

基於 TRIZ 發展生活巧思樣式語言

A Pattern Language for Ingenuity for Life based on TRIZ

郭靖玫	周忠信	鄭有進	黃義閔
Ching-Wen Kuo	Jung-Sing Jwo	Yu Chin Cheng	Yi-Ming Huang
東海大學	東海大學	台北科技大學	東海大學
資訊工程系	資訊工程系	資訊工程系	資訊工程系
g03357001@thu.edu.tw	jwo@thu.edu.tw	yccheng@ntut.edu.tw	g03357005@thu.edu.tw

摘要

隨著科技進步及經濟發展，人類對於生活品質要求越來越高，「生活巧思」在日常生活中逐漸發酵，「生活巧思」可視為「創新」其中一個小環結，TRIZ 理論運用在各個領域上做為創新的工具已有數十年的歷史，其效果不容置疑。TRIZ 發展出很多套解決問題的工具，其皆為解題過程中可以運用的方法，想要使用工具解決問題必定要先熟悉工具使用方式。再則 TRIZ 工具通常一次只能解決一個問題，所以一個發明問題的整體解決方案，僅由 TRIZ 工具並無法獲知。

本論文首先引用了單一工程特性法則，將問題的分析變得更直觀。再使用樣式語言(Pattern Language)重新詮釋 TRIZ，發展出能為生活問題提出巧妙解決方法的「生活巧思樣式語言(A Pattern Language for Ingenuity for Life，縮寫為 PLIL)」。透過 PLIL 解決日常生活的難題將變得更有效率。不再需要經過 TRIZ 繁雜的解題過程就能為生活帶來更多的便利。

關鍵字：TRIZ、創新發明、樣式語言、40 項創新發明原則、39 項工程參數、矛盾矩陣、單一工程特性、生活巧思

一、緒論

在早期的社會，人們對於生活上的不便，往往有很大的容忍度。但隨著科技的進步及經濟的發展，人類對於生活品質的要求越來越高。針對日常生活中不便的事物，都期望可以有更好的解決方式。因此，「生活巧思」一詞成為近年來非常熱門的詞彙。西門子(Siemens)公司更以“Ingenuity for life”做為全新品牌的宣言。「巧思」其意為精巧的構思，故利用巧妙的想法來解決日常生活的難題稱之為「生活巧思」。「生活巧思」一詞似乎是承接「創新」一詞而來的。「創新」指的是利用現有的知識和技術提出有別於常人的見解，進而能改善或創造新的事物，以滿足理想中的需求。「創新」的範圍較為廣大，在各領域都能被談論。而本論文的範圍將僅限為「創新」運用在生活的環結，故引用「生活巧思」一詞。

提到「創新」，TRIZ 理論[1]在創新發明領域中，占有很重要的地位。TRIZ 是近年來廣為討論並逐

步為企業接納的創新發明方法。TRIZ 是由舊蘇聯時期學者 Altshuller[2]分析了近 250 萬份高水平的發明專利後，將這些專利發明的經驗以邏輯歸納的方式所創造出一套發明問題解決方法，經過數十年打磨，已廣為全世界各領域所接納。

TRIZ 分析及解決問題的工具很多，至今已被大量的運用在產品設計[3]、工業流程改善[4]、企業管理[5]，甚至到軟體架構[6]和教育方法[7]等領域。TRIZ 對於創新及解決問題的貢獻早已不容置疑，而其中最容易理解及最常被使用的工具，就屬矛盾矩陣[8]、39 項工程參數及 40 項創新發明原則[9]。這也是本研究主要使用的核心工具。

即便矛盾矩陣、39 項工程參數及 40 項創新發明原則已被視為 TRIZ 中最容易使用的方法，但其中還是有運用上的難題。劉志成[10][11]發現在解決產品問題的過程中，常會有矛盾訊息無法被確定的狀況。故劉志成提出改變矛盾矩陣的使用方式，發展出單一工程特性創新法則(2001)。其概念為將想改善或是避免惡化的工程參數對應 40 項創新發明原則出現的次數，次數越高表示此發明原則越能解決目前工程參數問題。單一工程特性創新法則大幅的提升了工程參數及發明原則的使用性，為不少工程師省去分析及尋找產品矛盾的過程。

由上論述，我們已經了解針對產品問題運用 TRIZ 工具尋求解決方法的其中一個途徑。但綜觀整個創新發明的過程，會遇到的技術問題通常不會只有一個。使用 TRIZ 工具，一次只能解決單一技術問題，導致整個解題的過程中，必須反覆的操作 TRIZ 工具，雖然最終還是可以藉由 TRIZ 工具的輔助找出創新發明的解決方法，但整個過程確也相當繁雜。為了改善 TRIZ 工具使用上的效率，本研究引用了建築設計學家 Alexander[12][13]在 1979 年提出的樣式語言(Pattern Language)，針對 TRIZ 工具中的矛盾矩陣、39 項工程參數及 40 項創新發明原則解題方法重新做一個詮釋。

樣式語言和 TRIZ 的關聯性已經有人提出來討論[14]。他們的核心精神都是為了讓一再重複發生的問題，可以尋求前人的解題經驗而得到答案。不一樣是，Altshuller 是利用專利分析發展出 TRIZ 解題工具，而 Alexander 則是利用描述建築空間設計問題及解法，發展出一套有既定格式的問題描述方式稱為樣式語言。

樣式(Pattern)是以一種正規的表達方式，可以

將問題及解決方法清楚表達，而樣式語言是由多個相關或是解決共通問題的樣式集結而成。因此一個完整的樣式語言，描述了整個解決問題方法的全貌。這可以幫助人們在解決問題的過程中，很快找到合適的解決方法，盡量避免使用錯誤或是不合適的方法。這個特點恰巧就是使用 TRIZ 工具解題無法達到的效益。

本研究將以解決 39 項工程參數問題為目標，制訂出 39 個樣式。這 39 個樣式將可以幫助解決生活中許多不便利的問題，進而發想出具有巧思的創新產品。「巧思」其意為精巧的構思，本論文之樣式語言能為生活問題提出巧妙的解決想法，故本論文將之稱為「生活巧思樣式語言 (A Pattern Language for Ingenuity for Life, 縮寫為 PLIL)」。透過 PLIL，可以更有效率的解決日常生活的不便及不理想，將不再需要透過 TRIZ 繁雜的解題過程，為生活帶來更多的便利。

二、文獻探討

2.1 TRIZ 理論

TRIZ 是由俄文 теории решения изобретательских задач 音譯成英文 Teoriya Resheniya Izobreatatelskikh Zadatch 的縮寫，其意思為“發明問題的解決理論 (Theory of Inventive Problem Solving)” [15]。歐美國家也有人稱為 Theory of the Solution of Inventive Problems，其縮寫為 TIPS [1]。

TRIZ 理論是由舊蘇聯時期的發明家 Altshuller 所創建，當時 Altshuller 在前蘇聯海軍的專利局工作。他總是思考著：人們在進行發明及解決技術問題時，是否有跡可循？而這個答案在他之後數十年的研究中被證實了，故 Altshuller 被尊稱為 TRIZ 之父。

Altshuller 從 1946 年開始，帶領數十家研究機構、大學及企業組成了 TRIZ 的研究團體，經過幾十年的時間，分析了世界近 250 萬份高水平的發明專利後歸納出一套邏輯化發明理論，就是現在被企業界廣為討論的 TRIZ 理論。

TRIZ 理論有幾個主要的分析及解決問題的工具 [16]：

- 矛盾矩陣 (Contradiction Matrix)
- 39 項工程參數 (Engineering Parameters)
- 40 項創新發明原則 (Inventive Principles)
- 物質-場分析 (Substance-Field Analysis)
- 76 標準解答 (76 Standard Solutions)
- 8 種技術演化類型 (8 Evolution of Technological Systems)
- 科學技術效應 (Scientific Technological Effects)
- 最終理想結果 (Ideal Final Result)
- ARIZ 發明問題解決演算法 (Algorithm to Solve an Inventive Problem)

TRIZ 的工具很多，但最常被使用的就屬矛盾矩陣、39 項工程參數及 40 項創新發明原則，因為

其用法頗為簡單且容易理解。如圖 1 所示，首先將自己領域碰到的技術問題，抽象化並轉成為 TRIZ 的標準問題。此時運用 TRIZ 矛盾矩陣，可以獲得 TRIZ 40 項發明原則中推薦的原則，透過標準問題與推薦原則，再將之類比並轉化成解決領域問題的方法，進而發想出可以滿足需求的創新發明。



圖 1 TRIZ 步驟

2.2 單一工程特性創新法則

TRIZ 矛盾矩陣是用來解決產品的技術矛盾的工具。它是一個 39x39 的矩陣，縱軸為欲改善的工程參數，橫軸為導致惡化的工程參數，而中間對應的數字為可以解決工程參數惡化的發明原則。例如：希望物體的體積變大，卻不要物體的重量增加，這情況可以對應到發明原則 2 抽離，將物體內不需作用的部份抽離，這樣既可以保留外部的體積大小不變，又可以讓整體重量不惡化。在運用矛盾矩陣來解題的過程中，矛盾通常是在改善某項工程參數時才會出現。如果在改善某項工程參數時，能夠預知另一項工程參數會惡化，這時可以使用矛盾矩陣來解決問題。一般的狀況常常是，工程師知道要改善某項工程參數，不清楚或是無法預知哪項工程參數會惡化。亦或是只清楚要避免某項工程參數惡化，但確不清楚哪些工程參數是造成惡化的原因。這種情況下就無法使用矛盾矩陣，故劉志成 [10][11] 提出了缺乏矛盾訊息下可以使用的單一工程特性創新法則。

單一工程特性創新法則表中間的數字代表 TRIZ 創新發明原則編號，縱軸為欲改善或避免惡化的 TRIZ 工程參數，橫軸為對應工程參數，創新發明原則出現的次數並以等級劃分，其中 A(19 次以上)，B(16~18 次)，C(13~15 次)，D(10~12 次)，E(7~9 次)，F(4~6 次)，G(1~3 次)，等級越前面代表被使用的次數越多，也表示此原則越可能解決目前工程參數的問題，如表 1。單一工程特性創新法則使用方式是找出欲改善的工程參數或是避免惡化的工程參數，再查詢單一工程特性創新法則表，找出等級最高的創新發明原則來做為解決工程參數問題的最優先考慮。不少使用 TRIZ 進行產品設計的論文 [17][18] 都將其納入產品設計流程中，有助於解決產品設計過程中找不到矛盾的瓶頸。

等級		A (19 次以上)	B (16~18 次)	C (13~15 次)	...
工程參數					
1	移動件重量	35		28	...
2	固定件重量	35	28,10,19,1,26	26	...
3	移動件長度	1,29	15	35,4,17	...
...
37	控制複雜性	35	28	27,26	...
38	自動化程度	35		2,28,26	...
39	生產性	35,10,28		1	...

表 1 單一工程特性創新法則表(部份)

2.3 樣式語言(Pattern Language)

樣式語言一詞是由建築設計學家 Alexander 在 1979 年提出的。Alexander 發現在建築設計中有許多相同的問題一再發生，即便是重覆發生，設計師還是不能避免發生錯誤。故他提出了一個樣式的概念。樣式具有一種正規格式的表達方式，可以清楚描述一個問題及其解決方法。因此藉由樣式可以將設計師的解題經驗清楚記錄下來並分享，幫助其他遇到相同問題的設計師可以很快的尋求合適的解決方案。樣式語言是由多個相關或是解決共通問題的樣式集結而成，透過樣式語言可以將問題關聯到其他樣式，以解決共通的問題，或是一個大問題裡的部份解決方案。

從 1994 年開始在美國舉辦的 PLoP (Pattern Language of Programming) 研討會是樣式語言的重要會議。由 PLoP 討論的經驗看來，樣式的寫作沒有一定正確的方法，但樣式語言是否可用經常取決於樣式的表達方式。Meszaros[19]以在 PLoP 的經驗整理了樣式的寫作技巧及方法，並以樣式語言的方式來表達及詮釋樣式。其針對樣式寫作項目的定義大略如下，樣式項目名稱將會以粗體字表示：

- 樣式必須包含項目：
 1. **樣式名稱(Pattern Name)**：由**問題及解決方案**配對出可以被參考的命名。
 2. **背景(Context)**：該**問題**是在何種情況下的解決方案。**背景**通常是以一個「情況」來描述，而非明確的說明。有時也會引用一個樣式來描述。**背景**會關係到**力**的描述。
 3. **問題(Problem)**：需要解決的具體問題。必須在**背景**限制下，以確保該**問題**是被保持在**解決方案**的範圍內。
 4. **力(Forces)**：在選擇一個**問題**的**解決方案**時，經常會有模糊的情況，所以藉由**背景**的暗示來描述**力**，使其可以關聯到**解決方案**。
 5. **解決方案(Solution)**：提出**問題**的解決方法。**問題**可能會有一個以上的**解決方案**，而選擇好的**解決方案**需考慮到**問題**的**背景**。**解決方案**需考慮**力**的影響，解決

某些**力**會有其他代價，某些則無。最佳**解決方案**就是解決由**背景**決定最需優先解決的**力**。

- 如果有助於提升樣式理解，可以選擇性添加下列項目：
 1. **表徵(Indications)**：表示可能存在問題的徵兆。
 2. **結果上下文(Resulting Context)**：在應用了該樣式之後，有了一個或多個新問題需要解決。這需要運用更多的樣式，可能是下一個樣式語言。
 3. **相關樣式(Related Patterns)**：其他讀者可能感興趣的樣式，包括：同樣問題的其他解決方案、更通用或是具體的樣式、解決該問題產生結果上下文的問題使用樣式。
 4. **示例(Examples)**：樣式應用的實體示例。
 5. **代碼示例(Code Samples)**：演示如何實現樣式的代碼示例。
 6. **理由(Rationale)**：解釋為何該解決方案為該背景之下問題的最佳解決方案。
 7. **別名(Aliases)**：該樣式其他已知的名稱。
- 樣式語言是讓讀者透過**樣式名稱**、**問題**及**解決方案**就可以了解這個樣式是否可能為讀者所需要的解決方案，再透過**背景**及**力**來確定樣式是否合適，最後可以透過**理由**、**結果上下文**、**關聯樣式**及**示例**的部份來獲取樣式的進一步資訊。

三、研究方法

本論文的目的是使用樣式語言做為表達方式，基於 TRIZ 理論發展出生活巧思樣式語言(PLIL)。PLIL 預計將提出 39 個樣式，分別用以解決 39 項工程參數衍生的技術問題。解決方案是依據單一工程特性創新法則推薦出最常使用的發明原則為基礎來描述。這有助於提升 TRIZ 理論的可讀性及使用性。礙於篇幅，本論文僅先呈現其中五個完整樣式，其工程參數對應樣式名稱及其使用之發明原則如表 2 所示。在第四節將舉出二個日常生活問題為例，演示如何運用 PLIL 樣式發想出具有巧思的問題解決方案，進而發明出創新產品。

項號	工程參數	樣式名稱	優先使用發明原則
3	移動件長度	橫材可入灶	1, 29, 15, 35, 04, 17...
25	時間浪費	分秒必爭	10, 35, 28, 18, 4, 32...
26	物質數量	寡能敵眾	35, 3, 29, 18, 10...
29	製造精確度	造工精確	32, 28, 10, 18...
35	適合性	因地制宜	35, 15, 1, 29...

表 2 工程參數對應 PLIL 之樣式名稱

3.1 樣式 1：橫材可入灶

- 問題：作用時會移動的物體，其物體本身的長度或是作用的距離範圍如何可以不受到限制？例如：橫向的木材如何可以放進爐口比木材窄爐灶裡面，或是如何可以畫出不受限制長度的直線(一般的尺可能只有15或30公分)？
- 背景：
 1. 移動狀態的物體，物體長度造成其他影響。
 2. 物體可移動的距離受到其他因素所限制。
- 力：
 1. 移動狀態的物體的長度如果沒有限制的話，會造成其他問題的衍伸。例如：重量會變重、面積會變大、體積會變大、不方便使用等等。
 2. 物體可移動的長度範圍如果受到限制，會造成其他問題。例如：不好操控、不好移動、不方便使用等等。
- 解決方案：
 1. 可以將長度分割成區段，必要使用的時候可以很容易組裝回去。
 2. 讓長度變得有彈性，可以使用液體或氣體的材質來取代原本固體的材質。
 3. 做成可以動態伸縮的裝置。
 4. 改變其他條件來讓長度可以變化。例如增加水壓可以讓水噴的高度更高。
 5. 做成非對稱的長度設計。
 6. 改變長度的維度。例如河道截彎取直，河水可以更快的流入大海。

3.2 樣式 2：分秒必爭

- 問題：如何讓完成工作所花費的時間更精簡？
- 背景：完成一個工作的過程中，可能會需要一項或多項操作，全部操作完成的連續時間計算視為所花費的時間。
- 力：
 1. 工作的過程中，各項因素造成的時間花費變長。例如：操作項目多、操作複雜、面積大作用時間長、資訊不足辨識時間長等等。
 2. 工作過程中包含沒有效益的時間浪費。例如：項目與項目間的等待、物體與物體間不需作用的間隙、系統出錯的復原時間等等。
- 解決方案：
 1. 改變其他條件，以加快操作的時間。例如：增加系統動力、提高振動頻率、增加色差。
 2. 改變系統的操作方式以優化時間花費。例如：改變流程，以非對稱流程取代對稱流程。
 3. 不相關的工作可以預先並行完成，減少

項目間的等待時間。

3.3 樣式 3：寡能敵眾

- 問題：如果數量因需求而改變，如何不改變數量也能滿足需求？
- 背景：無論是工作量、或是物體數量、物料量，會因需求而改變。例如：想要加速生產，會需要設置更多的產線或更多人力。
- 力：
 1. 需求範圍擴大時，需要運用的資源種類數量加增。例如：更多資訊、更多不同設備、更多技術等等。
 2. 同樣的需求但數量提升，原既有的資源數量也要提升。例如：增加產線、增加原料、增加時間、增加設備等等。
- 解決方案：
 1. 以極少數量就能完成多項需求。例如：一個設備能局部執行不同工作，。
 2. 減少資源數量消耗。例如：固體資源的消耗一次就是一個固定量，如果使用液體或氣體取代，可以更彈性的微量調整消耗數量，減少浪費。
 3. 提升工作執行次數。例如使用振盪方式來執行工作。
 4. 預先執行工作或安置元件，當需求改變時，不需再調整現狀。

3.4 樣式 4：造工精確

- 問題：如何讓操作之後的結果，或是生產出來的產品更符合需求？
- 背景：操作的結果或生產出來的產品是否符合需求會影響其整體效益。
- 力：
 1. 人為操作造成任何可測量值的誤差。例如：重量、長度、面積等等。
 2. 其他外部因素會造成生產過程失去標準。例如：資訊喪失、有害物質產生、元件穩定性低等。
- 解決方案：
 1. 降低任何可能造成人為或機器的偏差因素。例如：改變顏色或增加色差、預先安置輔助元件或作用、以振盪的方式作用等等。
 2. 改變系統的運作方式以防止生產過程失去標準。

3.5 樣式 5：因地而異

- 問題：在外部環境的變化下，如何讓物體也能隨之調適？
- 背景：外部環境變因太多，元件必須滿足隨時改變的需求。
- 力：
 1. 元件因受作用的物體特性不同而改變其操作。例如長度不同、面積大小不同、可承受力不同等等。
 2. 不論外部的環境改變，作用的結果必須

一致。例如：外部溫度變化、空間變化、壓力變化等等。

3. 因應隨時都在改變的需求。例如：不論高矮胖瘦的人坐同一張椅子都要能舒適。
- 解決方案：物品的任何性質都可以考慮設置成可以隨時動態調整。例如：使用液體或氣體會比固體更有彈性調整，或將物體分割成更細的區塊，使其更容易調整。

四、研究結果-生活實例演示

4.1 案例 1

- 情境：

如果想對紙做直線剪裁，以往一般做法有兩種：1.拿筆跟尺先將直線畫在紙上，再用剪刀依直線做裁剪。2.拿尺、美工刀和墊板，直接用美工刀靠著尺做切割。依照上述的做法會發現，整個過程是繁雜費時的，且裁切出來的直線常常會有偏差。

- 案例分析及使用 PLIL 解決方法：

由上面情境將問題分析如圖 2 所示。過程操作繁雜造成時間浪費的問題，可以使用 PLIL **分秒必爭**樣式來解決。裁切直線偏差的問題可以使用 PLIL **造工精確**樣式來解決。而另一個較細部的問題是，在使用尺來輔助直線裁切，裁切的長度會受到尺的長度所限制，如果裁切的長度較長，就必須反覆的將尺對齊再做裁切，這個問題可以使用 PLIL **橫材可入灶**樣式來解決。以下將藉由一個創新產品來做應用的說明。

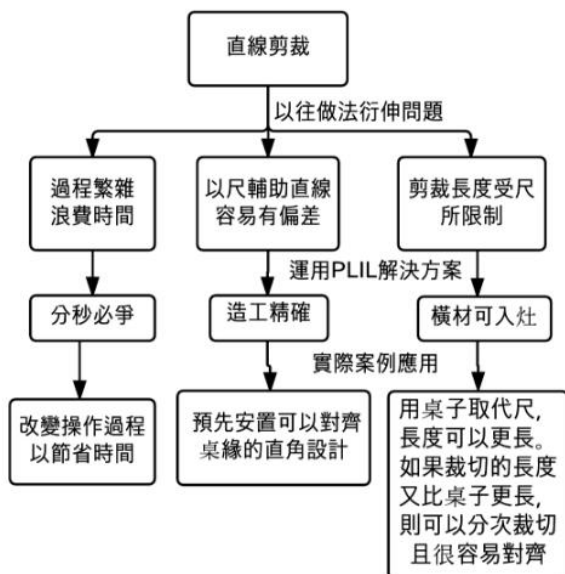


圖 2 運用 PLIL 解決直線剪裁問題

- 創新產品應用說明：

直線剪刀(Vector scissors)[20]是由匈牙利產品設計師 Fekete Tamas 所研發的，如

圖 3 所示。他將剪刀握柄的地方多做了一個可以對齊桌緣的直角設計，如圖 4 所示，改變操作，以桌子取代尺將更容易對齊，因此可以節省操作時間。使用剪刀直接對齊桌緣，比起以平面的尺來對齊，造成直線偏差的人為因素大幅降低。用桌子取代尺，長度可以更長。如果裁切的長度又比桌子更長，則可以分次裁切且很容易對齊，如圖 5 所示。



圖 3 Vector scissors 產品實照



圖 4 Vector scissors 對齊桌緣的直角設計

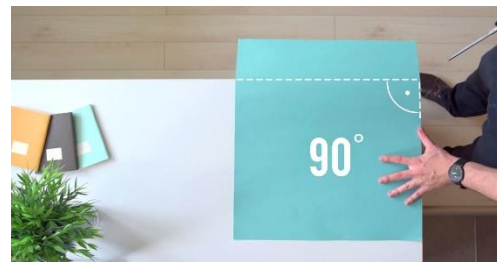


圖 5 利用桌緣來對齊紙張的邊緣是非常容易的

4.2 案例 2

- 情境：

午後的時光想要和朋友坐在花園裡享受咖啡，擺設足夠的椅子是必要的，但椅子足夠了卻又缺少了擺放咖啡的桌子。如何可以在有限的空間裡，適時的滿足坐位數量，又可以滿足放置咖啡的需求？

- 案例分析及使用 PLIL 解決方法：

由上面情境將問題分析如圖 6 所示。需要足夠椅子和桌子的問題可以使用 PLIL **寡能敵眾**樣式來解決。另外可以分析的問題是，坐位和擺放咖啡的空間是否可以隨情

況做調整，這時可以使用 PLIL 因地而異樣式來解決。

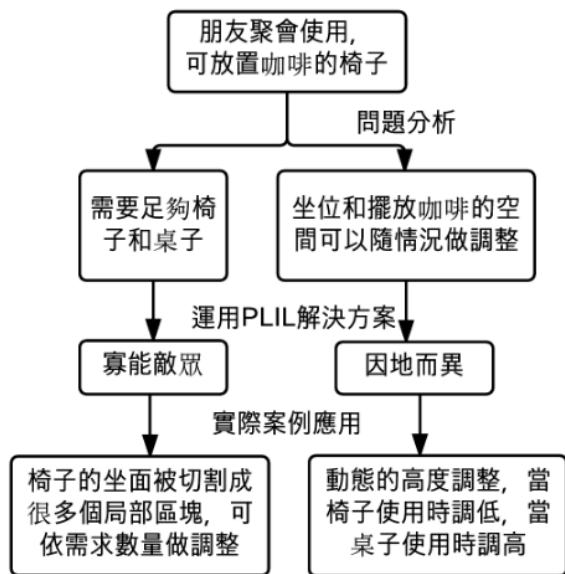


圖 6 運用 PLIL 解決咖啡椅問題

● 創新產品應用說明：

Coffee Bench by BEYOND Studio[1]是由波蘭 BEYOND 工作室的 Karolina Tylka 設計師所設計，想法是要讓椅子空間可以更智慧的靈活使用。如圖 7 及圖 8 所示，椅子的坐面被切割成很多個局部區塊，這些區塊可以動態的旋轉調節平面的高度，要當坐位使用時，就將高度調低，要當桌子或手把使用時，就將高度調高，使用面積可視需求做調整。



圖 7 Coffee Bench 可以智慧的旋轉調節



圖 8 Coffee Bench 可以靈活調整坐位及桌子的使用面積

五、結論與建議

本論文利用 TRIZ 的理論基礎，使用樣式語言重新詮釋成生活巧思樣式語言 (PLIL)，並完整呈現出其中五個樣式。接著運用 PLIL 的樣式來與日常生活遇到的問題做關聯、分析及找出解決方案。最後舉出創新產品真實案例，藉以證明其方案的可行性。由此說明 PLIL 做為日常生活問題的解決及創新產品的設計是有效的。

日後的研究將會著重於將 39 個樣式完整詮釋，並對每個樣式找出運用的創新產品實例，以提升 PLIL 的可讀性及使用性。而是否能真實運用達成創新產品的研發，也將是我們未來納入研究的一個方向。

本論文目前的階段都是將工程參數問題獨立討論，未來也會考慮到 39 個發展出來的樣式是否具有關聯性。能否在使用某項樣式時，也能關聯到另一個樣式來做輔助。例如，時間問題和速度問題是否相關，面積問題和壓力問題是否相關，這都是可以研究的。相信做完後續的研究工作，以 PLIL 的便利性，一定有助於提升及改善當前的生活品質。

參考文獻

- [1] Barry, K., Domb, E., & Slocum, M. S. (2010). TRIZ-What Is TRIZ?. The TRIZ Journal, 603-632.
- [2] Lerner, L. (1991). Genrich Altshuller: father of TRIZ. Russian Magazine Ogonek, <http://www.aitriz.org>.
- [3] 鄭勝隆. (2004). 以 TRIZ 輔助汽車內裝產品設計之研究. 設計研究所在職專班碩士論文 (民國 93 年).
- [4] 潘進丁. (2006). 應用 TRIZ 方法探討產品開發流程之改善—以筆記型電腦關鍵組件樞軸 (Hinge) 為例. 國立交通大學管理學院碩士在職專班工業工程與管理組碩士論文 (民國 95 年).
- [5] 朱映慈. (2006). TRIZ 四十原則對應於企業管理之探討. 國立交通大學經營管理研究所碩士論文 (民國 95 年).
- [6] Kluender, D. (2011). TRIZ for software architecture. Procedia Engineering, 9, 708-713.
- [7] 施乃瑜. (2013). 網路化 TRIZ 教學對國中學

生問題解決表現影響之研究. 臺灣師範大學科技應用與人力資源發展學系學位論文(民國 102 年).

- [8] Domb, E., Miller, J., & MacGran, E. (1998). The 39 features of Altshuller's contradiction matrix. *The TRIZ Journal*, 11, 10-12.
- [9] Tate, K., & Domb, E. (1997). 40 inventive principles with examples. *The TRIZ journal*.
- [10] 劉志成.(2003). 「TRIZ 方法改良與綠色創新設計方法之研究」. 國立成功大學機械工程學系碩博士班論文(民國 92 年).
- [11] Liu, C. C., & Chen, J. L. (2001). A TRIZ Inventive Design Method without Contradiction Information, *The TRIZ Journal*.
- [12] Alexander, C., Ishikawa, S., & Silverstein, M. (1977). *A pattern language: towns, buildings, construction* (Vol. 2). Oxford University Press.
- [13] Alexander, C. (1979). *The timeless way of building* (Vol. 1). New York: Oxford University Press.
- [14] Najari, A., Dubois, S., Barth, M., & Sonntag, M. (2016). From Altshuller to Alexander: Towards a Bridge between Architects and Engineers. *Procedia CIRP*, 39, 119-124.
- [15] Hua, Z., Yang, J., Coulibaly, S., & Zhang, B. (2006). Integration TRIZ with problem-solving tools: a literature review from 1995 to 2006. *International Journal of Business Innovation and Research*, 1(1-2), 111-128.
- [16] Philatov, V., Zlotin, B., Zusman, A., & Altshuller, G. (1999). *Tools of classical TRIZ*. Ideation Intl Inc, Detroit.
- [17] 丁怡文.(2006). 運用 QFD 與 TRIZ 在運動休閒服飾創新設計之研究. 大葉大學工業工程與科技管理學系碩士論文(民國 95 年).
- [18] 劉明盛, 吳芳瑩, 李詩涵, 許萍鳳, 李佳恩, & 洪紫瑜. (2013). 應用 TRIZ 創新策略於改善產品的功能—以打蛋器為例. *Int. J. Systematic Innovation*, 2(3), 13-25.
- [19] Meszaros, G., & Doble, J. (1998). A pattern language for pattern writing. *Pattern languages of program design*, 3, 529-574.
- [20] INDIEGOGO, Vector - Straight Cutting Scissors, <https://www.indiegogo.com/projects/vector-straight-cutting-scissors#/>
- [21] Jaime, D. Coffee Bench by BEYOND Standards, <http://design-milk.com/coffee-bench-by-beyond-standards/>