

國立交通大學

資訊學院 資訊學程

碩士論文

應用於多媒體教材製作之文件產生器的設計與實作

The Design and Implementation of a Document Generator
for the Multimedia Learning Content Creation



研究生：陳詩雯

指導教授：陳登吉 教授

中華民國九十九年五月

應用於多媒體教材製作之文件產生器的設計與實作
The Design and Implementation of a Document Generator
for the Multimedia Learning Content Creation

研究生：陳詩雯

Student : Shih-Wen Chen

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
A Thesis
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
May 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年五月

應用於多媒體教材製作之文件產生器的設計與實作

學生：陳詩雯

指導教授：陳登吉 博士

國立交通大學

資訊學院

資訊學程碩士班

摘要

隨著資訊科技與網路的發展，多媒體教材的應用與需求不斷地成長，如何製作一個較高品質的多媒體教材成為重要的議題。有別於探討多媒體教材之教學方法等教材內容本身的品質問題，本研究則是專注在管理多媒體教材製作過程中產生的品質問題，以軟體工程的方法，有效控管多媒體教材的製作過程，進而提升多媒體教材的品質。

在軟體工程領域裡，文件可以在軟體的開發及維護過程中，協助呈現軟體的工作產品，對於提升軟體的品質是有幫助。但是由於多媒體教材是由文字、圖片、聲音、影片、動畫等多媒體檔案所組成，其製作過程文件除了呈現一般軟體開發工具的文件產生功能所支援的文字與圖片的格式之外，還要能夠展示各種多媒體檔案的播放效果，讓專案成員可以瞭解整個多媒體教材的設計內容與開發過程。

因此本研究設計與實作一套可以支援多媒體檔案呈現的文件產生器，透過自動產生多媒體教材的製作過程文件及提供文件追蹤的功能，協助專案成員追蹤及確認每一個開發階段的正確性。並且實作 CMMI 之 PPQA 流程領域及發展多媒體教材製作過程之品質評估模型，藉由審核製作過程文件來評估各開發階段的設計成果，協助管理多媒體教材製作過程中的品質。最後，以一個教案作為實例，確認本研究的設計成果可以適用於多媒體教材的製作過程。

關鍵字：多媒體教材、軟體工程、文件產生器、品質管理

The Design and Implementation of a Document Generator for the Multimedia Learning Content Creation

Student : Shih-Wen Chen

Advisor : Dr. Deng-Jyi Chen

Degree Program of Computer Science
National Chiao Tung University

ABSTRACT

By development of technology information and internet, the application and demand of the multimedia learning contents (MLCs) are growing up. How to create MLCs with higher quality becomes an important issue. In this research, we focus on the study of managing the quality issues of the MLCs development process and use the methodology of software engineering to control the MLCs development process.

In the software engineering area, we can present the work products of software by documentation during the software development and maintenance process. MLCs consist of various media files, such as text, image, audio, video and animation, so the documents of the MLCs development process should support displaying all kinds of media files to assist the developers in understanding the design and development process of MLCs.

This research proposed a document generator for the MLCs creation, which can collect all kind of information on the MLCs development process and provide a mechanism for tracing work products on documents. We also implemented the PPQA process area of CMMI and established a quality evaluation model to manage the quality of the MLCs development process. Finally, we use a lecture to demonstrate that the proposed document generator and quality management method can be applied to the MLCs development process.

Keywords: Multimedia Learning Contents (MLCs), Software Engineering,
Document Generator, Quality Management

誌謝

衷心感謝陳登吉教授耐心的指導，並且在研究的過程中，時時給予建議，才能順利完成本論文。

同時也要特別感謝孔崇旭教授給予的指導與建議，讓本論文可以更加完善。

而在研究的過程中，則要感謝同窗好友仲智、錦成、信江的互相鼓勵與一同進行研究，透過彼此的分享與討論，使我可以更清楚相關的研究資訊。

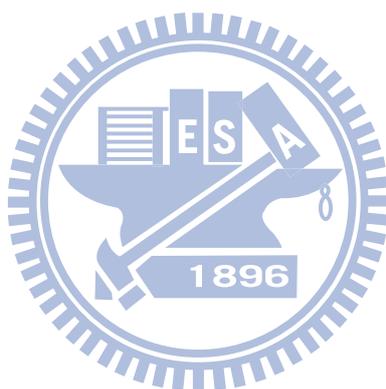
最後，要感謝我的家人在求學的這一段期間內給予的支持與鼓勵，特別要感謝我的父母親對我的照顧與付出，才能成就我的理想，感謝您們!!



目錄

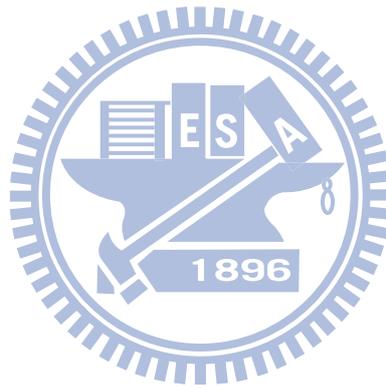
摘要	i
ABSTRACT	ii
誌謝	iii
目錄	iv
表目錄	vi
圖目錄	vii
一、 緒論	1
1.1 研究背景	1
1.2 研究動機與目的	2
1.3 研究方法	3
1.4 章節概要	4
二、 相關研究	5
2.1 軟體品質	5
2.1.1 系統文件	5
2.1.2 品質管理	5
2.2 CMMI	6
2.2.1 表述模式	7
2.2.2 流程領域	8
2.2.3 PPQA 流程領域	10
2.3 SCORM	12
三、 多媒體教材品質評估模型	14
3.1 製作過程文件	14
3.2 品質狀態圖	17
3.3 品質評估模型	26
四、 系統分析與設計	29
4.1 多媒體教材品質管制系統	29
4.2 需求分析	30
4.2.1 文件產生器	30
4.2.2 品質管理	31
4.2.3 使用案例圖	31
4.3 系統設計與實作	34
4.3.1 開發環境	34
4.3.2 系統架構	34
4.3.3 文件產生器	36
4.3.4 文件追蹤機制	38

4.3.5	品質管理	39
五、	應用實例	41
5.1	範例說明	41
5.2	文件管理	44
5.2.1	檔案播放方式	44
5.2.2	文件設定	45
5.2.3	文件關聯定義	48
5.3	教材管理	49
5.4	品質管理	54
六、	結論	57
6.1	總結	57
6.2	未來發展方向	58
	參考文獻	59
	附錄	61



表目錄

表 1	CMMI Process Areas.....	9
表 2	製作過程文件列表.....	16
表 3	「ABC in Phonics」之製作過程文件.....	43



圖目錄

圖 1	數位教材五大面向檢核圖	2
圖 2	The Three Critical Dimensions.....	6
圖 3	Continuous Representation.....	7
圖 4	Staged Representation	8
圖 5	CMMI Model Components	10
圖 6	PPQA Context Diagram	12
圖 7	Conceptual Illustration of a Content Aggregation.....	13
圖 8	多媒體教材開發生命週期	14
圖 9	設計元件關係圖	15
圖 10	製作過程文件關聯圖	17
圖 11	品質狀態圖	18
圖 12	品質狀態圖－課程架構文件	19
圖 13	品質狀態圖－SCO 描述文件	19
圖 14	品質狀態圖－分鏡設計文件	20
圖 15	品質狀態圖－腳本設計文件	21
圖 16	品質狀態圖－UI 說明文件	21
圖 17	品質狀態圖－素材清單（文字）	22
圖 18	品質狀態圖－素材清單（圖片）	23
圖 19	品質狀態圖－素材清單（聲音）	23
圖 20	品質狀態圖－素材清單（影片）	24
圖 21	品質狀態圖－素材清單（動畫）	25
圖 22	品質狀態圖－教材成果檔	25
圖 23	專案模型圖	26
圖 24	多媒體教材品質管制系統	29
圖 25	使用案例圖	32
圖 26	系統環境圖	34
圖 27	系統架構圖	35
圖 28	一般的文件產生方式	36
圖 29	提議的文件產生方式	36
圖 30	文件格式設定方式	37
圖 31	Object 型態資料處理方式	38
圖 32	XML 標籤資訊的記錄方式	38
圖 33	文件追蹤方式	39
圖 34	品質問題的記錄方式	40
圖 35	品質審核方式	40

圖 36	「ABC in Phonics」	41
圖 37	“What is on the Mat?” 課程內容	42
圖 38	新增檔案播放方式	44
圖 39	列出所有檔案播放方式	44
圖 40	新增文件	45
圖 41	列出文件	45
圖 42	編輯文件內容與格式	46
圖 43	加入 Complex element	46
圖 44	加入 Simple element	47
圖 45	設定文件格式	47
圖 46	選擇相互關聯的文件	48
圖 47	設定文件關聯	48
圖 48	新增教材資訊	49
圖 49	上傳文件資料與多媒體檔案	50
圖 50	預覽 HTML 文件	50
圖 51	預覽 PDF 文件	51
圖 52	選擇要瀏覽及追蹤的文件	51
圖 53	瀏覽及追蹤文件	52
圖 54	下載文件	52
圖 55	瀏覽整份可列印文件	53
圖 56	新增品質問題	54
圖 57	列出品質問題	54
圖 58	提出矯正措施	55
圖 59	檢視品質報告（第一次審核結果）	55
圖 60	檢視品質問題清單（第一次審核結果）	56
圖 61	檢視品質報告（第二次審核結果）	56



一、緒論

1.1 研究背景

隨著資訊科技與網路的發展，多媒體教材（Multimedia Learning Contents, MLCs）的應用與需求不斷地成長，如何製作一個較高品質的多媒體教材成為重要的議題。國際上已有許多組織致力於推廣數位教材相關的標準，其中以美國國防部 ADL 先導計畫（Advanced Distributed Learning Initiative）制定的 SCORM（Sharable Content Object Reference Model）[1] 最受重視並引為依據，但 SCORM 標準中雖然建立了教材再用與共享的機制，卻無法確保多媒體教材的品質。

由於多媒體教材兼具「教學」與「軟體」兩個專業領域的特質，所以在多媒體教材的開發上，除了滿足數位化教材製作的標準之外，我們希望可以運用軟體工程的方法，來協助有效控管多媒體教材的品質問題。而多媒體教材的品質問題可以分為二大面向來探討[2]：

(1) 教材內容本身的品質

教材內容本身的品質取決於教材製作者本身的創意與巧思，以及教材製作者所設計的教材內容的呈現方式，例如使用具有啟發性的教案或運用特定的教學方法來達成教學的目標。

(2) 教材製作過程中的品質

教材製作過程中的品質則是需要透過控管整個多媒體教材的製作過程，讓一份好的教材內容在製作成多媒體教材的過程中，可以減少製作上的錯誤，使得最後完成的多媒體教材可以符合預期的品質標準。

所以教材內容本身的品質傾向由教材設計者所決定，因此我們的研究則是專注在管理多媒體教材製作過程中可能產生的品質問題，藉由有效控管與持續改善多媒體教材的製作過程，提升多媒體教材的品質。

目前國內在數位教材品質的認定上，主要是依據教材內容（Content）、導引與追蹤（Navigation & Tracking）、教學設計（Instructional design）、教學媒體（Instructional media）與創意（Creativity）五大面向對教材成品進行檢核[3]，如圖 1 所示。對於教材的製作過程，則沒有相關的檢核標準與認證制度。



圖 1 數位教材五大面向檢核圖
資料來源：數位學習品質認證中心[3]

而在軟體工程領域裡，CMMI (Capability Maturity Model Integration) 是一個國際性的軟體品質規範，它主要是針對軟體的開發程序，匯集了最佳的實踐方式，讓組織透過流程改善提供更好的產品與服務[4]。CMMI 制定的二十二個流程領域 (Process Area, PA) 裡，PPQA (Process and Product Quality Assurance) 流程領域的目的在於提供專案成員與管理人員客觀地洞悉流程與相關的工作產品。藉由實作 PPQA 流程領域，客觀地評估軟體的開發流程與工作產品，以及追蹤、矯正所發現的品質問題，則可以協助管理軟體的開發流程，建立品質確保的機制。

軟體開發文件是呈現軟體與軟體開發流程唯一可見的方式[5]，利用文件記錄與呈現軟體開發過程中的資訊，可以協助專案成員瞭解整個軟體的設計內容與開發過程，並作為專案成員彼此之間的溝通工具，對於軟體品質亦是關鍵的構成要素。一般軟體開發專案可以依照開發階段交付文件等工作產品作為專案進度的檢核點，而軟體開發文件亦可用以檢核每一個開發階段的設計成果是否符合需求。

1.2 研究動機與目的

雖然維護軟體開發文件對於提升軟體品質是有幫助的，但配合系統變更即時更新文件則需要投入較多的時間與成本。因此許多軟體開發工具均有提供文件產生的功能，以協助軟體開發人員隨時依照所設計的內容重新產生文件。

多媒體教材的開發流程同樣可以用文件的方式呈現每一個開發階段的設計成果，但是由於多媒體教材是由文字、圖片、聲音、影片、動畫等多媒體檔案所組成[6]，其製作過程文件所呈現的資訊，除了文字與圖片的格式之外，還要可以展示各種多媒體檔案的

播放效果，因此有別於一般軟體開發工具所支援的文件產生功能，我們需要一個可以支援多媒體檔案格式的文件產生器，以協助產生多媒體教材的製作過程文件。

而且多媒體教材製作過程中的每一個開發階段都是環環相扣的，各階段的工作產品也是彼此相關的，所以在任一開發階段發現品質問題時，專案成員需要逐一確認每一個階段的正確性，雖然製作過程文件可以透明化多媒體教材的開發資訊，以人工的方式在每一份獨立的文件中進行追蹤，仍顯得耗時費力，因此文件之間若能提供自動追蹤的機制，則可以協助專案成員更有效率地確認每一個開發階段的正確性。

另外，在品質管理的基本假設中，開發程序的品質會直接影響最後交付產品的品質[5]。而多媒體教材每一個開發階段的品質問題也都會影響到下一個開發程序，及時發現多媒體教材製作過程中的品質問題，並加以改善，可以減少品質問題所產生的連環效應，但一般多媒體教材的製作過程仍缺乏品質管理的機制，使得多媒體教材製作過程中的品質問題只能仰賴專案成員的開發經驗個別進行評估與改善。

為了有效控管多媒體教材製作過程的品質，我們需要一個品質評估的模式可以用來客觀地呈現多媒體教材的品質狀態，但目前數位教材的品質認證制度只針對教材成品制定檢核標準，而相關的品質評估工具中，其檢核項目亦是以教材成品作為品質評估的標的[7]，尚無適用於整個多媒體教材製作過程的品質評估方法。

因此本研究的目的則是設計與實作一套可以應用在多媒體教材製作過程的文件產生器，藉由自動產生多媒體教材的製作過程文件與提供文件追蹤的功能，協助專案成員追蹤及確認每一個開發階段的正確性，另外，並實作 CMMI 之 PPQA 流程領域，以及發展多媒體教材製作過程之品質評估模型，以建立品質確保的機制，透過有效控管多媒體教材製作過程中的品質問題，讓多媒體教材能有一定的品質。

1.3 研究方法

首先我們探討了多媒體教材的品質議題，並針對管理多媒體教材製作過程的問題，提出解決的方法。接下來我們將依照下列步驟進行研究：

1. 探討軟體品質的相關議題。
2. 瞭解 CMMI 在提升軟體品質方面的觀念與方式，及實作 PPQA 流程領域應注意的細節。
3. 依照多媒體教材的開發生命週期，整理每一個開發階段所產生的資訊，及定義製作過程文件。
4. 針對多媒體教材製作過程中可能產生的品質問題，發展多媒體教材的品質狀態圖。

5. 設計多媒體教材製作過程之品質評估模型。
6. 進行系統的分析、設計與實作。
7. 以一個教案作為實例，實際將本研究的設計成果應用於多媒體教材的製作過程，並評估其適用性。

1.4 章節概要

本論文共分為六個章節，各章節之內容依序摘要如下：

第一章，說明多媒體教材製作過程的品質議題，並提出本論文的研究動機、目的與研究方法。

第二章，探討軟體品質的相關文獻與 CMMI 規範。

第三章，介紹多媒體教材的開發生命週期，及依照多媒體教材的開發流程定義的製作過程文件、品質狀態圖與品質評估模型。

第四章，描述系統需求與設計架構，並詳細說明主要功能模組的實作方式。

第五章，透過實例展示本研究的設計成果。

第六章，提出本論文的結論與未來可以繼續發展的方向。



二、相關研究

因多媒體教材兼具「教學」與「軟體」兩個專業領域的特質，為了提升多媒體教材品質，不僅在教學方面需要滿足數位化教材的製作標準，針對其軟體的特性，亦要符合軟體開發流程的品質規範。所以本章節將探討軟體品質的相關文獻，以及在軟體工程領域裡，透過流程改善提升軟體品質的 CMMI 規範。

2.1 軟體品質

如何提升軟體品質是軟體工程領域的重要議題之一，在製造業裡，品質代表的是生產出來的產品是否有符合規格，品質就是符合需求[8]，同樣地，一個高品質的軟體就是可以需要滿足客戶的需求。軟體的品質因子包含正確性 (Correctness)、效率 (Efficiency)、可擴充性 (Expandability)、彈性 (Flexibility)、完整性 (Integrity)、互通性 (Interoperability)、可維護性 (Maintainability)、可管理性 (Manageability)、可移植性 (Portability)、可靠性 (Reliability)、可重用性 (Reusability)、安全性 (Safety)、耐受性 (Survivability)、可用性 (Usability) 與可驗證性 (Verifiability) [9]。

2.1.1 系統文件

系統文件是瞭解軟體的重要來源[10]，藉由文件彙整軟體開發過程中的資訊，可以協助專案成員瞭解整個軟體的設計內容與開發過程，並作為專案成員彼此之間的溝通工具。以可讀且具有結構性的文件來呈現軟體的工作產品是相當重要的，好的文件是一個有一定品質的軟體其開發及維護過程所不可或缺的[11]，它可以提升軟體開發生命週期的生產力，並且降低後續軟體變更與驗證作業的維護成本，是軟體品質的關鍵構成要素[12]。

而軟體工作產品之間的追蹤機制 (traceability) 則廣泛被認為是一個有效管理軟體開發與維護過程的重要因子，追蹤機制對於幫助程式理解、系統維護、影響分析，以及重用既有軟體亦是十分重要的[13]。在軟體的工作產品之間，一組精確的追蹤關聯 (relation) 可以支援軟體的發展，協助專案成員進行許多開發的作業，包含確認需求的實作、分析需求變更的影響，取得開發原理與設計決策，以及支援變更後的迴歸測試[14]。

2.1.2 品質管理

品質管理提供軟體開發過程一個獨立的檢核機制，品質管理的程序則是透過檢視專案交付的產品，來確定有符合組織的標準與目標，而在品質管理的基本假設中，開發程序的品質會直接影響最後交付產品的品質[5]。

ISO 9000 標準是可以用在發展品質管理系統的一套國際性標準,ISO 9001 則是這些標準中最普遍且適用於關心設計、開發與維護產品之品質程序的組織,雖然 ISO 9001 標準不是特別用於軟體的開發流程,但卻制定了可以應用在軟體的一般原則,而其程序定義則應包含所需文件的描述,以展示在產品的開發過程中這些被訂定的程序都有被遵循。ISO 9000 標準只是簡單地認為藉由公司使用程序的定義及運用相關文件作為管理程序,可以明確地證明有依照程序進行作業,但是卻沒有保證所訂定的程序是否是最適合的,或是可以讓產品有一定的品質[5]。

CMMI 規範用在提升軟體品質的觀念和 ISO 9000 標準是一致的。CMMI 主要是針對軟體的開發程序,匯集了最佳的實踐方式,讓組織透過流程改善提供更好的產品與服務,其制定的二十二個流程領域敘述在開發軟體時應該要做什麼,而組織的任務則是去記述如何做到這些事情。不過 CMMI 並不要求組織撰寫許多的文件,而是充分運用軟體開發過程中的資訊產生需要的文件[15][16]。

2.2 CMMI

CMMI 是由卡內基美隆大學 (Carnegie Mellon University, CMU) 的軟體工程學院 (Software Engineering Institute, SEI) 所發展及對外授權,其主要目的在於協助組織持續改善其開發及維護流程,以提供更好的產品與服務。

SEI 在幫助組織開發及維護高品質產品與服務的研究中,發現組織可以專注在人員 (people)、程序與方法 (procedures and methods),以及工具與設備 (tools and equipment) 這三個維度來提升營運,而流程 (processes) 則是支持著這幾個維度,讓組織可以調整營運方式、擴大規模、整合知識、協調資源及檢視經營趨勢,進而幫助組織達到營運目標[4],如圖 2 所示。

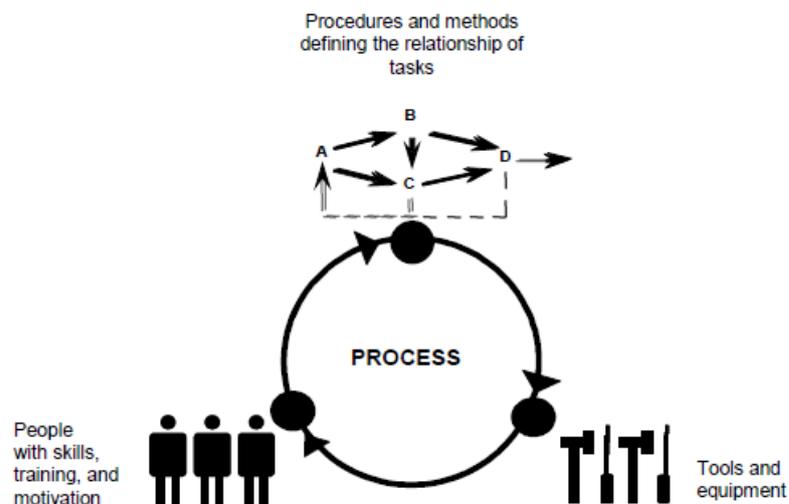


圖 2 The Three Critical Dimensions

資料來源：CMMI Product Team, 2006 [4]

而基本上，整合性的流程改善可以為組織降低成本、釐清營運目標、簡化重要的工作流程及適應環境變化[17]。由於 SEI 也採納了流程管理的假設，即「系統或產品的品質高度地受到其開發及維護流程的品質所影響」，因此 CMMs (Capability Maturity Models) 則專注於改善組織的流程。

2.2.1 表述模式

CMMI 匯集了軟體開發程序的最佳實踐方式，制定出二十二個流程領域，組織可以選擇使用連續式表述模式 (Continuous Representation) 或階段式表述模式 (Staged Representation) 來進行流程改善與評價。

當使用 CMMI 進行流程改善時，連續式表述模式提供了最大的實作彈性，組織可以選擇改善單一流程相關的問題，或是針對組織的商業目標從事數個領域，而每個流程領域也可以選擇不同的能力度等級 (Capability Level) 作為流程改善的目標，如圖 3 所示。

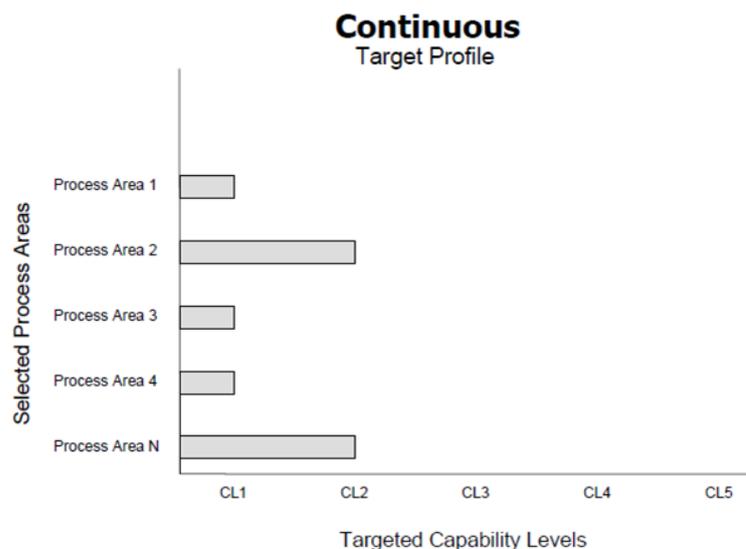


圖 3 Continuous Representation

資料來源：CMMI Product Team, 2006 [4]

階段式表述模式則提供了一個有系統與結構的方式依階段達到以模式為基礎的流程改善，它將流程領域依照成熟度等級 (Maturity Level) 進行分組，如圖 4 所示，讓組織可以由初始化階段至最佳化階段，有次序地實施流程領域以完成流程改善。

Staged Selected Maturity Level

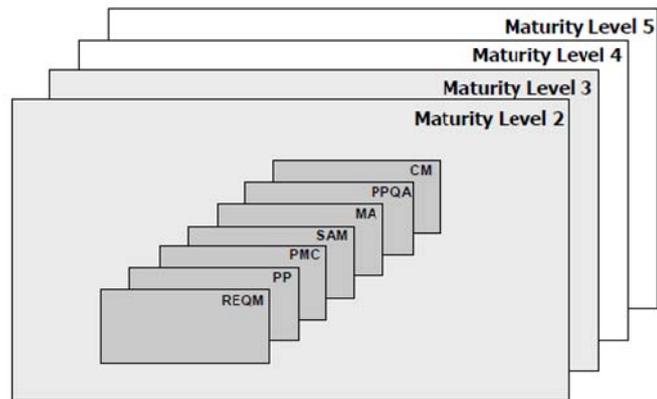


圖 4 Staged Representation

資料來源：CMMI Product Team, 2006 [4]

由於連續式表述模式可以讓組織依照其商業目標選擇適合的流程領域來實施，因此若是非常清楚那些流程是組織中需要改善的，並且已經瞭解各個流程領域之間的相依關係，則使用連續式表述模式是一個不錯的選擇[4]。

2.2.2 流程領域

流程領域共分為流程管理 (Process Management)、專案管理 (Project Management)、工程 (Engineering) 及支援 (Support) 四個類別：

(1) 流程管理

包含跨專案的活動，即定義、計畫、部署、實施、監視、控制、評價、量測及改善流程。基本的流程管理流程領域提供組織有能力記錄及分享最佳的實踐方式、組織性的流程資產，以及跨組織的學習。進階的流程管理流程領域則提供組織改善的能力，以達成品質與流程效能的量化目標。

(2) 專案管理

包含專案管理的活動，即計畫、監視及控制專案。基本的專案管理流程領域描述建立與維護專案計畫、建立與維護承諾、監視專案進度、採取矯正措施，以及管理供應商協議等活動。進階的專案管理流程領域則描述依組織標準流程建立定義的程序、依組織工作環境標準建立專案工作環境、合作及協調相關成員、管理風險、組成及支援整合的團隊以實施專案，以及量化管理專案定義的程序等活動。

(3) 工程

包含工程領域的開發與維護活動。工程流程領域將工程規範相關的流程整合成單一的產品開發流程，以支援產品導向的流程改善策略。

(4) 支援

包含支援產品開發與維護的活動。基本的支援流程領域描述運用在所有流程領域的主要支援功能。進階的流程領域則提供專案與組織改善的支援能力。

各流程領域所對應的類別及成熟度等級如表 1 所示。藉由瞭解流程領域之間的互動關係，以及區分基本的或進階的流程領域，可以協助組織以更有效的方式實作 CMMI。

表 1 CMMI Process Areas

類別	型態	流程領域	縮寫	成熟度等級
流程管理	基本	Organizational Process Focus	OPF	3
		Organizational Process Definition +IPPD	OPD+IPPD	3
		Organizational Training	OT	3
	進階	Organizational Process Performance	OPP	4
		Organizational Innovation and Deployment	OID	5
專案管理	基本	Project Planning	PP	2
		Project Monitoring and Control	PMC	2
		Supplier Agreement Management	SAM	2
	進階	Integrated Project Management +IPPD	IPM+IPPD	3
		Risk Management	RSKM	3
		Quantitative Project Management	QPM	4
工程	—	Requirements Development	RD	3
		Requirements Management	REQM	2
		Technical Solution	TS	3
		Product Integration	PI	3
		Verification	VER	3
		Validation	VAL	3
支援	基本	Configuration Management	CM	2
		Process and Product Quality Assurance	PPQA	2
		Measurement and Analysis	MA	2
	進階	Decision Analysis and Resolution	DAR	3
		Causal Analysis and Resolution	CAR	5

資料來源：整理自 CMMI for Development, Version 1.2 [4]

流程領域的組成元件分為必要的、期望的及有益的三種，如圖 5 所示。必要的組成元件是組織實施流程領域所必須達成的，包含每一個流程領域各自定義的特定目標 (Specific Goal, SG)，以及對應等級的一般目標 (General Goal, GG)。期望的組成元件則是為達成必要的組成元件所定義的，包含特定執行方法 (Specific Practices, SP) 與一般執行方法 (General Practice, GP)。有益的組成元件可以幫助組織思考如何達到必要的與期望的組成元件，其中典型工作產品 (Typical Work Product) 則是每一個特定執行方法可以參考的產出實例。

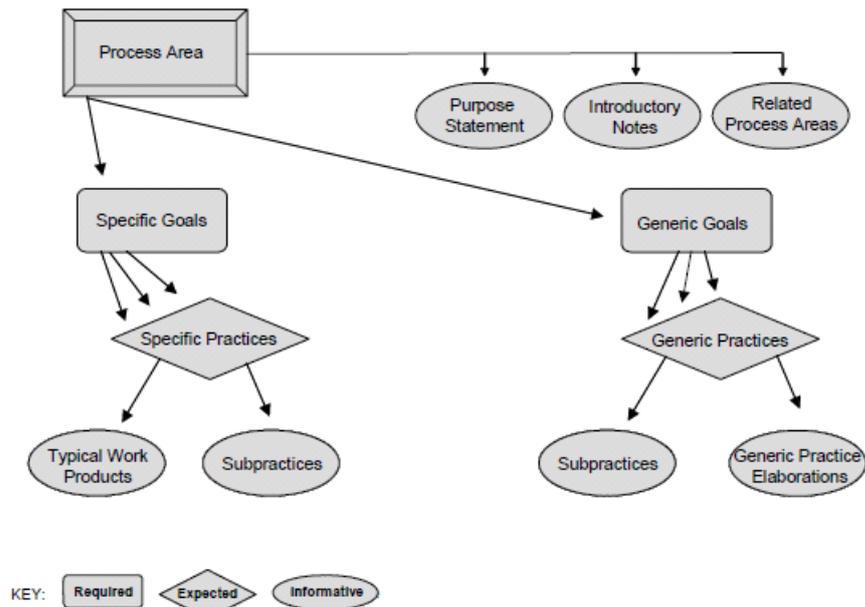


圖 5 CMMI Model Components

資料來源：CMMI Product Team, 2006 [4]

CMMI 的每一個流程領域都有定義特定目標與特定執行方法，由於組織在實作流程領域時，需要達到流程領域的特定目標及對應等級的一般目標，而藉由完成流程領域的特定執行方法所描述的活動，則預期可以達成其特定目標。

2.2.3 PPQA 流程領域

PPQA (Process and Product Quality Assurance) 流程領域的目的在於提供專案成員與管理人員客觀地洞悉流程與相關的工作產品。PPQA 流程領域可以確保組織的政策、業務與流程有被遵循[18]，其特定目標與特定執行方法如下：

- SG 1 客觀地評估流程與工作產品
(Objectively Evaluate Processes and Work Products)
 - SP 1.1 客觀地評估流程
(Objectively Evaluate Processes)
 - SP 1.2 客觀地評估工作產品與服務
(Objectively Evaluate Work Products and Services)

- SG 2 提供客觀的洞察力
(Provide Objective Insight)
 - SP 2.1 溝通及確保不符問題的解決方法
(Communicate and Ensure Resolution of Noncompliance Issues)
 - SP 2.2 建立記錄
(Establish Records)

第一個特定目標及其特定執行方法強調品質評估時的「客觀性」對於專案的成功是相當關鍵的，組織應先定義品質問題的呈報層級與如何確保客觀性。在執行時則藉由指定的品質衡量標準（criteria）來評估流程與工作產品，並鑑別每一個不符的問題與可供改善流程的經驗。

第二個特定目標及其特定執行方法則是需要客觀地追蹤及溝通不符的問題，確保每個問題都有提出解決方法，並透過分析不符的問題及記錄品質確保的活動，適時地提出品質評估結果與品質趨勢報告。

而 PPQA 流程領域的環境圖（Context Diagram）則如圖 6 所示。在環境圖中可以更清楚地看到品質評估時是透過審核工作產品來確認及呈報不符的問題，專案成員們則針對每個不符的問題進行溝通及提出解決方法，而整個品質確保的活動將會留下記錄。

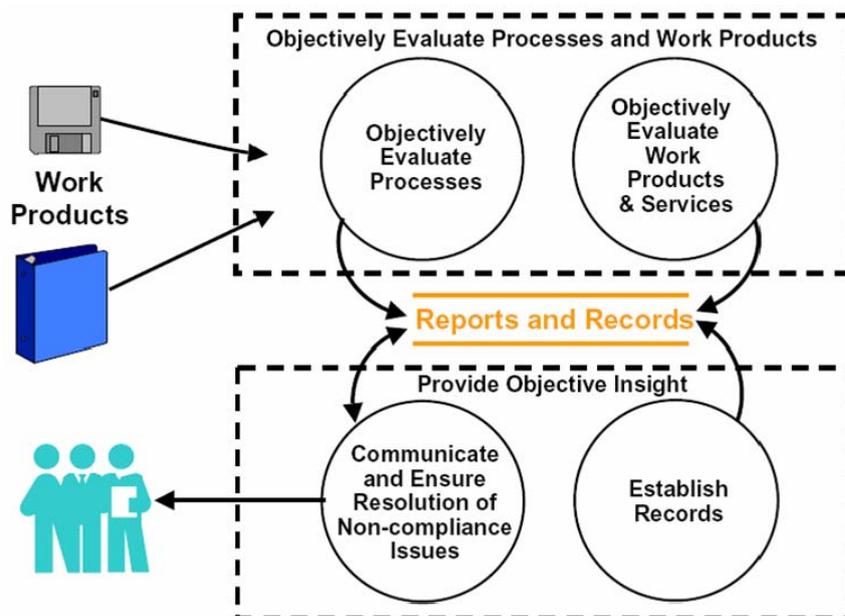


圖 6 PPQA Context Diagram

資料來源：Mike Phillips, 2004 [19]

2.3 SCORM

SCORM 是由美國國防部 ADL 先導計畫所制定的，目的在於建立教材再用與共享的機制。符合 SCORM 標準的學習內容具有可重用 (Reusable)、易取得 (Accessible)、可耐用 (Durable)、可互用 (Interoperable)、可適用 (Adaptable) 與經濟性 (Affordable) 的特性[6]。

ADL 制定的 SCORM 標準分為 CAM (Content Aggregation Model)、RTE (Run-Time Environment) 與 SN (Sequencing and Navigation) 規範[1]。其中，CAM 描述了 SCORM 內容模型的組成元件，由小而大包含素材 (Asset)、SCO (Sharable Content Object)、活動 (Activities)、內容組織 (Content Organization) 與內容整合 (Content Aggregation)，如圖 7 所示。

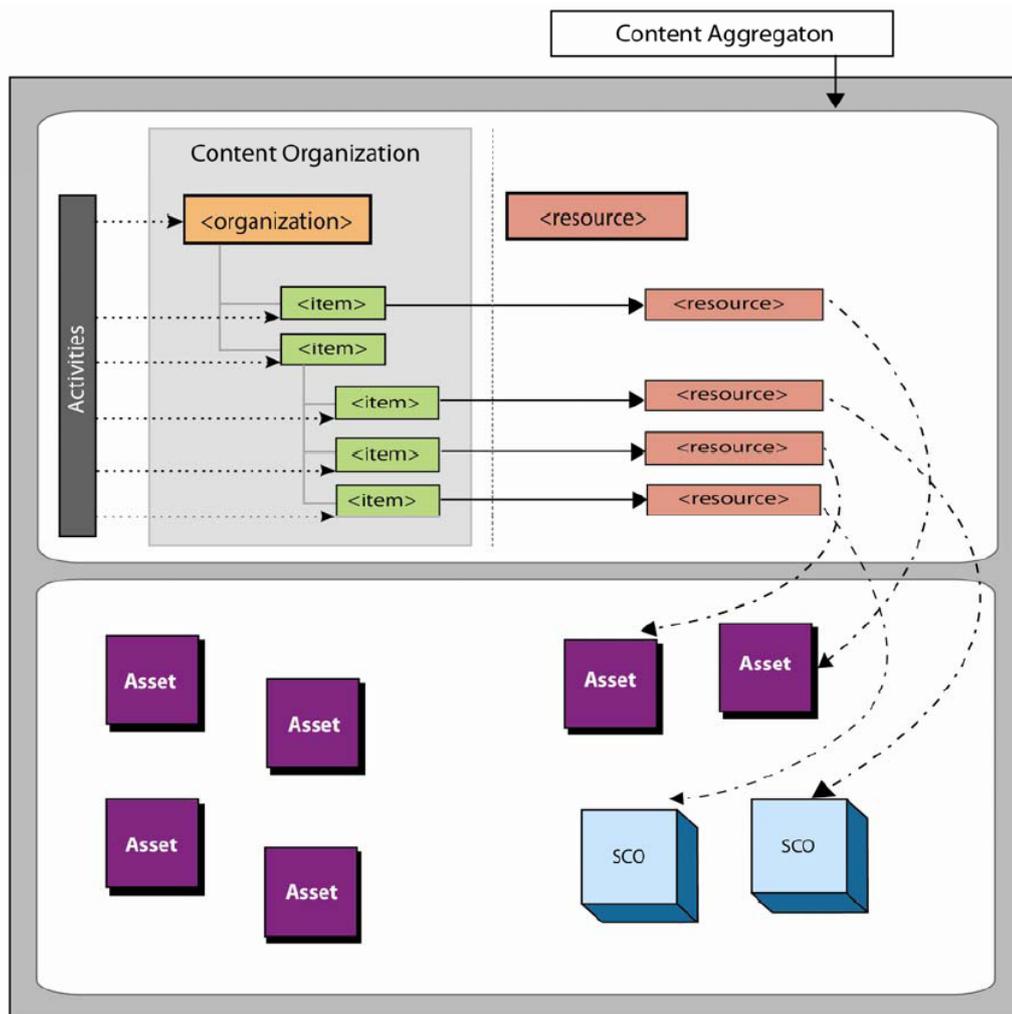


圖 7 Conceptual Illustration of a Content Aggregation

資料來源：ADL, 2009 [20]

而在此結構中，底層的學習資源（learning resource）是由素材與 SCO 所組成。素材是學習資源最基本的組成元件，它是以電子化呈現的媒體（media），例如文字、圖片與聲音等。SCO 則由一個以上的素材所組成，代表著可以發行的單一學習資源，並可使用 RTE 與 LMS（Learning Management System）進行溝通。

三、多媒體教材品質評估模型

在本章節中將介紹多媒體教材的開發生命週期，並依照多媒體教材的開發流程定義製作過程文件、品質狀態圖與品質評估模型。

3.1 製作過程文件

軟體工程領域裡，軟體開發程序（Software Process）一般包含以下幾個主要的活動[5]：

- (1) 軟體規格（Software Specification）
- (2) 軟體設計與實作（Software Design and Implementation）
- (3) 軟體確認（Software Validation）
- (4) 軟體發展（Software Evolution）

而如同軟體的開發程序，多媒體教材的開發生命週期（MLCs Process Life Cycle, MPLC）也可分為課程導入期、課程規劃期、課程製作期、測試結案，以及課程維護期幾個階段[6]，如圖 8 所示：

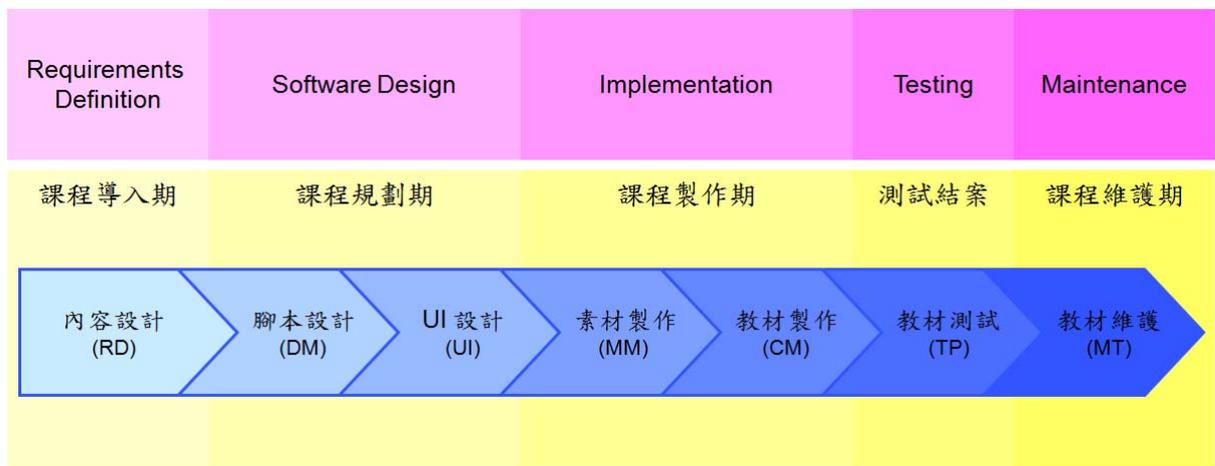


圖 8 多媒體教材開發生命週期

在多媒體教材的製作過程中，專案成員則依照程序逐步完成每一個開發階段的設計內容：

- (1) 課程導入期／教學內容設計（RD）

根據教學目的、教學對象與教學環境等需求，使用原創設計或是現有的學習教案，作為多媒體教材的原始教學內容。

(2) 課程規劃期／單元腳本設計 (DM)

設計多媒體教材的分鏡腳本，說明多媒體教材的內容所要表達的意境或觀念，並描述開場、退場，以及每一個演員的互動劇情。

(3) 課程規劃期／場景 UI 設計 (UI)

設計多媒體教材的使用者介面，包含每一個分鏡場景的背景圖片與音樂，並安排每一個演員的演出位置，以及每一幕的切換與瀏覽的方式。

(4) 課程製作期／素材製作 (MM)

依據分鏡腳本設計與製作需要使用的多媒體素材，包含文字、圖片、聲音、影片及動畫等各種媒體格式。

(5) 課程製作期／教材製作 (CM)

使用多媒體編輯軟體或程式開發工具，將多媒體素材依照分鏡腳本中的劇情安排與 UI 的空間規劃，實際製作成多媒體教材。

而多媒體教材製作過程中的主要工作產品可以分為課程架構 (Structure)、學習元件 (SCO)、分鏡 (Scene)、劇情腳本 (Script)、使用者介面 (UI) 與演員 (Actor) 等設計元件，其關係如圖 9 所示。

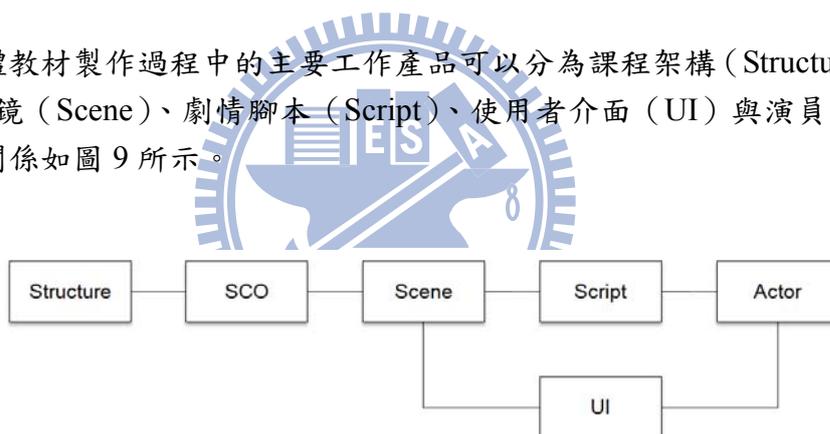


圖 9 設計元件關係圖

所以依照多媒體教材的開發生命週期與各開發階段的設計成果，我們定義了課程架構文件、SCO 描述文件、分鏡設計文件、腳本設計文件、UI 說明文件、素材清單與教材成果檔等製作過程文件，以記錄及呈現多媒體教材開發過程中的資訊，如表 2 所示。

表 2 製作過程文件列表

開發階段	設計元件	製作過程文件		
		文件名稱	資料類型	內容說明
教學內容設計 (RD)	Structure	課程架構文件	文字	依據原始教材內容規劃的多媒體教材架構，包含章節內容等資訊。
	SCO	SCO 描述文件	文字	描述每一章節對應的 SCO 學習資源的資訊。
單元腳本設計 (DM)	Scene	分鏡設計文件	文字	描述每一個 SCO 學習資源的分鏡內容。
	Script	腳本設計文件	文字	描述每一個分鏡的劇情腳本，包含開場、退場與互動劇情，以及使用的演員資訊。
場景 UI 設計 (UI)	UI	UI 說明文件	文字、 圖片	描述每一個分鏡的 UI 設計內容，包含演員配置與場景規劃示意圖等資訊。
素材製作 (MM)	Actor	素材清單 (文字)	文字、 圖片	列出多媒體教材使用的文字演員資訊。
		素材清單 (圖片)	文字、 圖片	列出多媒體教材使用的圖片演員資訊。
		素材清單 (聲音)	文字、 聲音	列出多媒體教材使用的聲音演員資訊。
		素材清單 (影片)	文字、 影片	列出多媒體教材使用的影片演員資訊。
		素材清單 (動畫)	文字、 動畫	列出多媒體教材使用的動畫演員資訊。
教材製作 (CM)	MLC	教材成果檔	多媒體	依據多媒體教材架構，展示製作完成的多媒體教材。

由於多媒體教材製作過程中的每一個開發階段都是環環相扣的，各階段的工作產品也是彼此相關的，所以製作過程文件之間也存在著相依性的關係，如圖 10 所示。而藉由文件關聯 (relation) 的建立，可以更清楚地呈現多媒體教材製作過程中的資訊，並可作為後續追蹤工作產品的基礎。

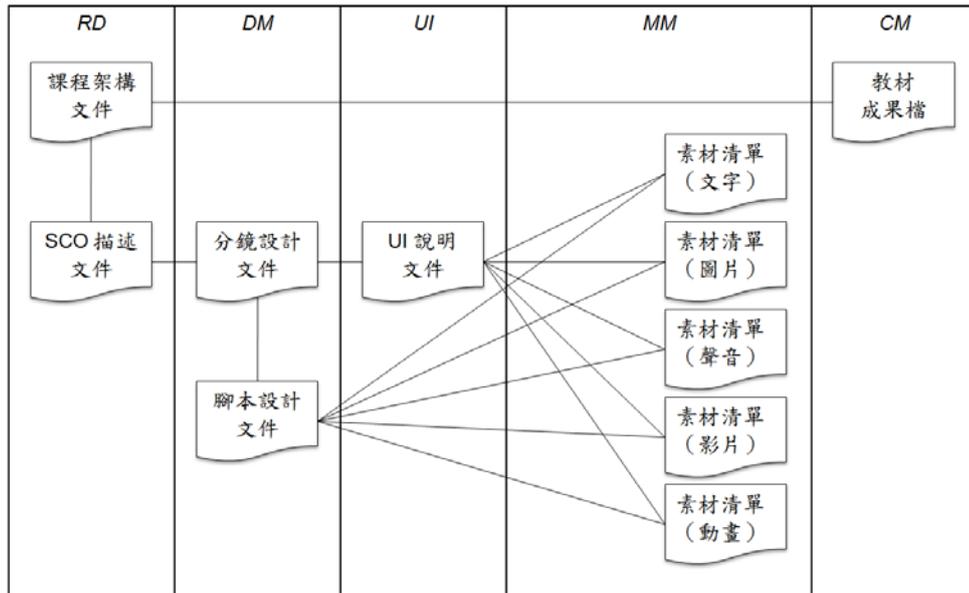


圖 10 製作過程文件關聯圖

3.2 品質狀態圖

對於多媒體教材製作過程中的品質，我們可以透過檢核製作過程文件的方式，評估每一個開發階段的產出是否符合需求且具有一定的品質，並可藉由品質狀態圖明確定義各階段工作產品的品質狀態，如圖 11 所示。



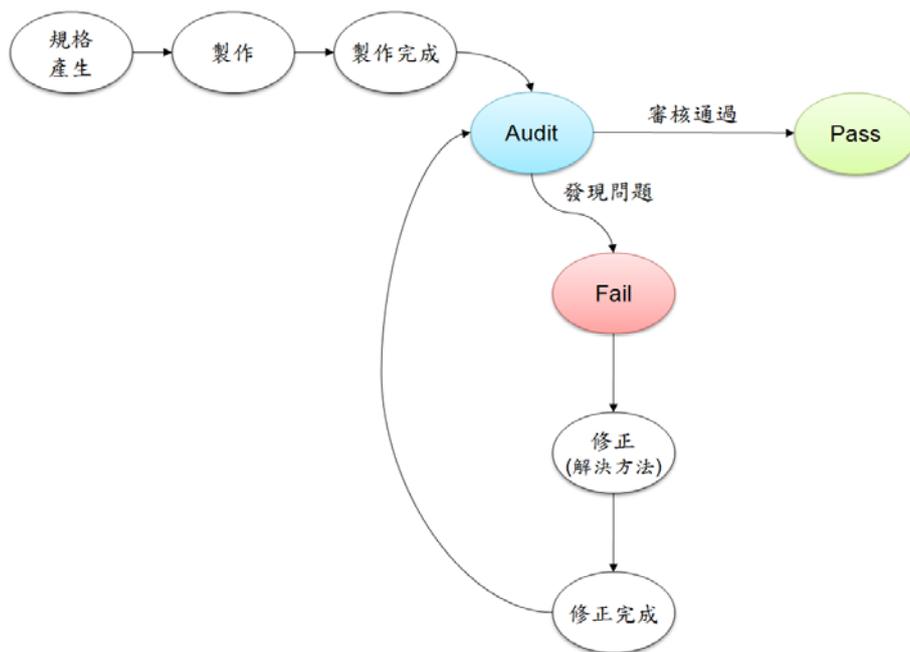


圖 11 品質狀態圖

當完成任何一個階段的設計內容時，品管人員即可透過製作過程文件審核工作產品的內容，審核通過的工作產品則可進到「Pass」的狀態，但若發現任何品質問題時，該工作產品則進到「Fail」的狀態，此時專案成員需要針對所發現的問題做修正，並於修改完成後再次經由品管人員進行審核，直至所有工作產品都通過審核，即完成該階段的品管作業。

由於多媒體教材每一個開發階段的工作產品的性質不同，且各工作產品的審核重點與可能發生的品質問題亦不盡相同，因此針對每一份製作過程文件的內容，我們定義了不同的品質狀態圖，以詳實記錄每一個工作產品發生的品質問題，並適當地進行修正：

(1) 課程架構文件

課程架構文件包含章節內容等資訊，審核時主要評估每一個章節的內容完整性、語意描述清晰度，以及分段是否正確，並且需要確認是否有錯別字或其他的品質問題，其品質狀態圖如圖 12 所示。

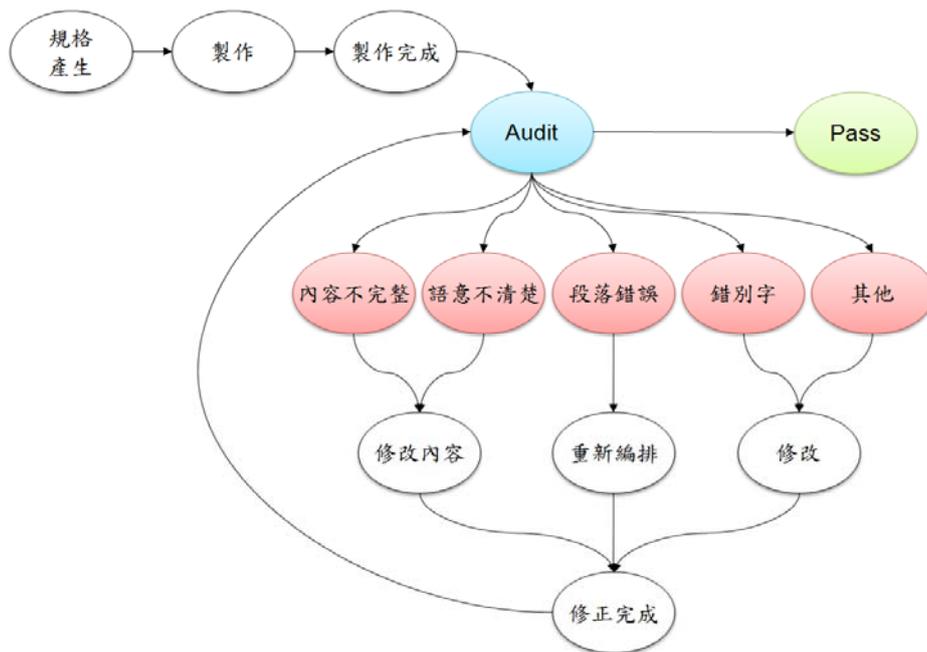


圖 12 品質狀態圖－課程架構文件

(2) SCO 描述文件

SCO 描述文件包含每一章節對應的 SCO 學習資源的資訊，審核時主要評估 SCO 的內容描述是否完整，並且需要確認是否有其他的品質問題，其品質狀態圖如圖 13 所示。

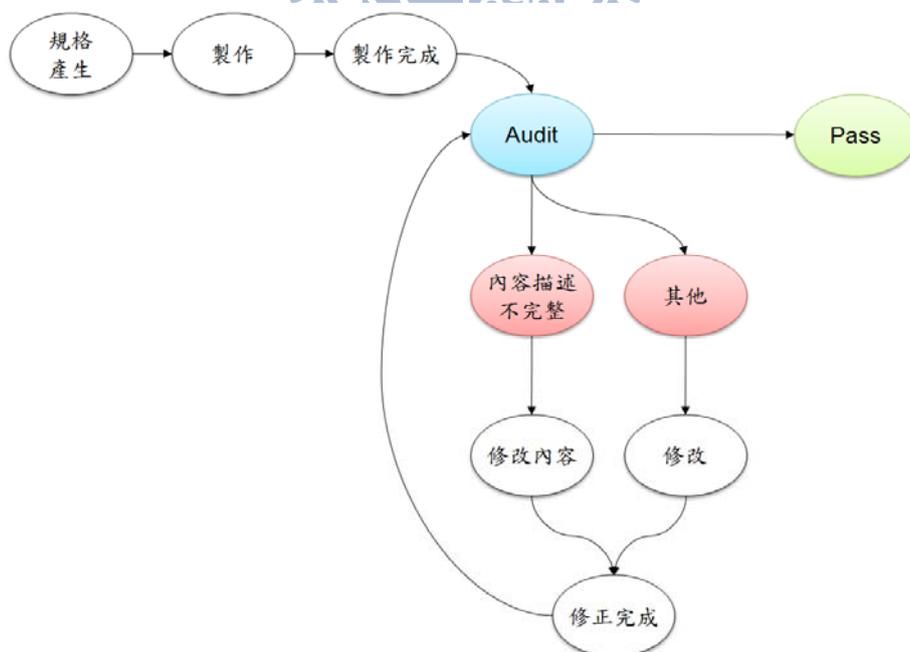


圖 13 品質狀態圖－SCO 描述文件

(3) 分鏡設計文件

分鏡設計文件包含 SCO 學習資源的分鏡內容，審核時主要評估分鏡數量、內容安排與演員數量是否適當，並且需要確認是否有其他的品質問題，其品質狀態圖如圖 14 所示。

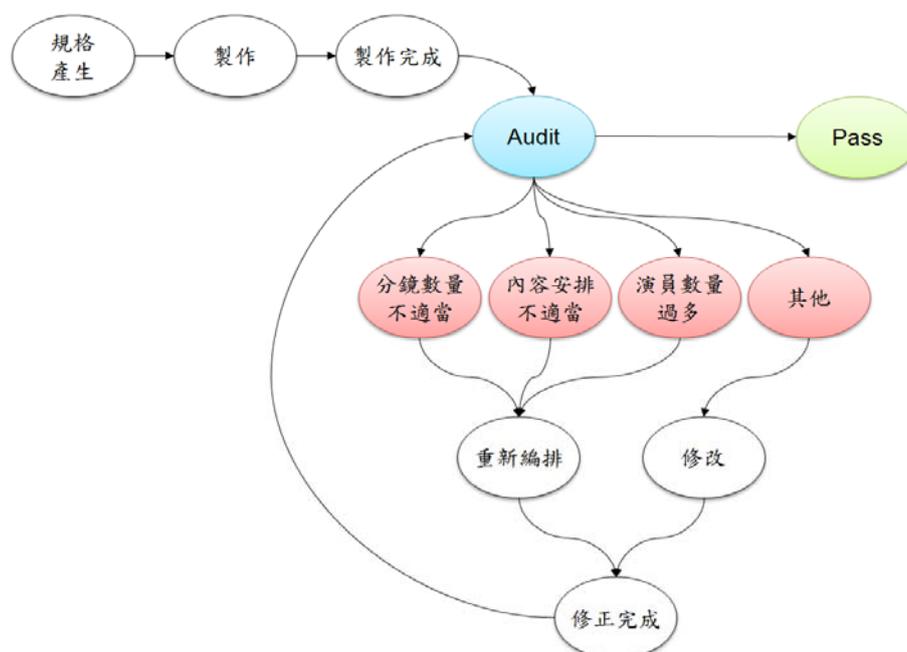


圖 14 品質狀態圖—分鏡設計文件

(4) 腳本設計文件

腳本設計文件包含分鏡的劇情腳本與使用的演員資訊，審核時主要評估每一個劇情的語意描述清晰度、內容正確性，以及腳本流程與演員間的互動方式是否適當，並且需要確認是否有其他的品質問題，其品質狀態圖如圖 15 所示。

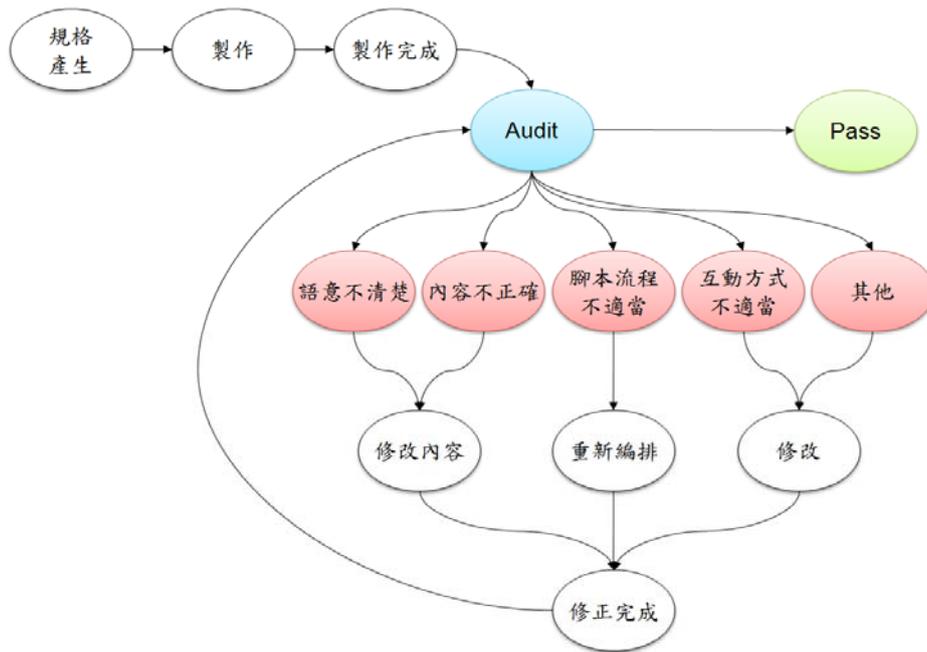


圖 15 品質狀態圖－腳本設計文件

(5) UI 說明文件

UI 說明文件包含分鏡的 UI 設計內容，審核時主要評估每一個場景的物件配置方式與整體的設計風格是否適當，並且需要確認是否有其他的品質問題，其品質狀態圖如圖 16 所示。

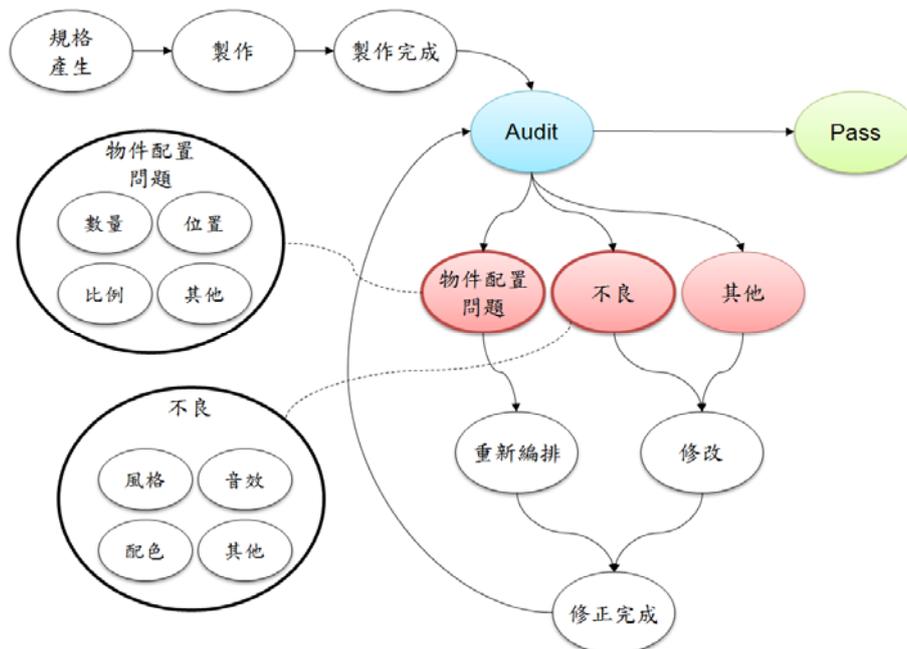


圖 16 品質狀態圖－UI 說明文件

(6) 素材清單（文字）

素材清單（文字）包含文字演員資訊，審核時主要評估每一段文字是否與內容相符，以及呈現的方式是否適當，並且需要確認是否有其他的品質問題，其品質狀態圖如圖 17 所示。

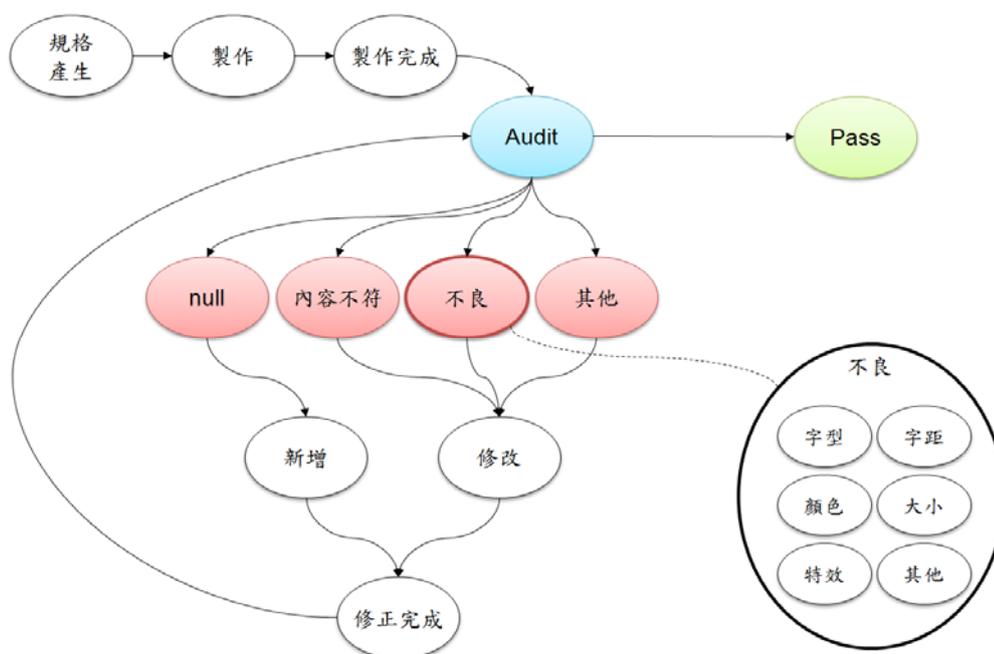


圖 17 品質狀態圖—素材清單（文字）

(7) 素材清單（圖片）

素材清單（圖片）包含圖片演員資訊，審核時主要評估每一個圖片檔案的正確性，以及呈現的方式是否適當，並且需要確認是否有其他的品質問題，其品質狀態圖如圖 18 所示。

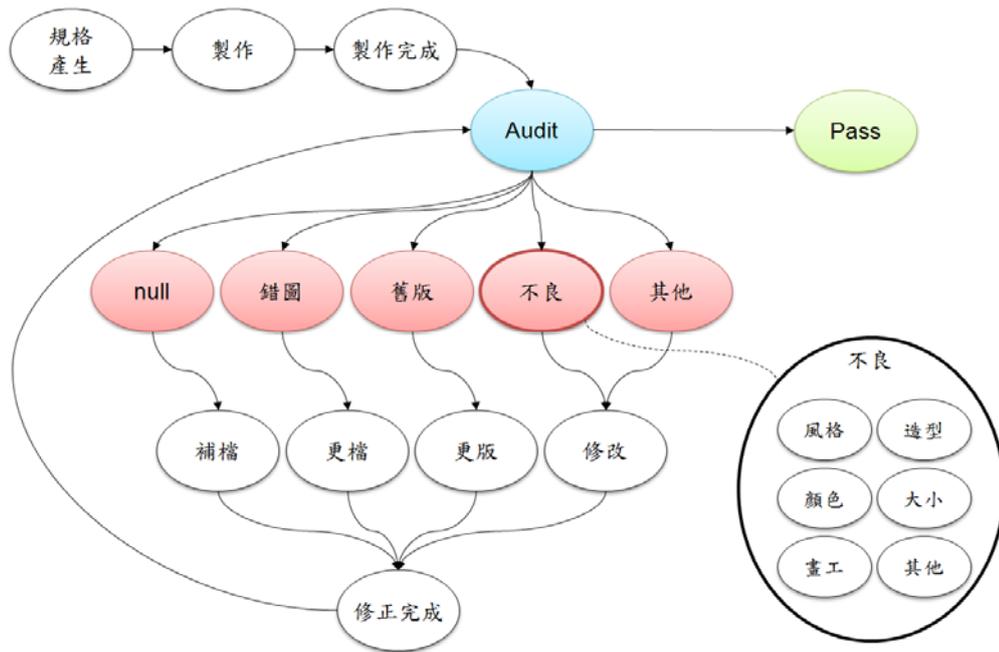


圖 18 品質狀態圖－素材清單（圖片）

(8) 素材清單（聲音）

素材清單（聲音）包含聲音演員資訊，審核時主要評估每一個聲音檔案的正確性，以及播放的方式與效果是否適當，並且需要確認是否有其他的品質問題，其品質狀態圖如圖 19 所示。

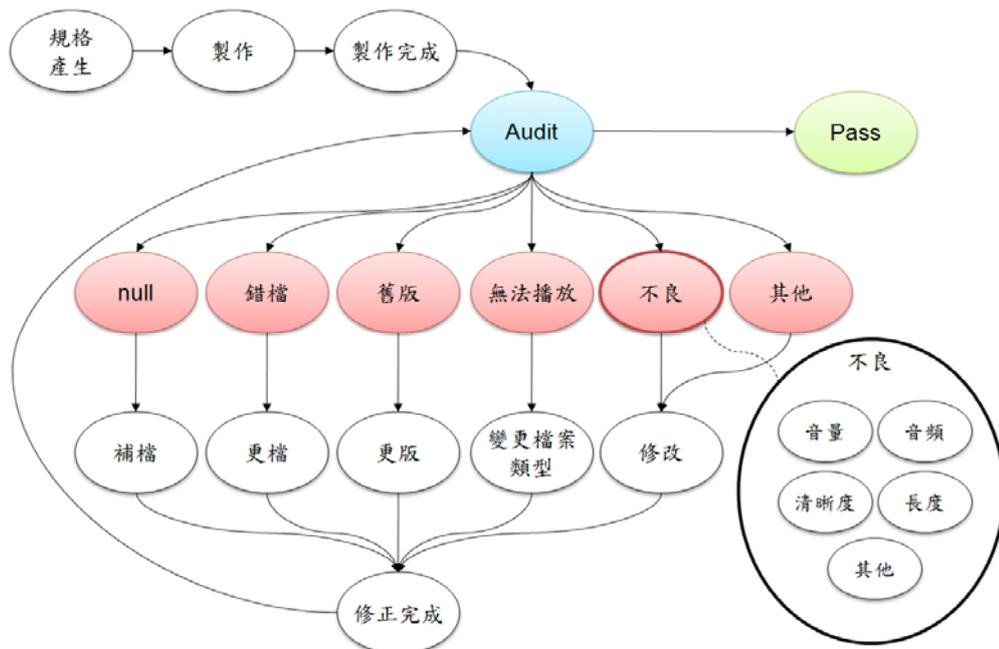


圖 19 品質狀態圖－素材清單（聲音）

(9) 素材清單（影片）

素材清單（影片）包含影片演員資訊，審核時主要評估每一個影片檔案的正確性，以及播放的方式與效果是否適當，並且需要確認是否有其他的品質問題，其品質狀態圖如圖 20 所示。

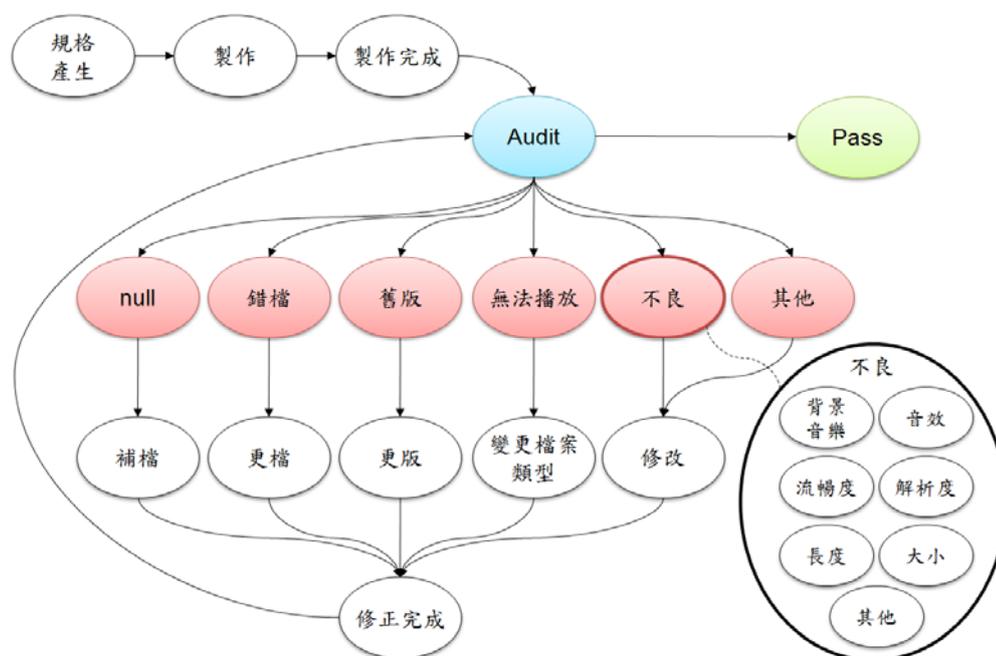


圖 20 品質狀態圖—素材清單（影片）

(10) 素材清單（動畫）

素材清單（動畫）包含動畫演員資訊，審核時主要評估每一個動畫檔案的正確性，以及播放的方式與效果是否適當，並且需要確認是否有其他的品質問題，其品質狀態圖如圖 21 所示。

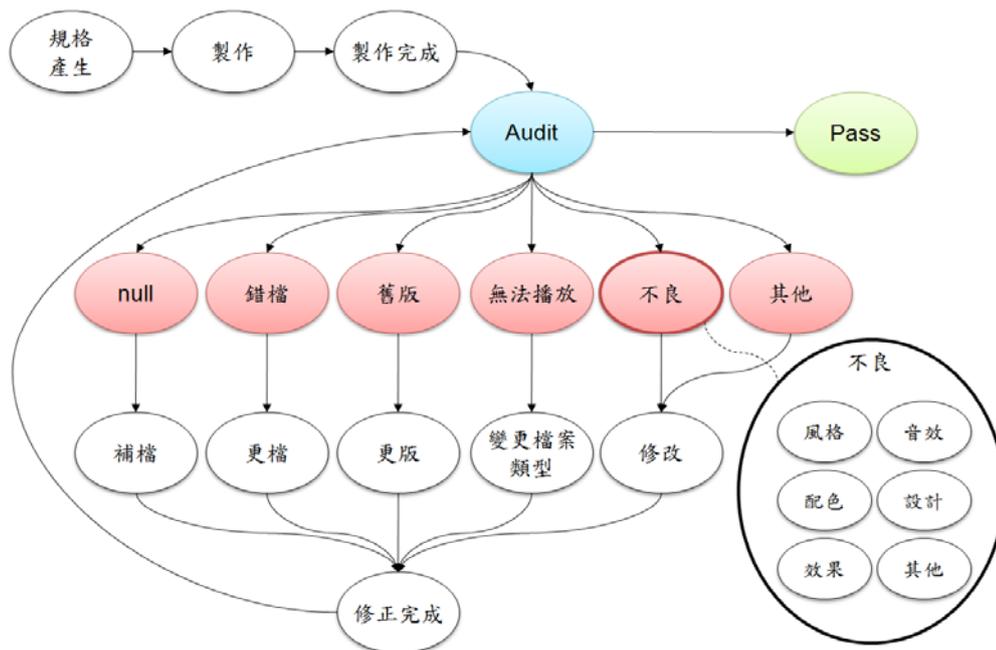


圖 21 品質狀態圖－素材清單（動畫）

(11) 教材成果檔

教材成果檔包含製作完成的多媒體教材，審核時主要評估每一個教材檔案的正確性，以及播放的方式與效果是否適當，並且需要確認是否有其他的品質問題，其品質狀態圖如圖 22 所示。

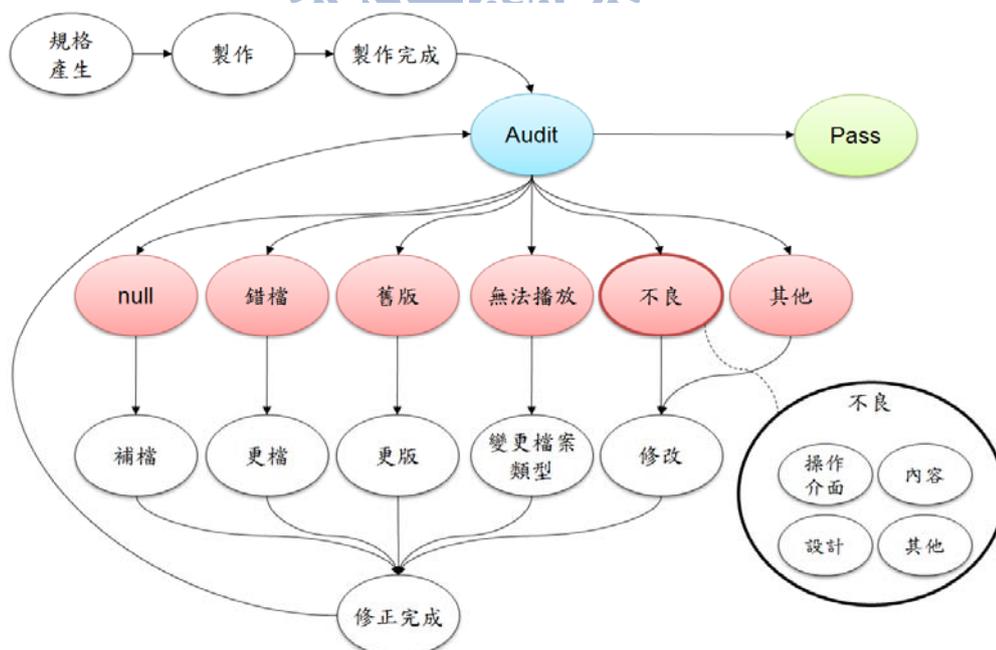


圖 22 品質狀態圖－教材成果檔

3.3 品質評估模型

在評估多媒體教材的品質時，我們將多媒體教材的品質依照教材（Lecture）、開發階段（Phase）、製作過程文件（Document）與項目（Item）四個層級分別進行評估。其中，教材的品質取決於已完成的開發階段的品質，而各開發階段的品質是由該階段的製作過程文件的品質所計算出來的，最底層的項目則反應每一個開發階段的工作產品的品質，並決定製作過程文件的品質。整個多媒體教材品質評估的專案模型圖如圖 23 所示。

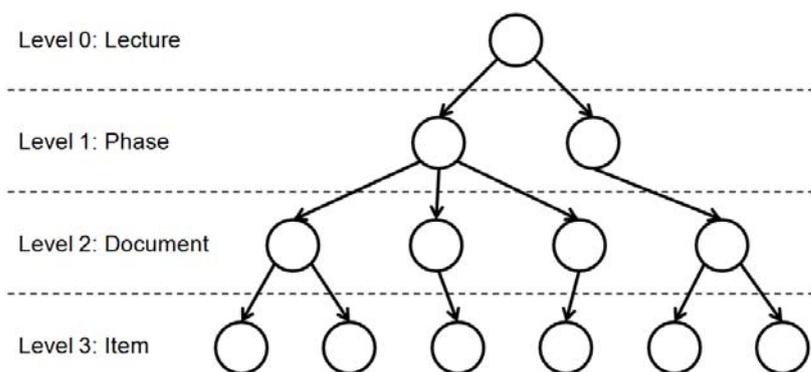


圖 23 專案模型圖

因此在多媒體教材的製作過程中，可以藉由下列二式來評估多媒體教材的品質：

$$S_A = \frac{\sum_{i=1}^n S_i^{child}}{n} \quad (1)$$

$$S_{Item} = \begin{cases} 1 \cdot C_1^{(n_{audit}-1)} & \text{if pass,} \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

首先，由於 Level 0 至 Level 2 的每一節點的品質分數取決於其子節點的品質分數，因此當需要評估特定節點的品質時，則可運用 (1) 式取得子節點品質分數的平均值，並作為該節點的品質分數。

而最底層的 Item 則可由 (2) 式取得品質分數，其中品質係數 (C_1) 之值介於 0 與 1 之間 ($0 \leq C_1 \leq 1$)。由於 Item 的品質分數代表品質審核的結果，當通過審核時即可得到 1 分，但為了呈現經過多次審核的結果，所以乘上品質係數以計算出實際可獲得的品質分數。例如：當品質係數設定為 0.8 時，第一次就通過審核的 Item 可得到 1 分，而第二次審核才通過的 Item 則只能得到 0.8 分。

因此我們可以藉由以下演算法取得特定節點的品質分數：

```
GET-SCORE(A)
1  if A has child
2      then score[A] ← average score of completed child nodes
3      else if pass
4          then score[A] ← score of audit result
5          else score[A] ← 0
6  return score[A]
```

當取得 Lecture 的品質分數之後，則可以運用類比的方法，由已完成的多媒體教材的資訊來預測此多媒體教材製作完成時的品質。預測的方式是先依照預先設定的誤差值計算出品質分數的區間，接著在已完成的多媒體教材之中，搜尋品質分數符合此區間的多媒體教材，若找不到符合的多媒體教材，則將誤差值加倍後再度進行搜尋，當找到符合的多媒體教材之後，再依序比對下一個層級的品質分數，直到 Item 層級為止，最後將所有符合的多媒體教材其 Lecture 層級的品質分數取平均值，即為預測的品質分數。

由於每一份教材最底層的 Item 內容與數量不相同，因此在比對 Item 層級的品質分數時，是以工作負荷（數量÷開發人力）取代品質分數來作為比較的基準，藉此降低不同多媒體教材之間的差異性。

此品質預測方法的限制則是需要先有已完成的多媒體教材的資訊，才可以進行預測，而且每一個多媒體教材的品質分數都必須是使用相同的品質係數所得到的，以避免因品質評估的計算方式不同而有誤差。品質預測的演算法如下：

GET-CRITERION(*A*, *type*)

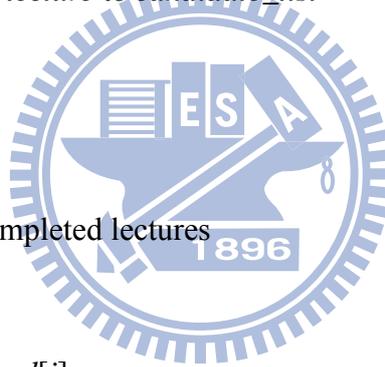
```
1  if type ≠ Item
2      then criterion[A] ← GET-SCORE(A)
3      else criterion[A] ← workload (number of item/man-day) of A
4  return criterion[A]
```

GET-CANDIDATE-LIST(*lecture_list*, *criterion*)

```
1  candidate_list ← NIL
2  e ← default error coefficient ( $0 \leq e \leq 1$ )
3  if lecture_list ≠ NIL
4      while candidate_list = NIL and  $e \leq 1$ 
5          lower ← criterion · (1 - e)
6          upper ← criterion · (1 + e)
7          for each lecture in lecture_list
8              if evaluation of lecture between lower and upper
9                  add lecture to candidate_list
10         e ← 2 · e
11 return candidate_list
```

PREDICT-SCORE(*L*)

```
1  lecture_list ← list of completed lectures
2  i ← 0
3  while i < 3
4      for each node at level[i]
5          if node[L] is completed
6              criterion ← GET-CRITERION(node[L], NIL)
7              lecture_list ← GET-CANDIDATE-LIST(lecture_list, criterion)
8  for each node at level[2]
9      if node[L] is completed
10         criterion ← GET-CRITERION(node[L], Item)
11         lecture_list ← GET-CANDIDATE-LIST(lecture_list, criterion)
12 predict_score ← average score of lectures in lecture_list
13 return predict_score
```



四、系統分析與設計

本章節將介紹多媒體教材品質管制系統的整體架構，以及描述本研究之文件產生器與系統中其他模組的關係。接著則針對本研究提出的文件產生器與品質管理機制，進行需求分析及系統架構設計，並詳細說明主要功能模組的實作方式。

4.1 多媒體教材品質管制系統

多媒體教材品質管制系統（MLCs Quality Control System, MQCS）建構了多媒體教材開發流程的品質控管輔助機制，原始規劃的系統架構包含 MDE（MLCs Dependency Editor）、MCD（MLCs Change Detector）、MDID（MLCs Inconsistent Detector）、Files/Documents Management Module 與 MAT（Multimedia Editor/Multimedia Authoring Tool）模組[2]。

之後的研究則加入流程控管（Process Control）、驗證（Verification）與 MDG（MLCs Directed Graph）的設計，以協助提升多媒體教材製作過程中的品質[21][22][23]。而本研究則是實作應用在多媒體教材製作過程的文件產生器（Document Generator），並進一步發展多媒體教材製作過程之品質管理機制，透過有效控管多媒體教材製作過程中的品質問題，讓多媒體教材能有一定的品質。因此多媒體教材品質管制系統的整體架構如圖 24 所示。

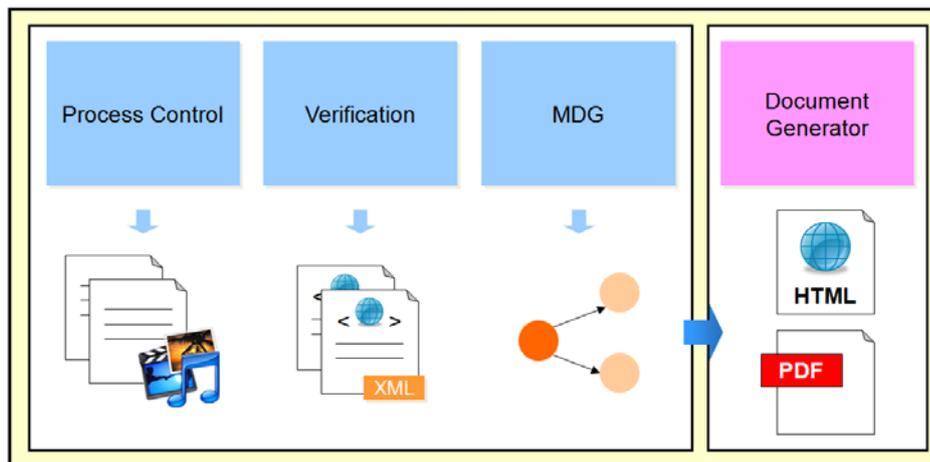


圖 24 多媒體教材品質管制系統

對於一份多媒體教材而言，透過多媒體教材品質管制系統各個模組所提供的資訊，並運用文件產生器自動產生多媒體教材的製作過程文件，最終的工作產品則可以包含開發階段的設計內容、設計元件關聯圖、教材原始碼與教材成品這四個面向的資訊，提供專案成員以不同的角度來檢視多媒體教材。

4.2 需求分析

為提升多媒體教材製作過程中的品質，我們提出以一套可以應用在多媒體教材製作過程的文件產生器，協助專案成員透過製作過程文件進行追蹤及確認每一個開發階段的正確性，另外，並建立品質管理的機制，以有效控管多媒體教材製作過程中的品質問題。因此接下來我們將介紹文件產生器與品質管理機制的主要功能需求，以及說明系統的使用案例圖。

4.2.1 文件產生器

依照多媒體教材的特性與多媒體教材品質管制系統中各模組的設計方式，我們規劃的文件產生器需要具備以下的功能，以實際應用在多媒體教材的製作過程，並且和多媒體教材品質管制系統的其他模組共同運作：

(1) 支援多媒體檔案格式

多媒體教材的製作過程文件除了基本的文字資訊之外，還需要展示各種多媒體檔案的播放效果，因此在設定文件的格式時，則要能夠指定各種多媒體檔案的播放方式，讓產生的製作過程文件可以適當地呈現多媒體教材各開發階段的設計成果。

(2) 提供文件追蹤功能

文件追蹤功能則需要可以設定製作過程文件之間的關聯資訊，以正確建立文件之間的追蹤機制，讓專案成員可以透過製作過程文件追蹤各開發階段的設計內容。

(3) 上傳資料與多媒體檔案

由於文件產生器的資料來源是由多媒體教材品質管制系統的其他模組所提供的，因此需要讓專案成員可以依照多媒體教材的開發階段上傳各階段的資料與多媒體檔案，最後再由系統依照所設定的文件格式與上傳的資料產生製作過程文件。

(4) 輸出可列印文件

為了讓使用者可以更方便地傳遞及保存多媒體教材的製作過程文件，因此不僅需要能夠線上瀏覽製作過程文件，並且需要可以輸出可列印文件的功能，將多媒體教材的開發資訊彙整成一份文件。

4.2.2 品質管理

由於我們希望透過實作 CMMI 之 PPQA 流程領域，以建立品質管理的機制，因此依據 PPQA 流程領域定義的特定目標與特定執行方法歸納出以下三個實作的重點：

(1) 客觀性 (Objectively)

品質審核需要可以由獨立的品管團隊 (QA Team) 透過製作過程文件來評估多媒體教材的品質，藉此以比較客觀的角度提出問題與建議，進而協助專案成員與管理人員瞭解系統的品質狀況，並持續進行改善。

(2) 解決方法 (Resolution)

專案經理 (Project Manager, PM) 則必須可以針對每一個品質問題提出解決方法，並且在確認所有的問題都有對應的矯正措施之後，再度由品管團隊進行審核。

(3) 記錄 (Log)

記錄問題對於提升品質亦是相當重要的，藉由分析所發現的問題可以協助瞭解及改善開發流程[24]。因此整個品管活動均需要留下記錄，並且可以讓專案成員直接在製作過程文件上檢視及追蹤問題。

4.2.3 使用案例圖

因此總結功能需求，理想中應用於多媒體教材製作過程的文件產生器，必須可以支援多媒體檔案格式、提供文件追蹤功能、可以上傳資料與多媒體檔案及輸出可列印文件；而品質管理的部分，則要達到 PPQA 流程領域所定義的特定目標，即由品管團隊審核教材品質、確認每一個問題都有提出矯正措施，以及記錄品管活動的資訊。而整個系統的使用案例圖 (Use Case Diagram) 則如圖 25 所示。

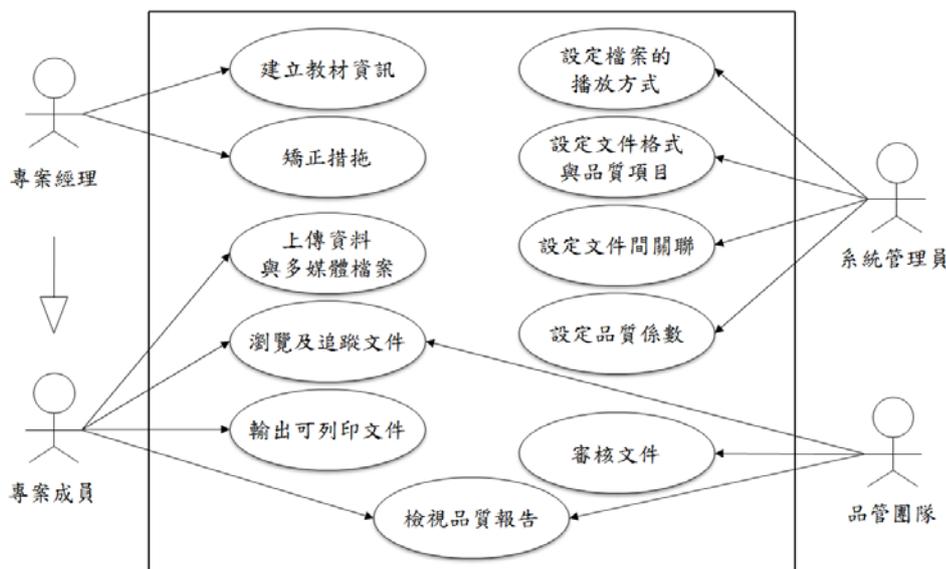


圖 25 使用案例圖

接著說明每一個使用案例的內容：

(1) 設定檔案的播放方式

系統管理員必須預先設定各種多媒體檔案的播放方式，當系統產生製作過程文件時，即可依照設定在文件中呈現多媒體檔案。

(2) 設定文件格式與品質項目

系統管理員依照多媒體教材的開發流程，逐一設定各開發階段的製作過程文件的資訊，以及設定每一份文件的格式與品質項目，當進行品質審核時，即可依照品質項目審核文件中的設計元件。

(3) 設定文件關聯

系統管理員依照製作過程文件之間的相依性關係，建立文件之間的關聯資訊，當系統產生製作過程文件時，就可以透過此設定進行文件追蹤。

(4) 設定品質係數

系統管理員統一設定多媒體教材品質評估模型的品質係數，在計算專案模型中任一節點的品質分數時，即可呈現經過多次審核的結果。

(5) 建立教材資訊

專案經理建立教材的基本資料，包含教材名稱、教材目標、教材目的等資訊，以初始化一份多媒體教材的開發專案。

(6) 上傳資料與多媒體檔案

專案成員依照多媒體教材的開發流程，依序上傳各開發階段的資料與多媒體檔案。

(7) 瀏覽及追蹤文件

專案成員與品管團隊可以瀏覽系統產生的製作過程文件，並可依照文件之間的關聯資訊，進行文件追蹤。

(8) 輸出可列印文件

專案成員可以在多媒體教材開發完成時，下載及列印整份多媒體教材的製作過程文件。

(9) 審核文件

品管團隊針對每一份製作過程文件，逐一審核文件中的設計元件內容，並記錄所發現的品質問題。

(10) 矯正措施

專案經理針對每一個品質問題進行追蹤及提出解決方法，並在系統記錄每一個問題的矯正措施。

(11) 檢視品質報告

專案經理與品管團隊可以檢視系統依照品質問題與矯正措施產生的品質報告，其內容包含多媒體教材的品質狀態等資訊。



4.3 系統設計與實作

4.3.1 開發環境

系統的開發環境如圖 26 所示。

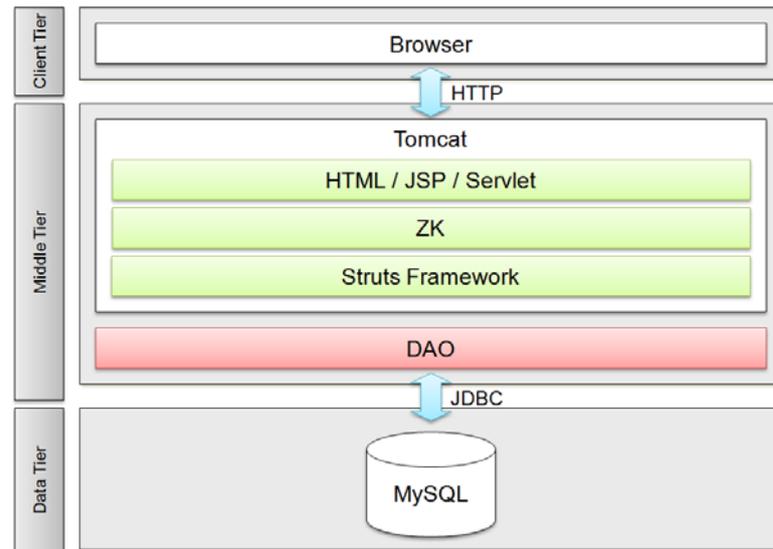


圖 26 系統環境圖

由於網路的普及與便利性，我們將以 web 的方式進行系統開發，讓前端使用者可以直接透過瀏覽器操作系統，不需要再另外安裝軟體。

在應用程式端則使用 Tomcat 作為 Web Container 以提供服務，並實作 Struts Framework，以透過 MVC (Model-View-Control) 的設計模式提升系統的維護性。另外，並在前端使用者介面使用 ZK，以 Ajax (Asynchronous JavaScript and XML) 技術改善網頁操作的流暢度。

資料層級則以 MySQL 作為資料庫，並透過資料存取物件(Data Access Object, DAO) 對資料進行操作。

4.3.2 系統架構

整個系統的主要模組分別為文件產生器 (Document Generator)、文件追蹤機制 (Document Tracing Mechanism)、品質管理 (Quality Management)、品質評估模型 (Quality Evaluation Module) 與品質報告 (Reporting)，如圖 27 所示。

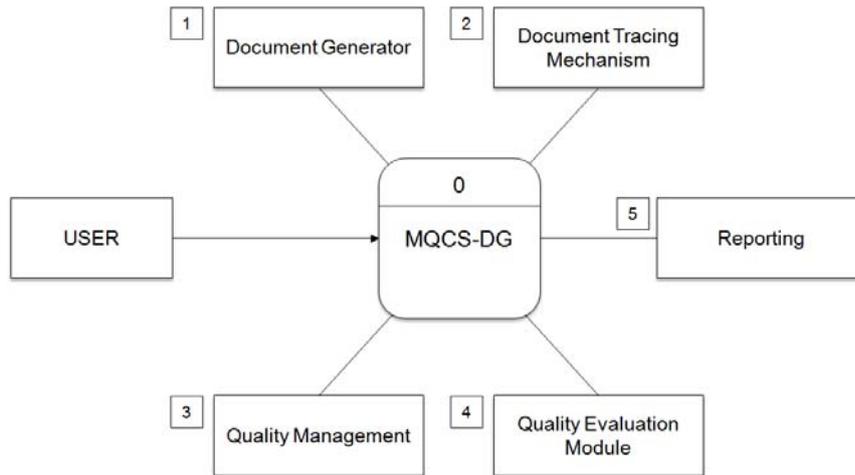


圖 27 系統架構圖

各模組的功能說明如下：

(1) 文件產生器 (Document Generator)

依照定義的文件格式與上傳的文件資料，產生 HTML (HyperText Markup Language) 與 PDF (Portable Document Format) 格式的製作過程文件，提供以線上瀏覽的方式查看文件，或是利用可列印文件的方式輸出整份文件。

(2) 文件追蹤機制 (Document Tracing Mechanism)

依照製作過程文件之間的關聯資訊，協助在文件之間進行追蹤，當有追蹤的需求時，則先透過此模組開啟所有相關的文件，再依其內容分別定位到相關的設計元件。

(3) 品質管理 (Quality Management)

管理及記錄所有的品管活動，藉由直接審視製作過程文件的內容，確認設計內容是否符合需求，並且針對所有發現的品質問題提出矯正措施。

(4) 品質評估模型 (Quality Evaluation Module)

運用品質評估模型衡量多媒體教材的品質狀態及預估多媒體教材製作完成時的品質。

(5) 品質報告 (Reporting)

匯集品質管理記錄的品質問題與矯正措施，以及運用品質評估模型所得到的多媒體教材的品質狀態，產生品質報告。

而接下來的小節將詳細說明主要功能模組的實作方式。

4.3.3 文件產生器

一般文件產生的方式可以分為輸入 (Input)、轉換 (Transform)、輸出 (Output) 三個步驟，如圖 28 所示。

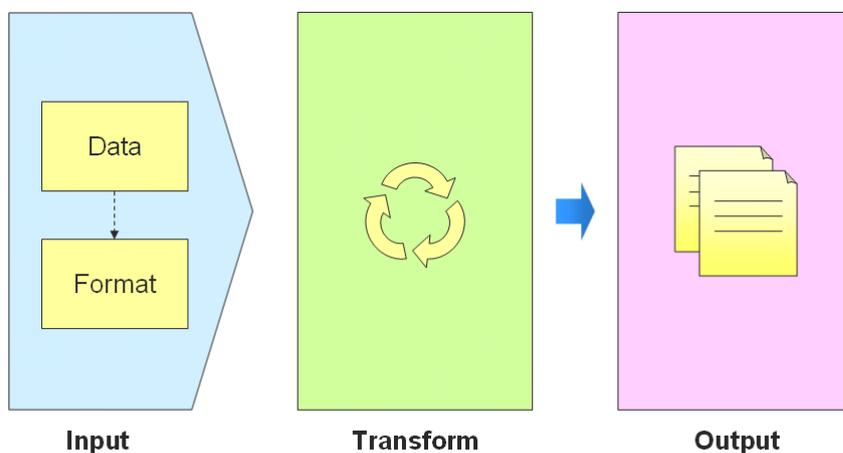


圖 28 一般的文件產生方式

輸入方面必須先依據資料的內容設定好文件的格式，接著就可以針對不同的資料內容來產生文件。由於多媒體教材的製作過程文件需要支援多媒體檔案的呈現，因此對於應用在多媒體教材製作過程的文件產生器，我們提出的文件產生方式則如圖 29 所示。

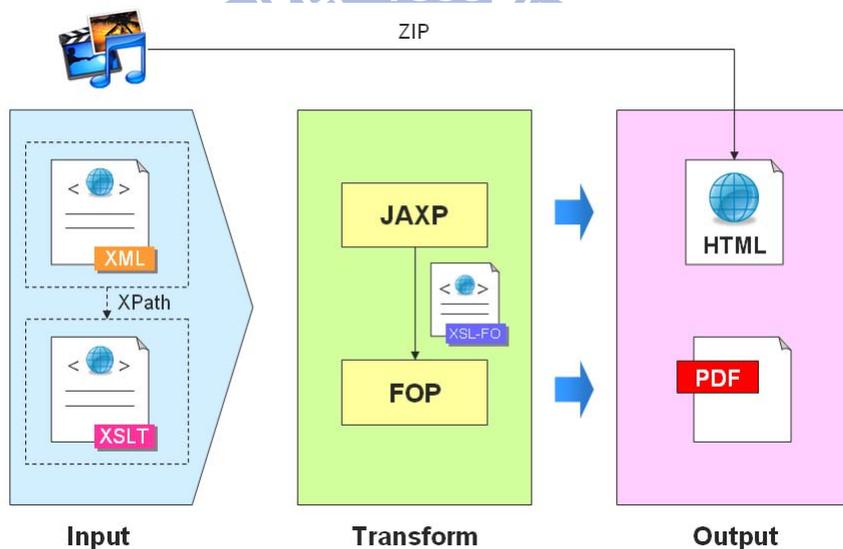


圖 29 提議的文件產生方式

其中，輸入部分的資料內容是以 XML (Extensible Markup Language) 作為儲存格式，以方便解析文件，及在各系統模組之間傳遞資訊。文件的排版與格式設定則使用 XSLT (Extensible Stylesheet Language) 進行格式的定義及轉換，以呈現出所需要的文件

格式。

轉換的部分則是運用現有的 JAXP (Java API for XML Processing) 模組來處理 XML 的資料，並依照 XSLT 的格式定義，將 XML 轉換為 HTML 的文件。而 PDF 的部分，則需要再多一層轉換的工作，也就是利用 JAXP 模組先輸出 XSL-FO (XSL Formatting Objects) 的文件，然後再透過 Apache FOP (Formatting Objects Processor) 套件，協助將文件轉換成 PDF 的格式。

而多媒體檔案則由使用者以壓縮檔的方式整包上傳到系統後，再由系統依照對應的目錄結構進行解壓縮。

在此文件轉換的架構中，最重要的部分在於如何設定每一個資料在文件中的位置，特別是如如何安排多媒體檔案的呈現方式，而這些都是在文件格式設定的步驟來完成。設定文件格式時，必須依照 XML 的文件定義檔 (XML Schema) 來安排每一個資料在文件中的呈現方式，並利用 XPath (XML Path Language) 呈現在 XSLT 的文件中，如圖 30 所示。

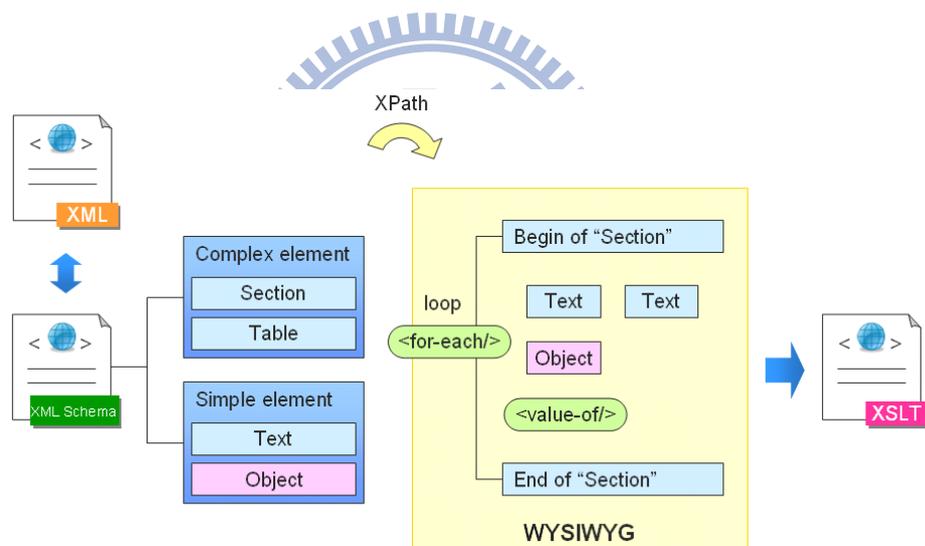


圖 30 文件格式設定方式

針對 XML Schema 文件中的 Complex element，我們將以 loop 的結構來表示，以顯示節點下的所有資料，而在轉換成 XSLT 的文件時，則以 XSLT 語法中的 <for-each/> 來呈現；Simple element 則以文字的方式直接顯示出來，而在轉換成 XSLT 時，就利用 <value-of/> 的方式來呈現。

而為了支援多媒體檔案的呈現，在 Simple element 的部分，我們另外設計一個 Object 的型態，當文件中的資料如果是一個多媒體檔案的名稱且要用多媒體檔案的方式來呈現時，則在做格式設定的時候，就需要選擇用 Object 的型式來顯示，接下來系統就會特別處理 Object 型態的資料。

由於檔案名稱中的副檔名是一個可以很容易且快速辨別檔案型態的方式，所以針對 Object 型態的資料，系統將會在 XSLT 的文件中，加入 `<choose/>` 的語法，以實際依照多媒體檔案的副檔名判斷其檔案類型，並嵌入這個多媒體檔案對應的播放語法，讓多媒體檔案可以正常地顯示，如圖 31 所示。

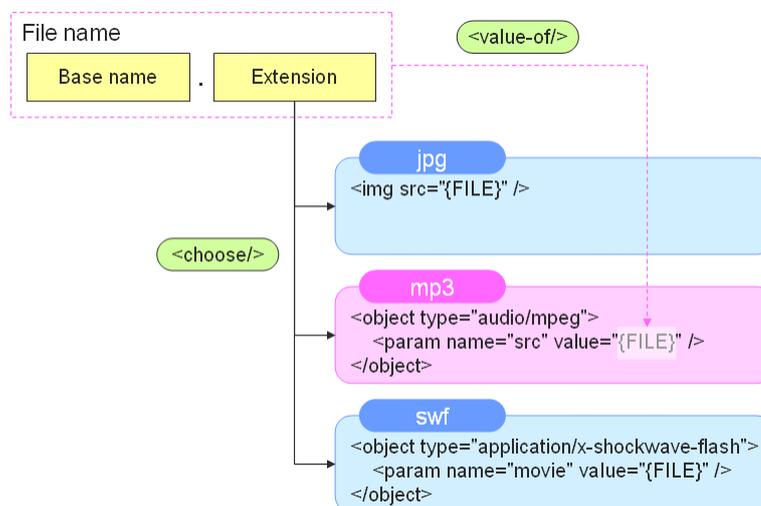


圖 31 Object 型態資料處理方式

4.3.4 文件追蹤機制

文件之間要能彼此進行追蹤，很重要的一點是要知道文件中的資料是什麼。雖然 XML 是一份可以解釋自己資料內容的文件，但當我們將 XML 文件轉換為 HTML 的格式時，基本上已移除了它的標籤名稱的資訊，只將資料的部分顯示出來而已，無法由產生的 HTML 文件知道原來的資料是什麼。因此我們利用 XPath 的型式，以書籤的方式將原來 XML 文件中的標籤資料，嵌入在最後產生的 HTML 文件中，以便辨識文件中的資訊，如圖 32 所示。

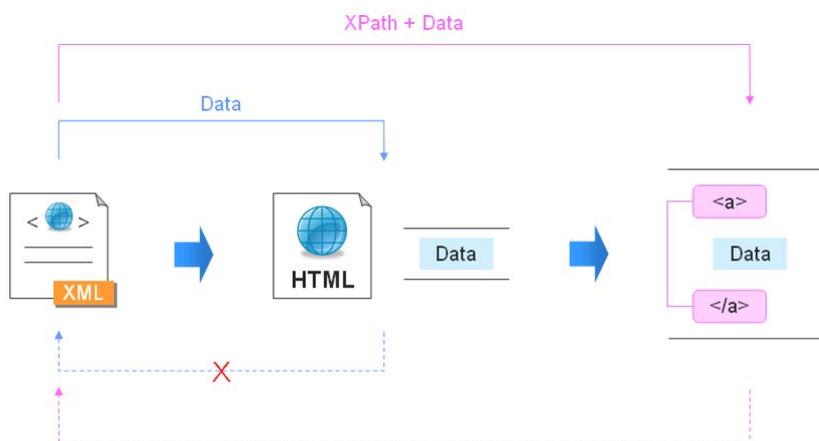


圖 32 XML 標籤資訊的記錄方式

而文件關聯的設定，同樣是透過 XPath 來做連結。假設文件 A 與文件 B 是彼此相關的，而且我們希望在瀏覽文件 A 的時候，就可以隨時追蹤到文件 B 中的資訊，因此我們需要設定文件 A 中的「/A/b/id」的資料可以對應到文件 B 中的「/B/id」的資料，並將這個關聯的資訊儲存在資料庫中，等到實際產生資料時，就可以依照資料的內容進行追蹤。

此時假設我們對文件 A 中「/A/b/id」是 001 的資料有興趣，我們就可以透過之前設定的關聯，找到文件 B 中「/B/id」這個 XPath，接著再到文件 B 中找出「/B/id」之值是 001 的資料，即可達到文件追蹤的功能。

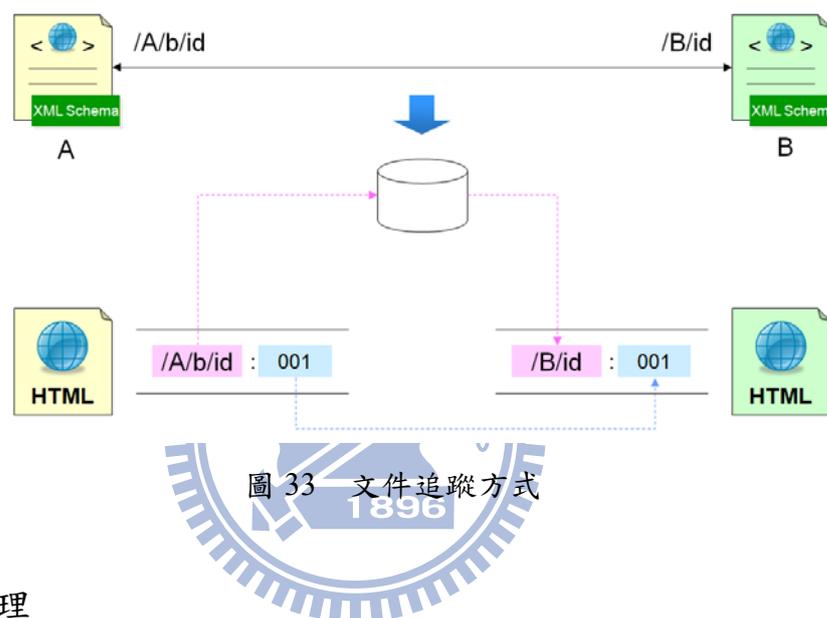


圖 33 文件追蹤方式

4.3.5 品質管理

最後談到品質管理的部分，因為我們希望可以讓專案成員直接在製作過程文件上檢視及追蹤問題，但是我們在文件中加上的書籤資訊可能會有重複的情形發生，因此對於文件中的每一個品質問題，則需要同時儲存資料在文件中呈現時的座標值，以確定品管團隊所描述的問題是指文件中那一個部分的資料，如圖 34 所示。

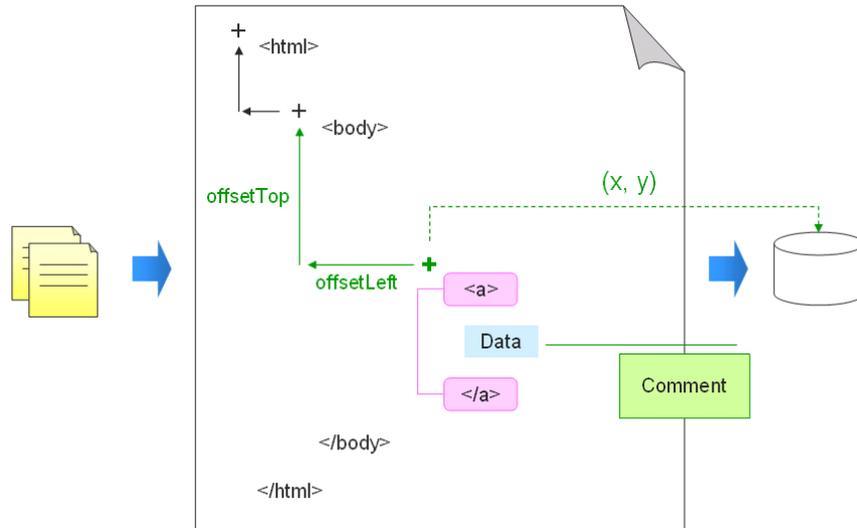


圖 34 品質問題的記錄方式

由於我們需要記錄每一個品質問題出現在文件中的位置，因此在每次進行品質審核之前，則必須將這個時點的文件先做快照 (snapshot)，接著再針對複製出來的文件進行品質審核。

而如果在品質審核的過程中有發現任何品質問題時，專案經理則必須進行追蹤及提出解決方法。當修正完成且確認所有的問題都有對應的矯正措施之後，就可以交由品管團隊再度進行審核，此時同樣由系統將此時點的文件做一次快照，品管團隊再依照複製的文件進行審核，如圖 35 所示。

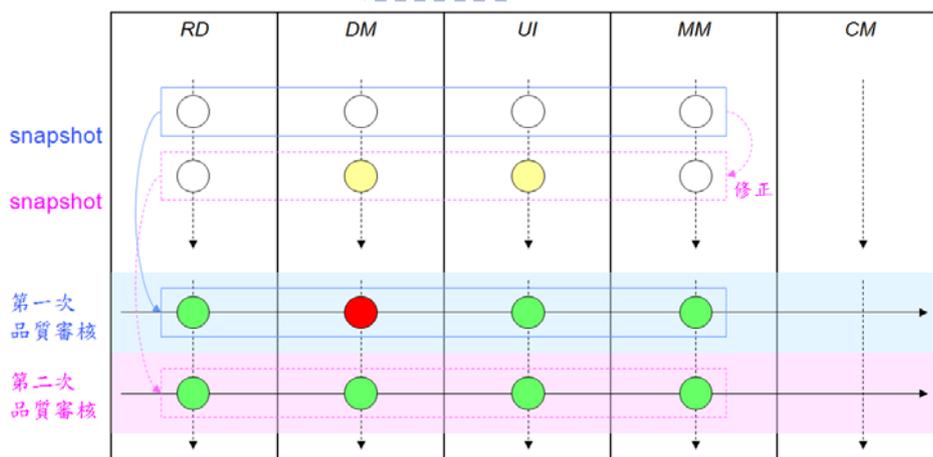


圖 35 品質審核方式

最後確認文件中的所有設計內容都有符合品質的標準時，即完成這個階段的品管活動。

五、應用實例

本章節將以「ABC in Phonics」之“What is on the Mat?” 章節內容作為應用實例，實際展示本研究的設計成果。

5.1 範例說明

此章節展示的多媒體教材範例的原始教案「ABC in Phonics」是新竹市政府配合行政院「挑戰 2008 國家發展計劃—E 世代人才培育計畫」，邀請專家學者以趣味性、生活化的方式所編寫的系列英語教材之一，如圖 36 所示。

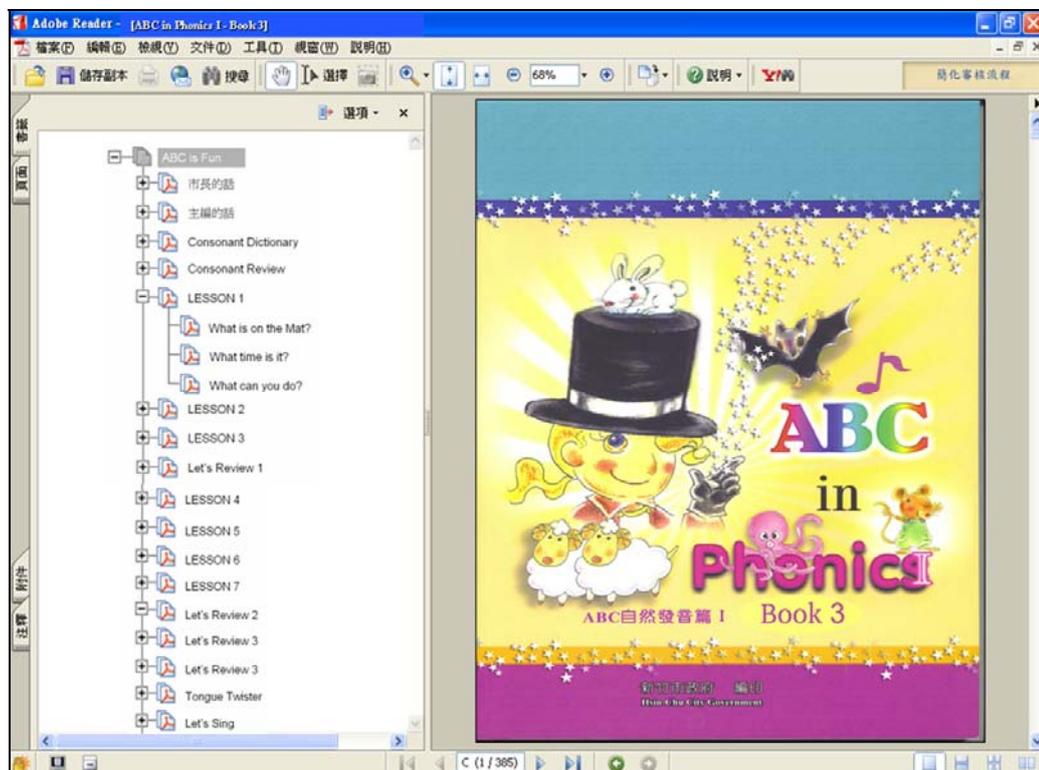


圖 36 「ABC in Phonics」

在此我們選定第三冊中的“What is on the Mat?” 章節作為 SCO 的原始教學內容，並進行多媒體教材的開發，其課程內容如圖 37 所示。

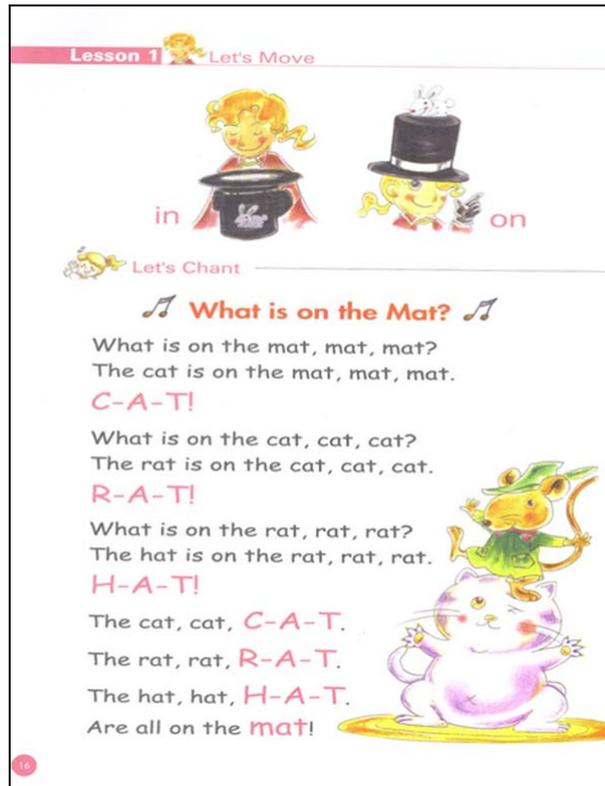
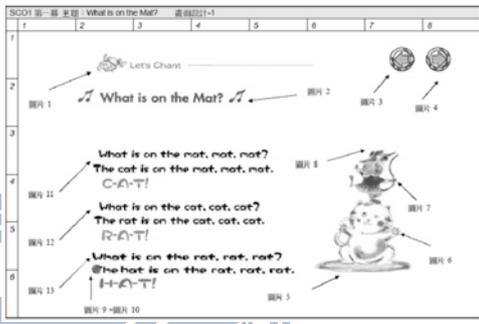


圖 37 “What is on the Mat?” 課程內容

依照多媒體教材的開發流程，匯集每一個開發階段的設計成果，就可以依序產生多媒體教材的製作過程文件。整份教材的製作過程文件摘要如表 3 所示。

表3 「ABC in Phonics」之製作過程文件

開發階段 (Phase)	製作過程文件 (Document)	項目 (Item)		
		內容摘要	數量	開發人力
教學內容設計 (RD)	課程架構文件	「ABC in Phonics」第三冊 LESSON1 之章節內容	3	1
	SCO 描述文件	“What is on the Mat?” 內容說明	1	1
單元腳本設計 (DM)	分鏡設計文件	第一幕為「What is on the mat, mat, mat?」至「H-A-T!」, 第二幕為「The cat, cat, C-A-T」至結束	2	1
	腳本設計文件	每一幕分為開場、互動與退場劇情; 描述演員的播放順序與內容	6	1
場景 UI 設計 (UI)	UI 說明文件	每一幕的場景規劃說明, 例如: 	2	1
素材製作 (MM)	素材清單 (文字)	無	0	0
	素材清單 (圖片)	列出素材內容與說明, 例如: 	28	2
	素材清單 (聲音)	列出素材內容與說明, 例如: 1. 鈴聲 2. 女生:「Page sixteen. Let's Chant. What is on the Mat?」	12	2
	素材清單 (影片)	無	0	0
	素材清單 (動畫)	無	0	0
教材製作 (CM)	教材成果檔	製作完成的多媒體教材	1	1

5.2 文件管理

這一節將介紹系統管理員如何在系統中設定各種檔案的播放方式、管理每一開發階段的文件，以及設定文件的格式、品質項目與文件間的關聯資訊。

5.2.1 檔案播放方式

- 步驟一：新增檔案類型及設定嵌入語法。

MQCS

ADMIN

- » 教材管理
- » 品質管理
- » 文件管理
- 檔案播放方式
 - 文件設定
 - 文件關聯定義
- » 系統管理

名稱: swf

說明: swf

副檔名: swf

嵌入語法: `<object type="application/x-shockwave-flash" data="{FILE}" width="200" height="200"><param name="movie" value="{FILE}"></param></object>`

播放器: 瀏覽...

Submit Cancel

Degree Program of Computer Science, National Chiao Tung University

圖 38 新增檔案播放方式

- 步驟二：完成設定。

MQCS

ADMIN

- » 教材管理
- » 品質管理
- » 文件管理
- 檔案播放方式
 - 文件設定
 - 文件關聯定義
- » 系統管理

編號	名稱
1	gif
2	htm
3	jpg
4	mp3
5	png
6	swf

View New Edit Delete

Degree Program of Computer Science, National Chiao Tung University

圖 39 列出所有檔案播放方式

5.2.2 文件設定

- 步驟一：選擇開發階段，並以素材製作（MM）階段為例新增「素材清單（圖片）」文件，以及上傳文件定義檔。

MQCS 登出

ADMIN [RD] [DM] [UI] [MM] [CM]

» 教材管理
» 品質管理
» 文件管理
- 檔案播放方式
- 文件設定
- 文件關聯定義
» 系統管理

文件名稱: 素材清單 (圖片)
文件簡介: 素材清單 (圖片)
類型:
文件定義檔: C:\upload\actor.xml
次序: 1

Degree Program of Computer Science, National Chiao Tung University

圖 40 新增文件

- 步驟二：完成文件新增，接著選擇「Set Style」設定文件格式。

MQCS 登出

ADMIN [RD] [DM] [UI] [MM] [CM]

編號	文件名稱	次序	流程
6	素材清單 (圖片)	1	4

Degree Program of Computer Science, National Chiao Tung University

圖 41 列出文件

- 步驟三：編輯文件上需要呈現的內容及設定文字格式。

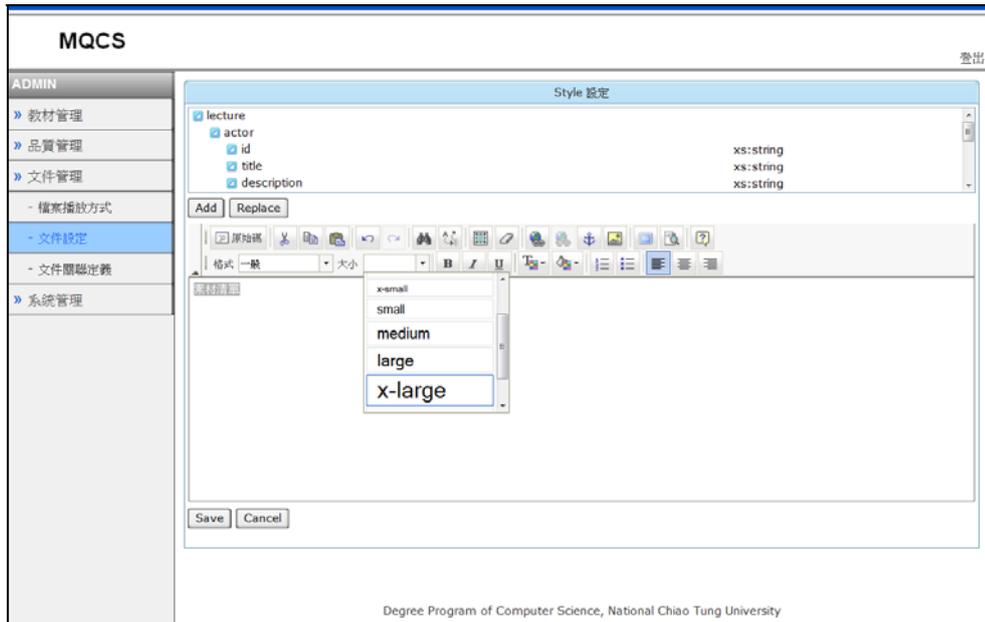


圖 42 編輯文件內容與格式

- 步驟四：依照文件定義檔編排資料的位置，首先要選擇文件定義檔列出的元素，並按「Add」進行新增。此時 Complex element 會出現 SECTION 及 TABLE 供選擇，而 Simple element 則會出現 TEXT 及 OBJECT 供選擇。

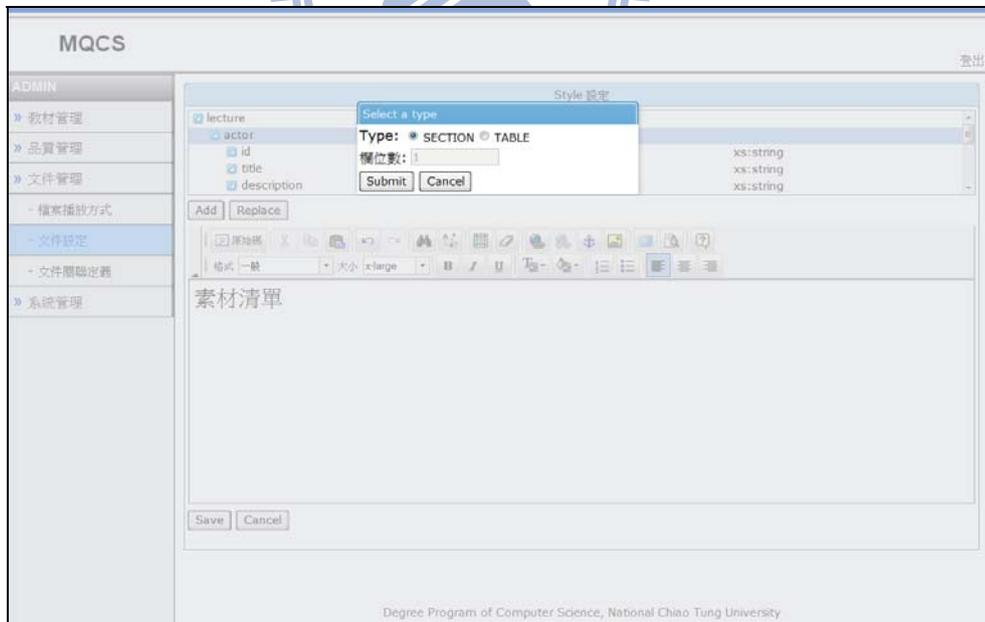


圖 43 加入 Complex element

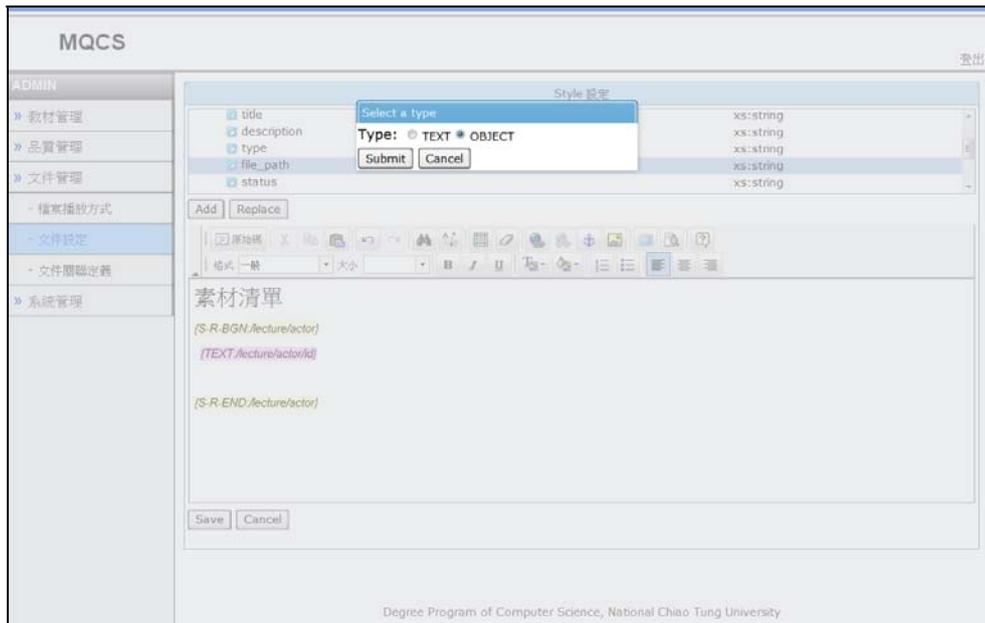


圖 44 加入 Simple element

- 步驟五：於設定完成後按「Save」即可儲存；為展現 TEXT 與 OBJECT 型態的不同，在此我們將「/lecture/actor/file_path」，分別以 OBJECT 與 TEXT 的方式呈現在文件中。

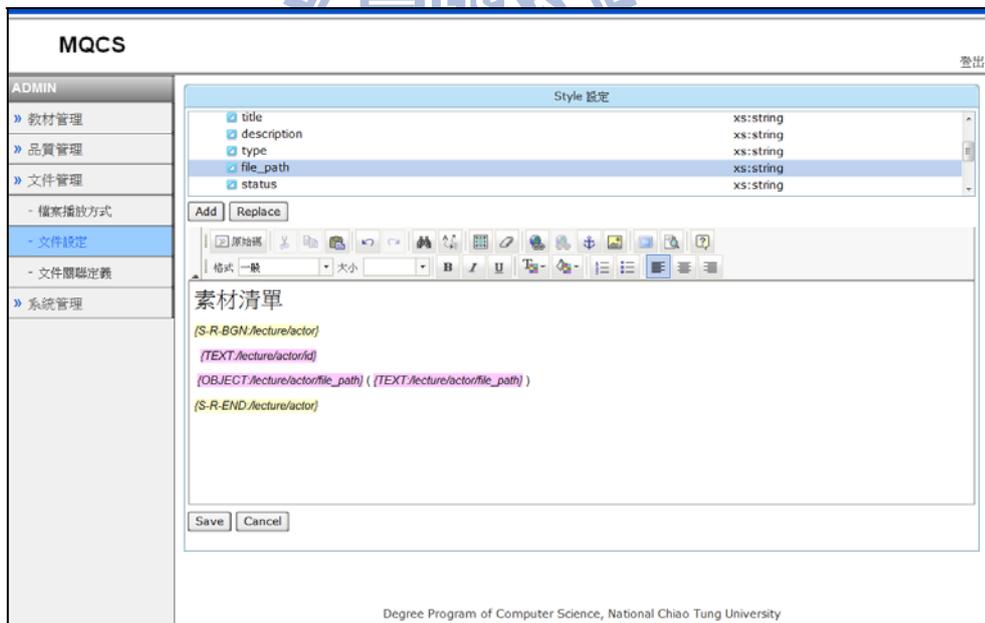


圖 45 設定文件格式

5.2.3 文件關聯定義

- 步驟一：選擇要設定關聯的文件。在這個部分，我們以「腳本設計文件」與「素材清單（圖片）」為例進行設定。



圖 46 選擇相互關聯的文件

- 步驟二：依文件定義檔選擇要設定關聯的節點，並按「Save」即可完成設定；此例中，我們設定了「腳本設計文件」中的「/lecture/script/actor/id」與「素材清單（圖片）」中的「lecture/actor/id」有關聯。

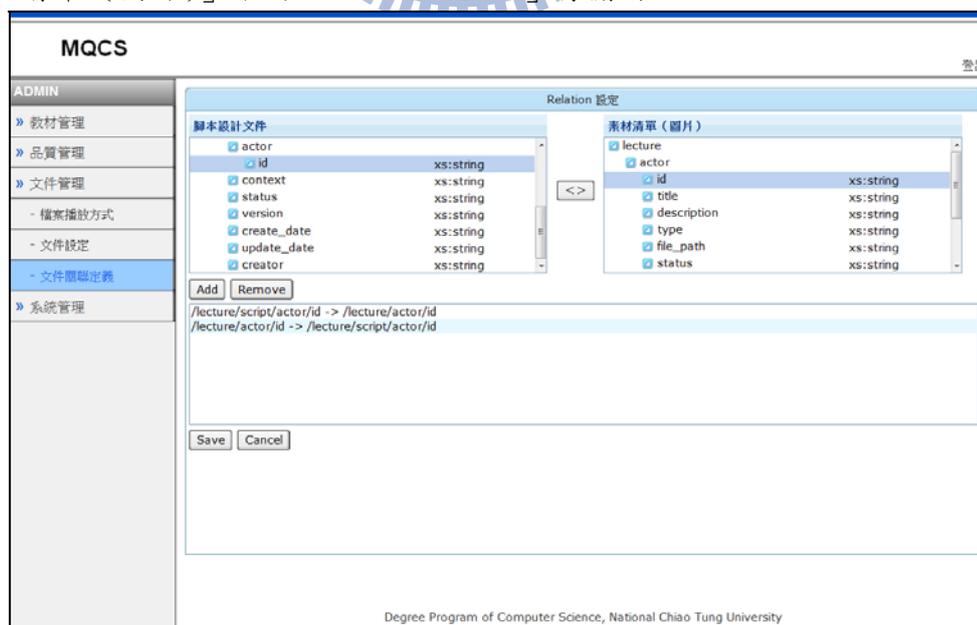


圖 47 設定文件關聯

5.3 教材管理

在此節中，我們將實際在系統中建立「ABC in Phonics」教材的資訊，及上傳每一個開發階段的資料與多媒體檔案，並對產生的製作過程文件進行線上瀏覽與追蹤，最後則輸出一份可列印的 PDF 文件。

- 步驟一：專案經理在系統中建立教材的主要資訊。

The screenshot shows the MQCS system interface. On the left is a navigation menu with options like '教材管理', '教材資訊建立', '資料及檔案上傳', '文件瀏覽與追蹤', '可列印文件輸出', and '品質管理'. The main area is a form for adding curriculum information for 'ABC in Phonics'. The form fields include:

教材名稱	ABC in Phonics
教材簡介	「ABC in Phonics」是新竹市政府配合行政院「挑戰2008國家發展計劃—E世代人才培育計畫」，邀請專家學者以趣味性、生活化的方式所編寫的系列英語教材之一。
教材主旨	新竹市國民小學英語教學
教材目的	推動英語教育
教材目標	以趣味性、生活化的方式教學
備註	
教材原始檔	C:\upload\ABC in Phonics. [瀏覽...]
狀態	

At the bottom of the form are 'Submit' and 'Cancel' buttons. The footer text reads 'Degree Program of Computer Science, National Chiao Tung University'.

圖 48 新增教材資訊

- 步驟二：專案成員依照開發流程上傳資料與壓縮後的多媒體檔案，並且可以對上傳的文件內容做預覽的動作。在 HTML 文件中，OBJECT 型態的資料內容是依照檔案播放方式的設定做呈現，但 TEXT 的資料內容則以文字的方式顯示，如圖 50 所示；而 PDF 文件中的 OBJECT 資料內容則是以連結的方式讓使用者連結到系統中進行瀏覽，如圖 51 所示。



圖 49 上傳文件資料與多媒體檔案



圖 50 預覽 HTML 文件

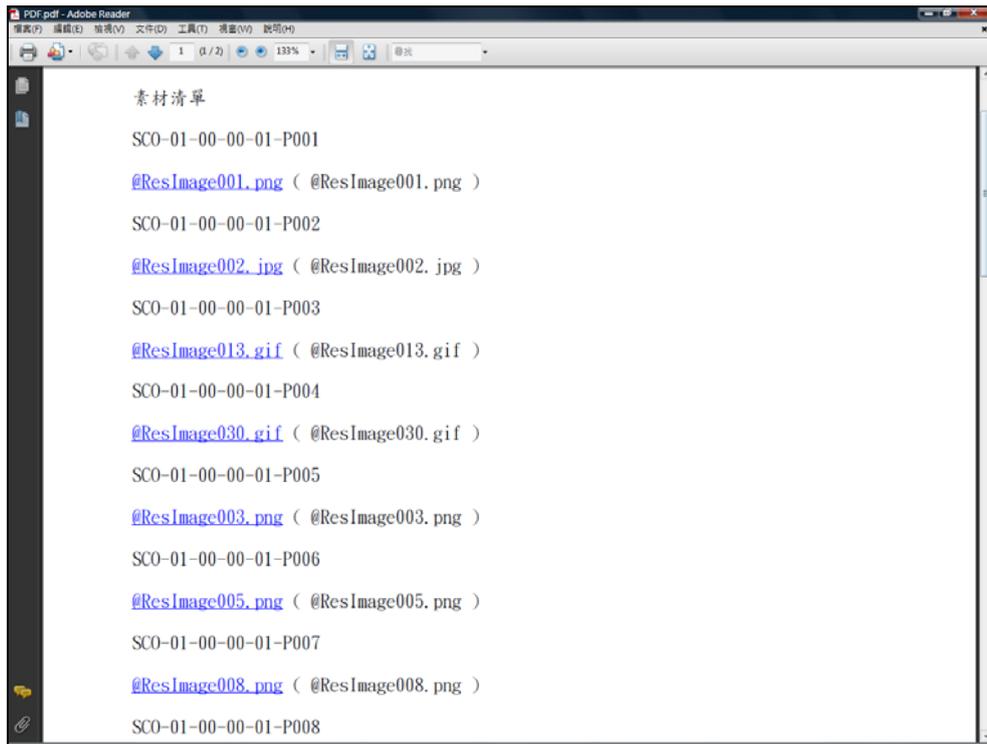


圖 51 預覽 PDF 文件

- 步驟三：專案成員選擇需要瀏覽及追蹤的文件。



圖 52 選擇要瀏覽及追蹤的文件

- 步驟四：當對資料有興趣且需要追蹤文件時，點選文件中的資料（藍色星號），再按「Trace Document」，系統即會依照文件關聯設定自動在已開啟的其他文件中，定位出相關的資料內容（紅色星號）。

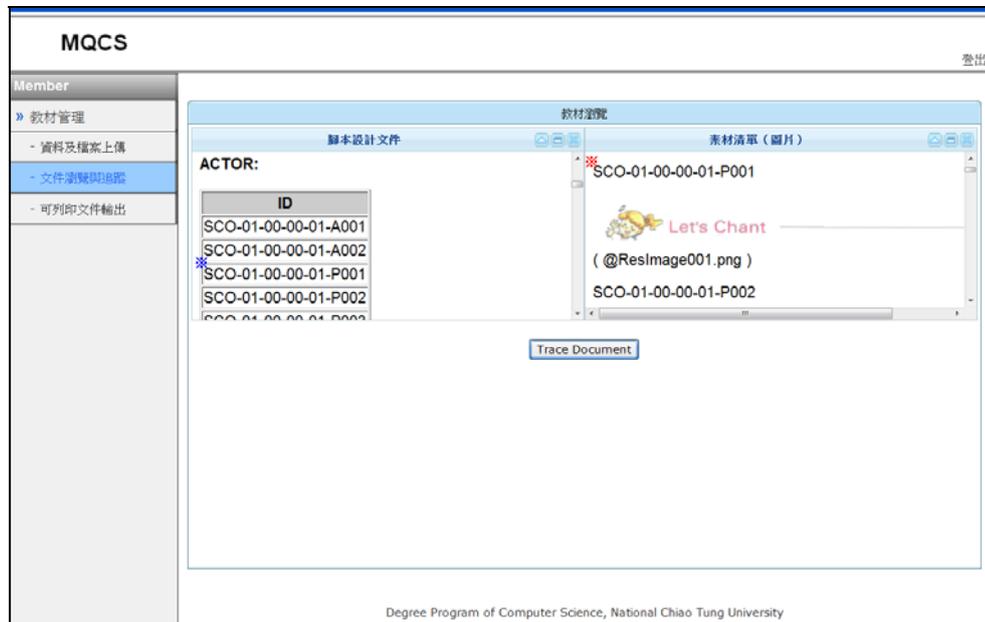


圖 53 瀏覽及追蹤文件

- 步驟五：在多媒體教材開發完成後，專案成員可下載整份製作過程文件。而系統產生的製作過程文件如圖 55 所示。

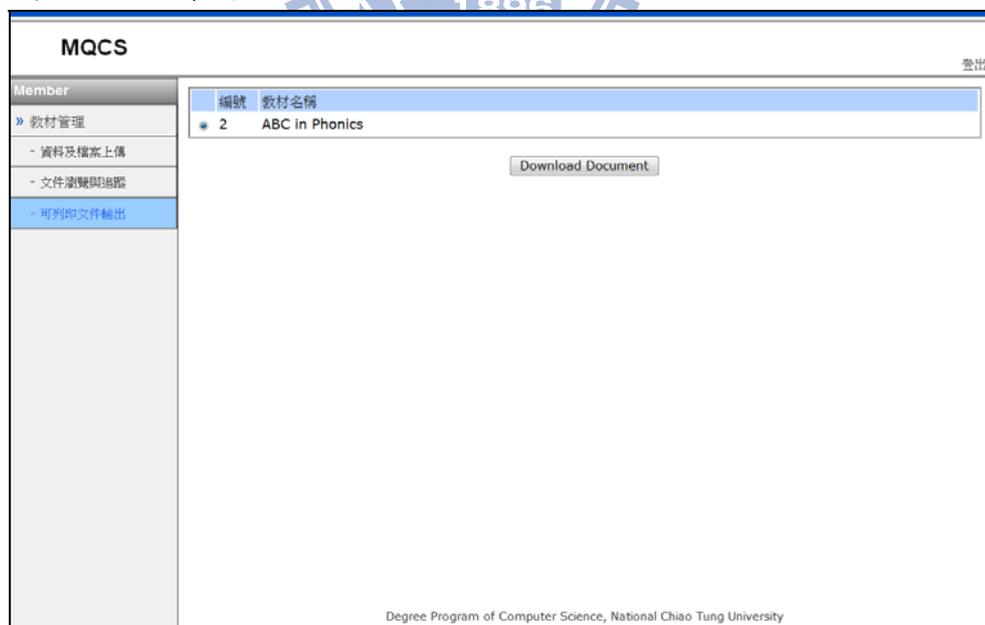


圖 54 下載文件

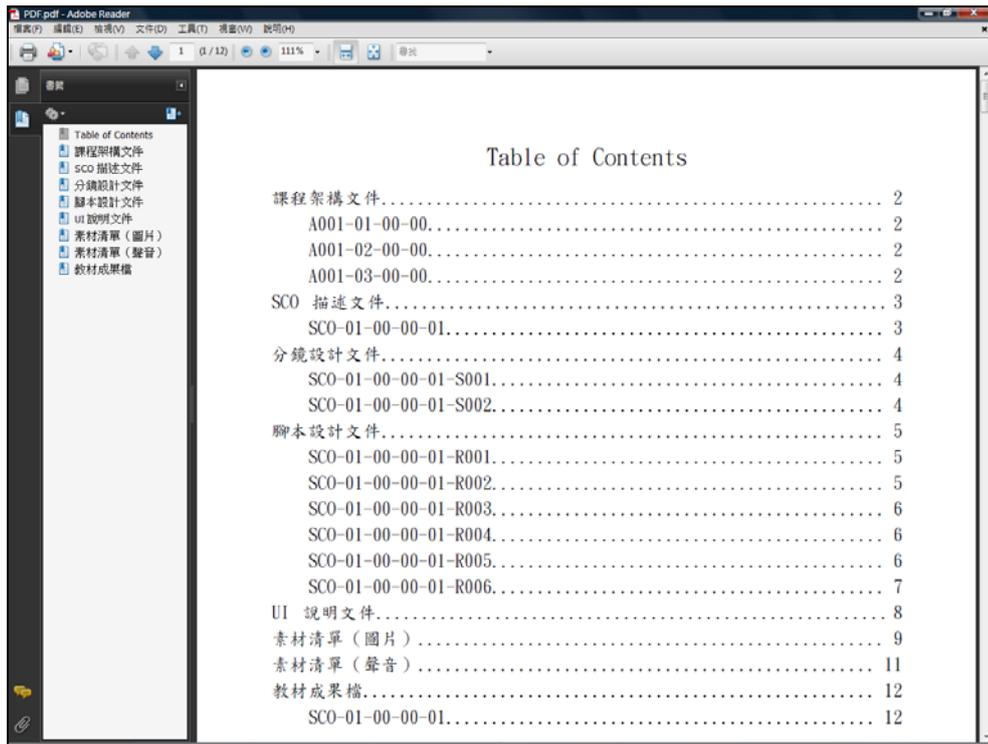


圖 55 瀏覽整份可列印文件



5.4 品質管理

最後我們將以一個品質問題，說明如何進行品質審核及提出矯正措施。

- 步驟一：品管團隊瀏覽及追蹤文件，並於發現品質問題時，加上問題描述。

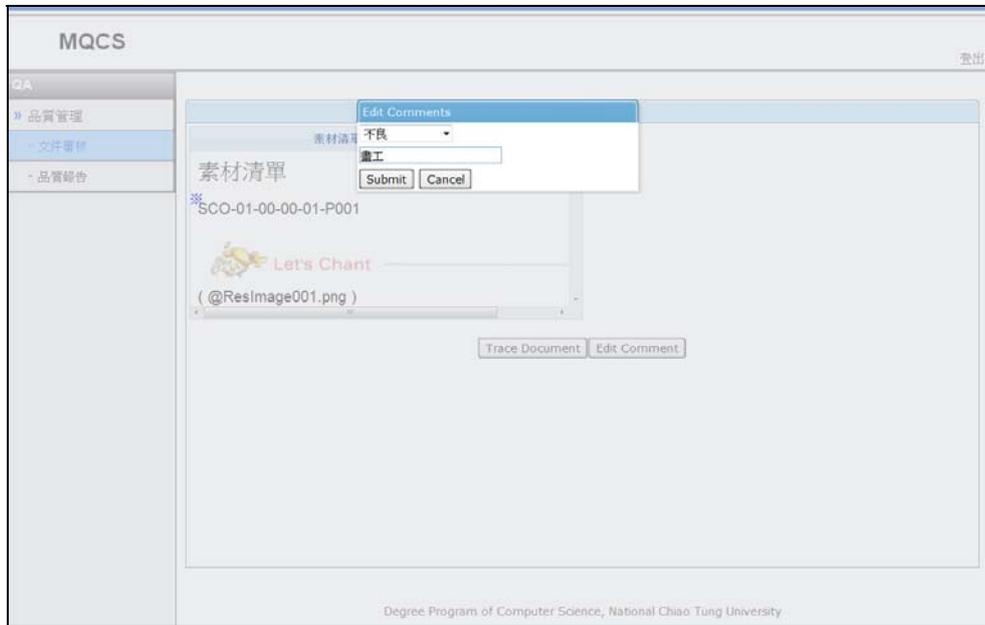


圖 56 新增品質問題



圖 57 列出品質問題

- 步驟二：專案經理提出矯正措施。



圖 58 提出矯正措施

- 步驟三：產生品質報告（品質係數設定為 0.8）。

品質報告				
教材	開發階段	文件名稱	項目數量	問題數量
ABC in Phonics 【98.07%】	教學內容設計 (RD) 【100%】	課程架構文件 【100%】	3	0
		SCO 描述文件 【100%】	1	0
		分鏡設計文件 【100%】	2	0
	單元腳本設計 (DM) 【100%】	腳本設計文件 【100%】	6	0
		場景 UI 設計 (UI) 【100%】	UI 說明文件 【100%】	2
	素材製作 (MM) 【92.27%】	素材清單 (圖片) 【92.86%】	28	2
		素材清單 (聲音) 【91.67%】	12	1

圖 59 檢視品質報告（第一次審核結果）

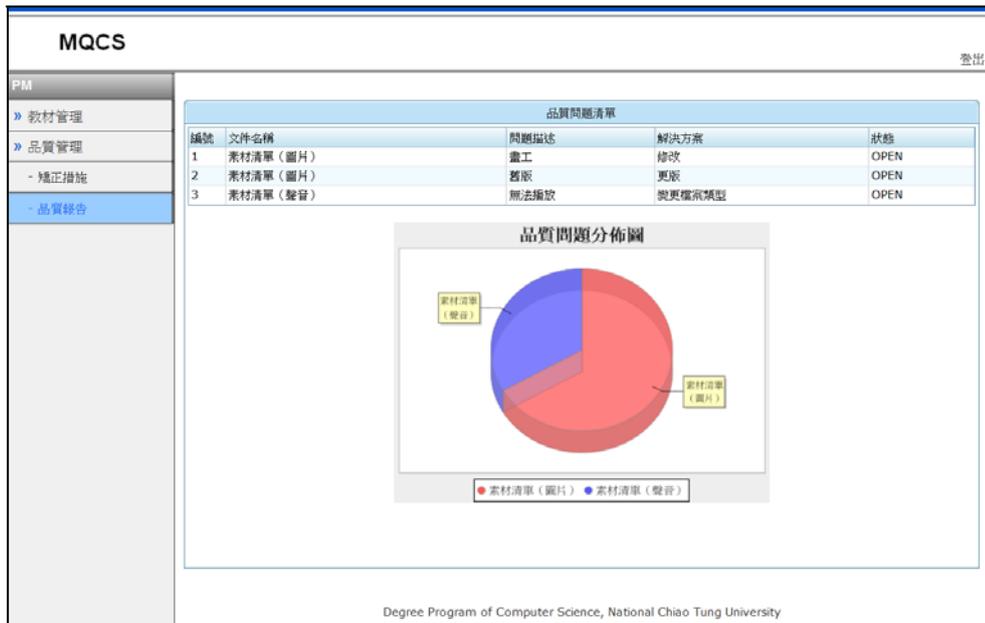


圖 60 檢視品質問題清單 (第一次審核結果)



圖 61 檢視品質報告 (第二次審核結果)

六、結論

6.1 總結

為提升多媒體教材的品質，本研究針對多媒體教材製作過程中的品質問題，提出以一套可以應用在多媒體教材製作過程的文件產生器，以及發展多媒體教材製作過程之品質管理機制與品質評估模型，來協助專案成員確認開發階段的正確性及控管多媒體教材製作過程中的品質問題。

由於多媒體教材兼具「教學」與「軟體」兩個專業領域的特質，因此我們探討了軟體品質的相關文獻與 CMMI 規範，希望運用軟體工程的方法，有效控管多媒體教材的品質問題。並且依照多媒體教材的開發生命週期定義多媒體教材的製作過程文件，及發展多媒體教材的品質狀態圖與品質評估模型，客觀地呈現多媒體教材的品質狀態。

最後則依照主要的功能需求進行分析、設計與實作多媒體教材製作過程之文件產生器與品質管理機制，並以一個教案作為實例，確認本研究的設計成果可以適用於多媒體教材的製作過程。

因此本論文在協助提升多媒體教材的品質方面，主要的貢獻如下：

(1) 透明化多媒體教材的開發過程

運用本研究設計與實作的文件產生器所產生的製作過程文件，可以在多媒體教材的製作過程中，適當地展示各種多媒體檔案的播放效果，進而協助專案成員更確切地瞭解整個多媒體教材的設計內容與開發過程。而藉由文件追蹤機制的建立，則可以協助專案成員更有效率地確認每一個開發階段的正確性。

(2) 建立品質管理機制

本研究以實作 CMMI 之 PPQA 流程領域的方式，建立多媒體教材的品質管理機制，藉由審核製作過程文件，有效地控管多媒體教材製作過程中的品質問題，並且可以持續改善多媒體教材的開發流程，提升多媒體教材製作過程中的品質。

(3) 客觀地呈現多媒體教材的品質狀態

透過本研究發展的品質狀態圖與品質評估模型，可以清楚地呈現多媒體教材製作過程中產生的品質問題，以及客觀地評估與預測多媒體教材的品質，讓專案成員可以隨時檢視多媒體教材的品質狀態，並及時修正所產生的品質問題。

6.2 未來發展方向

最後，我們提出以下可以繼續發展的方向，讓一個應用在多媒體教材的文件產生器可以更趨完善：

(1) 提供文件格式範本

由於目前在系統中需要逐一設定每一份製作過程文件的格式，因此若是可以提供文件格式的設定範本，不僅可以增進文件格式設定效率，也可以協助訂定組織的文件標準。

(2) 加強多媒體檔案的播放方式

對於各種多媒體檔案格式，目前在文件中只能以相同的播放方式作呈現，如果可以針對每一個多媒體檔案提供不同的播放方式，則更可以支援多媒體教材的多樣性。

(3) 建立進度評估模型

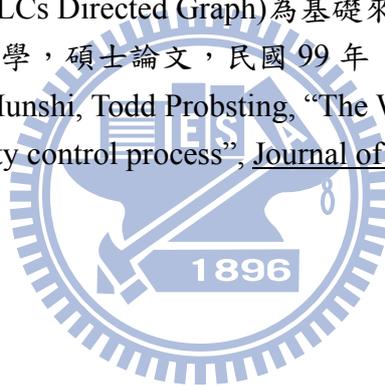
透過本研究發展的品質評估模型可以清楚地瞭解多媒體教材的品質狀態，而對於專案進度的評估，則需要另外發展多媒體教材的進度評估模型，以隨時評估及預測多媒體教材的開發進度，並協助管理多媒體教材的開發時程。



參考文獻

- [1] Advanced Distributed Learning (ADL), “SCORM 2004 4th Edition”,
<http://www.adlnet.gov>
- [2] 連瑞斌, 「以 CMMI 為基礎的多媒體學習內容開發流程品質管制方法」, 國立交通大學, 碩士論文, 民國 96 年
- [3] 數位學習品質認證中心, 「數位教材」, <http://www.elq.org.tw>
- [4] CMMI Product Team, CMMI for Development, Version 1.2, Software Engineering Institute, Carnegie Mellon University, 2006
- [5] Ian Sommerville, Software Engineering, 8th Edition, Addison-Wesley, 2007
- [6] 陳登吉, 「創意教學策略及資訊融入教學方法」, 國立交通大學多媒體教材編輯設計與實作課程講義
- [7] Tracey L. Leacock, John C. Nesbit, “A Framework for Evaluating the Quality of Multimedia Learning Resources”, Journal of Educational Technology & Society, vol. 10, no. 2, pp.44-59, 2007
- [8] Philip B. Crosby, Quality Is Free: The Art of Making Quality Certain, McGraw-Hill, 1979
- [9] Michael S. Deutsch, Ronald R. Willis, Software Quality Engineering: A Total Technical and Management Approach, Prentice-Hall, 1988
- [10] Shahida Sulaiman, Norbik Bashah Idris, Shamsul Sahibuddin, “Production and Maintenance of System Documentation: What, Why, When and How Tools Should Support the Practice”, Proceedings of the Ninth Asia-Pacific Software Engineering Conference, 2002
- [11] Christiano de Oliveira Braga, Arndt von Staa, Julio Cesar Sampaio do Prado Leite, “Documentu: A Flexible Architecture for Documentation Production Based on a Reverse-engineering Strategy”, Journal of Software Maintenance: Research and Practice, vol 10, pp. 279-303, 1998
- [12] Mira Kajko-Mattsson, “A Survey of Documentation Practice within Corrective Maintenance”, Empirical Software Engineering, vol. 10, pp. 31-55, 2005
- [13] Andrea De Lucia, et al., “Enhancing an Artefact Management System with Traceability Recovery Features”, Proceedings of the 20th IEEE International Conference on Software Maintenance, 2004
- [14] Patrick Mäder, Orlena Gotel, Ilka Philippow, “Rule-Based Maintenance of Post-Requirements Traceability Relations”, Proceedings of the 16th IEEE International Requirements Engineering Conference, 2008
- [15] Margaret K. Kulpa, Kent A. Johnson, Interpreting the CMMI: A Process Improvement Approach, Auerbach, 2003

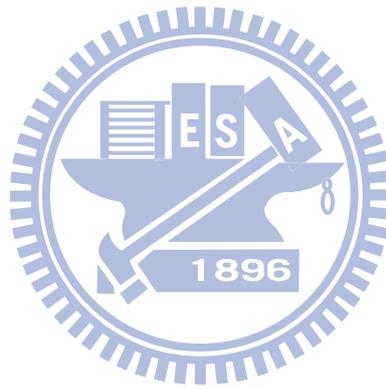
- [16] Maria Habib, et al., “Blending Six Sigma and CMMI – An Approach to Accelerate Process Improvement in SMEs”, Proceedings of the 12th IEEE International Multitopic Conference, 2008
- [17] Dennis M. Ahern, Aaron Clouse, Richard Turner, CMMI Distilled: A Practical Introduction to Integrated Process Improvement, Addison-Wesley, 2001
- [18] Mary Beth Chrissis, Mike Konrad, Sandy Shrum, CMMI: Guidelines for Process Integration and Product Improvement, 2nd Edition, Addison-Wesley, 2006
- [19] Mike Phillips, “CMMI V1.1 and Appraisal Tutorial”, Software Engineering Institute, Carnegie Mellon University, 2004
- [20] Advanced Distributed Learning (ADL), SCORM 2004 4th Edition Content Aggregation Model (CAM) Version 1.1, 2009
- [21] 張錦成, 「符合 CMMI 需求管理流程領域之多媒體教材內容製作品質管制系統」, 國立交通大學, 碩士論文, 民國 98 年
- [22] 陳信江, 「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」, 國立交通大學, 碩士論文, 民國 98 年
- [23] 李仲智, 「以 MDG (MLCs Directed Graph) 為基礎來尋求一多媒體教材素材的品質管制方法」, 國立交通大學, 碩士論文, 民國 99 年
- [24] Pankaj Jalote, Rajesh Munshi, Todd Probsting, “The When-Who-How analysis of defects for improving the quality control process”, Journal of Systems and Software, vol. 80, pp. 584-589, 2007



附錄

A 「ABC in Phonics」授權函

本論文第五章應用實例展示的「ABC in Phonics」英語教材是由新竹市政府所編印，基於智慧財產權，因此已先取得新竹市政府授權作為學術用途使用。申請及同意授權的函文內容如下：



(1) 交通大學申請授權函

國立交通大學 函

機關地址：30010 新竹市大學路1001號

聯絡電話：03-5712121分機56601

傳 真：03-5721490

電子郵件：yolanda@cs.nctu.edu.tw

聯絡人：楊耀萱

裝 受文者：

發文日期：中華民國99年4月30日

發文字號：交大資工字第0991004014號

速別：普通件

密等及解密條件或保密期限：普通

附件：

訂 主旨：請同意授權本校資訊工程學系陳登吉教授及其指導學生使用 貴府自編英語
教材「ABC in Phonics」為論文研究之應用實例，請 查照惠復。

說明：

- 一、本校資訊工程學系陳登吉教授及其指導學生陳詩雯（學號：9679513）及李仲智（學號：9679537）因論文研究所需，請同意授權使用 貴府自編之英語教材「ABC in Phonics」做為論文研究之應用實例。
- 二、論文研究內容：運用軟體製作流程規範，設計與實作應用於多媒體教材製作之品質管制方法，透過控管多媒體教材的開發流程，提升多媒體教材的品質。
- 三、擬使用 貴府自編英語教材「ABC in Phonics」第三冊中「What is on the Mat?」章節之內容，實際說明將教材內容製作成多媒體教材的過程及展示如何使用論文中實作之多媒體教材品質管制系統。

線

正本：新竹市政府

副本：本校資訊工程學系、本校資訊學院碩士在職專班

(2) 新竹市政府同意授權函

新竹市政府 函

機關地址：30051新竹市中正路120號

承辦人：楊慧瑜

電話：03-5216121#273

電子信箱：05172@ems.hccg.gov.tw

受文者：國立交通大學

發文日期：中華民國99年5月4日

發文字號：府教學字第0990046737號

速別：普通件

密等及解密條件或保密期限：普通

附件：

主旨：本府同意授權 貴校資訊工程學系陳登吉教授及其指導學生陳師雯、李仲智

使用本府自編英語教材「ABC in Phonics」為論文研究之應用實例，請

查照。

說明：

一、復 貴校99年4月30日交大資工字第0991004014號函。

二、本府同意僅限學術研究使用，請勿作為其他用途使用。

正本：國立交通大學

副本：本府教育處

99/05/05
08:38:51

(3) 新竹市政府姓名更正函

新竹市政府 函

機關地址：30051新竹市中正路120號

承辦人：楊慧瑜

電話：03-5216121#273

電子信箱：05172@ems.hccg.gov.tw

受文者：國立交通大學

發文日期：中華民國99年5月18日

發文字號：府教學字第0990052958號

速別：速件

密等及解密條件或保密期限：普通

附件：

主旨：有關本府同意授權 貴校資訊工程學系陳登吉教授及其指導學生陳詩雯、李仲智使用本府自編英語教材「ABC in Phonics」為論文研究之應用實例，姓名誤繕更正乙案，請 查照。

說明：

- 一、本府99年5月4日府教學字第0990046737號函諒達。
- 二、本府同意授權 貴校資訊工程學系陳登吉教授及其指導學生陳詩雯、李仲智使用本府自編英語教材「ABC in Phonics」為論文研究之應用實例，其中學生姓名陳詩雯誤繕為陳師雯，特此更正。

正本：國立交通大學

副本：本府教育處

99/05/19
08:09:03