

網路化電腦輔助程式設計學習之研究

A study on computer-assisted programming learning through the Web

陳明溥

國立臺灣師範大學 資訊教育學系

E-mail: mpchen@ice.ntnu.edu.tw

摘要

本研究旨在探討概念模型(conceptual model)、程式設計經驗(programming experiences)、電腦先備知識(prior knowledge)、及認知型態(cognitive style)對透過全球資訊網學習環境學習遞迴程式設計學習成效之影響。在實驗教學中,動態及靜態複製模型(copies model)以電腦輔助教學之形式經由全球資訊網進行遞迴觀念之解說、示範及演練。實驗結果顯示:(1)在程式碼評量(code evaluation)方面:程式設計經驗與認知型態顯著影響學習的成果;但電腦先備知識及複製模型類型則無顯著影響;(2)在程式碼產生(code generation)方面:發現有顯著之程式設計經驗 x 電腦先備知識交互作用(interaction),就低程式設計經驗者而言,電腦先備知識顯著影響遞迴程式設計之學習成效;但就高程式設計經驗者而言,複製模型類型則顯著影響遞迴程式設計之學習成效(動態複製模型>靜態複製模型)。關鍵字:全球資訊網、電腦輔助教學、程式設計

壹. 網路化學習

近年來,由於電腦科技不斷的推陳出新及網際網路的蓬勃發展,使得各類型的資訊

得以透過電腦網路快速而廣泛的傳達到世界各地,縮短了原有的時空距離。在此電腦科技快速發展的條件下,教育界也隨之突破藩籬,透過網際網路在不受時空限制下提供學習者各種不同的學習活動。全球資訊網應用在教學上,不但可以用來呈現多元化的教材,更可以因應學習者的個別差異,幫助學習者建立自主性的學習模式及吸收更廣的知識。根據過去的研究(楊家興,民 82; Jonassen, 1991)指出,全球資訊網應用於教學時具有下列五項特性:1. 多元化的資訊;2. 彈性的知識體系;3. 充分共享的知識庫;4. 自主性的遠距學習環境;以及 5. 具高度互動性的學習環境。由於電腦網路的快速、便捷,在短時間內便能提供學習者龐大而豐富的資訊,然而卻也常因資訊過多無法吸收而產生「認知超載」(cognition overload)及在學習過程「迷失」(disorientation)的困擾,進而影響網路化學習的成效。

雖然全球資訊網在教學應用上可以將教師、學生及各知識域內容帶進同一時空,使學習者能夠依照個別差異進行學習,但是其在教學應用上仍有一些潛在問題尚待進一步克服(顏榮泉,民 85),如:1. 初學者適應(adaptation)和迷失(disorientation)的問題;2. 過量的學習資訊;及 3. 學習者控制能力(learner control)的不足等問題。孫春在

(民 84)也指出：由於全球資訊網加入了動態教材、自由瀏覽學習、多元化的討論及共同學習等新的因素，使得遠距合作環境下的教學模式較以往都複雜，不論是學習動機、認知結構、溝通形式、人機介面、評量標準及方式等均需要重新深入探討。因此，本研究主要的目標是將認知負載理論(cognitive load theory)應用於網路化學習活動及其遊歷(navigation)之教學設計上，以克服因遊歷行為而產生的迷失與資訊過多而產生的認知超載問題，並進一步探討概念模型(conceptual model)、程式設計經驗(programming experiences)、電腦先備知識(prior knowledge)、及認知型態(cognitive style)對透過全球資訊網學習環境學習遞迴程式設計學習成效之影響。

貳. 程式語言學習

“遞迴”由於缺乏生活上的例子，常常造成初學者在學習遞迴概念時產生極大的困擾(Anderson, Pirolli & Farrell, 1988)，但是由於“遞迴”在程式設計方面是一種功能強大的解決問題技術，因此一直是程式語言教學上一項重要的觀念。研究者們(Pirolli & Anderson, 1985; Kessler & Anderson, 1989)指出：要使初學者能成功地習得遞迴概念必須仰賴適當的心智模型(mental model)才能奏效。在探討迴圈(iteration)與遞迴間之學習遷移(transfer)情形時，研究學者們發現：先學習迴圈對後續學習遞迴有實質上的幫助；但是先學習遞迴卻對後續學習迴圈上卻沒有任何助益(Anzai & Uesato, 1981; Kessler & Anderson, 1989)。Er (1984)在研究遞迴程式設計教學後則指出：初學者學習遞迴概念時的困擾並不是因為遞迴概念太難，而是因為遞迴呼叫時參數的傳遞使得學習遞迴變得複雜難懂。如果教學上能將遞迴過程與參數傳遞以更具體的方式呈現，那麼學習者是否

就能學的更好呢？

Mayer (1981) 針對程式設計初學者所做的研究發現：程度較差的初學者如果給予具體的(concrete)系統模型時，其學習表現會有顯著的提昇。Pirolli 與 Anderson (1985)更進一步建議：類比推理(analogical reasoning)是學習程式設計上一種基本而有效的方法；其他相關研究(Anderson, Farrell & Sauers, 1984; Carbonell, 1983; Gentner & Gentner, 1983; Gick & Holyoak, 1983; Pirolli & Anderson, 1985)也發現透過典型範例(example)學習者可以仿倣(類比)寫出語法正確的程式，進而發展出適切的心智模型。Pirolli (1991)更進一步證實使用典型範例不但可以減少程式設計錯誤的發生率，更能有效的提昇學習程式設計的效率。過去，在程式設計教學上許多研究者提出各種不同的方法與概念模型來幫助學習者快速有效的學習遞迴程式設計，這些方法有：類比模型(Murnane, 1991)、圖示遞迴結構模型(Give'on, 1990)、數學推理模型(Aho & Ullman, 1992; Ford, 1984)、樹狀模型(Koffman, 1992; Krue, 1982)、以及複製模型(Kahney & Eisenstadt, 1982; Kessler & Anderson, 1989)等。由於複製模型(copies model)最適合以電腦輔助學習之形式來呈現(Wilcocks & Sanders, 1994)，因此本研究以複製模型做為遞迴程式設計教學之主要概念模型。

參. 概念模型設計

雙碼理論(dual-coding theory, DCT) (Paivio, 1971, 1986; Clark & Paivio, 1991)主張：人們的心智結構與資訊處理過程中皆包含有文字表示式(verbal representation)與圖像表示式(imaginal representation)二種不同類型之資訊。圖 1 所示為雙碼理論之多媒體學習示意圖(Mayer & Sims, 1994)。如圖 1 中 1 號箭頭所示，當感知器接受到文字(verbal)

資訊時，這一些文字資訊將經由文字編碼過程(verbal encoding process)在工作記憶體(working memory)中被進一步轉換為文字系統之心智表示式(mental representation of verbal system)，此過程稱為：建立文字表示鏈結(building a verbal representational connection)。相同的，如圖 1 中 2 號箭頭所示，當感知器接受到視覺(visual)資訊時，這一些視覺資訊將經由視訊編碼過程(visual encoding process)在工作記憶體(working memory)中被進一步轉換為視訊系統之心智表示式(mental representation of visual system)，此過程稱為：建立視訊表示鏈結(building a visual representational connection)。圖 1 中 3 號箭頭表示建立“文字系統心智表示式”與“視訊系統心智表示式”間之“參照鏈結”(referential connection)。學習者的學習成效如何則視“文字表示鏈結”、“視訊表示鏈結”、以及“參照鏈結”等三種鏈結建立之品質而定。研究者們(Mayer & Gallini, 1990; Mayer & Sims, 1994)也發現：文字資訊與視覺資訊同時呈現比先後呈現更能有效的幫助建立“參照鏈結”提昇學習效果。因此，就 DCT 之觀點而言，學習者需要同時使用“心智文字系統”與“心智視訊系統”來有效處理資訊，如果能促進此三種鏈結之建立，就能有效提昇學習的成效。

如同 DCT 所主張的，許多相關的研究(Cooper & Sweller, 1987; Sweller, 1988, 1989)也指出：需要學習者在內心中整合不同來源之學習資源(如：範例與說明)的教學，往往耗費學習者過多心力而影響學習效果。因此，在使用遞迴複製模型進行程式設計教學時，如果能善用電腦輔助學習之互動特性，以動態方式逐步呈現遞迴的過程與參數之傳遞(如圖 2 所示)，是否能如 DCT 所主張”有助於參照鏈結之建立，進而提昇學習效

果”正是本研究所要探討的重點。

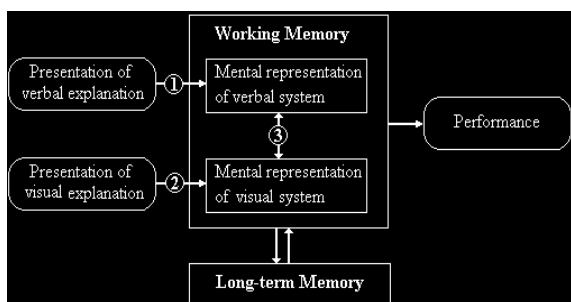


圖 1. 雙碼理論之多媒體學習示意圖。(摘自 For whom is a picture worth a thousand words? Extensions of a dual-coding theory of multimedia learning, by Mayer and Sims, 1994, in *Journal of Educational Psychology*, 86(3), 389-401.)

肆. 研究方法

本研究依據 DCT 之觀點將遞迴複製模型以網路化電腦輔助教學之形式設計成為時五十分鐘之教學，對象為 153 位高職一、二年級資料處理科學生。研究採性向處理交互作用 (aptitude-treatment-interaction, ATI) 設計，並以概念模型(動態 vs. 靜態)、程式設計經驗(高 vs. 低)、電腦先備知識、及認知型態(cognitive style)為研究之獨立變數，以遞迴程式設計後測成績做為條件變數(criterion variables，包括：程式碼評量, code evaluation 及程式碼產生, code generation)分別進行多重回歸分析(multiple regression)。本研究所使用之工具(instrument)有：(1) 先備知識測驗, (2) 遞迴成就測驗, (3) 網路化遞迴電腦輔助教學軟體二套(動態遞迴複製模型 vs. 靜態遞迴複製模型), 及(4) 網路化遞迴程式設計學習態度問卷一份。

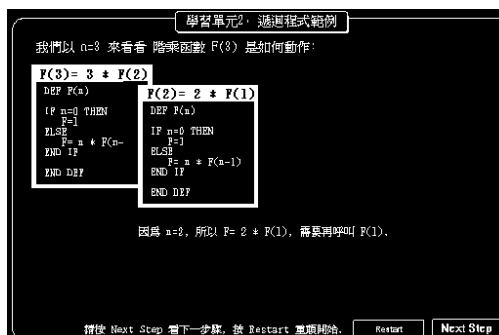


圖 2. 動態遞迴複製模型，以動態方式逐步呈現遞迴的過程與參數之傳遞。

$$Y_{\text{程式碼評量}} = 3.679 + .389 YR + .05 CS$$

伍. 重要研究結果

本研究多重回歸分析之基本模型 (baseline model) 為:

$$Y = b_0 + b_1 \text{GRP} + b_2 \text{BCC} + b_3 \text{YR} + b_4 \text{CS}$$

GRP: 概念模型類型 (動態複製模型 vs. 靜態複製模型)

BCC: 電腦先備知識

YR: 程式設計經驗(高 vs. 低)

CS: 認知型態

一. 程式碼評量(code evaluation)之分析

如表一所示，由於交互作用 (GRPxBCC、GRPxCs、YRxBCC、YRxCs) 皆未達顯著水準，因此，移除交互作用後，再進行主效果(main effects)之分析。

表一

由多重回歸分析基本模型(baseline model)移除之交互作用

	Beta In	t	Sig.	Partial Correlation
GRPxBCC	-.423	-1.067	.288	-.089
GRPxCs	.526	1.603	.111	.133
YRxBCC	.011	.023	.982	.002
YRxCs	-.213	-.687	.493	-.057

遞迴成就測驗(程式碼評量)之主效果 (main effects) 回歸分析摘要如表二所示。程式設計經驗與認知型態主效果皆達顯著水準， $F(1, 144) = 2.278, p = .024$ ；及 $F(1, 144) = 2.557, p = .012$ 。主效果之 R-square 為 .070，顯示在遞迴程式碼評量成就測驗上 7.0% 的變異數可歸因於程式設計經驗與認知型態之影響。就此二項主效果而言，由標準化後 (SD=1) 之 Beta 值得知，程式設計經驗與認知型態對遞迴程式設計學習之影響大致相等 (0.182 vs. 0.204)。程式碼評量分析後可得知回歸方程式為：

表二

遞迴成就測驗(程式碼評量)之主效果回歸分析摘要

Model		Unstdized Coefficients	Std. Error	Stdized Coefficients	t	Sig.
1	(Const)	3.650	.490		7.445	.000
	GRP	-.232	.170	-.108	-1.365	.174
	YR	.322	.177	.151	1.814	.072
	BCC	7.765E-02	.052	.125	1.492	.138
2	(Const)	4.831E-02	.020	.191	2.379	.019
	YR	3.313	.425		7.801	.000
	BCC	.308	.178	.144	1.732	.085
	BCC	8.028E-02	.052	.129	1.539	.126
3	(Const)	4.706E-02	.020	.186	2.313	.022
	(Const)	3.679	.354		10.402	.000
	YR	.389	.171	.182	2.278	.024
	CS	5.169E-02	.020	.204	2.557	.012

Dependent Variable: PE (程式碼評量)

二. 程式碼產生(code generation)之分析

如表三所示，由於交互作用 (YRxBCC) 達顯著水準， $F(1, 143) = -2.544, p = .012$ ，

表三

程式碼產生之多重回歸分析及顯著之之程式設計經驗 x 電腦先備知識 (YRxBCC) 交互作用

Model		Unstdized Coefficients	Std. Error	Stdized Coefficients	t	Sig.
1	(Const)	3.679	.672		5.477	.000
	GRP	-.548	.232	-.188	-2.358	.020
	YR	.622	.243	.214	2.559	.012
	BCC	4.913E-02	.071	.058	.689	.492
2	(Const)	-1.838E-03	.028	-.005	-.066	.947
	(Const)	.372	1.458		.255	.799
	GRP	-.556	.228	-.191	-2.435	.016
	YR	3.009	.968	1.035	3.109	.002
3	BCC	.564	.214	.664	2.634	.009
	CS	-6.853E-03	.027	-.020	-.250	.803
	YRxBCC	-.355	.140	-1.168	-2.544	.012

因此多重回歸分析之基本模型 (baseline model) 先依程式設計經驗(高 vs. 低) 分組後再分別分析之，此時的多重回歸分析方程式成為： $Y = b_0 + b_1 \text{GRP} + b_2 \text{BCC} + b_3 \text{CS}$ 。

1. 低程式設計經驗組：

低程式設計經驗組遞迴成就測驗(程式碼產生)之主效果(main effects)回歸分析摘要如表四所示。電腦先備知識主效果達顯著水準， $F(1, 73) = 2.095, p = .040$ 。主效果之 R-square 值為 .057，顯示在遞迴程式碼產生成就測驗上 5.7% 的變異數可歸因於電腦先備知識之效果。因此可以得知回歸方程式為：

$$Y_{\text{低程式設計經驗組}} = 2.476 + .208 BCC$$

表四

低程式設計經驗組遞迴成就測驗(程式碼產生)之主效果回歸分析摘要

Model		Unstdized Coefficients		Stdized Coefficients		t	Sig.
		B	Std. Error	Beta			
1 (Const)	(Const)	2.932	.858			3.416	.001
	GRP	-.379	.349	-.126		-1.086	.281
	BCC	.196	.102	.225		1.923	.058
2 (Const)	CS	1.693E-02	.041	.049		.410	.683
	(Const)	3.022	.825			3.663	.000
	GRP	-.355	.342	-.118		-1.038	.303
3 (Const)	BCC	.206	.099	.236		2.078	.041
	(Const)	2.476	.636			3.891	.000
	BCC	.208	.099	.238		2.095	.040

Dependent Variable: PG (程式碼產生)

2. 高程式設計經驗:

高程式設計經驗組遞迴成就測驗(程式碼產生)之主效果(main effects)回歸分析摘要如表五所示。概念模型類型(動態複製模型 vs. 靜態複製模型)主效果達顯著水準， $F(1, 72) = 2.471, p = .016$ 。主效果之 R-square 值為 .078，顯示在遞迴程式碼產生成就測驗上 7.8% 的變異數可歸因於概念模型類型之影響。因此可以得知回歸方程式為：

$$Y_{\text{高程式設計經驗組}} = 5.563 - .745 GRP$$

表五

高程式設計經驗組遞迴成就測驗(程式碼產生)之主要效果回歸分析摘要

Model		Unstdized Coefficients		Stdized Coefficients		t	Sig.
		B	Std. Error	Beta			
1 (Const)	(Const)	7.074	.935			7.568	.000
	GRP	-.794	.300	-.298		-2.650	.010
	BCC	-.145	.096	-.170		-1.510	.135
2 (Const)	CS	-3.878E-02	.036	-.120		-1.064	.291
	(Const)	6.679	.859			7.778	.000
	GRP	-.767	.299	-.288		-2.567	.012
3 (Const)	BCC	-.151	.096	-.177		-1.578	.119
	(Const)	5.563	.492			11.308	.000
	GRP	-.745	.302	-.280		-2.471	.016

1 (Const)	7.074	.935		7.568	.000
GRP	-.794	.300	-.298	-2.650	.010
BCC	-.145	.096	-.170	-1.510	.135
CS	-3.878E-02	.036	-.120	-1.064	.291
2 (Const)	6.679	.859		7.778	.000
GRP	-.767	.299	-.288	-2.567	.012
BCC	-.151	.096	-.177	-1.578	.119
3 (Const)	5.563	.492		11.308	.000
GRP	-.745	.302	-.280	-2.471	.016

Dependent Variable: PG (程式碼產生)

陸. 結論與建議

由研究結果顯示：(1) 在就遞迴程式碼評量(code evaluation)方面，程式設計經驗與認知型態顯著影響學習的成果；但電腦先備知識及複製模型類型則無顯著影響。換言之，程式設計經驗愈多，遞迴程式碼評量的表現也愈好；而認知型態越是趨向場地獨立(field independent)，遞迴程式碼評量的表現也愈好。而就程式設計經驗與認知型態之效果比較而言，兩者對遞迴程式設計學習之影響大致相等(0.182 vs. 0.204)。(2) 在遞迴程式碼產生(code generation)方面：整體而言，程式設計經驗愈多、電腦先備知識越好、認知型態越是趨向場地獨立，則其遞迴程式碼產生的表現也愈好；而且動態概念模型的學習效果比靜態概念模型來的好。

就程式設計經驗再細分，低程式設計經驗者其電腦先備知識顯著影響其遞迴程式設計之學習成效，此現象顯示程式設計經驗較不足者其心智模型的建立主要是仰賴其個人之電腦先備知識；換言之，初學者如果有較佳的電腦先備知識則其遞迴程式設計之學習成效也越好。但就高程式設計經驗者而言，使用動態概念模型學習比使用靜態概念模型有較好的學習效果。此結果顯示即使是對程式設計經驗較豐富的學習者，動態概念模型也能有效幫助其建立適切心智模型，進而有更好的學習成效。

本研究結果顯示：教學上提供動態之概念模型，可以有效幫助學習者建立適切之心智模型，使有經驗的程式設計者在網路化學

習環境下有更好的學習表現；但就程式設計經驗不足者而言，如何有效補強及建立足夠的電腦先備知識是進行網路化學習前的一項要務。由於本研究未直接比較在網路環境與非網路環境之程式設計學習成效，因此建議後續可就此議題再做深入探討。

柒. 致謝:

本研究承蒙國科會科教處專題研究計劃補助(NSC 88-2520-S-003 -005)始得以順利完成。

捌. 參考文獻

- Aho, A. V. & Ullman, J. D. (1992). Foundations of computer science. New York: W. H. Freeman.
- Aman, J. (1992). Gender and attitude toward computers. In C. D. Martin & E. Murchie-Beyma (Eds.), In Search of Gender Free Paradigms for Computer Education, (pp. 33-46). International Society of Technology Education, Eugene, OR.
- Anderson, J. R., Farrell, R. & Sauers, R. (1984). Learning to program in LISP. Cognitive Science, 8, 87-129.
- Anderson, J. R., Pirolli, P. L. & Farrell, R. (1988). Learning to program recursive functions. In M. T. H. Chi, R. Glaser & M. Farr (Eds.), The nature of expertise (pp. 153-183). Hillsdale, NJ: Lawrence Erlbaum.
- Anzai, Y. & Uesato, Y. (1981). Learning recursive procedures by middle school children. Proceedings of the Fourth Annual Conference of the Cognitive Science Society, 100-102.
- Carbonell, J. G. (1983). Derivational analogy and its role in problem solving. Proceedings of the National Conference on Artificial Intelligence. Washington, DC: University of Maryland and George Washington University.
- Carlson, R. & Wright, D. (1993). Computer anxiety and communication apprehension: Relationship and introductory college course effects. Journal of Educational Computing Research, 9(3), 329-338.
- Clarke, V. A. (1992). Strategies for involving girls in computer science. In C. D. Martin & E. Murchie-Beyma (Eds.), In Search of Gender Free Paradigms for Computer Education (pp. 71-86). International Society of Technology Education, Eugene, OR.
- Cooper, G. & Sweller, J. (1987). The effects of schema acquisition and rule automation on mathematical problem-solving transfer. Journal of Educational Psychology, 79, 347-362.
- Er, M. C. (1984). On the complexity of recursion in problem-solving. International Journal of Man-Machine Studies, 20, 537-544.
- Ford, G. (1984). An implementation-independent approach to teach recursion. SIGCSE Bulletin, 16(1), 213-216.
- Gentner, D. & Gentner, D. R. (1983). Flowing water or teeming crowds: Mental models of electricity. In D. Gentner & A. L. Stevens (Eds.), Mental models (pp. 99-129). Hillsdale, NJ: Lawrence Erlbaum.
- Gick, M. L. & Holyoak, K. J. (1983). Schema induction and analogy transfer. Cognitive Psychology, 15, 1-38.
- Give'on, Y. S. (1990). Is recursion well defined? Computers and Education, 14(1), 35-41.
- Jonassen, D. H. (1991). Hypertext as instructional design. Educational Technology Research and Development, 39(1), 83-92.
- Kahney, H. & Eisenstadt, M. (1982). Programmers' mental models of their programming tasks: The interaction of real world knowledge and programming knowledge. Proceedings of the Fourth Annual Conference of the Cognitive Science Society, 143-145.
- Kessler, C. M. & Anderson, J. R. (1989). Learning flow of control: Recursive and iterative procedures. In E. Soloway & J. C. Spohrer (Eds.), Studying the novice programmer (pp. 229-260). Hillsdale, NJ: Lawrence Erlbaum.
- Koffman, E. B. (1992). Pascal (4th ed.). Reading, MA: Addison Wesley.
- Krue, R. L. (1982). On teaching recursion.

- SIGCSE Bulletin, 14(1), 92-96.
- Mayer, R. E. (1981). The psychology of how novices learn computer programming. Computing Surveys, 13, 121-141.
- Mayer, R. E. & Sims, V. K. (1994). 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.
- Murnane, J. (1991). Models of recursion. Computers and Education, 16(2), 197-201.
- Pirolli, P. L. (1991). Effects of examples and their explanations in a less on recursion: A production system analysis. Cognition and Instruction, 8(3), 207-257.
- Pirolli, P. L. & Anderson, J. R. (1985). The role of learning from examples in the acquisition of recursive programming skills. Canadian Journal of Psychology, 39, 240-272.
- Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. Cognitive Science, 12, 257-285.
- Sweller, J. (1989). Cognitive technology: Some procedures for facilitating learning and problem solving in mathematics and science. Journal of Education Technology, 81, 457-466.
- Ward, M. & Sweller, J. (1990). Structuring effective worked examples. Cognition and Instruction, 7, 1-39.
- Wilcocks, D. & Sanders, I. (1994). Animating recursion as an aid to instruction. Computers and Education, 23(3), 221-226.
- 孫春在 (民 84)，超媒體網路與遠距合作式電腦輔助學習。科學科技與媒體，21，29-37。
- 楊家興 (民 82)，超媒體：一個新的學習工具。科學科技與媒體，12，28-39。
- 顏榮泉 (民 85)，全球資訊網輔助學習系統之建構模式：以生活科技課程為例。國立台灣師範大學工業科技教育學系碩士論文。