

一個支援適性化學習的程式設計練習系統

An Adaptive Programming Learning Environment

林紹陽、董少桓、林宗德
國立雲林科技大學資訊管理系
{ M10223011, tungsh, lintt }@yuntech.edu.tw

曾筱倩
資訊工業策進會
tseng.tsc@gmail.com

【摘要】由於程式設計師的需求日益增加，以及網路與雲端技術的進步，使得世界各地出現許多在瀏覽器中練習寫程式的網站。然而，這些網站中練習題的編排方式，都是由淺入深、一視同仁的方式編排，並沒有考慮到有程式設計經驗的學習者的需要。因此，對已經會寫程式，但卻想學習另一個程式語言的學習者而言，這種以初學者的需求來編排練習題的方式，勢必會浪費這些有經驗的學習者的時間。因此，本研究簡述了一個支援適性導引功能的程式設計練習系統的設計與實施。經由實驗評估之後顯示，這種適性化的編排方式，能夠增加高程度學習者的學習效率。

【關鍵詞】 程式設計教學、適性導引、資訊科學教育

***Abstract:** Due to increasing demands of programmers and advances in internet technologies, many learning-by-doing web sites that provide programming exercises for learners to practice have been established. The exercises in these web sites are designed for novice learners and are sequenced linearly regardless of learners' performance. As a result, an experienced learner who need to learn another programming languages may waste valuable learning time on exercises that are already familiar. To address this problem, this paper presents the design and development of a programming learning system that offers learners different practice paths depending on learners' performance. Empirical study reveals that adaptive programming exercises can improve the learning efficiency for learners in upper level of a programming class.*

Keywords: programming education, adaptive guidance, computer science education

1. 背景

由於雲端以及智慧型手機應用的日益普及，形成了全球各地對程式設計師的廣大需求。許多國家也體認到，國民整體的程式設計能力，不但可以增加就業，更是國家競爭力的重要指標；因此，程式設計將不僅僅是少數人才的特殊技能，而是一般大眾應該具備的基本知識。

由於這股風潮，再加上實際練習寫程式，是學習程式設計最有效的方法；因此，許多以練習為主要學習方式的「練習導向」程式教學網，也如雨後春筍般的出現(Code.org, 2014; Lynda.com, Inc., 2015; Ryzxa, Inc., 2015; Treehouse Island, Inc., 2015; Udacity, Inc., 2015)。這些網站的特色是提供「由淺入深、單一練習路徑」的程式練習教材供使用者學習。然而，網際網路使用者的特性是多元。就學習某個程式語言而言，有的學習者是完全的初學者，而有的則可能是有經驗的程式設計師。因此，能夠藉由調整原本單一路徑的練習題目，來為不同程度的學習者提供適性化的練習路徑，便可以增進學習者的學習效率。

有鑒於此，本研究介紹了一個適性化程式練習系統——PLWeb。PLWeb 提供了能夠將「由淺入深、單一練習路徑」的練習題目，調整成為不同程度的學習者，提供不同學習路徑的適性化功能。PLWeb 的教師出題與學生練習的功能已經使用多年(Tung, Lin, & Lin, 2013)。在 2014 年，我們為它新增了上機考試與適性化配題的功能。在接下來的論文中，我們將簡要的歸納適性化程式設計學習領域的相關文獻，簡述 PLWeb 系統的主要功能，說明其適性化配題系統的設計，接著進行使用的評估，最後提出結論與未來研究的方向。

2. 相關研究

自古以來，「因材施教」便是一種理想的教學方式。而在程式設計教學領域，也有眾多符合因材施教精神的系統與文獻被發表。根據 Brusilovsky & Peylo (2013) 所使用的名稱，程式設計的因材施教系統，可以約略分成智慧型家教 (Intelligent Tutoring)、智慧型合作學習 (Intelligent Collaborative Learning) 與適性化超媒體 (Adaptive Hypermedia) 三大類。

智慧型家教系統能夠分析學生程式碼的錯誤所在，並給予適當的提示，以幫助學生完成程式設計的練習 (Weber & Brusilovsky, 2001)。智慧型合作學習系統則使用適性分組或虛擬學習同伴與虛擬家教，來協助學習 (Chan & Baskin, 1990; Chou, Chan, & Lin, 2003)。

適性化超媒體系統則可以分為適性呈現 (Adaptive presentation) 和適性導航 (Adaptive navigation support) 兩種。適性呈現的目的是將教材依照學生的學習目標、已有的知識與其它資訊來調整呈現教材的方式。適性導航則是藉著調整超連結的外觀與次序，來幫助學生在超媒體網站中找到適合的內容 (Brusilovsky & Peylo, 2013)。

在程式語言教學領域，JavaGuide 是少數以適性導航的方式，藉由提供學生預測程式執行結果的題目，來協助學生瞭解程式碼執行過程的系統 (Hsiao, Sosnovsky, & Brusilovsky, 2010)。JavaGuide 是以個人學習進度或者團體時間的平均作為標準，藉者提示學生是否切換題目，來達到適性化的效果。然而，這個系統的練習方式，只限於觀察與預測程式碼的執行結果，而沒有提供實作性質的適性化練習。

3. PLWeb 功能簡介

PLWeb 提供了包括：教材管理、課程管理、學習狀態顯示等透過伺服器與瀏覽器所提供的功能 (Tung et al., 2013)。然而，它的程式編輯器則是與使用者互動最多的部份。PLWeb 的程式編輯器是從一個以 Java 撰寫的開放源碼程式編輯器——jEdit，修改並延伸而來。程式編輯器中包含四個區域 (圖 1)：題目描述區 (左上)、編輯區 (右上)、輸出樣本區 (左下)、以及測試區 (右下)。題目描述區是一個迷你瀏覽器，它可以將

題目的內容顯示出來。如果該題目提供部分解答，則部分解答會被自動的載入編輯區中。而學生可以將編輯區中的內容以填入空格、延伸功能或自行全部編寫的方式來完成練習題。

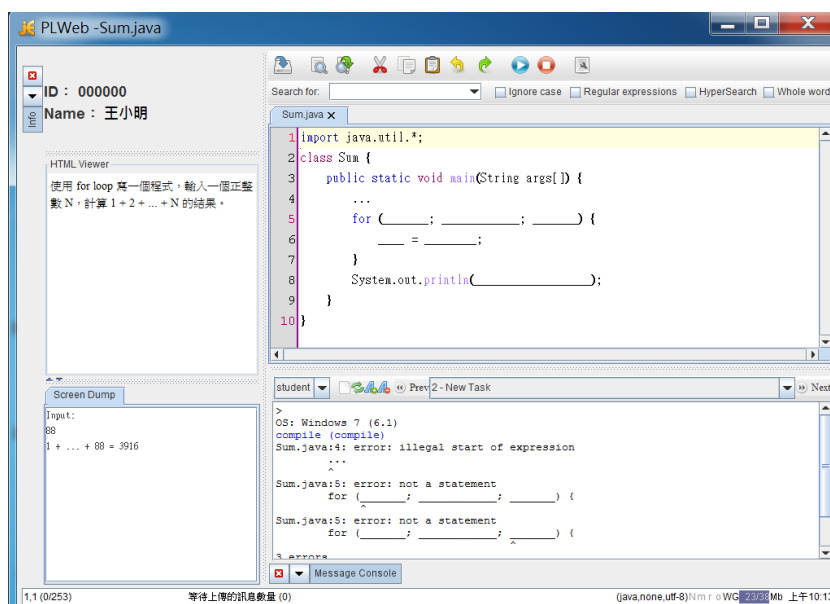


圖 1、程式編輯器

當學生在網頁中按下某一教材之「開始練習」按鈕時，系統便會將程式編輯器伴隨著一組練習題從伺服器下載到客戶端。學生可以依照题目的描述，以及輸出樣本區的內容，在編輯區中開始練習。學生若按下「執行」按鈕，程式編輯器就會呼叫預先安裝在學生電腦中的編譯器，將編輯區中的程式進行編譯與執行。程式編輯器會自動比對學生程式的輸出和輸出樣本區的內容是否相同，以判斷學生的程式是否通過測試。而測試的結果會被自動地送至伺服器，並儲存在資料庫中。此外，為了防止作弊或進行上機考試，教師也可以提供經過加密的隱藏式測試案例。學生在通過輸出樣本區的測試要求後，可以接著按下「提交」按鈕。這時隱藏式的測試案例，便會進一步的檢測學生的程式是否確實達到题目的要求，而不只是以使用 printf 或其它不合題目要求的方式印出輸出樣本區的答案。

在結束一題練習題的撰寫及測試之後，學生可以按「Next」按鈕，繼續練習。這種設計可以讓學生以循序漸進的方式，一題接著一題的學習程式設計，而不需要浪費時間預備接下來練習每一題時所需要的作業環境。此外，有些題目也可以提供部分解答，因此能幫助學生把注意力集中在主要概念的學習，而不需要分心於次要的程式語法或其它細節上。

程式編輯器不僅可以讓學生練習程式設計，還可以協助教師製作練習題目。當程式編輯器處於作者模式時，它會在編輯區中同時開啟三個頁籤：題目描述、解答以及部分解答。教師可以任意切換並編輯這三個頁籤的內容。當教師按下「執行」按鈕後，便可以在命令提示視窗中輸入測試資料。而命令提示視窗中的內容會自動地被儲存起來，當作輸出樣本區的內容。此外，為了預備上機考試的試題，教師也可以開啟編輯隱藏式測試案例的頁籤。如果程式編輯器發現有隱藏式測試案例的檔案存在，則編輯器也會將隱藏式測試案例中的資料當成程式的輸入資料，並且將相對應的輸出結果加密。在結束練習題的建立之後，練習題作者可以按「Next」按鈕，繼續建立下一題練習。在完成所有的練習題之後，作者可以按下「上傳」按鈕，將練習題打包並上傳至伺服器。

PLWeb 的上機考試功能讓學生在監考人員的巡視下無法作弊。上機考試在 Windows 作業系統的實現方式，是藉著下載的程式編輯器在使用者端取得系統權限後，便將所有視窗桌面上的程式全部關閉。而為了防止學生使用 Ctrl-Alt-Delete 三鍵同時按下，並使用工作管理員自行啟動其他軟體，程式編輯器會每隔一秒便自動關閉可能已被啟動的工作管理員。這個方式讓學生除了使用仍然與主機相連的程式編輯器外，沒有任何其他的應用程式可以使用，因而得到了一個簡易、容易使用而且實施成本低的上機考試環境。

4. 適性化學習功能

PLWeb 的適性化學習功能，可以讓教師將每一個教學單元的練習題，拆成數個題組，讓學生可以有多樣化的學習路徑。舉例來說，在 Java 程式語言中，一支程式通常由幾個部分所組成，例如：迴圈、條件式、例外處理與方法等，教師可以將一支練習程式的組成元件，設計成不同的練習程式，並形成一個題組。例如以下是為已經有程式設計經驗的學習者設計的「由難而易」的題組，如果學習者能夠順利的回答最具挑戰性的第 4 題，那麼就沒有必要再練習其餘的題目。為了便於說明，題號小的題目代表比較簡單的練習，而題號大的題目則代表比較難的練習：

題目 4：請顯示 1 到 10 中各個「偶數」的階乘。

程式碼：

```
for (int i = 1; i <= 10; i++) {
    int ans = 1;
    if (i % 2 == 0) {
        for (int j = 1; j <= i; j++)
            ans *= j;
        System.out.println(ans);
    }
}
```

題目 3（比題目 4 簡單的題目）：請利用迴圈計算 5 的階乘。

程式碼：

```
int result = 1;
for (int i = 1; i <= 5; i++)
    result *= i;
System.out.println(result);
```

題目 2（比題目 3 簡單的題目）：請利用「迴圈」，輸出 1 到 10。

程式碼：

```
for (int i = 1; i <= 10; i++)
    System.out.print(i);
```

題目 1（更為簡單的題目）：請判斷輸入值是偶數還是奇數。

程式碼：

```
int input = reader.nextInt();
if (input % 2 == 0)
    System.out.println("偶數");
```

```
else
    System.out.println("奇數");
```

題目 4 是由「迴圈」與「條件式」所組成，因此教師可以針對迴圈以及條件式兩個主題來設計其它的練習題（題目 3、2 與 1）。如果學生在練習題目 4 時，無法在設定的時間內完成，代表學生可能對迴圈或條件式的概念比較不熟，系統將以提醒的方式，提醒學生是否更換此題組中較簡單的題目來練習。以一組九題的題組為例，一種編排題組中難易度的方式可能如圖 2 所示。其中 4、3、2、1 為一個子題組，6、5 為一個子題組，9、8、7 為一個子題組（如果這三個子題組的題目以由簡而難的方式組合成一個題組，那麼其題號的編排方是將是 1, 2, ..., 9）。如果程度較好的同學，能夠在時限內完成這三個子題組中最難的題目，那麼這位同學的學習路徑為 4 → 6 → 9；而程度稍差的同學，其路徑則可能為 4(未完成) → 3(完成) → 4(完成) → 6(超時但完成) → 5(完成) → 9(完成)（如圖 3 所示）。因此，學生程度不同，練習時所花的時間也不一樣，而可以同時達到為程度較佳的學生節省練習時間，並且為程度較差的學生增加練習機會的效果。教師也可以將原本「由淺入深、單一練習路徑」排列的練習，重新組合成適性化的練習題組。讓學生在沒有老師從旁指導協助時，仍然能夠擁有個人化的學習路徑，因而能達到「因材施教」的效果。

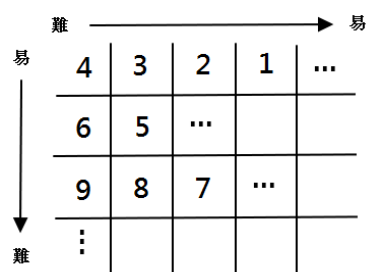


圖 2、適性化學習題組

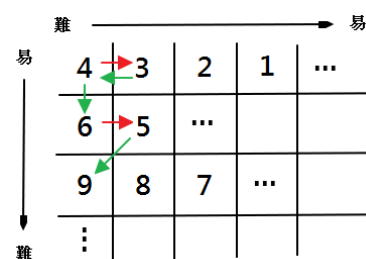


圖 3、適性化學習路徑圖

由於第一個子題組的第一題（題目 4）是該題組最難的題目，而最後一題是最簡單的題目（題目 1）。因此當學生在嘗試完該子題組中所有的題目之後，若仍然無法完成最簡單的題目時，就是教師介入給予個人指導的時機，這時系統顯示學習狀態的儀表板可以提醒教師哪些學生需要協助，而教師便可以優先關注這些同學的練習。

我們是以「PLWeb 上線 6 年以來所有學生練習過此題的時間加總平均」以及「上一屆所有學生練習過此題的時間加總平均」，各占 50% 的比例來做為每一題題目的時間限

制。然而，PLWeb 已上線 6 年，其中可能因資訊化程度或者教學方法有所改變，導致平均水準不一，因此加入上屆 50% 之平均以減少上述原因所造成的落差。

5. 實驗與評估

我們在 103 學年度的下學期為 PLWeb 適性化功能的學習效率進行了實驗評估。實驗對象是雲林科技大學資訊管理系四技部一年級的學生。這些學生已經在一年級上學期以 Scheme 語言學習過諸如條件判斷、迴圈等不同的程式設計概念與方法，而他們在下學期將學習與 Scheme 語言語法非常不同的 C 語言。

實驗是以該班學生上學期 Scheme 語言的期末成績為分組依據，並且在經過 t 檢定分析後，確認兩組之間的程式設計能力並無明顯差異 ($p = .778$)。接著再將這兩組同學分成高程度組和中低程度組。實驗前，教師先對實驗組和控制組以相同的內容授課，接著開放實作練習，實驗組使用「適性化學習」的功能，而控制組沒有使用適性化的功能。

實驗內容為迴圈，實施時間為三小時，前兩小時為授課與練習時間，後一小時為測驗時間，練習題共 13 題，而測驗題共 5 題。測驗題與練習題都是迴圈程式設計的實作題。在實驗過程中，我們發現使用適性化學習的高程度組的同學，並不需要兩小時，便練習完需要練習的題目，而提早進行測驗。測驗的結果可以評估兩種學習方式之學習效率的差異。「學習效率」的定義是「測驗答對題數除以練習題數」後所得到的比率。因此，若將兩位在測驗時答對的題數相同的同學相比，如果其中一位練習的題數較少，那麼則代表這位同學有比較高的學習效率。

實驗樣本數為 61 人，以是否使用適性化功能為自變項，學習效率為依變項，進行 t 檢定，以探討使用適性化功能是否能夠對整班、高程度或中低程度組的學習效率有影響。

兩組學生的學習效率如表 1 所示，經由 t 檢定分析後，結果顯示，兩種不同的學習方式對不分高低程度時的全班學生的學習效率為有顯著的差異 ($p = .012 < .05$)，而實驗組的學習效率的平均數 (.466) 要大於控制組的學習效率的平均數 (.345)，代表使用適性化的方式來學習，對全班學生的學習效率有幫助。

接著再對兩組高程度組的同學進行 t 檢定，結果顯示，兩種不同的學習方式對高程度學生的學習效率有顯著的差異 ($p = .000 < .05$)，而實驗組學習效率的平均數 (.627) 大於控制組學習效率的平均數 (.388)，代表使用適性化的方式來學習，對高程度組學生的學習效率有顯著的幫助。

最後再對兩組中低程度組學生進行 t 檢定，結果顯示，兩種不同的學習方式對中低程度組學生的學習效率沒有顯著差異 ($p = .225 > .05$)。然而，適性化組的平均數仍然高於非適性化組的平均數。

表 1 實驗結果統計表

	適性化學習(實驗組)			非適性化學習(控制組)			
	人數	平均	標準差	人數	平均	標準差	顯著性
全班	30	.466	.218	31	.345	.137	.012*
高程度組	9	.627	.153	9	.388	.033	.000***
中低程度組	21	.397	.207	22	.328	.159	.225

* $p < .05$, ** $p < .01$, *** $p < .001$

根據實驗結果，我們可以發現適性化學習對高程度的學習者最有顯著的幫助。因為使用適性化學習，能夠幫助程度較好的學生不用化費時間在練習已經熟悉的題目上，而他們即使不練習這些題目，也能得到較高的測驗成績。而適性化學習對中低程度組同學的幫助比較不明顯的原因可能是：實驗限定兩組同學的練習時間都為兩小時，而這些中低程度的同學，由於受到時間的限制，以至於都沒有完成足夠的練習，因而沒有得到理想的學習效果。

6. 結論與未來方向

本研究以「適性化」的學習方式，透過 PLWeb 教學平台來驗證適性化程式設計實作題，是否對學生的學習效率有幫助。實驗的結果顯示，適性化程式設計實作題對高程度的學習者最有明顯的幫助；而對中低程度的學習者而言，雖然有幫助，但並不明顯。

在這次的實驗裡，我們沒有讓使用適性化功能的同學，能夠自由選擇先做比較簡單一點的題目的機會，因此學生必須不斷的嘗試完成一題可能太難的題目，直到系統設定的練習時間結束時，才能換題。這樣的方式，也會讓中低程度的同學，浪費時間在自己還不熟悉的題目上；因此，我們將改進這個缺點，以進一步的提升中低程度學生的學習成效。

7. 致謝

本研究依經濟部補助財團法人資訊工業策進會「104 年度資策會創新前瞻技術研究計畫(1/1)」辦理。

參考文獻

- Brusilovsky, P., & Peylo, C. (2003). Adaptive and intelligent web-based educational systems, *International Journal of Artificial Intelligence in Education*, 13(2-4), 156-169.
- Chan, T. W., & Baskin, A. B. (1990). Learning companion systems. In C. Frasson, & G. Gauthier (Eds.), *Intelligent Tutoring Systems: At the crossroads of artificial intelligence and education* (pp. 6-33). Norwood: Ablex Publishing.
- Chou, C.-Y., Chan, T.-W., & Lin, C.-J. (2003). Redefining the learning companion: the past, present, and future of educational agents. *Computers and Education*, 40(3), 255-269.
- Code.org (2014). *Hour of code home page*. Retrieved from <http://hourofcode.com>
- Hsiao, S. I-H., Sosnovsky, S., & Brusilovsky, P. (2010). Guiding students to the right questions: adaptive navigation support in an e-learning system for Java programming. *Journal of Computer Assisted Learning*, 26(4), 270-283.
- Lynda.com, Inc. (2015). *Lynda.com home page*. Retrieved from <http://www.lynda.com>
- Ryzxa, Inc. (2015). *Codecademy home page*. Retrieved from <http://www.codecademy.com>
- Treehouse Island, Inc. (2015). *Treehouse home page*. Retrieved from <https://teamtreehouse.com>
- Tung, S. H., Lin T. T., & Lin, Y. H. (2013). An exercise management system for teaching programming, *Journal of Software*, 8(7), 1718-1725.
- Udacity, Inc. (2015). *Udacity home page*. Retrieved from <https://www.udacity.com>

Weber, G., & Brusilovsky, P. (2001). An adaptive versatile system for web-based instruction, *International Journal of Artificial Intelligence in Education*, 12(4), 351-384.