

模擬程式執行教材開發與練習編輯工具之 設計與實作

孫鵬飛

元智大學資訊工程所

s956051@mail.yzu.edu.tw

周志岳

元智大學資訊工程所

cychou@saturn.yzu.edu.tw

摘要

對於資訊相關科系的學生，甚至是一般欲接觸程式設計的人來說，學習程式設計是一種生活中很少遭遇的學習情境。而在整個學習過程中，學習者更是容易因為觀念不清，不斷的在嘗試中錯誤，又從錯誤中學習，也就是試誤學習(trial-and-error learning)的發生。為了有效提昇學生的學習效率，本研究開發了一套編輯器(authoring tool)型式的電腦輔助學習系統(CAL system)，讓使用者能模擬程式執行的過程並且編輯程式執行的步驟以及每一步驟中變數的資料內容。本系統亦提供教師編輯教材的工具，教師所編輯的內容視作專家的行為並以此與初學者所編輯之內容做分析比較。除此之外，更架設一線上交流學習平台，供學習者之間做非同步式交流，讓學習者不僅能藉由電腦輔助，更藉由同儕之間相互交流，全面提昇學習效果。

關鍵詞：程式設計、編輯器、電腦輔助學習系統

Abstract

For the students major in Computer Science and other people who want to learn program design, it is a little life experience learning environment for learning programming. In the whole process of learning, the learner is easy in the attempt to keep the mistakes, because the concept is unclear. Learning from mistakes, which is called trial-and-error. In order to effectively enhance the learning efficiency of students. The researcher develops a type of computer-assisted learning system like authoring tool, so that users can simulate the process of the implementation of program and

editing data of each step. The system also provides teachers a tool for editing materials, teachers editorial content and act as experts and beginners to edit the content of the analysis and comparison. Moreover, it set up on the frontline exchanges learning platform for students to do between non-synchronous AC, learners are not only learning by computer-assisted system, but through peer interaction to enhance the learning effects.

Keywords: Programming, Authoring Tool, CAL system

1. 研究背景與動機

學習程式設計一向是電腦科學最基礎也最難以入門的地方，初學者往往就在這個關口打退堂鼓，學習一門外國語言對大部分人來說都是很困難的，更何況是學習一門與電腦溝通的人工語言。許多學者已經針對程式設計入門這個議題做了許多研究，從傳統的學習程式設計時所遭遇到的困難，到以電腦軟體作為輔助學習的工具。目前為止，已有許多增進程式設計學習的系統產生，相信隨著研究不斷的深入，即使不懂程式語言的人也能輕鬆製作程式的時代將很快來臨。

1.1 電腦化輔助學習

編輯器即為一種軟體工具，它能让教材設計者、教育學家、老師以及學習者去設計一個互動的多媒體學習環境，而且不必具備程式語言的相關知識[11]。編輯器就是假設在沒有程式設計師和設計者缺乏或者指具備少許程式設計能力的情況下，可以發展和設計教學應用軟體[7]。

本研究應用的電腦輔助工具即為編輯器，使用者不但能將編輯器所提供的內容製作成教學教材，編輯完成後還能給於學習者互動

學習的關係，此系統不但可以解決教學者在教授程式設計前不了解學習者程度的困擾，也可藉此釐清學習者的概念，以便降低學習迷失的問題。

1.2 程式設計學習

學習程式設計是電腦相關科系最基礎且最重要的課題之一。一般程式設計課程的教學目標，主要在於告訴學生什麼是程式、如何去設計程式以及培養學生具備解決問題的觀念[10]，Dalbey和Linn在1985年提出了以「認知鏈」(Chain of Cognitive)的架構來了解程式設計所需的認知需求，其中包含了三個表示程式設計可能的學習結果：(1)程式語言的特性；(2)程式設計的技能 and (3)一般解決問題的能力。Dalbey 和Linn也一致認為程式設計最明顯的學習結果通常只有程式語言的特性和功能性知識的獲得，而程式設計的技能 and 解決問題的能力的學習成果則較不明顯，大部分的初學者都較不容易達到這兩個階段。

程式設計思考與教學的意義在於：

1. 視程式設計為資訊專業的一環：這種能力的養成可以幫助資訊產業發展原創性的軟體內容。
2. 視程式設計為資訊素養的一環：程式設計固然是專業，但其實對大多數的使用者而言，雖不必去寫程式，但是基礎程式設計概念，例如記憶位址，流向控制等基礎應用還是可以列為資訊素養的一部份。更重要的是這種訓練也可以再一次機會對邏輯思考進行訓練[1]。

最近幾年，國內對於程式設計教學方面的研究有多個方向，其中包括教學方法之探討，例如教師以多媒體演示概念模型及學習者自行操作多媒體概念模式之不同的教學呈現方式，對程式語言學習成效之影響。另外，也有一些研究專注於程式設計中的某些特殊課題的學習系統，如遞迴程式設計學習系統之發展。

在探討學習程式語言對於變數之概念發展的認知情形方面，依據Costa[6]所提出模式中的六個階段，包括

1. 產生與聚焦（找到一個變數名稱）
2. 賦予類別（變數的屬性）
3. 提出關係（推論相關資訊）
4. 應用概念網路（從已知推論未知）
5. 預測效果（擬定應用計畫）
6. 規劃實施步驟（將變數置入程式碼中），可以

應用到變數使用的歷程中。就概念發展的前兩階段，本研究利用電腦輔助工具增進學習者對於變數概念清楚的程度。

對於一般學習程式語言的學生而言，學習成效不佳可以歸類為以下三個原因：

1. 學生缺乏電腦的概念模型
 2. 教師太強調語法的教學，忽略問題解決方法
 3. 使用不合適的學習工具
- 在學習程式語言的過程中，教學策略與學習工具均會影響學生的學習成效[4]。

心理學家桑代克(Edward L. Thorndike 1874-1949)的學習理論說到，學習是一連串刺激與反應聯結行程的過程。每個刺激與反應聯結都是經由不斷的嘗試而建立的，這種學習方式也就是試誤學習(trial-and-error learning)[2]。

目前已有許多程式設計學習的相關研究，但大多是基於理論上的探討，少有實務上的研究。一般的教學系統多是強調在教授的部份，讓學生讀取教材後，再做試題測試，與傳統教學法無異，學生在學習程式設計時，最常發生的就是概念無法釐清，容易產生邏輯上的錯誤，因為不了解程式的執行過程，而寫出錯誤的程式碼，一再的發生試誤學習，故若能讓學生模擬程式的執行過程，就能有效解決這一部份的障礙，以增進學習效能。

2. 研究目的

綜觀目前有關程式設計學習方面的研究，主要的研究議題大致分為兩個方向，其一是討論學習程式設計時所遭遇的瓶頸，其二是開發可以輔助並促進程式設計學習的系統，這兩種方向都對提昇學習效能有所幫助。

在促進學習效果上，運用專家與初學者之比較，探討出專業內涵的內容、形式、與形成過程，然後以此教導初學者，使其獲得專業內涵的捷徑，為一有效的教學方法[8]。也就是說，經由與專家行為的比較，可以節省初學者許多試誤學習的時間，加速在思考或操作上的突破。本研究擬將初學者對於學習程式設計中變數應用的邏輯思考加以紀錄，並分析比較其中的內涵，找出容易產生的錯誤概念，然後透過適當的教育方法，讓學習程式設計更有效率。基於上述之背景，本研究讓學生模擬程式執行，逐行分析程式碼內容，並以此建構一套電腦輔助系統讓學生編寫程式執行過程，在整個過程中觀察學生發生試誤學習的情況，並研究改善的方法。

我們假設透過模擬程式執行方式，可以讓學生有效熟悉變數應用的概念，並深入了解程式到底做了什麼，在這樣的假設為前提之下，本研究希望達到的目的如下：

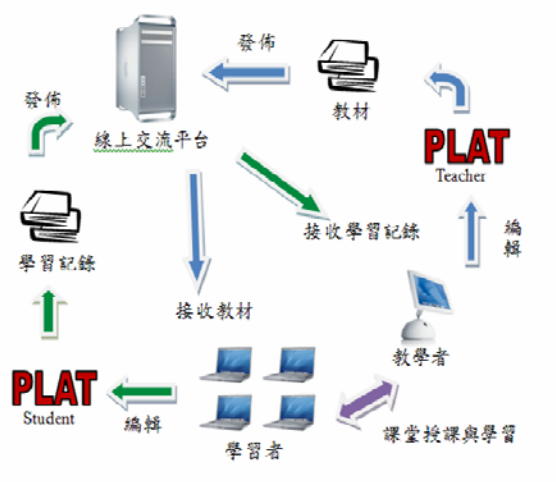
1. 以模擬程式執行過程的方法，讓學生學習程式設計的實際成效如何？
2. 觀察並紀錄學生在使用 PLAT(Program Learning Author Tool)系統練習時的行為與結果，藉由與專家的行為做比對，分析其與學習成效的關聯性。
3. 在 PLAT 系統加入自動比對功能，讓系統能利用教師的答案來偵測學生的答案，達到自動判斷與修正的目的。

3.PLAT(Program Learning Author Tool)系統介紹

本研究欲發展一套軟體系統，用於輔助程式設計學習，供進行實驗的學生使用，學習者使用本程式來學習程式設計，能更有效的熟練與清楚變數的使用與變數值的變化，程式後端也會紀錄相關的編輯紀錄。另外，本程式也可提供教學者作為教材編輯之用。

3.1 開發環境

PLAT 系統採用 .net framework 2.0 作為開發平台，實作的程式語言為 C#語言，建立在 Windows XP 的作業系統上，系統為單機板架構，應用程式分為教師用教材開發編輯器與學生用模擬練習編輯器，單純執行程式時無須網路環境，但系統提供了線上交流平台，作為學生端之系統紀錄檔和教師端之教材檔交流之用。其系統架構如圖一所示。



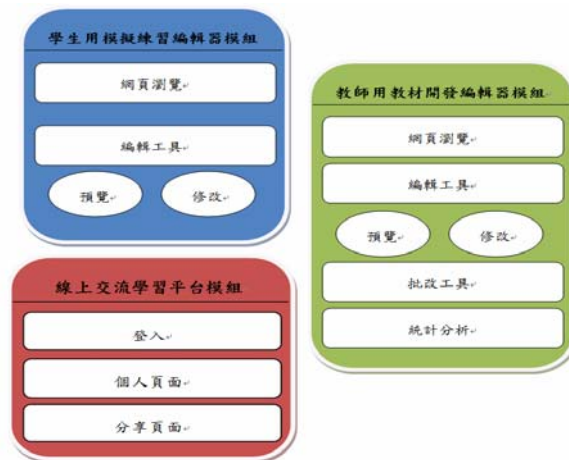
圖一.系統架構圖

3.2 介面設計

PLAT 系統包含學生用模擬練習編輯器、教師用教材開發編輯器以及線上交流平台三個應用程式，本系統以模擬程式執行過程的方式，讓學生熟悉變數和變數值的變換以及程式執行的步驟，學生用模擬練習編輯器依據這個概念，設計成讓學生依照程式執行的步驟，編輯每一步驟中所包含的變數與其變數值，並觀察學生是否了解正常的程式執行過程。

除了上述特點之外，系統的另外一項特點即是應用程式分為教師用與學生用，教師用教材開發編輯器主要功能在編輯教材供學生練習使用，另外也可批改學生的學習記錄，整套系統的運作包含教師與學生，教師使用程式編輯教材，將教材置於線上交流平台，學生可經由網路自行下載教材，再使用練習編輯器來模擬程式執行流程，並將學習記錄上傳到線上交流平台，接收到的資料可提供教師作為教學的參考依據。

根據以上兩項設計理念，本計畫希望學生透過操作模擬練習編輯器來了解程式執行的過程以及變數的應用，以預期學生使用正確的學習工具與學習方法，達到提升學習動機與學習成效的結果，並輔以增加教師參與的程度，讓每個學生的模擬程式執行過程的結果，都可以讓教師取得，用作分析學生的先備知識程度、學習成效以及日後教學進度和教材編輯的依據，從傳統教學方面也給予一定程度的學習成效提昇。本系統各個子系統分別詳述於 3.3 節(學生用模擬練習編輯器)、3.4 節(教師用教材開發編輯器)、3.5 節(線上交流平台)。PLAT 系統各模組功能如圖二所示。



圖二. PLAT 系統各模組功能

3.3 學生用模擬練習編輯器

本計畫所設計的學生用模擬練習編輯器，主要對象為剛接觸程式設計的初學者，學生使用模擬練習編輯器來扮演原本是電腦執行程式的工作，並且編輯程式執行的步驟與其中的變數資料，透過這樣的學習過程讓學生熟悉程式的運作方式。

一般而言，學習者在初學程式設計時，最大的問題往往不在於對語法的不熟悉，而是對程式執行過程的概念不夠清楚，進而在簡單的邏輯錯誤中打轉，費時費力卻學習成效不彰。所以，本計畫的模擬練習編輯器可以讓學習者在學習初期就以模擬程式執行方式，透過編輯程式執行每一個步驟的資料，熟悉每一行程式碼有哪些變數，是什麼資料型態，並計算出最後的變數值。學習者在經過整套的程式執行流程練習之後，會更加釐清對於程式執行過程的概念，避免浪費無謂的時間在處理邏輯錯誤的問題上。

學生用模擬練習編輯器的兩項主要功能為「網頁瀏覽」和「編輯工具」，內容詳述如下：

(一) 網頁瀏覽

學習者進入系統之後的預設畫面即為「網頁瀏覽」功能，程式會自動連上線上交流平台也就是元智大學的「個人 Portal」頁面，學習者可以利用「網頁瀏覽」功能進入線上交流平台，所有課程的相關資訊都會公佈在平台中，並提供學習者接收練習題目以及發佈學習記錄。

(二) 編輯工具

「編輯工具」為學生用模擬練習編輯器的核心功能，學生使用「編輯工具」編輯程式的執行過程，並填寫每一步驟中的變數資料，學習者首先要開啟一道練習程式，然後依據程式碼的執行順序新增步驟，並編輯程式執行內容，包含變數名稱、資料型態、變數值、程式註解和輸入與輸出畫面，編輯好的程式步驟會顯示在畫面左側的預覽視窗。除此之外，編輯工具也提供修正功能，可以直接在編輯好的步驟上做修改，以及全步驟預覽功能，在全部的步驟都編輯好之後，讓學習者可以一次觀察全部的程式執行步驟。

3.4 教師用教材開發編輯器

教師用教材開發編輯器，主要是為了讓教

學者編輯學習用的教材，並且提供能夠批改學習者編輯的程式執行步驟的功能，給予學習者指導與回應，最後紀錄所有學習者的學習成績加以分析，用做以後教學教材與學習進度的評估。

教師用教材開發編輯器的三項主要功能為「網頁瀏覽」、「編輯工具」和「批改工具」，由於「網頁瀏覽」和「編輯工具」的功能與學生用模擬練習編輯器大致相同，以下將只對「批改工具」的內容詳述：

(一) 網頁瀏覽

此功能與學生用模擬練習編輯器提供的完全相同。

(二) 編輯工具

基本上的功能與學生用模擬練習編輯器相同，但多出題目編輯以及選項編輯兩項功能，提供教師編輯程式碼教材使用。

(三) 批改工具

「批改工具」主要在協助教師更有效率的查看以及分析學生的學習記錄，在批改工具介面中會顯示教師所預先編輯的正確程式執行步驟，並與學生的程式執行步驟的做比對，而教師可以對學習記錄的每一個程式執行步驟進行對錯的判斷，並寫下錯誤原因，預期未來將功能提昇為初步的自動批改，讓程式挑出與預設正確步驟不一樣之處，再由教師來定奪。另外「批改工具」也有簡單的統計功能，計算該學習者錯誤及正確的步驟數，預期未來將功能提昇為統計圖表，顯示各步驟的對錯人數，以利教師作為教學的參考依據。

4. PLAT(Program Learning Author Tool)系統實作

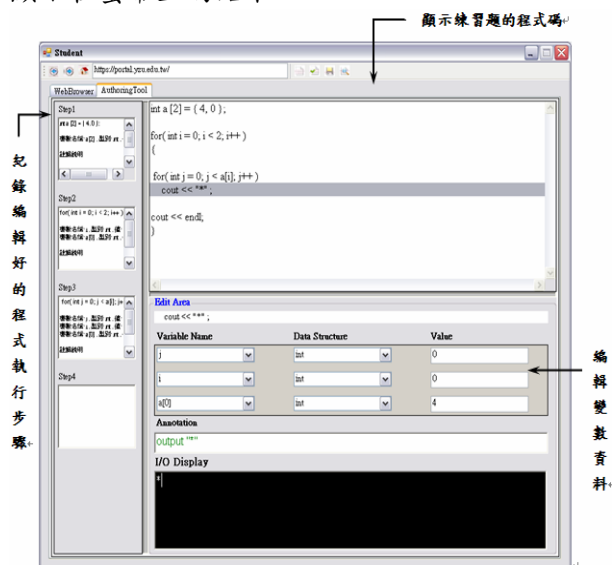
PLAT 系統為程式設計提供新的學習方向，讓學習者自行追蹤程式執行流程，此外，與教學者保持良好的互動，有效的提昇學習成效與學習時間，教學者也可即時掌握學習者的學習進度，增進教學品質。

以下我們將依據學生學習的過程，逐一展示並解釋 PLAT 系統的使用流程，當學生開啟系統，會進入元智大學的「個人 Portal」頁面（圖三），此時學生必須輸入自己的帳號密碼以進入「個人 Portal」頁面，所有課程的相關資訊都將於「個人 Portal」上作分享與交流。



圖三. 登入個人 Portal

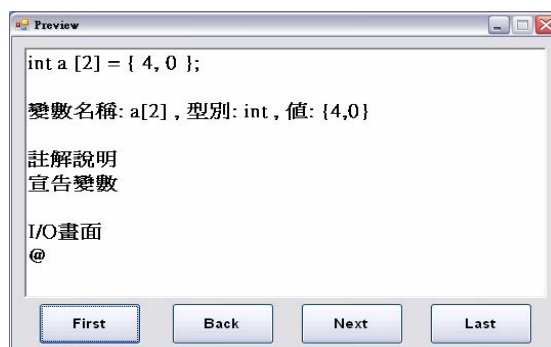
學生從「個人Portal」頁面下載練習題檔之後，將使用「編輯工具」介面（圖四）開啟練習題檔，「編輯工具」介面提供學生模擬程式執行流程的功能，首先我們說明介面的右半部份，右上方顯示練習題檔的程式碼，學生可直接點選要編輯的程式碼，點選之後程式碼會顯示在右下方的編輯區域，編輯區域中有四塊編輯模組，由上而下依序是，（一）顯示目前所要編輯的那一行程式碼，（二）編輯目前所選擇的程式碼中所包含的變數資料，（三）對於當前程式執行的步驟編寫註解，（四）編輯程式執行後顯示在螢幕上的結果。



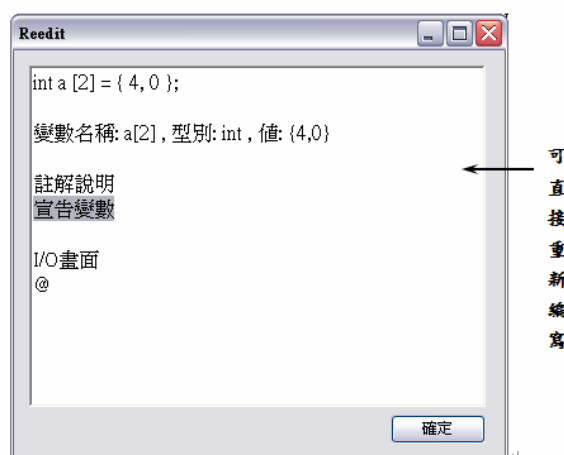
圖四. 編輯工具

接著看到介面的左半部份，該部份紀錄編輯好的所有程式執行步驟，供使用者預覽自己所編寫的程式執行流程，使用者可透過預覽模式（圖五）觀看所有的執行步驟，另外系統也

提供修正模式（圖六）來重新編寫錯誤的執行步驟。



圖五. 預覽模式



圖六. 修正模式

學生編輯完之後，將學習記錄上傳到「個人Portal」頁面供教師作為教學的參考，教師從「個人Portal」頁面下載學生的學習紀錄，使用「批改工具」介面（圖七）來進行批改，「批改工具」介面是屬於教師用編輯器的獨有功能，它能有效幫助教師統察學生的學習紀錄，並予以批改及統計，目前在批改方面還只支援教師自行批改，介面最左方是教師預先編輯好的正確程式執行步驟，中間是學生所編輯的程式執行步驟，最右邊為批改紀錄區，由教師決定每一個步驟的正確與否，並勾選錯誤的內容，如有額外需要說明的都可記錄在選單下面，介面的最下方有簡單的統計，計算答對與答錯的題數，本計畫預期未來要系統加入自動批改比對一些制式的回答，以及以圖表方式表現全班學生的統計紀錄。



圖七. 批改工具

5. 實驗設計

本研究將以模擬程式執行過程的方法建構的一套練習編輯系統，讓資訊工程系大一使用這套系統來輔助程式設計學習，並希望藉由這套系統提昇學習者的學習成效，另外也提供教師編輯教材的工具，期能在教學方面也加以配合，有效增進整的學習時間與成效。

針對學生用模擬練習編輯器的使用者進行學習態度問卷調查，受訪者預計 120 位，並將問卷調查結果作一歸納。期能藉由問卷調查，使系統效能更臻於完備。以下就實驗對象、實驗設計加以說明。

5.1 實驗對象

本研究預計的實驗對象係元智大學資訊工程系修習「程式設計（二）」課程之 120 位大學部學生，使用學生用模擬練習編輯器來學習程式設計。每位同學在進行教學實驗前均已修習過程「程式設計（一）」課程，但尚未學習到實驗的教學內容。

5.2 實驗流程

本研究所使用之工具(instrument)有：(1)程式語言先備知識測驗，(2)成就測驗，(3)PLAT 電腦輔助學習編輯器二套(學生用 vs. 教師用)，及(4)程式設計學習態度問卷。

實驗採取前測—後測設計，總實行時間約為兩週，第一週先進行前測，再經過一段時間的學習課程後，讓學生使用學生用模擬練習編輯器來練習回家作業，並於隔週課堂上進行後測，實驗完成後填寫問卷調查。

實驗過程如下：

1. 前測：程式語言先備知識測驗，進行時間

15分鐘。

2. 學習活動說明：研究者為了能夠讓學生能夠順利使用模擬練習編輯器進行回家作業，在前測之後將為模擬練習編輯器說明教學，使學生在解題作業時，將軟體操作能力差異的影響降到最低，進行時間20分鐘。
3. 程式設計課程：程式設計課程學習，進行時間100分鐘。
4. 電腦化程式語言學習：學生回家後自行使用模擬練習編輯器完成回家作業。
5. 後測：隔週成就測驗，進行時間25分鐘。
6. 問卷調查：在後測完畢後讓使用模擬練習編輯器的學生填寫程式設計學習態度問卷，進行時間5分鐘。

6. 結論與未來研究方向

本文以程式設計學習為基礎，提出一套促進學習時效以及師生互動的編輯練習系統，經過模擬程式執行的步驟與流程，藉此了解學生在進行程式碼執行理解之過程，以及犯下的錯誤，並作為教師教學以及編輯教材的參考依據。

在後續研究上，將以此實驗得到的數據為依據，建立適合邏輯推理的學生錯誤模型，進而發展出一套智慧型電腦輔助學習系統，並強調在於將抽象的程式設計邏輯概念實體化，以彌補學生理解上的不足，以及學生學習過程中，適合學生學習之教材篩選策略。

參考文獻

- [1] 劉明洲，“程式設計之創造性解題思維研究，” *國際科技教育整合思考研討會 (ITTE'99)*，國立台東師範學院。3月2-5日，1999。
- [2] 蔡益元，“程式緒與自然程式進化路徑，” 中原大學資訊工程系碩士論文，2004。
- [3] 陳關弘、張瑞觀，“應用部落格於學習歷程檔案系統建置-以國小六年級電腦課為例，” *2006EITS 台南科大的數位學習研討會*，高雄師範大學資訊教育研究所，2006。
- [4] 鍾靜宜，“教學策略與學習工具對高中程式語言學習之影響，” 國立台灣師範大學資訊教育研究所碩士論文，2003。
- [5] Clark, J. M. & Paivio, A., “Dual coding theory and education.,” *Educational Psychology Review*, 3(3), 149-210.,1991.

- [6] Costa, A., “Developing minds: *A resource book for teaching thinking.*,” Alexandria, VA: ASCD,1985.
- [7] Hedberg, J., & Harper, B., “Visual metaphors and authoring,” *ITFORUM*, 1998
- [8] Johnson, R. P., “Computer, cognition and curriculum: Retrospect and prospect,” *Journal of Educational Computing Research*, 4(1), 1-29.,1988.
- [9] Mayer, Richard E. , & Sims, Valerie K., “For whom is a picture worth a thousand words? Extensions of a dual-coding theory of multimedia learning.,” *Journal of Educational Psychology*, 86(3), 389-401., 1994.
- [10] Marcia C. Linn., & John Dalbey., “Cognitive consequences of Programming Instruction: Instruction, Access, and Ability.,” *Educational Psychologist*, Vol.20, No.4, 191-206.,1985.
- [11] Nada Dabbagh, “Authoring Tools and Learning Systems: A Historical Perspective.,” George Mason University, 2001