計者能夠開發一套結構化的教學內容(或學習活動) ，所以 Content Organization 可被視為是學習資源的結構化地圖。這樣的教學內容(或學習活動)亦可視為一個教材，用來指引學習者透過結構化的學習活動來使用學習資源。因此導入教學設計者的經驗所開發的教學內容(或學習活動)都可視為是教材，這是 SCORM 在教學的設計上針對教材與教學內容(或學習活動)具有相當大的彈性 (Flexibility) 與重複使用性 (Reusability)。

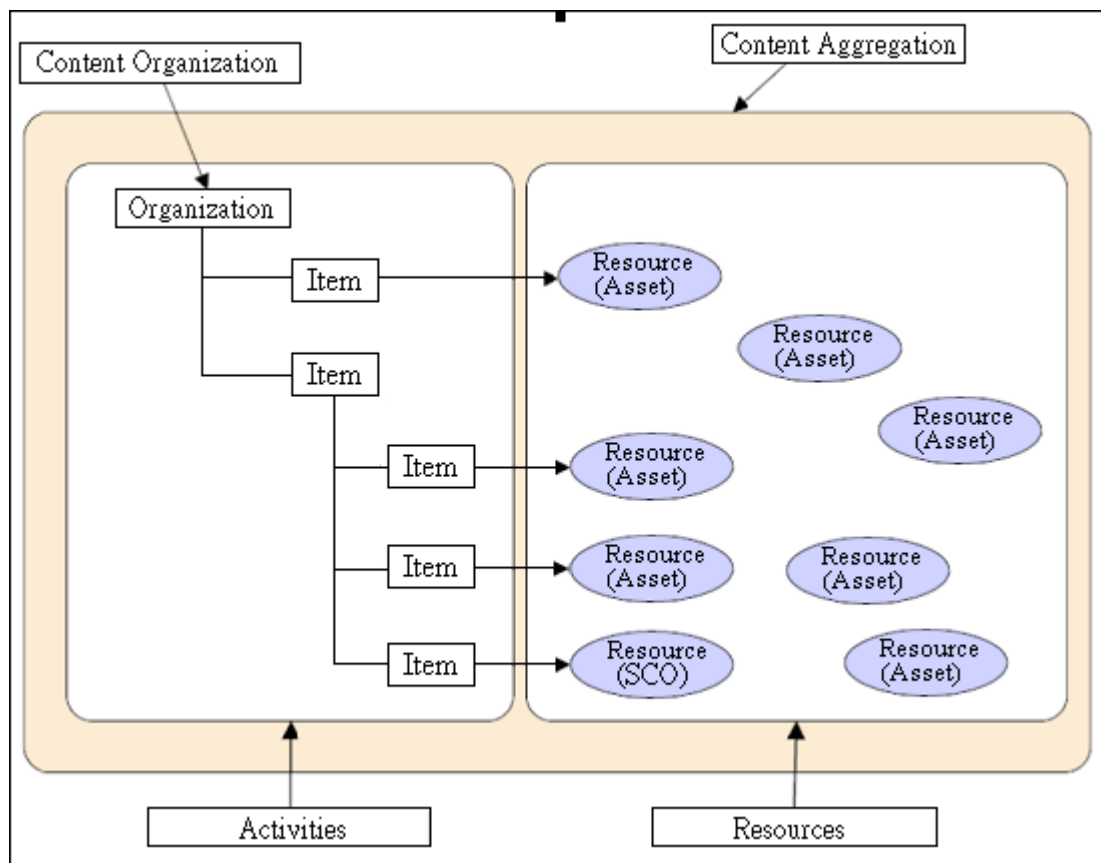


圖 3 : Content Organization (Source: ADL, 2004)

2.1.4 共享內容物件(Sharable Content Object)

Sharable Content Object (SCO) 是由一個或多個以上的 Asset 構成之並且在 SCORM RTE 的規範下設計之 LMS-to-LMS 與 LMS-to-Learner 之間所傳遞教學資源的最小單位[7][8][9]，如圖4 所示。一個 SCO 教學設計上必須具備獨立性，即 SCO 彼此之間不可在教學應用上具有高相似的本質與內容，因此在教學設計時，根據課程需要重新組裝各個 SCO 才不會發生內容相似的學習資源重複出現於課程當中，所以 SCO 是教學應用的最小學習資源，且學習資源得以有效地再利用。SCO Meta-data 的功能與 Asset Meta-data 一樣皆是用來說明該 SCO 本身具有的內容與涵義。然而，SCO 與 Asset 本質上的差異在於前者具有教學的意義，後者僅對數位化媒體本身的實質意義做出描述。

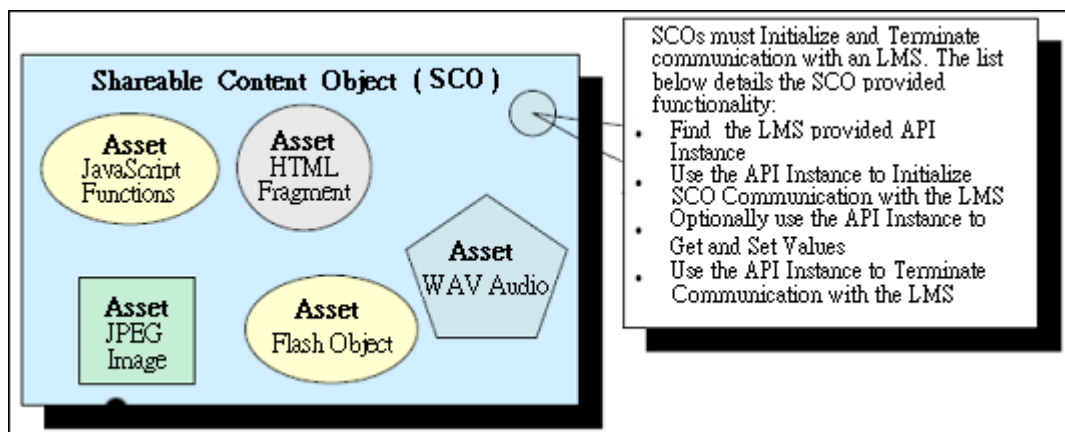


圖 4 : Sharable Content Object (Source: ADL, 2004)

2.1.5 執行環境(Run Time Environment)

執行環境 (Run-Time Environment, RTE) [10]，為了讓不同廠商製作的教材能在不同的學習管理平台 (LMS) 上使用，SCORM 的 RTE 描述了內容物件的運行機制，內容物和 LMS 之間的傳遞機制應用程式介面以及資料交換模型。為此 LMS 制定了三個標準：

(1) Launch

學習元件的啟動機制。SCORM 定義了 Assests 和 SCOs 兩種學習元件，啟動機制確立了 LMS 和學習元件之間的溝通方式，讓學習管理平台依照特定的條件啟動學習元件，學習管理平台可根據學習者的學習狀況以及教材包裹中所定義的學習順序等啟動課程。

(2) API

學習元件與學習管理平台之間用於溝通的應用程式介面，相當於學習元件與 LMS 之間的通信機制。API 包含開始、結束、獲得、儲存等狀態，是嵌入在學習資源物件裡，當學習資源與學習管理平台溝通時就是藉由 API 與之溝通，用於接受或儲存資料。

(3) Data Model

定義學習元件與學習管理平台間的資料交換模型。當在 Client 端要啟動一個 SCO 時就要透過 API Adapter 所提供的 function 以及事先定義好的資料交換共同格式 data model，才能使兩者之間正確的傳遞資料。若以程式設計的觀點

來看，API 就如同是程式設計中的函式，而 Data Model 就是函式中傳遞的參數了。如圖5 所示，說明了學習管理平台(LMS) 如何透過 RTE 和學習元件互動的流程。

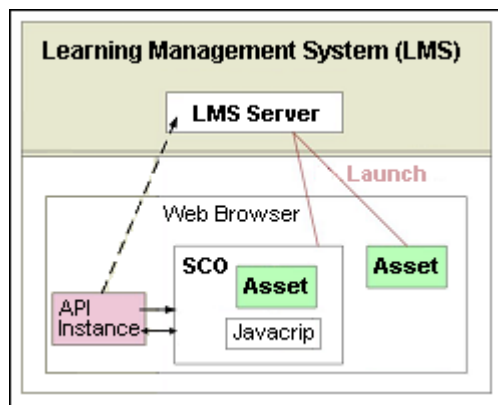


圖 5 : Run-Time Environment 概念示意圖

2.2 軟體品質



一個產品的「品質」是包含有顧客滿意度涵義的名詞，在品質控制的研究理論中，品質代表所提供的產品與服務是否具有成本效益性、即時性、以及符合顧客的需求[11][12]；而軟體品質則是指控制軟體計畫的成本、時程及結果之可預期性，使得能承擔軟體發展、使用及其風險管理的能力[13]。

軟體工程 (Software Engineering) 研究領域中關於如何提昇軟體品質有詳細的探討。在軟體的開發上，軟體品質要達到的目標有二，其一是符合產品需求規格，其二是符合客戶的需求及期望。欲達到上述目標，就必須藉助適當的軟體品質管理的機制，來達成設計優良品質軟體產品所需穩定的開發環境[14]。

常見的軟體品質特性包含正確性 (Correctness)、可靠性 (Reliability)、效能 (Performance)、安全性 (Safety)、可重用性 (Reusability)、可使用性 (Availability)、可移植性 (Transferability)、友善性 (Friendliness)、可測試性 (Testability)、可驗證性 (Verifiability)、可維護性 (Maintainability) 以及可管理性 (Manageability)

等等。國際標準組織 ISO 定義了軟體品質是包含產品或服務特徵和特性的整體，具備符合需求的能力，其品質因子包含有效率性（efficiency）、更改彈性（flexibility）、安全性（integrity）、互動性（interoperability）、可維護性（maintainability）和可攜性（portability）（ISO 8402-1986）。

由於要開發一個具備一定規模的完整軟體專案，所需具備的因素相當複雜。尤其當專案愈大時，要維持軟體品質其困難度更呈級數式增加。因此需要具有完整規劃以及制度化的軟體品質管理作業模式，以確保最終開發完成的軟體產品其品質表現符合預期的水準。因為影響整體軟體系統品質表現的，除了軟體開發工程師的技術成熟程度，還需要軟體開發團隊其他內部及外部組織中的每一成員包括軟體專案經理、系統維護人員、品質審核工程人員、銷售人員、管理人員以及顧客的配合因素等[15]。軟體流程品質是指軟體發展過程中的組成因子，它包含了使用的技術、工具、人員、組織、訓練等。藉由實施軟體流程改善措施，可以協助軟體開發團隊找出軟體品質缺陷的關鍵項目，並設計出改善軟體品質問題的軟體品質矯正措施機制，以提昇軟體產品的品質水準。目前軟體流程的品質驗證管理流程以 CMMI 最為業界所採用，有關 CMMI 的說明將在後續章節中做詳細介紹。

2.3 能力成熟度整合模式(CMMI)

能力成熟度整合模式（Capability Maturity Model Integration, CMMI）是由美國卡內基美隆大學(Carnegie Mellon University, CMU) 的軟體工程學院 (Software Engineering Institute, SEI) 所公佈提出，其主要目的是用以評估某一組織的軟體開發能力、協助持續改善軟體流程與軟體品質及提升專案與開發管理能力。它涵蓋了軟體產品從起始到交付與維護的生命週期，制定了發展與維護活動的最佳執行方法，是目前最具代表性的國際通用標準之一[16][17]。

根據行政院主計處於民國94年4月辦理「政府機構資訊系統委外開發之專案管理概況」問卷調查，其中有高達 78% 的受調查政府機構委外過程中曾發生問題，有 68% 於專案驗收時因品質不符發生爭議的情形[18]。有鑑於此，國內在經濟部

工業局及資策會和學界共同努力之下，也逐步建立了推動機制，並成功取得 CMMI 模式中文化的授權，而 CMMI 中文版也已正式在 SEI 官方網站發布。目前為止已有許多國內企業導入 CMMI 並獲致不同程度的效益，經濟部工業局網站有公佈國內企業導入 CMMI 的成效調查報告[18]。

2.3.1 CMMI 歷史沿革

CMMI 的提出可追溯至 1984 年，當時美國國防部所執行的軟體委外專案，因為無法評估軟體公司對軟體標案之承接及執行能力，於是委託卡內基美隆大學 (Carnegie Mellon University, CMU) 的軟體工程學院 (Software Engineering Institute, SEI) 進行研究，希望幫助軟體產業建立一套工程制度，使得在軟體發展上能有效的持續執行組織流程改善，藉以評估組織作業能力、改善軟體之發展過程及發展能力，並且協助軟體發展者持續進行改善軟體流程成熟架構及軟體品質，進而提昇軟體發展能力，達成縮短軟體發展時程、降低成本及確保軟體品質等目標。自 1991 年起 CMMs 陸續發展應用於許多專業領域，例如系統工程、軟體工程、軟體採購、及整合的產品與流程發展(Integrated Process & Product Development) 等模式。圖6 為能力成熟度家族歷史。

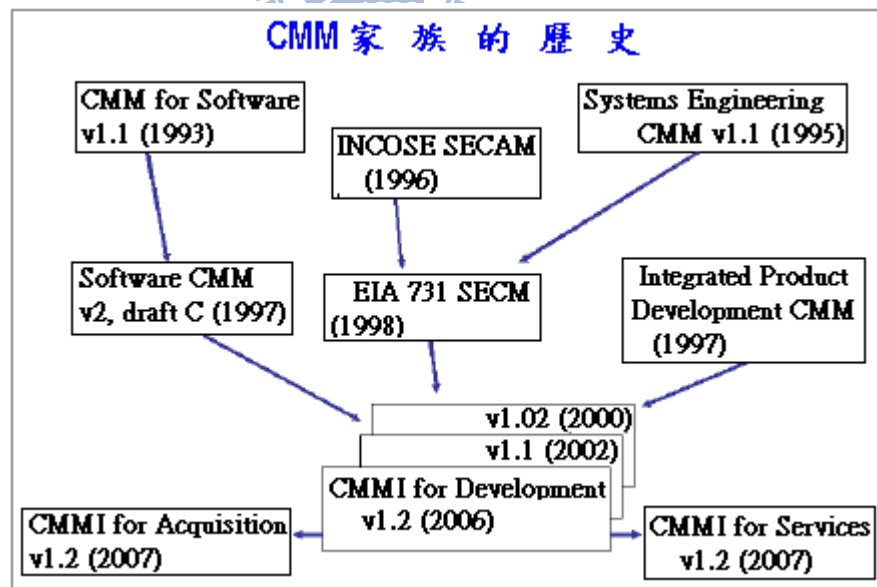


圖 6 : CMMI Derivation (Source : CMMI-DEV V1.2)

2.3.2 CMMI 表述模式

CMMI 提供了兩種不同的表述模式，以支持不同的組織流程改善方法，包括連續式表述以及階段式表述。連續式表述協助組織選擇一個流程領域（或一組流程領域）和改善與其相關的流程。這個表述使用能力度等級，描述與個別流程領域有關的改善。階段式表述使用預先定義的幾組流程領域，以定義組織流程改善的途徑，這個改善途徑使用成熟度等級描述。每一個成熟度等級提供一組流程領域，描述不同的組織行為。圖7 說明了連續式與階段式表述的結構。

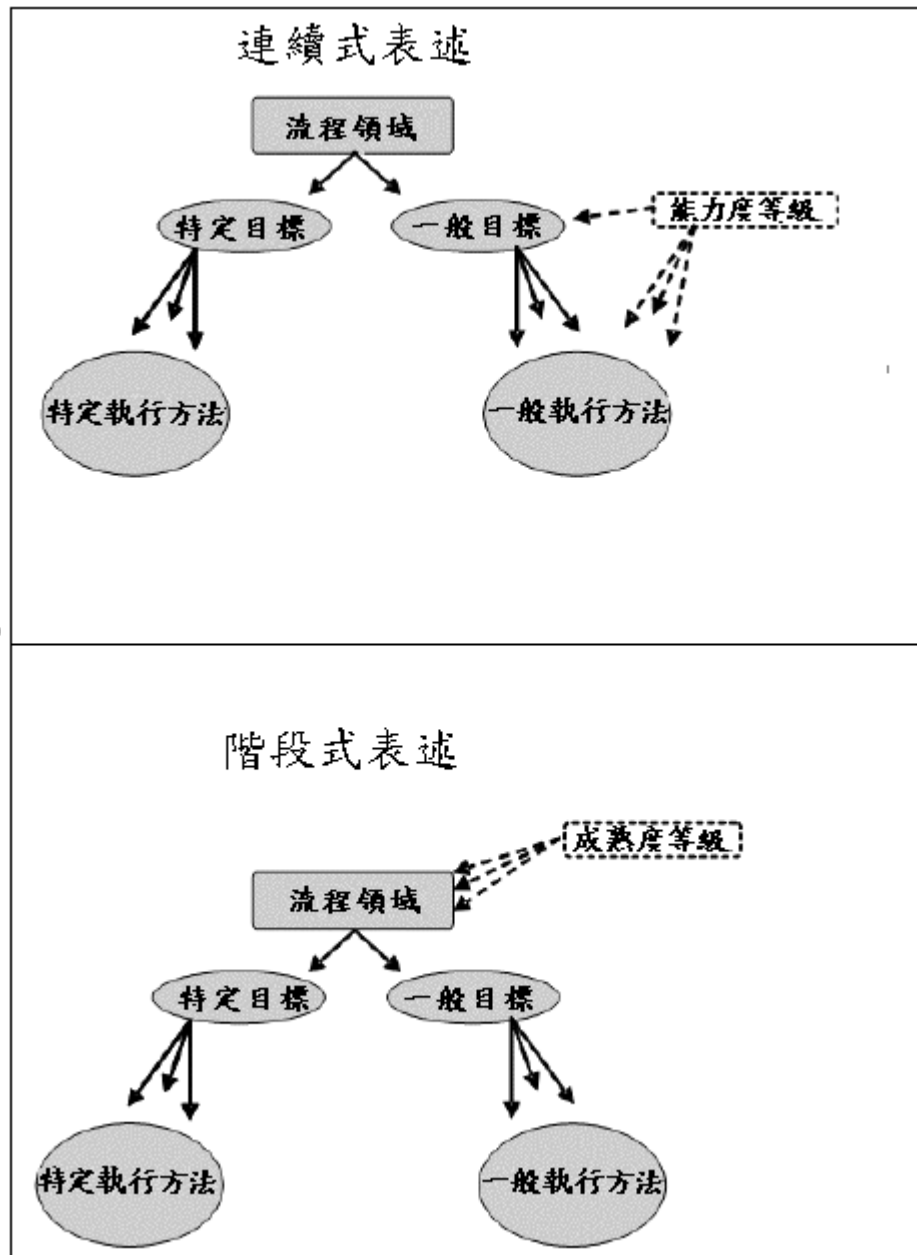


圖 7：連續式與階段式表述的結構 (Source : CMMI-DEV V1.2)

階段式表述強調軟體流程改善是循序漸進式，每個層級的完成都是下個層級的基礎。而連續式的表述則是強調企業可以依據自身的企業目標與願景來自由的選擇流程領域加以改善。表1 為階段式與連續式表述之間的差異。

表 1：連續式與階段式表述的差異 (Source : CMMI-DEV V1.2)

連續式表述	階段式表述
組織依據其流程改善目標，選擇流程領域與能力度等級。	組織依據成熟度等級，選擇流程領域。
使用能力度等級度量改善。能力度等級： <ul style="list-style-type: none"> ▪ 度量跨組織的特定流程領域的能力度。 ▪ 等級從0 到5。 	使用成熟度等級度量改善。成熟度等級： <ul style="list-style-type: none"> ▪ 度量跨組織的一組流程領域成熟度。 ▪ 等級從1 到5。
能力度等級摘要當做目標與追蹤流程改善績效。	成熟度等級當做目標與追蹤流程改善績效。
對等的階段式允許一個組織使用連續式表述進行流程改善並引申成熟度等級以當做評鑑的一部分。	並不需要一個對等的機制來反推連續式表述。

2.3.3 能力度與成熟度

階段式表述使用成熟度等級，應用於跨流程領域的組織流程改善的達成。這些等級有預測下一個從事的專案之一般產出的方式。成熟度有五個等級，編號為 1 到 5，每個等級的內容簡述如下：

等級 1：沒有固定流程、無法提供穩定環境、資源，經常超出專案時程及預算。成功經驗無法重複，偶而會成功、大都只靠少數有經驗的人才能完成。

等級 2：建立了基本的專案管理過程。按部就班地發展系統、追蹤費用、根據專案進度表來進行發展。對於相似的專案，可以重覆使用以前的經驗及成果。

等級 3：軟體發展工程和管理活動已標準化，且被集結成為一個組織的標準和流程資產。所有軟體的發展和維護都在這個標準基礎上制定與執行。

等級 4：對軟體發展過程和產品品質都有很好的歸納，產品成果和發展過程都可用數量方式控制。可界定流程變異之特殊原因，並適當矯正，軟體發展過程及產品品質的定量管理。

等級 5：經發展過程的定量回饋機制，不斷產生新思想，並研擬新技術來最佳化相關過程。組織及專案必須追求持續的、可度量的過程改進，包括缺失預防、技術更新管理和流程改造管理。

連續式表述使用能力度等級，應用於個別流程領域的組織流程改善的達成。這些等級對一個流程領域有遞增地改善流程的方式。有六個能力度等級，編號為 0 到 5，每個等級的內容簡述如下：

等級 0：沒有執行或部分執行的流程。無法滿足流程領域的一個或多個特定目標，以及因沒有制度化部分執行流程的理由，這個等級沒有一般目標。

等級 1：滿足流程領域特定目標的流程，支援並使所需工作能夠產出工作產品。由能力度第1 級所導致的重大改善可能會隨著時間而失去，因為它們沒有被制度化。

等級 2：具有基礎建設支援流程，任用具備技能的人員，並給予足夠的資源以產出可控制的產品；納入相關的關鍵人員；監督、控制及審查；並評估遵循流程說明的程度，確保現有的執行方法能在有壓力的情況下，仍維持運作。

等級 3：流程根據組織的調適指引調適組織標準流程，並納入工作產品、度量與其他流程改善資訊至組織流程資產。專案的標準、流程說明與程序由組織標準流程調適而得，以符合特定專案或組織單位，因而更具一致性。

等級 4：使用適當的統計和其它量化的技術進行控制。建立品質和流程績效的量化目標，並以該目標為管理流程的準則。以統計的術語瞭解品質和流程績效，並在流程生命週期受到管理。

等級 5：利用瞭解流程中的共同變異原因，改善流程。最佳化流程以漸進與創新的改善，專注於持續改善流程績效。

表2 為能力度與成熟度的比較[16][17]。

表 2：能力度與成熟度等級的比較 (Source : CMMI-DEV V1.2)

等級	連續式表述的能力度等級	階段式表述的成熟度等級
等級 0	不完整級	無
等級 1	執行級	初始級
等級 2	管理級	管理級
等級 3	調適級	調適級
等級 4	量化管理級	量化管理級
等級 5	最佳化級	最佳化級



2.3.4 流程領域

流程領域 (Process Area) 是一個領域下相關執行方法的集合，當它們共同執行時，滿足一系列被視為對改善該領域是重要的目標。CMMI for Development Version 1.2 規範整合了以下二十二個流程領域：

- 原因分析及解決方案(Causal Analysis and Resolution, CAR)
- 建構管理(Configuration Management, CM)
- 決策分析與解決方案(Decision Analysis and Resolution, DAR)
- 整合專案管理+IPPD(Integrated Project Management, IPM+IPPD)
- 度量與分析(Measurement and Analysis, MA)
- 組織創新與發展(Organizational Innovation and Deployment, OID)
- 組織流程定義+IPPD(Organizational Process Definition, OPD+IPPD)
- 組織流程專注(Organizational Process Focus, OPF)
- 組織流程績效(Organizational Process Performance, OPP)

- 組織訓練(Organizational Training, OT)
- 產品整合(Product Integration, PI)
- 專案監控(Project Monitoring and Control, PMC)
- 專案規劃(Project Planning, PP)
- 流程與產品品質保證(Process and Product Quality Assurance, PPQA)
- 量化專案管理(Quantitative Project Management, QPM)
- 需求發展(Requirements Development, RD)
- 需求管理(Requirements Management, REQ M)
- 風險管理(Risk Management, RSKM)
- 供應商協議管理(Supplier Agreement Management, SAM)
- 技術解決方案(Technical Solution, TS)
- 確認(Validation, VAL)
- 驗證(Verification, VER)

CMMI 模式組件被群組成三個類型：必要的、期望的與助益的[17]。

- (1) 必要的組件：說明一個組織要滿足某一個流程領域所需要達成的成果，此成果必須很明顯地被一個組織的流程所執行。
- (2) 期望的組件：包含特定執行方法和一般執行方法。在目標被認定已滿足之前，執行方法或其可行的替代方案，都必須表現於組織已規劃和已實行的流程之內。
- (3) 助益的組件：提供細部描述協助組織著手思考，如何達成必要的組件與期望的組件。

流程領域與相關的模式組件其關係，如圖8 所示。

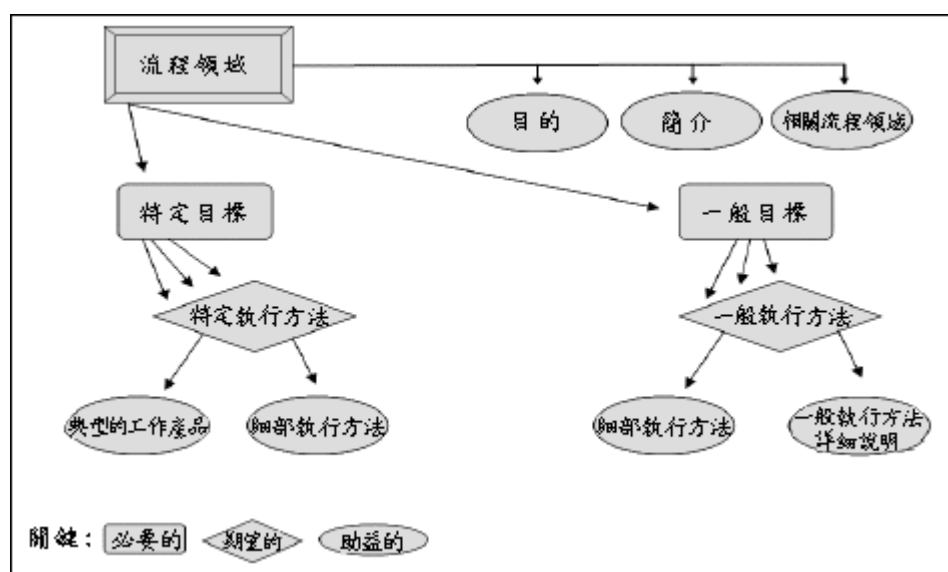


圖 8：流程領域與相關的模式組件 (Source : CMMI-DEV V1.2)

2.3.5 驗證流程領域

本研究以設計並實作出符合 CMMI 驗證流程領域 (Verification Process Area) 成熟度 Level 3 的多媒體教材品質驗證系統為目標，因此有必要對驗證流程領域作詳細說明。驗證 (Verification) 的目的，在於確保選定的工作產品符合其指定的需求規格。根據 CMMI 規範說明，驗證包括產品及中間工作產品的驗證，將其與選定的客戶需求、產品需求及產品組件需求加以比較。

整個流程領域中，產品及產品組件的意涵也包括服務及其組件。驗證是一種漸進式過程因為它發生於產品及工作產品的發展過程中，從需求驗證開始，經工作產品到最終完成產品的驗證。在 CMMI 中所規範之驗證流程領域的連續式表述之特定目標及特定執行方法如表3 所示。

表 3：驗證流程領域連續式表述之特定目標及執行方法

連續式表述	特定目標 / 執行方法	
SG1 驗證準備	SP 1.1	選擇需驗證之工作產品
	SP 1.2	建立驗證環境
	SP 1.3	建立驗證程序及準則
SG2 執行同仁審查	SP 2.1	準備同仁審查
	SP 2.2	進行同仁審查
	SP 2.3	分析同仁審查資料
SG3 驗證工作產品	SP 3.1	執行驗證
	SP 3.2	分析驗證結果

(Source : CMMI-DEV V1.2)

依據軟體工程理論，系統驗證與確認的規劃，應該是在開發程序的初期即開始進行。對應到多媒體教材品質管理系統上是從需求定義階段即開始進行驗證程

序。典型的驗證模組包括需求規格驗證、高階設計驗證、正規化規格驗證、細部設計驗證以及產品驗證。驗證與確認的規劃程序應該在這些工作之間取得平衡。

系統必須進行的工作包括進行驗證與確認的靜態與動態方法、制定教材檢查及測試的標準與程序、建立檢查清單做為程式檢查之用，以及定義教材測試計畫等。測試計畫除了用來協助安排資源和估計測試時程外，也提供給負責教材設計與進行系統測試的測試人員使用。圖 9 是一個典型的靜態和動態的驗證與確認週期。

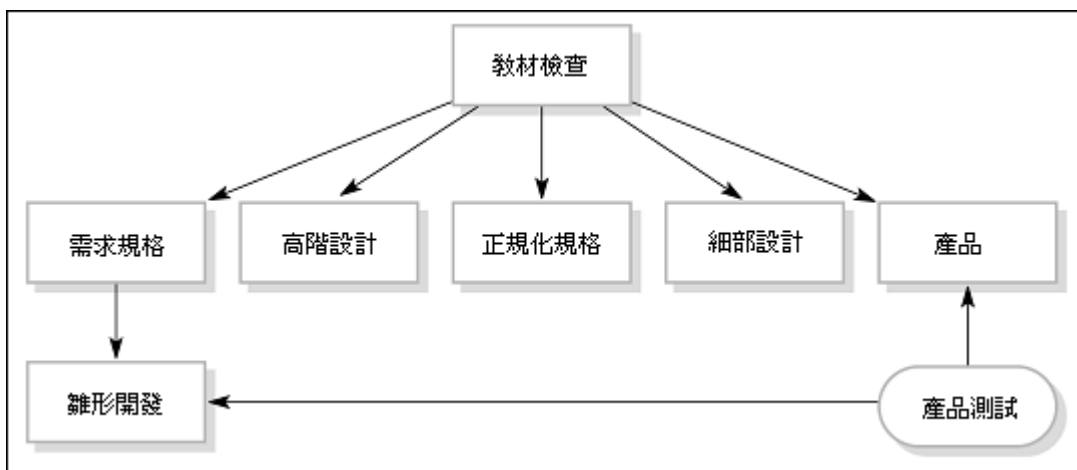


圖 9：靜態和動態的驗證與確認 (Source : Software Engineer)

2.3.6 預期效益

在國內，經濟部工業局曾在提升資訊軟體品質計畫網站中提出導入 CMMI 的效益統計數據[19]。在國外，根據 SEI 於 2003、2004 年的分析報告[20][21]指出，導入 CMMI 廠商的平均效益如下：

1. 生產力提高 35%。
2. 產品上市時間縮短 19%。
3. 產品發行後缺失率減少 39%。
4. 節省成本對投入軟體流程改善成本的比例為 5：1。
5. 產品錯誤率約降低了 15%。
6. 生產力提高了 30%。

由此可知流程改善是組織要持續進行品質改善的重要關鍵，CMMI 模式提供了改善組織、流程規範指引，以及發展、取得及維護產品或服務的管理能力。

CMMI 模式已建立作業驗證架構，來幫助組織依據本身的成熟度或能力度、建立改善的優先順序，實施改善方案。此外，組織更可藉由導入 CMMI 模式，改善流程及提供指引，確保流程的穩定度、能力度及成熟度，並協助提昇產品品質，有效降低開發成本與後續維護成本，強化組織於市場上的競爭力。



三、系統分析與流程設計

多媒體教材兼具「教學」與「軟體」兩種專業領域特性，要驗證多媒體教材的一致性就必須針對這兩大專業領域進行規範遵循及方法管理。在教學領域方面，多媒體教材必須依據 SCORM 的相關規範，以確保開發出來的教材產品能正確有效的引導學習者進行學習，達成教學目標。在軟體領域方面，為了有效的進行檢查驗證，避免產生多媒體教材內容一致性的錯誤，必須從多媒體需求管理流程開發針對需求規就進行系統化的審查。而在驗證流程管理方面，國際上，以美國卡內基美隆大學 (Carnegie-Mellon University, CMU) 的軟體工程研究所 (Software Engineering Institute, SEI) 所提出的國際性標準能力成熟度整合模式 (Capability Maturity Model Integrated, CMMI) 整合了大部分國際標準的內容最具有代表性。CMMI 優越的流程改善、管理的能力及品質規範，近年來也獲得世界上大部分軟體開發組織的認同與採用。

本研究的主要目標，在於提出一套於多媒體教材驗證管理方法，從多媒體教材的製作過程中，實作以 CMMI 驗證流程領域的各項管理方式，來達到製作出符合能力度第三級的多媒體教材驗證管理系統。因此本章將針對 CMMI 驗證流程領域進行分析，並整合多媒體教材開發的生命週期，設計、製作出符合上述目標的多媒體教材驗證管理系統 (Multimedia Verification Management System, MVMS)。

3.1 CMMI 驗證流程領域分析

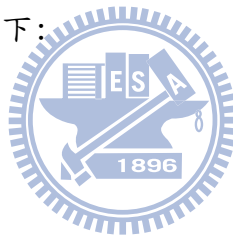
CMMI 驗證流程領域 (CMMI Verification process area) 的主要目的，在於提出一套適於產品在開發製作過程中實施的驗證方法，以確保選定的工作產品符合其指定的需求。根據 CNS 14836 (ISO/IEC12207:1995) 的定義：由檢查與提出客觀證據，以證實規定之要求已被達成；並且在該標準文件第 6.4 節提到：「驗證的過程，乃是在判斷某活動的軟體產品，是否達成前項活動加諸其上之需求或條件的過程」。由此可知，驗證工作最主要的目標就是保證工作產品與需求規格的一致性。

因此，從進行需求管理工作階段開始，直到教材產品開發完成的整個流程。當中包括審查需求規格、制定規格驗收測試計劃、審查系統規格、制定系統整合測試計劃、檢查系統設計、制定子系統整合測試計劃、檢查細部設計、制定模組單元製作測試計劃等，然後配合子系統整合測試、系統整合測試、直到最後的驗

收測試。每一個階段都要確實執行適當的驗證步驟，才能確保整個流程都符合驗證程序，如此最終的產品也才能達到符合需求的目的。

良好的驗證管理是有效執行專案驗收的基礎，無法適當地執行規格審查、系統檢查、及各階段驗證步驟，無法掌握各項規格需求變更，都將會導致專案品質不良、延遲交貨及客戶不滿意等。因此，驗證管理的主要目的就是要避免上述情況的發生，並有效地管理驗收程序，以確保所有專案之最終產品及產品組件均符合使用者需求。此外還必須能有效控驗證過程中發生錯誤時的回報機制，以協助教材製作者進行矯正措施，這也就是驗證管理對專案的重要性。驗證工作在專案生命週期各階段會持續演進為需求規格、審核計畫、系統規格、測試文件、測試結果分析等之內容。這些項目是在不同發展階段的工作產品，但前後之間關係密切，任何一項工作產品內容變更都有可能對其他項目造成影響。亦即，驗證會影響專案的整個流程。因此，對驗證工作能愈早進行就愈能減輕開發過程中，因某一階段的缺失而影響到下一階段。

本研究參考 CMMI 中有關於驗證流程領域 (Verification Process Area) 中的相關規範，共有三個特定目標 (Specific Goal, SG) 及八個特定執行方法 (Specific Practices, SP)，其內容如下：



SG 1 驗證準備

執行驗證準備。

SP 1.1 選擇需驗證之工作產品

選擇需驗證之工作產品及每一工作產品使用之驗證方法。

SP 1.2 建立驗證環境

建立並維護支援驗證工作的環境。

SP 1.3 建立驗證程序及準則

建立並維護所選定的工作產品的驗證程序與準則。

SG 2 執行同仁審查

對選定的工作產品執行同仁審查。

SP 2.1 準備同仁審查

準備對選定的工作產品進行同仁審查。

SP 2.2 進行同仁審查

準備對選定的工作產品進行同仁審查。

SP 2.3 分析同仁審查資料

分析同仁審查的準備、執行及結果資料。

SG 3 驗證工作產品

依照其指定的需求, 驗證所選定的工作產品。

SP 3.1 執行驗證

對選定的工作產品執行驗證。

SP 3.2 分析驗證結果

分析所有驗證活動的結果。

本研究針對多媒體教材開發製作過程, 以 CMMI 中驗證流程領域的八項規範為基礎, 歸納並分析出系統開發的四大模組, 分別為:

1. 驗證準備 :

驗證包括產品及中間工作產品的驗證, 將其與選定的客戶需求、產品需求及產品組件需求加以比較。整個流程領域中, 產品及產品組件的意涵也包括服務及其組件。驗證是一種漸進式過程因為它發生於產品及工作產品的發展過程中, 從需求驗證開始, 經工作產品到最終完成產品的驗證。

2. 同仁審查 :

同仁審查是驗證的重要部分, 也是經證實可以有效去除缺失的機制。發展一套瞭解工作產品及產出流程的方法, 以利防制缺失並界定流程改善的有利機會。同仁審查為有系統的檢查工作產品, 以界定缺失及其他變更需求。此項工作是由產品製作人員的同仁來執行。

3. 驗證執行 :

驗證是一種漸進式過程因為它發生於產品及工作產品的發展過程中, 從需求驗證開始, 經工作產品到最終完成產品的驗證。於產品及工作產品發展過程中, 逐步執行驗證, 促使及早發現問題, 並能及早移除缺失。驗證結果節省了耗用在尋找問題過程中, 將問題獨立出來及重做的可觀成本。

4. 矯正措施界定 :

當專案的執行結果出現偏離計畫或需要部分修正時, 必須採取管理矯正措施

直到結案。矯正措施包括分析議題、採取矯正措施、管理矯正措施。

為了清楚的表示這四大模組在多媒體教材開發過程中與相關人員的關連性，本研究以一個驗證流程模組示意圖來表示其相關性，如圖10 所示。

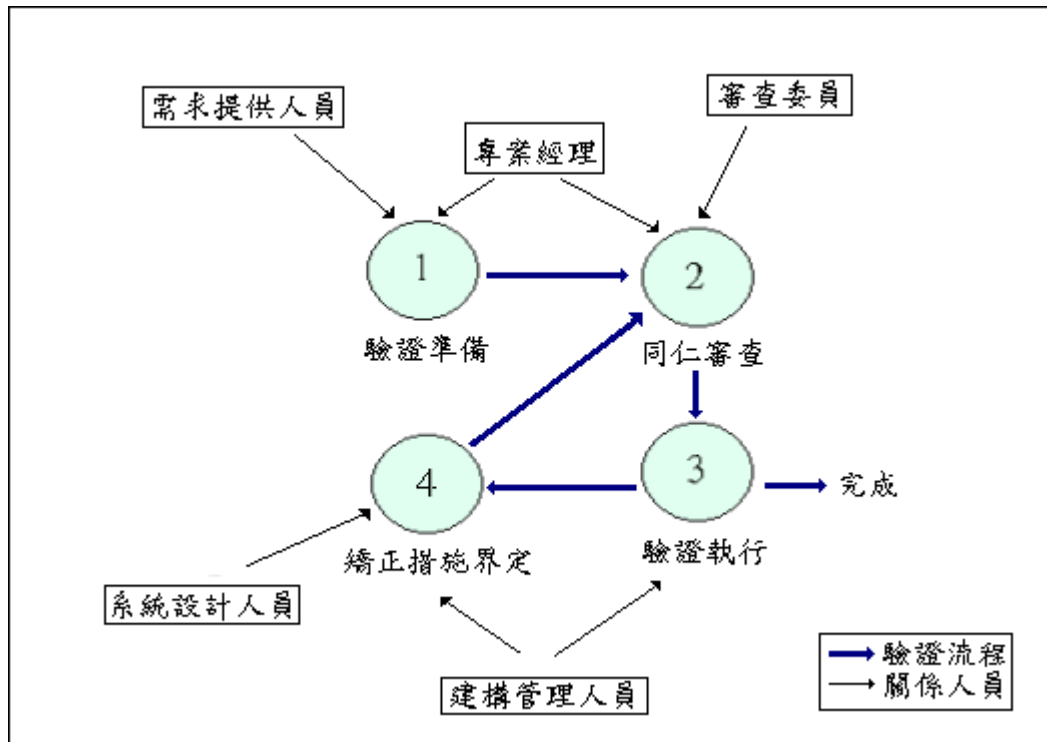


圖 10: 驗證流程四大模組示意圖

3.2 功能性需求與非功能性需求

需求區分為功能性需求、非功能性需求與專業領域需求，本節主要在探討與分析多媒體教材驗證管理系統的功能性需求及非功能性需求這兩項。功能性需求即是系統該提供什麼樣的功能，非功能性需求是指系統的特性與限制。依據 CMMI 驗證流程領域的特定目標 (SG) 與特定執行方法 (SP)，本研究分析出所要開發的 MVMS 系統之四大模組所需之功能性需求與非功能性需求，以表列的方式表示如下。

1. 功能性需求

MVMS 系統之功能性需求，是以 CMMI 驗證流程領域的特定訂目標(Specific Goal) 及特定方法 (Specific Practices) 所需執行之功能項目為基礎所發展的，主要由驗證準備、同仁審查、驗證執行、矯正措施界定等四大模組所組成。其實際執行方法是結合軟體工程理論來實作，表4 即為其詳細項目。

表 4：MVMS 系統功能性需求表

功能性需求	CMMI VER SG/SP	描述
界定驗證工作產品	選擇需驗證之工作產品	選擇需驗證之工作產品及每一工作產品使用之驗證方法
建立驗證環境	建立驗證環境	建立並維護支援驗證工作的環境
建立驗證程序及驗證準則	建立驗證程序及準則	建立並維護所選定的工作產品的驗證程序與準則
準備同仁審查作業	準備同仁審查	準備對選定的工作產品進行同仁審查
同仁審查	進行同仁審查	針對所選定的工作產品進行同仁審查，並由同仁審查的結果界定議題
審查資料分析	分析同仁審查資料	分析同仁審查的準備、執行及結果資料
驗證工作產品	執行驗證	對選定的工作產品執行驗證
驗證分析	分析驗證結果	分析所有驗證活動的結果
建立矯正措施計畫	建立矯正判斷準則及矯正措施計畫	建立矯正措施準則、矯正計畫活動內容
執行矯正措施	針對議題，採取矯正措施	監控矯正措施直到計畫完成

2. 非功能性需求

MVMS 系統之非功能性需求如下：

- (1) 終端 - 伺服器 (Client - Server) 架構
- (2) 多人同時上線使用 (Multi User)
- (3) 後端資料庫伺服器 (Database Server) 連結
- (4) 網際網路瀏覽系統操作介面 (Web-Based Interface)
- (5) 權限管控措施及相關系統開發工具
- (6) 作業系統 (Operation System) 平台限制
- (7) 使用的系統發展資源等項目

表5 為 MVMS 非功能性需求列表。

表 5：MVMS 系統非功能性需求表

非功能性需求	描述	類別
WEB-BASE 多人使用	網頁介面 可多人同時連線作業	建置環境
系統權限	教材製作者和審核者 各自不同權限	安全性
可移植性	可跨平台作業系統	可攜性
發展資源	JAVA JDK Tomcat WEB Server MySQL Database	系統資源



3.3 多媒體教材製作生命週期

多媒體教材是創意呈現的結果表現，如何利用多媒體來呈現教材所要表達的內涵意境或闡釋的精神理念是製作多媒體教材時最重要的課題。多媒體教材的製作首先必須選定腳本 (script)，腳本涵蓋了多媒體教材內容所要表達的理念。而腳本中依照情節需要使用了各種不同型態的多媒體資料來協助豐富教材內容，使得更貼切的表達出教材創作者的理念或更貼近於讀者。透過腳本內容的情節串聯起這些多媒體資料之間的空間及時間關係，多媒體教材就如同由包括文字、影像、聲音、動畫擔任的演員，依照設計好的劇情合力演出一齣戲劇。而教材讀者則扮演觀眾角色，經由演出與觀賞的互動中，達到學習或傳達知識的目的。

參考交通大學軟體工程實驗室所提出的研究內容 [22]，可歸納出多媒體教材的開發生命週期的基本步驟如下：

1. 多媒體教材的教案設計。
2. 根據教案的內容設計分鏡腳本 (Storyboard)。
3. 根據分鏡腳本 (Storyboard) 的劇情需要，設計多媒體教材的場景及使用者介面 (User Interface)。
4. 根據分鏡腳本 (Storyboard) 的劇情、多媒體教材的場景與使用者介面 (User

Interface) 的需要，設計教材所需的多媒體素材演員(如文字、圖片、動畫、影片、音樂)。

5. 使用適當的多媒體編輯工具，根據分鏡腳本 (Storyboard) 的劇情內容，將多媒體教材的場景及使用者介面 (User Interface) 與素材，組合成一完整的多媒體教材。

在軟體工程 (Software Engineer) 領域裡，將軟體製作程序 (Software Process) 有系統地進行規劃、分析、設計、程式製作、測試與維護等工作步驟，已經廣泛為軟體開發者所遵循與應用，藉此提高軟體的生產力和軟體的品質。軟體製作程序的主要的階段 [14] 如下：

1. 軟體需求確認階段 (Requirements Definition Phase)
2. 軟體設計階段 (Design Phase)
3. 軟體實作階段 (Implementation Phase)
4. 軟體測試及驗證階段 (Verification and Validation Phase)
5. 軟體維護階段 (Maintenance/Evolution Phase)

如同開發軟體一樣，多媒體教材本身也是「軟體」的一種，其開發流程亦適用於一般軟體工程中的軟體製作生命週期中的各個階段。不同的是有別於一般軟體程式，因多媒體教材軟體額外具有「教學」的性質及使用多媒體形式呈現的特性，其製作過程的生命週期與一般軟體程式稍有不同，其主要的發展階段詳述如下：

1. 課程導入期：

主要的工作是根據教學目的、教學對象與教學環境的需求，使用原創設計或是選定現有適當的學習教材或教案，作為開發多媒體教材的原始教學內容與依據

2. 課程規劃期/單元腳本設計：

主要的工作是設計多媒體教材的分鏡腳本，此分鏡腳本是用來說明多媒體教材內容所要表達意境或觀念。腳本內的劇情同時需要描述這些多媒體素材之間與場景的空間及時間的關係。

3. 課程規劃期/場景設計：

主要的工作是設計多媒體教材的使用者介面，美術編輯者依據多媒體教材的分鏡腳本中每一幕場景的需要，設計多媒體教材內容的使用者介面。

4. 課程製作期/素材製作：

主要的工作是根據多媒體教材的分鏡腳本劇情的需要，設計腳本中每段劇情所需要使用不同資料格式的多媒體素材，來協助了解教材內容的意境表達或觀念。常見的多媒體資料格式種類包括聲音、影片、圖片、動畫及文字。

5. 課程製作期/教材製作：

主要的工作是多媒體教材內容的編輯及製作，需要使用多媒體編輯軟體或程式開發工具，根據分鏡腳本的劇情描述，將多媒體素材及場景之間的空間及時間的關係，製作成一幕幕的多媒體教材內容。而此步驟所使用的多媒體編輯軟體或程式開發工具，對於多媒體教材的最終成品有關鍵性的影響。

在多媒體教材開發生命週期 (Multimedia Learning Contents Process Life Cycle) 中的每一個階段，都必須經過同仁審查的方式進行審查步驟，並作為下一個階段開發的依據，以此方式來確保製作出來的多媒體教材產品品質，如圖 11 所示。

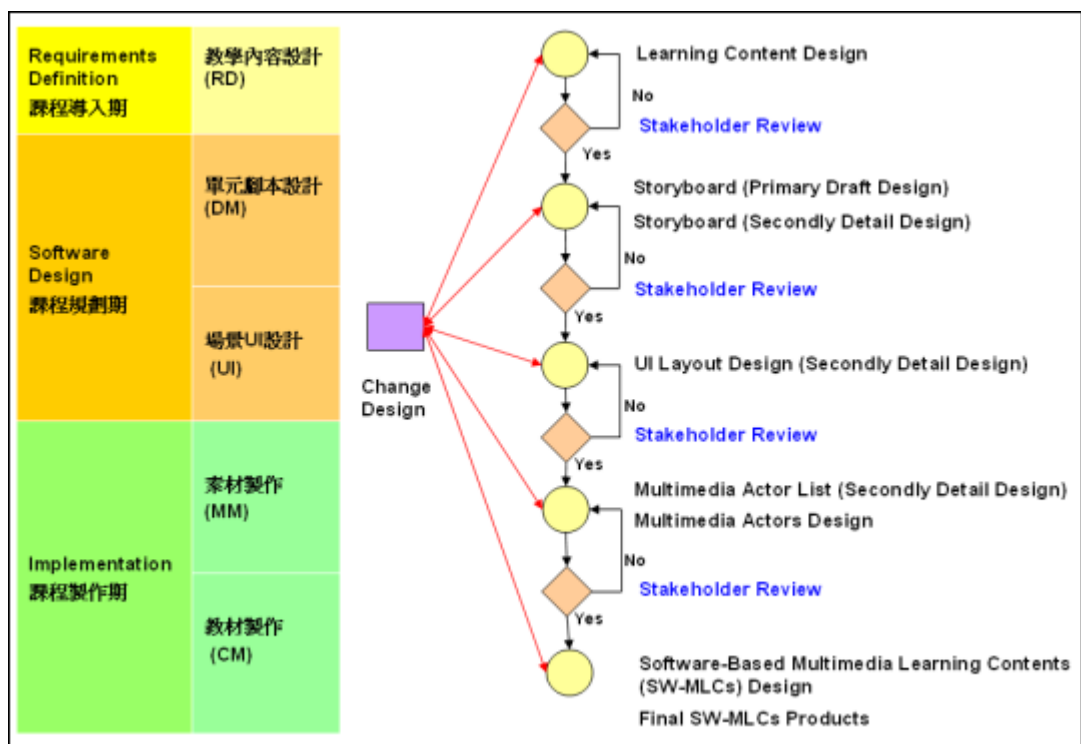


圖 11：多媒體教材開發生命週期流程圖 (Source: 交通大學軟體工程實驗室)

3.4 多媒體教材單元的相依性關係

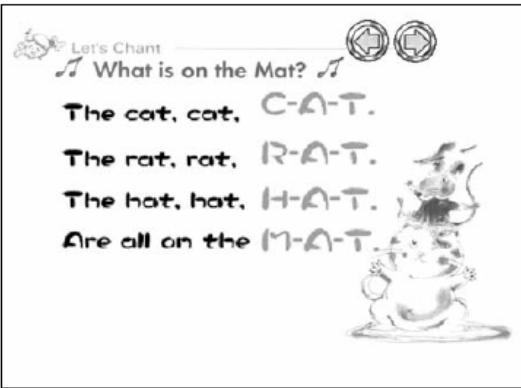
因為在多媒體教材開發生命週期 (Multimedia Learning Contents Process Life Cycle) 的各個階段，都可能發生需求變更 (Requirements Change) 情況，而導致不一致性 (Inconsistency) 品質問題的產生。解決此不一致性問題的方法是在需求變更之前，預先評估在教材開發過程中，由於修改劇情、場景或是多媒體素材對於最後成品所造成影響的範圍大小，再根據影響的範圍做變更後驗證確認。舉例而言，假設所設計的多媒體教材的分鏡腳本內容如表6 所示，此腳本內容中包含了一個場景 (Scene, SNE) 與三段劇情 (Natural Language Script, NLS)，每一段劇情都有使用到各自的多媒體演員素材 (Multimedia Actor, ATR)，三段劇情總共使用了四個演員素材，而共同使用了同一個場景設計 (User Interface, UI)。

依據表6 的分鏡腳本 (Storyboard) 內容，可以歸納出以下的相依性關係

1. 當劇情 NLS1 的內容被修改時，場景 UI1 及多媒體素材演員 ATR1、ATR2 的設計就會受影響。
2. 當劇情 NLS2 的內容被修改時，場景 UI1 及多媒體素材演員 ATR2、ATR3 的設計就會受影響。
3. 當劇情 NLS3 的內容被修改時，場景 UI1 及多媒體素材演員 ATR1、ATR2、ATR4 的設計就會受影響。
4. 當場景 UI1 的設計被修改時，多媒體素材演員 ATR1、ATR2、ATR3、ATR4 及最後多媒體教材成品 CM1 的設計就會受影響。
5. 當多媒體素材演員 ATR1、ATR2、ATR3、ATR4 的設計被修改時，最後多媒體教材成品 CM1 的設計就會受影響。

根據上述的分析可以推導出劇情、場景或是多媒體素材演員之間所存在的相依性關係 (Dependency Relationship)，並估算出修改設計後所造成的影響範圍。例如，當發生劇情 NLS3 申請變更設計時，則在審核通過劇情 NLS3 申請變更設計的同時，必須將場景 UI1 及多媒體素材演員 ATR1、ATR2、ATR4 同時變更為待審核狀態。直到劇情 NLS3 變更完成，劇情 NLS3，場景 UI1 及多媒體素材演員 ATR1、ATR2、ATR4 全部重新通過審核後，才進入下一階段。

表 6：分鏡腳本範例

Scene: SNE1	
User Interface: UI1	
	
Natural Language Script	Multimedia Actor
NLS1: Let's Chant What is on the Mat? 前後方向按鈕的圖片，隨開場場景出現，播放鈴聲。	ATR1 [Pic1] ATR2 [Aduio1]
NLS2: 播放鈴聲，女生說「Page sixteen. Let's Chant. What is on the Mat?」	ATR2 [Aduio1] ATR3 [Aduio2]
NLS3: 畫面出現對話 男生說「The cat, cat, C-A-T.」女生說「The rat, rat, R-A-T.」 男生說「The hat, hat, H-A-T.」 男生和女生同時說「Are all on the M-A-T.」	ATR1 [Pic1] ATR2 [Aduio1] ATR4 [Aduio2]
Final Product: CM1	

3.4.1 多媒體教材製作與非循環有向圖

由多媒體教材的製作過程中，可以發現每一個製作階段會在時間和空間上呈現特定的順序性結構關係，此特定順序性即為多媒體教材的相依性關係。根據圖學理論[23]，我們可以發現「非循環有向圖 (Directed Acyclic Graph, DAG)」很適合用來表示上述關係的資料結構。其中 DAG 中的節點(Vector) 即是多媒體教材製作的每一階段，而 DAG 中的邊 (Edge) 則代表了多媒體教材的相依性關係。以表6 的分鏡表內容為例，將多媒體教材相依性關係 (Dependency Relationship) 建立一個以 DAG 來表示的相依性模型，。在 DAG 中的每一節點 (Node)，代表多媒體教材在開發生命週期 (Process Life Cycle) 的每一階段中所需完成的工作單元。在 DAG 中的每一個邊 (Edge)，代表多媒體教材開發生命週期的每一階段與

相鄰階段之間的工作存在相依性關係 (Dependency Relationship) 與完成的順序性 (Sequence) 關係。以下是多媒體教材在開發生命週期的 DAG 圖型中所使用的節點類型的描述：

- 分鏡節點 (Scene node)：
此節點代表多媒體教材內一個獨立的分鏡畫面作業單位。此作業單位包含在分鏡腳本 (Storyboard) 中。在此節點內必須完成紀錄相關劇情、多媒體素材與使用場景的關係性。
- 劇情節點 (Natural language script node)：
此節點代表多媒體教材內的一個獨立分鏡中，多媒體素材所需要演出的劇情。如同分鏡節點一樣，此劇情亦包含在分鏡腳本 (Storyboard) 中。在此節點必須以自然語言說明多媒體素材依照劇情的時間與空間的關係性進行表演行為。
- 場景節點 (User Interface node)：
此節點代表多媒體教材內的一個獨立分鏡中，多媒體素材演出相關劇情時，所必須使用的舞台及背景。此背景同樣在分鏡腳本 (Storyboard) 中。在此節點內必須完成場景及使用者介面的設計工作。
- 演員素材節點 (Actor node)：
此節點代表多媒體教材內的一個獨立分鏡中，依照劇情的情節需要參與演出的多媒體素材演員。這些素材演員也都記錄在分鏡腳本 (Storyboard) 中。在此節點必須完成多媒體演員素材的設計工作。
- 教材節點 (Code module node)：
此節點代表多媒體教材內的一個獨立分鏡中，多媒體素材根據此分鏡的相關劇情，在特定場景內進行演出的呈現結果。此結果也必須記錄在分鏡腳本 (Storyboard) 中。

結合 DAG 圖學理論，在此我們用 $G(\text{Graph})$ 表示多媒體教材製作圖， $V(\text{Vector})$ 表示多媒體教材製作的各單元的集合， $E(\text{Edge})$ 表示多媒體教材的相依性關係的集合。於是我們得到以下的 (1)、(2)、(3) 關係式

$$G = (V, E) \quad (1)$$

$$V = \{v_1, v_2, v_3, \dots, v_i\} \quad (2)$$

$$E = \{e_1, e_2, e_3, \dots, e_j\} \quad (3)$$

其中，

v_1 表示 SNE1 節點，

v_2 表示 NLS1 節點，

v_3 表示 NLS2 節點，

...

v_i 表示 CM1 節點，以此類推。

e_1 表示 SNE1 到 NLS1 的邊，

e_2 表示 SNE1 到 NLS2 的邊，

e_3 表示 SNE1 到 NLS3 的邊，

...

e_j 表示 ATR4 到 CM1 的邊，以此類推。

由節點 (nodes) 與每一個邊 (edges) 所組成的 DAG 模型，清楚的描述了多媒體教材開發生命週期的每一階段變更時的影響範圍大小。此外，非循環有向圖 (DAG) 具有方向性以及非循環的特性[24]，可以表示多媒體教材開發中每一工作階段的相依性關係 (Dependency Relationship)。因此適合作為雙向追蹤 (Bidirectional Traceability) 以及不一致性偵測 (Inconsistent Detection) 的多媒體教材基礎模型。圖12 即為根據表6 所繪製的 DAG 模型。

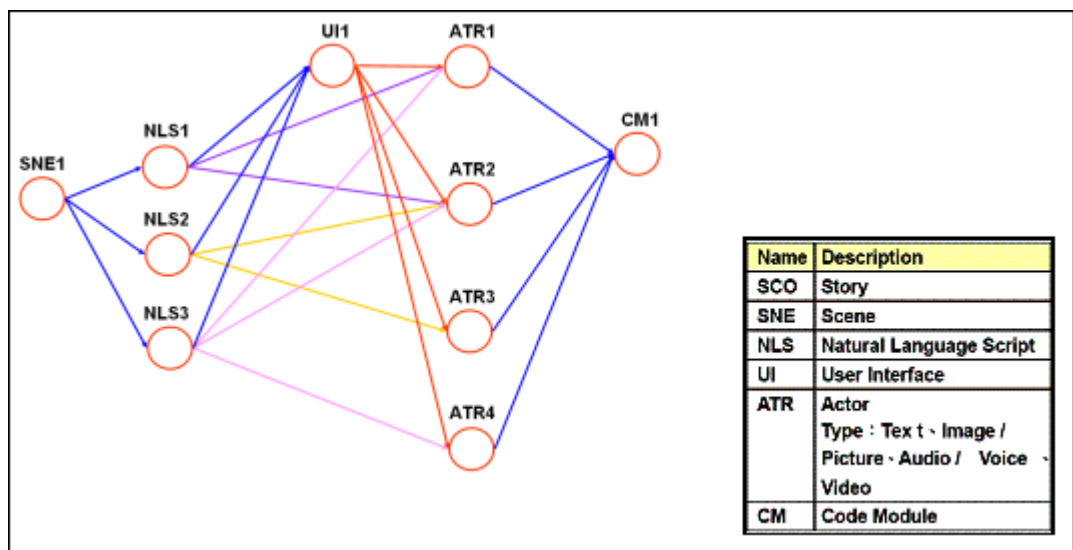


圖 12：多媒體教材開發週期非循環有向圖

3.4.2 多媒體教材單元資料結構與演算法

針對圖12 的 DAG 模型，為兼顧分析上的容易度及演算效率，我們採用將 DAG 轉為多元樹狀結構(Multiple Trees Structure) 方式做分析，並建立以鏈結串列(Link-List) 為基礎的資料結構來追蹤各單元節點的相依性關係(Dependency Relationship)。為了解說方便，在本節中我們以 ” 節點 ” 來表示多媒體教材單元。首先我們建立 DAG 中的節點的資料結構，以名稱 ITEMdata 表示，ITEMdata 是自定義的類別變數。此類別變數中包含該節點的編號、標題、驗證旗標以及連結到其他節點的串列類別型態，其中節點編號(ID) 在 DAG 中必需是唯一的，以作為辨識依據，而變數 LEVEL 是整數型態用於紀錄此節點是教材中的章節、SCO、劇情、分境表、場景或是演員。變數 NEXT 是連接到同一階層(即 LEVEL 相同)的下一節點，OUTdata 則是連接到下一階層(LEVEL+1)的起始節點。在實際應用中 ITEMdata 可以加入任何所需要的資料型態組合，圖13 是基本的 ITEMdata 建構子(Constructor) 函式的虛擬碼。

```
1 class ITEMdata
2 {
3     String ITEM;
4     String ID;
5     int LEVEL;
6     boolean VERIFICATIONflag; // verification flag
7     ITEMdata NEXT; // next node
8     ITEMdata OUTdata; // output level node
9
10    ITEMdata(String ITEMIn,String IDIn,int LEVELIn)
11    {
12        ITEM ← ITEMIn;
13        ID ← IDIn;
14        LEVEL ← LEVELIn;
15        VERIFICATIONflag ← false;
16        NEXT ← NULL;
17        OUTdata ← NULL;
18    }
19 }
```

圖 13：多媒體教材節點建構子虛擬碼

接著，定義一個名稱為 GRAPH 的 ITEMdata 陣列型態的變數代表整個 DAG 模型的進入點。GRAPH 陣列的索引值 0 是連接到教材節點串列起始處，GRAPH 陣列的索引值 1 是連接到章節節點串列起始處，GRAPH 陣列的索引值 2 是連接到 SCO 節點串列起始處 … 以此類推。接著開始將 DAG 轉換成多元樹狀結構。轉換原則有兩點，第一點是將每一個節點加入到 GRAPH 中的適當階層，第二點是該節點亦同時加入上一層父節點的 OUTdata 串列中，並且不可重複加入。由於在一開始時整個多元樹狀結構是不存在的，因此 GRAPH 的每一個索引值的內容都被設為 NULL。

在加入新節點時必須指定節點階層以及其父節點編號，系統先將此新節點加入到指定的階層陣列中，此處指定的階層是由變數 LEVEL 決定。接著還必需在其父節點的 OUTdata 串列中加入一個複製此節點內容的新節點。方法是先根據階層變數 LEVEL - 1 和 變數 PARENTid 在 GRAPH[LEVEL - 1] 的串列中搜尋到其父節點。找到父節點後，再搜尋父節點的 OUTdata 串列中是否已有此節點？若沒有，則另外複製一個相同於此節點內容的一個新節點，並將其加到父節點的 OUTdata 串列的尾端。搜尋的演算法請參考圖 14。若父節點的 OUTdata 串列中已有此節點，則不再加入。透過重複上述演算步驟將每一個節點單元依序加入 GRAPH 中，就可以完整將 DAG 轉換為多元樹狀結構了。

```

1   boolean SEARCH(ITEMdata SOURCEitem,String ITEMid)
2   {
3       ITEMdata DATAptr;
4       int FOUND=0;

9       DATAptr ← SOURCE->OUTdata;
10      while ( (FOUND == 0) && (DATAptr != NULL) )
11      {
13          if (DATAptr->ID==ITEMid)
14          {
15              FOUND=1;
17          }
18          DATAptr ← DATAptr.NEXT;
19      }
20      if (FOUND==1)
22          return true;
25      else return false;
48  }

```

圖 14：節點搜尋算法虛擬碼

此處必須要用到複製節點功能，其作法是先利用 ITEMdata 物件的建構子 (Constructor) 產生一個新節點，再將要被複製的原始節點的屬性 Attribute) 逐一指定給新節點，最後將此新節點當作回傳物件傳回給呼叫端，演算法如圖15 所示。

```
1  ITEMdata COPY_NEW_ITEM(ITEMdata SOURCEitem)
2  {
3  ITEMdata NEWitem=new ITEMdata(null,null,0);
4
5  NEWitem.ITEM ← SOURCEitem.ITEM;
6  NEWitem.ID ← SOURCEitem.ID;
7  NEWitem.LEVEL ← SOURCEitem.LEVEL;
8  NEWitem.NEXT ← SOURCEitem.NEXT;
9  NEWitem.OUTdata ← SOURCEitem.OUTdata;
10
11 return NEWitem;
12 }
```

圖 15：複製節點演算法

上述將 DAG 轉換成為普通樹狀結構的演算法，最主要步驟是把 DAG 中的每一節點另外複製後分別連結到兩個地方，一個是在對應的 GRAPH 階層陣列中，此部份是作為本身階層節點單元的串列紀錄。另一個是在該階層陣列父節點的 OUTdata 串列（如果此節點本身已經是對應到 GRAPH 的第一層階層 (GRAPH[0]) 則因為沒有父節點所以不需要複製到父節點的 OUTdata 串列），這一部份則是為了紀錄該節點對上一層節點的相依性追蹤用途。因為是用複製節點的方式分別連結，並且複製的新節點的 NEXT 串列被指定為 NULL，因此即可將多個父節點連結到一個子節點的情況轉換為每一個父節點只連結到一個子節點。如此可以簡化後續演算法並提升搜尋效率。詳細的新增節點流程圖顯示於圖16。

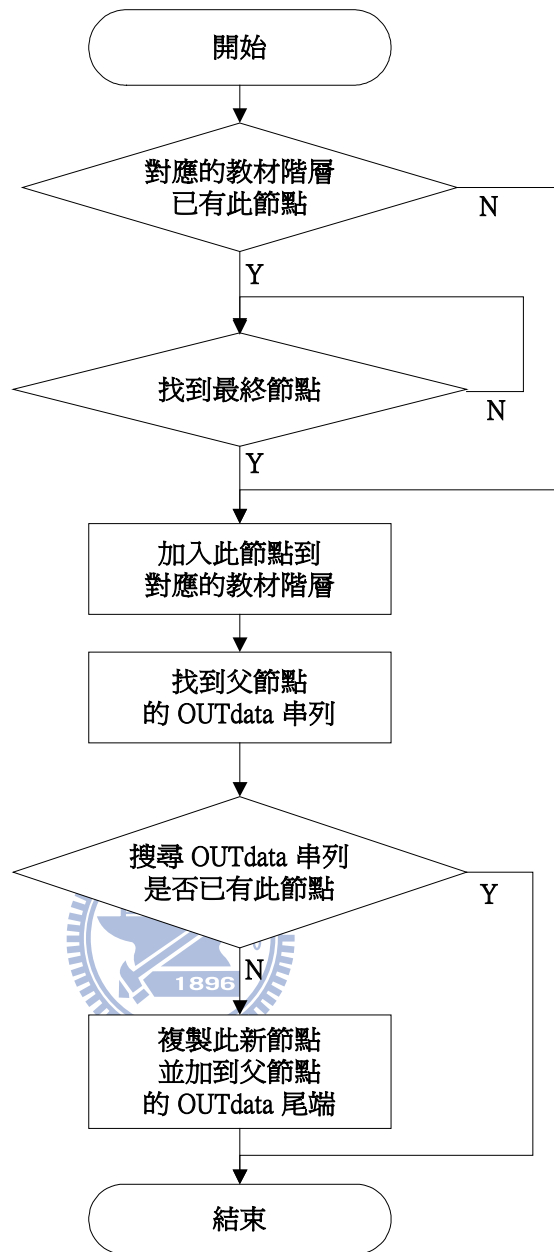


圖 16：加入新節點到 GRAPH 流程圖

以下以圖 12 的 SNE1、NLS1、NLS2、NLS3 等四個節點依序加入 GRAPH 變數的分解步驟作解說，其他節點依此類推。

(一) 初始狀態

GRAPH 的各個索引都指向 NULL，如圖 17 所示。

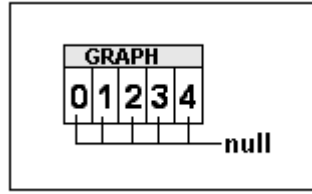


圖 17：GRAPH 的初始狀態示意圖

(二) 加入 SNE1 節點

GRAPH[0] 指向 SNE1 節點，如圖18 所示。

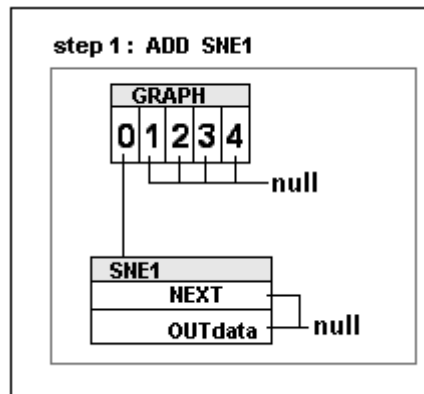


圖 18：加入 SNE1 到 GRAPH 的狀態示意圖

(三) 加入 NLS1 節點

GRAPH[1] 指向 NLS1 節點，並且複製 NLS1 成為一個新節點將其連結到父節點串列的 SNE1 的 OUTdata 串列，如圖19 所示。

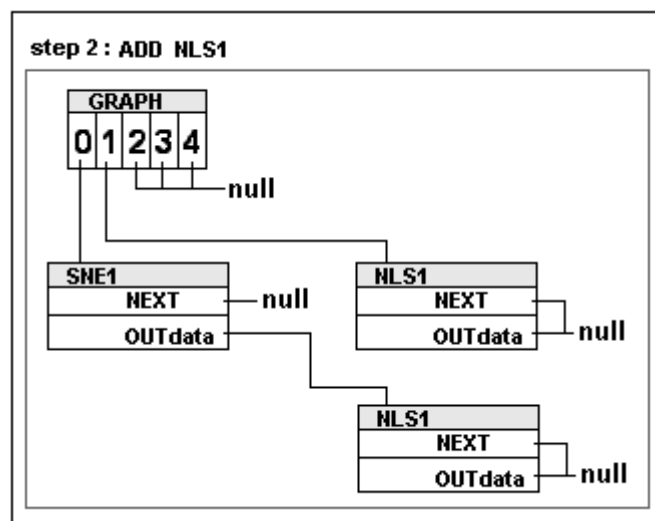


圖 19：加入 NLS1 到 GRAPH 的狀態示意圖

(四) 加入 NLS2 節點

NLS1 的 NEXT 指向 NLS2 節點，並且複製 NLS2 成為一個新節點將其連結到父節點串列的 SNE1 的 OUTdata 串列尾端，如圖20 所示。

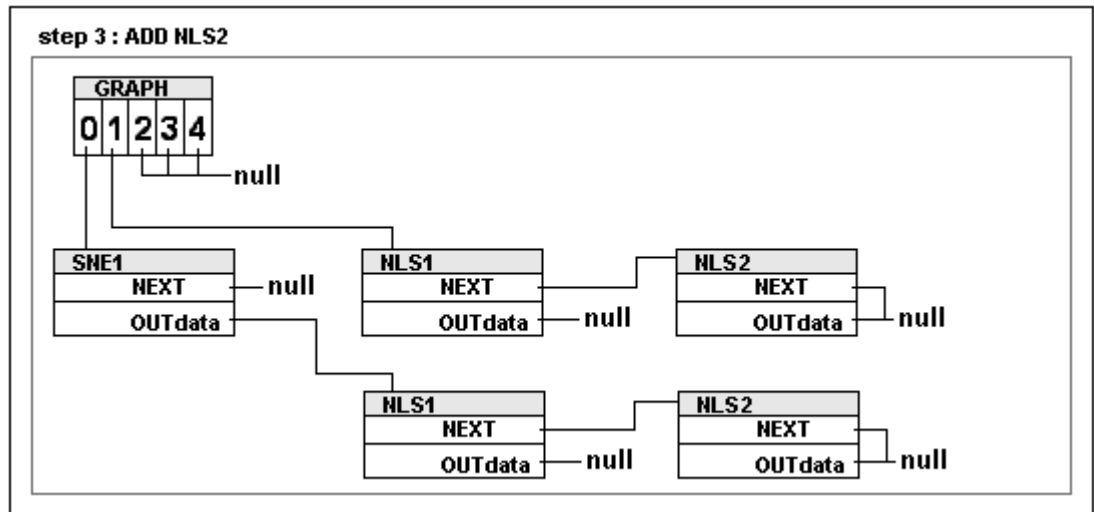


圖 20：加入 NLS2 到 GRAPH 的狀態示意圖

(五) 加入 NLS3 節點

NLS2 的 NEXT 指向 NLS3 節點，並且複製 NLS3 成為一個新節點將其連結到父節點串列的 SNE1 的 OUTdata 串列尾端，如圖21 所示。

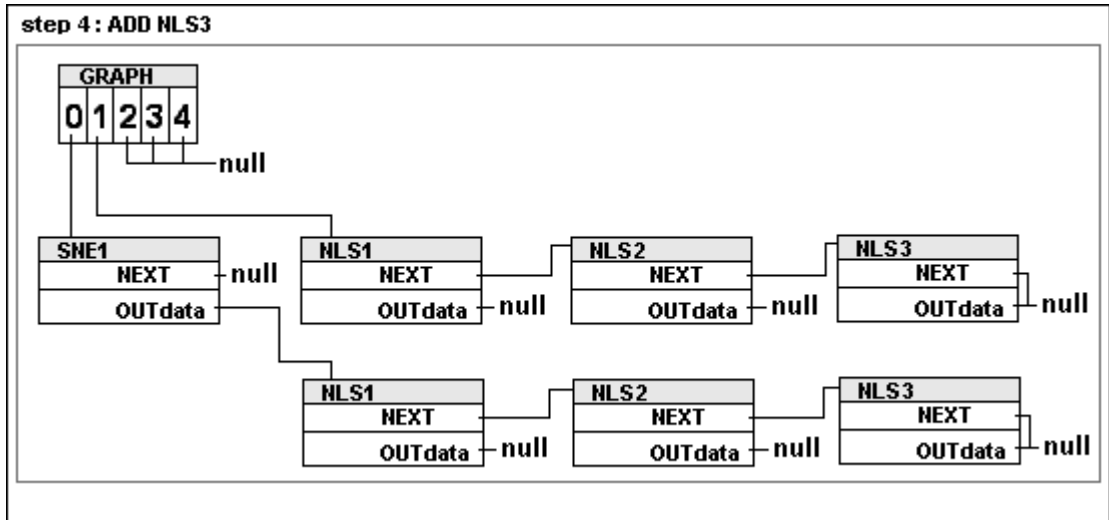


圖 21：加入 NLS3 到 GRAPH 的狀態示意圖

圖22 是建立以 GRAPH 為進入點的多元樹狀結構模型的演算法。其中第 7 列到第 8 列是判斷 GRAPH 中指定的索引階層是否已有節點串列,若還沒有節點串列(該索引階層指向 NULL),則將新加入的節點直接指定給 GRAPH。第 9 列到第 26 列是當 GRAPH 已經有節點串列,則搜尋此節點串列中是否已有此節點,若已有此節點則不重複加入。若還沒有,則將此節點加入到節點串列的尾端。第 28 列第 46 列則是在建立上一層之父節點的 OUTdata 串列作類似的操作。經過上述過程建立的模型,在串列中的節點不會重複出現,所以在後續做節點的新增、刪除、修改、搜尋等動作時,大幅降低了運算的複雜度並且有效提昇了操作效率。

```

1  ITEMdata ADD(String ITEM,String ITEMid,String PARENTid,int LEVEL)
2  {
3      ITEMdata SAMEdata = null,PREdata = null,DATAptr = GRAPH[LEVEL],NEWdata,OUTptr;
4      FOUND ← false;
5      HASsame ← false;
6      NEWdata ← new ITEMdata(ITEM,ITEMid,LEVEL);
7      if (DATAptr == null)
8          GRAPH[LEVEL] = NEWdata;
9      else {
10         while ((HASsame == false)&&DATAptr != null)
11             {
12                 PREdata ← DATAptr;
13                 if (DATAptr.ID == ITEMid)
14                     {
15                         HASsame ← true;
16                         SAMEdata ← DATAptr;
17                     }
18                 DATAptr ← DATAptr.NEXT;
19             }
20             if (HASsame == true)
21                 {
22                     NEWdata ← NEWdata.COPY_NEW_ITEM(SAMEdata);
23                     NEWdata.NEXT ← null;
24                 }
25             else PREdata.NEXT ← NEWdata;
26         }
27     if (HASsame == false) NEWdata ← NEWdata.COPY_NEW_ITEM(NEWdata);
28     if (LEVEL != 0)
29         {
30             DATAptr = (LEVEL==LEVELactor) ? GRAPH[LEVEL-2] : GRAPH[LEVEL-1];
31             while ((FOUND == false) && (DATAptr != null))
32                 {
33                     if (DATAptr.ID == PARENTid)
34                         {
35                             OUTptr ← DATAptr.OUTdata;
36                             if (OUTptr == null)
37                                 DATAptr.OUTdata ← NEWdata;
38                             else {
39                                 while (OUTptr.NEXT != null) OUTptr ← OUTptr.NEXT;
40                                 OUTptr.NEXT ← NEWdata;
41                             }
42                             FOUND ← true;
43                         }
44                     DATAptr ← DATAptr.NEXT;
45                 }
46         }
47     return NEWdata;
48 }

```

圖 22：加入新節點到 GRAPH 演算法

四、系統實作與導入

4.1 系統平台與開發工具

為因應多媒體教材開發製作可以多人同時上線進行，本系統採用網路瀏覽 (Web based) 的軟體操作介面，審核人員可以在瀏覽器 (Browser) 上操作執行，透過網際網路 (Internet) 連接本系統。為了讓系統有較高的效率與更好的集中管理和控制能力，本系統使用主從架構 (Client - Server)，如圖23 所示。

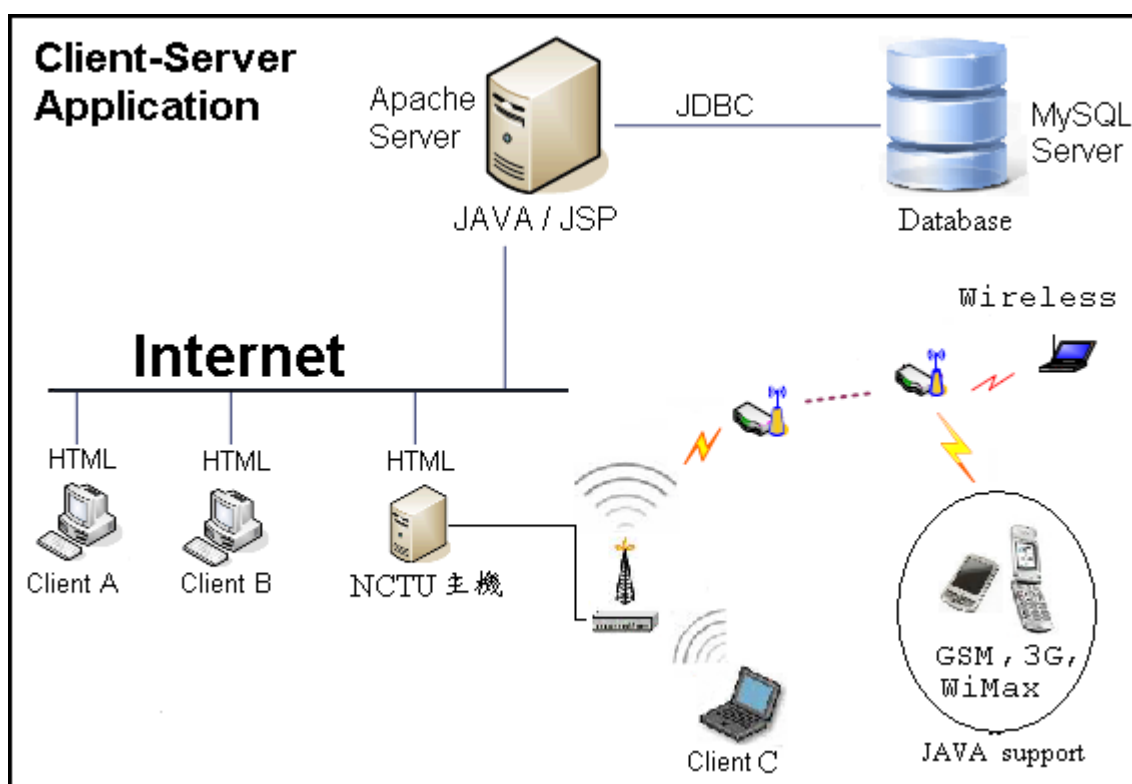


圖 23：MVMS 系統環境平台

客戶端 (Client) 通常是個人電腦或小型工作站，本身就具備完整獨立作業能力。伺服器端 (Server) 則是一台較大型的伺服器電腦主機，而在客戶端及伺服器端之間則藉由 TCP/IP 通信協定連結，形成通訊網路來互相傳遞資料。由客戶端

發出服務的請求，訊息傳給伺服器後，再由伺服器經由命令解譯決定所執行的動作。當執行動作需要存取資料時則經由資料庫驅動協定 (JAVA JDBC) 負責聯繫資料庫系統進行相關資料記錄處理傳遞，最後再將處理資料結果傳送到客戶端，資料的傳遞協定如圖24 所示。

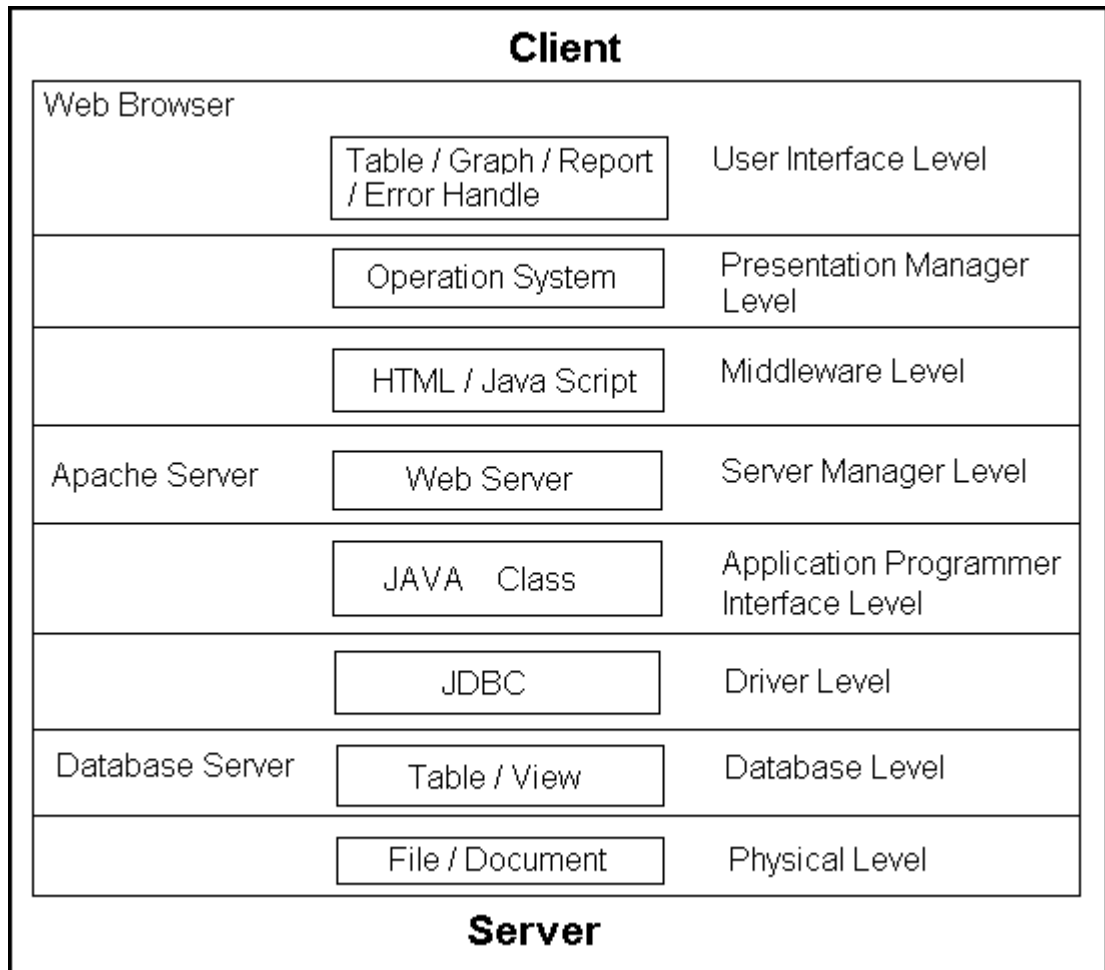


圖 24 : MVMS 系統資料傳遞示意圖

典型的多媒體教材品質管控系統 (Multimedia Quality Control System ,MQCS) 通常包含了以下幾個模組功能:

- (1) 需求管理 (Requirement Management) :
負責多媒體教材的需求新增、需求變更、需求審查、需求追溯等功能作業。
- (2) 開發工具規格整合 (Specification Integrated) :
針對各種多媒體教材製作工具的規格限制，提供整合轉換功能。
- (3) 驗證管理 (Verification Management) :
驗證多媒體教材產品與規格的一致性。
- (4) 確認管理 (Validation Management) :

確認多媒體教材產品符合顧客需求。

(5) 文件管理 (Documentation Management) 等功能。

提供管理系統中各子系統之間資料轉移所需的各式文件，並管理各子系統所產生之對應文件。

MVMS 系統可以整合到多媒體教材品質管理系統中如圖25 所示，使得多媒體教材品質管理系統的設計可以用模組化方式完成，在系統的製作和維護管理上都更具彈性而且更有效率。

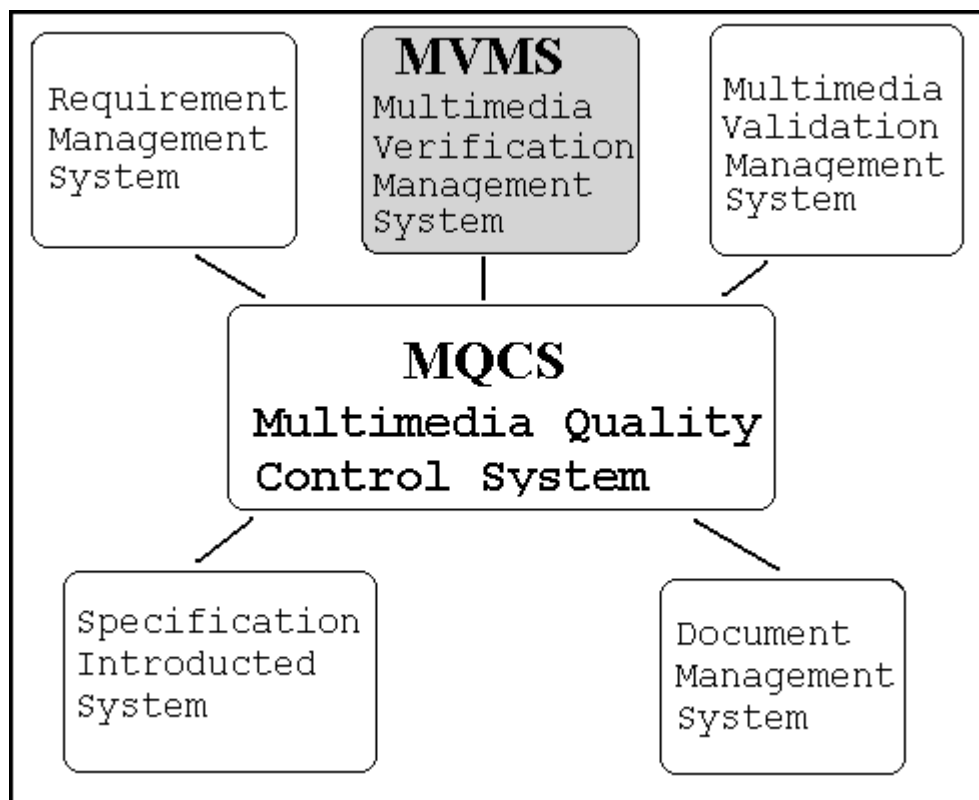


圖 25：典型多媒體教材品質管理系統

本系統採開放式架構開發環境，選擇的開發工具均是以開放源碼 (Open Source) 開發工具為主，使用者執行環境也是考慮以一般教材製作者普遍的操作環境。網路連線部分可以利用有線傳輸如纜線數據機 (Cable modem)、非對稱數位用戶迴路 (Asymmetric Digital Subscriber Line ,ADSL)、光學纖維 (Optical Fiber)，無線傳輸部分則如無線上網筆記型電腦 (Wireless Notebook Computer)、個人數位助理 (Personal Digital Assistant ,PDA) 或其他行動通訊裝置等，都可以透過網際網路連線使用本系統。

系統環境說明如下：

開發環境：

- 系統開發工具： JAVA JDK 1.5.014
- 程式語言： JAVA / JSP / HTML

伺服器端：

- 作業系統：Linux 平台
- 網站伺服器：Apache Tomcat 5.5
- 資料庫伺服器：MySQL 5.0

使用者端：

- 作業系統：Linux 平台、Microsoft Windows 95 以上。
- 網頁瀏覽器：Firefox 1.0 以上、Microsoft Internet Explorer 4.0 以上。

4.2 MVMS 系統架構



MVMS 系統架構主要分為教材驗證子系統和控制處理子系統兩部分，如圖26 所示。

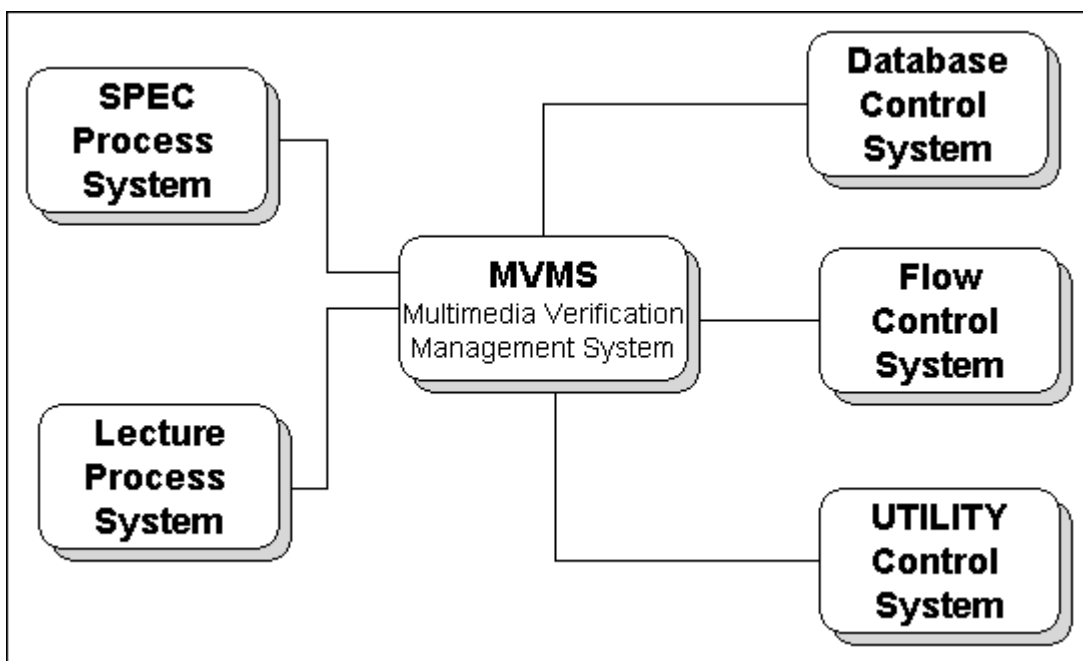


圖 26：MVMS 系統架構

教材驗證子系統負責處理有關多媒體教材相關的處理工作單元，包括規格模組、教材模組、章節結構。功能敘述如下：

1. 規格模組：記錄教材各單元的詳細規格，以及對應的轉換方式。
2. 教材模組：處理教材各單元的驗證，以及各單元間的相依關係。
3. 章節結構：將若干 SCO 單元依其所歸屬的章節關係組成一個子單元。

控制處理子系統部分則負責提供包括流程控制、資料庫存取操作、公用程式庫、繪圖處理、統計運算以及 XML 文件輸出等功能模組。功能敘述如下：

1. 流程控制：處理資料流的傳遞。
2. 資料庫存取：提供對資料庫系統的新增、刪除、更新、查詢等操作處理。
3. 公用程式庫：提供包括型態轉換、檔案系統操作等功能。
4. 繪圖處理：處理 DAG 及統計輸出圖形。
5. 統計運算：處理資料統計運算。
6. XML 文件：提供 XML 文件格式輸出。

圖27 是以統一塑模語言 (Unified Modeling Language,UML) 的類別圖 (Class Diagram) 來呈現 MVMS 定義的類別。

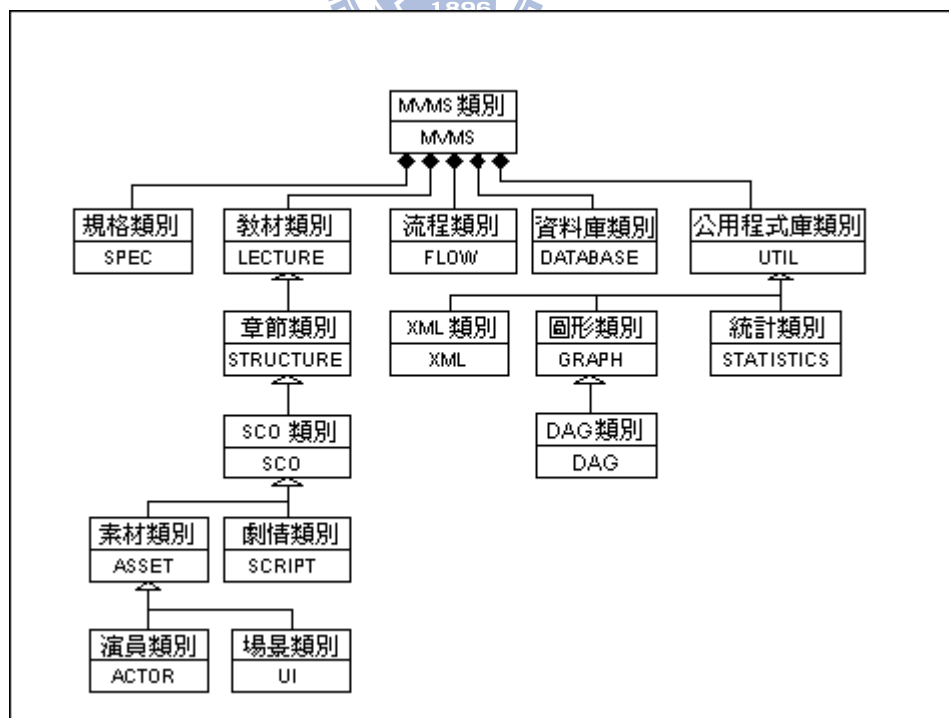


圖 27：MVMS 類別圖

4.3 系統製作

MVMS 系統功能性需求分為驗證準備、同仁審查、驗證執行、矯正措施界定四大模組，整合多媒體教材開發生命週期各階段系統製作方式，詳述如下：

1. 驗證準備

(1) 選擇需驗證的工作產品

首先建立一個新的驗證專案，然後選擇要驗證的多媒體教材當作驗證的工作產品。

(2) 建立驗證環境

WEB Server : Apache Tomcat Server

Database : MySQL Database

Compiler : JAVA Development Kit (JDK)

Program : JAVA、JSP、HTML

(3) 建立驗證程序及準則

驗證程序包括靜態檢查程序和動態驗證程序兩部分。驗證程序主要是將多媒體教材各個製作單元進行需求與規格的一致性檢核，如表7所示。

表 7：多媒體教材驗證準則

單元	驗證程序	驗證準則
教材單元	教材驗證程序	工作產品與需求規格一致性
章節結構單元	章節結構驗證程序	
SCO 單元	SCO 單元驗證程序	
分鏡腳本單元	分鏡腳本驗證程序	
劇情單元	劇情驗證程序	
演員素材單元	演員素材驗證程序	
場景素材單元	場景素材驗證程序	
相依性關係	相依性關係驗證程序	

靜態檢查是在教材製作過程中即開始對內容加以檢查，是查核教材內容找出錯誤、遺漏或異常的地方，其目的是為了降低當產品完成後再接受測試時的錯誤機率。本系統中選定以教材規劃審查、單元劇情概觀檢查、SCO 單元檢查、素材

單元檢查等子項目作為靜態檢查的檢查單元，如圖 28 所示。圖 29 是以 UML 的使用案例圖 (Use Case Diagram) 來呈現靜態檢查程序。

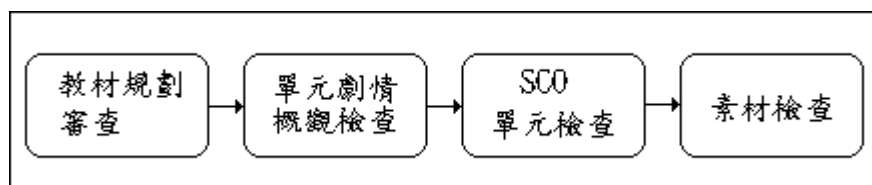


圖 28：多媒體教材靜態檢查程序

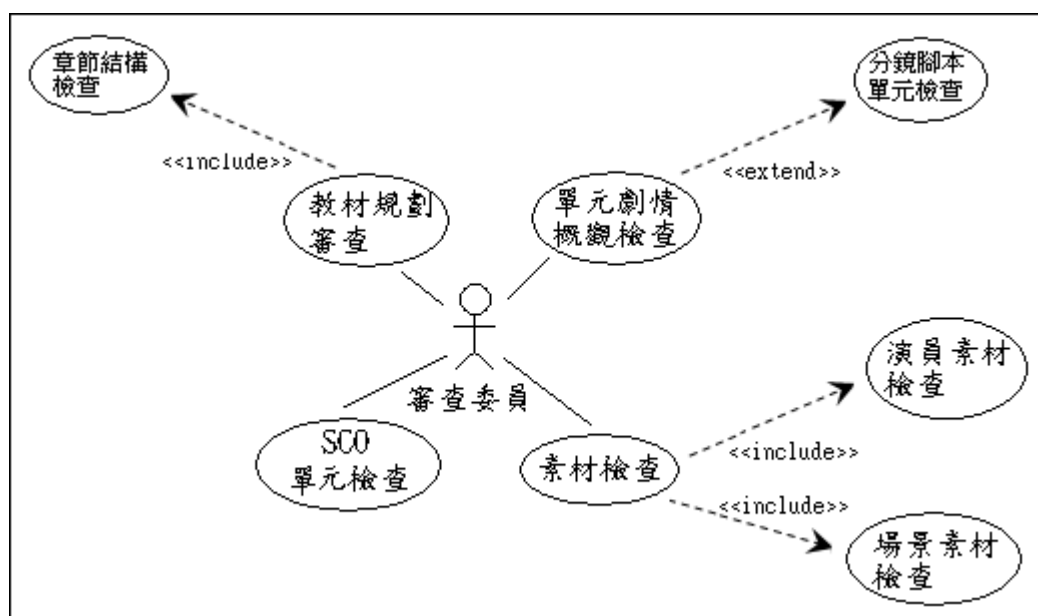


圖 29：多媒體教材靜態檢查使用案例圖

動態驗證程序建立以下五個驗證單元分別是微單元驗證、模組驗證、子系統驗證、系統整合驗證以及可接受性驗證，動態驗證的各驗證單元如圖 30 所示。



圖 30：多媒體教材動態驗證單元

動態驗證的各單元驗證細項列示如下。

- 微單元驗證：場景素材、演員素材。
- 模組驗證：素材類別、劇情單元類別、章節結構、教材類別、規格模組。
- 子系統驗證：附件資料驗證、劇情串接驗證、互動式劇情驗證。
- 系統整合驗證：劇情素材整合、附件資料整合、整合相依性。
- 可接受性驗證：單元可接受性、模組可接受性。

2. 同仁審查

根據 CMMI 規範的說明，同仁審查是利用有條理的方法步驟來檢查工作產品，界定需移除之缺失並建議其他需變更事項。此項工作必須是由產品製作人員以外的其他同仁執行。同仁審查是驗證步驟中重要而且有效的驗證方法，經由檢查、結構化逐步審查或其他經證實的審查方式來執行。

在本研究中同仁審查除了應用於多媒體教材專案上之外，亦應用在產出之工作相關分析報表和文件上。

從多媒體教材的開發生命週期來看，同仁審查必須在系統規格制定、需求制定等步驟開始時即加入審查機制，以確保制定的規格符合需求。同仁審查的功能使用案例如圖32 所示。

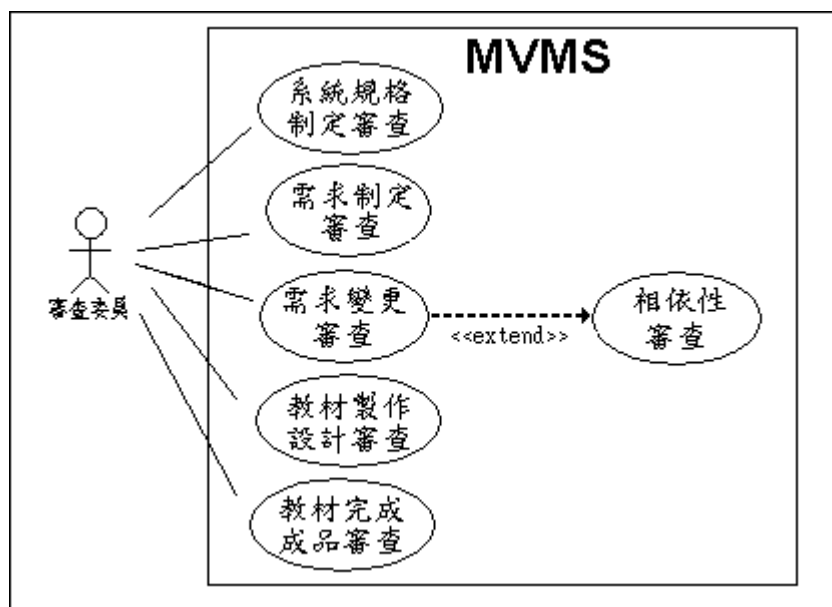


圖 31：同仁審查使用案例圖

(1) 準備同仁審查

同仁審查的準備工作包括準備審查工作產品，也就是指定多媒體教材專案。指定審查人員，以及制定審查原則。審查原則是依據教材內容、教學設計、學習導引、學習評量等原則考量予以制訂，參考表8 說明。



表 8：同仁審查原則

審查項目	審查原則	審查角色
教學主題	內容學習性	審核委員
學習領域	內容領域專業性	
適用年級	內容學習性、教學適當性	
設計理念	內容正確性、教學創新	
能力指標	製作技巧、創意表現	
教學目標	內容表達性	
教學內容	學習一致性、教學適當性	
教學流程	學習一貫性	
教學資源	媒體運用、教材相容性	
版權問題	是否侵權	

此外，準備工作還必須訂定審查時程，審查時程表如表9 所示。

表 9：多媒體教材審查時程

審查時程	說明
系統規格制定	多媒體教材規格審查 輸出文件規格審查
需求制定階段	教材需求制定審查 單元規格需求審查 系統規格需求審查
需求變更申請	單元需求變更申請審查 相依性追蹤審查
教材製作設計	場景素材單元審查 演員素材單元審查 劇情單元審查 分鏡表單元審查 章節結構審查
教材製作完成	教材成品專案審查

(2) 進行同仁審查

在多媒體教材製作生命週期中，同仁審查會分別在各個單元階段進行，包括系統規格制定、需求規格設計、需求變更申請、教材單元製作以及教材製作完成等階段。此處同仁審查是會針對多媒體教材的需求(Requirement)、架構 (Architecture)、設計 (Design)、產品 (Product) 等進行技術性審查。圖32 是審查序列圖(Sequence Diagram)。

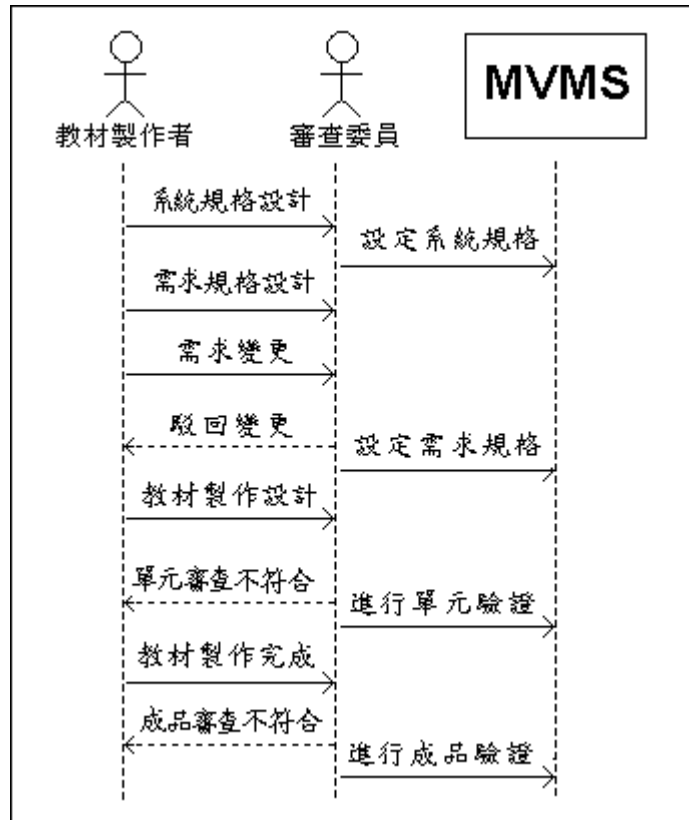


圖 32：同仁審查序列圖

同仁審查也可以針對教學個案或各個別內容主題訂定審查議題進行審查。例如以下的編續教學審查為例。

- A. 學生是否可以經由自己確實閱讀、填寫、進而學會教材。
- B. 是否提供正確答案，讓學生立即作答，核對學習結果。
- C. 是否可以讓學生選擇符合自己的興趣與需求學習。
- D. 是否提供教師可以根據學生學習的紀錄，適時修正教材。

同仁審查依據審核原則在教材設計及製作各階段協助查核教材各單元錯誤、遺漏或異常的地方。常見的單元錯誤如表 10 所示。

表 10：常見錯誤型態

錯誤型態	錯誤檢查	範例說明
資料錯誤	Data Type check	未提供面版資源檔案
	Initial check	初始位置超出邊界值
	Naming rule check	腳本名稱違反命名規則
流程錯誤	Script Sequence check	劇情連接順序錯誤
	Flow check	學習評量流程錯誤
	Synchronism check	演員退場同步機制錯誤
介面錯誤	Append File check	素材附件檔未提供
	Version check	同時連結不同版本素材
例外處理	Reverse trace check	回溯追蹤節點單元錯誤
	Except process	例外處理警示系統

(3) 分析同仁審查資料：

當所有待審的工作產品都已經完成詳細審查，審查委員必需針對審查結果製作審查報告，報告內容包括缺失項目、審查效率、原因分析以及建議改善事項等，並擬定缺失改善計畫作為改善依據。缺失改善的分析步驟說明如表11 所示。

表 11：缺失改善分析步驟

分析步驟	說明
缺失問題	教材製作者及審查團隊共同於審查過程中發現的缺失或問題
原因分析	討論及確定缺失的發生原因並提出預防的建議或措施，以避免類似的問題再次發生
工作產品修整	視審查結果決定修整內容及範圍
進度稽核	依照進度進行工作成果的稽核
後續追蹤	持續追蹤直到缺失處理完成為止

3. 驗證執行

(1) 執行驗證程序

依據前面驗證準備單元所建立的驗證程序進行驗證步驟，圖33 以 UML

的活動圖 (Activity Diagram) 來表示驗證執行活動。

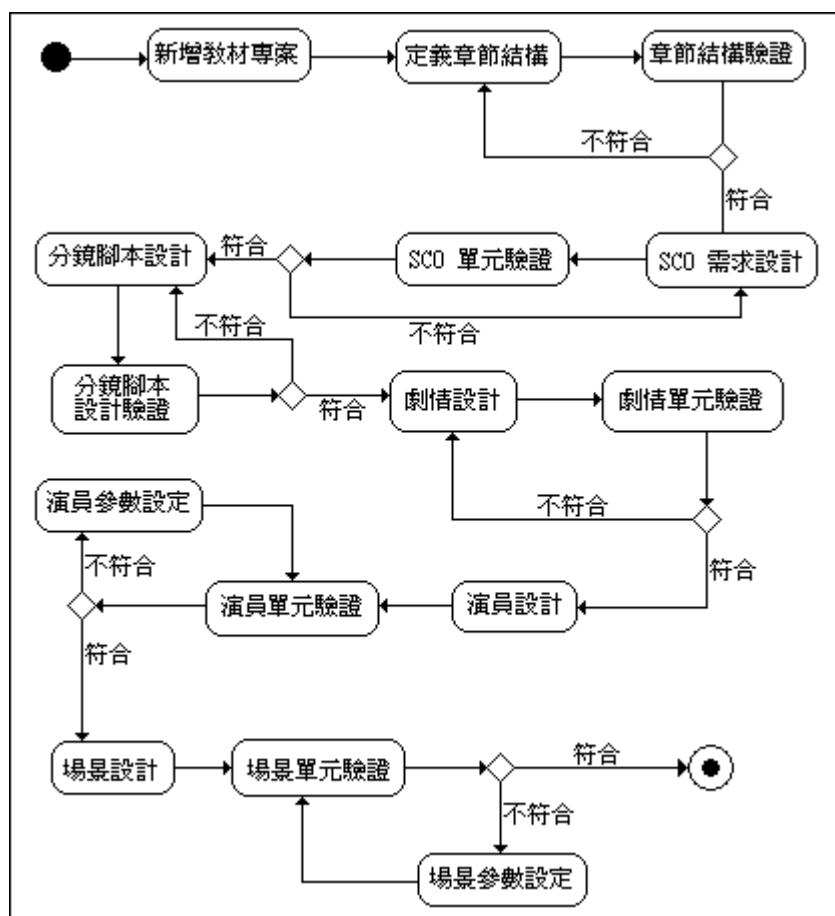


圖 33：驗證執行活動圖

(2) 分析驗證結果

驗證分析將各單元內部詳細規格值及實際值加以比對，並針對不一致項目分析其原因以及影響範圍。例如，當演員微單元的位置及透明度項目發生不一致情形，依據分析結果可以確認若演員的位置還在誤差範圍內，則對執行結果並無影響，但透明度的誤差過大則可能導致轉場特效雖然有執行但操作者無法看見此效果。

此外，驗證分析也針對缺失項目作統計運算，並繪製出在統計學中品質管控方法常用的柏拉圖 (Pareto Chart) 協助尋找關鍵不一致項目以及盒圖 (Box Plot) 幫助分析缺失項目分佈情形，以提供系統設計者及教材製作者作為改善依據。

(3) 提供驗證報告

當教材專案完成驗證步驟，系統根據多媒體教材開發資料庫當中的教材組成元件、制定規格與檔案異動的相關資訊，自動分析並整理出多媒體教材品質驗證報表。其中包括腳本分鏡表驗證清單、素材附件驗證清單、演員規格驗證清單、場景規格驗證清單以及多媒體教材單元驗證狀態表等內容。多媒體教材開發人員可依據報表所分析的不一致性資訊，檢查並修正相關的教材組成元件，直到整個教材通過完整驗證程序。

4. 矯正措施界定

(1) 分析議題

由於造成多媒體教材品質問題的原因眾多，除了規格不一致的情形之外，尚包括一些難以用系統化方式解決的主觀議題。例如素材製作的精緻程度、視覺上的美觀感覺、教材內容深度、畫面呈現技巧等。這些類型的品質問題較適合用成立特定議題方式加以分析改善以與一般性的缺失問題作區隔，如此可以提高系統的實用性，以下是建立特定議題時的一些參考項目。

(一) 問題界定

問題的陳述。

問題的可能成因。

問題的影響範圍。

相關系統設定條件。



(二) 備選方案分析

方案策略目的。

備選方案描述。

預期效益分析。

可行性分析。

(三) 方案執行建議

方案採行依據。

預期效益分析。

(2) 採取矯正措施

矯正措施的制定，必須依據缺失問題的性質提供針對性的措施。例如針對可量化的數據規格，可以由系統依據誤差範圍自動計算適當數據配合矯正改善。而對於特定議題的處理，則必須經由議題分析結果來制定矯正措施。

為解決缺失問題之發生原因，而擬定改善對策，並找出潛在可能影響品質之因素，以防止缺失問題再度發生。擬定之改善對策首先必須確認適用範圍以及權責單位，此處主要是多媒體教材製作者和審核委員。此外，若是大型教材專案，則還必須包含專案經理、資源素材供應商等人員。

執行矯正措施的作業實施內容包含提供缺失問題報告，詳細描述缺失狀況，進行缺失問題之調查，並正確分析以確認原因。接著擬定出可執行之改善對策據以執行。審核委員負責監督確認實施效果，若矯正效果符合預期，則建立檔案紀錄，以供後續預防措施制定之參考。若矯正效果不符預期，則必須重新分析異常原因，並修正矯正措施，直到執行成效符合預期結果。矯正措施實施作業參考如圖34 所示。

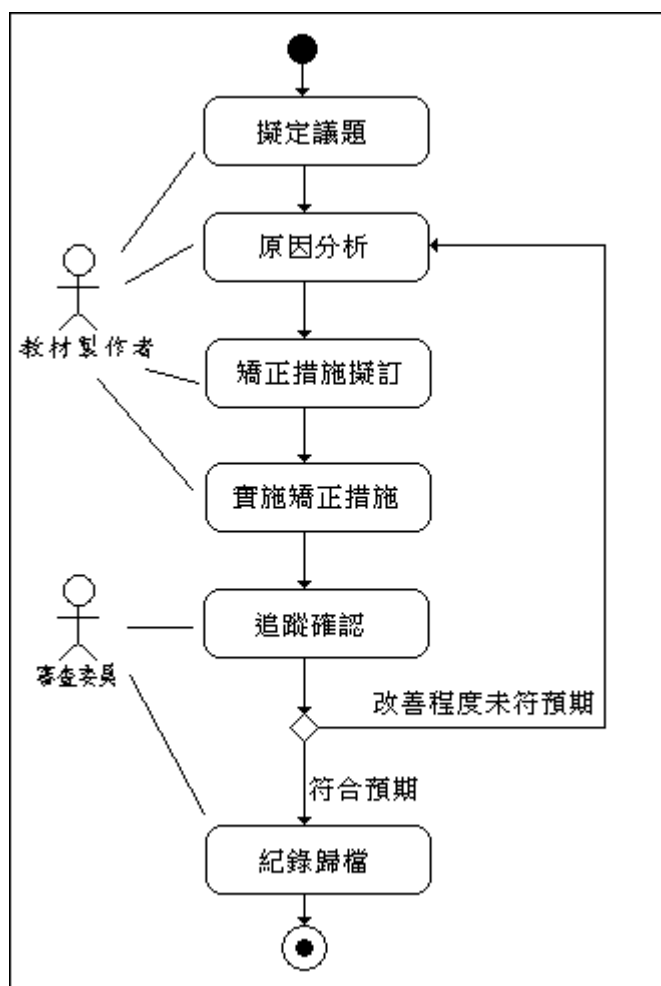


圖 34：矯正措施作業活動圖

五、實例應用與分析

在本章節中將以實際多媒體教材為例，說明如何使用第四章所建構的 MVMS 各模組功能及流程來展現如何運用多媒體教材開發流程配合執行驗證管理。本章節將實際使用數個多媒體教材原始教案，作為系統測試樣板範例，並分析其驗證結果。本章節的主要解說範例是多媒體教材 “What is on the Mat?” 這是新竹市政府為配合行政院「挑戰2008 國家發展計劃—E 世代人才培育計畫」以提昇全民英語能力，推動英語教育為首要的政策方針，自民國九十一年度起全面於新竹市實施國民小學一至六年級的英語教學教材。本章節將以實例展示配合驗證流程領域方法將多媒體教材原始教案 “What is on the Mat?” 進行實際驗證步驟，並分析其驗證結果。

此外，本系統亦提供將歷史教材專案相關驗證數據進行統計分析功能，並根據系統需要繪出如柱狀圖 (Bar Chart)、趨勢圖 (Trend Chart)、盒圖 (Box Plot)、柏拉圖 (Pareto Chart)、等相關統計圖表數據，藉以分析造成不一致性主要項目及可能原因以提供使用者和系統作為流程改善依據。



5.1 驗證準備

1. 選擇工作產品

此處根據英語教學課本的全文電子檔案內容，選擇要驗證的多媒體教材的原始教案，並選定英語課本的教案章節內容為驗證工作產品。此處我們選定教材 “What is on the Mat?” 作為驗證工作產品。“What is on the Mat?” 是新竹市政府為配合行政院「挑戰2008 國家發展計劃—E 世代人才培育計畫」以提昇全民英語能力，推動英語教育為首要的政策方針，並且自民國九十一年度起全面於新竹市實施國民小學一至六年級的英語教學教材。本工作產品內容包含一個章節結構，一個分鏡腳本單元，九段劇情，劇情編號從 SCR001 到 SCR009，十三個演員素材，演員編號從 Actor000 到 Actor012，以及一個場景素材 編號 UI001 ，其中演員素材包括圖像型態及文字型態。教材播放畫面如圖 35 所示。

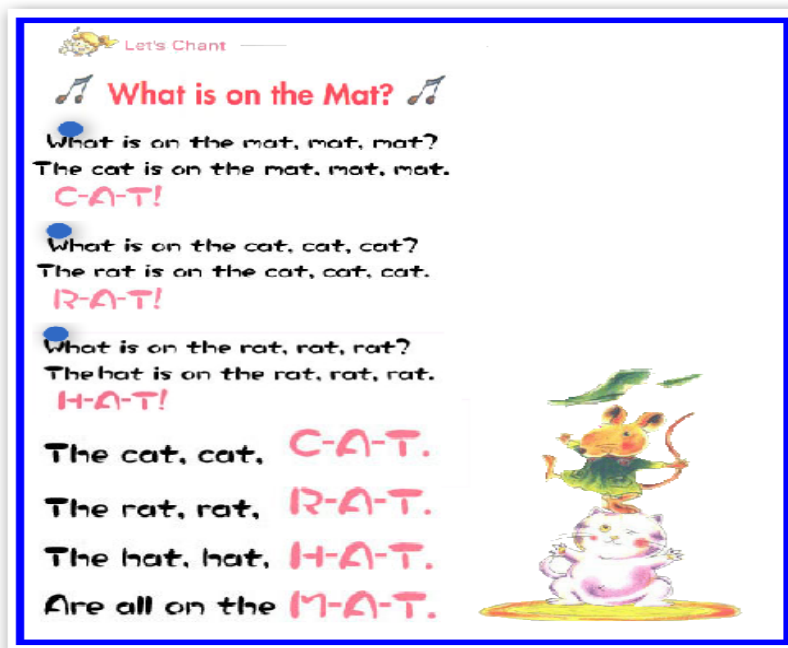


圖 35：多媒體教材 “What is on the mat?” (Source：英語教學教材)

2. 教材需求內容

根據原始教材需求分析，本教材可以分解為一個分鏡表，九段劇情，十三個演員素材並且所有劇情共用一個場景素材。圖 36 中以劇情三作為原始教材的劇情編排方式解說，圖 37 則是原始教材對應到 SCORM 規範的 Content Organization 對應關係。

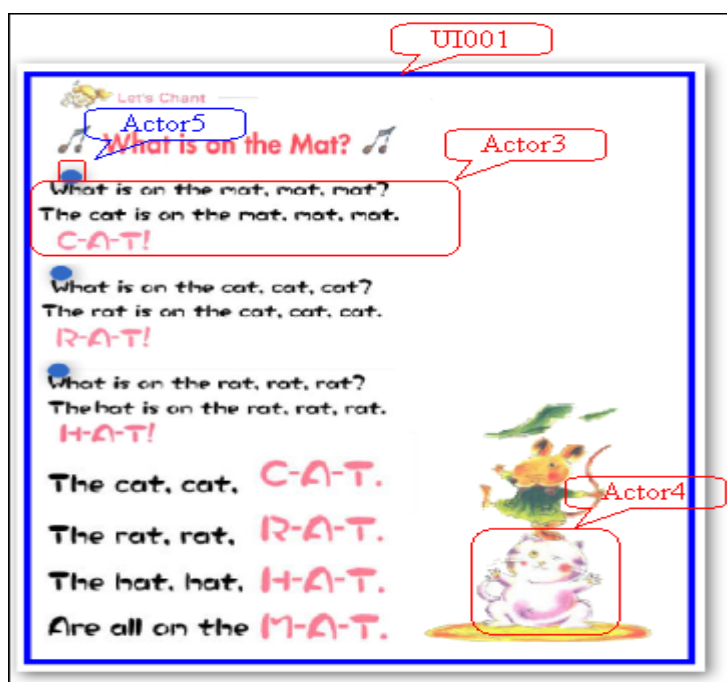


圖 36：劇情三素材對應圖

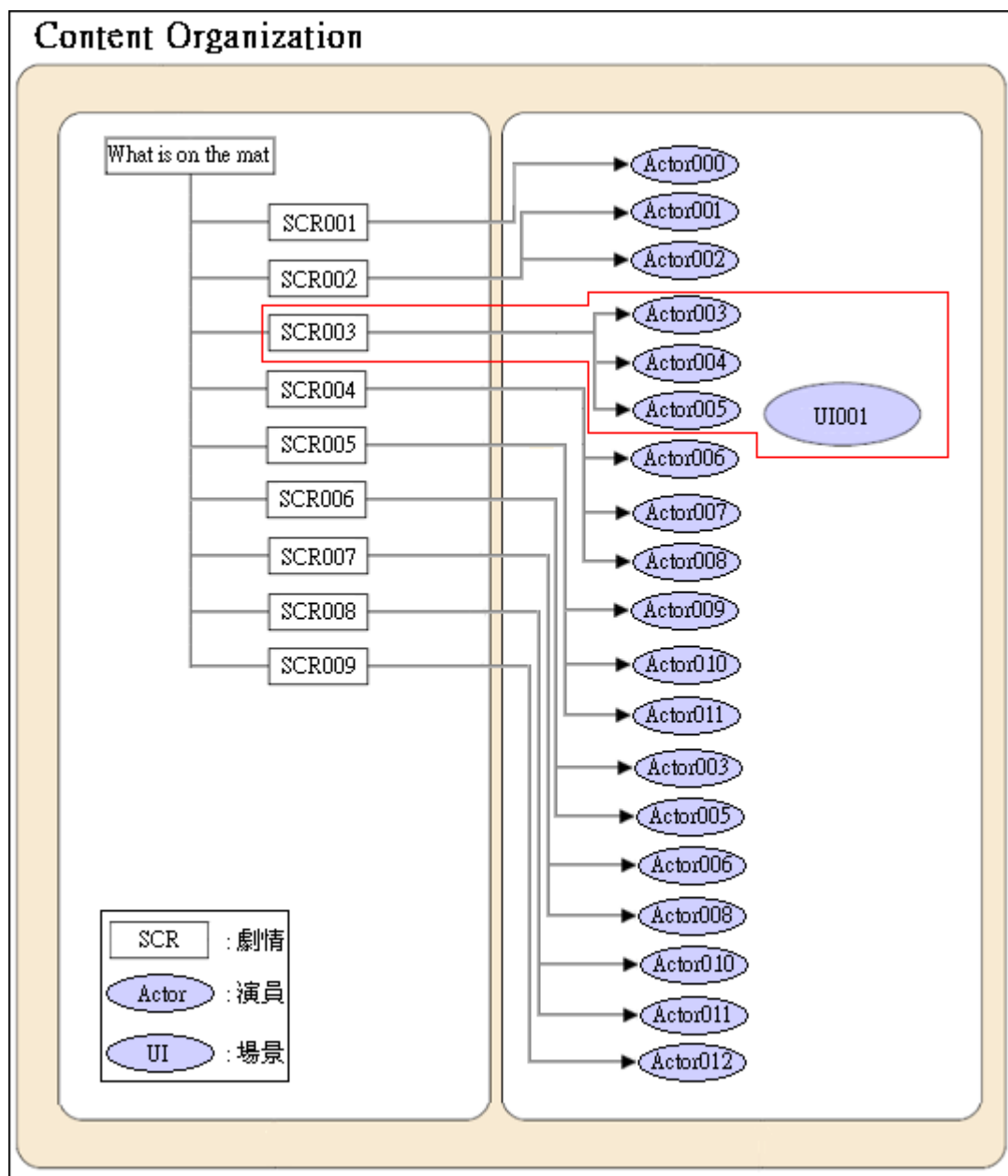


圖 37：Content Organization 對應關係

在圖 37 的內容組織圖中將多媒體教材的九個劇情對應至編號從 SCR001 到 SCR009。十三個演員素材對應至編號從 Actor000 到 Actor012。場景素材對應至編號 UI001。圖中也以劇情三（編號 SCR003）作為解說範例，該劇情包含了三個演員素材分別是 Actor003、Actor004、Actor005，以及共用的一個場景素材 UI001。圖 38 是建立多媒體教材主要單元結構執行畫面，圖 39 是建立素材清單執行畫面。

MVMS 多媒體教材 兒童英語教材 成人英語教材		教材資料	SCO 單元	素材清單	場景	演員
MQCS		建立教材單元結構				
同仁審核		項目	編號	組成單元	標題	
單元驗證		教材	LEC001	STR001	what is on the mat	
教材驗證		章節結構	STR001	SCE001	chapter 1	
相依性驗證		分鏡腳本	SCE001	SCR001	script 1	
矯正措施界定		劇情1	SCR001	Actor000,UI001	Lets chant	
不一致性統計表		劇情2	SCR002	Actor001,Actor002,UI001	What is on the mat	
不一致性項目分佈		劇情3	SCR003	Actor003,Actor004,Actor005,UI001	mat mat mat	
演員屬性關係曲線		劇情4	SCR004	Actor006,Actor007,UI001	cat cat cat	
各別教材統計分析		劇情5	SCR005	Actor010,Actor009,Actor011,UI001	rat rat rat	
		劇情6	SCR006	Actor003,Actor005,UI001	mat	
		劇情7	SCR007	Actor006,Actor008,UI001	cat	
		劇情8	SCR008	Actor010,Actor011,UI001	rat	
		劇情9	SCR009	Actor012,UI001	Are all on the mat	

圖 38：建立教材主要單元結構畫面

MVMS 多媒體教材 兒童英語教材 成人英語教材		教材資料	SCO 單元	素材清單	場景	演員
MQCS		建立素材清單				
同仁審核		項目	編號	檔案內容	附件檔案	標題
單元驗證		場景	UI001		@RESEBG001.jpg	1st UI
教材驗證		演員0	Actor000		@RESIMAGE001.png	let's chant
相依性驗證		演員1	Actor001		@RESIMAGE002.jpg	title
矯正措施界定		演員2	Actor002		@RESIMAGE003.png	mat
不一致性統計表		演員3	Actor003		@RESIMAGE004.jpg	what is on the mat
不一致性項目分佈		演員4	Actor004		@RESIMAGE005.png	cat
演員屬性關係曲線		演員5	Actor005		@RESIMAGE006.png	ball1
各別教材統計分析		演員6	Actor006		@RESIMAGE007.jpg	what is on the cat
		演員7	Actor007		@RESIMAGE008.png	rat
		演員8	Actor008		@RESIMAGE009.png	ball2
		演員9	Actor009		@RESIMAGE010.png	hat
		演員10	Actor010		@RESIMAGE011.png	what is on the rat

圖 39：建立素材清單畫面

根據表 12 的單元結構以及表 13 的素材清單，系統經組合分析後自動產生一個包含一組分鏡表單元、九段劇情單元、十三個演員素材，並且所有劇情共用一個場景素材的多媒體教材非循環有向圖 (DAG)，如圖 40 所示。

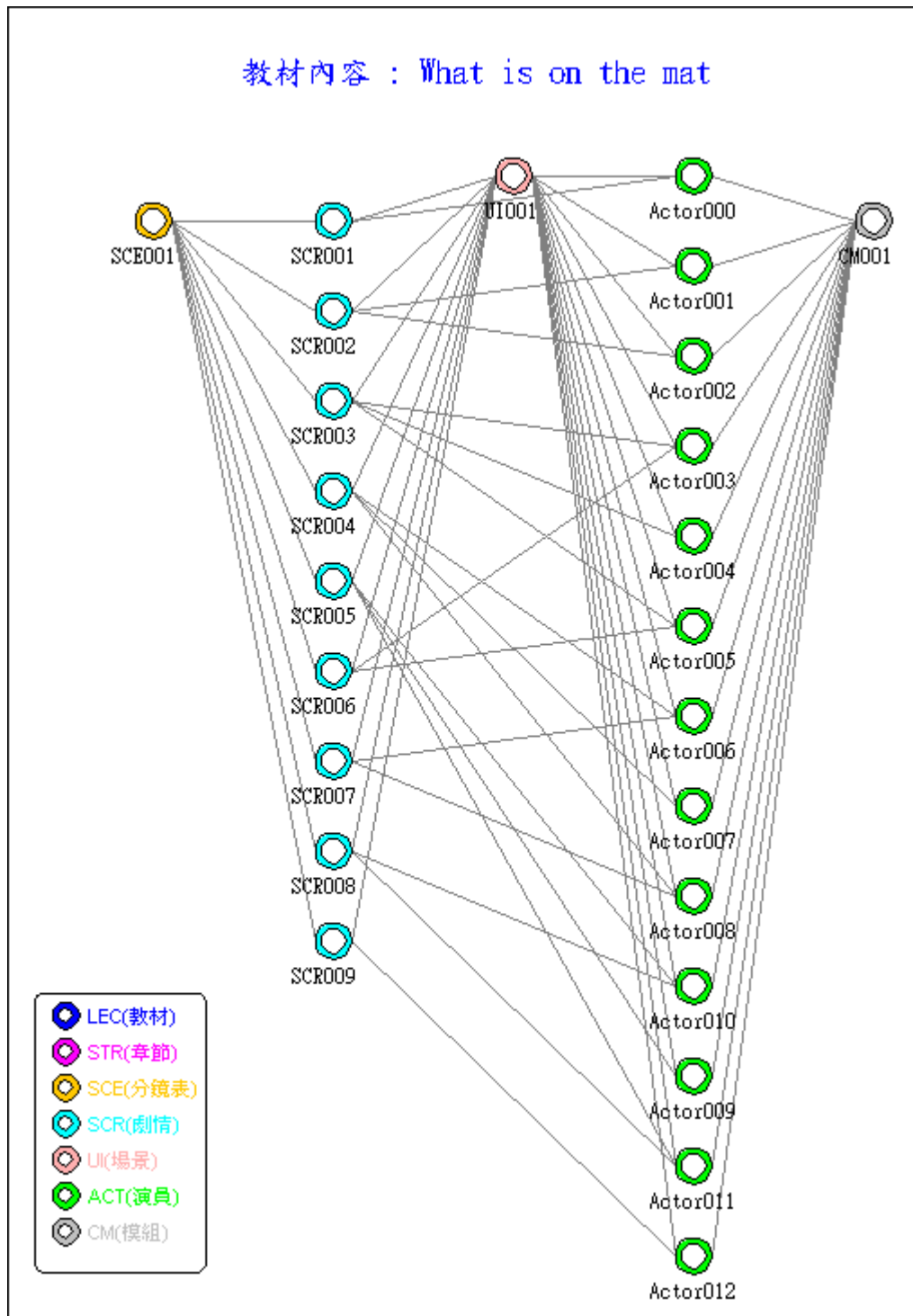


圖 40：教材 “What is on the mat?” 非循環有向圖 (Source: MVMS 繪製)

3. 建立相關驗證程序

依據教材劇情組織結構經系統重新組織後建立相關驗證程序，各單元、

組織對應驗證程序建立執行畫面如圖 41 所示。

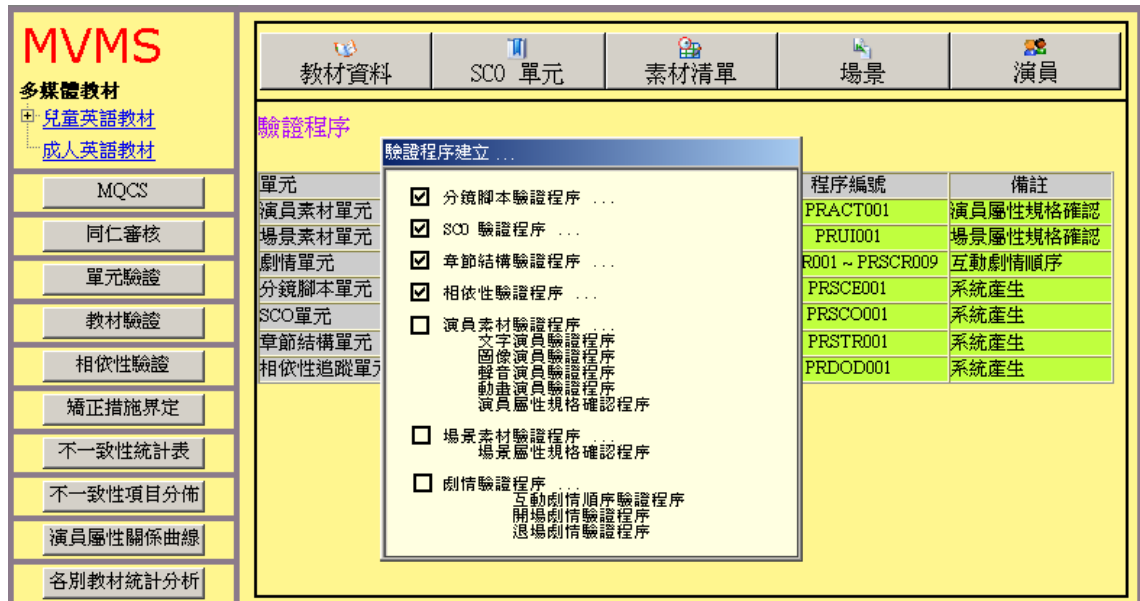
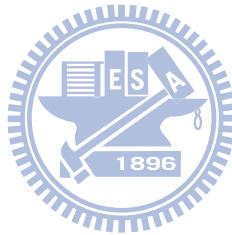


圖 41：建立驗證程序執行畫面

5.2 同仁審查



系統紀錄的同仁審查資料如下單元素材、附加檔案、教材規劃文件、學習策略、評量表、教材內容清單 … 等相關資料，並且針對各單元內容作適當的審核。審核委員依據審核原則針對教材相關文件內容、附檔進行審查，並在系統上確認審查項目進度，如圖 42 所示。確認完成後，系統顯示審查結並將記錄儲存於資料庫中。

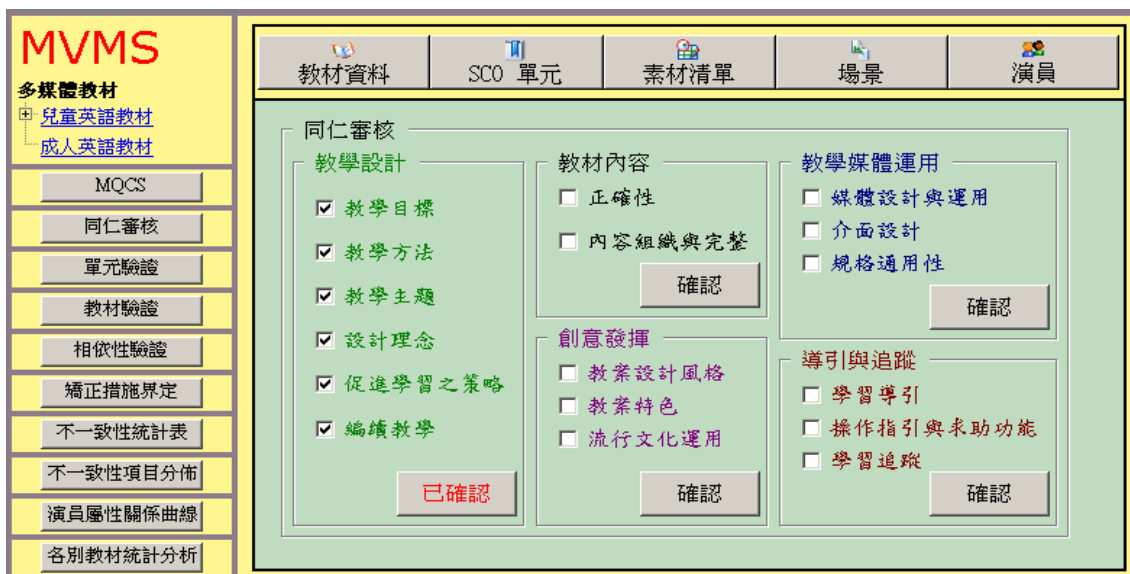


圖 42：同仁審核確認執行畫面

圖 43 是以教學設計中的編續教學審核為例，當操作者點選編續教學核對盒 (Check Box) 時顯示編續教學子項目視窗供審核者確認，並將確認後的資訊存入審核紀錄中。

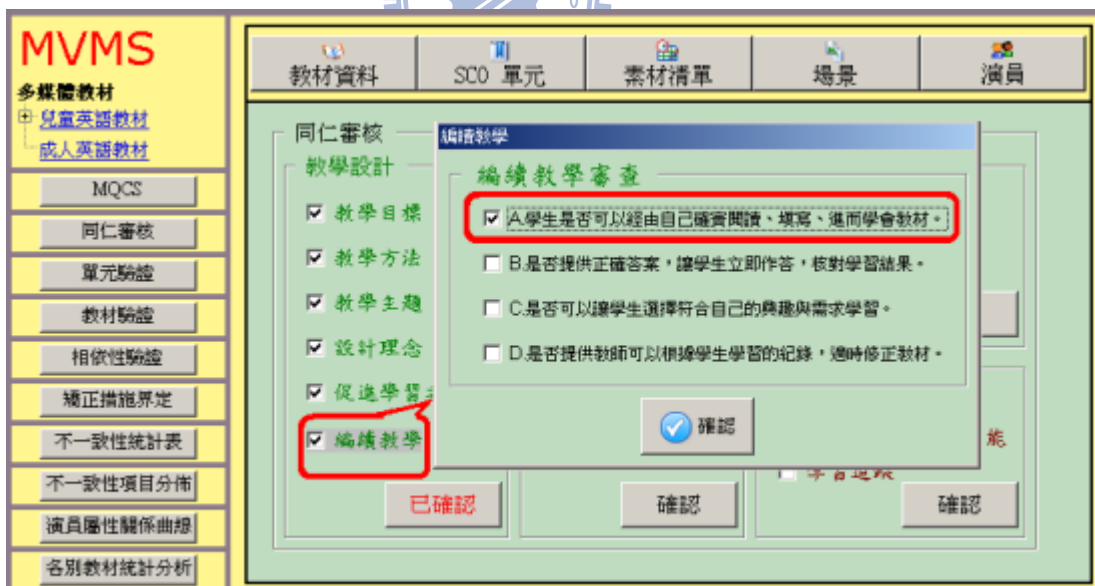


圖 43：編續教學審核畫面

5.3 驗證執行

當教材各單元經過審核後，系統建立審核紀錄資料並將教材進行狀態設定為驗證準備狀態，教材製作者可選擇進行系統驗證。驗證進行時系統根據教材章節架構以及各單元擬定義好的驗證程序執行驗證以及需求規格，針對各單元分別進行屬性驗證及相依性驗證，圖 44 是劇情單元驗證執行中畫面，每一單元經驗證後會將結果儲存於單元驗證紀錄中，最後並將匯整至驗證輸出清單。若最終驗證結果已正確無誤，則系統除了匯整驗證輸出清單之外，還將另外建立 XML 結構文件於資料庫中，使用者可以經由輸出功能選項輸出 XML 文件，作為與其它多媒體教材製作工具的文件格式轉換平台。



圖 44：劇情單元驗證進行

以演員素材屬性驗證為例，當驗證過程中發現透明度屬性值不符需求規格，系統顯示錯誤屬性欄位並將實際值和規格值對照輸出。隨後進行的相依性追蹤驗證將教材中的相關節點單元標示出相依性錯誤如圖 45 所示。

MVMS 多媒體教材 兒童英語教材 成人英語教材		教材資料	SCO 單元	素材清單	場景	演員
MQCS	驗證單位	屬性驗證	相依性驗證	審核記錄驗證	附加檔案驗證	
同仁審核	場景素材單元	√	√	√	√	
單元驗證	演員素材單元	×	×	√	√	
教材驗證	劇情單元	√	×	√	--	
相依性驗證	分鏡腳本單元	√	×	√	--	
矯正措施界定	SCO 單元	√	×	√	--	
不一致性統計表	章節結構單元	√	×	√	--	
不一致性項目分佈	教材專案單元	√	×	√	--	
演員屬性關係曲線	錯誤單位	編號	錯誤項目			
各別教材統計分析	演員素材單元	Actor004	透明度(實際值=60,規格值=70)			
	劇情單元	SCR003	相依性			
	分鏡腳本單元	SCE001	相依性			
	場景素材單元	UI001	相依性			
	章節結構單元	STR001	相依性			
	教材專案單元	LEC001	相依性			

圖 45：單元驗證清單執行畫面

5.4 矯正措施界定

審核人員針對本教材的分析，列出三項議題如下：

1. 將語言元素融入情境學習的分析。
2. 章節編排缺乏與學習者互動探討。
3. 如何加強提供學習者支援與協助。

同時，也針對個別議題提出矯正建議措施，提供教材製作者的矯正參考依據，如圖 46 所示。經由矯正執行後再重新進入審核、驗證步驟，直到整個驗證流程完成，最後再將相關矯正紀錄儲存於資料庫中。

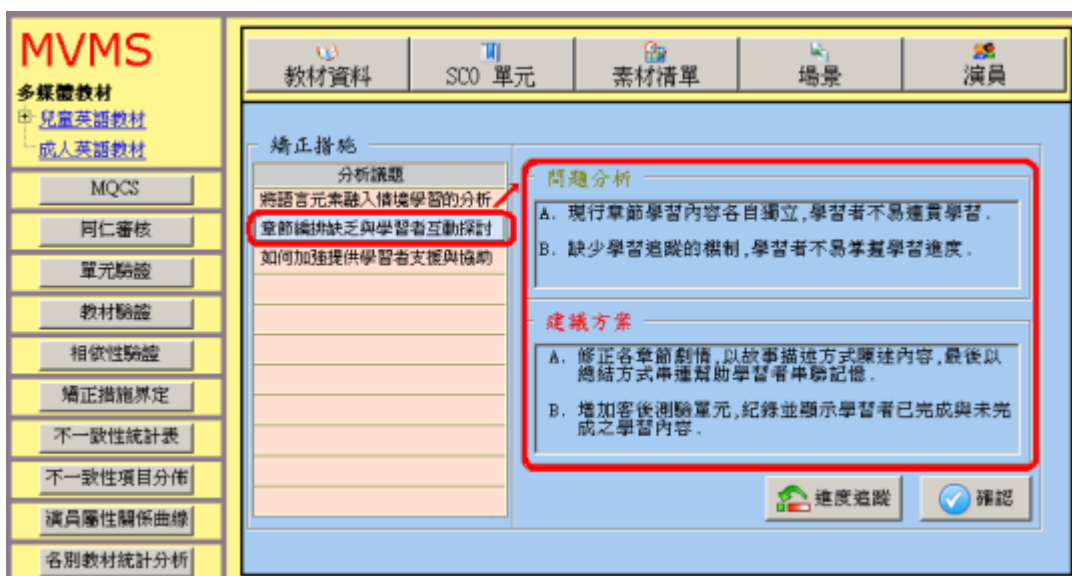


圖 46：矯正措施執行畫面

5.5 系統導入成效



MVMS 系統結合多媒體教材開發生命週期的每一個階段進行單元審核與驗證管理，確保多媒體教材從單一素材、SCO 架構到整合模組單元都符合教材規格從而達成產品與需求規格一致性的目標。經實際導入後其成效如下：

1. 確保實際工作教材產品與需求規格的一致性，提高多媒體教材內容品質。
2. 確保教材模組之間規格一致性，解決教材模組本身品質一致性問題。
3. 確保教材單元模組統整規格一致性，解決教材模組之間整合的問題。
4. 可以整合到多媒體教材的需求管理系統，解決產品需求變更時的一致性要求。
5. 結合統計學方法協助找出關鍵不一致性原因。
6. 透過輸出 XML 格式文件，提供不同多媒體教材製作系統之間一個一致性的內容格式轉換平台。

5.6 系統導入問題探討

在本小節中我們利用統計方法針對教材範例中驗證結果的不一致性項目相關數據作統計運算分析。在本節中的討論中，都是以下列六個驗證項目，演員屬性、演員退場方式、場景屬性、轉場特效、劇情順序、互動劇情等作為分析項目。

圖 47 的統計柱狀圖 (Bar Chart) 是針對教材 LEC001 的上述六個項目統計結果。由圖中可以知道演員屬性是教材 LEC001 的關鍵不一致性錯誤項目。

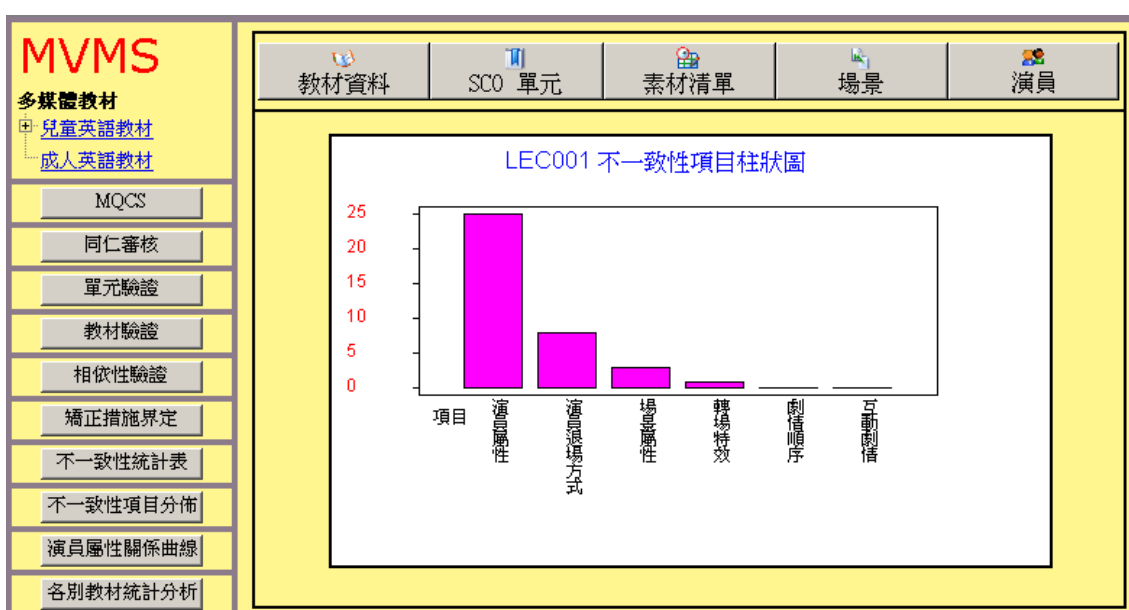


圖 47：教材 LEC001 錯誤項目統計柱狀圖 (Bar Chart)

圖 48 則是教材製作者使用本系統陸續再新增八個多媒體教材專案後，系統累計上述六個驗證項目數據，經彙整後統計的結果整理於表 12。



圖 48：多媒體教材範例

表 12：多媒體教材範例驗證數據表

教材主題	編號	演員數目	場景數目	劇情數目	演員屬性	演員退場方式	場景屬性	轉場特效	劇情順序	互動劇情
What is on the mat	LEC001	13	1	9	25	8	3	1	0	0
鐵道之旅	LEC002	25	14	13	30	5	21	0	2	0
北國探險	LEC003	16	5	15	24	7	12	0	1	3
原住民歷史簡介	LEC004	18	8	22	28	4	14	0	1	5
希臘風情	LEC005	20	17	15	27	1	11	2	3	4
中國文字演變	LEC006	24	12	18	29	1	8	4	3	8
小城故事 -- 淡水小鎮的故事	LEC007	13	8	12	22	2	11	0	2	5
認識聲音	LEC008	12	8	21	23	2	12	2	0	7
印地安傳說	LEC009	22	12	20	25	3	6	1	4	5

從表 12 中，同樣以演員屬性、演員退場方式、場景屬性、轉場特效、劇情順序、互動劇情等六個項目為母體樣本數，繪製成統計盒圖 (Box Plot) 來幫助觀察樣本數據群的分佈情形，如圖 49 所示。

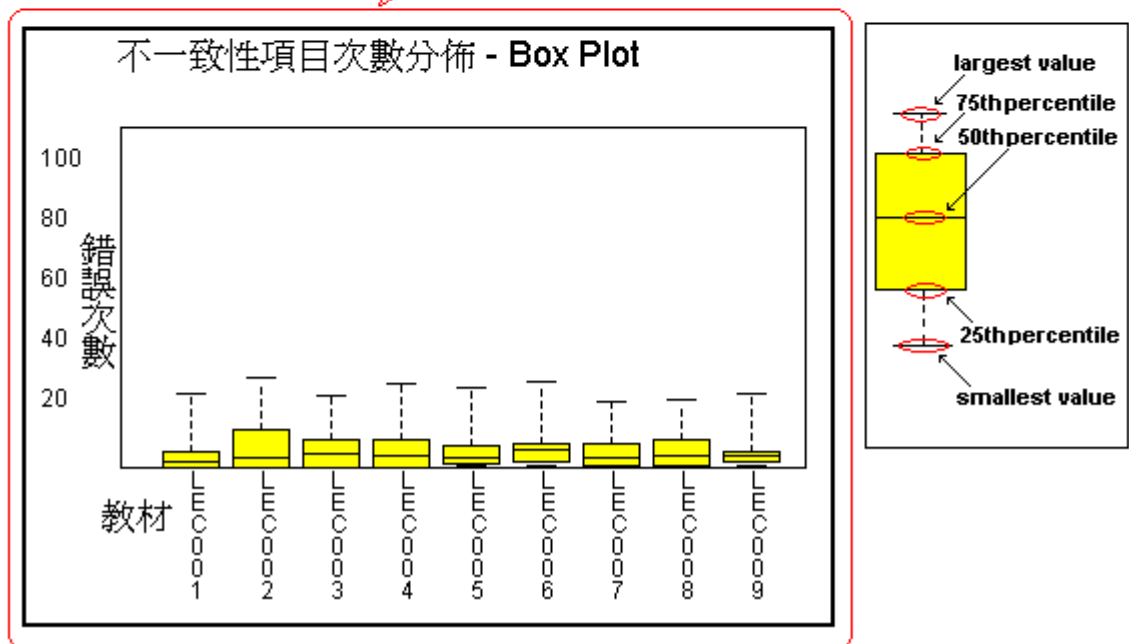
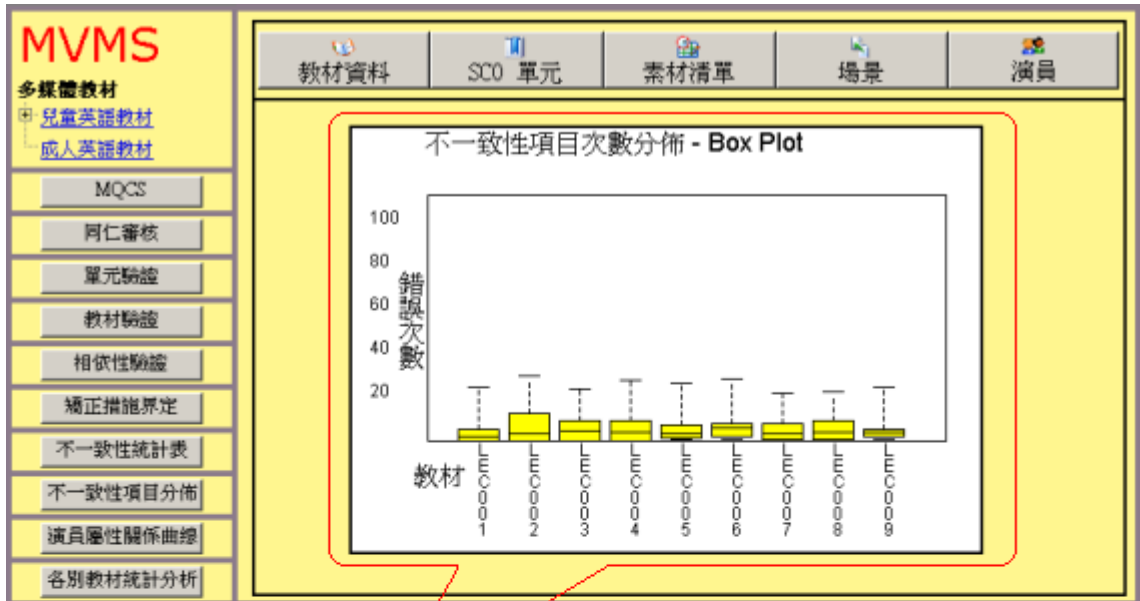


圖 49：多媒體教材範例不一致性項目盒圖

觀察圖 49 的九個範例的錯誤項目分佈，其中，最小值(smallest value)、第一四分位數(25th percentile)和第三四分位數(75th percentile)之間都很接近，唯有最大值(largest value)和其它數值有明顯的差距。這結果顯示出至少有一個項目產生不一致性的機率高出其它項目，因為其次數明顯高出許多。若再與表 12 的作對照可以得知此最大數值項目就是演員屬性項目，也就是少數關鍵的不一致性項目。在統計學上柏拉圖 (Pareto Chart) 是用來表示出關鍵少數項目的常用方法，圖 50 是教材 LEC001 的不一致性項目柏拉圖。

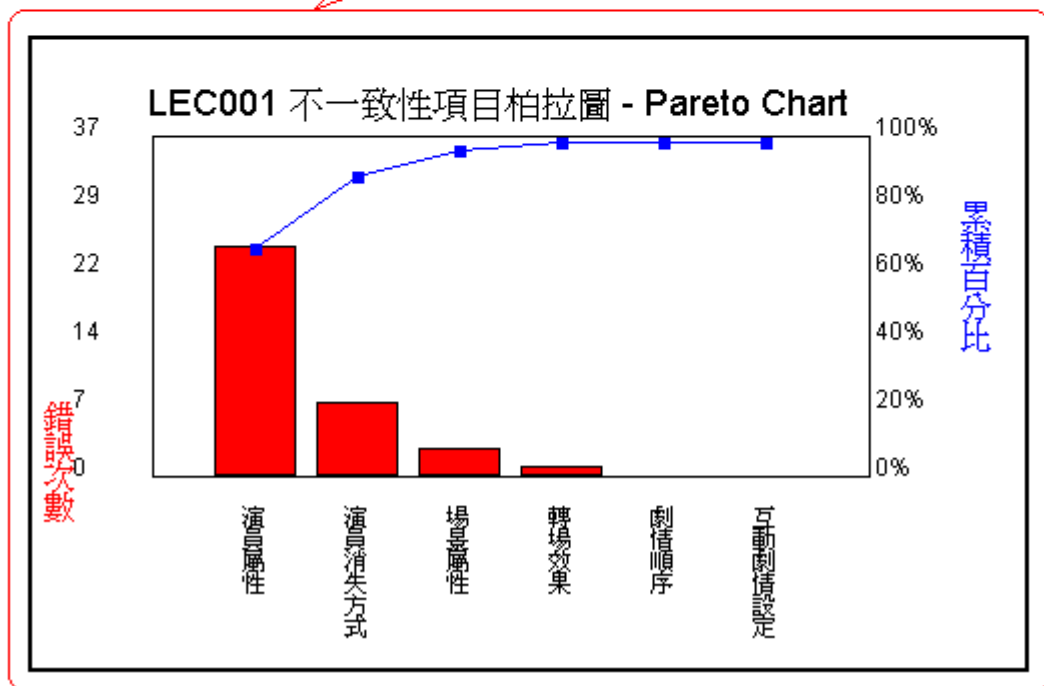
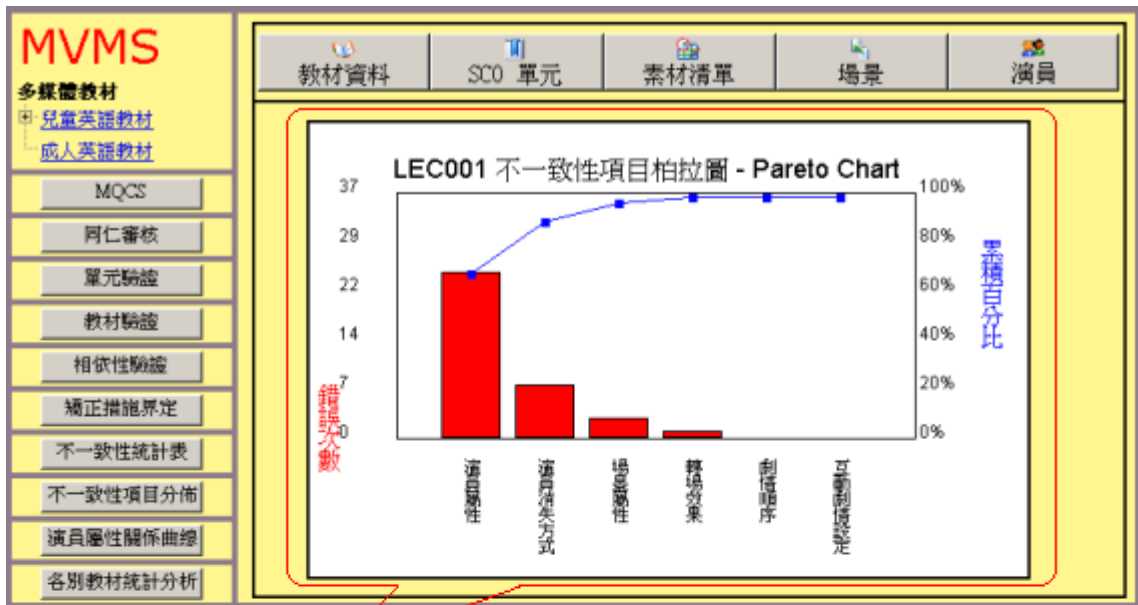


圖 50：教材 LEC001 不一致性項目柏拉圖

因為每一個演員都有演員屬性需求規格，因此當演員數量增加時，則演員屬性發生不一致性錯誤的機率也跟著提高，亦即演員數量和演員屬性欄位值有正相關特性。圖 51 所示即是以演員屬性錯誤次數和演員數量分別為縱、橫軸所繪製的關係曲線圖，其中，橫軸是由小到大遞增排列，從圖中可以發現曲線有向上遞增的趨勢。若將表 12 的每一個範例作上述統計，都會得到類似的結果。

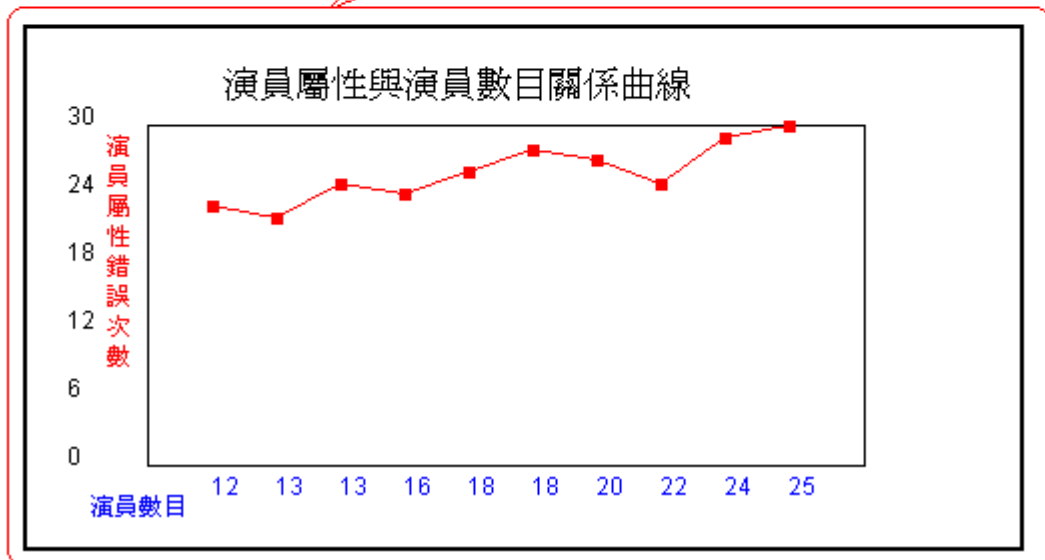
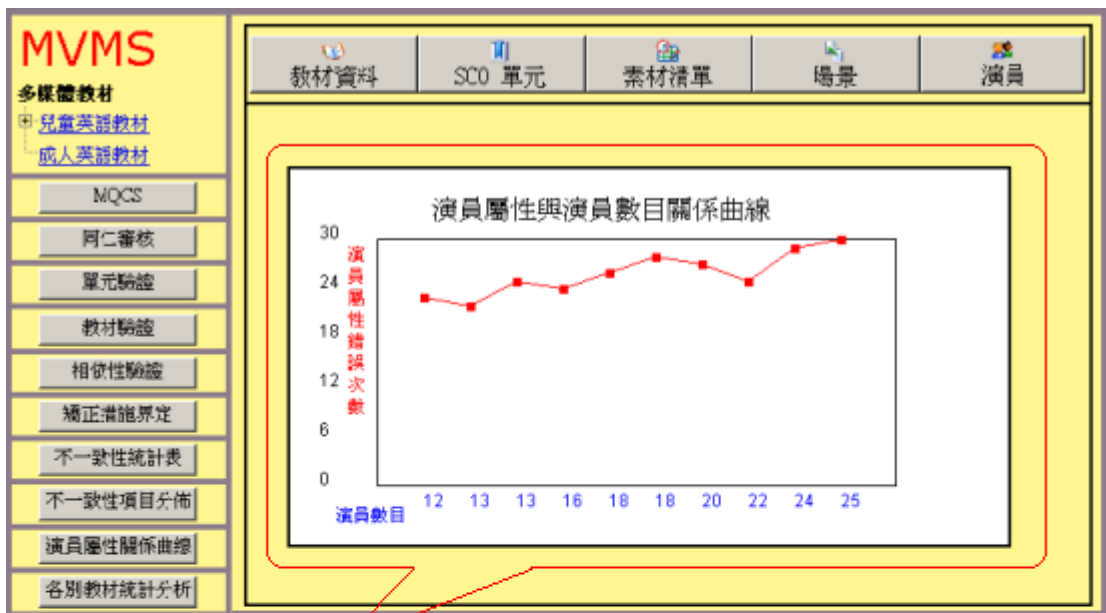


圖 51：演員屬性與演員數目關係曲線

圖 51 顯示的結果讓我們發現系統存在的一個明顯的問題，即是當規格制定越詳細，則發生不一致性錯誤的情況也跟著提高，因為系統中訂定的演員屬性規格遠多於其他單元，所以演員屬性發生不一致性錯誤的情形也多於其它項目。表 13 是參考智勝國際公司的多媒體教材開發工具編輯手軟體所制定的動畫演員規格列表[25]。

為了解決上述問題，系統作了下列三項修正：

1. 加入誤差容許機制

在不影響教材內容正確性的範圍內，提供誤差容許機制，降低錯誤發生情況。

2. 以預設選項取代絕對數值規格
 針對某些欄位規格並不影響內容的正確性，只會影響畫面美觀性，例如演員位置屬性，此種屬性可以使用將畫面切割成四個象限或九個方格的選項作法來取代用絕對座標表示，降低錯誤發生情況。
3. 區分必要項規格與非必要項規格
 必要項規格是會影響教材內容品質及正確性的項目(例如，演員名稱、屬性)，非必要項規格則相反(例如，轉換特效)。系統只在必要項規格不一致發生時判定為驗證錯誤。對於非必要項規格的不一致情況則以提供警示訊息方式因應。

經過上述改善機制，在不影響教材內容正確的範圍內，的確有效的降低不一致性錯誤，更提高了系統的實用性。

表 13：動畫演員屬性規格 (Source：編輯手提供)

動畫演員訊息	
欄位名稱	欄位說明
Type	動畫演員類型識別字
Name	演員名稱(唯一)，背景動畫演員(BGActor)，背景音樂演員(BGMActor)
Caption	演員說明(給教材編輯者)
Position	演員位置(X-axis, Y-axis, Depth)
Size	演員大小(Width, Height)
Alpha	演員透明度: 0 ~ 1
Rotate	演員旋轉角度, 0 ~ 360
Filter	濾鏡效果，無(0)、模糊(1)、凹凸(2)、陰影(3)、發光(4)
Visible	編輯模式下，設定演員隱藏(0)或顯示(1)
Lock	編輯模式下，設定演員解除鎖定(0)或鎖定(1)
Replaceable	演員可在樣板模式下置換內容(1)
FrameCount	動畫張數
BorderInfo	1. 顯示外框，0：不顯示，1：顯示 2. 外框顏色 3. 外框粗細
FillInfo	1. 顯示填色，0：不顯示，1：顯示 2. 填色顏色
ScaleInfo	1. 縮放倍率 2. 顯示範圍(Left) 3. 顯示範圍(Top) 4. 顯示範圍(Width) 5. 顯示範圍(Height)
ShowGUI	顯示面版
Skin	面版資源檔案
SegCount	動畫路徑數量
RecordingCaption	錄音檔名稱

Segment	<ol style="list-style-type: none"> 1. BeginPicIdx：開始的 frame index，通常都是-1 2. EndPicIdx：結束的 frame index，通常都是-1 3. PathLoopCount：動畫路徑播放的次數，-1代表無限次播放 4. SoundLoopCount：聲音播放的次數，-1代表無限次播放 5. pathPlayTime：路徑播放秒數 6. delayTimeToPlay：延遲幾秒開始播放，1000代表1秒 7. playWhenStart：播放參數(請見PlayWhenStart欄位說明) 8. alpha：結束的 alpha 數值 9. rotation：結束的旋轉數值 10. resourceFName：演員的資源檔案，動畫演員是圖片，文字演員是文字檔案 11. soundFName：播放聲音的檔案名稱 12. pathFinalSize：演員播放結束的大小 13. effect：動畫效果: 0:無, 1:由快變慢(easeoutexpo), 2:由慢變快(easeinexpo), 3:終點反彈(easeoutback), 4:起點反彈(easeinback) 14. path_mode：套用路徑模式: 0:使用者定義, 1:飛入(flyin), 2:飛出(flyout) 15. path_direction：路徑方向: 0:上(up), 1:下(bottom), 2:左(left), 3:右(right)
PlayWhenStart	<p>演員播放的模式，共十種，0：關閉，1：啟動</p> <p>PlayWhenSceneStart：均為0</p> <p>NoAnimateWhenSceneStart：演員開始演出前，停止動作。</p> <p>NoAnimateAfterPlay：演員結束演出後，停止動作。</p> <p>DisappearWhenSceneStart：演出前是否隱藏演員。</p> <p>DisappearAfterPlay：演出後是否隱藏演員。</p> <p>PlayFromOriginalPosition：是否以路徑起點為演出起始點，0：以路徑終點，1：以路徑起點。</p> <p>LoopPathFromOriginalPosition：是否原地連續演出，0：是，1：往前連續演出。</p> <p>NestedActorScenario：是否為劇情演員。</p> <p>NotWaitSoundOverWhilePlaying：是否不等聲音播完後結束，0：等聲音播完結束。</p> <p>NotWaitPathOverWhilePlaying：是否不等路徑演完後結束，0：等路徑演完結束。（若上兩個數值均為0，表示會等聲音及路徑演完後結束）</p>
RatioEquality	等比縮放，0：關閉，1：開啓
MirrorLR	左右翻轉，0：不翻轉，1：左右翻轉
MirrorUD	上下翻轉，0：不翻轉，1：左右翻轉
AutoDirectChange	路徑播放時，自動隨路徑翻轉

六 結論與未來研究方向

6.1 結論

本論文最主要的目的是提出以 CMMI 驗證流程領域為基礎的多媒體教材驗證方法，並實作出驗證管理系統，用來解決多媒體教材製作過程產生的不一致性問題。由於多媒體教材最終成品是以軟體作為呈現與儲存型態，因此本研究參考軟體工程方法論中的驗證與確認方法，結合 CMMI 驗證流程領域的特定方法與特定目標，設計出符合多媒體數位教材開發流程的驗證機制。希望藉此系統的導入能結合多媒體教材的需求管理系統，有效的控管及改善多媒體教材的開發程序，進而能開發出高品質的多媒體教材。彙整本研究的貢獻如下：

1 解決多媒體教材開發的一致性問題

針對多媒體教材開發流程的品質驗證管理問題而開發出一套驗證管理系統，能在多媒體教材製作過程中的各階段，進行階段驗證，及時發現問題進行改善。

2 導入以 CMMI 驗證流程領域為基礎的驗證方法達成一致性目標

以驗證準備、同仁審查、驗證執行、矯正措施界定四大功能模組的流程運作，確保多媒體教材產品與定義規格一致，同時整合了 CMMI 的驗證流程領域規範，確保多媒體教材產品的品質。

3 結合統計學方法協助分析關鍵不一致性項目

系統以資料庫中的記錄數據為統計樣本，輸出相關統計圖表，幫助系統設計者找出關鍵性不一致性項目加以改善，以獲得最大改善效益。

4 輸出標準 XML 格式檔案，整合多媒體教材製作工具軟體

本系統提供輸出標準格式腳本語言 XML 檔案，提供不同多媒體教材製作工具之間正確且高效率的一致性內容格式轉換平台。

6.2 未來研究方向

未來，本研究建議可以朝下列幾點進行更進一步的探討與研究。

- 1、由於多媒體教材專案中的需求文件及相關表格相當繁多，再加上專案發展過程中的設計變更，使得要維護專案需求規格文件相當不易且耗時。本研究建議可針對多媒體開發系統發展規格文件自動產生系統 (Document Generator)，有效率地產生、管理與維護系統規格文件。
- 2、由於資訊技術的發展快速，考量未來可能有更新的創作工具、開發平台以及交換機制，為了符合各種新形式的多媒體學習物件規範與使用者學習介面的需要，評估將 CMMI 其他的流程領域 (例如，建構管理、專案管理等)、規範與作業實施建議項目，導入多媒體教材開發流程的管理系統，來達到建立與維護工作產品建構之完整性。
- 3、多媒體設計工具軟體繁多，本研究只整合了智勝國際發行的編輯手開發工具軟體。未來若能整合更多其他多媒體設計工具軟體，則能大幅提高本研究所提出的品質管制手法的應用廣度。



參考文獻

- [1] Advanced Distributed Learning Department of Defense (DoD), [On-line]. Available: <http://www.adlnet.org>
- [2] IEEE Learning Technology Standards Committee (LTSC) IEEE P1484.12 Learning Objects Metadata Working Group, [On-line]. Available: <http://ltsc.ieee.org/wg12/>
- [3] ADL, “SCORM 2004 2nd Edition Overview”, [On-line]. Available: <http://www.adlnet.org/>
- [4] Advanced Distributed Learning (ADL) initiative, SCORM Specifications – The SCORM Content Aggregation Model Version 1.2, October 1, 2001
- [5] Advanced Distributed Learning (ADL) initiative, SCORM Specifications – SCORM Version 1.3 Application Profile Working Draft Version 1.0, March 26, 2003
- [6] ADL, “SCORM Content Aggregation Model (CAM) Version 1.3.1”, [On-line]. Available: <http://www.adlnet.org/>
- [7] Sharable Content Object Reference Model: The SCORM Book 1: The SCORM Overview, [On-line]. Available: <http://www.adlnet.org>
- [8] Sharable Content Object Reference Model: The SCORM Book 2: The SCORM Content Aggregation Model, [On-line]. Available: <http://www.adlnet.org>
- [9] Sharable Content Object Reference Model: The SCORM Book 3: The SCORM Run-Time Environment, [On-line]. Available: <http://www.adlnet.org>
- [10] ADL, “SCORM Run-Time Environment (RTE) Version 1.3.1”, [On-line]. Available: <http://www.adlnet.org/>
- [11] Deming, W., “Out of the Crisis”, MIT Center for Advantaged Engineering Study, 1986.
- [12] Jamieson A., “Introduction to Quality Control”, Reston Publishing 1982.
- [13] Schulmeyer, G. et al, “The Handbook of Software Quality Assurance”, 3rd Edition, Prentice-Hall, 1999.
- [14] Ian Sommerville, Software Engineering, 8th Edition, Addison-Wesley, 2007.
- [15] Pressman, R., Software Engineering: A Practitioner’s Approach, 6th Edition McGraw-hill, 2005.
- [16] MU/SEI, CMU/SEI-2006-TR-08, CMMI for Development(CMMI-DEV), V1.2, CMU/SEI, August 2006.
- [17] CMMI 產品團隊, 「適用於發展的能力成熟度整合模式 (CMMI-DEV) 1.2 版」, 財團法人資訊工業策進會, 民國九十七年。

- [18] 行政院主計處，政府機關資訊通報，第 212 期專題報導。
- [19] 經濟部工業局，提升資訊軟體品質計畫網站— CMMI 的效益，
<http://www.cmmi-taiwan.org.tw/cgi-bin/big5/cmmi/po00?q1=picdata&q22=6&q3=ea055>
- [20] Software Engineering Institute, Carnegie Mellon Goldenson, D.R., Gibson, D.L.,
“Demonstrating the Impact and Benefits of CMMI: An Update and Preliminary
Results”; CMU/SEI-2003-SR-009; 2003.
- [21] Geoff Draper, Rickhefner Ph.D., “Applying CMMI Generic Practices with Good
Judgment”, SEPG Conference Tutorial, March 2004.
- [22] 陳登吉，「國立交通大學資訊工程暨多媒體工程研究所多媒體編輯課程講
義」。
- [23] Thulasiraman, K.; Swamy, M. N. S. (1992), "5.7 Acyclic Directed Graphs",
Graphs: Theory and Algorithms, John Wiley and Son.
- [24] Cormen, Thomas H. Leiserson, Charles Eric. Rivest, Ronald L., Introduction to_
algorithms, Cambridge, Mass.: MIT Press, 2001.
- [25] 智勝國際，“編輯手2004”，<http://www.caidiy.com/>

