

# 一個支援 QTI 規格之評量批閱系統的設計

## Design of a Scoring System for Supporting QTI Specifications

劉建宏 李益誠 陳淑玲\*

國立台北科技大學 資訊工程學系, 南台科技大學 管理與資訊學系\*

E-mail{cliu, s3598006}@ntut.edu.tw, slchen@mail.stut.edu.tw\*

### 摘要

隨著網際網路的普及與發達, 數位學習(e-learning)逐漸受到重視, 也成為教育學習領域未來的趨勢。由於測驗是評量學生學習狀況的一個重要方法, 因此線上評量系統便成為數位學習中一個重要的工具。目前被廣泛採用的 QTI(Question and Test Interoperability)試題規格支援多種的題型與自動批閱的樣版, 但沒有定義對於問答或申論題型其人工批閱的相關規範。有鑑於此, 本論文提出一個支援 QTI 規格之評量批閱系統的設計, 除了整合 QTI 既有的試題自動答覆處理功能, 也提供了人工批閱工具, 讓教師能在網頁上直接做類似於傳統批閱的動作, 並可呈現回饋的圖形與文字, 以提高支援 QTI 規格之評量批閱系統的應用。

**關鍵詞:** QTI, 數位學習, 線上測驗, 線上人工閱卷。

## 1 前言

隨著網際網路的蓬勃發展, 促進了數位學習與遠距教學等學習趨勢, 各級學校或教學訓練單位無不致力於建構此類教學平台, 提供數位化教學內容來輔助課堂上的教學, 以提升教學品質與學習成效, 讓學生能享有另一種學習管道與提升自我能力的方式。由於測驗是評量學生學習狀況的一個重要方法, 線上評量系統的發展便成為數位學習中一個重要的課題。透過線上評量的進行, 除了可以檢視學生學習的成果, 了解學生學習的情況, 還能作為往後老師教學的依據與參考, 進而輔助學習的進行與促進學習的效率。因此一個可提供評量傳送、試卷和試題編輯、批閱試卷與測驗管理等功能之線上評量系統, 便成為發展數位學習的一個重要工具。

目前有許多的評量系統被提出 [1, 3, 4], 這些系統可以評量學習成效和輔助數位學習, 不過多數的系統僅提供是非與選擇題型, 缺少問答題型的支援, 或是沒有採用合乎國際標準的試題規範。因此在評量的實際應用與題庫的交換共享上有所限制。而 QTI[9]是目前國際廣泛採用的評量規範之一, 目的在於促進試卷與試題的共享。QTI 不但定義試卷與試題的組成架構, 亦規範評量答覆處理的機制來控制試題批閱的流程。此外, QTI 支援包含問答題在內等多種題型, 並提供內建的答覆處理機

制樣版, 試題可直接套用這些樣版來進行自動批閱的動作。雖然答覆處理機制樣版可以提供試題批閱之流程控制的彈性。然而這些樣版主要適用於答案簡單的題型, 如是非題和選擇題。對於答案繁複的題型, 如問答題或申論題, QTI 尚未加以規範, 人工批閱的方式仍有其需要。因此, 若能整合自動答覆處理機制與線上人工批閱的功能, 將有助於促進 QTI 規格的評量系統的應用。

QTI 相關的研究與評量系統的實做目前已有許多。K. Johnson[11]等人比較了數種試題的規格, 包括 GIFT、Aiken、Missing Word、Blackboard、WebCT、Course Test Manager 與 QTI 等, 並認為 QTI 規格具有完整性與相容性等優勢, 能符合大多數評量系統的需求。R. Garcia-Robles[5]等人根據 QTI Lite[10]設計與實做了 QAed 系統, 該系統能提供試題編輯、預覽、匯入、匯出、與搜尋等功能; 同時隱藏複雜的 QTI 規格, 讓使用者不需要學習 QTI 規範。E. G. Pacurar[6]等人提出一個整合 QTI player 與 QTI editor 的網路平台, 稱為 netUniversit é。主要用來幫助具有基礎網站開發技巧的教師能設計出自己的教學網站。此外, 在考慮符合經濟效益的情況下, H. Vogten[14]等人提出 CCSI (CopperCore Service Integration)的架構, 用以將學習設計引擎(Learning Design Engine)與其他如 QTI 引擎、討論區、搜尋引擎等學習服務整合在一起, 使得與數位學習有關的系統能夠有效地利用彼此的資源, 以結合成為功能更完整的系統。而在線上人工閱卷方面, 許政穆[2]等人提出一個 Web-based 線上閱卷系統。該系統使用 VML[13]的技術將批閱之文字與符號, 以疊加的方式覆蓋在原始作答內容上, 並利用 CSS 和 HTML 的<DIV>標籤來固定批閱內容與作答內容之相對位置, 以避免因為瀏覽解析度的不同, 造成不正確的檢視結果。

在這篇論文中, 我們提出一個 QTI 試題評分系統。此系統除了支援 QTI 規格所定義的自動答覆處理機制外, 也整合人工批閱的方法以支援問答題型的批閱, 讓線上評量系統對於數位學習的支援更為完整。在自動批閱部分, 我們主要實作一個評量傳送引擎以支援 QTI 內建的答覆處理機制樣版。在人工批閱部分, 我們利用 SVG(Scalable Vector Graphics)[12]的技術, 讓教師可以在網頁上直接進行人工批閱的動作, 並可呈現類似於傳統測驗的批閱結果與回饋內容。

本篇論文的組織如下：第二部分對 QTI 的試題規格以及測驗流程與試題中基本標籤之間的關係做一個簡單的介紹。第三部分介紹整個測驗與批閱試卷的流程。第四部分說明自動批閱的設計，包含了 QTI 答覆處理樣版的評分方式。第五部分介紹整個閱卷系統之設計，包含人工批閱的流程、人工批閱樣版的設計、SVG 圖形與 HTML 文字的呈現方法、以及自動批閱與人工批閱試題的整合。第六部分展示本系統主要的功能與操作流程。最後一部分對本系統做總結，以及提出未來可能的研究方向。

## 2 QTI 試題與評分處理的簡介

### 2.1 QTI

IMS(Instructional Management Systems)[7]是目前對數位學習規範投入最深，也最積極的單位，其致力於發展推廣網路學習活動的開放技術標準，QTI 是 IMS 為分享測驗試題與評估工具而定的規範，並用 XML 語言來描述試題與測驗，其目的在使不同單位所製作的試題或試卷，能夠動態地組合而再利用，讓試題或試卷可以提供給不同評量系統(assessment systems)使用，節省製作試題或試卷的時間和金錢。

在 QTI 的規範裡，試卷和試題之間的關係是定義在其 ASI(Assessment、Section、Item)的資訊模型(information model)中。基本上一個題組(section)包含一個以上的試題(item or assessment item)或是其他題組，一個試卷零件(testPart)包含一個以上的題組，一個試卷(assessment)則由一個以上的試卷零件所構成。而不同的試題與題組可以整合成一個題庫(item bank)，以便於共享。

### 2.2 Item

在 QTI Version 2.1 標準中，一個試題主要包含有(1)答覆宣告(response declaration)一定義解答與答覆相關的資訊變數，並儲存考生之作答內容以供答覆處理機制使用；(2)結果宣告(outcome declaration)一定義評量結果與暫存資訊變數，用來儲存自動批閱的結果與試題狀態的改變，通常藉由狀態的改變來決定回饋元素(feedback elements)與 modal 回饋(modal feedbacks)的顯示與否；(3)試題主體(item body)一定義題目內容、考生互動方式(interactions)，選項(choices)、與回饋元素等資訊；(4)答覆處理機制(response processing)一定義答覆和評分(scoring)處理所需的法則(rule)，評量傳送引擎通常會根據這些法則去修改結果宣告定義之變數，更新得分或試題狀態，提供下一階段試題回饋的呈現；和(5)modal 回饋一定義經過答覆處理機制運作之後欲顯示的回饋內容，與回饋元素不同的地方在於，modal 回饋內不能包含考生互動方式。

如圖 1 所示為一 QTI 試題的 XML 範例。該試題為一單選題，題目使用了 XHTML 標籤呈現一張

圖片(sign.png)，和<choiceInteraction>互動方式詢問考生此圖片代表之意義，且有三個選項供考生選擇。而且定義了正確答案為"ChoiceA"，以及使用了 QTI 內建之答覆處理機制樣版：Match Correct，以供自動批閱處理；Match Correct 其意義為當考生選擇的選項識別字(identifier)與正確答案定義之識別字相同時，表示此題答對，可得 1 分，否則表示此題答錯，得分為 0。



圖 1 QTI 試題 XML 範例

### 2.3 Response Processing

Response Processing 旨在批閱考生之作答以及計算該試題之得分。當考生送出作答後，評量傳送引擎根據答覆處理機制的定義來判斷作答內容之正確性，一般來說最後會將該試題的得分儲存至 SCORE 之結果變數中。如果該試題有定義 Modal Feedback，則在結束答覆的處理後將 Modal Feedback 的內容呈現給考生觀看。另外，答覆處理機制如同一般的程式語言，具有變數、運算子、函式、以及程式流程的控制等特性。

## 3 QTI 試題評分系統流程之設計

為了簡化 QTI 試題自動與人工批閱功能的整合，我們將自動與人工批閱的流程分開處理。基本上 QTI 的問答題型並不會定義其答覆處理機制，因此我們可以透過試題的答覆處理機制定義來決定是否需要進行自動或人工的批閱。

圖 2 為測驗與批閱試卷的流程設計。在測驗的過程中，考生每送出一道試題的作答，評量傳送引擎需要判斷此試題是否定義其答覆處理機制，若有則進行自動批閱的動作，接著將作答內容與批閱的結果與得分儲存下來，否則直接儲存作答內容。考生完成整份試卷的作答後，教師可以進入查詢系統，檢閱試卷目前的批閱情況，同時可點選需要人工批閱之試題以進行閱卷動作。在人工批閱的過程中，教師使用閱卷工具列在作答文字上註記，並輸入此試題的得分。完成批閱後，儲存人工批閱結果

與得分。

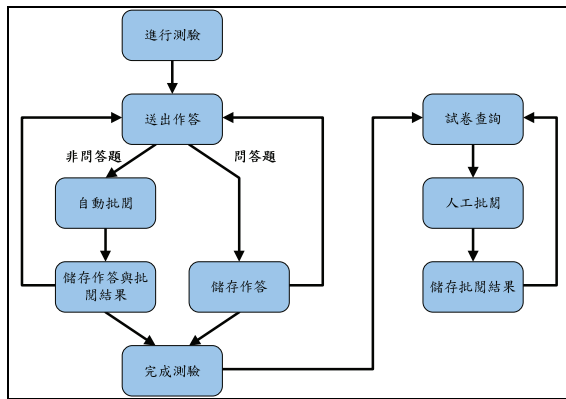


圖 2 測驗與批閱試卷的流程

#### 4 自動批閱試卷的設計

在自動批閱部分，當考生送出一道試題的作答後，評量傳送引擎會執行答覆處理機制的規則（該試題有定義答覆處理機制時），根據這些規則來決定該試題的得分與回饋內容；我們直接使用 QTI 提供之答覆處理機制樣版來作為自動批閱試題之規則。如圖 3 所示為 QTI 定義之互動、答覆處理、與 modal feedback 三個階段之間的流程關係。

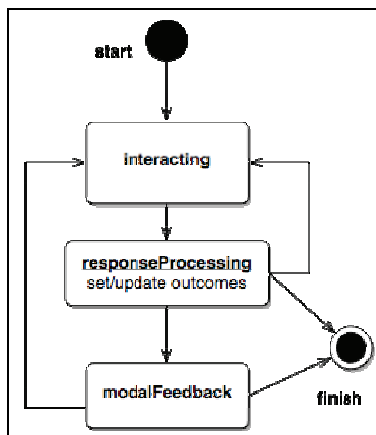


圖 3 QTI 試題答覆處理的流程 (摘自 [8])

start 為一道試題的開始，首先評量傳送引擎會將試題的初始內容呈現給考生，接著進入 interacting 階段；在每一次的作答嘗試中，當考生將作答內容送出(submit)後，會進入 responseProcessing 階段，評量傳送引擎先將該作答內容儲存至對應之答覆宣告的變數中，接著從頭執行一次在答覆處理機制定義之答覆處理規則，在此過程中會去更動到結果變數的值，以提供總分的統計與回饋的使用。完成答覆處理後，如果試題有定義 Modal Feedback，則會進入 modalFeedback 的階段，根據 Modal Feedback 所參考到的結果變數來決定回饋的顯示與否；否則

完成本道試題的作答或是回到 interacting 階段，進行下一次的嘗試作答。

圖 4 為本系統中評量傳送引擎在處理每一份試卷的工作流程，流程分為 Pre-processing、Interaction、Response Processing 與 Modal Feedback 四個階段，後面三者分別對應至圖 3 的三個階段。在 Pre-processing 階段，評量傳送引擎首先會根據測驗的定義擷取出試卷的相關資訊，接著根據試卷定義之選題策略做一個試題擷取的動作，表示從資料庫中取得該試題的資訊，包含完整的 QTI XML 內容。接著宣告一個 Response Processor 物件，存放在 session scope 中，在作答與評估的過程中重複使用此物件，直到評量傳送引擎傳送新的試題時，才從 server 中移除之，目的是要保存每次答覆處理所更新的結果變數內容。透過 XMLBeans[15]將試題的 XML 內容文字轉換為 XMLBeans 定義的試題物件，接著將試題物件傳入 ResponseProcessor 物件中轉換為更精簡且適合用來做答覆處理機制的物件。而在完成試題的呈現後，評量傳送引擎會等待考生的作答，直到接收到答覆後才進入答覆的處理階段。完成答覆的處理與呈現回饋後，如果此道試題已經完成作答，系統會儲存考生的作答內容與批閱結果，接著擷取出下一道試題。

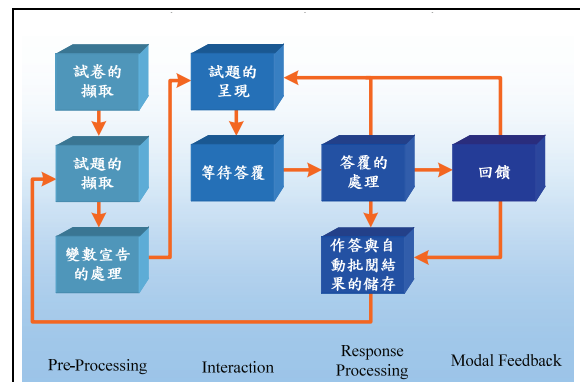


圖 4 評量傳送引擎之工作流程

QTI 提供了三種答覆處理機制的樣版，一般除了問答題型的試題均可直接套用之。三種樣版的說明與使用方法如下：

#### ■ Match Correct

此種樣版適合用於具有單一答案與非完全答對即為完全答錯之特性等試題，如：是非題、單選題。其意義為使用 QTI 定義之 match 運算子去判斷答覆變數 RESPONSE 的值，與 RESPONSE 事先定義之正確答案(correctResponse)是否相同；如果相同表示此題答對，接著將 SCORE 變數內容設為 1，否則將 SCORE 變數內容設為 0。由此可知，使用此樣版必須在答覆宣告處宣告 RESPONSE 變數與定義 RESPONSE 之正確答案，同時在結果宣告處定義 SCORE 變數。另外不論 RESPONSE 有沒有設值，此樣版都會給予 SCORE 變數一個值。如圖 5 所示

為 Match Correct 樣版內容。

```
<?xml version="1.0" encoding="UTF-8" ?>
<responseProcessing xmlns="http://www.imsglobal.org/xsd/imsqti_v2p0"
  xmlns: xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.imsglobal.org/xsd/imsqti_v2p0_imsqti_v2p0.xsd">
  <responseCondition>
  <responseIf>
  <match>
  <variable identifier="RESPONSE" />
  <correct identifier="RESPONSE" />
  </match>
  <setOutcomeValue identifier="SCORE">
  <baseValue baseType="integer">1</baseValue>
  </setOutcomeValue>
  </responseIf>
  <responseElse>
  <setOutcomeValue identifier="SCORE">
  <baseValue baseType="integer">0</baseValue>
  </setOutcomeValue>
  </responseElse>
  </responseCondition>
</responseProcessing>
```

圖 5 Match Correct 樣版內容

## ■ Map Response

此種樣版適用於具有多個答案與部分得分特性試題，如：複選題、配合題、填充題。其意義為使用 QTI 定義之 *mapResponse* 運算子去判斷答覆變數 *RESPONSE* 的值，與 *RESPONSE* 事先定義之 *mapping* 結構之對應關係。當遇到像複選題這種需要去判斷每個選項是否正確以及每個選項的得分時，我們就需要去定義一個 *mapping* 的結構。*mapping* 結構中包含一個以上的 *mapEntry*，每個 *mapEntry* 定義了一個鍵 *mapKey* 以及對應的值 *mappedValue*，如圖 6 所示為一個 *mapping* 的範例。

```
<mapping lowerBound="0" upperBound="2"
  defaultValue="-2">
  <mapEntry mapKey="H" mappedValue="1" />
  <mapEntry mapKey="O" mappedValue="1" />
  <mapEntry mapKey="CI" mappedValue="-1" />
</mapping>
```

圖 6 mapping 範例

以複選題來說這個 *mapEntry* 表示當選擇了一個識別字為“H”之選項時，可得到“1”分。同時 *mapping* 提供了 *lowerBound* 與 *upperBound* 兩個屬性去控制最後累計值的範圍。由此可知，使用此樣版必須在答覆宣告處宣告 *RESPONSE* 變數與定義 *RESPONSE* 之正確答案與 *mapping* 結構，同時在結果宣告處定義 *SCORE* 變數。另外當 *RESPONSE* 為 NULL 時，此樣版會將 *SCORE* 值設為 0。

## ■ Map Response Point

此種樣版同樣適用於具有多個答案與部分得分特性試題，與 Map Response 樣版不同處在於支援圖形式作答的題型，例如在地圖上指出台灣的位置。另外與 Map Response 不同處在於需定義 *areaMapping* 的結構，在 *areaMapping* 中包含一個以上的 *areaMapEntry*，*areaMapEntry* 定義了形狀 *shape*，描述形狀位置與大小的座標 *coords* 與對應值 *mappedValue*，如圖 7 所示為一個 *areaMapping*

的範例。

```
<areaMapping defaultValue="0">
  <areaMapEntry shape="circle"
    coords="11,18,12" mappedValue="1" />
  <areaMapEntry shape="circle"
    coords="15,23,12" mappedValue="1" />
  <areaMapEntry shape="circle"
    coords="96,114,12" mappedValue="1" />
</areaMapping>
```

圖 7 areaMapping 範例

當考生選擇的點座標位於圖 7 中三個圓形之一的範圍內時，可得到 1 分。同樣地，使用此樣版必須在答覆宣告處宣告 *RESPONSE* 變數與定義 *RESPONSE* 之正確答案與 *areaMapping* 結構，同時在結果宣告處定義 *SCORE* 變數。

## 5 人工線上批閱試卷的設計

### 5.1 線上批閱的問題

線上批閱主要有三個挑戰：(1)需要在網頁上顯示批閱的文字與圖形，我們不修改原先作答內容，而是使用疊加的方式在作答文字上呈現圖形；(2)批閱的結果在批閱者端與受測驗者查詢端呈現的效果要相同，不能因為使用者使用各種瀏覽器不同的字體、字型大小等設定造成不同的呈現效果，可能會使得批閱者的批閱內容文不對題；(3)自動批閱與人工批閱試題的整合。第一與第二個問題我們使用 SVG 與 CSS 技術來解決，另外提出了自動批閱與人工批閱之試題整合的機制，讓人工閱卷系統能很輕易地加入原先的評量傳送引擎中，使得線上評量系統在題型與閱卷的支援上更為完整。

利用 SVG 實作線上閱卷的功能，基本上需要完成兩部分：一個 Editor 和一個 Viewer。Editor 主要是提供一個幾何圖形（線段、矩形、圓形、橢圓）和文字編輯的工具，讓使用者能在 HTML 網頁上進行圖形的新增、刪除、移動與編輯屬性並產生對應的 SVG 敘述，同時要提供輸入試題得分的欄位。另外為了方便老師批閱，應提供打勾、畫叉、問號與箭頭等常用的符號。而 Viewer 主要是將 Editor 所產生的 SVG 敘述寫入 HTML 網頁中對應的位置，並使得原先 HTML 網頁的文字內容不受影響。值得注意的是，由於不同的瀏覽器可能有不同的網頁呈現(rendering)設定，而使得 SVG 圖文和 HTML 網頁內容的相對位置和大小受到影響，產生位移而造成錯誤。因此我們需要結合 CSS 或 XSL 來固定網頁的字型、大小、寬度等瀏覽器 rendering 的設定，讓老師和學生在不同的瀏覽器下能看到相同批閱結果。因此 Editor 也包含了 Viewer 的功能。

SVG 是 W3C 所公佈一種描繪向量圖形的 XML

標記語言，它可以在瀏覽器上輕易地畫出向量幾何圖形和文字，並具有漸層色彩、濾鏡效果、動畫、樣式功能、和使用者互動的能力。由於 SVG 可以結合 Java Script、DHTML (Dynamic HTML)、Cascading Style Sheets(CSS)、和 XSL Style Sheet，透過 Java Script 及 DHTML 程式的控制和 CSS(或 XSL)的排版，我們可以將 SVG 的幾何圖形和文字動態地重疊到純文字的 HTML 網頁上，使得老師的評量回饋和學生回答問題的答案能呈現在一起，而達到與傳統紙上閱卷相同的效果。

## 5.2 線上批閱樣版的設計

為了方便人工閱卷的初始化動作，我們建立一個空白的 SVG 文件樣版。如圖 8 所示，此 SVG 文件中包含了事件處理，初始化，畫布，狀態列，及物件選取外框。

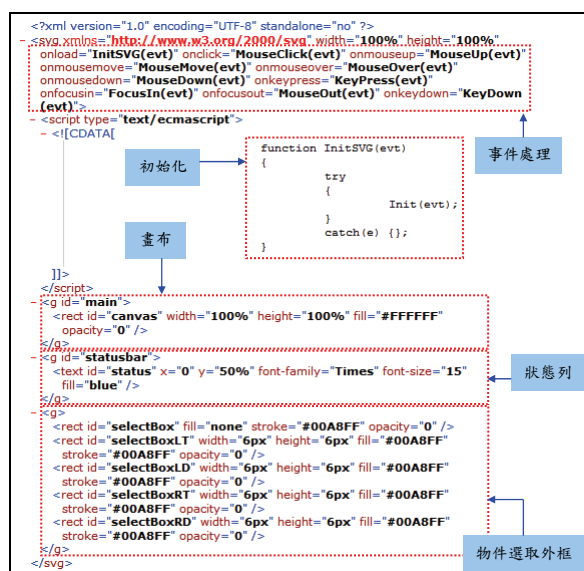


圖 8 線上批閱樣版的內容

以下說明這個樣版包含的詳細內容之設計。

### ■ 事件處理

包含了滑鼠點擊、滑鼠移動、鍵盤輸入與 SVG 文件初始化的事件處理，每個事件都會呼叫外部 JavaScript 的函式。

### ■ 初始化

呼叫 JavaScript 的初始化函式，用來初始化閱卷工具列屬性，如文字大小，文字字體，外框顏色與寬度，填滿顏色，透明度等設定。

### ■ 畫布

一個群組元素，在畫面上呈現的所有圖形與文字元素都包含在此群組中。在結束人工閱卷的動作時，僅將此群組的 SVG 內容儲存在資料庫中，而不儲存此樣版其餘的 SVG 元素；當下次要做批閱動作時，先將空白樣版中畫布的內容清空，再將資料庫中畫布的內容加入到此空白樣版的畫布群組

中。在畫布中還有一個與 SVG 文件同樣大小的透明矩形，用來防止閱卷的過程中接收到下層 HTML 的事件。

### ■ 狀態列

使用一個文字元素來顯示訊息，可提供除錯或提示使用者的功能。

### ■ 物件選取外框

定義了一個選取物件時顯示在物件外圍的選取外框，包含了一個無填滿區域的矩形及四個角落方塊。

## 5.3 人工批閱試卷的流程

由於評量的過程是將試題採逐題呈現的方式，試題的順序則由系統隨機決定，以利受測者作答。在本系統中，除了問答或作文等文字作答的互動方式外，其餘的題型均能使用答覆處理機制來做自動批閱，而沒有定義答覆處理機制的題型，為方便人工閱卷，本系統必須將這些試題彙整，且結合上述 SVG 幾何圖形和文字的編輯工具，製作成一個網頁，以利閱卷的進行。在批閱完畢後，將試題得分及相關的 SVG 圖文存回資料庫。當學生查閱評量結果時，再將 SVG 圖形和評量回饋等資料與原來的試卷整合，以利學生檢視成績和瞭解自己需要更正之處。如圖 9 所示為本系統線上閱卷的架構，為了便於儲存管理與整合，SVG 標記圖文與原始試卷是分開儲存的，也使得閱卷者無法更改受測者的作答。



圖 9 線上閱卷之架構

## 5.4 作答內容與 SVG 圖形的呈現方式

在 HTML 的文件中欲插入 SVG 文件需要利用 `<embed/>` 標籤，如：

```
<embed src="empty.svg" width="600" height="300" wmode="transparent">
```

在此處屬性 src 指定 SVG 文件的來源路徑，屬

性 width 與 height 則分別指定 SVG 圖形的寬度與高度，而屬性 wmode="transparent" 則表示此文件的背景為 100% 透明；如果不使用 transparent 值，即使是空白的 SVG 文件，仍會有預設的白色背景而完全遮住在底下的 HTML 文件內容。

如圖 10 所示，我們欲將考生作答內容與 SVG 的圖形以疊加的方式呈現，首先將給予考生輸入作答文字的 HTML 標籤 <textarea> 與 SVG 圖形嵌入標籤 <embed/> 分別放入兩個 <div/> 標籤中，透過 CSS 指定兩個 <div/> 標籤的座標位置為相同，並指定 <textarea> 與 <embed/> 標籤的大小為相同，使其作答內容與 SVG 圖形的區域恰好重疊在一起，接著使用屬性 z-index 指定兩個 <div/> 標籤的在 Z 軸上的順序（即上下順序），同時使用 CSS 語法固定 <textarea> 標籤內容文字的大小，避免使用者更動網頁字型的大小造成考生作答內容與 SVG 圖形位置不相符的情況。

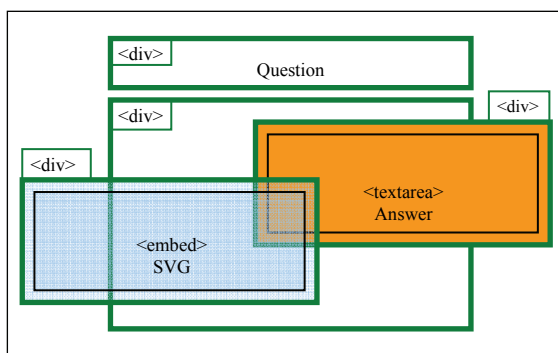


圖 10 作答內容與 SVG 呈現方式示意圖

## 6 範例說明

在本節以一個實際的例子，針對考生完成整份試卷的測驗過程後，說明本批閱系統的使用。

當考生送出每道試題之後，評量傳送引擎會判斷此試題是否可進行自動批閱之動作，若是則運行其答覆處理樣版定義之答覆處理規格，計算試題的得分，並將作答內容與批閱結果儲存起來；否則直接將作答內容儲存起來。直到考生完成整份試卷作答後，閱卷者方能進行試卷查詢的動作。如圖 11 所示為試卷查詢的畫面，分別為兩道選擇題的呈現與自動批閱的結果與得分；第三題為一道作文試題，由於尚未進行人工批閱的動作，因此畫面上只呈現考生作答的內容，當閱卷者進行批閱之後，可在作答文字上看到人工批閱後的註記圖案與文字。

接著說明人工閱卷之流程。

1. **選擇試卷**：從測驗群組與科目中選擇某位考生之試卷。
2. **選擇試題**：系統會將考生的整份試卷列出，接著使用在問答題下方之「人工批閱試題」按鈕進入批閱畫面。

3. **進行人工批閱**：如圖 12 所示，使用畫面上方之閱卷工具列來進行螢光筆，畫外框線，畫圓，打勾，加入註記文字等功能。同時可分別設定框線與填滿區域之顏色與透明度；另外在畫面上畫出的任何物件都可隨意移動與刪除。
4. **輸入得分**：最後輸入此道試題的得分。
5. **送出批閱結果**。



圖 11 試卷作答與批閱結果查詢



圖 12 進行人工批閱

## 7 結語與未來研究方向

在這篇論文中我們提出一個支援 QTI 規格之評量批閱系統的設計，以提高 QTI 評量批閱系統的應用。此系統整合 QTI 自動答覆處理機制與線上人工批閱的功能。當考生完成測驗後，我們根據 QTI 內建的答覆處理機制樣版來自動批閱試題；而文字作答的題型則使用本系統設計的批閱工具做類似傳統批閱的動作。透過 SVG 的技術，我們可以疊加的方式在試題作答內容上，呈現人工批閱的文字與圖形。

關於未來的研究方向，在自動閱卷的部分，目前本系統只實做出 Match Correct 與 Map Response 兩種樣版的批閱，未來將完整支援 Map Response Point 樣版。而 QTI 提供之答覆處理樣版在特殊的情況下仍不敷使用，例如不支援題目答錯時的倒扣處理，因此本系統在未來也將提供答覆處理樣版的

編輯功能。另外在人工閱卷的部分，線上批閱工具雖然提供各種基本幾何圖形的繪製，例如：線段、矩形、橢圓等，但應提供一般人工批閱工作中常用的符號，如：問號、打勾、打叉、打圈、箭頭、插入符號(^)、與刪除線等工具，讓批閱者能更快速地完成批閱動作。

## 參考文獻

- [1] 涂嘉壽，一個具回饋效果的線上評量系統之設計與實作，碩士論文，國立中山大學資訊工程學系，2003.
- [2] 許政穆，林建丞，「應用於網路教學的Web-based線上閱卷工具」，2001年台灣區網際網路研討會(TANET 2001)，pp. 215-219，2001。
- [3] 陳新豐，林邦傑，余民寧，“線上題庫系統之研發，”2002年台灣區網際網路研討會(TANET 2002)，pp. 426-430，2002.
- [4] 陳澄曲，互動式教學評量系統之研究與實作，碩士論文，國立清華大學資訊工程學系，2002.
- [5] R. Garcia-Robles, J. Blat, S. Sayago, D. Griffiths, F. Casado and J. Martinez, “Supporting Usability and Reusability Based on eLearning Standards,” Proceedings of the IEEE International Conference on Advanced Learning Technologies, 2004.
- [6] Ecaterina Giacomini Pacurar, Philippe Trigano, Sorin Alupoaie. “A QTI editor integrated into the netUniversité web portal using IMS LD”, Journal of Interactive Media in Education (JIME), retrieved October 25, 2005.
- [7] Instructional Management Systems (IMS). <http://www.imsglobal.org/>.
- [8] IMS Global Learning Consortium, Inc., IMS Question and Test Interoperability Information Model, version 2.1 Public Draft Specification, 2006.
- [9] IMS Question and Test Interoperability(QTI). <http://www.imsglobal.org/question/index.html>.
- [10] IMS QTILite. <http://www.imsglobal.org/question/#version1.2lite>.
- [11] Kevin Johnson, Timothy Hall and Derek O’Keeffe, “Generation of Quiz Objects (QO) with a Quiz Engine Developer (QED)”, IEEE International Workshop on Wireless and Mobile Technologies in Education, 2005.
- [12] SVG 1.1 Specification. <http://www.w3.org/TR/SVG/>.
- [13] Vector Markup Language (VML). <http://www.w3.org/TR/NOTE-VML>.
- [14] Hubert Vogten, Harrie Martens, Rob Nadolski, Colin Tattersall, Peter van Rosmalen and Rob Koper, “CopperCore Service Integration; Integrating IMS Learning Design and IMS Question and Test Interoperability”, ICALT, 2006.
- [15] XMLBeans. <http://xmlbeans.apache.org/>.