

國小學生學習視覺化程式設計對後設認知與運算思維發展之研究

A Study on the Development of Metacognition and Computational Thinking in Elementary Students Learning Visual Programming

張芷寧¹, 張智凱^{2*},

¹臺灣台南大學數位學習科技學系

²臺灣台南大學數位學習科技學系

* chihkai@mail.nutn.edu.tw

【摘要】 本研究採探索性單組前後測設計，探討運算思維（CT）融入 Scratch 數位說故事教學對 22 位國小學生後設認知之影響。研究結合 Jr. MAI 量表、cCTt 測驗、Dr. Scratch 實作評量與行為分析進行三角檢證。結果顯示，學生在教學介入後知識的認知（MK）構面呈現顯著增長趨勢，但在認知的調節上則無顯著差異，初步反映存在「知行落差」的現象。實作分析揭示，學生的程式特徵存在抽象化與邏輯的學習瓶頸，學生因習慣線性堆疊指令而缺乏模組化思維，導致除錯認知負荷過高。研究建議未來教學應強化自定義積木訓練，並透過長期縱向實驗驗證後設調節能力的內化成效。

【關鍵字】 運算思維；後設認知；Scratch；數位說故事；知行落差

Abstract: This exploratory study investigates the impact of integrating Computational Thinking (CT) into Scratch-based digital storytelling on the metacognition of 22 elementary students. Triangulating data from the Jr. MAI, cCTt, Dr. Scratch, and behavioral analysis, results revealed significant gains in Knowledge of Cognition (MK) but not in Regulation of Cognition (MR), evidencing a "know-do gap". Practical analysis identified abstraction and logic as primary learning bottlenecks; students' reliance on linear command stacking rather than modular thinking increased cognitive load during debugging. Future pedagogy should emphasize custom block training to help students bridge this gap, with longitudinal studies recommended to verify long-term effects.

Keywords: Computational Thinking, Metacognition, Scratch, Digital Storytelling, Know-Do Gap

1. 前言

在基礎教育階段，視覺化程式語言雖能降低語法門檻，使學生專注於邏輯推理，但其積木操作的便利性亦常導致學生陷入「盲目試誤」的淺層學習，阻礙演算法思維的內化 (Lye & Koh, 2014)。程式學習的核心涉及複雜的認知監控與動態調節，這與 Flavell (1979) 提出的「後設認知」高度呼應。然而，教學現場常觀察到學生具備策略知識卻無法落實於實作的「知行落差」現象。本研究選定數位說故事 (Digital Storytelling, DST) 為場域，主因在於其涉及角色同步與多線程邏輯，能激發學生更高強度的監控行為。本研究採取探索性研究取向，透過為期五天的教學觀察，初步分析知行落差的行為轉化機制，以期為後續教學設計提供依據。

2. 文獻探討

2.1. 運算思維與視覺化程式設計

運算思維 (CT) 涉及問題解決與系統設計的核心素養 (Wing, 2006)。視覺化程式設計如 Scratch，雖降低語法門檻，但便利的積木操作若缺乏引導，常使學生傾向盲目試誤而非深層邏輯規劃，進而阻礙思維內化 (Lye & Koh, 2014)。本研究選定數位說故事 (DST) 作為場域，主因其高度涉及角色間的對話同步、廣播傳遞與平行處理。這種多線程的時序協調具備較高的認知複雜度，能觸發學生的認知衝突，迫使學生啟動監控機制來檢查邏輯，是探究後設認知調節的理想環境。

2.2. 後設認知理論與認知發展

後設認知分為知識的認知 (MK) 與認知的調節 (MR)。在程式學習中常觀察到學生具備策略知識 (MK)，實作中卻無法落實調節行為 (MR) 的「知行落差」。根據 Piaget (2009) 發展理論，國小學生多處於具體運思期，尚未完全具備抽象化能力。由於除錯需在腦中建構抽象執行模型，缺乏此能力的學生難以進行心理模擬，導致其實作仍訴諸對具體積木的隨機嘗試。因此，抽象化能力的不足被視為阻礙覺察轉化為調節實踐的關鍵瓶頸。

3. 研究方法

3.1. 研究對象與設計

本研究採用探索性之單組前後測準實驗設計。對象為 22 位國小二至六年級學生，針對樣本異質性，將年級視為關鍵變項，探討不同認知發展階段對學習成效之影響。受限於樣本數較小 ($N=22$)，本研究不進行強因果推論，而是透過質性行為軌跡補足統計效能之限制。

3.2. 實驗流程與教學設計

本研究實施為期五天的教學介入，採用專題式學習策略，課程以互動式數位故事為核心，將運算思維要素與後設認知之計畫、監控、評估歷程深度融合。詳細教學流程如表 3-1 所示。

表 3-1、教學架構表。

階段	課程內容重點	運算思維核心技能	後設認知目標
I	基礎操作、導入對話	模式識別、基礎演算法	計畫：目標、資源分配。
II	角色動畫、廣播機制	分解、抽象化	規劃：建構流程圖。
III	條件判斷、迴圈應用	演算法設計、模式識別	監控：測試邏輯、檢查變數。
IV	專案統整與偵錯	抽象化、偵錯	評估：全面測試與反思。

3.3. 研究工具與資料分析

本研究結合量化測驗、實作評量與質性行為編碼進行三角檢證。在量化測驗方面，採用 Sperling 等人 (2002) 發展、Kim 等人 (2017) 驗證雙因素結構之 18 題 Jr. MAI，以及 El-Hamamsy 等人 (2022) 開發之 17 題 cCTt。兩量表之原信度 Cronbach's α 分別為 .82 與 .85，針對本研究 22 位樣本重新測得 Jr. MAI 之 MK 與 MR 構面分別為 .89 與 .85，cCTt 測驗前後測信度分別為 .66 與 .76。確認其在本次小樣本施測之穩定性。在實作評量上，運用 Dr. Scratch 針對最終專案進行靜態程式碼分析，評估同步、抽象化與邏輯等七個維度；該工具主要反映「程式碼結構特徵」，作為外顯實踐表現之客觀代理指標，而非直接等同於內在的高階認知能力。

在質性資料與學習分析方面，本研究全程側錄學生螢幕畫面，並參考 Brennan & Resnick (2012) 與 Flavell (1979) 之理論架構建立行為編碼表。最後，利用 Orange 資料探勘平臺進行 K-means 聚類，輸入特徵設定為 MK/MR 增益值與實作得分，採歐氏距離度量，並透過側影係數 (Silhouette Score) 評估以決定最佳 K 值。此分析不僅協助鑑別具備知行落差特徵之目標群體，亦透過計算各樣本至群集中心 (Centroid) 的距離，客觀選出最具代表性之典型個案如 S005，作為後續微觀行為軌跡分析之主要對象。

4. 結果與討論

4.1. 整體學習成效分析

配對樣本 t 檢定分析教學介入之初步成效，結果如表 4-1 所示。學生在後設認知量表中的 MK 構面上呈現增長趨勢 ($t=2.337, p<.05, d=0.50$)，具中度效應量。然而，這種知識的增長並未同步反映在 MR 構面上 ($p>.05$)。此現象初步揭示了「知行落差」。MR 的不顯著可能受限於小樣本導致之二類錯誤風險。此結果初步顯示教學活動有助於建立學生的策略覺察，但在解釋上仍須考慮單組設計下可能存在的成熟效應等干擾因素。

表 4-1、運算思維與後設認知能力前後測成效分析 ($N=22$)

測驗維度	前測 M	SD	後測 M	SD	t	p
後設認知總分	58.95	12.87	60.86	11.36	1.796	.087
知識的認知(MK)	29.82	6.56	31.55	5.32	2.337*	.029
認知的調節(MR)	29.14	6.64	29.32	7.21	0.189	.852
運算思維總分	12.39	2.68	12.43	3.16	0.121	.905

* $p<.05$

4.2. 年級差異與能力瓶頸

分析結果顯示，年級變項差異達顯著水準（ $F=4.004, p<.05$ ）。其中以四年級學生的進步幅度最為顯著（ $M=+4.25$ ），顯示視覺化環境高度契合具體運思期之發展。五年級的負增益現象，則可能反映其對抽象邏輯有更高需求，或受限於課程難度未達其認知挑戰。

為進一步解析「知行落差」的成因，本研究利用 Dr. Scratch 進行運算思維維度的細部分析。如圖 4-1 所示，實作熱圖顯示學生在「同步（84.06%）」與「平行處理（82.61%）」維度表現最佳。而「抽象化（33.33%）」與「邏輯（17.39%）」維度表現欠佳。此數據反映學生在可觀察的程式結構上較少展現模組化特徵，而非直接代表內在認知能力的欠缺。這種缺乏模組化設計的傾向，增加了除錯時的認知負荷，可能是阻礙後設監控機制執行的主因。

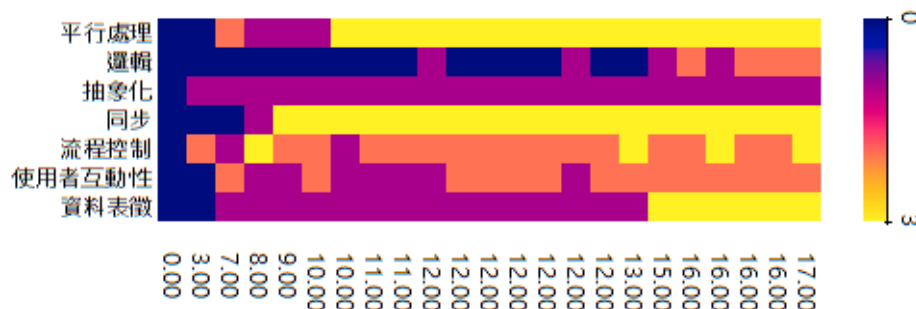


圖 4-1、學生於 Scratch 專案之各項運算思維維度表現熱圖 (N=22)

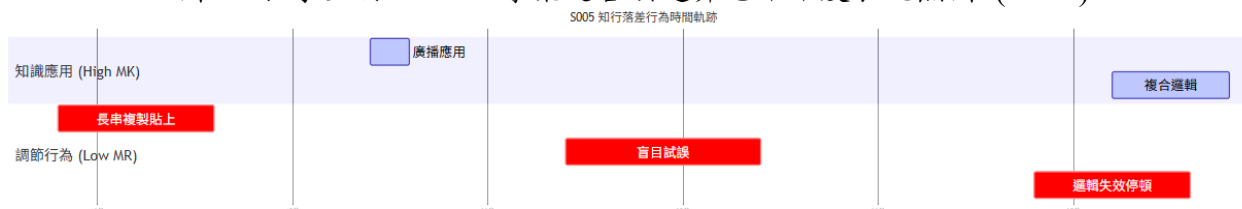


圖 4-2、S005 知行落差行為時間軌跡

4.3. 實作歷程之行為模式分析

為了進一步驗證上述推論，本研究結合 Orange 資料探勘與螢幕錄影編碼進行深入分析。本研究鑑別出具備高 MK 但低 MR 特徵的群體。並客觀選取最接近群集中心的典型個案 S005 進行微觀分析，探討其實作歷程中知行落差的核心機制：

1. 盲目試誤與監控機制缺失：研究發現，S005 在問卷中宣稱具備高度檢查能力，但在面對程式邏輯錯誤時，錄影顯示其行為表現出較強的衝動性特徵。如圖 4-2 所示，該學生在 5 分鐘內頻繁點擊執行並進行 12 次盲目試誤。此模式反映學生在實作當下缺乏有效調節（MR）來抑制衝動，使策略知識難以轉化為實質除錯行動。
2. 抽象化層次受限：S005 處理長腳本時僅仰賴複製貼上而非封裝自定義積木。此現象初步顯示，若未突破程式結構產出上的抽象化瓶頸，高階策略難以落實於實作，使運算思維停滯於淺層迭代。

5. 結論與建議

本研究結果顯示，教學介入初步提升了學生在知識的認知（MK）構面之表現，但並未同步反映於認知的調節（MR），突顯出學生雖具備策略知識，卻難以轉化為具體計畫與監控行為的知行落差現象。實作分析進一步指出，學生習慣於指令的線性堆疊，在抽象化與邏輯等程式碼結構特徵上遭遇瓶頸；缺乏模組化思維導致除錯時認知負荷過高，進而阻礙了後設監控機制的有效執行。此外，四年級學生的後設認知增益最為顯著，初步顯示此階段為導入結構化運算思維訓練的關鍵發展期。

基於上述探索性發現，未來教學設計應強化自定義積木與模組化訓練，以協助學生跨越知行落差。受限於單組前後測設計與極小樣本規模（N=22），本研究推論仍具侷限性。未來研究應擴大樣本並納入對照組，進行長期縱向實驗，以進一步驗證後設認知調節能力從短期覺察轉化為穩定實踐的長效影響。

參考文獻

- Brennan, K., & Resnick, M. (2012). New Frameworks for Studying and Assessing the Development of Computational Thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*, Vol. 1, Vancouver, 13-17 April 2012, 25 p. - References—Scientific Research Publishing. (n.d.). Retrieved March 27, 2026, from <https://www.scirp.org/reference/referencespapers?referenceid=3062876>
- El-Hamamsy, L., Zapata, M., Martín, E., Mondada, F., Dehler Zufferey, J., & Bruno, B. (2022). The Competent Computational Thinking Test: Development and Validation of an Unplugged Computational Thinking Test for Upper Primary School. *Journal of Educational Computing Research*, 60, 073563312210817.
- Flavell, J. (1979). Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry. *American Psychologist*, 34, 906–911.
- Kim, B., Zyromski, B., Mariani, M., Lee, S. M., & Carey, J. C. (2017). Establishing the Factor Structure of the 18-Item Version of the Junior Metacognitive Awareness Inventory. *Measurement and Evaluation in Counseling and Development*, 50(1–2), 48–57.
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61.
- Piaget, J. (2009). Intellectual Evolution from Adolescence to Adulthood. *Human Development*, 15(1), 1–12.
- Sperling, R. A., Howard, B. C., Miller, L. A., & Murphy, C. (2002). Measures of Children's Knowledge and Regulation of Cognition. *Contemporary Educational Psychology*, 27(1), 51–79. <https://doi.org/10.1006/ceps.2001.1091>
- Wing, J. M. (2006). Computational thinking. *Commun. ACM*, 49(3), 33–35.