

國立交通大學

土木工程學系

博士論文

動態想法地圖：探討輔助分散式想法連結之研究

Dynamic Idea Maps - A Study in Supporting Distributed Idea Linking



博士生：賴怡成

指導教授：張登文、劉育東

中華民國九十五年六月

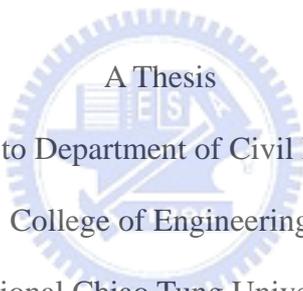
動態想法地圖：探討輔助分散式想法連結之研究

Dynamic Idea Maps - A Study in Supporting Distributed Idea Linking

研究生：賴怡成
指導教授：張登文
劉育東

Student : Ih-Cheng Lai
Advisor : Teng-Wen Chang
Yu-Tung Liu

國立交通大學
土木工程學系
博士論文



A Thesis
Submitted to Department of Civil Engineering
College of Engineering
National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Doctor

In

Architecture

June 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年六月

動態想法地圖：探討輔助分散式想法連結之研究

學生：賴怡成

指導教授：張登文、劉育東

國立交通大學土木工程學系博士班

摘 要

設計本身是一種設計者與想法之間的互動性行為。在早期設計階段中，此互動性行為常透過二人以上的參與者共同討論而產生多樣性的設計想法，而想法聯想的連結（linking）與分散性（distribution）問題是了解互動性行為的關鍵。針對此分散性的想法連結問題，本研究建構一個名為「動態想法地圖」(Dynamic Idea Maps, 簡稱 DIM) 的運算模型，試圖建構及了解上述問題的連結機制與連結有效性，進而在分散式的設計環境中，輔助多重參與者產生多樣性且關聯性的設計想法。

本論文的研究步驟，經由三個階段進行：文獻回顧與模型建立，系統實作，及實驗檢驗。在第一階段，經由背景研究與文獻回顧，探索與建立分散式想法連結的主要機制：呈現、回憶與溝通，這些機制結合案例基礎推理與軟體代理人技術，提供本研究建構 DIM 的基礎。在第二階段，DIM 架構在一個多重代理人平台 (DARIS)，以具體發展成一個輔助分散性想法連結系統。在第三階段，本研究實際執行一個設計實驗，予以驗證 DIM 之運算的可行性與連結的有效性。

經由上述的研究方法，最後的結果證明想法聯想的分散性連結問題，可以經由運算的方式了解並實踐，同時，其所發展的輔助系統有助於參與者在分散性的設計環境中，產生多樣性的想法與想法之間的有效性連結。此外，在研究過程中的發現，也幫助了解分散式想法連結所伴隨在設計運算上的複雜性問題，而這些問題將提供作者未來之後續研究。

關鍵字：想法聯想、分散性連結、運算模型、案例基礎推理、軟體代理人

Dynamic Idea Maps - A Study in Supporting Distributed Idea Linking

Student: Ih-Cheng Lai

Advisor : Dr. Teng-Wen Chang
Dr. Yu-Tung Liu

Department of Civil Engineering
National Chiao Tung University

ABSTRACT

Design is an interactive behavior involving the evolution of ideas between two or more participants in discussion, especially during the early design stage. Idea association is an important behavior for generating diverse ideas through the dynamic exchange of varied knowledge possessed by the participants. Linking and distribution within idea association are the key issues for understanding the interactive behavior. To understand the mechanism and effectiveness of the distributed linking, this research proposes a computational model named Dynamic Idea Maps (DIM), which supports participants to generate correlated and diverse ideas in a distributed design environment.

There are three research steps in this thesis. By applying the technology of case-based reasoning and software agent, the first step is to establish the mechanisms of the distributed idea linking, which includes representation, recalling and communication. Furthermore its computational model is proposed. In the second step, some computational components within DIM are computed for supporting distributed idea linking by using a multi-agent framework (DARIS). In the third step, the computational feasibility and effective linking of DIM are evaluated through a design experiment.

Consequently, this research indicates that the distributed idea linking of idea association can be understood through DIM. Also, the DIM system can support participants to generate diverse ideas and effective linking in a distributed design environment. Following the development of DIM, future studies such as agent learning, case adaptation or supporting the creative problem solving can be pursued.

Keywords: idea association, distributed linking, computational model, case-based reasoning, software agent

誌 謝

經過四年的煎熬，終於畢業了！

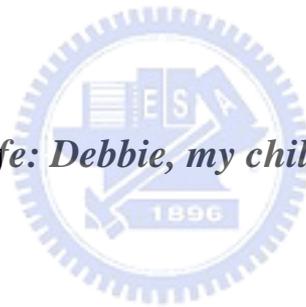
首先，感謝上天，在您的照顧下讓我得到這些人的幫助，也讓我勇敢的堅持完成學業。接著，由衷的謝謝對我學術上付出的一群人：指導教授 - 張登文老師，謝謝您的耐心指導，讓一個沒有運算概念的我可以順利完成學位，也是我在這個專業領域的引導員。建築研究所前所長（口試委員之一） - 劉育東老師，您大力支持我進入交大建築研究所，提供一個具有挑戰性的學術環境鞭策我，不僅開啟我研究的視野，也拓展我建築專業領域的經驗。

感謝口試委員 - 邱上嘉老師、郭中人老師及葉李華老師，各位的寶貴意見除了啟發我另一層次的思考，也讓我的博士論文更加紮實。另外，感謝陳珍誠老師及簡聖芬老師對於論文的建議，足以讓論文內容更為周延。

感謝政瑋及勝誠，由於二位程式寫作上的協助與指導，論文才可以順利進行。還有感謝一群認真參與實驗的朝陽科技大學建築系學生，因為你們的加入，老師才可以快速畢業。

最後，感謝一路相挺的家人 - 我的父母、孩子們及牽手，沒有他們的支持與容忍，很難熬過這四年，希望您們分享到這份喜悅。

For my parents, my wife: Debbie, my children: Jesse and Pauline



目 錄

一、緒論

1.1 想法聯想與連結	2
1.2 數位技術與連結	3
1.3 研究方法	5
1.3.1 研究方法回顧	5
1.3.2 研究方法建立	5
1.4 論文概述	7
1.4.1 研究問題	7
1.4.2 研究假設	8
1.4.3 研究目標	8
1.5 論文內容架構	8

二、研究背景：從設計搜尋到分散式想法連結

2.1 設計搜尋	10
2.2 連結與呈現	12
2.2.1 連結知識呈現	13
2.2.2 概念地圖	14
2.2.3 設計案例呈現	16
2.2.4 媒材表現	17
2.3 連結與規則	18
2.3.1 想法聯想的三個原則	19
2.3.1.1 相似原則	20
2.3.1.2 對比原則	20
2.3.1.3 相鄰原則	21
2.3.2 語言行動	22
2.4 連結與記憶	24
2.4.1 記憶組織	25
2.4.2 案例回憶	25
2.4.2.1 索引	26
2.4.2.2 檢索	26
2.4.2.3 選擇	27
2.5 連結與決策	28
2.5.1 角色扮演	28
2.5.2 相互作用	30
2.5.3 代理人	32



2.5.3.1 代理人與分散式設計	32
2.5.3.2 代理人溝通語言	33
2.6 小結	35

三、文獻回顧：探索輔助分散式想法連結系統

3.1 輔助呈現	37
3.1.1 Archie-II	38
3.1.2 PRECEDENTS	39
3.1.3 FABEL	42
3.1.4 WEBPAD	43
3.2 輔助回憶	44
3.2.1 Archie-II	44
3.2.2 PRECEDENTS	45
3.3.3 FABEL	47
3.4.3 DIM (初期研究模型)	48
3.3 輔助溝通	49
3.3.1 ADLIB	50
3.3.2 MADS	51
3.3.3 CBR-TEAM	52
3.3.4 DARIS	52
3.4 邁向一個輔助分散式想法連結模型	54



四、DIM-動態想法地圖

4.1 元素	58
4.2 知識	59
4.2.1 想法元件	60
4.2.2 連結	61
4.2.2.1 屬性連結	61
4.2.2.2 元件連結	62
4.2.3 想法地圖	64
4.2.3.1 想法地圖定義	64
4.2.3.2 IE 地圖	66
4.2.3.3 ICF 地圖	67
4.2.3.4 ICF 詞典	68
4.3 原則	69
4.3.1 解譯	70
4.3.2 連結	72
4.3.2.1 相似性連結	73

4.3.2.2 對比性連結	75
4.3.2.3 相鄰性連結	77
4.3.3 選擇	79
4.3.4 想法地圖轉化	80
4.3.4.1 IE 地圖轉化	80
4.3.4.2 ICF 地圖轉化	81
4.4 連結過程	81
4.4.1 外部相互作用	82
4.4.2 內部相互作用	83
4.4.3 連結溝通語言	84
4.4.3.1 發送者與接收者	84
4.4.3.2 訊息內容	85
4.4.3.3 語言行動	86
4.4.4 訊息傳遞過程	87
4.4.5 角色學習	90
4.5 小結	92

五、DIM 之系統執行與實例

5.1 系統架構	98
5.1.1 系統元素	99
5.1.2 系統技術	100
5.2 系統環境	101
5.2.1 代理人身分定義	103
5.2.2 代理人生命週期	103
5.2.3 代理人溝通	104
5.2.4 代理人 Yellow Pages 服務	105
5.3 代理人執行	107
5.3.1 角色代理人	108
5.3.1.1 工作記憶	109
5.3.1.2 規則基礎	112
5.3.1.3 規則組織與區劃	118
5.3.2 場記代理人	120
5.3.2.1 工作記憶	120
5.3.2.2 規則基礎	121
5.3.3 場景代理人	127
5.3.3.1 工作記憶	127
5.3.3.2 規則基礎	128
5.3.4 舞台代理人	130



5.3.4.1 工作記憶	131
5.3.4.2 規則基礎	132
5.4 系統實例	134
5.4.1 代理人的知識獲得與建立	134
5.4.2 角色如何扮演、想法如何連結？	137
六、DIM 系統之連結有效性評估	
6.1 評估方式與內容	143
6.1.1 評估方式	143
6.1.1.1 連結指標	145
6.1.1.2 連結形式指標	146
6.1.1.3 連結路徑長度	146
6.1.2 評估內容	147
6.2 設計實驗：設計工作室的集體腦力激盪會議	147
6.2.1 設計工作	147
6.2.2 實驗受測者	148
6.2.3 實驗過程	149
6.2.3.1 工作環境	149
6.2.3.2 會議流程	150
6.2.3.3 腳本內容	150
6.3 評估與分析	152
6.3.1 連結有效性評估	153
6.3.2 想法組合與連結分析	157
6.4 發現與討論	158
七、結論與建議	
7.1 研究貢獻	164
7.2 未來研究	165
參考文獻	
附錄	
附錄 A DIM 代理人主要程式碼	181
附錄 B 角色代理人工作記憶案例	203
簡歷	



圖目錄

- 圖 1 從面對面的想法聯想到分散式的想法聯想 2
- 圖 2 搜尋過程與資訊狀態 (來源自 Woodbury, 1991) 11
- 圖 3 以節點與連結呈現設計知識 (來源自 Coyne et al., 1990) 13
- 圖 4 一個概念地圖 (來源自 Novak, 1998) 15
- 圖 5 貴族公寓 Hotel d'Orliance (左側) 與 Villa Meyer (右側) 16
- 圖 6 一個設計想法與其表現媒材 18
- 圖 7 相似原則 20
- 圖 8 對比原則 20
- 圖 9 相鄰原則 21
- 圖 10 某行政大樓的想法草圖與使用的早期設計案例 (KHL Architect Association 提供) 22
- 圖 11 某行政大樓的設計方案結果 (KHL Architect Association 提供) 22
- 圖 12 單向傳遞與雙向傳遞 24
- 圖 13 一對多、一對一與多對一傳遞 24
- 圖 14 集體腦力激盪的相互作用 (來源自 Chang 和 Lai, 2004) 31
- 圖 15 三種類型的互動關係 (來源自 Chang, 2004) 32
- 圖 16 Archie-II 使用者介面 (來源自 Domeshek 和 Kolodner, 1997) 39
- 圖 17 James Stirling 在 Stuttgart 設計的 Staatsgalerie 美術館 (1977-1984)(左側) 與柯比意在 Cambridge 設計的 Visual Arts Center (1961-1963)(右側)(來源自 Heylighen, 2000) 41
- 圖 18 使用 ICF 概要組織 Stuttgart 美術館案例 (來源自 Oxman, 1994) 45
- 圖 19 PRECEDENTS 之動態連結案例的機制 (來源自 Oxman, 1994) 46
- 圖 20 DIM 初期模型的案例基礎推理架構 (來源自 Lai, 2005b) 49
- 圖 21 ADLIB 系統架構圖 (來源自 Anumba et al., 2002) 50
- 圖 22 DARIS 系統架構圖 (來源自 Chang and Lai, 2004) 53
- 圖 23 DIM 模型架構 57
- 圖 24 想法元件、屬性與屬性連結的知識網路結構圖 62
- 圖 25 元件相似連結、元件對比連結與元件相鄰連結 (由上而下) 63
- 圖 26 一個 IE 地圖 65
- 圖 27 一個結合多媒體資訊的 IE 地圖 66
- 圖 28 想法元件中屬性值之置換 67
- 圖 29 ICF 地圖：議題地圖、概念地圖與形式地圖 68
- 圖 30 連結原則的三個機制：解譯、連結與選擇 70
- 圖 31 想法元件的解譯過程 71
- 圖 32 想法元件跳躍 (相似性連結) 73

圖 33	相似性連結	74
圖 34	想法元件跳躍 (對比性連結)	75
圖 35	對比性連結	76
圖 36	想法元件跳躍 (相鄰性連結)	77
圖 37	相鄰性連結	78
圖 38	連結過程與內外相互作用	82
圖 39	外部相互作用	83
圖 40	內部相互作用	84
圖 41	DIM 角色的競爭學習	91
圖 42	DIM 的系統架構	100
圖 43	DIM 的系統環境	102
圖 44	DIM 代理人的生命週期	104
圖 45	DIM 系統代理人之溝通機制	105
圖 46	DIM 的 Yellow Pages 服務	106
圖 47	角色代理人的知識基礎元素	108
圖 48	場記代理人的知識基礎元素	120
圖 49	場景代理人的知識基礎元素	127
圖 50	舞台代理人的知識基礎元素	131
圖 51	屬性值之關鍵字的關聯性建立	135
圖 52	議題、概念與形式地圖建立	137
圖 53	載入代理人視窗	138
圖 54	新增代理人平台視窗	139
圖 55	代理人管理視窗	139
圖 56	編輯 StA 腳本視窗	140
圖 57	一個 IE 地圖 (外部或內部) 與想法元件的多媒體資訊	141
圖 58	載入連結過程視窗	142
圖 59	內部相互作用連結過程顯示介面	142
圖 60	一個外部 IE 地圖	145
圖 61	設計實驗的設計工作：單一家庭住宅設計	148
圖 62	設計實驗的工作環境：Team 1 (左側) 和 Team 2 (右側)	150
圖 63	學生 E 的想法草圖 (上圖) 與想法組合的地圖範圍 (下圖)	152/153
圖 64	Team 1 在採光 (daylight) 場景產生的外部 IE 地圖	154
圖 65	Team 1 在動線 (circulation) 場景產生的外部 IE 地圖	154
圖 66	Team 2 在採光 (daylight) 場景產生的外部 IE 地圖	155
圖 67	Team 2 在動線 (circulation) 場景產生的外部 IE 地圖	156
圖 68	角色代理人 (RA_1) 在連結上的衝突性 (相似元件連結)	161
圖 69	角色代理人 (RA_1) 在連結上的衝突性 (相鄰元件連結)	161
圖 70	相互作用之網路拓樸關係：環狀、星狀、點對點與簇群狀	166

表 目 錄

表 1	輔助分散式想法連結之運算機制總表	56
表 2	一個設計想法：James Stirling 設計的 Stuttgart 美術館案例	59
表 3	想法元件的四個屬性與屬性值	61
表 4	不同想法地圖與 DIM 元素的關係	64
表 5	ICF 詞典中的詞典單元與其構成元素	69
表 6	外部相互作用中訊息傳遞過程與連結溝通語言的關係圖表	88
表 7	內部相互作用中訊息傳遞過程與連結溝通語言的關係圖表	90
表 8	DARIS 代理人與 DIM 元素匹配之關係圖表	98
表 9	DARIS 三個系統層級與 DIM 元素之關係圖表	99
表 10	一位參與者建立的想法元件	135
表 11	一位參與者建立的 ICF 詞典	136
表 12	集體腦力激盪會議的流程	150
表 13	設計實驗的腳本內容	151
表 14	Team 1 與 Team 2 產生外部 IE 地圖連結有效性比較圖表	157
表 15	Team 2 參與者產生替選方案之連結形式指標圖表	157
表 16	Team 2 參與者產生替選方案之連結路徑長度圖表	158



一、緒論

設計本身是一種設計者與設計想法之間的互動性行為。在早期設計階段 (the early design stage)，設計是須要經由多樣性的設計想法進行不斷的演化而形成。因此，許多設計問題的解決與想法的產生，常結合多重參與者，進行面對面 (face-to-face) 的腦力激盪式討論，而其中所產生的多樣性設計想法，可以提供設計者 (為具有建築專業知識的設計者) 發展未來可能替選方案的參考依據；另外，在此互動性行為中，設計者不僅與不同參與者相互激盪想法，同時，也會反應其自身的想法並進行想法之間的相互辯證。

在此互動性行為中，*想法聯想 (idea association)* 為是透過聯想 (association) 的行為去連結 (link) 到人類的記憶與想像力，用以產生多樣性想法之行為 (Webster Dictionary 2005)。設計者常運用想法聯想去啟動設計想法的產生，在過程中，設計者不僅連結本身的記憶去搜尋可能之想法外，同時也連結不同參與者的想法，以作為產生想法的依據。換言之，想法聯想可以被視為設計者剛開始產生互動性行為的關鍵，在想法聯想的過程中，如一個設計想法如何啟動另一個設計想法的發生，或一個設計想法和下一個設計想法的關係為何等，其關鍵皆在於連結 (linking)，經由產生想法之間連結的有效性¹，可以輔助設計者產生多樣性且關聯性的設計想法以解決設計問題。因此，為瞭解早期設計階段中之互動性行為，我們認為想法聯想的連結問題是本研究之主要議題。

進而，在“設計為技術的反映”的看法之下[1]，在建築設計的歷史發展，不同的技術常創造出不同設計的呈現與影響，而這些影響通常會使設計演化成多樣且具創造性。例如：工業化的大量生產改變了建築營建的過程，進而形塑了現代建築的國際式樣等。因此，當數位技術進入了設計領域，其所伴隨的數位技術化的機制，將或多或少改變本文所述之設計互動性行為，包括設計的思考、合作與溝通等方式。

此外，由於數位技術的大量使用 (例如網際網路) 與國際化的趨勢，建築設計走向分散式設計環境已為不可避免的趨勢。因此，在早期設計階段中，想法聯想的行為，也必將從早期單純面對面的腦力激盪式討論，而轉變為在不受時間與地理環境的限制下，在分散式的設計環境中進行想法聯想 (圖 1)。上述的這些數位化

¹Goldschmidt (1995) 認為設計為一種生產過程，稱之為設計生產 (design productivity)。而效率 (efficiency) 關係於設計思考，並對於設計的創造性與專家知識產生影響，因此，有效性 (effectiveness) 為評估此設計生產過程的主要依據，其評估方式建立在想法與連結之相互共存關係 (correlates) 的數量比。

技術的機制與趨勢的事實，也必將影響著想法聯想的連結問題。

為了進一步了解想法聯想的連結問題，與數位技術化機制如何影響此連結問題，進而影響到設計之互動性行為，在開始進入本研究之前，想法聯想與連結必須先有一個合理的定義。因此，作者先試圖從想法聯想的歷史發展與應用，進行有關其連結上之意義與機制的了解，同時，有關數位技術在連結上的應用與影響也必須加以說明。

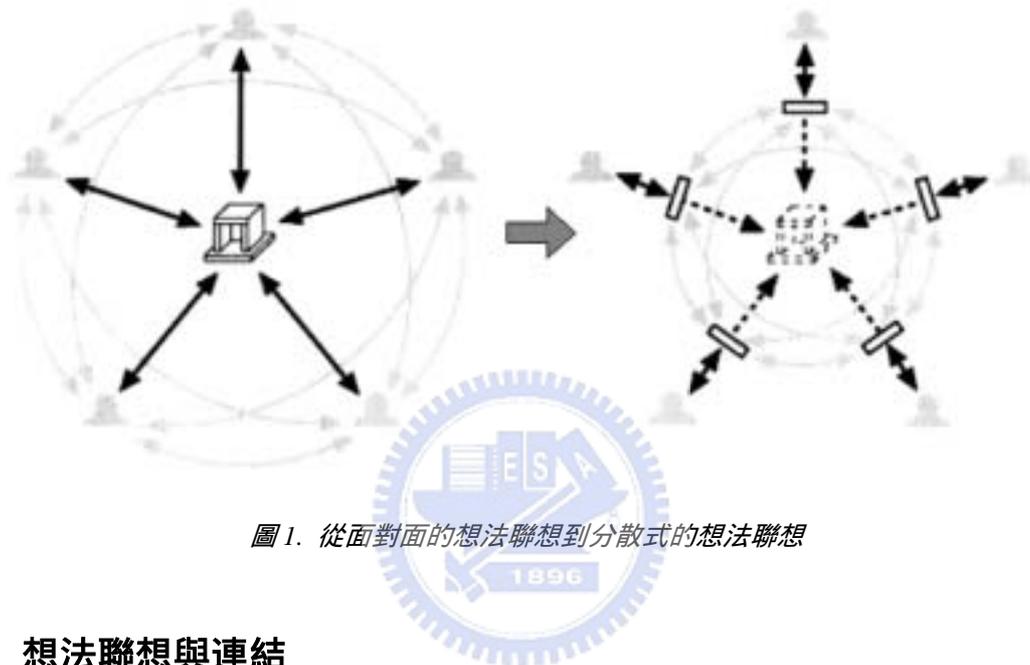


圖 1. 從面對面的想法聯想到分散式的想法聯想

1.1 想法聯想與連結

整個想法聯想的研究歷史最早起源於古希臘哲學[2]。在古希臘時代，Plato 與 Aristotle 首先具體建立想法聯想之理論架構。他們認為想法聯想為人類心理學之基本原則，同時也定下三條想法聯想的主要連結原則，包括「相近」(Contiguity)、「相似」(Similarity) 和「對比」(Contrast) 原則去連結人類思考與外在生存環境。到了 18 世紀，從 Bacon 到 Kant，這些哲學家更強調想法聯想中的連結過程，是藉由探討一個狀態導致另一個狀態的過程，將上述抽象的連結概念予以具體化；例如：Leibniz[3]主張想法聯想尋求因果關係的了解，此因果關係乃透過單子 (monads) 系統之間的連結，或 Bacon[4]藉由歸納 (induction) 的概念談狀態的改變。另外，在連結狀態的過程中，Locke[5]認為想法聯想是一種直覺 (intuition) 與反射性的行為，透過不同連結的過程，將想法連結到不同思考者的知識[6, 7]。這些論述提供了想法聯想中“連結”的理論基礎與連結原則，同時也強調想法聯想連結過程的重要性，但是，上述的這些描述仍屬於抽象及形而上之觀念。

在經過這個時代之後，想法聯想的問題才逐漸從哲學領域轉變為心理學的主要議題。在心理學研究範疇中，雖有不同的學者對於想法聯想提出不同的相關研究，

然而這些研究都是探討人類意識如何“連結”到心理狀態為主要議題，主要強調一切心理狀態都是由單一、簡單和不可再簡化的基本知識元素（如印象、記憶、感覺能力），藉由與上述想法聯想原則的結合，將人類的心理狀態去反應外在實際的問題。例如：Lewin[8]藉由拓樸關係解釋人類體驗外在環境過程中，對於不同感覺狀態之間連結關係的探討，Kohler[9]經由嘗試與錯誤（trial-and-error）的決策過程去進行狀態之間的比較與選擇，或 Freud[10]認為想法聯想是一系列問題解決（problem solving）的過程，經由人類潛意識的作用去找尋可能的解決方法等。因此，想法聯想不再只是抽象的連結概念，而是藉由人類思考的推理過程，將每一個狀態的連結關係，經由不同推理的機制如搜尋、決策、回想等，將不同狀態的想法一步一步的推進演化，並具體反應於不同的外在環境。這些在不同領域的研究都說明想法聯想的“連結”概念。

在近代想法聯想之研究中，Osborn[11]則首次將想法聯想運用於創造力的問題解決，並將想法聯想的行為結合集體腦力激盪（brainstorming）會議。此時，想法聯想是一種產生解決問題想法的行為，此行為是經由“連結”個人的長期記憶，與“連結”其他參與者的知識而達成。在設計領域中，以連結的觀念解決設計問題，被許多研究學者[12-15]廣泛應用在早期設計階段的創造性問題解決會議（或稱為腦力激盪會議），經由設計空間的廣泛探索，而得到整個設計方向的概觀外，同時，也可以提供他們組合設計想法的依據，以作為發展解決相關設計問題的可行方案[13, 16]。

為了“連結”其他參與者的設計想法，Goldschmidt[14, 17]和 van der Lugt [13]認為這些產生想法之間的關係建立是重要的關鍵，起源於古希臘之想法聯想的三個原則，經常被設計者用來產生多樣性的設計想法，與建立設計想法之間的關聯性。因此，想法聯想的連結伴隨而來的是，個人知識的互動與多重知識（multi-knowledge）的互動，這種因連結而產生分散知識的動態性互動，Chang 和 Lai[18]稱之為分散式互動（distributed interaction）。

1.2 數位技術與連結

當數位科技進入了設計領域，數位技術改變了之前論述想法聯想的行為與其連結的機制，例如，強大的電腦硬體發明，數位技術能有效的提供大量的資訊，在想法聯想的連結過程中提供更多具啟發與想像力的設計想法；由於人工智慧的發明，電腦可以被視為知識實體（knowledge entities）²，能自動化的進行推理與搜尋相關的資訊，用以輔助設計者在想法聯想的過程中連結與產生多樣性的設計想

²人工智慧的領域（包括分散式人工智慧），許多研究學者（Simon, 1965；Minsky, 1985；Russell 和 Norvig, 1995；Wooldridge, 2000）認為電腦如同人類為具有知識或行為能力的個體單元。Chang 和 Lai（2004）稱此個體單元為知識實體。

法；加上網路環境的發明，設計者能在不受限制於地理與時間的想法聯想過程中，連結分散性的設計知識。

有許多學者進行上述數位技術特質與其相關機制上的研究，這些研究提供探討想法聯想連結機制的運算可能方向。例如：藉由探討人類的思考模式，提出對大量資訊的知識呈現與其組織方式的研究，以讓人類能夠有效且迅速地連結人類的記憶，進而獲取所需的知識以解決目前的問題，包括如 Simon[19]的搜尋 (search) 策略、Archea[20]的猜謎 (puzzle-making) 策略或 Schank[21, 22]的回想 (reminding) 機制等。在設計運算領域中，一些研究藉由對數位技術中有關運算與邏輯機制的探討，建立自動化的連結機制與智慧性的輔助設計系統，並實際應用在不同的建築設計領域上，如 Coyne 等人[23]的知識系統 (Knowledge-based system) Jackson[24]的專家系統 (Expert system) Kolodner[25]的案例基礎推理 (Case-based reasoning, 簡稱 CBR) 或 Woodbury[26]的設計空間探索者 (Design space explorer) 等。

此外，Lai[27]在其初期研究所建構的動態想法地圖 (Dynamic Idea Maps) 模型，是強調單一設計者使用想法聯想行為的連結機制，以連結本身記憶與其它參與者的設計想法，並藉由 CBR 的運算機制將這些連結的設計想法，建構如地圖般的關聯性知識結構 (詳見 3.2.4 章節)。上述這些數位技術的機制與輔助系統都強調如何藉由適當的知識呈現方式反應當前的問題，並透過有效的搜尋機制，找尋資料庫的相關資訊，予以提出可能解決問題的方法，然而，在上述輔助系統的搜尋機制由於單一知識實體集中式的限制，因而缺乏搜尋資料庫相關資訊的動態性與多元性。

為了連結不同知識實體之間的分散式知識，在分散性數位技術的領域中，除了網際網路技術的應用外，許多研究學者也結合上述自動化的輔助系統，進行分散式知識的連結，例如：Oxman[28]建立的思考地圖 (Think-Maps)，為一個結合案例基礎推理與網際網路技術的輔助設計系統。此外，一些研究應用軟體代理人 (以下簡稱代理人) 的理論與系統，探討設計者 (包括人類與電腦) 之間的互動行為[29, 30]，並實際應用在不同設計領域的分散式設計系統[18, 31, 32]；為了支持多重知識實體之間的分散式互動，Chang 和 Lai[33]結合角色扮演的理論與多重代理人技術。這些研究皆為強調使用分散性的數位技術，建立多重知識實體在分散式環境中的連結，企圖經由分散性的搜尋機制以進行問題解決。

綜合以上的論述，在整個想法聯想發展過程中，想法聯想建立了一種強調連結問題的基本概念。其連結機制主要藉由適當的知識呈現方式反應當前的問題，並透過有效的推理機制，進行搜尋人類內在的記憶，在分散知識的互動性過程中，人類經由溝通行為，建立了人類內在記憶與其外在環境動態的連結關係。另外，在數位技術特質與其相關機制的研究中，不同數位化技術的機制，包括如知識呈現、

搜尋機制、分散性知識的連結等，都似乎反應了此連結問題在設計運算領域上的意義，上述描述的這些機制將提供本研究探討想法聯想有關連結的相關問題。

1.3 研究方法

當我們視想法聯想的連結問題為了解設計分散式互動的關鍵，同時，設計需反映時代的技術性時，因此，如何結合想法聯想的連結問題與數位化技術，以建構分散式設計環境中之有效性連結，將為了解數位時代之設計互動性的重要問題。為了進行此重要問題的探討，作者必須建立一個適當的研究方法予以進行本研究，因此，作者先就數位技術在設計領域之相關研究方法進行說明，並透過想法聯想之連結問題的特質論述，找尋這些特質與這些研究方法之間的關係，以建立本研究之研究方法。

1.3.1 研究方法回顧

在早期設計階段中，數位技術在設計領域的研究，主要以電腦的角度來看待設計上的問題，並進行電腦輔助設計的相關研究。基本上，這些相關研究認為設計為一個強調搜尋的問題解決過程，並經由設計認知與設計運算的兩個研究領域之研究方法，進行對相關設計之研究問題的探討[34]。

在設計認知的研究中，主要將人類在設計「黑箱作業」的心智活動[35]，透過人的認知觀點，探討設計者如何思考以及如何進行設計過程，以尋找設計者的行為結構(structure)或心智過程(mental process)，進而指出電腦應用目前的不足[36, 37]。口語分析(protocol analysis)為主要的研究方法，在探討設計過程與設計師認知活動的研究[38-43]；其技術為經由「放聲思考」與「影音回溯」，進行口語分析實驗的資料收集，並透過轉譯、編碼與資料分析等過程，與認知模型的建立以進行此相關研究。

在設計運算的研究中，由於電腦本身具有強大的運算能力、記憶容量、互動性介面與視覺回饋功能等特性，這些特性促使此類型的研究，在於發展適用於從事設計構想的電腦輔助系統[44, 45]。其研究方法強調經由電腦實際計算分析過後的運算化模型，並使用電腦模擬予以具體化(reification)。因此，其技術為架構在設計運算的相關機制(例如呈現、推理、搜尋、邏輯等機制)，找尋研究問題可以被運算化的因子，經由一個運算化模型的建立，與此模型系統的執行與模擬，進行此研究問題的了解與檢驗[23, 25, 26, 28]。

1.3.2 研究方法建立

研究方法的建立必須反應研究問題的特質，根據 Osborn[11]對於想法聯想的看法，

想法聯想的連結問題具有下列特質：

- 立即性：想法聯想為「量產生質」(quantity breeds quality) 的問題解決行為。因此，在有限的時間內，參與者（包括人類或電腦）必須經由設計想法的立即呈現，進行連結並產生設計想法的行動。
- 反射性：強調「量產生質」的問題解決方式，促使參與者必須透過反射性的動作，去反應外在不同的設計情境，以提供參與者採取適當的聯想行為，連結其內在的記憶以產生大量的設計想法。
- 關聯性：除了強調產生設計想法的多樣性外，想法聯想強調「建立在其他想法的基礎上」(build on other ideas) 的關聯性建立，經由這些關聯性的設計想法，提供參與者尋求想法組合 (seeking combination) 的依據。
- 衝突性：多重參與者在連結想法的過程中，由於參與者個人的主觀意見，導致不同參與者在想法聯想過程中，對於不同設計想法之看法上的差異，而產生參與者之間的衝突。進而，參與者經常會經由設計知識的改變，予以反應此衝突性。

此外，想法聯想行為是一種強調藉由生產大量的想法，進而提高解決問題方法的品質，在設計領域中，這種強調生產數量的設計思考活動，Goldschmidt[14, 46]稱之為設計生產 (design productive)。在設計生產的過程中，效率 (efficiency) 有關於此種設計思考活動，並反映在設計的創造性與專家知識，而有效性 (effectiveness) 為評估設計生產過程的主要方式。

根據上述研究方法之回顧，雖然，認知研究的口語分析提供我們了解設計者在設計過程的思考方式，但對於連結問題之立即性與反射性的特質，此方法在連結過程的資料收集，由於人類設計者之短期記憶不足、或口語速度不及思考速度等問題，對於這些特質的了解有其限制性。在設計運算研究中，由於，電腦強大的運算能力與記憶容量，提供設計運算中如呈現、推理、反射等自動化機制與視覺回饋的介面技術，進行了解此連結問題之反射性與立即性特質。

有關衝突性與關聯性的特質，是強調參與者（多重或個人）之多重知識間的連結關係，尤其在分散式的設計環境中。在設計認知的研究中，有相關的研究使用口語分析方法進行多重參與者之連結問題研究[13, 14]，但由於分散性的知識與其連結關係的資訊不容易收集，因此，只限於人類設計者面對面的連結關係研究，對於分散式設計環境中之多重知識之連結關係仍有其限制性。在設計運算研究中，由於，電腦的分散性的數位技術（如網際網路）與運算機制（如多重代理人），可

以經由設計運算中如搜尋、溝通等自動化機制與超連結技術，進行了解此連結問題之衝突性與關聯性特質。

綜合以上的論述，設計運算的研究方法有利於本研究對於此連結問題的探討，同時，相關此問題之認知領域上的知識，可以輔助此連結問題之運算機制的建立。因此，本研究之研究方法，主要以設計運算的角度看待此連結問題，並結合相關的設計認知領域知識，找尋此連結問題可以被運算化的因子，經由一個運算化模型的建立與其系統的執行與模擬，進行本研究有關連結問題的了解與評估。

由於，立即性強調此連結問題在視覺回饋的介面需求，反射性與關聯性強調連結問題在想法與其關係的連結建立，衝突性則強調多重知識實體在知識改變的機制，因此，根據之前論述的研究動機（強調利用數位化技術建構分散式設計環境中之有效性連結），本研究的評估的重點是連結問題在反射性與關聯性的特質，如何經由數位機制建立連結之有效性。

評估方式除了透過系統實作以驗證此連結問題運算化的可行性外，本研究也將執行一個多重參與者進行分散式想法連結的設計實驗，並藉由實驗中受測者所產生的設計結果，進行這些設計結果在連結有效性的比較分析，其比較分析方式則使用 Goldschmidt 的連結圖像理論（linkography）[46]（詳見 6.1 章節），量化分析想法與想法之間的連結，以檢驗本研究建立的分散式想法連結機制。至於，有關連結問題的立即性特質，本研究會在系統模擬與設計實驗中特別注意使用者的使用狀況，以提供未來介面設計的需求依據，同時，此設計實驗中的發現也將提供本研究進一步精練衝突性之相關機制。

1.4 論文概述

架構在之前的研究方法，本研究經由研究問題的界定與假設的提出，進行研究目標的說明，並提供建立研究步驟的依據。有關未來研究的延伸與可能的研究方向，將被討論在本論文的相關章節中。

1.4.1 研究問題

如前所述，分散式環境中的多重知識連結為一個重要的研究領域範圍，然而，如何在分散式的設計環境，結合想法聯想的連結問題與設計運算的機制，仍然是一個不清楚的研究領域。基於上述的原因，本研究產生兩個主要的研究問題：想法聯想中有關連結問題與其分散性的抽象行為，該如何被建構並加以運算化？而設計運算的理論與技術，又該如何建立一個數位工具，用以輔助本研究了解分散性的想法連結問題？

1.4.2 研究假設

當數位技術提供分散式知識在資訊、記憶與不同知識實體之間新的連結機制時，這些數位技術與運算理論的結合，將改變分散式知識之間的連結，並提供了解想法聯想之分散化連結機制的可能性。基於這些事實，本研究的假設為 1) 想法聯想的分散式連結問題，可以經由運算化的機制進行了解並建構；2) 而其發展的數位工具，在分散式的設計環境中，可以輔助多重參與者產生有效性的連結。

1.4.3 研究目標

上述的假設必須透過一個輔助分散式想法連結系統的建立而予以驗證。此輔助系統需能具體反應分散式想法連結機制，如何被實踐在設計運算相關理論和技術，而經由此輔助系統的建立，我們可以獲得多重參與者進行分散式想法連結的經驗，同時，了解此連結機制所伴隨之設計運算上的複雜性問題。

1.5 論文內容架構

架構在上述的研究方法與論文概述，本研究的論文內容架構具體反應了研究之步驟與其內容，其詳細說明如下：

- 在第二章中，作者首先架構在「設計搜尋」的概念下，找尋構成搜尋的運算機制（包括呈現、記憶、規則與決策），然後，經由這四個搜尋機制的面向，進行想法聯想之連結問題在設計運算與認知領域的背景回顧，以提出一個分散式想法連結的機制。
- 在第三章中，根據背景研究所獲得的分散式想法連結的機制（包括呈現、回憶與溝通），進行輔助呈現、回憶與溝通之輔助設計系統（包括案例基礎推理系統與多重代理人系統）的文獻回顧，並進行這些系統的比較與分析，以提供本研究建立一個輔助分散式想法連結模型的基礎。
- 在第四章中，本論文延續初期研究所建立的動態想法地圖（DIM 初期模型），更進一步的結合第三章所獲得在輔助分散式想法連結的運算機制，以精練並強化 DIM 初期模型在分散性的特質，進而建構一個輔助分散式想法連結的運算模型，而此模型的構成元素與其運算機制在本章有詳細說明。
- 在第五章中，DIM 經由架構在一個多重代理人架構（DARIS）的平台基礎上，予以具體化 DIM 模型的運算機制，並發展成一個輔助分散式想法連結系統，而此系統的使用方式藉由一個實際案例予以說明。有關 DIM 的系統實作與使

用方式在本章節有詳細說明。

- 在第六章中，為了驗證本研究之假設，作者執行一個設計實驗，並架構在 Goldschmidt [46]所建立連結圖像理論與其評估方式，進行 DIM 系統在輔助連結有效性之評估，同時，透過此設計實驗，了解多重參與者在分散式的設計環境中，如何藉由本研究建構的分散式想法連結機制，反應上述連結問題的特質，又如何輔助他們在早期設計階段的想法連結。
- 在第七章中，作者首先對本論文的研究提出結論與建議，並藉由此研究的發現，更進一步的說明本論文的研究貢獻，與作者未來後續的研究方向。



二、研究背景：從設計搜尋到分散式想法連結

想法聯想建立了一種強調連結問題的基本概念。連結問題的機制（以下稱連結機制）主要藉由立即性的呈現方式反應解決問題的想法，並透過反射性的搜尋策略連結參與者內在的記憶，同時，在分散性知識的互動過程中，經由多重參與者在內在記憶與外在設計情境的衝突，建立了設計想法之間關聯性的連結。由於，本研究目標是藉由一個輔助系統的建構，以了解分散式想法連結的相關問題，因此，為了能更進一步了解連結機制之相關背景在設計運算上的意義，作者架構在相關的設計運算觀點，予以進行此連結機制的背景回顧。

在設計運算的領域中，於「設計搜尋」(Design as search) 的概念下，發展出不同的搜尋模型，進而衍生為大部分的電腦輔助設計系統[23]。基本上，「設計搜尋」強調設計過程中的搜尋機制，因此，搜尋機制將提供本研究了解連結機制在設計運算領域中的意義。架構在搜尋機制的觀點下，首先須了解「設計搜尋」的相關背景，經由了解搜尋機制在設計運算中的意義，再以搜尋機制所包含的不同面向，進行連結機制的背景回顧，以提出一個分散式想法連結的機制。

2.1 設計搜尋

從 Simon 和 Newell[47]開始，在相關分析設計過程的研究中，設計被認為是一種解決問題 (problem solving) 的行為。為了提供設計者可以透過一個特定模式的設計過程，以提升設計問題解決的結果，有許多學者提出不同的設計過程模型[23]，例如：Alexander[48]的「分析與綜合」(analysis 和 synthesis) 模型、Pena 等人[49]的「計劃與設計」(programming 和 design) 模型、Asimow[50]的「分析、綜合與評估」(analysis、synthesis 和 evaluation) 模型，亦有學者架構在已提出的模型基礎上發展出另一組模型，例如：架構在 Asimow 提出的模型基礎上，Jones[35]提出「分歧、轉化與集中」(divergence、transformation 和 convergence) 模型，此模型的分歧係指將設計問題予以分解成若干片段，並透過搜尋大量相關的解決方法以界定設計空間的範圍，轉化則為將這些分解的片段以一種新的方式組織在一起，而集中為發現此新組織方式的可能結果並予以實踐。

為了解決設計問題，設計被視為設計者在設計問題的空間中，搜尋解決此設計問題方法的過程[19]。此搜尋過程為經由不同的狀態（設計的解決方法），找尋解決目前設計問題的方法，這些狀態主要包括起始狀態 (initial state)、中間狀態 (intermediate state) 及目標狀態 (goal state) [19]。在進行搜尋的設計空間中，設計者應用自己本身知識庫中程序知識 (procedural knowledge) 或規則的區塊

(chunk)，將設計從一個狀態移動到另一個狀態。因此，許多自動化的問題解決系統都來自於 Newell 和 Simon[47]的理論，這些自動化的系統稱為 General Problem Solver (簡稱 GPS)。

然而，設計者在問題解決的搜尋行為上，相似於上述 GPS 系統應用在人工智慧的搜尋策略，例如嘗試與錯誤搜尋 (trial-and-error)、深度優先搜尋 (depth-first search)、廣度優先搜尋 (breadth-first search) 和經驗法則搜尋 (heuristic search)，這些搜尋策略已經被運算化的應用在電腦輔助設計系統[38, 51]。此外，Anderson[52, 53]根據搜尋在推理的方向性，將設計搜尋的推理機制歸納成向前鏈接 (forward chaining) 與向後鏈接 (backward chaining)，而 Anderson 等人[54]更進一步指出，有經驗的設計師通常使用向前鏈接的推理機制進行搜尋，而沒有經驗的設計師通常使用向後鏈接的推理機制。

在設計運算領域中，Woodbury[26]認為上述之設計搜尋過程，在可能的設計空間中找尋解決問題的路徑，此設計空間是一群資訊集合 (information set)，包括起始狀態、中間狀態 (intermediate state) 和目標狀態，與在狀態之間的運算元 (operators)。每一個狀態代表可能完成或未完成的設計，因此，設計可視為在起始狀態與目標狀態之間，去發現適當運算元的序列或路徑[26] (圖 2)。在搜索過程中，Norman[55]認為設計者藉由特有的限制 (constraints) 因素，執行設計相關目標與決策之間的配對 (mapping) 關係以決定方式，以得到最佳化的解答。

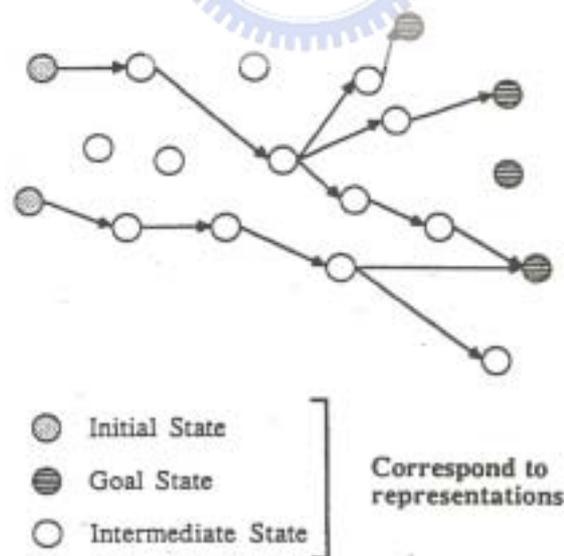


圖 2. 搜尋過程與資訊狀態 (來源自 Woodbury, 1991)

除此之外，在以設計支援系統為主的設計研究領域中，一個提供設計支援系統的架構，被稱之為設計空間探索者 (Design Space Explorer)，其引用一種最能被廣泛接受的設計行為：「搜尋」或稱為「探測」。設計空間探測者將設計比喻是一種在

有選擇性且可能無盡的設計空間裡的智慧型行為，在這個無止盡的設計空間裡，設計者“尋找”或“探測”所有可能的設計、發現新設計、記錄或沿用已完成的設計[56-59]。

在此無盡的設計空間中，運用適當的設計規則反應主要設計需求，然後以此類推到後來不同的設計狀況，進而產生設計空間中的元素（nodes, 設計狀態）和邊緣結構（edges, 狀態關係）。所以，這樣的思考系統刻意的被打開，設計者能在任何時間修改或改變設計說明、狀況設定、或隨後衍生的方法或限制，進而，此「探測」機制包括呈現（*representation*）規則（*rule*）記憶（*memory*）與決策（*policy*）[26]，且架構在此「探測」機制，有許多相關研究之發展，例如：設計空間探測者的運算模式及再現模型之研究[26, 60]，或設計空間探測者的系統建構之研究[61, 62]等。

總而言之，設計被視為解決設計問題的行為。為了解決設計問題，設計者必須在設計空間中，搜尋設計問題的解決方法。基本上，設計空間的組成分為 1)設計結構性的敘述，例如：圖形/節點（*graph/nodes*）；2)原始的設計問題（*initial design problem*）；3)設計轉化（*design transition*）的狀態，和 4)詳細解說的解決問題方法（*solution*）而 Woodbury[26]提出的「探測」理論，具體的建立了在設計空間的搜尋機制，主要包括呈現、記憶、規則與決策，簡言之，呈現敘述設計所內涵的資料結構，規則反應設計狀態之間的轉變條件，記憶則負責儲存搜尋過程及其設計空間的資料，決策擔任設計過程的流程控制。根據這四個搜尋機制的面向（呈現、記憶、規則與決策），本研究將進行連結機制在設計運算與認知領域的背景研究。

2.2 連結與呈現

所謂呈現係指表現一個特定領域概念的符號（*symbol*），提供設計知識基本的表達與組織方式[23]。當我們視想法聯想的連結機制為強調想法本身與想法之間的關聯性建立時，那想法的呈現是什麼？而關聯性的呈現又是什麼？

在古希臘時代，“想法”這個字的意義為“一個事物的出現”（*appearance of a thing*）。人類擁有一種獨特的能力去使用符號進而表達想法，並透過建構符號與符號間的連結關係解釋外在的事物[63]。在建築設計的領域，設計者常將一個設計予以分解成數個建築元素，然後藉由元素的符號與元素之間的關係建構建築設計的知識[23, 64, 65]，且經由強調元素本身與元素之間的關係，予以呈現有關聯性設計知識的資訊。

在設計領域中，van der Lugt [13]認為設計想法具有下列二個特性：1) 設計想法必須相關於正在進行的設計工作，並能提供某一特定設計問題的解決方法，所以，

設計想法可以被視為設計問題與其解決方法的組成；2) 設計想法也必須提供參與者在解決設計問題討論中的溝通。Oxman[66]認為使用早期設計案例(precedents)，可以有效產生解決設計問題之想法，同時，在產生想法的過程中，有助於參與者之間的溝通。

另外，想法是一種思想(thought)，而思想也須透過文字、繪畫、圖像等媒材表現，提供人與人之間的溝通[67]。因此，除了呈現知識本身內在的符號與符號之間關係的建立外，使用適當的媒材表現設計想法有助於設計者對於想法的了解，進而提升參與者之間的溝通。基於上述的論述，本研究經由連結知識呈現、設計案例呈現與媒材表現了解連結與呈現的關係。

2.2.1 連結知識呈現

在設計過程中尤其是早期設計階段，設計者常藉由泡泡圖(bubble-diagram)的方式去呈現設計知識(包括物件與關係)，此種圖像化的類比(diagrammatic analogy)方式類似於 Quillian[68]所主張之語意網路(semantic network)。語意網路對應之設計知識呈現方式乃以符號(或稱物件)，與符號間關係為主要的構成元素；其強調圖像連接性的資訊(connectivity information)，主要以節點(nodes)和連結(arcs)為構成具有連結性知識結構的主要元素(圖 3)。

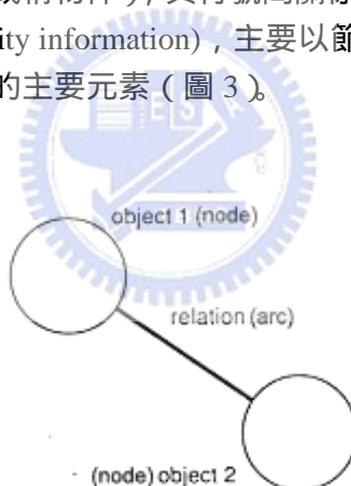


圖 3. 以節點與連結呈現設計知識 (來源自 Coyne et al., 1990)

在設計運算領域中，Coyne 等人[23]認為上述這些符號與關係必須根據不同的設計工作，賦予這些符號與關係的相關設計內容(包括如概念、實例、等級或屬性)，提供設計者瞭解及組織關聯性設計知識的方式。有一些學者提出不同的知識呈現方式詮釋具有關聯性的設計知識，例如：Coyne 等人[23]提出的事實基礎(fact-based)為呈現設計知識的物件與物件之間的關係，並藉由文字基礎為其象徵的事實(facts)強調連接性的知識；而 Minsky[69]提出的架構基礎(frame-based)為透過框架式的連結機制，將設計知識中物件的相關屬性以框架(frame)、空格(slot)與值(value)來呈現。這些強調連結關係的知識呈現，提供設計者瞭解並組織設計知識的方式，同時，在固定的設計知識結構下，符號與之間的關係可以

被增加。

另外，使用連結呈現語意為連結知識建構另一個重要的議題。Sowa[70]提出的概念圖像 (Conceptual Graphs) 除強調知識呈現的語意問題外，同時也強調符號之間的連結知識。由於連結知識可以表達複雜語意上的知識，因此，概念圖像被廣泛應用到處理自然語言 (natural language) 的運算領域，類似於 Sowa 的概念圖像，而 Novak[63, 71]與 Novak 和 Gowin[72]提出的概念地圖 (Conceptual Maps)，則經由概念 (符號) 具語意的連結 (關係) 與地圖般的結構，允許使用者能動態性的組織知識結構。

有別於之前論述之固定性的設計知識呈現方式，概念地圖提供動態性的知識建構機制，而且，廣泛的應用在有關連結知識建構的意義學習³ (meaningful learning) [28, 73]，包括如個人之概念知識的建構，和以團體基礎 (team-based) 之集體概念知識的建構[71]，經由這些概念知識的建構，幫助他們了解概念的意義並進行學習。為了更進一步了解概念地圖中，相關於連結與知識呈現的關係，本研究將繼續對概念地圖進行探討與了解。

2.2.2 概念地圖

概念地圖是 Novak 和 Gowin[72]在進行有關人類學習和知識建構研究的結果，他們提議知識的基本元件是概念 (concept) 與概念之間的關係敘述 (proposition)，同時，Novak[71]把概念定義為“在事件或物件，或者事件或物件的記錄中，被理解的一個規律 (regularity)，此規律可以使用一個標籤予以指定” (p 22)。而概念地圖是經由一個具語意單位的連結關係，連結兩個或更多概念的標籤所組成，因此，概念地圖是以二維向度的圖像方式呈現所有關聯性的概念 (圖 4)。

基本上，一張概念地圖是透過不同節點和連結所建立之圖像似 (graph-like) 的知識架構[63, 71] (圖 4)。節點是描述概念的要素，而連結則描述在兩個節點之間的關係，也就是概念之間的關聯性，同時，為了描述概念之間的關聯性，連結可以根據不同的需要決定是否貼上標籤 (label)。所以，概念地圖的圖像形式反應了關聯性知識的建構方式與欲描述的知識，可以應用在不同領域有關連結知識的建構，並有助學習者進行意義學習[73]。

³Ausubel (1962) 認為意義學習是強調描述性學習 (descriptive learning) 方法的學習理論。在此理論中，學習的基本想法為將吸收的新概念和建議放置在學習者原有的概念架構，同時，進來的概念因為與長期現有的知識相互作用而被組織且處理。因此，學習被認為是根據存在命題的網路或概念地圖延伸人類知識的過程。

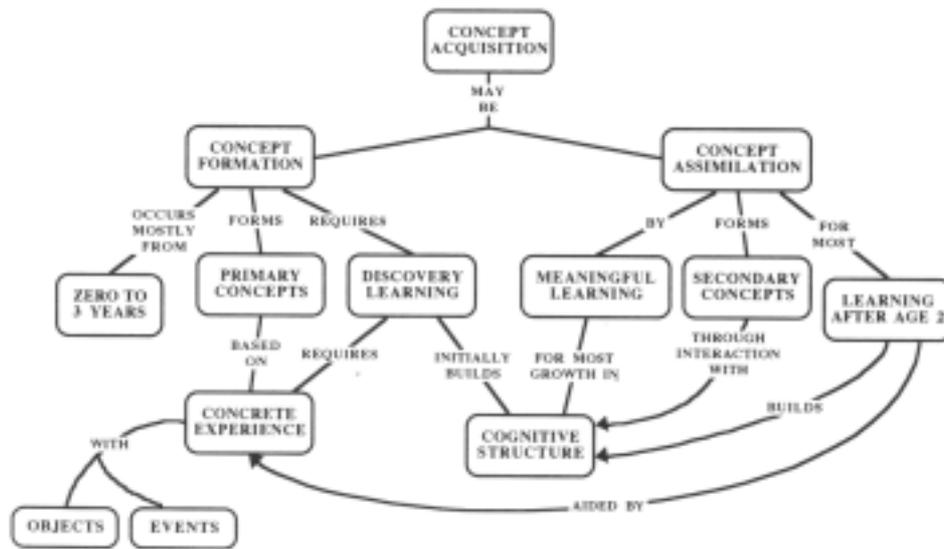


圖4. 一個概念地圖 (來源自 Novak, 1998)

概念地圖的連結知識結合了內部領域連結(in-domain linkages) 和跨領域連結(cross-domain linkages)的特質。內部領域連結強調知識在一個相同領域的網路架構邏輯中，例如：在呈現一種建築物類型，內部領域連結的知識侷限在此種類型相關的概念。然而，對於一個跨領域連結，知識則在二種以上的不同領域具有多重的呈現，例如：中庭在傳統街屋與購物中心二種不同建築類型中，各自對於中庭有不同的概念，經由跨領域連結可以將這些不同中庭概念連結在一起。在設計運算領域中，Oxman[28, 66]認為內部領域連結與跨領域連結都是設計思考中重要的特性，並結合概念地圖連結知識的建構方式，將個人或團隊基礎集體討論的設計想法予以組織與呈現，進而發展輔助連結知識建構的 CBR 系統(稱為 Think-Map)

總之，概念地圖是一種強調連結知識的圖像 (diagram) ，其用以探索人類在知識建構過程中，對於知識分享與知識集結的連結資訊，有助於設計者動態性的建構個人與其他參與者之間，產生設計想法的連結知識。一些研究學者使用概念地圖發展輔助建構連結知識的工具，這些工具經由將知識如地圖般的圖像方式呈現，提供使用者理解所有關聯性知識的主體架構，例如：CampTools[74]和 KMap[75]使用網際網路技術，連結分散式環境中的想法。除此之外，有些研究學者[71, 76, 77]透過不同的評估方式了解概念地圖與意義學習之間的關係。

此概念地圖強調關聯性知識的建構機制，可以反映在處理資料結構的圖像理論 (Graph Theory) 。基本上，圖像理論[78, 79]是一種類似概念地圖之圖像般的網路資料結構，其基本的元素包括節點 (vertices) 與邊 (edges) ，與邊的方向性 (direction) 路徑 (path) 等級 (degree) 等定理，有助於作者了解設計知識有關呈現、連結與資料結構的關係。

2.2.3 案例知識呈現

Schank[21, 22]認為利用過去的經驗可以有效的解決當下的問題，同時，他提出案例知識在設計中具有三種的基本意義與功能，包括 1) 案例是種知識的表達，即是經驗法則；2) 案例本身是種資訊結構，提供經過組織過的相關資訊，並有助於作設計決策時之評估；3) 案例是一種萃取的資源，亦即案例在尚未形成特定之事實前，為一種已被認定可望獲取有效知識的資源。

在建築設計的領域，早期設計案例為設計者主要的經驗[66, 80]，Akin[81]指出，從早期設計案例所獲得的概念想法是提供設計的抽象性概念與真實實踐之橋樑，且有助於建築設計的意義學習。基本上，早期設計案例是一種被讚賞（recognized）的過去設計作品[82]，透過嵌入在這些知名設計作品裡之獨特知識的特性，提供設計者參考的便利性[66]。因此，建築理論學者 Rowe 和 Kotter[83]在 Collage City 書中提倡建築歷史的重要性，他們認為建築與城市設計的想法主要來至於早期的設計案例，而建築師 Dennis[84]則引用 Le Corbusier 所設計的 Villa Meyer 的住宅設計為例，說明此設計案例有關自由平面的設計想法，主要來自於十八世紀在巴黎市中心的貴族公寓 Poche 的平面技法並轉化而成（圖 5）。總之，在連結想法的過程中，早期設計案例是提供設計者連結與產生設計想法的主要來源。

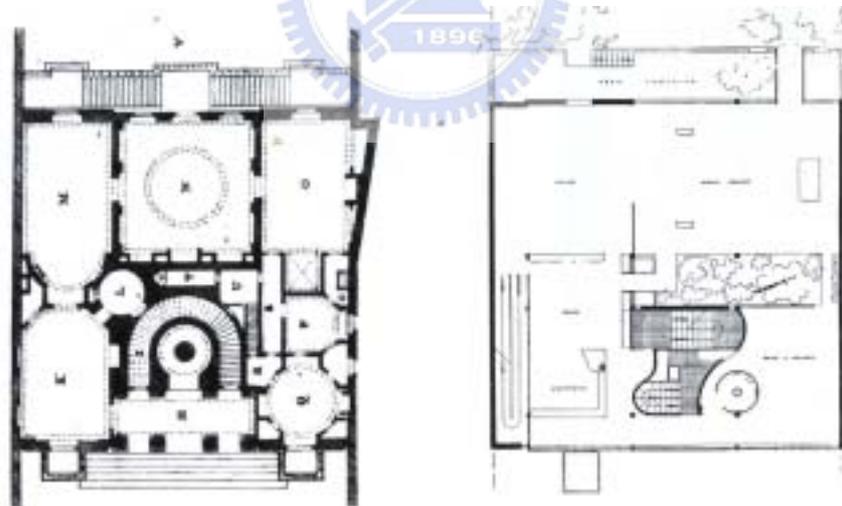


圖 5. 貴族公寓 Hotel d'Orliance (左側) 與 Villa Meyer (右側)

在設計運算領域，Gero[85]認為設計案例本身除了是一具體的個案外，其基本上是設計知識的濃縮或解決之方案，其不但縮短設計解答的途徑，而且可以提高設計的效率，因此，設計案例被認為是早期設計經驗透過壓縮（condensed）的知識，此知識為提供解決設計的方法[86, 87]。因為設計案例的重要，所以，有許多學者透過不同的方式在描述案例，包括如 Domeshek 和 Kolodner[88]的 story Schank[21]

的 script、Minsky[69]的 frame 等。此外，Kolodner[25]認為“案例通常代表實際的狀態，在此種狀態中，案例整合了許多複雜的資訊”，而 Maher 等人[80]則認為設計案例呈現必須考慮三個課題：1) 設計案例的內容、2) 設計案例記憶組織方式、與 3) 設計案例對於使用者的表現媒介。

根據 Maher 等人[80]認為設計案例的內容可以由設計繪圖、需求與方法、或機能、行為與結構 (Function-Behavior-Structure, 簡稱 FBS) 所構成，而案例知識的呈現則可以由文字呈現、物件導向呈現 (Object-Oriented representation) 圖像基礎呈現 (Graph-Base representation) 多媒體呈現或案例階層呈現而達成。這些案例知識的記憶組織方式包括平面結構 (Flat structure) 案例特徵基礎結構 (Feature-Based structure of cases) 和階層結構 (Hierarchical structure)，而平面結構是最為普遍的記憶組織方式，此記憶結構主要強調案例之間沒有階層與其他組織的關係[89]。除此之外，設計案例知識的表現也可以藉由文字、影像、或多媒體等表現方式，提供使用者與電腦進行設計溝通與互動[27, 80, 90]。

上述 Maher 等人[80]所提出的三個課題會根據設計目的(如設計生成、設計決策等)的差異而有所不同，並應用在許多不同的建築設計問題，例如：Domeshek 和 Kolodner[88]提出一個以故事型態的案例呈現方式去輔助設計學習；Oxman[66]提出的議題、概念與形式的知識呈現方式 (簡稱 ICF)，提供設計案例本身與不同設計案例之間的連結關係；另外，結合多媒體的資料視覺呈現，除了為案例內容的一部份，同時藉由此案例的視覺化呈現幫助設計者解決設計問題[80]。

而這些知識呈現已被實際的應用在案例基礎推理之輔助設計系統。根據 Maher 和 Pu[91]的分類，包括 Archie-II[92]、FABEL[93]、CASECAD[80]、SEED[94]、PRECEDENTS[66]等案例基礎推理系統，這些系統在設計案例呈現與組織的相關機制在第三章有詳細的說明與分析。

2.2.4 表現媒材

設計想法藉由不同的媒材 (如文字、草圖、影像照片) 表現，其目的是提供不同參與者可以進行溝通[27, 67]。有些學者[13, 95]認為使用文字表達設計想法較其他媒材能立即性的產生設計想法，進而助於產生大量的設計想法，另一方面，van der Lugt [13]和 Lai [27]認為影像照片等視覺呈現相較於文字，更能增加參與者之間的溝通。

根據設計行為的相關研究[23, 48, 65]，設計者習慣將建築設計予以分解成許多建築的元素與屬性，並使用這些元素的屬性，作為尋找解決設計問題相關設計想法的依據。為求設計溝通的有效性，設計者經常使用領域概念語彙 (domain conceptual vocabulary)，予以表達設計想法中建築元素與其屬性的意義，並建構它們之間的

關係[66, 96]。

設計也是一種視覺的反射行為[40]，所以，設計者經常使用草圖或影像圖片等視覺表現敘述設計想法。根據作者之前的研究[27]，視覺表現是提供設計者解決問題方法的實際案例之具體化表現。例如，設計者為解決水岸與廟城之間「動線」的設計問題時，一位設計者使用「漂浮建築」(floating building) 等領域概念語彙去表達設計想法中的設計問題與解決方法，同時使用草圖與柯比意設計的 Villa Savoye 的影像照片為實際案例，予以具體化此解決方法（圖 6）。

不同的設計者之間雖然具有相同的設計想法內容，但他們會使用不同的領域概念語彙描述這些相同的想法內容，此問題被稱之為「解譯」(interpretation) 上問題[93, 97]。在上述圖 6 的例子中，二位設計者 (A 與 B) 欲解決上述的設計問題時，當兩位設計者都以 Le Corbusier 設計的 Villa Savoye 為解決「動線」問題的解決方法時，設計者 A 使用「漂浮建築」描述其解決方法，而設計者 B 則使用「底層挑高」描述之。解譯問題提供每一位設計者根據各自的知識，對相同的設計想法有不同面向的看法與解釋[98]。



圖 6. 一個設計想法與其表現媒材

2.3 連結與規則

所謂規則乃藉由正規化語言 (formal language) 所描述之設計轉變 (transformation)。基本上，規則分為兩個部分：先期條件及轉換方式。先期條件提供對規則的限制條件(constraints)，轉換方式是設計轉換之執行方式及結果描述，因此，規則提供兩個設計狀態之間轉變的機制。另外，規則具有改變性的性質，此性質提供設計搜尋的開放性，換言之，一個規則的使用，經由不同的方法執行一個設計狀態，可能引發無數設計狀態的轉變。

想法聯想強調想法之間的連結機制，此種連結機制為一個設計想法的產生必須仰賴其它產生的設計想法，進而引發一個想法轉變到另外一個想法[11]，在設計領域中，Goldschmidt[46] 稱上述想法之間的轉變過程為設計移動 (design move) 此種設計移動的論點可以解釋個人或群體在設計想法之間的動態性連結關係[14]，同

時，也反應 Schon 和 Wiggins[40]二人強調設計有**反射性行動** (*reflective action*) 的特質，而了解反射性行動的兩個主要概念是**行動** (*act*) 與**反應** (*re-act*) [40]。

在連結想法的過程中，當設計者接收到其他參與者的設計想法時，設計者常把一個設計想法當成起始狀態，除了即時對它採取“行動”以產生新的設計想法外，並經由改變設計者的狀態，“反應”的將設計想法輸出，以連結其他的參與者的設計知識。由此可知，行動與反應是連結機制的重要關鍵，提供參與者之間的狀態改變的規則建立。

綜合上述，連結機制提供兩種層級在設計狀態之間的轉變規則：設計想法間的層級，與其設計者間的層級。在設計想法的層級，想法聯想有三個**原則** (*principles*) (包括**相似原則**、**對比原則**與**相鄰原則**) 提供設計想法之間的轉變機制；在設計者的層級，**語言行動** (*speech acts*) 支持參與者之間經由行動與反應的轉變機制，改變參與者之間在溝通過程中的互動關係。

2.3.1 想法聯想的三個原則

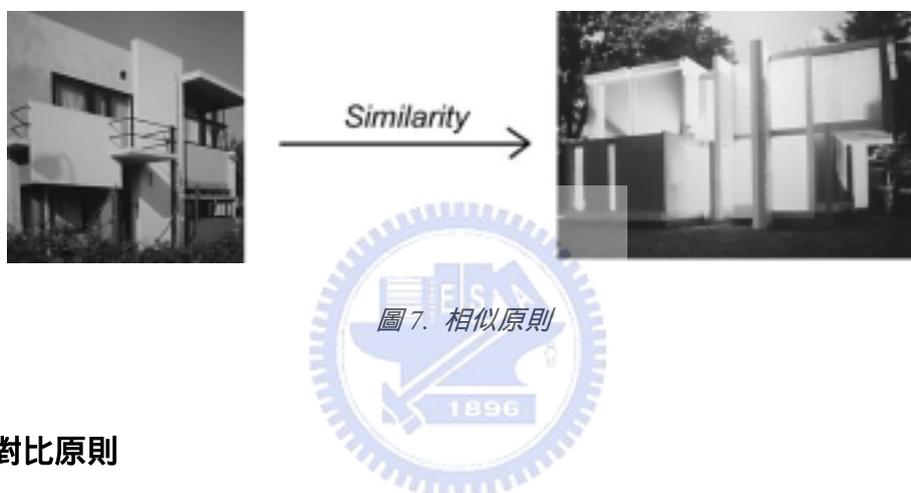
想法聯想即是想像力連結到記憶的部分，使一個想法導致另一個想法之現象[11]，Plato 與 Aristotle 認為此種連結想法的力量為人類心理學的基本原則，而且，Dewey[99]指出將這些想法相互地連結起來，即可增加人類創造性的思考。所以，在連結想法時，人類須分析並運用存在於不同想法之間的關聯性，例如：想法之間的「相似之處」、「不同之點」、「因果關係」或「前後次序」等關係，而古希臘人將這些關係歸納成三種主要的原則：相似原則、對比原則和相鄰原則。雖然，有其他學者提出許多聯想作用的原則[9, 10, 100]，但古希臘人所建立的三大原則，至今仍被視為最基本與普遍應用的想法聯想原則[11]。

在建築設計的領域，設計者經常會對於設計想法看法上的差異而作不同的解譯，並使用這三個原則為策略，以便連結與產生多樣性的設計想法，與設計想法之間的關聯性[101]。例如：相似原則為連結並產生具有相似性的設計想法與關係，而對比原則是連結並產生具有對比性的設計想法與對比關係，另外，相鄰原則為透過設計想法之間的因果關係，而連結並產生相鄰性的設計想法與相鄰關係。因此，這三個原則可以被視為設計想法轉變的規則，經由在設計空間進行設計想法之間的比較，進行連結與產生設計想法與其關係。

另外，有關想法聯想的三個原則（相似原則、對比原則和相鄰原則），因表現設計想法的媒介內容與資訊之不同而有所差異[11]。架構在之前論述（詳見 2.2.3），設計想法的內容包括設計問題與解決方法，而表現想法的媒介主要為領域概念語彙。對於想法聯想三原則在建築設計上連結想法的應用，則經由下列小節作進一步說明。

2.3.1.1 相似原則

相似性係指物件本身與其他物件具有相類似的屬性。設計者對於一個相同設計問題，通常會有不同但相似的解決方法，因此，設計者可以利用相似原則連結產生相似性的設計想法。例如：Peter Eisenmen 設計的 Frank House 和 Gerrit Rietveld 設計 Schröder House 在解決「牆面組構」的設計問題具有相似的解決方法。從上面的兩個設計案例，設計者透過相似原則可以產生兩個相似性的設計想法，此想法是使用「多層次牆」的方法解決「牆面組構」的設計問題，進而，這兩個解決方法所對應的設計想法建立相似性的關係（圖 7）。



2.3.1.2 對比原則

對比性係指物件之間的比較，且比較後的物件具有明顯差異的屬性。設計者藉由對比性的領域概念語言（例如，室內和室外、隱私和公共、實體和虛體關係等），經由連結具對比性領域概念語言的解決方法，進而產生對比性的設計想法。比較於 Le Corbusier 設計的 Villa Savoie 案例中，十八世紀巴黎傳統的貴族式住宅，在地面層採用「實體空間」的方法去解決「建築量體」上的問題。因此，設計者使用對比原則去連結 Villa Savoie 的設計案例，而產生設計想法為使用「虛體空間」的方法解決「建築量體」的設計問題，進而，這兩個解決方法所對應的設計想法建立對比性的關係（圖 8）。

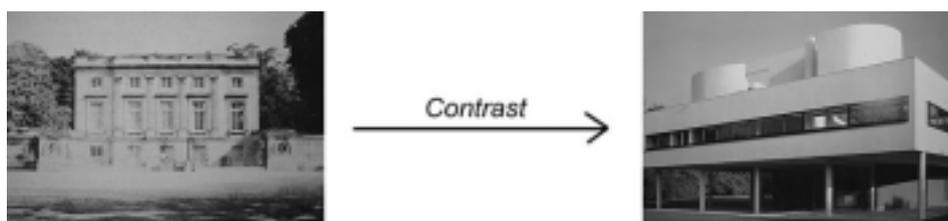


圖 8. 對比原則

2.3.1.3 相鄰原則

相鄰性係指物件之間具有因果性的關係。由於設計問題與解決方法之間存在著因果關係[19]，因此，當不同設計想法對於不同的設計問題有相同的解決方法時，設計者可以利用相鄰原則連結到這些相同的解決方法，進而產生相鄰性的設計想法（具有不同的設計問題）。例如，Adalberto 在坐落於玄崖邊基地所設計的 Villa Malaparte 設計案例，一個設計想法以使用「屋頂平台」的方法用來解決業主對於「動線」的設計問題，這個設計想法可以藉由相鄰原則而連結到另一個設計想法，此設計想法是使用「屋頂平台」的方法解決「景觀」的設計問題，進而，這兩個設計問題所對應的設計想法建立了相鄰性關係（圖 9）。

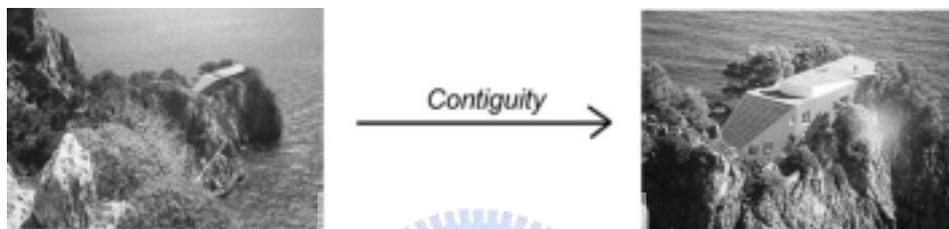


圖 9. 相鄰原則

此外，設計者通常會根據個人的喜好（preference），選擇符合目前情境所需要的設計想法[102]，在想法聯想的過程中，當不同參與者產生的設計想法經由三種想法聯想原則而建立不同關聯性的連結後，參與者會在產生的設計想法中，選擇多數設計者同意的設計想法為主要的共同想法，並以此共同想法為尋求想法組合的依據[11, 103]，Lai 和 Chang[104]稱此現象為設計的共鳴現象。例如：在下列的設計案例中，為某建築師事務所的三位設計專家為解決某大學行政大樓有關採光的設計問題，在早期設計階段的集體腦力激盪會議，經由想法聯想三個原則所產生的設計想法（包括想法草圖與設計案例）（圖 10）。

在這些設計想法中，三位設計專家選擇羅馬萬神殿的採光方式與空間關係為共同想法，在設計發展過程中，與此共同想法相關聯的設計想法提供他們尋求想法組合的依據，並產生最後的設計結果（圖 11）。在圖 11 中，此設計作品顯現了此尋求想法組合的線索，例如：在解決採光的設計問題，此設計者組合了古希臘的迴廊（stoas）羅馬萬神殿的圓頂、Renzo Piano 在辦公大樓設計案的遮陽板等手法，進行建築形式上的創造。

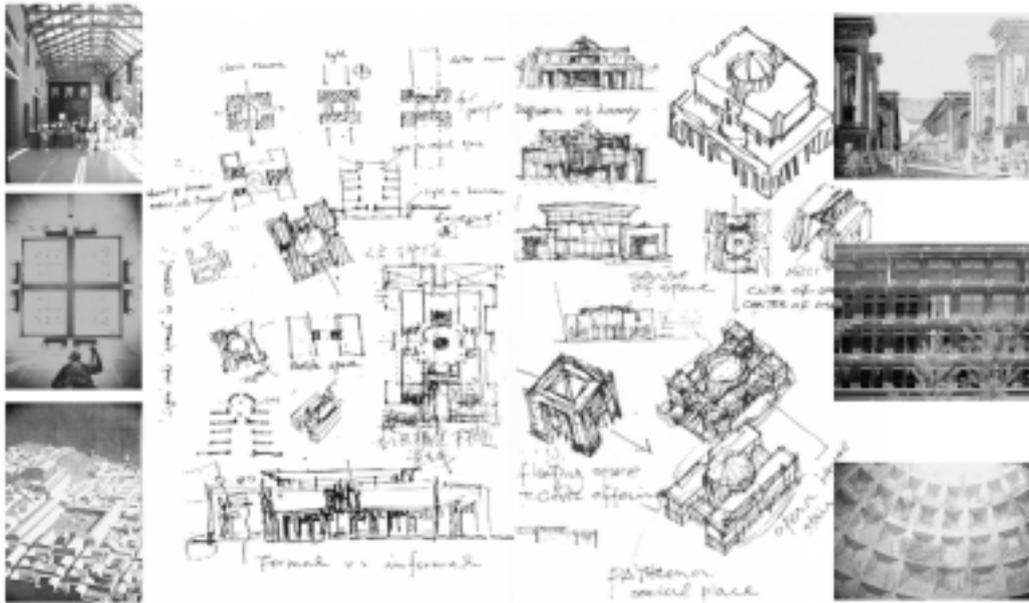


圖 10. 某行政大樓的想法草圖與使用的早期設計案例 (KHL Architect Association 提供)



圖 11. 某行政大樓的設計方案結果 (KHL Architect Association 提供)

2.3.2 語言行動

為了產生多樣性的想法，參與者會傳遞有關想法的訊息而進行彼此之間的溝通。在溝通的過程中，發送訊息的人（以下稱為發送者）與接收訊息的人（以下稱為接收者）因為彼此的互動而改變身分[105]，這種現象可以被視為參與者之間的身分狀態改變。由於，身分狀態改變會因為語言行動的差異，影響人類採取適當的

行動與反應，以回應不同的外在情境[106]，因此，語言行動提供本研究了解設計者身分狀態轉變的機制。

基本上，語言行動理論視溝通為一種行動（action），此行動被認為人類透過語言行動表現其意圖的行為機制[107]。換言之，語言行動為發送者與接收者在溝通時，使兩方對訊息產生的行動與反應。而藉由不同的語言行動，發送者與接收者才會進行傳遞訊息的相關動作。

語言行動理論開始於哲學家 John Austin[108]。Austin 認為自然語言語調（utterances-hereafter）的特定種類歸因於語言行動，而語言行動可以改變外在的狀態。為了類比於人類的實際行動，Austin 定義一些不同語言行動的動詞（又稱 performative），包括如要求（request）、告知（inform）和允諾（promise）等。架構在 Austin 的研究[108]，Searle[106]認為語言行動被執行在接收者（receiver）與發送者（sender）之間的溝通，進而將語言行動歸納成五種類型：

- 斷然性(assertive)：發送者對於傳遞的訊息，表現其自身立場的真實性，如通知。
- 命令性 (directive)：有方向性的企圖讓接收者去執行發送者委任的事情，如要求。
- 宣告性 (declarative)：帶入一些事件而導致一些狀態的改變，如宣告遊戲開始或結束，此語言行動類型反應行為在訊息傳遞過程的時間因素。
- 承諾性 (committing)：接收者對於發送者委任的事情所做的承諾行為，如答應、拒絕。
- 表現性 (expressive)：表現發送者或接收者對於傳遞的訊息，所反應的心理狀態，如感謝。

這五種類型的語言行動，其行動與反應的規則建立，除了支持參與者之間身分的改變，同時，也提供參與者在相互作用中的溝通機制[109]（有關相互作用與溝通機制在 2.5.2 和 2.5.3 小節有詳細說明）。由於語言行動為強調人類在溝通過程中，對於接收訊息所產生的行動與反應[106]，所以，行動與反應也會因為參與者之間傳遞訊息的方向與數量，而影響參與者之間的身分狀態轉變[110]。

所謂訊息傳遞方向係指發送者與接收者間之訊息傳遞的方向關係，此方向關係主要包括單向傳遞方式與雙向傳遞方式。單向傳遞方式為發送者傳遞訊息給接收

者，但接收者不會立即回傳給此發送者，而採取相關行動與反應，將新的訊息傳給下一個接收者，相對而言，雙向傳遞方式為發送者傳遞訊息給接收者，接收者會經由相關的行動與反應，產生新的訊息並立即回傳給此發送者（圖 12）。



圖 12. 單向傳遞與雙向傳遞

所謂訊息傳遞數量係指發送者與接收者之間的數量關係，此數量關係主要包括一對多、一對一與多對一的關係（圖 13）。所謂一對多關係為一個發送者同時傳遞訊息給多個接收者；一對一關係為一個發送者傳遞訊息給一個接收者；所謂多對一關係為多個發送者同時傳遞訊息給一個接收者。



圖 13. 一對多、一對一與多對一傳遞

在設計運算領域中，語言行動對於知識實體之行動與反應的機制，經常被應用在多重代理人之間的设计合作，且藉由代理人溝通語言（Agent Communication Language, 簡稱 ACL）中的 performatives 元素而執行。有關代理人與 ACL 的相關機制在 2.5 章節（連結與決策）有詳細說明。

2.4 連結與記憶

記憶支配儲存在設計空間中的設計知識，此設計空間必須藉由規則的執行而發現，同時，設計空間內的設計知識也被規則所定義，因此，記憶提供搜尋設計知識的場所，其關係著儲存中的設計知識如何被組織，以及如何連結到已存在於記憶中的設計知識。

想法聯想即是想像力連結到記憶的部分，使一個想法導致另一個想法之現象 [11]。此種連結記憶的力量，Plato 與 Aristotle 強調其為人類心理學的基本原則。基本上，記憶為儲存人類知識的容器，且記憶必須經由適當的組織，提供人類使

用記憶中的知識以解決問題[19]，為了連結記憶並有效解決問題，Schank[21]強調回想（reminding）行為與早期經驗的重要性，而在設計領域中，回想行為經常被設計者用來解決當下面臨的設計問題，同時早期經驗主要來自於設計者記憶中的早期設計案例[66, 80, 81]，因此，本章節試圖經由記憶的組織與回憶，進行了解連結與記憶在設計運算中的意義。

2.4.1 記憶組織

Simon[19]將人類記憶分為長期記憶（long-term memory）與短期記憶（short-term memory），並將記憶的基本單元定義為區塊（chunk）。長期記憶的主要功能為提供人類經由回憶而搜尋到久遠的知識，而為了有效回憶長期記憶的知識，許多研究學者提出不同的記憶組織方式組織知識，包括如 Simon[19]的命題式（propositional）結構 Quillian [68]的語意網路（semantic networks）結構或 Minsky[69]的框架（frame）結構等。短期記憶提供人類處理解決當下問題的相關知識，因此也稱之為工作記憶（working memory），此外，短期記憶在處理知識區塊的數量有它的限制性[111]。

為了更進一步了解人類記憶的相關機制，Schank [21, 22]提出動態記憶（Dynamic Memory）的理論。在動態記憶理論中，Schank 認為記憶為“人類建構對於發生事件與行為解釋的過程”，同時，他提出不同的方式組織記憶的結構，讓外界發生的問題能有效的連結到動態記憶中，而且立即的產生解決此問題的方法。Schank 提出的記憶組織包括 1) 記憶組織包裹（Memory Organization Packets）：場景基礎（scene-based）的記憶結構方式；2) 主題組織包裹（Thematic Organization Packets）：抽象且獨特領域（domain-independent）的記憶結構方式；3) 腳本基礎（script-based）：在一特定的文本（context）或一預先決定的行動系列（stereotypes sequence of actions）描述一適當系列事件（events）的記憶結構方式。

上述動態記憶理論的記憶組織方式被實踐在相關的運算領域中，並發展成不同記憶組織的結構，例如：平坦結構（flat structure）、特徵基礎結構（feature-based structure）或階層結構（hierarchy structure）[25]。

2.4.2 案例回憶

Schank[21, 22]認為使用早期經驗可以有效解決目前的問題，而回想行為提供人類有效的連結長期記憶，搜尋出早期經驗以解決當前問題的方法。因此，他提出不同回想機制去組織早期經驗的記憶，包括故事基礎（story-based）回想、交叉文本（cross-contextual）回想和目標基礎腳本（goal-based scenarios）回想。在回想過程中，人類透過回想而解決問題的主要方式是類比推理[21, 22]，另外，Carbonell[112, 113]使用類比推理建立了解決問題的運算機制，此運算機制提供案例基礎推理的基礎[25]。

在設計運算領域中，Maher 等人[80]更進一步認為，類比推理是經由連結到相關的早期經驗，促使人類瞭解當前的問題與處境，進而形成解決設計問題的方法。當早期經驗主要來自的案例知識時，這些回想方式被廣泛應用在案例基礎推理(簡稱 CBR)的運算理論，並被實際應用在不同的輔助設計系統[包括如 27, 66, 87, 92, 94]。

於案例基礎推理的理論中，Schank[21, 22]的回想機制被稱之為回憶 (recall) [25]。所謂回憶是從案例記憶 (case memory) 中進行搜尋解決問題方法的運算機制，而如何在適當的時間搜尋到合適的案例則是案例基礎推理的主要關鍵。基本上，回憶主要包括三個過程：索引 (indexing) 檢索 (retrieval) 與選擇 (selection) [80]。這三個過程的詳細說明如下：

2.4.2.1 索引

索引或稱案例的標籤，主要目的是從案例記憶中取回相關知識，可說是探索記憶結構的主要功能[66]。基本上，索引必須反應 CBR 系統在輔助解決問題之目的，以連結記憶結構中的相關案例，因此，在案例庫 (case base) 進行索引必須反應到設計問題的內容。另外，索引不能太過普遍，以免取回太多的可能案例，相反的，也不能過於特定，而造成某些可能解決問題案例的遺漏，同時，索引應該能擴展足以包含新的案例，與幫助預測新的資訊[89]。

如何建立一個好的索引方式關係於取回案例的速度與合適性，因此，有部分的研究在探討此類似的議題，例如：Oxman[66]使用故事索引 (story indexing) 的方式取代案例索引，並藉由領域概念語彙描述設計問題的內容，以解決案例索引缺乏問題形式、真實設計情境與解決方法的相互關係；Domeshek 和 Kolodner[92]透過有關索引字彙的問題，增加語彙的擴充性，而發展所謂相關基礎 (relevance-based) 的索引方式，另外，為增加檢索案例之間的動態關係，Tan[114]提出實體多重標籤 (multiple labeling) 的索引方式，以建立動態秩序的浮現關係 (emergent relations) 而取代傳統固定的分類方式。

2.4.2.2 檢索

在回憶案例的過程中，一個案例的搜尋稱之為檢索。根據新設計問題的相關內容說明，檢索的目的為鑑定相關設計案例，這些設計案例在設計過程中允許設計描述 (design specifications) 改變並且擴大，因此，檢索案例是在案例記憶搜尋的問題[115, 116]。基本上，檢索的策略包括搜尋案例記憶與發現相關案例的方法，而檢索的技術則必須依賴案例記憶的結構與索引的模式 (index schemes)，索引的計劃可以提供檢索過程較具選擇性並減少記憶容量所造成的影響，因此，不同方法組織並且使用案例記憶和索引模式導致檢索過程的不同。

Maher 等人[80]將檢索機制歸納成三種基本方式：目錄查詢 (list-checking)、概念推敲 (concept refinement) 以及相聯回憶 (associative recall)。目錄查詢是利用案例特徵的目錄決定相關案例的搜尋，而概念推敲是根據樹狀索引的案例呈現，並使用此樹狀之階層性的案例特徵進行循序搜尋，相聯回憶則以案例特徵間的關係基礎 (relationship-based) 之索引模式進行案例的搜尋。因此，CBR 系統應用這三種不同的檢索機制搜尋關聯性的設計案例。

為了從案例庫中檢索適合的案例，相似性評估 (similarity assessment) 扮演一個重要的因素。在檢索的過程中，每一個案例必須與目前的設計問題相互比較，針對每個案例中的相似程度，分別給予它們不同的等級，然後藉由檢索的搜尋機制，選取相似等級最高的案例，作為解決此設計問題的方法。基本上，相似評估大部分是藉由結構性相似 (structural similarity) 的方法進行符號比對 (pattern matching)，其包括比對案例在文字上的相似及比對其相似程度的數值。此外，為了增加符號比對上的彈性，部分學者則提出部分比對 (partial matching) 的方式，包括如文字採用不同階層 (hierarchy) 比對關係，或在數值上採用權重 (weight) 的比對方法[89]。

另外，Watson[116]將檢索案例的方法分成最近鄰接擷取 (nearest-neighbor retrieval) 和歸納性檢索 (inductive retrieval) 兩種技術。最近鄰接取回主要藉由對目標案例與來源案例去量化相似性，然而此方法雖然簡單，但有檢索速度上的問題；相對的，歸納性檢索雖解決取回速度的問題，但由於它受限於預先索引在決策樹 (decision tree) 的案例組織，所以，當案例遺失或不清楚時，可能會遇到無法檢索案例的問題。

2.4.2.3 選擇

在案例的選擇機制中，被檢索的相關設計案例必須經過相互比較與評估 (assessment) 後，“最佳” (best) 的設計案例才會被選擇用來解決目前的設計問題。Maher 等人[80]和 Leake[115]認為所謂“最佳”的設計案例是架構在最相似的比對 (the most similar match) 或最有用的比對 (the most useful match) 基礎上。基本上，案例的選擇主要可歸類成三種比對方式 1) 比對案例特徵的最大數目、2) 比對案例特徵的權重之合 (weight count) 與 3) 文本基礎的比對方式 (context-dependent matching) [80]。這些不同的比對方式主要為定義案例選擇的準則。

在比對案例特徵的最大數目中，此策略為在檢索的設計案例之間計算案例特徵的比對數量，當檢索的案例具有最高案例特徵的比對數量時，則此案例可視為最適合此設計問題的解決方法；在比對案例特徵的權重之合中，此權重特徵的方式允許設計者清楚地說明他/她關於案例特徵比對的優先權 (priorities) (例如，案例的

喜好程度) 選擇合適的案例；在文本基礎的比對方式中，則根據文本的內容量測關聯性重要性，並藉由這些檢索的案例確定新問題的特徵中哪個才是最重要的比對。有關索引、檢索與選擇應用在不同設計領域之輔助系統(主要為 CBR 系統)的運算機制，在第三章將有更進一步的說明。

2.5 連結與決策

決策是所有搜尋機制的指導方針，此指導方針為擔任設計過程的流程控制，同時，也為設計者和電腦在設計空間中相互控制的關鍵性元素。因此，決策被視為“改變控制”(meta control)的準則，以及控制設計過程互動於設計情境的策略，而在分散式的想法連結中，決策須強調分散式知識的連結過程該如何被控制，以便反應於設計情境。

設計行為會隨著設計情境而有所改變，進而影響設計過程[117, 118]，因此，在與不同參與者連結想法的過程中，設計者為了和設計情境的條件(包括如設計問題、參與時間、參與人數等)有所互動，設計者經常透過扮演不同的角色，連結個人的長期記憶與其他參與者的設計知識，以產生關聯性的設計想法，以便解決設計問題[103, 104]。例如，在一個有關住宅設計的腦力激盪會議中，為解決採光的設計問題，當設計者接收到其他參與者的設計想法時，她/他有時會扮演業主的角色去連結有關預算上的想法，有時也會扮演著不同設計專家的角色以產生設計想法，在此過程中，參與者扮演不同的角色並根據產生想法的順序，其扮演角色的行為會持續的進行直到限定時間終了，整個角色扮演的遊戲才會結束。因此，不同改變控制的指導方針引導著整個想法連結的過程，相對的，也影響到整個設計最後的結果。

因此，藉由角色扮演的觀點探討多重參與者與設計情境之間的互動性行為，其幫助本研究了解分散式知識的連結過程與控制策略。在認知心理學的領域中，Yardley-Matwiejczuk[119]的角色扮演(role-play)理論提供重要的見解，有助於了解上述連結想法的過程與控制的策略，包括如多重知識之分散性、知識分享、沉浸與情境分析等特質。另外，Chang[120]的相互作用理論提供作者建立角色與設計情境之間的互動關係。

2.5.1 角色扮演

角色扮演是在一個虛擬的情境中，獲得人們主觀性的知識與情境中互動行為的研究方法。Schank[121]認為角色扮演是為達某一種目的，利用扮演角色的方式融入此目的的情境，讓學習者依據劇本中規範的角色與任務，來選定角色並逐步設法完成任務，因此，角色扮演可以被視為一種以目的導向且融入情境的問題解決方式。此外，透過角色扮演的理論，Yardley-Matwiejczuk[119]認為在假設或模擬的行

為及環境中，可以進行分析人們產生行為的過程及其使用知識的方法。在設計領域中，由於角色扮演具有參與者進行設計合作的分散式與情境式的特質，因此，有部分學者[122, 123]利用角色扮演的機制，進行有關分散式設計的情境式互動關係之研究。Chang[120]認為這些角色扮演的相關研究可以歸類三種不同的應用，包括社會行為、真實模擬和衝突溝通。

在社會行為的研究中，角色扮演是根據情境的文本 (context) 與人在情境的位置，進行了解個人行為與情境之間的互動，因此經常被應用在有關社會性行為的心理研究。在心理研究中，Moreno[124]認為角色扮演有關代理 (agency) 機制、反射性與自發性的特質，是透過“好像” (as-if) 或“沉浸” (immerse) 的融入性概念而達成，此外，Moreno 認為角色扮演為個人與其他人雙方面的互動，並經由扮演角色的不同解譯，每個角色對於不同情境是具有反射性的特質。因此，融入性的概念，經常被應用在了解參與者在設計過程中的主觀性知識，並允許參與者根據他們的內在知識與外在設計情境進行反應[122, 123]。

另一方面，角色扮演強調真實模擬的特質。此真實模擬是由一些瞬間的真實印象 (snapshots) 內容，透過參與角色扮演的參與者之心理而重新建構，而劇本 (scenario) 或故事 (narrative) 敘述，採用連續性的分析過程進行了解此角色扮演中真實印象的內容。在建築設計領域，一些學者[123, 125, 126]使用角色扮演為隱喻，進行設計遊戲的模擬，藉由模擬參與者的可能情境、角色性格的互動控制與角色扮演規則的設定，以了解不同參與者在設計決策過程中有關設計行為與知識之間的瞬間關係。

在衝突溝通的研究中，Stanislavski[127]認為一個演員在戲劇中的行動發生在舞台中與其他演員之間的衝突，經由連續性面對面溝通所形成的行動序列，提供演員彼此之間了解衝突與採取適當的反射行為。因此，在此行動序列中，任何單一的事件 (event) 都會造成演員之間的衝突，這些演員之間衝突的反射行為將強化角色之間的溝通過程。因此，角色扮演被視為一個以衝突 (或特別事件) 為主的劇本基礎互動，其衝突機制會根據參與演出角色之差異而有所不同，並透過一些隱喻性元素 (包括如角色、導演、場景與腳本等) 予以描述[127, 128]。在這些隱喻性元素中，腳本描述在一個特定文本中 (例如角色扮演的戲劇文本) 事件的序列，其提供角色互動於不同戲劇情境與過程的組織性資訊[129]。

根據上述角色扮演相關應用的論述，角色扮演可以被視為在行動與反應 (acting/re-acting) 的過程中建構一個“真實生活”經驗或事件的方法[120]。同時，Lai 和 Chang[104]認為上述角色扮演的相關研究，可以歸納成三個重要機制：情境 (situation)、沉浸 (engagement) 和場景基礎互動 (scenario-based interaction)。情境為當人類遇到真實生活的條件或情況；根據此情境，人類相互之間進行作用

與反作用，此互動稱之為場景基礎互動；沉浸提供人類“好像”（as-if）置身於表演之中演員的真實感受。

- 情境：角色扮演的情境特質可以進行了解當設計者遇到不同的設計情境時的行為。此設計情境影響參與者對於扮演角色與想法聯想原則的選擇。
- 沉浸：誘導（induction）為角色扮演沉浸特質的主要行為。藉由不同角色的誘導，設計者可以沉浸於真實的設計情境進行想法的連結。
- 場景基礎互動：架構在一個特定的文本內容中，不同的設計者彼此之間進行互動，同時也互動於設計的過程與文本的內容。

這三個機制提供作者了解分散式知識的想法連結過程。在想法連結的過程中，情境的機制提供想法連結，在內部的知識和外部的知識之間，予以區分成一系列遊戲劇和相互作用的關係；而場景基礎互動的機制使用這些關係，描述想法連結的過程；而沉浸的機制為實現連結不同知識和想法之概念。另外，Chang[120]認為角色扮演的過程伴隨著角色之間的互動，而相互作用（interplay）為強調互動行為的概念，此概念有助於本研究更一進步的了解角色扮演，如何控制分散性知識彼此之間連結的機制。

2.5.2 相互作用

Chang[120]認為設計過程需要參與者（包括人類或電腦），經由和文本情境與特定設計知識了解而產生互動，此互動行為稱之為相互作用，例如：集體腦力激盪會議可以被視為相互作用的過程，此過程是多重訓練（multi-disciplinary）的參與者為產生設計想法，而彼此之間進行相互作用。簡而言之，相互作用必須伴隨著資訊的發生，與不同參與者之間進行互動的行為。

相互作用是一種互動的行為，這個行為牽涉一些必須應用到反射性動作（包括行動與反應）的複雜問題[120]，因此，相互作用應用在角色扮演時，當角色介入的方式已被清楚定義後，此相互作用的情境就會被引導出來，進而產生解決設計的方法。例如：在一個集體腦力激盪的會議中，一群設計者在進行相互作用時，在連結設計想法的過程中，當一個設計者產生想法時，其他的參與者會根據此設計想法進行反射性的動作，進而產生新的設計想法[27, 130]。

根據相互作用的理論，Chang[120]提出一個行動角色模型（Acting Role Model, 簡稱 ARM），以提供分散式知識在設計過程中彼此互動之運算化基礎。ARM 的基本想法是來自角色扮演的理論，基於角色/演員和舞台/發生地點的分離，ARM 根據

他們的能力和情境式反射去描述個別的互動性，而不同的角色代表著不同的知識和行為去反應不同的設計情境。此外，在演員和角色之間有多對多的關係，這個關係可以透過多元情境的描述能力表現在最後的設計結果。

基本上，ARM 架構主要有由三個部份構成：1) 元素部分、2) 知識基礎與 3) 相互作用過程。每一個元素擁有特定相互作用的特質（互動知識基礎），予以反應最後的設計結果（或稱 play）。元素部分主要包括：角色、演員、舞台、腳本與場景，每一種元素有其本身的機制，並在一個特定互動的機制中彼此之間具有相互作用的關係（圖 14）。

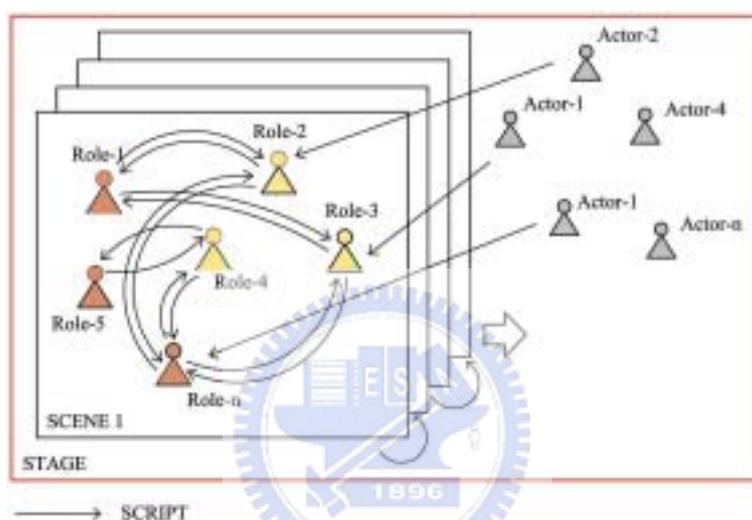


圖 14. 集體腦力激盪的相互作用（來源自 Chang 和 Lai, 2004）

知識基礎包括三種形式：角色技能和知識、控制策略（control strategy）和設計結果，這三種形式分別結合演員（或角色）、場景和舞台，賦與它們相互作用的相關機制。在一特定時間裡，這些知識基礎相互影響呈現整個戲劇內容，例如：集體腦力激盪的相互作用過程中（圖 14），這些元素結合獨特的知識基礎，與其情境達到動態的互動[18]。在相互過程中，相互作用行為主要有三種類型的互動關係，包括（1）角色對自我；（2）角色對角色，和（3）角色對環境[120]（圖 15），因此，在想法連結的過程中，設計者扮演不同的角色去產生設計想法，除了設計者本身與角色之間的相互作用外，同時此角色也會與不同角色與外在環境（如不同參與者或設計問題）進行相互作用。

在設計運算領域，當代理人為一個具有自主性（autonomous）、反射性（reflective）與溝通性（communicative）的知識個體時[131]，不同代理人可以自主性的在分散式的環境中進行溝通且採取適當的反射性行動。這些特性結合角色扮演的機制，可以提供多重代理人（電腦）與設計者（人類）在分散式的設計環境中自動地參與角色扮演，且在不同的設計情境產生互動，進而連結與產生多樣性的設計想法。

因此，藉由了解代理人在分散式互動的運算機制（包括如代理人建構與組織、訊息傳遞、語言行動等），有助於具體化本研究之分散式想法連結的機制。

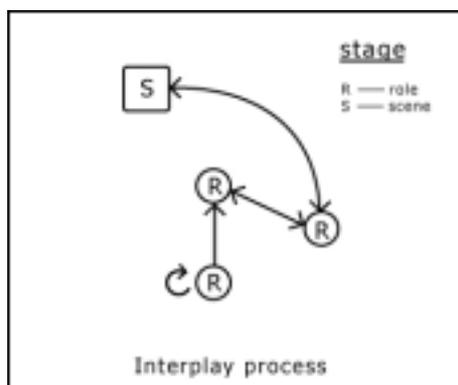


圖 15. 三種類形的互動關係 (來源自 Chang, 2006)

2.5.3 代理人

以軟體代理人 (software agent, 簡稱代理人) 為基礎的分散式人工智慧 (Distributed Artificial Intelligence) 是目前最為廣泛應用的技術[29]。然而，到目前為止，代理人並沒有一個明確的定義，因此，許多研究學者藉由探討代理人的不同行為能力，來定義所謂代理人機制，並進而解決相關的問題[29, 132, 133]。在這些學者提出的代理人行為中，除了普遍強調代理人需具有自主性與反射性的行為能力外，而為了不同代理人能彼此相互合作，Wooldridge 和 Jennings[131]認為代理人也需具備社會的行為能力，因此，Wooldridge[107]根據這些代理人的機制提出與早期專家系統的差異性，其差異性包括 1)專家系統必須借由使用者為中間人 (middleman) 去互動外在的環境；2)專家系統沒有反射性的行為能力；3)專家系統沒有社會性行為，例如合作、協調與妥協等行為。

Woodridge 和 Jennings[131]對代理人的定義，後來被廣泛的應用在不同設計領域上的問題，包括如有關代理人設計行為的認知研究[包括如 30, 134, 135]、代理人與網際空間的互動研究[包括如 136, 137, 138]、與分散式設計的機制建立與應用[包括如 31, 32]。其中代理人在分散式設計的機制，為強調多重知識實體之間的设计合作，與系統實際的執行與應用，有助於作者了解並具體化分散式想法連結機制在運算化的可行性。

2.5.3.1 代理人與分散式設計

基本上，代理人架構的建立主要決定於領域問題本身與分散性行為[29]。例如，Ligtenberg 等人[139]使用多重代理人的模擬 (multi-agent simulation) 來決定都市平面

分配的問題；Liu 等人[32]創造出一合作性的設計環境，提供設計師與軟體代理人在設計過程中進行資訊交換等的互動行為，藉由一個層級化的多重代理人系統建築與代理人動態與靜態的知識結構，建構一個動態管理代理人的合作性模型；Anumba 等人[31]以鋼的框架結構設計問題為例，根據鋼結構設計的內容（context），決定此合作過程中多重代理人不同的角色（包括如結構設計代理人、建築服務代理人等），與多重代理人在合作過程中進行互動與妥協的機制；Aly 和 Krishnamurti[140]所提出的物件就是代理者（object-as-agent）的主動性概念，將設計中的門與窗視為代理者與設計師進行。

在這些不同多重代理人系統的例子中，每一個多重代理人系統的不同代理人必須經由適當的組織以進行分散式互動。基本上，代理人架構的組織方式可以分為階層性（hierarchical）方式與層級性（layered）方式[107, 141]。在階層性組織方式中，主要將一個設計工作分解成不同且具有階層性關係的工作單位，每一個工作單位分別由特定的代理人負責，根據這些工作單位之間的從屬性（dependence）關係，這些代理人在各階層性的組織中進行互動性溝通，以解決設計問題，例如：Anumba 等人[31]與 Liu 等人[32]建立的代理人架構。

在層級性的組織方式中，主要將設計工作分解成不同工作單位，並將這些工作單位分別放置於不同層級，在每一個層級中，分別由一個以上的代理人共同處理相同的工作單位，根據這些工作單位之間的層級性關係，Muller（1996）認為這些代理人在各自的層級與不同層級之間，進行互動性的溝通予以解決設計問題，例如：Ligtenberg 等人[139]與 Aly 和 Krishnamurti[140]建立的代理人架構。另外，為強調設計過程的分散性互動，Chang 和 Lai[18]所發展的 DARIS 系統架構，結合角色扮演理論，與兩種不同的代理人組織方式而達成。

在上述不同多重代理人架構中，每一個代理人都能自動的反應設計情境與其他代理人的知識，同時，代理人之間的合作形成一個分散式的知識連結關係。為建立這些分散式知識的連結關係，代理人之間的溝通是一個重要的機制，而代理人溝通語言則是代理人進行溝通的主要運算機制，此運算機制提供控制代理人相互作用的策略。

2.5.3.2 代理人溝通語言

在運算領域中，代理人理論是強調彼此間的溝通[107]。為支持代理人之間的溝通，代理人溝通時所使用的語言稱之為代理人溝通語言，而於2.3章節（連結與規則）提及的語言行動，提供代理人溝通語言機制建立的基礎[107]。為了建立多重代理人之間的溝通，並強調這些代理人與不同系統之間的互動，有很多代理人溝通語言的溝通模式（formalisms）已經被發展而成[142, 143]，主要包括KQML[144]、FIPA[145, 146]和MASIF[147]。

KQML (Knowledge Query Meta Language) 是第一個使用Searle[106]的語言行動理論，並實際應用在代理人進行溝通的語言[107]。雖然，KQML是目前許多代理人平台最廣泛使用的代理人溝通語言，且已被發展需多不同的版本，但由於每種版本未被標準化，因此，存活 (live) 在不同代理人系統的代理人無法進行彼此之間的溝通。

FIPA (The Foundation for Intelligent Physical Agents) 為一種強調代理人溝通之標準化的代理人溝通語言。此標準化為建立代理人能在內部 (within) 的代理人平台，和跨越(across)的代理人平台中，進行最大可能性的系統互操作性(interoperability) [148]。因此，FIPA提供指定的規格，允許使用者可以自由地產生她們/他們自己的代理人軟體架構。

基本上，KQML和FIPA是訊息基礎 (message-based) 的代理人溝通語言，主要由 performatives 與不同屬性共同定義的模式，包括如發送者 (sender)、接收者 (receiver)、內容 (content)、語言 (language) 與實體論 (ontology)。performatives 為代理人進行溝通時作用或反作用的語言行動，發送者與接收者為分別指定特定發送與接收訊息的代理人，內容為代理人之間訊息傳遞的內容，語言為表現此訊息內容使用的語言和語法，而實體論為訊息使用的專門術語 (terminology)。例如：FIPA代理人溝通語言模式的語法如下：

```
(inform
  :sender agent1
  :receiver agent2
  :content (price good2 150)
  :language sl
  :ontology hpl-auction
)
```

MASIF (Mobile Agent System Interoperability Facility) 是強調代理人可以在不同的代理人平台之間，進行移動性的代理人溝通語言，但無法與非移動性的代理人進行溝通，換言之，在不同代理人平台中的代理人，MASIF限制這些代理人進行異質 (heterogeneous) 的訊息傳遞。

總而言之，KQML 和 FIPA 是強調代理溝通語言定義代理人之間的相互作用，而 MASIF 則強調代理人在遠程方法引用 (Remote Method Invocation) 的移動性特質。但由於 KQML 與 MASIF 的 performatives 未能被標準化，不能提供代理人在不同的代理人平台進行系統互操作性，因此，對於不同代理人存活在不同的代理人平

台，進行知識的連結是有它的限制性。比較於 KQML 和 MASIF，FIPA 標準化的特性可以提供分散式環境中的設計合作，例如分散式的想法連結。

除了標準化的特性，Poslad 等人[149]認為 FIPA 將 performatives 簡單化成 20 個標準的 performatives，可以提供彈性的代理人溝通方式，包括如分散性（或非集中性）的點對點（peer-to-peer）提供更多分散知識的連結，普遍性的訊息基礎語言方式提供代理人一致性的語言行動介面，非同步性（asynchronous）的訊息基礎互動支持不同代理人與使用者（人類）之間的溝通。

2.6 小結

從 Simon 和 Newell[47]開始，「設計搜尋」是在設計運算領域中廣泛被接受的概念。當我們視連結機制反映「設計搜尋」的概念時，Woodbury[26]提出的「探測」理論具體的建立在設計空間的搜尋機制，其所包含的四個搜尋機制（呈現、記憶、規則與決策），提供本研究了解及建立連結機制在設計搜尋的意義與機制。

「呈現」關係著設計須包含的資料結構，主要包括設計知識的呈現與組織方式。在連結機制中，呈現須反應設計知識（設計想法）的關聯性建構，而概念地圖[63]強調概念與概念之間的關係，有助於單一與多重的知識實體在分散式的設計環境連結設計想法。另外，設計想法主要來自早期設計案例的概念性知識，而結合領域概念語言與多媒體的媒材表現，有利於多重知識實體之間的设计溝通。

「規則」支持設計狀態之間轉變的機制。古希臘人建立想法聯想的三個原則（包括相似原則、對比原則與相鄰原則），提供連結機制在設計想法之間轉變的方式與限制條件。此外，語言行動理論[106]則支持參與者在連結想法的相互作用中，藉由改變他們的身分狀態，以採取適當的反射性動作（包括行動與反應）連結並產生設計想法。

「記憶」則負責儲存搜尋過程及其設計空間的資料。在連結機制中，透過適當的組織儲存在記憶中的設計知識，知識實體使用回想的行為，連結儲存在他們各自記憶（包括長期記憶與短期記憶）中的早期設計案例知識，或其他知識實體所產生的設計知識。此回想行為可以藉由 CBR 理論[25]中有關「回憶」的運算機制（包括索引、檢索與選擇）而了解。

「決策」擔任設計過程的流程控制。在連結機制中，決策關係著多重知識實體連結分散式知識之控制過程，而角色扮演理論[119]提供分解設計工作並組織多重知識實體的方式，同時，控制連結過程互動於不同設計情境的策略。此外，相互作用支持多重知識實體，彼此之間進行溝通而產生關聯性和衝突性的關係建立，而

ARM[120]提供此相互作用運算化的可能機制。這些相關於決策的機制，可以經由多重代理人之分散性運算技術，與代理人溝通語言而達成。

上述提到的連結機制，構成本研究分散式想法連結的基礎。基本上，此分散式想法連結可以被視為一種分散式的搜尋機制，在此機制中，主要透過知識實體（人類或機器）在各自的記憶中進行設計搜尋，並將搜尋到的這些設計知識，分散於多重知識實體所建立的設計環境中，且經由知識實體彼此之間的溝通，建立分散性設計知識之間的關聯性。因此，上述連結機制中的「記憶」可以結合「規則」中的三個想法聯想原則，提供多重知識實體在他們各自內在的記憶中，進行「動態性回憶」（簡稱回憶），以連結早期設計案例的概念知識，並產生解決問題的設計想法。

而連結機制的「決策」可以結合「規則」中的語言行動，提供多重知識實體在外在的設計情境中進行「互動性溝通」（簡稱溝通），以連結不同參與者的設計知識，同時，支持參與者之間的相互作用與他們在設計情境上的互動。最後，此分散性想法連結機制可以精練成由「呈現」、「回憶」與「溝通」三個主要機制所構成，而它們的關係可以藉由下列的等式表示之。

$$\text{呈現} + \text{回憶} + \text{溝通} = \text{分散式想法連結}$$

為了更進一步了解此分散式想法連結機制在設計運算領域中意義，作者將在下一章（第三章）進行有關於輔助呈現、回憶與溝通的輔助設計系統之文獻回顧，以期獲得相關的運算化機制，並建立一個輔助分散式想法連結的運算模型。

三、文獻回顧：探索輔助分散式想法連結系統

根據本研究的研究目標以及上一章論述之分散式想法連結機制（詳見 2.6 章節），本研究輔助系統建立必須具有三個主要的運算機制：呈現、回憶與溝通。因此，本章節主要針對 1) 輔助呈現、2) 輔助回憶、3) 輔助溝通相關的輔助設計系統進行文獻回顧，經由這些輔助設計系統，學習它們有關輔助三種連結機制的數位技術與運算機制。

回憶與呈現主要強調單一知識實體，如何透過反射性與關聯性的連結機制，使用早期設計案例知識進行想法連結，溝通則強調多重知識實體之間，如何藉由關聯性或衝突性的連結機制，在分散式的設計環境中進行想法連結。在設計運算領域中，案例基礎推理（以下稱為 CBR）的運算機制，是透過運算化知識呈現與回憶過程，進行搜尋適當的設計案例知識以解決設計問題，而以代理人理論發展的多重代理人的運算機制，為經由代理人之間的互動性溝通，進行分散式的設計合作。因此，相關之案例基礎推理系統或多重代理人系統，分別提供本研究進行輔助呈現與回憶，和輔助溝通之文獻回顧的依據。

關於文獻回顧的方式，首先對於每個系統案例設計的動機和目的，進行簡短的背景介紹，然後，根據分散式想法連結的機制，進行系統在相關運算機制的探討。最後，作者針對每個系統案例所提供的運算機制，架構在上述分散式想法連結的機制（呈現、回憶與溝通），進行這些系統之運算機制的相互比較與討論，以達到提供本研究建構輔助分散式想法連結模型與系統的目的。

3.1 輔助呈現

在進行有關輔助呈現與回憶之 CBR 系統案例的文獻回顧之前，作者先說明 CBR 系統案例選擇的依據。自從 1981 年，Carbonell[112, 113]使用類比推理所建立解決問題的運算機制，奠定了 CBR 運算理論的基礎，並啟發了許多建築設計領域的研究學者，使用 CBR 的運算技術輔助設計者解決相關的設計問題[80]。基本上，Maher[91]將 CBR 在建築設計的輔助方式歸納為二種類型：1)藉由設計案例的回憶，以解決建築設計問題，包括使用生成（generative）平面排列為設計案例的 SEED[94]，和使用早期設計案例的 Archie-II[88, 92, 150]、PRECEDENTS[66]、WEBPAD[90]、FABEL[93] 和 DIM 初期模型[27]；2)透過設計案例的調整，以解決建築設計問題，包括 CADRE[87]與 IDIOM[151]。

架構在上述分散式想法連結的機制，使用早期設計案例進行回憶之 CBR 系統案例，將為本研究主要的文獻回顧範圍，因此，作者選擇 Archie-II、PRECEDENTS

和 FABEL 進行輔助呈現 CBR 系統之文獻回顧，另外，結合網際網路的 WEBPAD 系統也在文獻回顧的範圍。在進行上述系統有關呈現機制的文獻回顧中，本研究主要強調早期設計案例內容如何反應設計問題、案例知識如何呈現並儲存在電腦的資料庫、和案例知識如何表現以幫助使用者了解。有關設計案例的記憶組織由於關係到回憶機制，因此，在 3.2 章節（輔助回憶）再予以探討。

3.1.1 Archie-II

Archie-II 是一種輔助建築師進行設計的 CBR 系統，由喬治理工學院的人工智慧研究室（AI Lab of Georgia Tech's College）與建築系共同合作建立。此 CBR 系統源自於早期發展稱之為 Archie 的 CBD 輔助系統，而 Archie 為最早應用在建築設計領域的 CBR 系統。Archie-II 主要提供建築師有關公共建築的早期設計案例，以便輔助他們在早期設計階段進行公共建築的設計。在 Archie-II 系統中，設計案例主要的任務是提供建築師在早期設計階段相關設計問題的解決方法、可能犯錯的警告以及相關評估的原則。

設計案例

由於，目前建築設計的回饋對未來設計的用途經常被忽略，所以，評估性的資訊可以提供建築師在早期設計過程中，使他/她更知道設計決策暗示性（implication）的紀錄。因此，Archie-II 系統的案例知識，主要來自於對實際完成的建築設計進行評估所得到的資訊。而設計決策的結果，來自於一棟建築物有關設計執行、建築使用與維護的使用者，藉由儲存這些使用者的故事與故事之間的關係，Archie-II 啟發建築師對於建築生命週期的重視。

在 Archie-II 中，設計案例主要包括一些法院和圖書館的公共建築設計案例。在這些建築設計既有的描述（包括藍圖和說明），主要強調設計的評估，而 Archie-II 儲存的內容包括成功故事和負面的資訊，都是建議建築師在早期設計過程中遵循或避免的準則。但由於法院或圖書館的設計描述過於龐大，並不能有效幫助建築師做設計決策，因此，Archie-II 將這些案例評估經由分解成容易管理的知識區塊（包括故事與準則），予以組織 Archie-II 的案例資料庫。

案例呈現

為了解 Archie-II 的案例知識呈現，作者藉由 Archie-II 使用者介面的不同視窗，予以說明這些知識區塊的呈現方式（圖 16）。基本上，Archie-II 的使用者介面的視窗被分成 8 個主要框架（frame），在左上角裡的黑框架是組態控制板（mode control panel），使用者可以在 Archie-II 中包含的五個組態進行選擇，這五個組態為訊息庫（library）、說明（description）、平面（plan）、筆記本（notebook）和歷史（history），

例如在圖 16 中，使用者目前已經選擇平面組態，而被選取的組態會影響此框架下的第二個控制面板的標籤和內容，此外，Archie-II 也提供視窗，供使用者瀏覽（包括如移動或變焦）CAD 繪圖的建築平面。

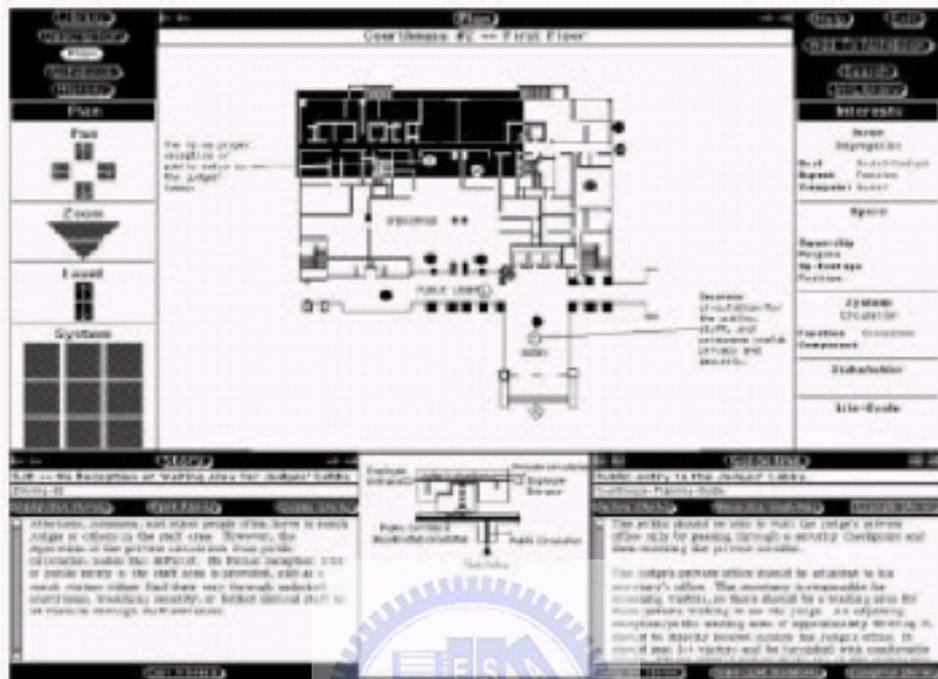


圖 16. Archie-II 使用者介面（來源自 Domeshek 和 Kolodner, 1997）

在中間大的框架內容也會隨著選擇的組態而改變。在上述平面組態例子中，它使用一些經過註釋的簇群故事，說明一座特別建築物的平面圖。在黑色右上角裡的按鈕提供使用者一些基本的系統選擇，包括如幫助鍵、退出鍵或記憶體的搜尋鍵。另外，興趣（interest）框架允許使用者指定搜尋的標準，例如：在圖 16 中，為使用者對於動線系統如何影響建築區域分割的故事感興趣。根據這些相關興趣的結合，系統自動的選擇註釋的故事並展示在主要的框架中。

故事主要展示在螢幕底部的左邊框架中，在此故事框架中，除了完整的故事正文之外，還包含故事的來源與標題，和一連串有關於故事和方針的連結，而方針被展示在右邊底部的框架中。另外，在故事和方針框架之間的方形部分裡，故事和方針可以被更進一步的詳細說明，例如，在這個顯示視窗中，為一個抽象化的法院平面圖，主要強調故事中有關不同區域和動線路徑的議題。

3.1.2 PRECEDENTS

PRECEDENTS是由Rivka和Robert Oxman[96]所發展的CBR系統，主要應用在建築設計教學。此系統主要強調在早期設計概念發展階段，有關博物館空間結構(spatial

organization)⁴[152]的設計問題。此系統的案例知識呈現重視早期設計案例有關博物館的概念知識的呈現，包括如設計想法、外在概念觀點與相關的設計原則，因此，學生可以藉由運算化的早期設計案例圖書庫，了解早期設計案例的概念知識與案例之間的關係。

設計案例

PRECEDENTS儲存過去有關重要博物館設計的空間組構想法。就如同Archie-II中的法院和圖書館，這些博物館案例所包含既存的圖像式描述，對於教導學生有關博物館空間組構的概念過於龐大，而且，由於這些案例沒有明確表達設計後面的概念，使它們的解譯完全倚賴使用者對於設計領域的熟悉度，因此，Rivka和Robert Oxman[96]決定透過設計故事（design stories）把這些選擇的設計分解，使這些案例成為更容易管理的區塊。然而，不同於Archie-II，PRECEDENTS只使用一種形式的設計故事。基本上，一個設計故事為文字註釋的片段，主要描述一個設計唯一的概念性論點，此論點來源自於博物館案例中有關早期設計概念特別部分的意義，例如：

“The *urban site* in Stuttgart which was given for Stirling to design the municipal museum consists of two main parallel roads that requires *public access* in order to provide for *urban continuity*. The museum, on the other hand, requires a *private access* system that does not accommodate public circulation. Stirling resolved this conflict in a unique and interesting way by cutting the building and creating a *path through* the building around an central court.”(Neue Staatsgalerie, Stuttgart, James Stirling)

這例子說明一個設計故事如何透過一個特定的設計概念（path through）連結一個特別設計議題（urban continuity），而穿越中央中庭（central court）之兼具公共性與私密性的路徑，予以具體化“path through”的設計概念。這些在議題、概念與形式之間的連結實際上建構了系統中有關早期設計案例的知識內容。此外，不同的設計故事可能被註釋在一種單獨的設計中，同時，一個案例的設計故事之議題或概念可能也會適用在其他案例的設計故事。例如，“urban continuity”的議題除了啟

⁴空間組構為建築設計概念發展中扮演著重要的角色，透過分析不同的建築構件與其之間的關係，構成了建築空間的意涵與形式（Ching, 1996）。另外，Ching（1996）與Clark和Pause（1985）認為空間組構為反應建築空間的秩序與關係，主要透過空間的構成元素（包括形式、空間、動線、組織、比例尺寸等）而達成。

發斯特林的Staatsgalerie設計案例的中央中庭（central court）外，也可以啟發於柯比意的Visual Arts Center設計案例的穿越走道（walkway）（圖17）。因此，PRECEDENTS所收集設計故事的內容，主要藉由分析博物館設計案例的關鍵作品文章內容之建築領域概念語彙而形成。

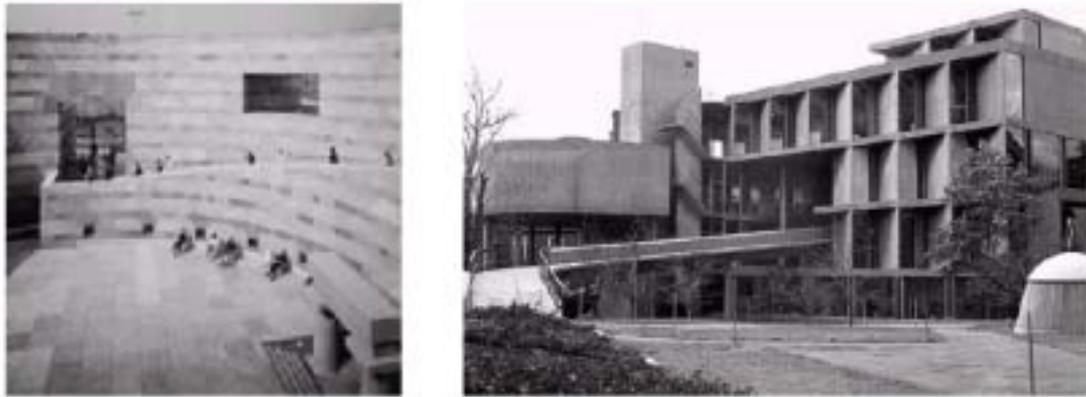


圖 17. James Stirling 在 Stuttgart 設計的 Staatsgalerie 美術館（1977-1984）（左側）與柯比意在 Cambridge 設計的 Visual Arts Center（1961-1963）（右側）（來源自 Heylighen, 2000）

案例呈現

基本上，PRECEDENTS的案例知識並不是在於案例本身，而是透過分解的概念區塊，其稱為設計故事。基本上，設計故事的呈現主要架構在issue-concept-form（簡稱ICF）的知識呈現概要，而ICF的知識呈現概要則由議題（issue）概念（concept）與形式（form）所構成。議題是一個設計案例中必須被強調的設計問題（例如：在Staatsgalerie博物館的“urban continuity”議題），概念是相關這些議題之抽象的解決方法（例如“path through”），而形式則是具體化這個解決方法之設計人工製品（artifact）（或稱為設計手法），例如：“the ramp in the central court”。

早期設計案例通常使用圖解的形式呈現，以提供一個完整架構的設計解決辦法。對這張圖解的插圖來說，幾個設計故事皆以簡短正文的形式予以註釋，這使得圖解的呈現具有雙重的作用（包括圖解說明被註釋的設計故事，和將這些故事予以簇群成早期設計案例），而這些設計故事的描述主要來自於領域概念語彙，也就是所有關於建築設計中空間組構之核心概念，例如：都市連續性（urban continuity）、方向性（orientation）等，透過進行建築文學（architectural literature）的內容分析予以定義。因此，在PRECEDENTS中，一個設計故事的知識呈現包括三個組成部分：一個文字片段（實際故事）、一張相對應早期設計案例的圖像實例（illustration）、且一個議題和概念的網路連接來自其他早期設計案例的故事（圖17）。

PRECEDENTS 系統的 ICF 知識呈現概要，可以應用在到不同建築類形的案例知識，如公共建築、住宅或教堂等，容易建立不同案例知識之間的連結關係，有助於單一知識實體，彈性與開放性的組織記憶中設計知識。但由於案例知識的分析與建立過於主觀性和集中性，對於不同知識實體進行分散式的想法連結，缺乏對案例知識解譯的開放性，相對的影響產生設計想法的多樣性。

3.1.3 FABEL

FABEL 由一個德國國家資訊科技研究中心(簡稱 GMD) 協會所建立，此系統的主要目的是輔助建築師與工程師，解決建築設計中有關設備的複雜問題。FABEL 主要由不同的工具所建構而成，這些工具稱之為專家 (specialists)，每位專家分別負責案例基礎推理的過程 (例如回憶與調整)，同時，這些專家擁有自己的案例庫、案例呈現與記憶組織，彼此共同合作在一個虛擬的環境中。因此，每位建築師與工程師能根據本身的需求，搜尋到設計的相關資訊。

設計案例

FABEL 主要案例資料為建築案例的 CAD 圖畫。這些圖畫包含了龐大的設計組成元素，例如：牆，窗戶，門或樓梯等。由於這些 CAD 圖畫的執行與管理非常困難，因此，這些圖畫被分解成較容易管理的片段，也就是元素的子集 (sub-sets)，這些子集構成 FABEL 基本的案例。原則上，這個基本元素可以藉由增加設計者個人的意見、需求與限制而被補充和擴大。

案例呈現

為了案例的回憶和調整，有兩種方式呈現 FABEL 的案例知識：設計組成元素 (也就是原始的形式) 的收集，和一些專門性與具體性的解釋。這些設計組成元素被儲存成具有幾何性和語意性的 CAD 物件 (CAD-object)，幾何性決定物件的尺寸和位置，語意性包括設計組成部分的子系統 (如空間、構造、樓層、屋頂、天花板、牆、與其它建築設備構件)，功能 (如使用性、連接性)、提案 (如把分區、元素)，和規模 (如建築基地，整體建築、樓層、房間)。

除了上述說明的案例呈現外，FABEL 還有七種不同“色調”(coloured) 的案例呈現，也就是，透過 FABEL 專家特定的案例解譯，這些案例呈現包括兩類的特徵向量 (feature vectors)、兩類的型態 (gestalts)、條件、圖像和點陣圖。為了考量效率，在用進行案例回憶與調整之前，這些專家的案例呈現在處理期間被建立，同時，動態的儲存在專家的案例庫中。雖然，這些不同“色調”的案例知識呈現，可以提供設計者使用不同方式的解譯案例，但解譯方式的選擇過於主觀性，因此，只能限定在特定的建築類型與問題解決。

3.1.4 WEBPAD

WEBPAD 是網路基礎 (web-based) 的設計系統，結合了 CBR 理論與電腦輔助設計技術[28, 90]。此系統的主要目的是提供建築師在分散的網路設計環境中，得到 CAD 繪圖檔案的相關設計，以便設計者或設計工作室可以發展自己的案例庫。此外，WEBPAD 系統也延伸 PRECEDENTS 的系統功能，並發展成一個應用於輔助建築設計學習的設計工具，稱之為 Think-Maps[28]。

設計案例

基本上，WEBPAD 系統根據 PRECEDENTS 的 ICF 知識呈現方式去呈現案例的知識與其案例知識的記憶組構，但不同於 PRECEDENTS，WEBPAD 還結合案例知識內有關 CAD 繪圖。因此，在 WEBPAD 系統中，ICF 不僅提供故事元素、故事與設計案例之間的連結，同時也連結視覺表現的設計案例資訊(例如：CAD 繪圖)。為了能有效連結上述不同的資訊內容，WEBPAD 的案例基礎包含二個不同的層次結構，上層結構為以文字為主的概念網路案例基礎，下層結構以圖像視覺為主的案例基礎，每一個案例的視覺內容可以經由在文字層索引而連結。

WEBPAD 的主要想法是希望每個設計者或者設計辦公室能發展他自己的案例庫。這個案例庫可以在區域 (local) 或者透過網際網路與其他 WEBPAD 使用者共享。以整個建築設計定義案例容易造成設計案例本身的複雜度，另一方面，當視建築組成元素 (例如：房間) 為案例時，此方式會導致此元素與其在整體建築關係之解釋的困難度。因此，WEBPAD 的設計案例包括整個建築設計與其個別組成元素的相關描述。

案例呈現

架構在 PRECEDENTS 所發展之 ICF 知識呈現概要，WEBPAD 結合 CAD 的繪圖以呈現設計案例知識。WEBPAD 已經發展一個包括三個不同層的案例呈現之多層 (multi-layered) 模型，包括 *工程層 (project layer)* 提供整體呈現並且形成與工程有關資料的內聚 (cohesive) 框架結構；*繪圖層 (drawing layer)* 包含一個整體建築設計的 CAD 繪圖呈現；*元素層 (component layer)* 呈現在繪圖層中分開元素的圖像與文字資訊。

WEBPAD 提供不同設計者在分散式的設計環境中，透過網際網路技術進行分享與連結設計案例知識，因此，WEBPAD 除了連結單一知識實體的案例知識，也連結其他多重知識實體的設計知識；案例知識可以根據不同使用者做動態性的增加，而 Java applet 可以提供使用者可以在不同的 CAD 軟體進行案例回憶。雖然，WEBPAD 提供參與者進行分散式的想法連結，但機器 (或電腦) 只是一個連結分

散知識的溝通平台，而無法自動性的參與分散式想法連結。

3.2 輔助回憶

分散式想法連結中的回憶主要強調記憶組織與案例回憶的機制。記憶組織為探討 CBR 系統的早期設計案例如何架構在知識呈現的基礎上，組織設計案例以儲存在案例庫中；而案例回憶為了解如何透過不同搜尋過程（包括如索引、檢索與選擇）的推理機制，有效的連結到案例庫的設計案例知識以產生設計想法。因此，在輔助回憶之文獻回顧，除了延續上一章節的 CBR 系統案例外，另外，一個強調想法連結機制的 DIM[27]初期研究模型，其主要結合想法聯想三個原則與 CBR 運算機制，也在本章節探討的範圍。

3.2.1 Archie-II

記憶組織

Archie-II 的案例記憶組織主要包含故事和準則 (guidelines) 二種類型的知識區塊，並提供使用者能迅速且容易使用自己的方法，搜尋到適當的案例。在 Archie-II 的故事，對於一個公共建築設計的使用評估，應強調主要優點和缺點部分的特徵，這些特徵都值得儲存以供將來使用。藉由以簡短的文字描述這些特徵有趣部分，Archie-II 將案例分解成三種類型的故事，包括：點故事 (point stories)、互動故事 (interaction stories) 和簇群故事 (cluster stories)。

除故事之外，Archie-II 包含第二種稱之為準則的知識區塊。基本上，準則是透過綜合許多特別的經驗，而提出一種思考的設計方式，但不提供解決設計問題的絕對答案。Archie-II 的版本描述包含許多組故事和相似數量的準則，這些資訊被連結在一個網路，包括例子和反例 (counter-examples)、普遍化和特別化。

案例回憶

回憶的方式允許使用者適時性的搜尋到正確的案例，或適時性的搜尋到正確案例中的適當知識區塊。因此，Archie-II 提供兩種索引方式：描述性索引 (*Descriptive indices*) 與關聯性索引 (*Relationship indices*)。描述性索引係透過一個設計議題與下列四個特徵之一 (或多個)，其所共同定義的故事進行設計案例的檢索，這四個特徵包括建物的空間、功能的組成、保管者 (stakeholder) 的遠景和建物不同階段的生命週期。另一種索引方式為關聯性索引，其功能類似於直接嵌入作者的文字忠告 (pointers)，予以引導使用者對於相關訊息的注意力。

簡言之，描述性索引支持使用者能直接搜尋設計案例，允許設計者在同一時間內，

連結並搜尋到一個設計案例具有不同面向的資訊；關聯性索引允許使用者瀏覽有關的故事和準則，使 Archie-II 能夠跟隨一位設計者，經由在不同設計案例之間的動態性移動 (juggling)，建立設計想法之間的連結。另外，在 Archie-II 中，雙方向的連結 (bi-directional) 可以被建立在上述三種不同類型的故事之間。

因此，Archie-II 的案例回憶企圖反映建築師的設計思考方式，也就是設計者不會將設計分開考慮，而是強調不同議題之間的關聯性。然而，複雜的索引系統對於案例的管理 (增加、移除、修正等) 有相當的困難度，同時，知識獲得主要以公共建築類型為主，對於其他建築類型的設計案例缺乏開放性，這些問題導致案例回憶在連結設計知識的侷限性。

3.2.2 PRECEDENTS

記憶組織

在 PRECEDENTS 中，案例記憶主要經由不同的設計故事予以組織。架構在 ICF 概要之知識呈現的三個元素 (議題、概念和形式)，此記憶組織是一個相關於設計領域高層次 (high layer) 故事的關聯性網路，經由這三樣元素相互連結，不僅連結自己案例的故事 (圖 18)，同時也跨文本的連結 (cross-contextual linkage) 到其他案例的故事。因此，整個記憶組織是一個具有語意上連結的知識網路結構，提供 PRECEDENTS 之回憶的基礎。取代於回憶完整的設計案例，PRECEDENTS 透過相關領域概念語彙的議題和概念符號，回憶相關的設計故事。

透過高層次回憶領域概念語彙，可以支持使用者直接的進行搜尋並且瀏覽。議題和概念的領域語彙是 PRECEDENTS 案例記憶的詞彙 (lexicon)，因此，從此知識網路結構的任何節點 (包括議題或概念) 中，相關的故事可能被回憶，同時，也可以搜尋到這些故事連結的早期設計案例，提供使用者瀏覽到不同的早期設計案例。

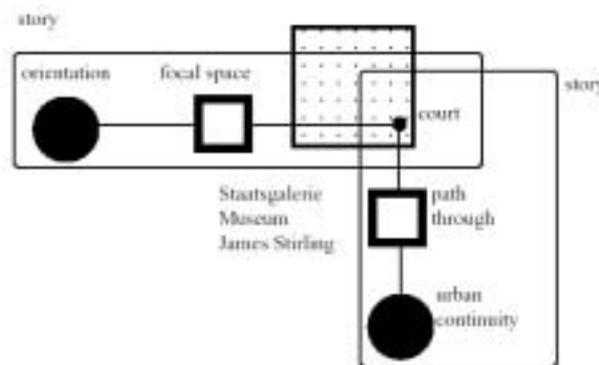


圖 18. 使用 ICF 概要組織 Stuttgart 美術館案例 (來源自 Oxman, 1994)

案例回憶

像Archie-II一樣，PRECEDENTS提供兩種回憶相關案例的方式：直接搜尋與瀏覽。第一種方式是使用者在系統的介面中，輸入指定ICF概要中議題的關鍵字後，系統利用這些議題的關鍵字作為在語義網路詞彙的入口，且搜尋相關的設計故事並對應出這些設計故事的早期設計案例。如果使用者為了檢查其他解決辦法的概念，或者希望在相同的早期設計案例探索不同的概念，使用者能使用故事的索引瀏覽案例庫。經由啟動在語意網路之相類似設計概念的連結（linkages），並利用一個關聯性（associative）的索引系統，提供使用者在多樣性的設計故事與早期設計案例，搜尋到其他解決相同設計議題的方法。

圖19說明Stuttgart美術館之設計故事中的詞彙（領域概念語彙），提供使用者連結到其他相關案例的設計故事。在此例子（圖19）中，當使用者指定“orientation”的議題之後，PRECEDENTS可以使用Staatsgalerie博物館案例的設計故事（orientation, focal space, court entry），藉由連結到古根漢美術館（Guggenheim museum）案例中設計故事的相同議題值（orientation），而搜尋到此設計故事中斜坡的形式。另外，使用者也可以使用Staatsgalerie博物館案例之設計故事的形式值（court entry），經由連結其它設計故事中的相同形式值，而搜尋到其他設計故事（但具有不同的議題與概念），如Le Corbusier設計的Visual Arts Center之設計故事（decorative emphasis, path through, exposition）。

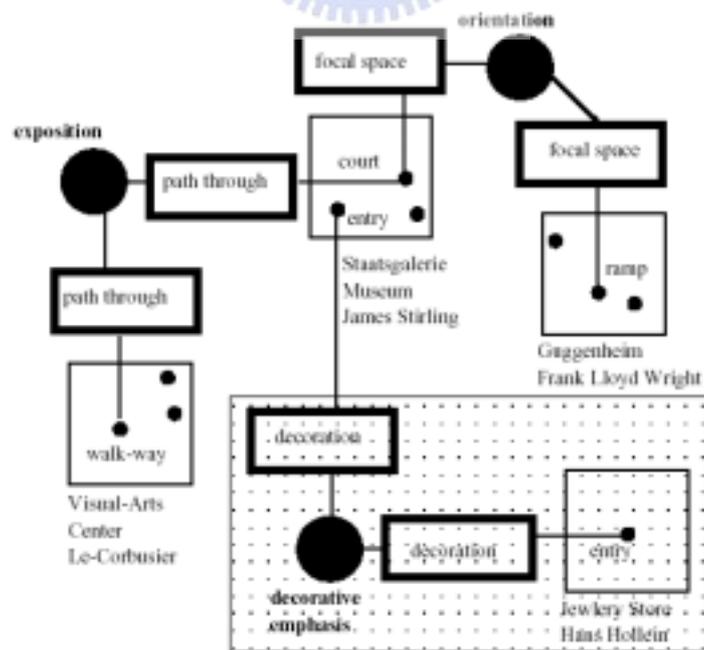


圖19. PRECEDENTS 之動態連結案例的機制（來源自 Oxman, 1994）

另外，PRECEDENTS也提供使用者經由連結到其他類型(非博物館類型)的建築，而分享到其他建築類型案例的設計故事。例如：使用者可以透過Staatsgalerie博物館的概念(court entry)，而連結到建築師Hans Hollein設計的珠寶商店之設計故事。在瀏覽過程中，使用者啟動Staatsgalerie案例與其他設計案例故事之間的連結，系統自動地帶出相對應圖解的實例，這些實例可能導致使用者發現新的且未預期的設計概念。

PRECEDENTS 系統的記憶組織，主要由不同設計故事透過 ICF 概要的知識呈現元素，將設計故事相互連結而形成。在回憶的過程中，可以經由連結到相似的設計故事，而搜尋到相關的案例知識，此回憶機制有助於知識實體在各自的記憶組織中，連結到多樣性的設計想法。由於 PRECEDENTS 系統對於設計故事之間的連結，沒有提供標籤以給予連結上的意義，因此，案例知識間缺乏有連結意義的關係建立，且在記憶回憶的過程中，使用者也無法利用連結意義(標籤)的特性，而失去設計案例在搜尋過程中的動態性與關聯性。

3.2.3 FABEL

記憶組織

FABEL的專家不僅在解釋案例上有所不同，同時，他們也使用不同的方法組織案例庫，其方法包括線形(linear)、階層形(hierarchical)、聯結形(associative)等。例如：在階層形的方法中，點陣圖(bitmap)被安排在一個階層形的記憶組織，在此階層形記憶組織中，階層最低反應最粗糙的解決方法；圖像(graph)被區劃成許多次圖像(sub-graph)；另外，terms可以被動態的組織成目錄、階層或被區劃成許多sub-term。

案例回憶

在FABEL內，檢索的目的是連結到相似於輸入需求(如設計問題)的設計案例。由於兩個設計可能在不同的地方是相似的，FABEL提供多重檢索的方式，也就是由單獨的專家檢索，或多重的專家檢索，同時，此相似的特質會隨著設計情境、用戶意圖，或個人喜好而產生差異。因此，每位檢索專家需比較設計之間一個或組合的特徵，且查詢首先被動態性的解釋並且改變成專家喜好的呈現，而後根據不同的案例呈現，在相同案例呈現的案例庫進行模式比對(pattern matching)。

為了此模式比對，每位專家都有一種適當的比較方式。例如：當使用特徵向量進行比較時，一種距離基礎(distance-based)的搜索演算法，把輸入查詢的片段與一個案例的目錄相互比較後，輸出最相似的案例排序目錄(sorted list)，而使用者可以指定檢索案例的數量，或相似度的距離值以選擇合適案例的範圍；當使用圖像

進行比較時，一個圖像演算法在圖像的案例庫進行輸入圖像的比對後，最後，輸出一個有結構性的相似圖像。

原則上，所有的檢索方式可以根據使用者的需求，隨意的被組合、連續或並聯，因此，FABEL 以“雇用”一個或多個之前描述的檢索專家，而提供四組固定的檢索團隊。第一組檢索團隊結合了特徵向量與預先指定的型態，並藉由類型和包含物體的數量進行案例比較，此團隊稱之為 ASM。第二組團隊稱之為 FAV，在 FAV 團隊中，這些型態與特徵向量彼此結合在一起，以便藉由規模、物體型態、數量和類型進行案例的比較。在第三組稱之為 ODM 的團隊中，ODM 只有一位雇用的專家，而此專家經由物體的密度地圖進行案例的檢索。最後，則是 ASPECT 團隊，此團隊允許結合並測量相似性的特徵，以便定義每一個檢索案例的相似性測量。

FABEL 支持設計案例不同的解譯與呈現方式，並予以建構不同的記憶組織，提供系統設計案例庫的豐富性。另外，不同檢索專家的建立與組織，並經由不同搜尋機制或彼此之間的搭配，可以輔助設計者在不同的設計情境，動態性的搜尋到多樣性的案例知識。然而，對於不同專家知識的整合，與其衍生龐大案例庫的管理仍有它的限制性。

3.2.4 DIM (初期研究模型)

DIM 的初期研究模型 (DIM 初期模型) 是 Lai[27]建立的 CBR 架構，其用來輔助設計者在一個互動性的集體腦力激盪會議中，藉由想法聯想行為以連結並產生多樣性的設計想法。DIM 初期模型主要結合想法聯想的三個原則 (相似性、對比性與相鄰性)，進行早期設計案例的回憶、調整或組合，並動態性產生如地圖般的知識網路結構，而這種結合想法聯想行為所建構的知識網路結構稱為動態想法地圖。

記憶組織

DIM 初期模型主要強調單一設計者在早期設計階段，產生有關於空間組構的設計想法。因此，在 DIM 初期模型中的案例知識，強調設計案例中有關設計概念知識的呈現，主要包括相關於空間組構上的設計問題，與其概念性的解決方法。其案例知識主要藉由領域概念語彙 (或稱關鍵字) 表現，並結合了多媒體，包括如影像照片、設計草圖或動畫等所構成。

DIM 初期模型的記憶組織是如地圖般的知識網路結構，主要由不同的想法元件，與想法元件之間的連結所構成。每一個想法元件包含了設計問題與設計問題的解決方法，而且想法元件之間主要有三種不同的連結關係 (相似關係、對比關係與相鄰關係)。另外，為了支持設計案例之間不同但類似的領域概念語言的共通性，DIM 初期模型包含一個詞典 (dictionary)，負責解譯不同但類似的領域概念語言，

以提供使用者能動態且彈性的連結到相關的設計案例的知識(圖 20) 這些記憶組織與元素(包括想法元件與連結), 被建構在框架(frame)和欄位(slot)的案例資料庫中。

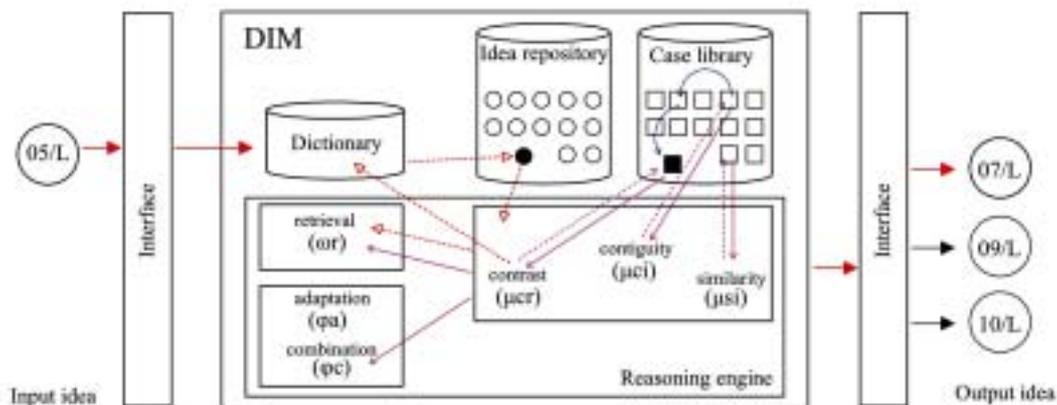


圖 20. DIM 初期模型的案例基礎推理架構

案例回憶

架構在上述的記憶組織方式, DIM 利用想法聯想的三個原則(相似原則、對比原則與相鄰原則), 建立回憶的限制條件與轉變方式, 提供設計者能動態性的搜尋與瀏覽, 並進而連結與產生多樣性設計想法。例如: 設計者使用相似原則與相鄰原則, 連結早期相似的想法與設計案例, 同時, 相似原則也允許此設計者繼續地連結本身的其它相似設計案例。另外, 設計者可以使用對比原則, 連結到早期對比的設計想法, 此對比的設計想法可以繼續的藉由相似原則, 去連結早期具對比性的設計案例。

簡言之, DIM 結合三種想法聯想的原則, 除了可以在記憶組織中, 賦予想法元件之間連結關係的意義外, 同時, 這些原則結合回憶機制, 可以提供設計者一個理解的方式去組織設計案例與想法的資訊, 同時, 在不改變整體共同的设计概念下, 參與者可以動態性的連結並產生多樣性的設計想法, 以解決設計問題。

3.3 輔助溝通

在分散式的設計環境中, 溝通主要強調多重知識實體(包括人類或電腦)能自主性的進行訊息傳遞。在設計運算領域, 以代理人理論發展的多重代理人(multi-agent)系統, 係強調代理人之間溝通的數位技術, 提供本研究在了解輔助溝通的主要運算機制。因此, 有關輔助溝通系統案例的選擇, 為強調多重代理人之間的分散式設計合作, 對於代理人系統結合自動細胞(cellular automata)機制進

行多重代理人模擬 (multi-agent simulation, 簡稱 MAS), 將不在本研究系統回顧的範圍。

除了之前論述 WEBPED 使用網際網路技術連結不同設計者 (人類) 之間的设计知識外, 多重代理人系統則是強調經由軟體代理人 (電腦) 的分散性特質 (自主性、反射性與溝通性), 提供電腦與參與者在分散式的設計環境中, 自動化的連結設計知識。在下面系統文獻回顧中, ADLIB 和 MADS 系統是透過多重代理人系統, 經由電腦彼此之間的合作而解決設計問題, 而 CBR-TEAM 系統則結合多重代理人與案例基礎推理技術, 使用分散式的搜尋機制進行設計合作。另外, DARIS 啟發於角色扮演理論, 經由分散式互動的實行, 提供多重知識實體可以在內外不同設計情境中進行互動性的溝通。

3.3.1 ADLIB

ADLIB系統是用來解決有關鋼結構的建築設計問題[31]。基本上, 此系統將有關鋼結構設計的涵構(context), 定義為代理人建構機制與代理人之間協商策略的依據, 主要將大量的建築桿件結構拆解成單一構件之代理人, 並透過代理人之間協商與合作以解決鋼結構的設計問題。

在ADLIB系統中, 主要由兩種不同類型的代理人組成: 介面代理人 (interface agent) 與專家代理人 (specialist agents)。介面代理人主要使用者與代理人間的互動關係, 專家代理人則由有關不同領域鋼結構知識的代理人組成, 其包括結構設計的代理人 (SDA) 安全建議代理人 (SAA) 等, 這些不同類型的代理人建立在一個階層性 (hierarchical) 的架構中, 在不同階層之間與其階層內的代理人, 均有不同的溝通模式 (圖20)。

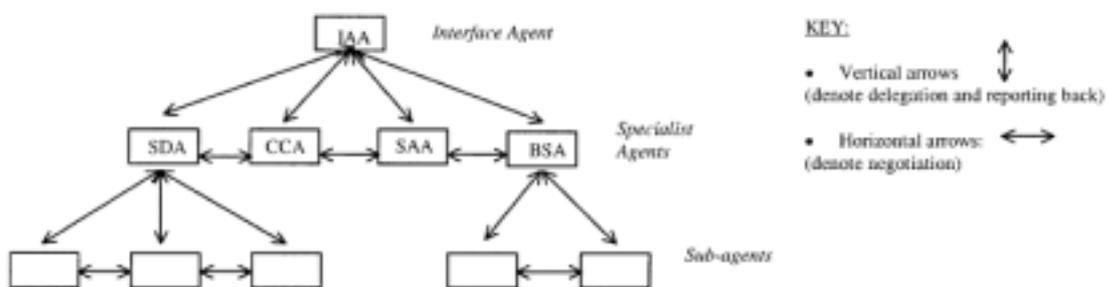


圖 20. ADLIB 系統架構圖 (來源自 Anumba et al., 2002)

為了讓上述不同領域知識代理人能有效的溝通, 在 ADLIB 系統中, 這些代理人是具有相同的代理人之知識實體論 (ontology), 以利代理人在合作過程中知識的瞭解與分享, 因此, 在解決鋼結構的設計問題中, 鋼結構的基本元素 (包括樑、柱、

椽和連系材等)與鋼結構長寬高的參數值為知識呈現最基本的元素。此系統為達到跨平台(interoperability)的目的,因此,ADLIB 系統主要以 Java 為根據的 TCP/IP 傳遞為主,其他訊息傳遞協定(protocol)還包括 HTTP、KQML 和 KIF,這些訊息傳遞協定方式主要根據在 ZEUS 代理人環境中的不同功能層而定。

ADLIB 系統提供不同代理人如何進行合作與溝通等社會性的問題,主要藉由程式知識實體論(agent ontology)之建構機制,連結不同領域代理人的分散知識,讓這些擁有不同設計資訊或知識的代理人,能自動處理部分設計問題的能力。在合作過程中,由於設計決策由代理人自行決定,因此此系統提供很大的彈性去反應不同的設計情境;同時,ADLIB 可由 World Wide Web 直接使用此資訊,提供使用者一個容易且便宜的分散式合作設計的環境。

3.3.2 MADS

MADS (Multi-Agent Design System)系統主要是由 Greco and Brown[153]所建構的多重代理人系統。此系統主要強調設計代理人在設計合作過程中有關設計學習(design learning)的議題。其主要的想法是經由代理的學習期望(learning expectations)去進行彈性的代理人學習以提昇設計的效率。

學習期望是提供多重代理人在動態的設計情境中,能適當地反應不同發生事件的學習方式,主要的機制是建立在代理人設計決策與結果評估的關係上。在 MADS 系統架構中,代理人之間沒有固定階層性的關係,它們之間的互動關係主要建立在相同層級中,預先定義好的代理人行為規則上,透過學習期望的機制去動態性連結不同類型的代理人,去反應外在設計環境的事件。

此系統主要由兩個主要的系統層與四種不同類型的代理人所構成。其系統層包括設計層(design layer)與互動層(interactive layer),而四種不同類型的代理人為設計者(Designers)、批評者(Critics)、讚美者(Praisers)與評估者(Estimators)。因此在設計層,主要為這些代理人處理個別設計領域知識的問題,在互動層,為這些代理人在處理設計合作(cooperation)、衝突(conflict)和溝通(communication)的互動性問題。在 MADS 中,多重代理人使用 KQML 的代理人溝通語言進行設計合作,系統執行在 CLIPS 的規則基礎(rule-based)環境中,其機器學習的元素主要根據 MLC++機器學習庫的包裹(wrapper)技術與歸納(inductive)學習。

MADS 系統將系統分為二個系同層去分別處理代理人的設計領域知識與互動式關係,可以提供分散式知識實體間較為動態的連結關係,因此在系統學習上也提供不少彈性,包括如同時具有獨立性(independently)與一致性(concurrently)的學習、學習過程會根據設計情境而改變等。然而,有關期望數量與其之間的衝突仍有它的侷限性。

3.3.3 CBR-TEAM

CBR-TEAM 是結合案例基礎推理技術的多重代理人系統，此系統是由 Massachusetts-Amherst 大學所建立[154]。基本上，此系統是由原先的 TEAM[155] 系統改良而成用來處理有關蒸氣冷凝器 (steam condenser) 的設計。

此系統主要將蒸氣冷凝器設計分解成若干元素，分別由不同的代理人負責設計。在設計的過程中，這些具有個別案例基礎推理 (CBR) 的代理人藉由擷取與使用案例庫中得設計元件去進行設計，當遇到設計衝突時，這些代理人使用談判式擷取策略 (negotiated retrieval strategy) 去解決衝突問題。此談判式的擷取策略主要是藉由擷取案例的子案例 (subcase) 與不同的反射性行為 (如啟動一個關鍵的子案例、結合既有的部分案例等)，而達到不同代理人之間的同意，而組裝蒸氣冷凝器個別設計的元素。

基本上，CBR-TEAM 系統是一種分散式的案例基礎推理系統，此系統將設計工作予以分解，並由不同代理人分別執行，除了解決 CBR 資訊超載的系統執行問題，同時此系統也連結兩種不同的設計情境：內部的個別知識與外部的分散知識。另外，由於每一個代理人具有部份或不完整的設計知識，因此設計過程中發生的衝突而造成連結中斷，是分散式連結機制必須加以考慮的。CBR-TEAM 系統的談判式檢索策略藉由案例知識呈現的方式與其代理人反射性的行為機制，提供解決設計衝突的可能方法。

3.3.4 DARIS

DARIS (Dynamic Agent-based Role Interplay System) 是由 Chang 和 Lai[18]所建構的代理人系統，此系統主要結合 Yardley-Matwiejczuk[119]的角色扮演理論架構與代理人的運算機制，用來執行分散式互動設計行為，包括如合作、腦力激盪與學習。基本上，DARIS 主要由三個不同層級所構成，包括內部設計情境層 (internal design situation layer)、溝通層 (communication layer)、外部設計情境層 (external design situation layer)。每一個層有它自己的機制並互動於其他層 (圖 22)。

在內部情境層中，由一群有自己的技能與知識，並能反應設計情境的角色代理人所構成，其中有部分的角色代理人具有合作的知識；在外部情境層中，包含場景代理人與舞台代理人主要描述角色代理人何時與何者來參與這個表演。另外，提供相互作用、知識與學習案例的儲存；另外，溝通層則提供不同代理人之間的溝通環境，同時，此層存在一個代理人專門負責在代理人平台中訊息傳遞的任務。

另外，DARIS 主要由五種不同的代理人構成，包括使用者代理人(UA)、角色代理

人(RA)、導演代理人(DA)、舞台代理人(StA)與場景代理人(ScA)。每一種代理人都有自己的機制與任務，並互動於相同與不同類的代理人。例如：RA 為具有知識基礎並進行設計推理任務、ScA 主要控制角色名單、時間限制與檢驗結果任務；而 UA 和 DA 主要負責使用者互動於 DARIS 環境的介面（圖 22）。

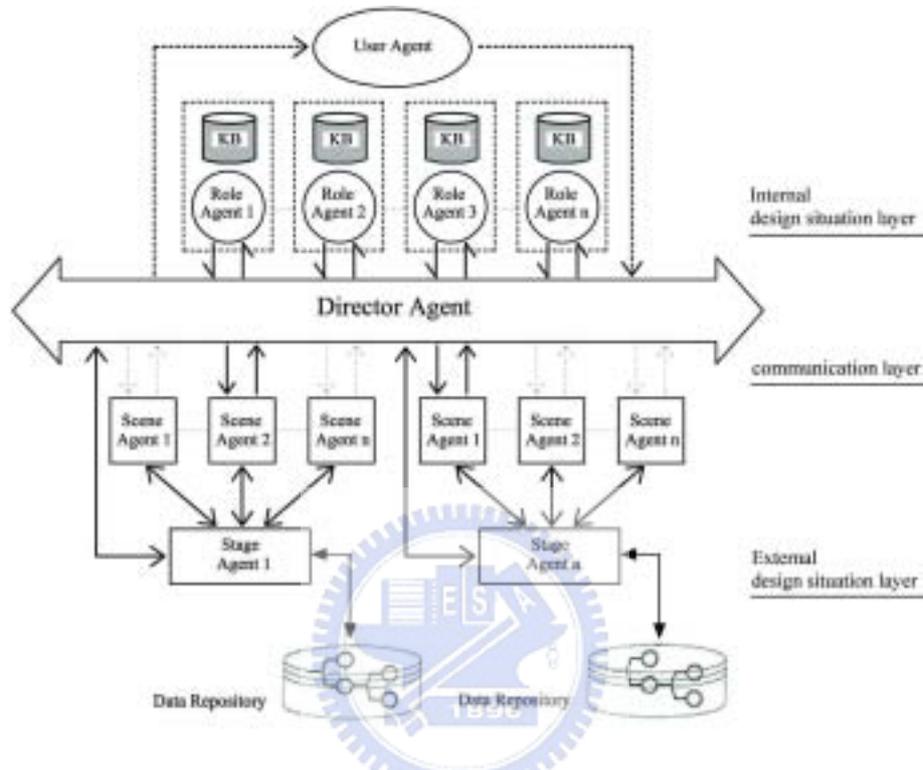


圖 22. DARIS 系統架構圖（來源自 Chang 和 Lai, 2004）

為了去執行上述的概念模型，DARIS 包括了三個主要的系統元素 1) 代理人的建築架構；2) 代理人的溝通語言，與 3) 溝通的協定模式。在系統的建築架構中，主要以 FIPA（Foundation for Intelligent Physical Agents）為代理人的抽象建築架構。在代理人溝通語言中，則主要為四個元素所構成，包括溝通語言的形式、內容、語意與實作。而其中形式主要以線性流動的形式（linear stream of characters）作為不同層之間的內部溝通。在內容部分根據主要設計問題加以分層，除了將知識實體論加以界定外，也提供其可變性。

DARIS 的代理人互動環境主要架構於 JADE 的代理人平台（Java Agent Development Framework）[156]，且 FIPA 為主要的代理人溝通語言。在 DARIS 的代理人架構中，每一個使用者（UA 與 DA）個自有一個 JADE 代理人平台（platform）控制其它代理人的互動關係；在每一個代理人平台中，這些代理人被啟動在 JADE 及時（run-time）環境中的容器（container）。另外，使用者之間能使用 HTTP 的訊息傳輸協定，在不同的地理環境位置進行相關訊息的交換。

3.4 邁向一個輔助分散式想法連結模型

綜合以上輔助系統的文獻回顧，這些輔助系統在輔助呈現、回憶與溝通的運算機制詳見表 1。在輔助呈現的系統案例中，這些 CBR 系統都包含具體的建築設計案例，這些案例被呈現在不同規模與類型的案例知識，從整體建築物設計(Archie-II、PRECEDENTS、WEBPAD)、強調建築空間組織(PRECEDENTS、DIM 初期模型)或設計組成元素(FABEL)。為了案例知識管理性的考慮，設計案例都分解成許多知識區塊，例如：故事(Archie-II、PRECEDENTS)、想法元件(DIM 初期模型)或設計組成元素(WEBPAD、FABEL)。另外，這些 CBR 系統都使用關鍵字伴隨著多媒體資訊以具體表現案例知識。

在輔助回憶的系統案例中，這些 CBR 系統提供不同的方式組織長期記憶的案例知識，例如：使用索引方式並結合不同的記憶結構組織案例記憶(Archie-II FABEL)，和利用 ICF 概要建立案例知識的基本構成元素，並經由這些構成元素彼此之間的連結，連結不同設計案例以組織案例記憶(PRECEDENTS、WEBPAD、DIM 初期模型)。這些 CBR 系統除了使用比對條件和運算元的規則建立，進行設計案例的搜尋外(Archie-II、PRECEDENTS、WEBPAD)，此外，結合不同的回憶方式(如 FABEL 的專家團隊，或 DIM 初期模型的連結原則)，可以增加使用者在搜尋案例的過程中，動態的連結到不同關聯性的設計案例，有助於不同設計問題的解決。雖然，FABEL 可以提供對於設計案例不同的解譯方式，以避免設計案例知識獲得的主觀性，但不同專家間的多重知識之關聯性建立有其限制性，而在 DIM 初期模型中詞典提供的解譯功能足以削減 FABEL 的限制性。

在輔助溝通的系統案例中，這些不同多重代理人環境，都將設計工作分解成許多工作單元，並給予不同代理人，然後，根據設計工作的流程與工作單元的從屬性，將這些代理人安排在階層性(ADLIB、MADS、CBR-TEAM、DARIS)或層級性(DARIS)的組織。此外，這些代理人之間的互動過程，經由 KQML(ADLIB、MADS、CBR-TEAM)或 FIPA(DARIS)代理人溝通語言而予以控制。雖然，這些多重代理人環境都利用 HTTP、IIOP 等通訊協定方式，提供代理人在區域內或跨平台的溝通，但由於 FIPA 的 performatives 標準化的特性，因此，比較於使用 KQML 的多重代理人環境，DARIS 提供較優越的跨平台開放度與相容性。

從上述的輔助設計系統，作者不但學習到有關輔助呈現、回憶與溝通的運算機制，而且這些運算機制提供作者建立一個輔助分散式想法連結之運算模型的基礎。有關作者提出之運算模型的呈現機制，主要強調使用早期設計案例解決有關空間結構的設計問題，為了案例知識的管理，這些早期設計案例必須被分解成許多知識區塊(即設計想法)，並透過結合多媒體資訊以增進使用者之間的溝通。

在作者提出之運算模型的記憶機制中，記憶組織必須強調知識區塊之間的關聯性，該關聯性藉由 ICF 概要而實踐，同時，利用 ICF 概要建構此模型在記憶組織的運算機制。此外，ICF 概要結合 DIM 初期模型的三種連結原則（相似原則、對比原則與相鄰原則），提供此模型進行動態性的模式比對，以連結多樣性且具關聯性的設計想法。而為了連結不同知識實體所產生的設計想法，此模型也須提供具有解譯不同關鍵字（領域概念語彙）功能的詞典。

為了輔助多重參與者在分散式的設計環境中連結設計想法，在作者提出之運算模型的溝通機制下，此模型必須具有跨平台與相容性的優勢，而 DARIS 具有此模型跨平台與相容性特性。除了前述的特性外，DARIS 的特性也有使用角色扮演理論，並將此理論結合階級性與層級性的代理人組織方式，這些特性將提供本研究了解分散性想法連結機制中之相互作用，與其連結問題上的衝突性。因此，DARIS 的多重代理人架構有助於作者建立此模型在溝通上的運算機制。



	呈現		回憶		溝通	
	呈現	記憶	規則	決策	規則	決策
Archie-	內容：公共建築的設計評估知識 呈現：三種故事與準則 媒材：結合圖像與註解文字	組織：描述性與關聯性索引 回想：直接搜尋與瀏覽	一組比對條件和一系列的運算元	使用者控制	一組比對條件和一系列的運算元	使用者控制
PRECEDENTS	內容：博物館建築的概念知識 呈現：故事 媒材：結合圖像與註解文字	組織：ICF 概要 回想：直接搜尋與瀏覽	一組比對條件和一系列的運算元	使用者控制	一組比對條件和一系列的運算元	使用者控制
FABEL	內容：公共建築的設備工程知識 呈現：CAD 圖畫組成元素（包括七種解譯） 媒材：結合圖像與註解文字	組織：關聯、線型和階層式索引 回想：四種專家團隊（ASM、FAV、ODM 和 ASPECT）	多組比對條件和一系列的運算元	比對條件與使用者控制	多組比對條件和一系列的運算元	比對條件與使用者控制
WEBPAD	內容：博物館建築的概念知識 呈現：建築物與三個組成階層 媒材：結合圖像與註解文字	組織：ICF 概要 回想：Java applet 並結合網際網路	一組比對條件和一系列的運算元	使用者控制	一組比對條件和一系列的運算元	使用者控制
DIM (初期研究模型)	內容：建築設計空間組織的概念知識 呈現：想法元件與連結 媒材：多媒體	組織：問題、概念與實例 回想：三種原則搜尋（分開或組合）	三組比對條件（相似性、對比性與相鄰性）和一系列的運算元	使用者控制	三組比對條件（相似性、對比性與相鄰性）和一系列的運算元	使用者控制
ADLIB	不在文獻回顧範圍	不在文獻回顧範圍	KQML 之 ACL 提供的比對條件和一系列的運算元	使用階層性代理人組織控制	KQML 之 ACL 提供的比對條件和一系列的運算元	使用階層性代理人組織控制
MADS	不在文獻回顧範圍	不在文獻回顧範圍	KQML 之 ACL 提供的比對條件和一系列的運算元	使用階層性代理人組織控制	KQML 之 ACL 提供的比對條件和一系列的運算元	使用階層性代理人組織控制
CBR-TEAM	不在文獻回顧範圍	不在文獻回顧範圍	KQML 之 ACL 提供的比對條件和一系列的運算元	使用階層性代理人組織控制	KQML 之 ACL 提供的比對條件和一系列的運算元	使用階層性代理人組織控制
DARIS	不在文獻回顧範圍	不在文獻回顧範圍	FIPA 之 ACL 提供的比對條件和一系列的運算元	使用階層性代理人組織控制	FIPA 之 ACL 提供的比對條件和一系列的運算元	使用階層性結合層級性代理人組織控制（結合角色扮演理論）

表一. 輔助分散式想法連結之運算機制總表

四、DIM – 動態想法地圖

動態想法地圖 (Dynamic Idea Maps, 簡稱 DIM) 是一個輔助分散式想法連結的運算模型。所謂分散式想法連結是強調多重參與者在分散式的設計環境中, 可以不受地理位置與時間條件的限制下彼此共同合作, 並有效的連結與產生多樣性以及關聯性的設計想法, 例如: 在設計競圖的過程中, 設計團隊的參與者可以在不同的工作地點相互激盪想法, 經由這些參與者產生想法之間的關聯性建立, 而提供他們發展競圖之替選方案的參考依據。基本上, DIM 是精練並強化 DIM 初期模型 [27] (詳見於 3.2.4) 在想法聯想中連結問題之分散性特質, 因此, DIM 除了延續其初期模型的 CBR 運算機制外, 並結合多重代理人的分散性運算機制 (包括自主性、反射性與溝通性), 用以建構分散式想法連結的連結機制與連結有效性, 進而在分散式的設計環境中, 輔助多重參與者產生多樣性且關聯性的設計想法。

啟發於 ARM[12]的相互作用理論, DIM 主要包括五個互動性元素 (*components*), 包括角色、演員、舞台、場景與腳本, 這些元素可以依據不同的設計情境進行分散式的想法連結。而為了支持這些元素能彼此共同合作而完成最後的設計結果, DIM 還包含知識 (*knowledge*)、原則 (*principles*) 與連結過程 (*linking process*) 的設計知識, 以提供這些元素進行設計合作的相關行為與機制 (圖 23)。

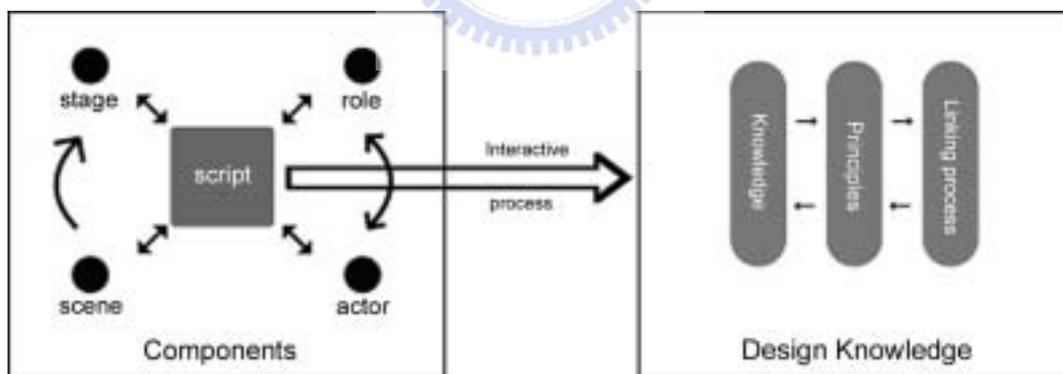


圖 23. DIM 模型架構

在這些設計知識中, 知識提供 DIM 元素關聯性的知識呈現與記憶組織方式, 原則支持這些元素在不同的設計情境, 能經由反射性的回憶, 而連結到記憶並產生設計想法, 而連結過程則允許這些元素經由彼此之間的溝通, 而建立這些元素在不同相互作用中之衝突性與關聯性的關係, 進而, 這些元素會因為彼此之間的衝突而相互競爭, 進而演化它們本身內在的設計知識。此外, 在連結過程中產生的設計結果, 皆可透過立即性的視覺回饋而表現。

由於，本研究是探討具有建築專業知識的多重參與者，進行早期設計階段的分散式想法連結，因此，DIM 模型架構在下列的條件前提下：1) 這些元素具有相同的知識本體論 (ontology)，同時，2) 使用相類似的建築領域概念語彙描述相關想法的設計知識。

4.1 元素

DIM 主要組成元素是角色、演員、舞台、場景與腳本。基本上，DIM 的主要關係建立在演員和角色之間，而角色和演員具有多對多的關係，換言之，一名演員能扮演很多角色，且一個角色也可以由不同的演員所扮演，且在場景的持續期間內，這些角色和演員經由彼此之間的相互作用而產生設計結果。在設計領域中，此現象反映著設計者會透過扮演不同的角色而產生適當的設計想法，以回應不同參與者產生的設計想法[104]，而相互作用會因為設計者之間在溝通過程的差異而不同，進而影響最後的設計結果。

除此之外，在 DIM 中，設計知識間的從屬性 (dependency) 可以透過一群合作的角色而指定，或藉由腳本的時間表特徵 (time-schedule-feature) 而定義。基於上面的描述，DIM 元素提供本研究建立分散式想法連結相關的行為機制 (如相互作用)，有關這些元素的詳細說明如下：

- 演員 (actor): 一位演員是一場戲劇的參與者。在 DIM 中，參與戲劇 (分散式想法連結) 的演員主要包括 *使用者* 和 *導演* 二種類型，使用者係指參與此分散式想法連結的設計者 (可以是人類或電腦)，而導演是主導整個分散式想法連結的協調者。在想法連結的過程中，這些演員根據此分散式想法連結的腳本內容，扮演不同的角色以連結與產生設計想法。
- 角色 (role): 角色代表在戲劇中進行表演動作的演員，是 DIM 主要的元素。在一個分散式的想法連結過程中，角色被視為演員為了執行某設計工作，被賦予連結與產生設計想法的行為。基本上，一位角色必須具備知識和原則 (詳見 4.2 與 4.3 章節)，知識建立角色的記憶與其組織方式，原則提供角色連結與產生想法的能力，且支持角色互動於不同設計情境的機制。

另外，*場記* (script holder) 可以被視為一種特殊類型的角色，主要負責每一個場景中角色之間的協調工作，包括安排角色演出的順序、演出時間的控制、儲存不同角色共同演出的設計結果。

- 舞台 (stage): 舞台代表一場戲劇表演的環境，通常由一個以上的場景所構成。在 DIM 中，舞台反映一個設計工作之主要目的，透過不同場景在設計問題、

時間、順序與其他條件限制的定義下而達成。此外，舞台負責貯藏所有場景的相關內容（包括場景的腳本內容與設計結果），與整個分散式想法連結的過程。

- 場景 (scene) : 場景是根據特定的時間表，表現程序性需求。在同一個時間中，只有一個場景能在舞台上表現，而根據一個特定設計問題與時間限制下，藉由不同角色與演員之間的互動，而完成此場景的程序性需求。同時，每一個場景負責儲存不同演員在此場景共同產生的設計結果。
- 腳本 (script) : 腳本為描述上述元素的內容敘述，主要包括舞台描述、角色定義、設計工作、行為動作與訊息內容等，其功能是描述這些元素本身，並定義這些元素彼此合作的準則。另外，DIM 元素之間溝通規則的描述也是一種特殊的腳本。

4.2 知識

在 DIM 中，知識主要來源自早期設計案例，並有關於建築設計空間組構的設計知識，而知識的呈現方式是如地圖般的知識網路結構，此知識網路結構稱之為**想法地圖 (idea maps)**。基本上，DIM 包含不同類型的想法地圖，主要由想法元件與連結所構成，這些想法地圖是建立不同 DIM 元素記憶組織的方式，且提供這些元素進行分散式想法連結之相關行為的設計知識。

表 2. 一個設計想法：James Stirling 設計的 Stuttgart 美術館案例

設計問題	都市連續性
抽象概念	開放空間、路徑穿越
實際製品	<p>中庭</p> 

4.2.1 想法元件

在 DIM 中，設計想法被稱為*想法元件* (*ie*)，其構成 DIM 知識的基本單元。一個想法元件主要由三個屬性所組成：設計問題、抽象概念，與實際製品 (artifacts)，例如：在德國 James Stirling 設計的 Stuttgart 美術館案例，建築師為解決基地「都市連續性」的設計問題，使用「開放空間」、「路徑穿越」的抽象概念去反應此設計問題，最後使用「中庭」的設計手法，予以具體化此抽象概念 (表 2)。

上述想法元件的三個屬性可以類比於 Oxman[66]的議題-概念-形式 (簡稱 ICF) 之知識呈現概要，*議題* (*i*) 可視為設計問題，*概念* (*c*) 可視為抽象概念，*形式* (*f*) 可視為實際製品，而為了反映設計者對於設計想法有不同的喜好程度，想法元件還包含*喜好* (*p*) 的屬性。因此，每個想法元件包含了四個屬性，分別為議題、概念、形式與喜好，其詳細定義如下：

- *議題* (*i*): 議題必須有關於設計者正在進行的設計工作，同時在一個特定條件下，對此設計問題上定義一個特殊的觀點。例如：在長條街屋的住宅設計中，狹長型基地且有兩長向緊鄰臨房的設計條件下，有關於「採光」與「通風」的設計問題。
- *概念* (*c*): 概念是關係於議題的解決方法，同時也是一個解決設計問題的抽象概念。例如：在上述的設計工作中，設計者使用「開口」的抽象概念，用以解決有關於通風或採光的設計議題。
- *形式* (*f*): 形式是一種特定設計的人為製品 (artifact)，它是將上述的概念予以具體化。因此，一個形式可視為藉由實際建築設計的元素，直接反映於解決一個設計議題的設計手法。例如：上述「開口」的抽象概念，可以透過建築師 Tadao Ando 的住宅設計案 Row House Sumiyoshi 的中庭、或建築師 Le Corbusier 的住宅設計案 Villa Savoye 的水平開窗等設計手法而表現。
- *喜好* (*p*): 喜好是角色根據其主觀意見，對於每一個想法元件的喜好程度。基本上，喜好會影響角色選擇設計想法的優先順序，喜好程度越高的想法元件，越容易被角色所使用，反之亦然。另外，當想法元件被選擇次數的頻率越高時，此想法元件的喜好值也會隨著增加。

喜好屬性之值是一個整數值，其整數值越高則代表角色越喜好此想法元件，同時，它被此角色選擇的機會也越大，反之亦然。而其他三個屬性之值主要由領域概念語彙所組成，議題屬性之值由一個領域概念語彙所組成，概念屬性之值由一個以上的領域概念語彙所組成，而形式屬性之值則由一個領域概念語彙和多媒體資訊

(包括如影像照片、動畫或 URL 等) 所組成。這四個屬性值可以詳見表 3。

表 3. 想法元件的四個屬性與屬性值

想法元件	議題 (i)	概念 (c)	形式 (f)	喜好 (p)
屬性值	一個 領域概念語彙	一個以上 領域概念語彙	一個 領域概念語彙與多 媒體	整數值

4.2.2 連結

連結 (—) 是想法地圖另一個重要元素。基本上, DIM 包含兩種類型的連結: 屬性連結 (\perp) 和元件連結 (\perp)。屬性連結是存在不同想法元件中屬性之間的連結, 元件連結是不同想法元件之間的連結。

4.2.2.1 屬性連結

屬性連結包括三種形式的連結, 其功能是建立不同想法元件中三個屬性之間的連結關係, 包括屬性相似連結 ($\overset{si}{\perp}$)、屬性對比連結 ($\overset{cr}{\perp}$) 與屬性相鄰連結 ($\overset{ci}{\perp}$)。根據想法元件的知識呈現方式, 這三種形式的連結具有下列的定義與其機制, 其說明如下:

- 屬性相似連結 ($\overset{si}{\perp}$): 當二個想法元件中的屬性, 其相同屬性的值具有相似性關係時, 則在這兩個想法元件中的相同屬性之間建立的連結, 稱之為屬性相似連結。例如: 當一個想法元件 (ie_a) 的議題值 (i_a), 與另一個想法元件 (ie_b) 的議題值 (i_b) 具有相似性關係時, 則 ie_a 中 i_a 與 ie_b 中 i_b 之間有屬性相似連結 ($\overset{si}{\perp}$)。
- 屬性對比連結 ($\overset{cr}{\perp}$): 當二個想法元件中的屬性, 其相同屬性的值具有對比性關係, 則在這兩個想法元件中的相同屬性之間建立的連結, 稱之為屬性對比連結。例如: 當某一個想法元件 (ie_a) 的概念值 (c_a), 與另一個想法元件 (ie_b) 的概念值 (c_b) 具有對比性關係時, 則 ie_a 中 c_a 與 ie_b 中 c_b 之間有屬性對比連結 ($\overset{cr}{\perp}$)。
- 屬性相鄰連結 ($\overset{ci}{\perp}$): 當二個想法元件中的屬性, 其相同屬性的值具有相鄰性關係時, 則在這兩個想法元件中的相同屬性之間建立的連結, 稱之為屬性相鄰連結。例如: 當某一個想法元件 (ie_a) 的議題值 (i_a), 與另一個想法元件 (ie_b) 的議題值 (i_b) 具有相鄰性關係時, 則 ie_a 中 i_a 與 ie_b 中 i_b 之間有屬性相

鄰連結 (ci)。

最後，藉由三種屬性(議題、概念與形式)之間所建立的不同形式之屬性連結(si 、 cr 、 ci)，這些屬性所對應的想法元件都可以被連結在一起，進而形成一個具有關聯性的知識網路結構 (圖 24)。

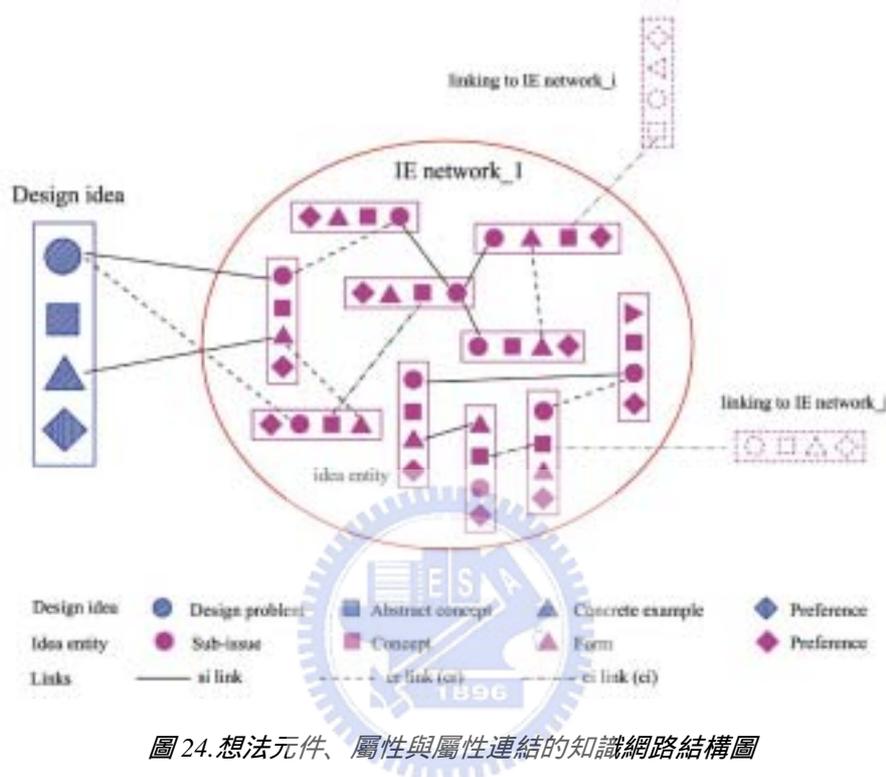


圖 24. 想法元件、屬性與屬性連結的知識網路結構圖

4.2.2.2 元件連結

想法元件之間的連結稱之為元件連結。元件連結主要包括元件相似連結 (si)、元件對比連結 (cr) 與元件相鄰連結 (ci)，而這三種不同形式的元件連結必須在兩個不同想法元件間，藉由兩種屬性之特定形式的屬性連結建立而形成。這三種元件連結的定義與其機制說明如下：

- 元件相似連結 (si)：元件相似連結為兩個想法元件間，具有相似性的解決問題方法。因此，當一個想法元件中的兩個特定屬性 (議題與概念，或議題與形式)，與另一個想法元件中相同的兩個特定屬性，彼此之間有屬性相似連結 (si) 時，這兩個想法元件有元件相似連結 (si) (圖 25)。
- 元件對比連結 (cr)：元件對比連結為兩個想法元件間，具有對比性的解決問題方法。因此，當一個想法元件中的兩個特定屬性 (議題與概念，或議題與形式)，與另一個想法元件中相同的兩個特定屬性，其中的議題有屬性相似

連結 (\underline{si}), 且概念或形式彼此之間有屬性對比連結 (\underline{cr}), 時, 這兩個想法元件有元件對比連結 (\underline{CR}) (圖 25)

- 元件相鄰連結 (\underline{CI}): 元件相鄰連結為兩個想法元件間, 具有相同的解決問題方法, 但此方法可以解決不同的設計問題。因此, 當一個想法元件中的兩個特定屬性 (概念與形式), 與另一個想法元件中相同的兩個特定屬性, 彼此之間有屬性相似連結 (\underline{si}) 時, 這兩個想法元件有元件相鄰連結 (\underline{CI}) (圖 25)

上述論述的想法元件、屬性、屬性連結與元件連結, 提供 DIM 建構不同類型的圖像式 (graph-like) 的知識網路結構, 其包含了節點 (想法元件或 ICF 屬性) 和邊 (屬性連結或元件連結), 而這些知識網路結構皆稱為想法地圖。這些不同類型的想法地圖, 提供 DIM 中不同元素的知識與其記憶組織, 有關這些想法地圖在下一章節詳細說明。

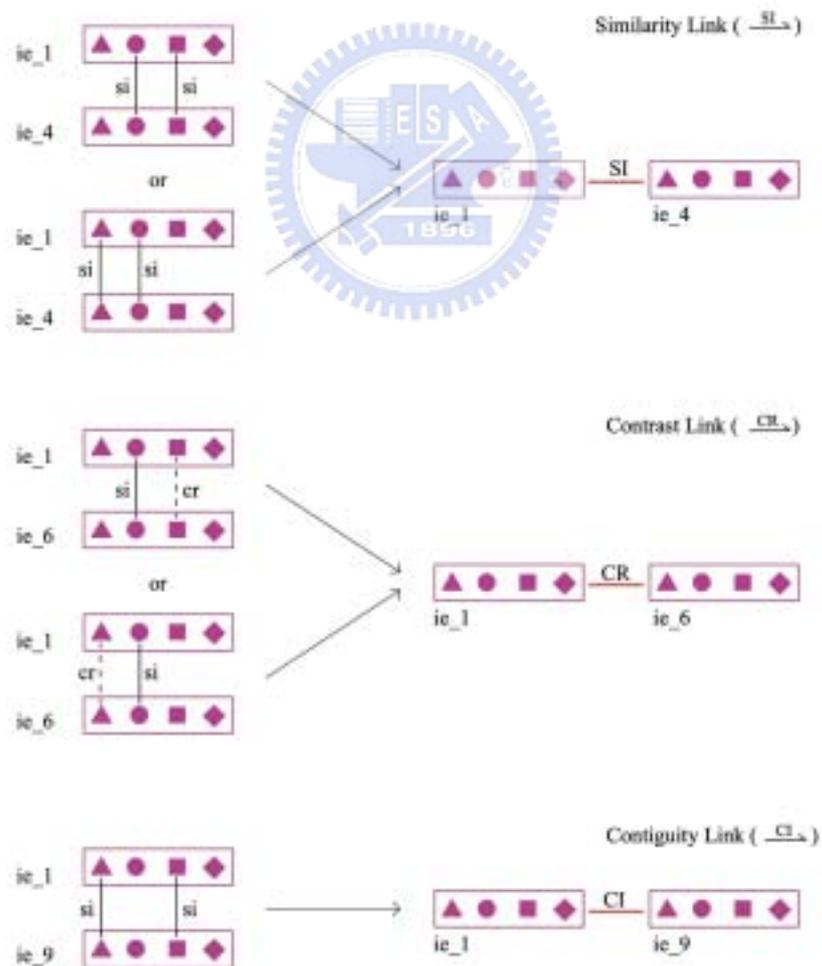


圖 25. 元件相似連結、元件對比連結與元件相鄰連結 (由上而下)

4.2.3 想法地圖

DIM 模型主要包括兩種不同型態的想法地圖：想法元件地圖（以下稱 IE 地圖， Map_{IE} ）與 ICF 地圖（ Map_{ICF} ）。IE 地圖主要由想法元件（ ie ）與元件連結所構成（包括 si 、 cr 和 ci ），而 ICF 地圖主要由想法元件中一個特定的屬性（包括 i 、 c 或 f ）與屬性連結（包括 si 、 cr 和 ci ）所構成。

IE 地圖的功能是提供角色組織記憶的方式，和其他 DIM 元素（包括舞台、場景和場記）儲存設計結果的知識呈現方式。為了區別 IE 地圖不同的功能，DIM 將 IE 地圖分為角色 IE 地圖（ Map_{IE_R} ）內部 IE 地圖（ Map_{IE_I} ）與外部 IE 地圖（ Map_{IE_E} ），角色 IE 地圖是角色的記憶組織，內部 IE 地圖是場記儲存設計結果的知識呈現方式，而外部 IE 地圖則是場景和舞台儲存設計結果的知識呈現方式。

除了 IE 地圖外，三種不同種類的 ICF 地圖：議題地圖（ Map_I ）概念地圖（ Map_C ）與形式地圖（ Map_F ），提供每一位角色在議題、概念與形式屬性值間，經由動態性的模式比對而增加角色在其角色 IE 地圖中，連結與產生想法元件的多樣性與開放性。這些不同種類想法地圖的功能與 DIM 元素之間的關係可以見表 4。

表 4. 不同想法地圖與 DIM 元素的關係

想法地圖					
IE 地圖			ICF 地圖		
角色 IE 地圖	內部 IE 地圖	外部 IE 地圖	議題地圖	概念地圖	形式地圖
角色	多重角色	多重演員	角色	角色	角色
記憶組織	設計結果	設計結果	記憶組織	記憶組織	記憶組織
角色的知識庫	場記的知識庫	場景的知識庫 舞台的知識庫	角色的知識庫	角色的知識庫	角色的知識庫

此外，為解決角色之間對於想法元件有不同解譯的問題，每一位角色的記憶還包括一個 ICF 詞典（ $Dictionary_{ICF}$ ），負責解譯不同角色產生想法元件中之議題、概念與形式的領域概念語彙，以提供角色在分散式的想法連結過程中，能有效的連結到不同角色的想法元件。有關 ICF 詞典在下列小節（4.2.3.4）有更進一步說明。

4.2.3.1 想法地圖定義

在進行說明下列不同想法地圖之前，作者先進行說明有關想法地圖的一般定義。

根據作者的先前研究[27]，參與者在連結想法的過程中，對於想法產生的順序並無直接關係，同時，兩個想法之間只存在一種想法聯想原則所建立的關聯性。因此，在 DIM 中的想法地圖中，在兩個不同想法元件之間的關係，最多只有一種類型的元件連結，而在兩個不同的想法元件中所包含的相同屬性間，DIM 可以允許三種不同類型的屬性連結，而建立想法元件之間的關聯性。

架構在 Wilson[78]的圖像理論，在 IE 地圖的圖像似結構中，想法元件稱之為節點 (vertex)，兩鄰接 (adjacent) 想法元件間的元件連結為邊 (edge)，兩節點之間至多為一個邊 (最少為無邊)，邊為無權重 (weight) 且無方向性 (direction)；另外，一個節點可以與一個以上的邊相連接，此節點相關聯 (incident) 的邊緣數量，稱之為想法等級 (degree)，且節點無迴圈 (loop)。另外，邊所關連的兩個節點無秩序 (order) 的關係。

在一個 IE 地圖中，起始想法元件 (或節點) 被視為一個輸入節點 (input vertex)；另外，所謂 IE 地圖的路徑 (path) 是指一個有限的連結序列中，這個連結序列從起始想法元件 (或輸入節點) 到終點想法元件的連結；在這個連結的序列，連結的數目被稱為路徑的長度 (length)，例如：在圖 26 中的 IE 地圖中， $ie_1 \xrightarrow{SI} ie_2 \xrightarrow{SI} ie_3 \xrightarrow{CR} ie_7$ 從 ie_1 到 ie_7 的路徑長度為 3。



圖 26. 一個 IE 地圖

在三種不同類型的 ICF 地圖中，每一種類型地圖之想法元件的屬性為節點，兩鄰接相同屬性的屬性連結為邊，兩節點之間至多為三個邊 (最少為無邊)，邊為無權重且無方向性；另外，一個節點可以與一個以上的邊相連接，此節點相關聯的邊之數量，稱之為屬性等級，且節點無迴圈。另外，邊所關連的兩個節點無秩序的關係。

同樣的，在任一種 ICF 地圖中，一個路徑是指一個有限的連結序列，這個連結序列是指從起始屬性到終點屬性的連結；在這個連結的序列，連結的數目被稱之為路徑的長度。根據上述地圖的定義，DIM 包括了不同種類的 IE 地圖與 ICF 地圖，這些地圖建構角色與其他 DIM 元素的記憶組織。

4.2.3.2 IE 地圖

在 DIM 中，主要有三種不同型態的 IE 地圖，包括角色 IE 地圖、內部 IE 地圖與外部 IE 地圖，每一種 IE 地圖皆有不同的特質與目的，同時，這些 IE 地圖主要都是由不同的想法元件與三種不同型態的元件連結（ SI 、 CR 和 CI ）所構成（如圖 25）。

- 角色 IE 地圖 (Map_{IE_R}): 角色 IE 地圖是一位角色內在的記憶組織，此記憶組織提供角色本身，在一特定的設計工作中，藉由不同類型的元件連結，連結到角色 IE 地圖中的想法元件，用以產生此設計工作相關設計問題的解決方法。此角色 IE 地圖被儲存在角色個別的記憶組織（表 6）。
- 內部 IE 地圖 (Map_{IE_I}): 在一個場景中，一位演員（使用者或導演）為解決一個指定的議題，在想法連結的過程中，經由扮演多重的角色而進行想法元件之間的連結，其最後產生的設計結果，是一個由不同想法元件與元件連結所建構的想法地圖，此想法地圖稱之為內部 IE 地圖。此內部 IE 地圖被儲存在場記的知識庫中（表 6）。
- 外部 IE 地圖 (Map_{IE_E}): 在一個場景中，多重演員（使用者和導演）為解決一個指定的議題，在想法連結的過程中，彼此之間進行想法元件之間的連結，其最後產生的設計結果，是一個由不同想法元件與元件連結所建構的想法地圖，此想法地圖稱之為外部 IE 地圖。此外部 IE 地圖同時被儲存在此場景和舞台的知識庫中（表 6）。

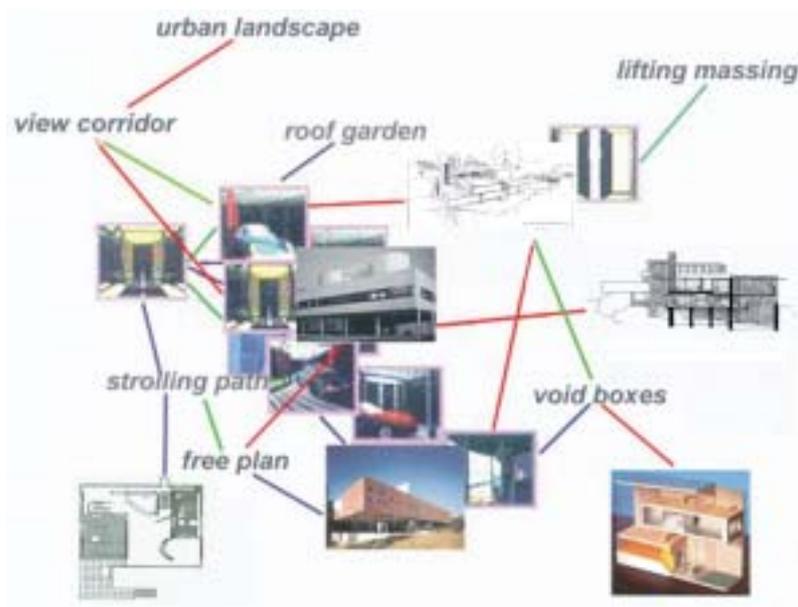


圖 27. 一個結合多媒體資訊的 IE 地圖

此外，這三種 IE 地圖藉由形式屬性值（領域概念語彙）與多媒體資訊（包括如設計案例的影像照片、建築繪圖或 3D 電腦模型等）的結合，而建構一個結合多媒體資訊的想法地圖（圖 27）。在想法連結的過程中，這些 IE 地圖會因為設計情境的改變（如設計問題、參與時間或參與人數等），或演員與角色彼此之間的相互作用關係，這些知識網路結構會產生動態性的轉化。

4.2.3.3 ICF 地圖

基本上，一個想法元件可以藉由不同型態的元件連結而連結到其他的想法元件，同時，此想法元件也可以藉由其屬性間之不同屬性連結（ si 、 cr 或 ci ），並經由 DIM 原則的機制（詳見 4.3 章節），將此想法元件中某特定屬性之值予以置換（*convert*），予以連結到其他的想法元件。

為了解釋此屬性置換的機制，作者使用解決“採光”議題的例子加以說明。當一個想法元件的屬性內容可以由（icfp）表示時，參與者產生的三個想法元件為 ie_a 、 ie_b 與 ie_c ，其屬性內容依序為（採光 虛空間 中庭 1）（採光 實空間 花園 1）與（光線 虛空間 花園 1），因此，當 ie_a 中的概念值“虛空間”被對比性的概念值“實空間”置換時，則在 ie_a 與 ie_b 之間建立元件對比連結（ $ie_a \xrightarrow{cr} ie_b$ ），而當 ie_a 中的議題值“採光”被相似性的議題值“光線”的置換時，則在這兩個想法元件之間建立元件相似連結（ $ie_a \xrightarrow{si} ie_b$ ），此外， ie_a 也可以藉由與採光具有相鄰性議題值“通風”的置換，而在 ie_a 與 ie_d （通風 虛空間 中庭 1）建立元件相鄰連結（ $ie_a \xrightarrow{ci} ie_d$ ）。有關上述這些想法元件屬性值之間的置換可以見圖 28。

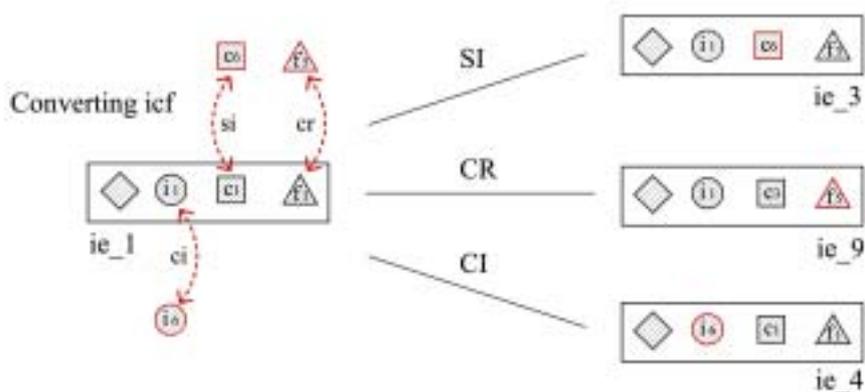


圖 28. 想法元件中屬性值之置換

DIM 包括三種類型的 ICF 地圖，以提供想法元件中屬性值能進行動態性置換，這些 ICF 地圖包括議題地圖、概念地圖與形式地圖。基本上，這些 ICF 地圖，主要由想法元件中三種屬性（議題、概念與形式）分別與不同類型的屬性連結，所共同構成的三種知識網路結構（圖 29），其主要功能是經由不同種類屬性值間之彈性

模式比對，而完成屬性值之間的置換。因此，角色可以經由連結不同的 ICF 地圖，而增加連結想法元件的多樣性與開放性。

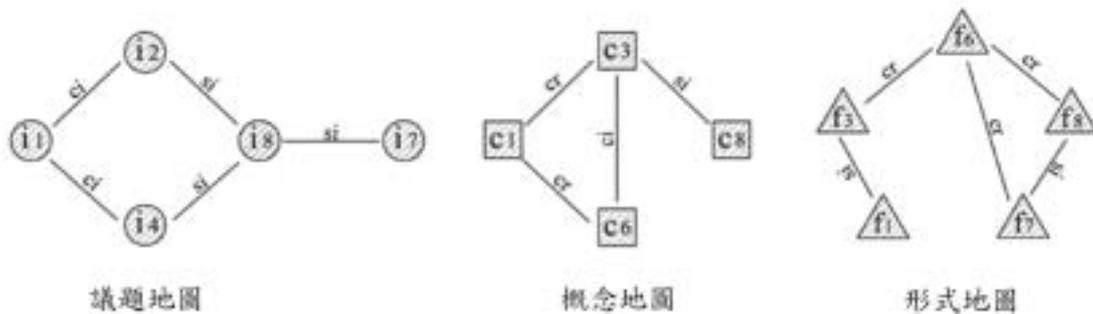


圖 29. ICF 地圖：議題地圖、概念地圖與形式地圖

- 議題地圖 (Map_I): 一群議題屬性值與不同屬性連結所構成的知識網路結構，稱之為議題地圖。角色可以經由議題地圖連結到其他關係（相似性、對比性或相鄰性）的議題值，而置換一個想法元件中的議題值。
- 概念地圖 (Map_C): 一群由概念屬性質與不同屬性連結所構成的知識網路結構，稱之為概念地圖。角色可以經由概念地圖連結到其他關係（相似性、對比性或相鄰性）的概念值，而置換一個想法元件中的概念值。
- 形式地圖 (Map_F): 一群由形式屬性質與不同屬性連結所構成的知識網路結構，稱之為形式地圖。角色可以經由形式地圖連結到其他關係（相似性、對比性或相鄰性）的形式值，而置換一個想法元件中的形式值。

這三種不同類型的 ICF 地圖 (Map_I 、 Map_C 、 Map_F) 與之前論述的角色 IE 地圖 (Map_{IE-R}) 是構成角色記憶組織的基礎。

4.2.3.4 ICF 詞典

在 DIM 中，每位角色的設計知識必須包含一個 ICF 詞典，用以解決角色之間在想法元件中議題、概念或形式之領域概念語彙的解譯問題，例如：當二位角色 (A 與 B) 使用 Stuttgart 博物館之「庭院」(court) 的形式值來解決「都市連續性」的議題時 (表 2)，角色 A 使用「庭院」(court) 的形式值描述之，而角色 B 使用「中庭」(courtyard) 的形式值描述之，然而，當存在於角色 A 之 ICF 詞典中的「庭院」有「解釋」「中庭」的領域概念語彙時，則角色 A 就能了解角色 B 在解決「都市連續性」的設計想法，並進行此想法元件的連結。

因此，ICF 詞典是由建築設計領域之不同的概念語彙所構成，提供每一位角色對於

其他角色產生想法元件中的屬性值，能進行有如詞典般的翻譯功能。這些包含在 ICF 詞典中的概念語彙被儲存在角色的知識庫中，在想法連結的過程中，當一位角色的 ICF 詞典，存有其他角色所產生想法元件之相關的概念語彙時，此角色才能進行解譯並進行這些想法元件的連結。但如果此詞典沒有相關的概念語彙時，則此角色無法進行解譯並進行這些想法元件的連結。另外，一個領域概念語彙可以同時具有在議題、概念或形式上的意義。

表 5. ICF 詞典中的詞典單元與其構成元素

ICF 詞典		
詞典單元	外部 - 內部	
單元一	view	landscape
單元二	sunlight	light
單元三	buffer	in-between
⋮		
單元 n	drying	ventilation

每個 ICF 詞典主要由許多兩兩成對的詞典單元所組成。每個詞典單元包括兩個部分：1) 每一個角色內部記憶組織（包括角色 IE 地圖與 ICF 地圖）所包含的領域概念語彙，與 2) 此角色另外能了解的外部領域概念語彙，這些外部與內部的領域概念語彙彼此不相同但具有相同的意義。例如：在角色 A 的 ICF 詞典中包含了 n 個詞典單元(表 5)，每個詞典單元包含角色的外部領域概念語彙(如 view、sunlight、buffer、drying)，分別對應與角色內部記憶組織的領域概念語彙(如 landscape、light、in-between、ventilation)。

4.3 原則

原則是角色進行想法連結的主要行為。DIM 包含三種不同的原則提供角色進行想法連結：*相似原則*、*對比原則*與*相鄰原則*。根據不同的設計情境，這三種原則經由不同的轉變規則與條件限制，提供角色動態性的選擇不同的原則形式進行推理，經由連結到角色記憶中的角色 IE 地圖、ICF 地圖與 ICF 詞典，而產生多樣性的想法元件。

在想法連結的過程中，當一個角色接收到其他角色的想法元件時，為了能在角色本身內在的記憶組織，搜尋到其他相關的想法元件，進而產生想法元件。這些原

則皆包括三個步驟以進行想法元件的搜尋，依次為解譯 (interpretation)、連結 (linking) 和選擇 (selection) (圖 30)。解譯步驟是提供角色在想法連結的過程中，透過解譯其他角色產生之想法元件中的屬性值，以利角色了解並接受這些想法元件的輸入，而連結步驟則是提供角色經由適當的連結方式 (包括相似性連結、對比性連結或相鄰性連結，詳見 4.3.2 章節)，搜尋到可能的想法元件，另外，選擇步驟是將角色連結到的所有想法元件，經由比較規則的條件建立，提供角色選擇最佳輸出的想法元件。而上述三種連結方式結合其他兩個步驟 (解譯與選擇)，是構成相似原則、對比原則與相鄰原則的主要機制。

除此之外，每一位角色在進行上述原則的步驟時，會因為不同的條件限制，而影響角色使用不同原則機制產生想法元件，而這些條件限制所建立的相關規則，稱之為事件 (event)。在 DIM 中，原則的三個步驟各自包含不同類型的事件，有關這些步驟與其事件類型的機制說明如下。

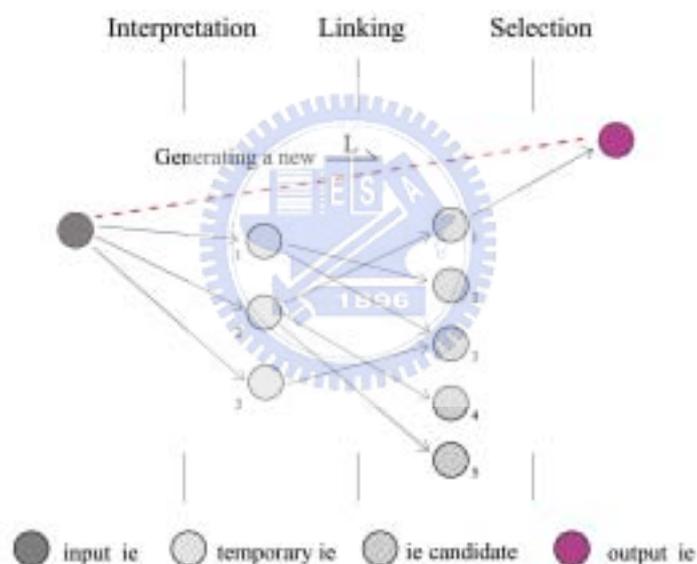


圖 30. 原則的三個步驟：解譯、連結與選擇

4.3.1 解譯

在想法連結的過程中，當一個角色 (以 $Role_1$ 為例) 接收到其他角色的想法元件時，此想法元件稱之為輸入想法元件 (input ie)，這些 input ie 中的三種屬性之值 (包括議題、概念與形式)，必須透過 $Role_1$ 知識庫中的 ICF 詞典，解譯成 $Role_1$ 之記憶組織 (包括角色 IE 地圖與三種 ICF 地圖) 存有的屬性值時， $Role_1$ 才能進行不同想法元件與屬性之間的動態連結，但如果 $Role_1$ 的 ICF 詞典沒有這些 input ie 中屬性相關的領域概念語彙，則 $Role_1$ 無法進行這些 input ie 的解譯，同時，也無法進行 $Role_1$ 內部記憶組織中其他想法元件的連結。

在想法元件解譯機制的過程中， $Role_1$ 所接收到的 $input_{ie}$ ，會經由 $Role_1$ 的 ICF 詞典對這些 $input_{ie}$ 中的屬性值進行解譯，而被解譯成功所形成之過度性的想法元件，將提供 $Role_1$ 在其內部的角色 IE 地圖與 ICF 地圖，進行關聯性想法元件與屬性之間的連結。由於這些想法元件的狀態是臨時性，因此稱之為臨時想法元件 ($temp_{ie}$) (圖 31)。

因此，在形成 $temp_{ie}$ 的過程中， $Role_1$ 對於所接收到之 $input_{ie}$ 內的屬性值 (包括議題、概念或形式)，首先必須經由 $Role_1$ 的 ICF 詞典進行解譯，並根據下列二種不同的事件 (包括解譯事件 1 和解譯事件 2) 而產生 $temp_{ie}$ 。

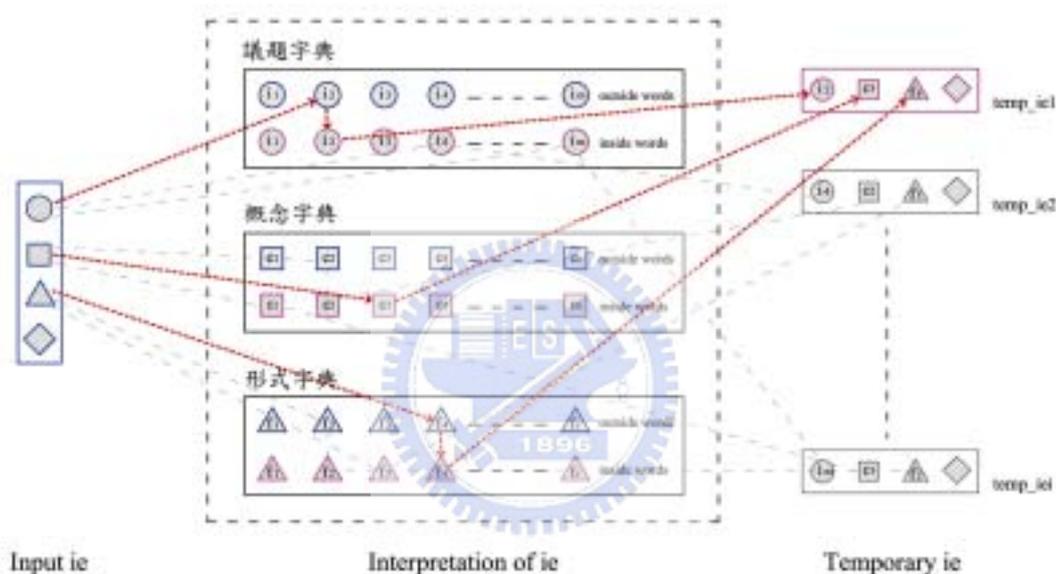


圖 31. 想法元件的解譯過程

解譯事件 1 ($Event_{i-1}$)

當 $Role_1$ 接收的 $input_{ie}$ ，此 $input_{ie}$ 的屬性值與 $Role_1$ 之 ICF 詞典中的詞典單元，有相同於 $Role_1$ 內部記憶組織的屬性值時，則此 $input_{ie}$ 的屬性值會被暫時放置在 $temp_{ie}$ 相同的屬性位置中。

解譯事件 2 ($Event_{i-2}$)

當此 $input_{ie}$ 的屬性值與 $Role_1$ 之 ICF 詞典中的詞典單元，沒有相同於 $Role_1$ 內部記憶組織的屬性值時，但此 $input_{ie}$ 屬性值有相同於 $Role_1$ 外部的領域概念語彙時，則此外部的領域概念語彙，會被相同之詞典單元的內部記憶組織的屬性值所置換，此內部記憶組織的屬性值會被放置於 $temp_{ie}$ 相同的屬性位置中。

解譯事件 3 ($Event_{i_3}$)

然而，當 $Role_1$ 不能在其 ICF 詞典中搜尋到此 $input_{ie}$ 的屬性值（具有相同此 ICF 詞典中之外部或內部的領域概念語彙）時，則此 $input_{ie}$ 的屬性值不會被放置於 $temp_{ie}$ 相同的屬性位置中，而以 nil 代替之。

最後，當此 $input_{ie}$ 中的屬性值經過上述不同的事件，經過解譯的屬性值彼此相互組合，並根據 $Role_1$ 本身對 $temp_{ie}$ 解譯的容許 (tolerance) 限制，產生此 $Role_1$ 的 $temp_{ie}$ (圖 31)。所謂容許限制是一位角色對於 $input_{ie}$ 在議題、概念與形式的三個屬性值，需要被解譯成其 ICF 詞典包含的領域概念語彙之屬性數量的接受度，並以數值 t 表示此容許限制。因此，在進行想法元件的解譯時，當角色指定的 t 值越高，代表此角色對於 $input_{ie}$ 接受度越小，相對的，當角色指定的 t 越低，代表此角色對於 $input_{ie}$ 接受度越大。

在 DIM 中，只有當 $input_{ie}$ 中有一個以上的屬性值，被解譯成記憶組織的屬性值時，才可以形成 $temp_{ie}$ ，並進行角色內部記憶組織的想法元件連結，因此，角色解譯的容許限制範圍在一與三之間 ($1 \leq t \leq 3$)。

4.3.2 連結

當 $Role_1$ 產生不同的 $temp_{ie}$ 後，這些 $temp_{ie}$ 會在 $Role_1$ 的記憶組織（包括角色 IE 地圖與三種不同 ICF 地圖），進行三種不同連結方式的想法元件連結，包括相似性連結、對比性連結與相鄰性連結。這些連結方式主要藉由 $temp_{ie}$ 中的三個屬性值（議題、概念與形式）與 $Role_1$ 記憶組織中想法元件的屬性值，經由相同類型屬性值的動態性模式比對，而搜尋到關聯性的想法元件。基本上，這些連結方式可以透過下列兩種機制進行想法元件的連結。

1. 當 $temp_{ie}$ 的三個屬性的值，與 $Role_1$ 角色 IE 地圖中的想法元件的三個屬性的值相同時，則根據不同類型的元件連結，在此角色 IE 地圖中進行此 $temp_{ie}$ 的想法元件跳躍 (flight of ideas)，進而連結到此角色 IE 地圖其他相關的想法元件 (圖 31、33 和 35)。
2. 當 $temp_{ie}$ 的三個屬性值其中有一個不相同時，為了提供具語意性與彈性的連結關係，除了可以直接連結 $Role_1$ 知識庫中的想法元件而產生它們之間的元件連結外，DIM 也提供 $temp_{ie}$ 中屬性值在不同類型的 ICF 地圖中進行屬性值置換，然後進行與其它想法元件間兩個相關屬性值的模式比對，最後在它們之間產生適合的元件連結 (圖 32、34 和 36)。

而這些被 $temp_ie$ 連結到的想法元件將是 $Role_1$ 選擇為輸出想法元件的候選者 (candidate), 其稱之為 *候選想法元件* (簡稱 $cand_ie$), 此外, 這些 $temp_ie$ 與其形成的 IE 地圖會被自動的儲存在 $Role_1$ 的知識庫中, 進而改變 $Role_1$ 原有的記憶組織。為了提供角色能有效的連結想法元件, 並建立想法元件間的連結關係, 這三種連結方式擁有各自的連結機制與目的, 其說明如下:

4.3.2.1 相似性連結

相似性連結強調在同一個設計問題 (議題) 下, 連結並產生具相似性解決此設計問題的想法元件, 進而產生想法元件之間的元件相似連結。因此, 在連結想法元件的過程中, 由於 $temp_ie$ 在相同屬性值數量的差異, 會遇到下列四種類型的事件 (包括連結事件 1、連結事件 2、連結事件 3 與連結事件 4), 予以連結到 $Role_1$ 其他相關的想法元件。

連結事件 1 ($Event_{1-1}$)

當 $temp_ie$ (以 $temp_ie_a$ 為例) 中的三個屬性之值 ($temp_i_a$ 、 $temp_c_a$ 與 $temp_f_a$), 與 $Role_1$ 角色 IE 地圖中想法元件 (以 ie_b 為例, 其屬性值分別為 i_b 、 c_b 和 f_b) 的三個屬性之值相同, 則 $temp_ie$ 藉由 ie_b 進行想法元件的跳躍, 進而連結到 $Role_1$ 角色 IE 地圖中與 ie_b 有元件相似連結 (SI) 的想法元件 ie_c (圖 32)。

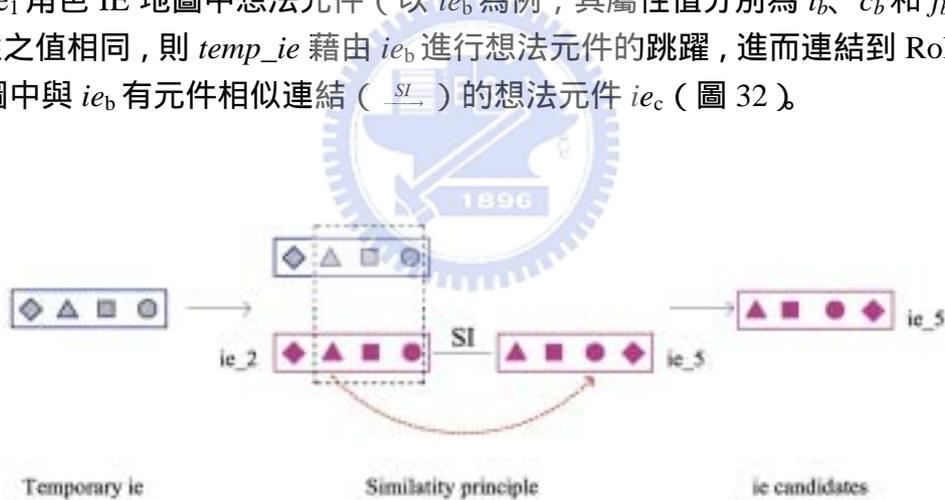


圖 32. 想法元件跳躍 (相似性連結)

然而, 當 $temp_ie$ 的三個屬性的值, 其中有一個不相同時, 經解譯完成 $temp_ie$ 中的二個屬性 (議題與形式, 或議題與概念) 的值, 藉其所屬的 ICF 地圖中屬性值的置換, 與 $Role_1$ 其他想法元件相對應的二個屬性進行圖像比對, 當兩個相對應之屬性的值, 相同或具有屬性相似連結 (si) 時, 凡具有不同於 $temp_ie$ 概念之值或形式之值的想法元件就會被連結 (圖 33), 主要有下列事件發生。

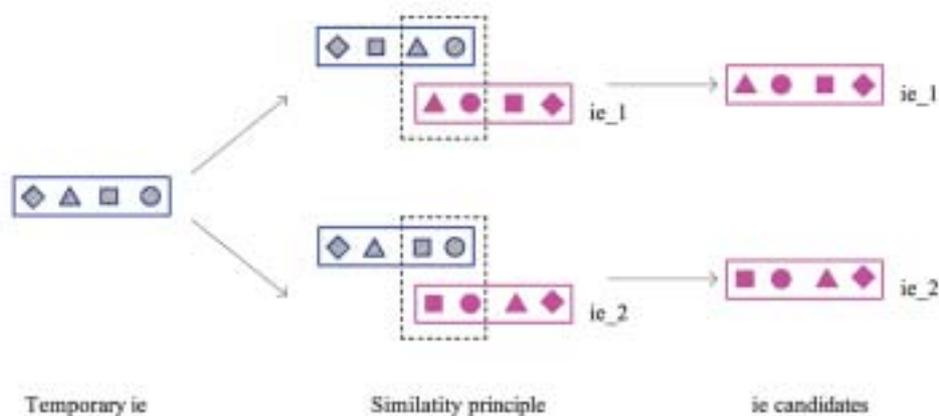


圖 33. 相似性連結

連結事件 2 ($Event_{i_2}$)

當 $temp_ie_a$ 中的二個屬性($temp_i_a$ 與 $temp_c_a$, 或 $temp_i_a$ 與 $temp_f_a$) 的值, 與 $Role_1$ 內部想法元件 ie_b 相對應的二個屬性(i_b 與 c_b , 或 i_b 與 f_b) 的值完全相同時, 且另外一個 ICF 屬性值不相同, 在此事件主要包括 1) 在形式值不相同, $temp_ie$ 的議題值與概念值和 ie_b 的議題值與概念值相同時, 或 2) 在概念值不相同, $temp_ie$ 的議題值與形式值和 ie_b 的議題值與形式值相同時, 則在 $temp_ie_a$ 與 ie_b 之間產生元件相似連結 ($\overset{sl}{\rightarrow}$), 同時將 $temp_ie_a$ 與 $ie_b \overset{sl}{\rightarrow} temp_ie_a$ 儲存在 $Role_1$ 的知識庫。

連結事件 3 ($Event_{i_3}$)

當 $temp_ie_a$ 中的二個屬性之值, 與 ie_b 相對應的二個屬性的值其中有一個不相同時, 且另外的一個 ICF 屬性值不相同, 在此事件主要包括 1) 在形式值不相同, $temp_ie$ 與 ie_b 的議題值相同, 且 $temp_ie$ 與 ie_b 的概念值不相同時; 2) 在形式值不相同, $temp_ie$ 與 ie_b 的議題值不相同, 且 $temp_ie$ 與 ie_b 的概念值相同時; 3) 在概念值不相同, $temp_ie$ 與 ie_b 的議題值不相同, 且 $temp_ie$ 與 ie_b 的形式值相同時; 或 4) 在概念值不相同, $temp_ie$ 與 ie_b 的議題值相同, 且 $temp_ie$ 與 ie_b 的形式值不相同時, 則

此 $temp_ie_a$ 的屬性值, 會根據相似原則的機制, 經由 $Role_1$ 一種相關類型的 ICF 地圖, 去連結具有屬性相似連結 ($\overset{si}{\rightarrow}$) 的屬性值, 假如 $temp_ie_a$ 的一個屬性值連結到相同 ie_b 的一個屬性值時, 則在 $temp_ie_a$ 與 ie_b 之間產生元件相似連結 ($\overset{sl}{\rightarrow}$), 同時將 $temp_ie_a$ 與 $ie_b \overset{sl}{\rightarrow} temp_ie_a$ 儲存在 $Role_1$ 的知識庫。

假如 $temp_ie_a$ 的屬性在 $Role_1$ 相關的 ICF 地圖中, 連結不到相同於 ie_b 的相關屬性值時, 則 ie_b 不會被 $temp_ie_a$ 所連結, 同時也不會將 $temp_ie_a$ 儲存在 $Role_1$ 的知識庫。

連結事件 4 ($Event_{i_4}$)

當 $temp_{ie_a}$ 中的二個屬性之值，與 ie_b 相對應的二個屬性的值都不相同時，且另外的一個 ICF 屬性值不相同，在此事件主要包括 1) 在形式值不相同，且 $temp_{ie}$ 與 ie_b 的議題值與概念值不相同時；2) 在概念值不相同，且 $temp_{ie}$ 與 ie_b 的議題值與形式值不相同時，則

此 $temp_{ie_a}$ 的屬性值，會根據相似原則的機制，經由 $Role_1$ 二種相關類型的 ICF 地圖，去連結具有屬性相似連結 (\underline{si}) 的屬性值，假如 $temp_{ie_a}$ 的二個屬性值連結到相同 ie_b 的二個屬性值時，則在 $temp_{ie_a}$ 與 ie_b 之間產生元件相似連結 (\underline{sl})，同時將 $temp_{ie_a}$ 與 $ie_b \xrightarrow{sl} temp_{ie_a}$ 儲存在 $Role_1$ 的知識庫。

假如 $temp_{ie_a}$ 的屬性在 $Role_1$ 相關的 ICF 地圖中，連結不到二個相同於 ie_b 的相關屬性值時，則 ie_b 不會被 $temp_{ie_a}$ 所連結，同時也不會將 $temp_{ie_a}$ 儲存在 $Role_1$ 的知識庫。其語言定義的描述如下：

4.3.2.2 對比性連結

對比性連結強調在同一個設計問題 (或稱議題) 下，連結並產生具對比性解決此設計問題的想法元件，進而產生想法元件之間的元件對比連結。因此，在連結想法元件的過程中，由於 $temp_{ie}$ 在相同屬性值數量的差異，會遇到下列三種類型的的事件 (包括連結事件 5、連結事件 6 與連結事件 7)，予以連結到 $Role_1$ 其他相關的想法元件。

連結事件 5 ($Event_{i_5}$)

當 $temp_{ie}$ 中的三個屬性之值，與 $Role_1$ 角色 IE 地圖中想法元件 ie_b 的三個屬性之值相同，則 $temp_{ie}$ 藉由 ie_b 進行想法元件的跳躍，進而連結到 $Role_1$ 角色 IE 地圖中與 ie_b 有元件對比連結 (\underline{CR}) 的想法元件 ie_c (圖 34)。

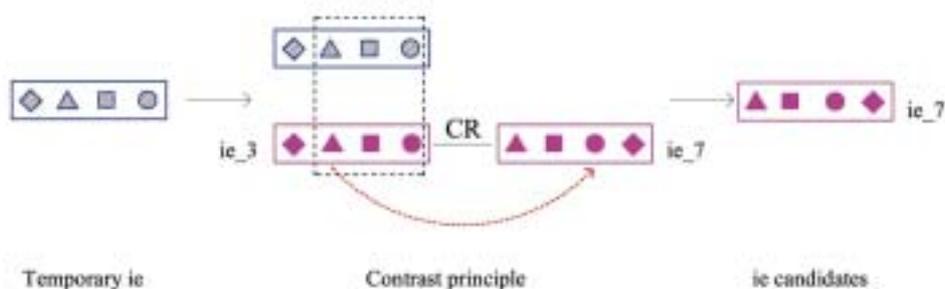


圖 34. 想法元件跳躍 (對比性連結)

然而，當 $temp_ie$ 的三個屬性的值，其中有一個不相同時，經解譯完成 $temp_ie$ 中的二個屬性（議題與形式，或議題與概念）的值，藉其所屬的 ICF 地圖中屬性值的置換，與 $Role_1$ 其他想法元件相對應的二個屬性進行圖像比對，當兩個相對應之屬性的議題值相同或具有屬性相似連結（ $_{si}$ ），且概念值或形式值具有屬性對比連結（ $_{cr}$ ）時，凡具有不同於 $temp_ie$ 概念之值或形式之值的想法元件就會被連結（圖 35）。

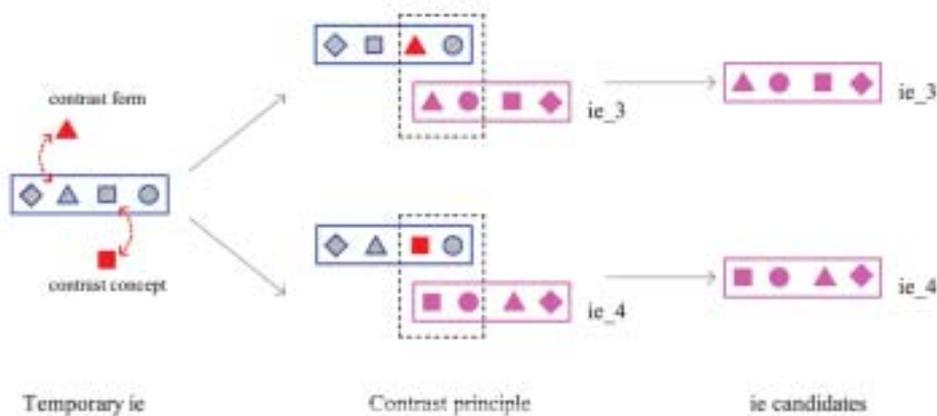


圖 35. 對比性連結

首先， $temp_ie_a$ 中的概念值（ $temp_c_a$ ）或形式值（ $temp_f_a$ ），會各別經由 $Role_1$ 的概念地圖或形式地圖，藉由這些地圖的屬性對比連結（ $_{cr}$ ），將 $temp_c_a$ 與 $temp_f_a$ 予以置換成對比性的概念值（ $_{cr}temp_c_a$ ）與形式值（ $_{cr}temp_f_a$ ），並開始進行下列三種事件的對比原則連結。然而，假如 $temp_ie_a$ 的概念值或形式值沒有在 ICF 地圖中，其中一個置換成功時，則不會進行下列事件的想法元件連結。

連結事件 6 ($Event_{1-6}$)

經過置換的概念值或形式值，當 $temp_ie_a$ 的議題值和 ie_b 的議題值相同時，且另外一個 ICF 屬性值不相同時，在此事件主要包括 1) 在形式值不相同， $temp_ie_a$ 與 ie_b 的議題值相同，且 $_{cr}temp_c_a$ 與 ie_b 概念值相同時，或 2) 在概念值不相同， $temp_ie_a$ 與 ie_b 的議題值相同，且 $_{cr}temp_f_a$ 與 ie_b 形式值相同時，則在 $temp_ie_a$ 與 ie_b 之間產生元件對比連結（ $_{CR}$ ），同時將 $temp_ie_a$ 與 ie_b $_{CR}$ $temp_ie_a$ 儲存在 $Role_1$ 的知識庫。

連結事件 7 ($Event_{1-7}$)

經過置換的概念值或形式值，當 $temp_ie_a$ 的議題值與 ie_b 的議題值不相同時，在此事件主要包括 1) 在形式值不相同， $temp_ie$ 與 ie_b 的議題值不相同，且 $_{cr}temp_c_a$ 與 ie_b 的概念值相同時；2) 在概念值不相同， $temp_ie$ 與 ie_b 的議題值不相同，且

$temp_f_a$ 與 ie_b 的形式值相同時，則

此 $temp_ie_a$ 的議題值，經由 $Role_1$ 的議題地圖，去連結具有屬性相似連結 (\xrightarrow{si}) 的屬性值，假如 $temp_ie_a$ 的議題值連結到相同 ie_b 的議題值時，則在 $temp_ie_a$ 與 ie_b 之間產生元件對比連結 (\xrightarrow{CR})，同時將 $temp_ie_a$ 與 $ie_b \xrightarrow{CR} temp_ie_a$ 儲存在 $Role_1$ 的知識庫。

假如 $temp_ie_a$ 的議題值在 $Role_1$ 的議題地圖中，連結不到相同於 ie_b 的議題值時，則 ie_b 不會被 $temp_ie_a$ 所連結，同時也不會將 $temp_ie_a$ 儲存在 $Role_1$ 的知識庫。其語言定義的描述如下：

然而，當 $temp_ie_a$ 的議題值和經過置換的概念值或形式值，與 $Role_1$ 內部想法元件 ie_b 相對應的二個屬性的值沒有相同時，則 ie_b 不會被 $temp_ie_a$ 所連結，同時也不會將 $temp_ie_a$ 儲存在 $Role_1$ 的知識庫。其語言定義的描述如下：

4.3.2.3 相鄰性連結

相鄰性連結強調在同時具有相同解決設計問題的方法下，連結並產生其他不同的設計問題（議題），進而產生想法元件之間的元件相鄰連結。因此，在連結想法元件的過程中，由於 $temp_ie$ 在相同屬性值數量的差異，會遇到下列五種類型的事件（包括連結事件 8、連結事件 9、連結事件 10、連結事件 11 與連結事件 12），予以連結到 $Role_1$ 其他相關的想法元件。

連結事件 8 ($Event_{i_8}$)

當 $temp_ie$ 中的三個屬性之值，與 $Role_1$ 角色 IE 地圖中想法元件 ie_b 的三個屬性之值相同，則 $temp_ie$ 藉由 ie_b 進行想法元件的跳躍，進而連結到 $Role_1$ 角色 IE 地圖中與 ie_b 有元件相鄰連結 (\xrightarrow{CI}) 的想法元件 ie_c (圖 36)。

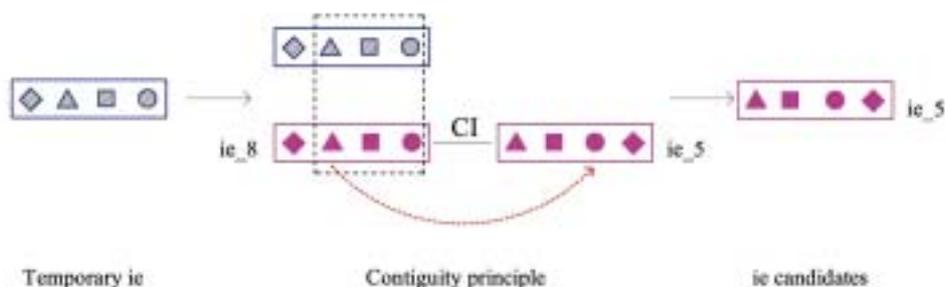


圖 36. 想法元件跳躍 (相鄰性連結)

然而，當 $temp_ie$ 的三個屬性的值，其中有一個不相同時，經解譯完成 $temp_ie$ 中的概念與形式之值，分別藉由概念地圖與形式地圖之屬性值的置換，與 $Role_1$ 其他想法元件的概念與形式之值進行圖像比對，當兩個相對應的屬性之值，相同或具有屬性相似連結 ($\overset{si}{\rightarrow}$) 時，凡具有不同於 $temp_ie$ 議題之值的想法元件就會被連結 (圖 37)，主要有下列事件發生。

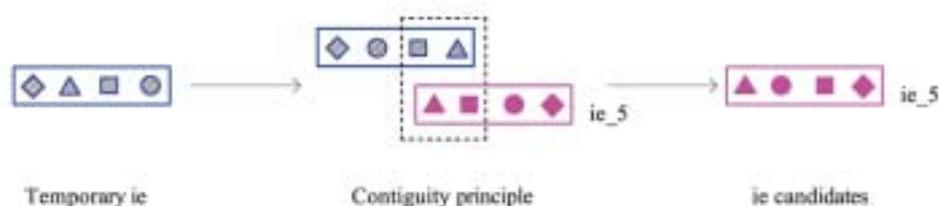


圖 37. 相鄰性連結

連結事件 9 ($Event_{1-9}$)

當 $temp_ie_a$ 中的概念屬性 ($temp_i_a$) 與形式屬性 ($temp_c_a$) 的值，與 ie_b 二個相對應屬性 (c_b 與 f_b) 的值完全相同，且議題值不相同時，則在 $temp_ie_a$ 與 ie_b 之間產生元件相鄰連結 ($\overset{cl}{\rightarrow}$)，同時將 $temp_ie_a$ 與 $ie_b \overset{cl}{\rightarrow} temp_ie_a$ 儲存在 $Role_1$ 的知識庫。

連結事件 10 ($Event_{1-10}$)

當 $temp_ie_a$ 中的概念屬性 ($temp_i_a$) 與形式屬性 ($temp_c_a$) 的值，與 ie_b 相對應的二個屬性 (c_b 與 f_b) 之值其中有一個不相同，且議題值不相同時，在此事件主要包括 1) 在議題值不相同， $temp_ie$ 與 ie_b 的概念值相同，且 $temp_ie$ 與 ie_b 的形式值不相同時；或 2) 在議題值不相同， $temp_ie$ 與 ie_b 的概念值不相同，且 $temp_ie$ 與 ie_b 的形式值相同時，則

此 $temp_ie_a$ 的屬性值，會根據相鄰原則的機制，經由 $Role_1$ 形式地圖或概念地圖，去連結具有屬性相似連結 ($\overset{si}{\rightarrow}$) 的屬性值，假如 $temp_ie_a$ 的一個屬性值連結到相同 ie_b 的一個屬性值時，則在 $temp_ie_a$ 與 ie_b 之間產生元件相鄰連結 ($\overset{cl}{\rightarrow}$)，同時將 $temp_ie_a$ 與 $ie_b \overset{cl}{\rightarrow} temp_ie_a$ 儲存在 $Role_1$ 的知識庫。

假如 $temp_ie_a$ 的屬性在 $Role_1$ 相關的 ICF 地圖中，連結不到相同於 ie_b 的相關屬性值時，則 ie_b 不會被 $temp_ie_a$ 所連結，同時也不會將 $temp_ie_a$ 儲存在 $Role_1$ 的知識庫。

連結事件 11 ($Event_{11}$)

當 $temp_ie_a$ 中的概念屬性 ($temp_i_a$) 與形式屬性 ($temp_c_a$) 的值，與 ie_b 相對應的二個屬性 (c_b 與 f_b) 之值都不相同時，且議題值不相同，則此 $temp_ie_a$ 的屬性值，會根據相鄰原則的機制，經由 $Role_1$ 的形式地圖與概念地圖，去連結具有屬性相似連結 (si) 的屬性值，假如 $temp_ie_a$ 的這二個屬性值連結到相同 ie_b 的二個屬性值時，則在 $temp_ie_a$ 與 ie_b 之間產生元件相似連結 (cl)，同時將 $temp_ie_a$ 與 ie_b cl $temp_ie_a$ 儲存在 $Role_1$ 的知識庫。

假如 $temp_ie_a$ 的屬性在 $Role_1$ 的形式地圖與概念地圖中，連結不到二個相同於 ie_b 的相關屬性值時，則 ie_b 不會被 $temp_ie_a$ 所連結，同時也不會將 $temp_ie_a$ 儲存在 $Role_1$ 的知識庫。

連結事件 12 ($Event_{12}$)

另外， $temp_ie_a$ 的議題值，可以經由 $Role_1$ 的議題地圖中具有與 $temp_i_a$ 有屬性相鄰連結 (ci) 的議題值置換後，此置換的議題值 $ci temp_i_a$ 和 $temp_ie_a$ 原有的概念值與形式值，與 ie_b 相對應的三個 ICF 屬性之值相同時，則在 $temp_ie_a$ 與 ie_b 之間產生元件相似連結 (cl)，同時將 $temp_ie_a$ 與 ie_b cl $temp_ie_a$ 儲存在 $Role_1$ 的知識庫。

假如經置換議題值的 $temp_ie_a$ ，與 ie_b 相對應的三個 ICF 屬性之值有一個不相同時，則 ie_b 不會被 $temp_ie_a$ 所連結，同時也不會將 $temp_ie_a$ 儲存在 $Role_1$ 的知識庫。

除此之外，上述的三種連結方式與其包含不同類型的事件，被動態性的組織在角色的記憶組織中，以提供角色能根據自己的喜好或不同的設計情境，動態性的選擇不同的原則形式 (相似原則、對比原則與相鄰原則)，進而連結與產生特定連結關係的想法元件。然而，當角色選擇的原則形式無法連結到角色記憶組織中的想法元件時，則 DIM 能自動且隨機的給予角色其它的原則形式，以利想法連結的進行。

4.3.3 選擇

在想法元件選擇的過程， $temp_ie$ 藉由不同連結原則所連結到的候選想法元件 ($cand_ie$)，根據每個角色喜好的連結原則，角色選擇一個具有最大喜好值的想法元件為最後輸出的想法元件 (稱之為輸出想法元件，簡稱 $output_ie$)；但假如有二個以上最大喜好值的想法元件時，則角色可以隨機的選擇一個想法元件為 $output_ie$ 。

如果當角色喜好的連結原則沒有連結到候選想法元件時，則角色可以藉由其它兩

種連結原則所連結的候選想法元件，隨機的選擇一個具有最大喜好值的想法元件為 *output_ie*。

為了提供角色能有效且動態性連結並產生想法元件，上述原則的三個步驟（解譯、連結與選擇）與連結步驟的三種連結方式（相似性連結、對比性連結與相鄰性連結），依據原則的步驟被依序地組織成不同的事件區塊：解譯、連結與選擇，而為了提供角色能動態性的選擇原則形式，上述連結步驟的事件區塊又依據三種連結方式予以區劃。因此，DIM 的整個事件區塊包括解譯、相似性連結、對比性連結、相鄰性連結和選擇，而其中三種連結方式之區塊的組織序列，可以依據角色的選擇而動態性地改變。

4.3.4 想法地圖轉化

這些經由角色使用原則而產生的想法元件與連結，在想法連結的過程中，提供不同種類的想法地圖進行動態性的轉化，包括 *IE 地圖轉化* 與 *ICF 地圖轉化*。

4.3.4.1 IE 地圖轉化

IE 地圖轉化係指 DIM 元素（角色、場記、場景）之知識庫，在想法連結的過程中，其所屬知識庫中的 IE 地圖會動態性的增加想法元件與元件連結。因此，IE 地圖轉化主要包括角色知識庫的 *角色 IE 地圖轉化*，場記知識庫的 *內部 IE 地圖轉化*，與場景和舞台知識庫的 *外部 IE 地圖轉化*。

- **角色 IE 地圖轉化**：角色 IE 地圖轉化為一位角色在進行想法連結時，經由解譯所產生的 *temp_ie* 與元件連結，會被動態性的增加在此角色的角色 IE 地圖中，因而轉化了原先的角色 IE 地圖，而此種轉化角色 IE 地圖的方式，改變了原有角色的記憶組織，可以被視為一種角色學習的機制。例如：在想法連結的過程中，一位角色接收到的 *input_ie* 經由解譯而形成 *temp_ie*，當這些 *temp_ie* 在連結的步驟中，與原有角色 IE 地圖產生元件連結時，則這些 *temp_ie* 與元件連結會被自動的增加至角色知識庫中的角色 IE 地圖。
- **內部 IE 地圖轉化**：內部 IE 地圖轉化為強調多重角色在進行想法連結時，這些角色共同產生的想法元件與元件連結，會被動態性的貯藏在場記的知識庫，相對的，也改變場記知識庫原有的內部 IE 地圖。例如：當多重角色在進行想法連結的過程中，每一位角色接收到 *input_ie*，經由不同原則所產生 *output_ie* 與相關的元件連結，會被自動的增加到場記知識庫中的內部 IE 地圖。
- **外部 IE 地圖轉化**：外部 IE 地圖轉化為強調多重演員（使用者與導演）之間在進行想法連結時，這些演員共同產生的想法元件與元件連結，會被動態性的

儲藏在場景和舞台的知識庫，相對的，也改變場景和舞台知識庫原有的外部 IE 地圖。例如：當不同演員在進行想法連結的過程中，每一個演員接收到 *input_ie*，經由不同原則所產生 *output_ie* 與相關的元件連結，會被自動的增加到場景與舞台知識庫中的外部 IE 地圖。

4.3.4.2 ICF 地圖轉化

在一個特定的場景中，ICF 地圖轉化為使用者與導演在此場景的外部 IE 地圖中，根據設計共鳴現象¹[104]，所共同選擇的想法元件，稱之為此場景的**關鍵性想法元件**，此關鍵性想法元件除了會被增加在參與此場景演出之角色的知識庫，同時，此關鍵性想法元件的三個屬性之值，也會在這些角色各自的 ICF 地圖中，根據三種類型之 ICF 地圖的屬性，個別增加此關鍵性想法元件之屬性值與屬性連結。由於角色的 ICF 地圖轉化，這些角色之 ICF 詞典中的領域概念語彙也會因而增加，進而提升角色的解譯能力。

有關角色在 ICF 地圖轉化的相關機制，牽涉到使用者與導演如何在外部 IE 地圖中使用此關鍵性想法元件的方式，因此，有關 ICF 地圖轉化的運算機制，將透過本研究在設計實驗中的發現與討論，予以更進一步的探討以提供作者未來的後續研究。

4.4 連結過程

連結過程是強調舞台、場景、角色、場記與演員（使用者與導演）元素，在分散式想法連結的過程中，這些元素能根據腳本的內容敘述，在二個不同的相互作用中（內部相互作用與外部相互作用）進行訊息傳遞的溝通機制。簡言之，連結過程提供 DIM 元素，可以在內在與外在的相互作用中進行互動性的溝通（圖 38）。

在 DIM 中，每一位演員（使用者與導演）在內部與外部的相互作用中，經由扮演不同的角色，且與其他演員進行互動性溝通而連結與產生想法元件。例如：在圖 38 中，User_1 經由扮演三位不同角色（Role_1、Role_3 和 Role_5）而連結想法元件，而在外部相互作用中，User_1 同時也 and User_5、User_4、User_3、User_2 和 Director 進行互動性的溝通，而連結不同演員之間關聯性的想法元件。因此，內部相互作用是強調演員與其扮演角色彼此之間的互動性溝通，而外部相互作用則是強調多重演員彼此之間的互動性溝通。此外，在一個分散式的想法連結中，每一位演員（使用者與導演）都可以操控其自身的內部相互作用，但只有導演可以操

¹在集體腦力激盪會議中，多重參與者通常會選擇關聯性最多的想法為集體共同的想法（Osborn, 1963），此現象也經常發生在設計領域中，Lai 和 Chang（2006）稱之為“設計共鳴現象”。

控外部相互作用（圖 38）。

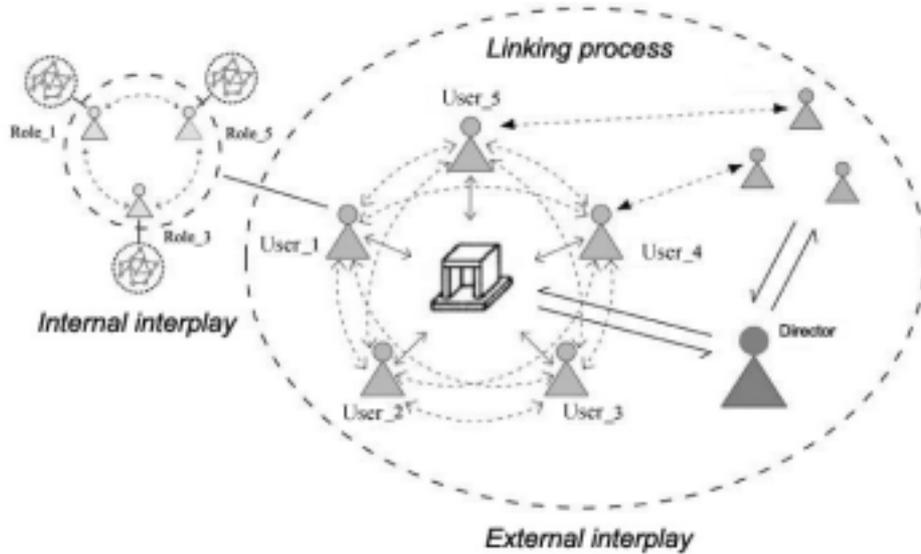


圖 38. 連結過程與內外相互作用

4.4.1 外部相互作用

在外部相互作用中，每一位演員為了產生設計想法，此演員必須反應不同的設計情境（包括如設計問題、參與時間或人數等），同時和舞台、場景與其他演員進行相互作用。在外部相互作用中，每一位演員產生的想法元件會被儲存在場景的知識庫，而其他演員再根據儲存在此知識庫的想法元件與其之間的元件連結，予以反應並採取適當的動作，以連結與產生其它的想法元件。

基本上，DIM 只允許一位導演存在於一個分散式的想法連結中，此導演除了與其他演員進行想法連結外，同時也負責操控整個外部相互作用的流程。在進行一個場景的外部相互作用時，導演首先將相關的訊息內容傳遞給舞台，再經由場景傳遞相關的訊息給不同演員，然後這些演員才開始進行外部相互作用的想法連結（圖 39）。

在連結過程中，每一位演員在各自的內部相互作用扮演不同角色，並被動態性的賦予這些角色的知識與原則，予以連結並產生想法元件。經由舞台、場景與不同演員之間的互動，這些演員所產生的想法元件與元件連結，會被自動的增加在場景知識庫中的外部 IE 地圖，此地圖將提供這些演員選擇關鍵性想法與尋求想法組合的依據。

當此場景的外部相互作用結束時，最後產生的外部 IE 地圖除了儲存在場景的知識庫外，同時，也儲存在舞台的知識庫。而根據設計的共鳴現象，場景會在其知識

庫中的外部 IE 地圖，自動的選擇一個具有最多想法等級的想法元件為輸出，然而，當最多想法等級的想法元件超過二個以上時，則場景將在這些想法元件中隨機的選擇一個。這個輸出的想法元件將提供這些演員解決此場景議題方法的參考依據。

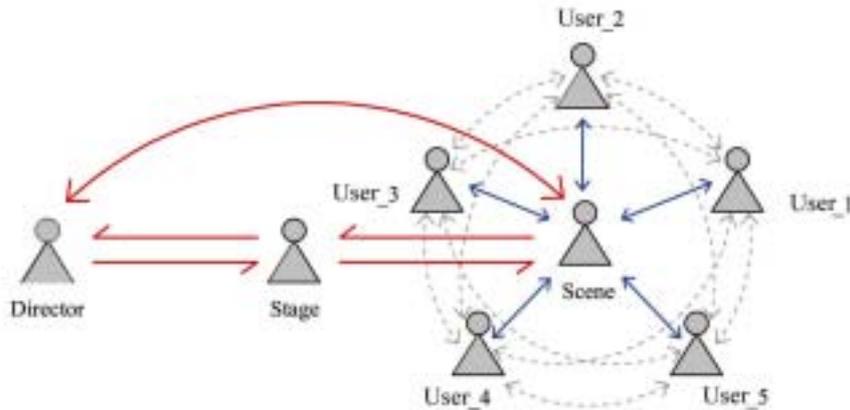


圖 39. 外部相互作用

4.4.2 內部相互作用

在內部相互作用中，每一位演員為了產生想法元件，此演員須在外部 IE 地圖中選擇一個想法元件，並藉由扮演不同的角色，且賦予每一位角色喜好的原則（相似原則、對比原則與相鄰原則）與解譯的容許限制，而連結到這些角色的記憶組織（包括角色 IE 地圖 ICF 地圖與 ICF 詞典），進而，連結與產生多樣性的想法元件。在此想法連結的過程中，DIM 提供一位場記負責傳遞與儲存這些角色產生的想法元件（圖 40）。

例如在圖 40 中，User_1 選擇五位角色（Role1、Role2、Role3、Role4 和 Role5）進行角色扮演，以連結與產生想法元件。在連結想法過程中，場記根據角色產生想法元件的順序，依序傳遞不同角色所產生的輸出想法元件，並經由 User_1、場記與這五位角色彼此之間的相互作用，這些角色所產生的想法元件與其之間的元件連結，被自動的增加在場記知識庫的內部 IE 地圖中，予以提供 User_1 選擇設計想法的依據。

當此內部相互作用結束時，根據設計的共鳴現象，場記會在此內部 IE 地圖中，自動的選擇一個具有最多想法等級的想法元件，以作為 User_1 輸出想法元件的參考依據。然而，當有二個以上具有最多想法等級的想法元件時，則場記會在這些想法元件中隨機的選擇一個。

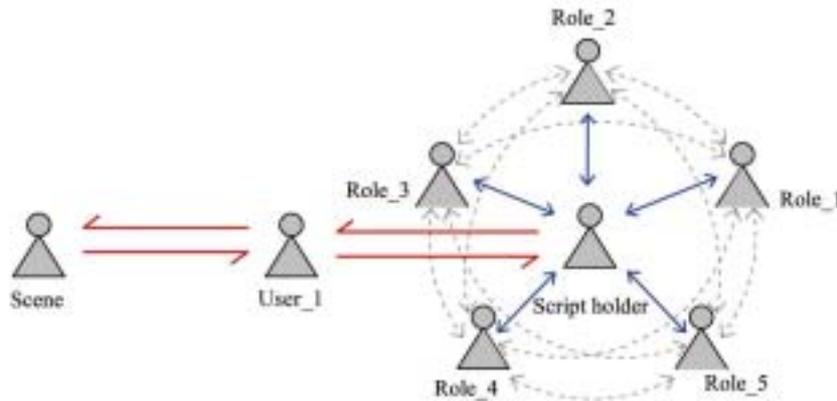


圖 40. 內部相互作用

4.4.3 連結溝通語言

為了使這些元素在內部與外部的相互作用中，能進行互動性的溝通，這些元素之間的溝通語言扮演一個重要的關鍵。在 DIM 中，提供這些元素溝通時所使用的語言，稱之為**連結溝通語言** (*Linking Communication Language*，簡稱 *LCL*)。

基本上，連結溝通語言主要包括下列的元素：**發送者** (*s*)、**接收者** (*r*)、**訊息內容** (*lc*)、**訊息語言** (*ll*)、**語言行動** (*la*) 與**知識本體論** (*lo*)。發送者與接收者決定 DIM 元素在相互作用中傳遞訊息的對象，訊息內容是提供這些元素進行相互作用時的相關資訊，訊息語言係指傳遞訊息內容所使用的語法 (*syntax*)，而語言行動則是提供這些元素對於訊息內容與對象進行反應，並採取適當的內部行動，另外，知識本體論為表現訊息內容所使用的符號語彙 (*vocabulary of symbols*)。

由於 DIM 模型為提供相同設計領域的設計者進行分散式的想法連結，因此，連結溝通語言架構在相同的知識本體論，同時，這些元素使用相同的訊息語言去描述訊息內容，以支持這些元素在相互作用中進行溝通。

4.4.3.1 發送者與接收者

在傳遞訊息過程中，發送者與接收者必須相互知道對方的身分“identity”，才可以採取其內部適當的互動行為，予以反應不同的相互作用關係。例如，在外部相互作用中，舞台 (*s*) 傳遞訊息給場景 (*r*)，同時，使用者 (*s*) 也傳遞訊息給此場景 (*r*)，因此，此場景必須知道這兩位接收者的身分，才可以採取適當的行動予以反應；另外，舞台與使用者的身分也提供此場景，提供此場景理解此訊息被傳遞的原因與其對外部相互作用的期望。

在內在相互作用中，發送者與接收者主要包括角色、場記與演員（包括設計者或導演），在外在相互作用中，主要包括演員、場景與舞台。因此，在訊息傳遞的過程中，這些元素會因為發送者與接收者身分的轉換，而採取不同的內部行動予以反應，另外，發送者與接收者之間訊息傳遞的方式，包括傳遞數量與傳遞方向，也會影響這些元素的內部行為機制。

在想法連結的過程中，一個使用者會將相關的設計議題傳遞給所有參與者，藉由使用者彼此之間的訊息交換，予以產生多樣性的設計想法。因此，在 DIM 的相互作用過程中，不同 DIM 元素在發送者與接收者之間的傳遞數量，主要包括一對多與一對一的關係，例如：在一對一的傳遞數量中，一個使用者 (s) 傳遞想法元件給場記 (r)，並要求場記儲存此想法元件在其知識庫中，而為找尋具有某一特定設計技能的角色參與，一個場景 (s) 傳遞設計技能需求的訊息給多個角色 (r)。

另外，在有關發送者與接收者之間的訊息傳遞方向，DIM 包括單向傳遞與雙向傳遞兩種方式。例如，在單向傳遞中，一個舞台 (s) 在外部相互作用中要求一個場景 (r) 開始進行某設計議題的想法連結，此場景會將此訊息傳遞給其他 DIM 元素，而不會立即回傳給此舞台，在雙向傳遞中，場記 (s) 在內部相互作用中向角色 (r) 要求找尋想法元件的計畫，當這些角色連結到相關的想法元件時，這些角色會立即傳遞產生的想法元件給此場記。上述的描述反映了語言行動與傳遞方向與數量與密切的關係，有關語言行動在 4.4.3.3 小節再進行說明。

4.4.3.2 訊息內容

訊息內容為包含在傳遞訊息中的實際資訊，是構成 DIM 腳本元素的主要內容敘述。在不同的相互作用中，訊息內容除了上述說明元素身分的定義外，主要還包括設計技能、設計問題、想法元件、連結原則、容許限制、參與時間與參與人數等，其詳細說明如下：

- 設計技能 (lc_{skill}): 設計技能的訊息內容是角色具有的設計能力，其設計能力主要根據不同的建築類型而決定，例如住宅類型、辦公室類型、廠房類型。此訊息內容主要提供場景找尋具有相關設計技能的角色參與想法連結的依據。其內容的值是文字。
- 扮演角色 (lc_{role}): 扮演角色的訊息內容是在某一個場景中，根據設計技能的要求，使用者與導演希望扮演之角色的身分。其內容的值是文字。
- 設計問題 (lc_{issue}): 設計問題的訊息內容是想法元件中議題屬性之值。在一個特定設計工作中，通常包含數個議題值為此設計工作欲解決的設計問題，其內容的值是文字。此外，一個設計問題可以被視為一個只有議題屬性值的想

法元件。

- 想法元件 (lc_{ie}): 想法元件的訊息內容包括三個屬性之值, 包括議題、概念與形式, 主要提供不同 DIM 元素在進行想法元件的連結與產生。其內容的值是一組文字。
- IE 地圖 (lc_{map}): IE 地圖之訊息內容包括兩個想法元件與它們之間的元件連結。基本上, IE 地圖訊息主要提供場記與場景產生內部 IE 地圖與外部 IE 地圖的主要訊息內容。其內容的值是一組文字。
- 連結原則 ($lc_{principle}$): 連結原則之訊息內容主要提供這些角色連結想法元件的行為, 包括相似性、對比性與相鄰性, 這三個原則也提供輸出與輸入想法元件之間的連結關係。其內容的值是文字。
- 容許限制 ($lc_{tolerance}$): 容許限制的訊息內容的訊息內容是提供這些角色在解譯想法元件的過程中, 對於三種屬性值被解譯數量的容許範圍。其內容的值是 1 到 3 的整數字。
- 演出時間 (lc_{time}): 參與時間的訊息內容是在一個場景下, 角色、使用者與不同的參與者進行有關此場景議題之想法連結的時間, 當參與時間結束, 則此場景的所有相互作用也跟著結束。其內容的值是數字。
- 參與人數 (lc_{number}): 參與人數的訊息內容為在一個場景中下, 導演指定使用者 (包括導演) 可以扮演角色的最大數量。其內容的值是數字。

除了上述的訊息內容外, 接收者也會根據其他的設計情境而告知發送者其他的訊息內容, 例如: 當場記向角色要求產生想法元件的計畫, 而此角色的記憶沒有任何想法元件時, 此角色會告知場記 “I have no design ideas”。

4.4.3.3 語言行動

想法聯想為強調產生多樣性設計想法的行為, 而在連結想法的過程中, 每一位參與者對於其他參與者所產生的設計想法不做任何的批評, 同時, 參與者之間的溝通無承諾性的行為存在[11]。因此, 在相互作用的過程中, 這些不同 DIM 元素在傳遞訊息所使用的語言行動類型, 主要包括斷然性類型的語言行動, 包括通知 (inform) 提出計劃 (propose) 與失敗 (failure), 與命令性類型的語言行動, 包括要求 (request) 要求計劃 (call for proposal, 簡稱 CFP)。為了動態性的改變內外相互作用的狀態, 宣告性類型的語言行動也會發生在此連結過程中, 例如: 適時要求 (*request-when*)。

- 通知 (la_{inform}): 發送者將他真實相信的訊息內容傳遞給接收者。例如: 在外部相互作用中, 場景通知舞台他已經將議題的訊息內容傳遞給角色。
- 要求 ($la_{request}$): 發送者要求接收者執行她們/他們內部相關的行為, 予以互動於不同的相互作用中。例如: 在外部相互作用中, 角色要求場記將其產生的想法元件, 放置在內部 IE 地圖中。
- 適時要求 ($la_{request_when}$): 在一定的時間範圍內, 發送者要求接收者執行她們/他們內部相關的行為, 予以互動於不同的相互作用中。例如, 在一個特定的時間範圍內, 舞台要求一個場景結束其內部相互作用。
- 要求計劃 (la_{cfp}): 在一個狀態下, 發送者要求接收者執行一個動作, 此動作所得到的結果, 必須回覆給此發送者, 此語言行動可以視為發送者和接收者之間的相互協商。例如: 在內部相互作用中, 場記向角色要求產生想法元件的計劃, 予以進行內部相互作用。
- 提出計劃 ($la_{propose}$): 發送者提出一個計劃給接收者, 此語言行動通常回應上述要求計劃的語言行動。例如: 在內部相互作用中, 角色產生的輸出想法元件, 必須向場記提出計劃。
- 失敗 ($la_{failure}$): 發送者告訴接收者他/她在執行上述的要求時, 所進行的行為失敗。例如: 在內部相互作用中, 由於角色對於其他角色輸入的想法元件沒有連結到任何相關的想法元件, 因此, 此角色告訴場記有關他產生想法元件的任務失敗。

4.4.4 訊息傳遞過程

訊息傳遞過程提供 DIM 元素進行分散性想法連結的流程控制。換言之, 訊息傳遞過程控制著發送者與接收者在一個特定的時間範圍中, 他們使用的連結溝通語言, 在內部與外部相互作用中的資訊流動關係。基本上, 整個外部相互作用的訊息傳遞過程如下:

1. 首先, 導演傳遞相關訊息給舞台, 同時要求舞台開始進行想法連結與儲存此訊息內容在舞台的知識庫;
2. 舞台告知導演將開始此想法連結;

3. 根據導演所安排的場景順序，舞台傳遞相關訊息給第一個場景，同時要求場景開始進行想法連結；
4. 場景告知所有參加的使用者（包括導演）有關此場景的相關訊息內容，予以進行想法連結。同時，不同的使用者和導演根據導演所安排的使用者順序開始進行內部相互作用的想法連結；
5. 當參與時間結束時，導演要求場景儲存所有使用者與導演產生的 IE 地圖，並產生一個具有想多想法等級的想法元件為此場景輸出想法的參考計劃；
6. 場景將提出想法元件計畫給導演；
7. 導演要求將此想法元件告知所有的使用者；
8. 舞台告知使用者此想法元件，並進行下一個場景的想法連結。

舞台執行上述相同的步驟，且執行完所有的場景或由導演強制結束上述步驟，此想法連結才完整結束，而這些相互作用的紀錄會根據不同的時間，動態性的儲存在舞台的知識庫，以提供連結過程的紀錄與資料的修正與利用。此外外部相互作用的訊息傳遞過程與連結溝通語言的關係可以見表 6。

表 6. 外部相互作用中訊息傳遞過程與連結溝通語言的關係圖表

訊息傳遞過程	發送者	接收者	訊息內容	語言行動
步驟一	導演	舞台	lc_{skill} 、 lc_{issue} 、 lc_{time} 與 lc_{number}	$la_{request_when}$
步驟二	舞台	導演	I will start the play	la_{inform}
步驟三	舞台	場景	lc_{skill} 、 lc_{issue} 、 lc_{time} 與 lc_{number}	$la_{request_when}$
步驟四	場景	使用者(多)	lc_{skill} 、 lc_{issue} 、 lc_{time} 與 lc_{number}	la_{inform}
使用者與導演的內部相互作用				
步驟五	導演	場景	Generating an ie	la_{cfp}
步驟六	場景	導演	lc_{ie}	$la_{propose}$
步驟七	導演	舞台	lc_{ie}	$la_{request}$
步驟八	舞台	使用者(多)	lc_{ie}	la_{inform}

在內部相互作用中，首先，使用者或導演從場景的外部 IE 地圖選擇一個想法元件，視為 *input_ie*（如果此場景剛開始進行，此想法元件只包含議題屬性值），然後進行此使用者或導演的內部相互作用，整個訊息傳遞過程的步驟詳細說明如下：

1. 在一定時間範圍內，一個使用者（以 User_1 為例）傳遞此場景的相關訊息內容，要求場記進行想法元件的連結，但如果為此場景的起始，則此想法元件為設計問題（只有議題值的想法元件）；
2. 當場記接收到此訊息時，場記告知 User_1 他開始進行想法元件的連結；
3. 根據角色的註冊順序，場記分別向不同的角色要求產生一個想法元件與元件連結的計畫；
4. 當角色完成此計畫時，角色向場記提出一個想法元件與元件連結的計畫；
5. 同時，角色要求場記將產生的 IE 地圖（內部相互作用輸入與輸出的想法元件，與其之間的元件連結）儲存至場記的知識庫；
6. 場記告知此所有角色此儲存的 IE 地圖；
7. 當限定的時間結束時，User_1 向場記要求提出一個具有想多想法等級的想法元件為此內部相互作用輸出的參考計畫。
8. 當場記完成計畫時，場記向 User_1 提出一個想法元件（*output_ie*）的計畫。
9. User_1 要求場景將產生的 IE 地圖（包括 *input_ie*、*output_ie* 與之間的元件連結）儲存至場景的知識庫。
10. 場景告知下一個使用者（或導演）開始進行想法連結，其詳細說明可以見於上述外部相互作用的步驟五與步驟六（表 7）。

另外，這些訊息內容與產生的內部 IE 地圖會根據不同的時間，會分別動態性的儲存在舞台和場記的知識庫，予以提供連結過程的紀錄與資料的修正與再利用。此內部相互作用的訊息傳遞過程與連結溝通語言的關係見表 7。

在上述訊息傳遞過程中，主要包括單向傳遞與雙向傳遞，而傳遞數量主要以一對一的方式進行。另外，在 DIM 中，所有的角色必須提供自己的身分與設計技能，

予以提供場景藉由發佈設計技能 (lc_{skill}) 的訊息，搜尋到具有此設計技能的角色進行此分散式的想法連結，此傳遞數量為一對多的方式進行。

表 7. 內部相互作用中訊息傳遞過程與連結溝通語言的關係圖表

訊息傳遞過程	發送者	接收者	訊息內容	語言行動
步驟一	使用者 (User_1)	場記	lc_{time} 、 lc_{role} 、 $lc_{principle}$ 、 $lc_{tolerance}$ 、 lc_{ie} (或 lc_{issue})	$la_{request_when}$
步驟二	場記	使用者 (User_1)	I will start the idea linking	la_{inform}
步驟三	場記	角色	$lc_{tolerance}$ 、 $lc_{principle}$ 與 lc_{ie} (或 lc_{issue})	la_{cfp}
步驟四	角色	場記	lc_{ie} 與 $lc_{principle}$	$la_{propose}$ 或 $la_{failure}$
步驟五	角色	場記	lc_{map}	$la_{request}$
步驟六	場記	角色 (多)	lc_{map}	la_{inform}
步驟七	使用者 (User_1)	場記	Generating an ie	la_{cfp}
步驟八	場記	使用者 (User_1)	lc_{ie}	$la_{propose}$
步驟九	使用者 (User_1)	場景	lc_{map}	$la_{request}$
步驟十	場景	使用者 (多)	lc_{ie}	la_{inform}

4.4.5 角色學習

在不同的相互作用中，參與的角色會因為彼此之間的衝突而相互學習，這種學習機制稱之為角色學習。在 DIM 中，一位角色產生的想法元件被使用者（或導演）選取的頻率，會決定此角色是否繼續活在一個場景的依據。因此，在一個場景中，存活的角色為了讓自己產生的想法元件能被使用者（或導演）所選取，而有

機會繼續存活在此場景中，這些角色會彼此衝突而相互競爭，進而演化其本身內部的設計知識。

在 DIM 中，角色學習包括兩種型態的學習機制：*競爭學習*與*演化學習*。競爭學習是在內部相互作用中，不同角色產生想法元件被使用者（或導演）選取頻率的高低，為這些角色爭取存活的依據，換言之，一位角色產生之想法元件被選擇的頻率越高，其存活的機率越大，而被選擇的頻率越低，其淘汰的機率越大（圖 41）。例如：在圖 41 中，使用者 User_1 與八位角色在 Scene_1 進行內部相互作用，當角色 Role_7 產生的想法元件被 User_1 選取的頻率最低時，Role_7 就會被淘汰在此想法連結的環境中。另一方面，當 Role_4 和 Role_5 產生的想法元件被 User_1 選取的頻率最高時，則 Role_4 和 Role_5 除了可以繼續存活在此想法連結的環境中，同時，這兩位角色也是 User_1 在之後場景扮演角色的優先選擇。

由於角色彼此之間的競爭學習，進而改變這些角色內部的設計知識，這種改變角色設計知識的學習方式稱之為*演化學習*。在演化學習中，因為不同角色之主觀意見所帶來的衝突性，可以經由角色調整其解譯的容許限制，而改變對於其他角色產生想法元件的接受度，同時，此角色的角色 IE 地圖也會因而轉化。除此之外，由於外部相互作用之衝突性所產生的設計共鳴現象，而導致角色在 ICF 地圖轉化的學習機制，作者將依據第六章的設計實驗發現，再進行此學習機制的討論與說明。

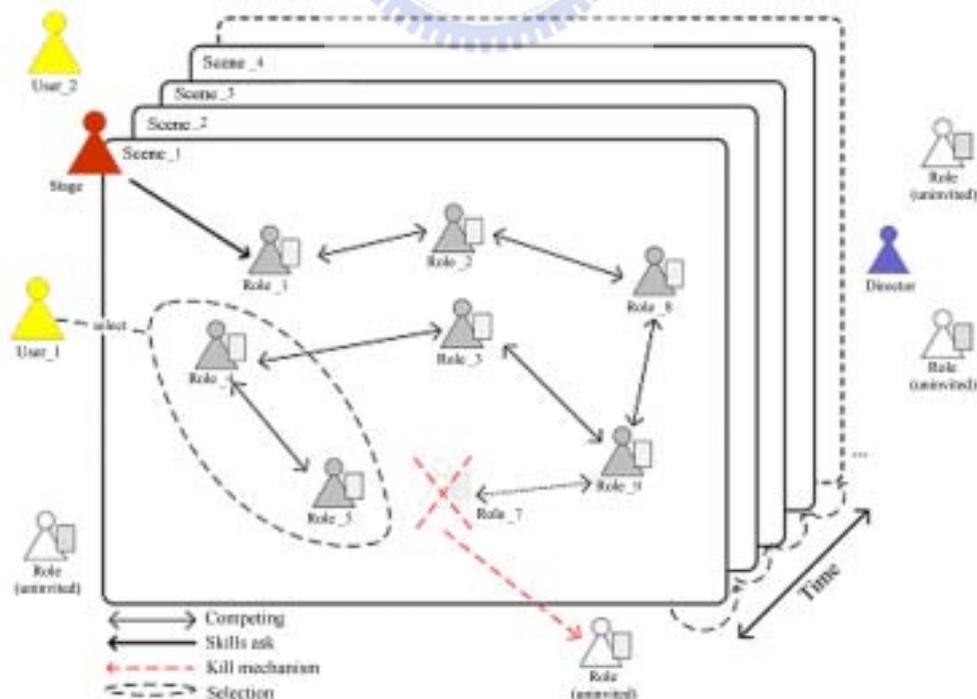


圖 41. DIM 角色的競爭學習

4.5 小結

DIM 是一個輔助分散式想法連結的運算模型，此模型提供多重參與者在分散式的設計環境中，可以不受地理位置與時間條件的限制下，彼此共同合作而連結與產生關聯性設計想法。在 DIM 中，一個設計工作（有關於產生設計想法）可以藉由 DIM 的五個元素予以分散化，這五個元素包括角色、演員、舞台、場景與腳本，在這五個元素中，角色（包括場記）、演員、舞台和場景可以被視為知識實體，其經由不同的相互作用而產生設計想法，而腳本則是一場分散式想法連結的內容敘述，其功能是定義這些不同相互作用的文本。此外，「知識」、「原則」與「連結過程」賦予這些元素相關的設計知識，以提供它們經由彼此間的相互作用，而進行分散式的想法連結。

基本上，「知識」提供 DIM 元素擁有關聯性的設計知識，和呈現這些設計知識與其記憶組織方式。知識的來源是早期設計案例，其基本單元為想法元件 (ie) 與連結 ($—$)，想法元件包含議題 (i)、概念 (c)、形式 (f) 與喜好 (p) 四個屬性，並藉由不同的媒材（領域概念語彙與多媒體）表現這四個屬性之值，而連結包含兩種類型的連結：屬性連結 ($—_L$) 和元件連結 ($—_E$)，每一種類型連結都由三種關係（相似性、對比性與相鄰性）的連結所構成。想法元件、屬性與不同類型連結的三種連結形式，共同建構 DIM 不同類型的想法地圖，這些想法地圖賦予不同 DIM 元素的記憶組織。

DIM 主要包括 IE 地圖與 ICF 地圖兩種型態的想法地圖。IE 地圖由想法元件與元件連結的三種連結形式 ($—_{SI}$, $—_{CR}$, $—_{CI}$) 所構成，而為了賦予並區別記憶組織在不同 DIM 元素的功能，IE 地圖包括角色 IE 地圖 (Map_{IE_R}) (即角色的記憶組織)、內部 IE 地圖 (Map_{IE_I}) (即場記的知識庫) 與外部 IE 地圖 (Map_{IE_E}) (即場景與舞台的知識庫)。ICF 地圖 (Map_{ICF}) 則是由三種屬性 (i, c, f) 分別與屬性連結的三種連結形式 ($—_{si}$, $—_{cr}$, $—_{ci}$) 所共同構成，其包括議題地圖 (Map_I)、概念地圖 (Map_c) 與形式地圖 (Map_f)，這些 ICF 地圖提供角色進行不同屬性值的置換，以增加連結想法元件的多樣性。在分散式的設計環境中，當角色接收到其他角色產生之想法元件時，ICF 詞典 ($Dictionary_{ICF}$) 則提供此角色進行解譯這些想法元件中屬性值的領域概念語彙。

「原則」賦予角色進行想法連結的行為，此行為是經由反射性的回憶以連結並產生想法元件。基本上，原則主要包括三個步驟：解譯、連結（包括相似性連結、對比性連結與相鄰性連結）與選擇，每一個步驟由不同類型的事件 ($Event_n$) 所組成，以提供角色能根據不同的設計情境，採取適當的行為以產生想法元件，上述這些原則的步驟與事件被適當的組織區劃，並儲存在角色的知識庫。此外，容許限制與想法元件中的喜好值，分別提供角色指定解譯屬性值的接受度與選擇想法

元件的優先順序。最後，這些角色產生的想法元件與連結，會在其記憶組織中進行不同想法地圖（ Map_{IE_R} 和 Map_{ICF} ）的轉化，同時，場記與場景分別在內部及外部相互作用中，立即性的呈現設計結果（分別是 Map_{IE_I} 和 Map_{IE_E} ）。由於 DIM 強調關聯性的知識建構與連結，因此，上面論述的「知識」與「原則」可以藉由 Wilson[78] 的圖像理論（Graph Theory）予以描述（詳見第 94, 95, 96 與 97 頁）。

另一方面，「連結過程」支持 DIM 元素在分散式的設計環境中，經由這些元素在內部或外部相互作用中的訊息傳遞，以連結具有關聯性的想法元件，進而產生設計結果（包括內部 IE 地圖與外部 IE 地圖）。基本上，演員（包括導演與使用者）角色與場記是在內部相互作用中進行訊息傳遞，而舞台、場景和演員（包括導演與使用者）是在外部相互作用中進行訊息傳遞，而這些訊息傳遞是建立在連結溝通語言（LCL）的機制上。連結溝通語言主要由發送者（ s ）、接收者（ r ）、語言行動（ la ）、訊息內容（ lc ）、訊息語言（ ll ）與知識本體論（ lo ）等屬性所構成，而其中不同類型的語言行動關係著這些元素對於接收訊息所採取的行動與反應，而訊息內容與傳遞訊息過程則是建立 DIM 腳本的主要內容敘述。

此外，不同角色由於主觀意見的差異而產生的衝突性，會引發這些角色進行彼此之間的競爭學習（包括淘汰與存活），同時，這些角色也經由演化學習而調整其原則中之解譯的容許限制，進而改變它們的記憶組織。上述這些 DIM 模型的運算機制將被具體執行並發展成一個輔助設計工具，以提供多重參與者可以在分散式的設計環境中進行想法連結。

Definition 1: Idea Entity

$$ie = (i, c, f, p)$$

$IE = \{(i, c, f, p) \mid i \in I, c \in C, f \in F, p \in P, r = 1 \sim n\}$, where i , c , f and p designate an issue, concept, a form and preference respectively.

$I = \{i_1, i_2, i_3, \dots, i_n\}$, where i is a keyword.

$C = \{c_1, c_2, c_3, \dots, c_n\}$, where c is a set of keywords.

$F = \{f_1, f_2, f_3, \dots, f_n\}$, where f is a keyword associated with multi-media.

$P = \{1, 2, 3, 4, 5\}$, where p is an integer.

Definition 2: Link

$\rightarrow = \{\underline{l}, \underline{L}\}$, where \underline{l} and \underline{L} are the non-dircectinal links between attributes and between ie respectively.

$\underline{l} = \{\underline{si}, \underline{cr}, \underline{ci}\}$, where \underline{si} , \underline{cr} and \underline{ci} designate a similarity link, a contrast link and a contiguity link respectively. These links exist between two attributes which are the same type of attribute.

$\underline{L} = \{\underline{SI}, \underline{CR}, \underline{CI}\}$, where \underline{SI} , \underline{CR} and \underline{CI} designate a similarity link, a contrast link and a contiguity link respectively. These links, which includes two types of \underline{l} , exist between two ie .

$$\underline{SI} = \{x \mid x = (i_a \xrightarrow{si} i_b) \cap (c_a \xrightarrow{si} c_b) \text{ or } x = (i_a \xrightarrow{si} i_b) \cap (f_a \xrightarrow{si} f_b)\}$$

$$\underline{CR} = \{x \mid x = (i_a \xrightarrow{si} i_b) \cap (c_a \xrightarrow{cr} c_b) \text{ or } x = (i_a \xrightarrow{si} i_b) \cap (f_a \xrightarrow{cr} f_b)\}$$

$$\underline{CI} = \{x \mid x = (c_a \xrightarrow{si} c_b) \cap (f_a \xrightarrow{si} f_b)\}$$

Definition 3: Idea Maps

There are two types of Idea Maps in DIM-2: Map_{ie} and Map_{cr}

$Map_{ie} = (IE, L)$ consist of

- 1) a finite set $IE = \{ie_1, ie_2, ie_3, \dots, ie_n\}$, whose elements are called idea entities, and
- 2) a subset L of the Cartesian product $IE \times IE$, the elements of which are called links

$Map_{ie} = (IE, L)$ determines a binary linking \underline{L} on the set IE of its idea entities , through the rules

$$ie_i \xrightarrow{L} ie_j \text{ if and only if } (ie_i, ie_j) \in L$$

$$L = \{(ie_i, ie_j) \in IE \times IE \mid ie_i \xrightarrow{L} ie_j \text{ and } ie_i \neq ie_j\}$$

Three types of Map_{ne} supports the memory organizations of different DIM-2 components.

$Map_{ne} = \{Map_{ne-r}, Map_{ne-l}, Map_{ne-e}\}$, where Map_{ne-r} is for a role, Map_{ne-l} is for a script holder, and Map_{ne-e} is for a scene and stage.

$Map_{scr} = \{Map_l, Map_c, Map_f\}$, where Map_{scr} includes Map_l , Map_c and Map_f , which are stored in the memory organization of a role.

$Map_l = (I, l)$ consist of

- 1) a finite set $I = \{i_1, i_2, i_3, \dots, i_n\}$, whose elements are called issue, and
- 2) a subset l of the Cartesian product $I \times I$, the elements of which are called links

$Map_l = (I, l)$ determines a binary linking \xrightarrow{l} on the set I through the rules

$i_i \xrightarrow{l} i_j$ if and only if $(i_i, i_j) \in l$

$l = \{(i_i, i_j) \in I \times I \mid i_i \xrightarrow{l} i_j \text{ and } i_i \neq i_j\}$

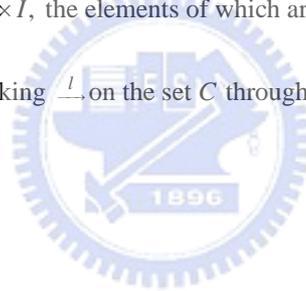
$Map_c = (C, l)$ consist of

- 1) a finite set $C = \{c_1, c_2, c_3, \dots, c_n\}$, whose elements are called concept, and
- 2) a subset l of the crocross product $I \times I$, the elements of which are called links

$Map_c = (C, l)$ determines a binary linking \xrightarrow{l} on the set C through the rules

$c_i \xrightarrow{l} c_j$ if and only if $(c_i, c_j) \in l$

$l = \{(c_i, c_j) \in C \times C \mid c_i \xrightarrow{l} c_j \text{ and } c_i \neq c_j\}$



$Map_f = (F, l)$ consist of

- 1) a finite set $F = \{f_1, f_2, f_3, \dots, f_n\}$, whose elements are called concept, and
- 2) a subset l of the crocross product $F \times F$, the elements of which are called links

$Map_f = (F, l)$ determines a binary linking \xrightarrow{l} on the set F through the rules

$f_i \xrightarrow{l} f_j$ if and only if $(f_i, f_j) \in l$

$l = \{(f_i, f_j) \in F \times F \mid f_i \xrightarrow{l} f_j \text{ and } f_i \neq f_j\}$

Definition 4: ICF Dictionary

$Dictionary_{scr}$ includes different sets of two similar domain conceptual vocabularies.

$Dictionary_{scr} = \{(a, b_r) \mid a \in A, b_r \in B, \text{ and } r = 1 \sim n\}$, where a and b designate an external domain and an internal domain conceptual vocabulary respectively.

$A = \{a_1, a_2, a_3, \dots, a_n\}$, where a is a keyword.

$B = \{b_1, b_2, b_3, \dots, b_n\}$, where b is a keyword

Definition 5: Events

In DIM-2, there are different types of events within Principle for roles to link and generate ie dynamically. The mathematical expressions of these events are described below:

$Event_{t_1}$

$$\left(\text{if } Map_e = (ie_b \xrightarrow{SI} ie_c) \text{ and} \right. \\ \left. temp_i_a = i_b \cap temp_c_a = c_b \cap temp_f_a = f_b \text{ then } temp_ie_a \xrightarrow{SI} ie_c \right)$$

$Event_{t_2}$

$$\left(\text{if } temp_i_a = i_b \cap temp_c_a = c_b \cap temp_f_a \neq f_b \text{ then } temp_ie_a \xrightarrow{SI} ie_b \right) \\ \left(\text{if } temp_i_a = i_b \cap temp_c_a \neq c_b \cap temp_f_a = f_b \text{ then } temp_ie_a \xrightarrow{SI} ie_b \right)$$

$Event_{t_3}$

$$\left(\text{if } temp_i_a = i_b \cap temp_c_a \neq c_b \cap temp_f_a \neq f_b \text{ and } temp_c_a \xrightarrow{si} c_b \right) \\ \left(\text{then } temp_ie_a \xrightarrow{SI} ie_b \text{ else return false} \right) \\ \left(\text{if } temp_i_a \neq i_b \cap temp_c_a = c_b \cap temp_f_a \neq f_b \text{ and } temp_i_a \xrightarrow{si} i_b \right) \\ \left(\text{then } temp_ie_a \xrightarrow{SI} ie_b \text{ else return false} \right) \\ \left(\text{if } temp_i_a \neq i_b \cap temp_c_a \neq c_b \cap temp_f_a = f_b \text{ and } temp_i_a \xrightarrow{si} i_b \right) \\ \left(\text{then } temp_ie_a \xrightarrow{SI} ie_b \text{ else return false} \right) \\ \left(\text{if } temp_i_a = i_b \cap temp_c_a \neq c_b \cap temp_f_a \neq f_b \text{ and } temp_f_a \xrightarrow{si} f_b \right) \\ \left(\text{then } temp_ie_a \xrightarrow{SI} ie_b \text{ else return false} \right)$$

$Event_{t_4}$

$$\left(\text{if } temp_i_a \neq i_b \cap temp_c_a \neq c_b \cap temp_f_a \neq f_b \text{ and } temp_i_a \xrightarrow{si} i_b \cap temp_c_a \xrightarrow{si} c_b \right) \\ \left(\text{then } temp_ie_a \xrightarrow{SI} ie_b \text{ else return false} \right) \\ \left(\text{if } temp_i_a \neq i_b \cap temp_c_a \neq c_b \cap temp_f_a \neq f_b \text{ and } temp_i_a \xrightarrow{si} i_b \cap temp_f_a \xrightarrow{si} f_b \right) \\ \left(\text{then } temp_ie_a \xrightarrow{SI} ie_b \text{ else return false} \right)$$

$Event_{t_5}$

$$\left(\text{if } Map_e = (ie_b \xrightarrow{CR} ie_c) \text{ and} \right. \\ \left. temp_i_a = i_b \cap temp_c_a = c_b \cap temp_f_a = f_b \text{ then } temp_ie_a \xrightarrow{CR} ie_c \right)$$

$Event_{t_6}$

$$\left(\text{if } temp_i_a = i_b \cap c_a \text{ contrast} = c_b \cap temp_f_a \neq f_b \right) \\ \left(\text{then } temp_ie_a \xrightarrow{CR} ie_b \right) \\ \left(\text{if } temp_i_a = i_b \cap temp_c_a \neq c_b \cap f_a \text{ contrast} = f_b \right) \\ \left(\text{then } temp_ie_a \xrightarrow{CR} ie_b \right)$$

$Event_{t_7}$

$$\left(\text{if } temp_i_a \neq i_b \cap \rho_{cs} temp_c_a = c_b \cap temp_f_a \neq f_b \text{ and } temp_i_a \xrightarrow{si} i_b \right) \\ \left(\text{then } temp_ie_a \xrightarrow{CR} ie_b \text{ else return false} \right)$$

$$\left(\begin{array}{l} \text{if } temp_i_a \neq i_b \wedge temp_c_a \neq c_b \wedge \rho_a temp_f_a = f_b \text{ and } temp_i_a \xrightarrow{si} i_b \\ \text{then } temp_ie_a \xrightarrow{CR} ie_b \text{ else return false} \end{array} \right)$$

Event_{t₈}

$$\left(\begin{array}{l} \text{if } Map_{\#} = (ie_b \xrightarrow{CI} ie_c) \text{ and } temp_i_a = i_b \wedge temp_c_a = c_b \wedge temp_f_a = f_b \\ \text{then } temp_ie_a \xrightarrow{CI} ie_c \end{array} \right)$$

Event_{t₉}

$$\left(\begin{array}{l} \text{if } temp_i_a \neq i_b \wedge temp_c_a = c_b \wedge temp_f_a = f_b \\ \text{then } temp_ie_a \xrightarrow{CI} ie_b \end{array} \right)$$

Event_{t₁₀}

$$\left(\begin{array}{l} \text{if } temp_i_a \neq i_b \wedge temp_c_a = c_b \wedge temp_f_a \neq f_b \text{ and } temp_f_a \xrightarrow{si} f_b \\ \text{then } temp_ie_a \xrightarrow{CI} ie_b \text{ else return false} \end{array} \right)$$

$$\left(\begin{array}{l} \text{if } temp_i_a \neq i_b \wedge temp_c_a \neq c_b \wedge temp_f_a = f_b \text{ and } temp_c_a \xrightarrow{si} c_b \\ \text{then } temp_ie_a \xrightarrow{CI} ie_b \text{ else return false} \end{array} \right)$$

Event_{t₁₁}

$$\left(\begin{array}{l} \text{if } temp_i_a \neq i_b \wedge temp_c_a \neq c_b \wedge temp_f_a \neq f_b \text{ and } temp_c_a \xrightarrow{si} c_b \wedge temp_f_a \xrightarrow{si} f_b \\ \text{then } temp_ie_a \xrightarrow{CI} ie_b \text{ else return false} \end{array} \right)$$

Event_{t₁₂}

$$\left(\begin{array}{l} \text{if } \rho_a temp_i_a = i_b \wedge temp_c_a = c_b \wedge temp_f_a = f_b \text{ then } temp_ie_a \xrightarrow{CI} ie_b \\ \text{else return false} \end{array} \right)$$

where *input_ie*, *output_ie*, *cand_ie* are input *ie*, output *ie* and candidate *ie* respectively. Also, *temp_ie*, *temp_i*, *temp_c*, *temp_f* are temporary *ie*, *i*, *c* and *f* respectively.

The Linking Communication Language (*LCL*) provides the communication mechanisms, including sending and receiving messages, among DIM-2 components. Through the *LCL*, these components can link *ie* and generate *Map_#* in the external and the internal interplays.

$LCL = \{(s, r, la, lc, ll, lo) \mid la \in LA \text{ and } lc \in LC\}$, where *s*, *r*, *la*, *lc*, *ll*, *lo* are sender, receiver speech acts, message content, language syntax and ontology respectively.

$$LA = \{la_{int_orm}, la_{request}, la_{request_when}, la_{ref}, la_{propose}, la_{failure}\}$$

$$LC = \{lc_{skill}, lc_{role}, lc_{issue}, lc_{ie}, lc_{map}, lc_{principle}, lc_{time}, lc_{number}\}$$

五、DIM 之系統執行與實例

在第三章有關輔助分散式連結系統的文獻回顧中，DARIS 是結合角色扮演理論的多重代理人環境，此環境反映 DIM 模型的分散性想法連結，有關其不同層級的相互作用與互動性的溝通機制，同時，DARIS 使用 FIPA 的代理人溝通語言，此語言所使用的 performatives 具有標準化特性，提供 DIM 不同元素（角色、演員、場景、舞台與腳本）在分散式的設計環境中，具有跨平台的開放度與相容性。上述的特性有助於具體化本研究建立的輔助分散式想法連結模型，因此，DIM 的系統執行主要架構在 DARIS 的多重代理人架構而建立。

在本章中，作者首先會對 DIM 系統如何架構在 DARIS 的方式進行說明，然後，再予以闡述 DIM 系統架構中的系統元素與系統技術。此外，根據 DIM 的系統架構，有關 DIM 系統環境的分散性運算機制，包括如代理人身分定義、生命週期、溝通、Yellow Page 服務也會在相關章節進行說明，以提供本研究實際執行 DIM 代理人（包括角色、場記、場景與舞台代理人）的基礎。最後，本研究會經由一個實際的案例說明 DIM 代理人的知識獲得與建構，和此系統實際的操作與應用。

5.1 系統架構

根據 DARIS 中三個系統層與五種代理人的系統架構中，DIM 架構在 DARIS 的方式，主要經由匹配（mapping）DIM 元素與 DARIS 代理人，同時，在 DARIS 的三個系統層級，適當地放置 DIM 元素並賦予相關的知識基礎而達成。在匹配 DIM 元素與 DARIS 代理人中，DIM 中的五個元素分別被匹配到 DARIS 的五種類型代理人（表 8），例如：在表 8 中，DIM 的場景元素匹配於 DARIS 的場景代理人，導演元素匹配於導演代理人（DA）。

表 8 . DARIS 代理人與 DIM 元素匹配之關係圖表

DARIS 代理人	DIM 元素
導演代理人 (DA)	導演
使用者代理人 (UA)	使用者
角色代理人 (RA)	角色、場記
舞台代理人 (StA)	舞台
場景代理人 (ScA)	場景

進而，這些經過匹配的 DIM 元素被分別放置在 DARIS 的三個系統層級（詳見

3.2.4), 並賦予它們相關的知識基礎(表 9)。在 DARIS 的內在設計情境層放置角色與場記, 並賦予它們知識、原則與連結過程中的溝通機制, 而在外在設計情境層放置舞台和場景, 且給予它們連結過程中的溝通機制, 此外, DARIS 的溝通層除了提供 DIM 元素之間訊息傳遞的頻道(channel)外, 同時提供演員(導演與使用者)在內在與外在系統層級中, 控制其它相關的 DIM 元素。而這些訊息傳遞過程的相關訊息內容與代理人描述, 都經由腳本的編輯而建立。

表 9. DARIS 三個系統層級與 DIM 元素之關係圖表

DARIS 系統層級	DIM 元素
內在設計情境層	角色、場記
溝通層	導演、使用者(提供 DIM 代理人溝通頻道)
外在設計情境層	舞台、場景

這些經過匹配並放置在三個系統層級的 DIM 元素稱為代理人, 因此, 導演、使用者、角色、場記、舞台和場景, 在 DIM 系統中, 分別被稱為導演代理人(DA)、使用者代理人(UA)、角色代理人(RA)、場記代理人(RAS)、舞台代理人(StA)和場景代理人(ScA)。此外, UA 和 DA 是參與一個分散式想法連結的參與者, 而這些參與者可以是人類或電腦。

跨平台的資訊溝通是 DIM 系統在進行分散式想法連結的主要關鍵, 由於, JAVA 程式語言擁有跨平台的優勢, 因此, DIM 之系統技術, 除了延用 DARIS 以 JAVA 為基礎的 JADE 代理人環境外, 並使用可以與 JADE 整合的相關語言程式(JESS)和軟體(GraphVIZ), 以具體化 DIM 不同代理人的知識基礎, 並建立一個跨平台的分散式代理人溝通環境。

5.1.1 系統元素

除了將 DIM 元素架構在 DARIS 的多重代理人環境外, 為了提供 DIM 系統能反射性的連結並產生想法元件與連結, 且透過訊息傳遞的過程, 建立分散式設計知識之間的關聯性與衝突性, 並立即性的呈現設計結果, DIM 的系統架構主要包括三個系統元素: 知識視覺化元素、知識基礎元素與溝通元素(圖 42), 這些系統元素詳細說明如下:

1. 知識視覺化元素: 知識視覺化元素賦予 DIM 代理人(RAS 和 ScA), 將設計結果(內部 IE 地圖和外部 IE 地圖)經由知識視覺化而呈現。這些視覺化的知識圖像, 提供參與者(UA 和 DA)在連結想法的過程中, 可以看見內在與外

在相互作用的連結關係，以利想法元件的選取與組合。

2. 知識基礎元素：知識基礎元素提供 DIM 代理人之知識基礎，包括在不同代理人 (RA、RAS、StA 和 ScA) 的知識庫，存取不同種類的想法地圖，同時，建立 RA、RAS 和 StA 連結想法的相關規則。簡言之，知識基礎元素賦予代理人存取和產生想法元件和連結之能力。
3. 溝通元素：在進行連結過程之相互作用時，溝通元素提供所有 DIM 代理人訊息傳遞的相關機制，主要包括代理人之間訊息傳遞的規則，與代理人溝通語言 (ACL) 的相關機制。簡言之，溝通元素賦予 DIM 代理人進行溝通的能力。

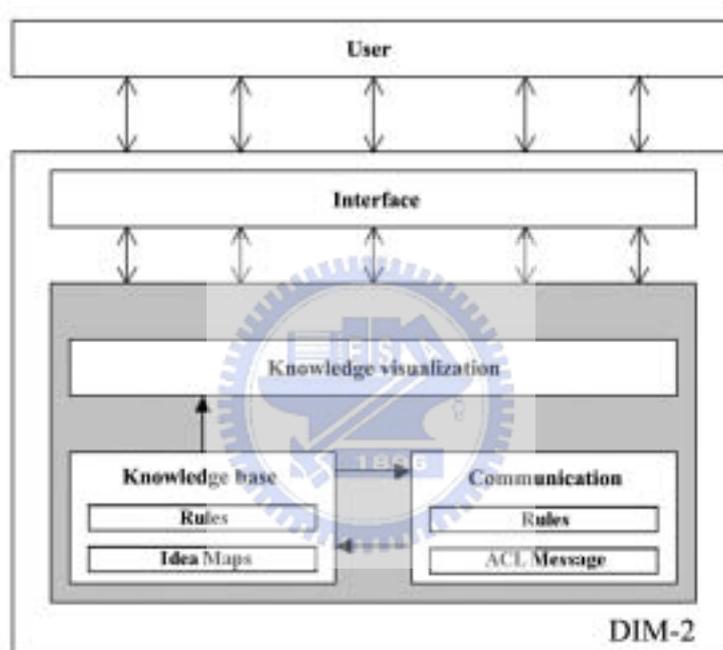


圖 42. DIM 的系統架構

5.1.2 系統技術

為了有效結合原有 DARIS 的系統技術，在這些 DIM 的系統元素中，溝通元素被執行在以 JAVA 為語言基礎之 JADE 的代理人環境中，同時，使用 FIPA 為代理人主要的代理人溝通語言，而知識基礎元素被執行在 JESS 的專家系統語言，提供相關代理人的系統邏輯推理與知識庫儲存，因此，DIM 系統是由一群結合 JESS 推理機制的代理人，在 JADE 的代理人環境中，進行分散式的想法連結。另外，DIM 的知識視覺化元素則藉由以 JAVA 為語言基礎的圖像視覺化 (graph visualization) 軟體 (GraphVIZ)，立即性的表現設計結果 (包括內部 IE 地圖與外部 IE 地圖)。

參與者在不同地理位置進行訊息傳遞，主要經由網際網路的 HTTP 訊息通訊協定

(transport protocol), 予以傳遞相關的訊息內容。此外, 每一個想法元件中形式屬性的多媒體資訊, 皆被儲存在一個集中的伺服器 (server) 中, 參與者可以藉由網路超連結 (hyper-link) 的方式, 表現每一個想法元件中的多媒體資訊。

5.2 系統環境

在 DIM 之系統環境中, 代理人必須“存活 (live)”在 JADE 的 runtime 環境¹中, 才能參與此分散式的想法連結。因此, 參與者 (包括 UA 與 DA) 必須具有擁有各自的代理人平台, 才可以將他們其他的代理人 (包括如 StA、ScA、與 RA 等), 放置在這些代理人平台內的容器中, 以進行此分散式的想法連結。由於 DA 必須控制整個外部相互作用的情境, 因此, StA 與 ScA 人只能“存活”在 DA 之代理人平台的容器中, 而 RA 與 RAS 則根據不同的參與者希望扮演的角色, 被分別放置在這些參與者的 runtime 環境中。此外, 每一個參與者之代理人平台, 除了包含一個既有的主容器 (Main container) 外, 參與者也可創造其他的容器, 並在這些不同容器中放置代理人。由於, 本研究強調不同知識實體之間的分散式連結關係, 因此, 不同容器之間的連結關係在此階段的系統執行暫時不予考慮。

在進行分散式的想法連結過程, DIM 系統提供參與者在區域性 (local) 的單一代理人平台中, 與其他代理人進行想法的連結, 同時, DIM 也提供參與者在全域性 (global) 的不同代理人平台中, 透過網路 HTTP 的訊息通訊協定方式, 與其他參與者之代理人平台的代理人進行想法的連結。在 DIM 的系統環境中, 每一個代理人平台的主容器包含兩個 JADE 內含的代理人 AMS (*Agent Management System*) 和 DF (*Directory Facilitator*), 其負責多重代理人在代理人平台中進行溝通的相關服務。

基本上, AMS 主要提供代理人身分 (agent identifier, 簡稱 AID) 命名 (naming) 的服務, 以確保每一位代理人在此代理人平台中有唯一 (unique) 的名字, 同時也負責代理人在此代理人平台中的生命週期 (包括如存活、行動、死亡)。此外, 根據這些代理人的 AID, DF 提供黃頁 (yellow page) 的服務, 此服務是支持一個代理人可以在不同的代理人平台中, 找尋其他相關的代理人以進行想法連結工作, 而為了顯示代理人存活在代理人平台中的相關資訊, RMA (*Remote Monitor Agent*) 負責將這些資訊經由使用者介面呈現給使用者, 以便使用者進行溝通與控制。因此, 一個 DIM 系統的分散式想法連結環境可以見圖 43。

¹Runtime 環境係指包含在代理人平台 (Platform) 中的“容器” (Container), 而容器是放置使用者載入代理人的空間。基本上, 每一台電腦只有一個代理人平台, 而一個代理人平台除了包含一個預先設定的容器, 稱之為主容器 (Main Container), 同時, 使用者可以根據自己的需求在此代理人平台中增加其他的容器。

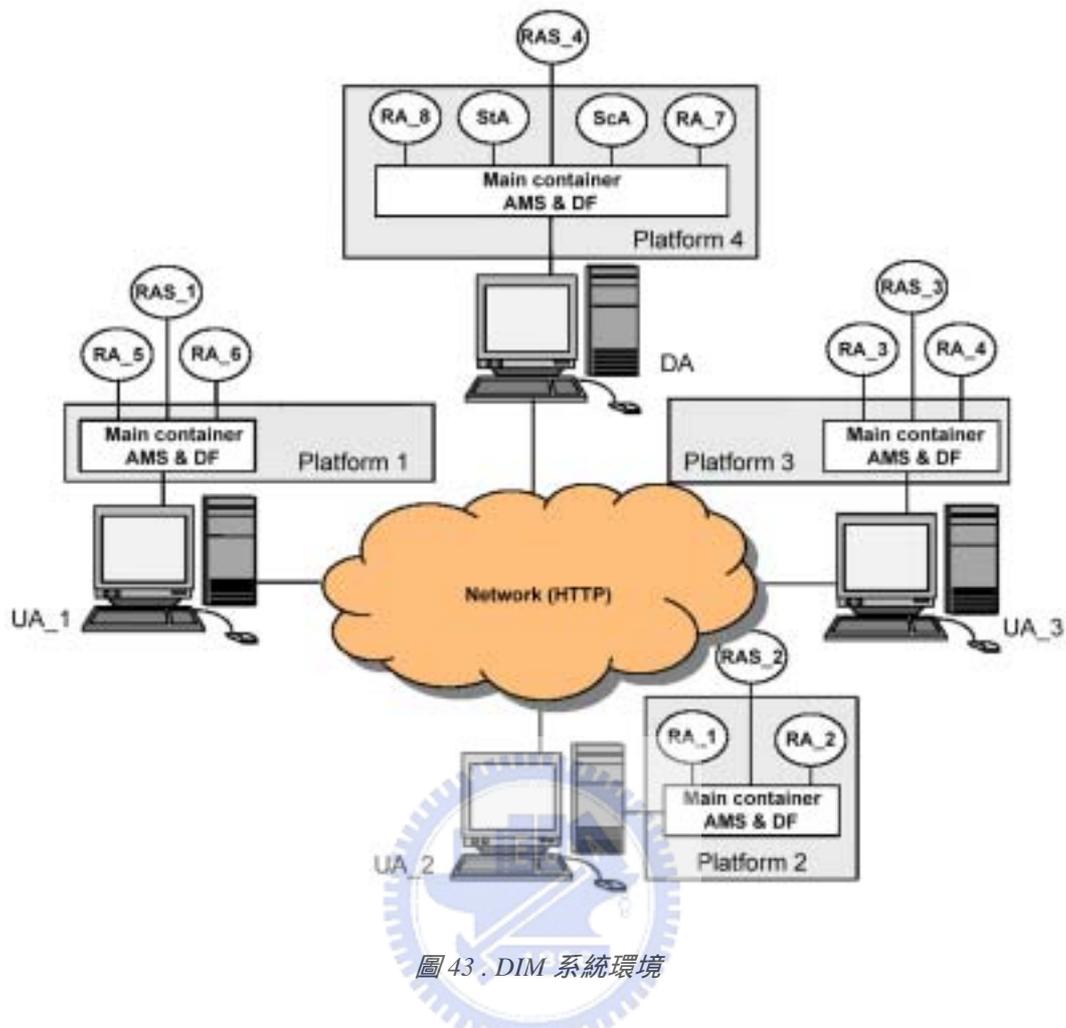


圖 43 . DIM 系統環境

在圖 43 的例子中，DA 與三位 UA (UA_1、UA_2 和 UA_3) 參與一個分散式的想法連結工作。在此 DIM 的系統環境中，除了每一個代理人平台之 Main container 包含個自的 AMS 與 DF 外，在 DA 的代理人平台 (Platform 4) 中，還包括 StA、ScA、RAS_4、RA_7 和 RA_8；在 UA_1 的代理人平台 (Platform 1) 中，還包括 RAS_1、RA_5 和 RA_6；在 UA_2 的代理人平台 (Platform 2) 中，還包括 RAS_2、RA_1 和 RA_2；在 UA_3 的代理人平台 (Platform 3) 中，還包括 RAS_3、RA_3 和 RA_4。這些不同代理人平台中的代理人，藉由網路的 HTTP 訊息通訊協定，彼此傳遞相關的訊息內容予以產生設計結果。

為了更進一步說明 DIM 系統環境中相關的分散式運算機制，以利本研究執行 DIM 代理人，有關代理人的身分定義、生命週期、行為、溝通機制與 Yellow Pages 服務必須先加以說明。基本上，身分定義提供多重代理人彼此認識而進行溝通，生命週期則是建立代理人在想法連結過程中 runtime 環境之狀態 (如存活、死亡等)，而溝通提供多重代理人之間的訊息傳遞機制，另外，代理人的 Yellow Pages 服務允許代理人搜尋適當服務內容的代理人，以便進行分散式想法連結的工作。

5.2.1 代理人身分定義

在DIM的系統環境中，所有參與分散式想法連結的DIM代理人(除了DA和UA外)，都需藉由繼承jade.core.Agent class和setup()的method而產生。

```
public class RA extends Agent {
    protected void setup() {
    }
}
```

為了進行分散式的想法連結，這些DIM中的代理人必須同時存活在不同代理人平台的容器中，同時，每一個代理人必須藉由jade.core.AID class定義代理人的AID而彼此認識，以利代理人彼此之間的訊息傳遞。基本上，一個AID物件(object)包含一個全域性且唯一性的名字和住址，主要以<nickname>@<platform-name>表示之，例如：有一個RA的nickname為RA1，此代理人存活在一個platform-name為lai:1099/jade的代理人平台中(1099為預先設定的RMI連接埠)，因此，此角色代理人的AID為RA1@ lai:1099/jade。進而，此代理人的AID可以從下列的語法而執行：

```
AID id = new AID (RA1@ lai:1099/jade);
```

5.2.2 代理人生命週期

所謂代理人生命週期是代理人在 runtime 環境中的啟始、終結與其它行為的執行序列[156]。由於，在分散式的想法連結過程中，代理人會根據不同的設計情境，而採取適當行為的執行序列以進行不同的相互作用，因此，代理人生命週期在想法連結過程中是一個必須被考慮的因素。基本上，代理人的生命週期建立在執行序列的三個階段：代理人起始階段、行為執行階段與終結階段，而這些執行序列主要藉由 setup()、doDelete()、action()、done()和 takedown()而達成(圖 44)。

在起始階段中，代理人必須被啟動而存活在一個代理人平台的容器中，並開始與其他代理人進行相互作用，然而，在終結階段，代理人被終結而消失在此代理人平台的容器中，且不能與其他代理人進行相互作用。另外，在行為執行階段，DIM必須先了解代理人(以 RA_1 為例)是否已經被告知終結(kill)在此環境中，如果答案為肯定(Yes)，則 RA_1 將清除其行為並退場；如果答案為否定(No)，則 RA_1 會從自己的啟動行為池(pool of active behavior)中得到下一個行為予以執行，如果執行完成時，則 RA_1 會從這個啟動行為池中移除這個行為(例如：RA_1 的接收訊息行為執行完成後，會從啟動行為池移除這個接收行為，而自動的得到

下一個連結原則的行為，予以進行想法元件的連結)。上述的流程會被的重複進行，直到 RA_1 被使用者 (UA 或 DA) 告知終結並清除這些行為。

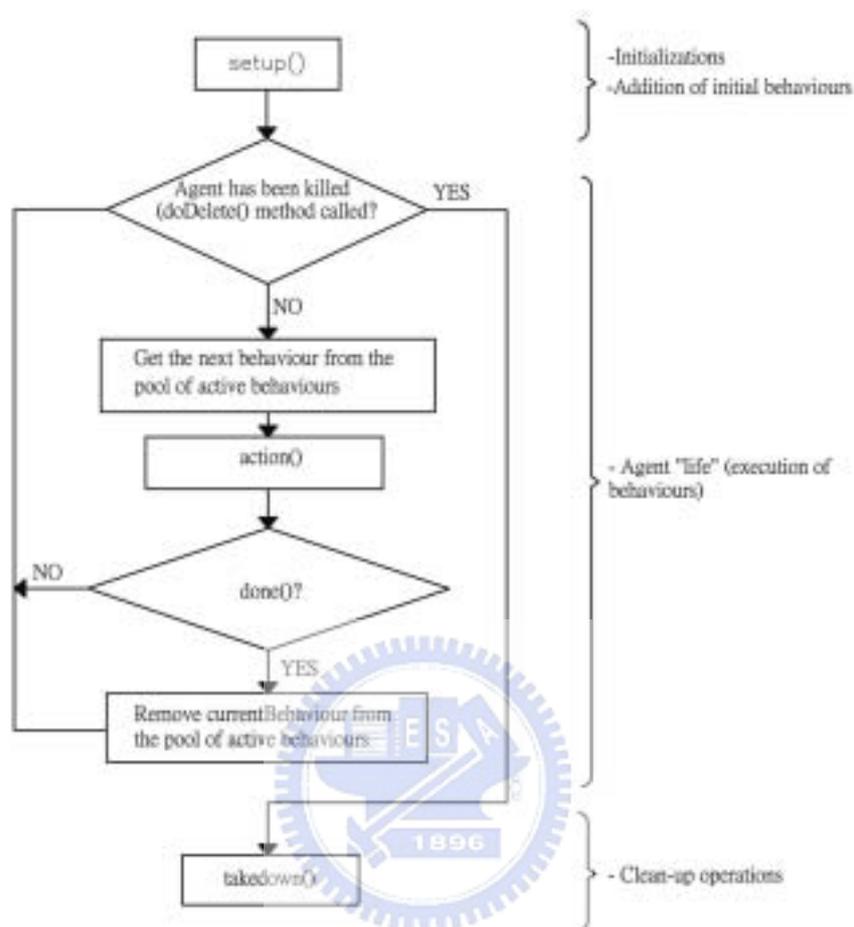


圖 44. 一個 DIM 代理人的生命週期

5.2.3 代理人溝通

在 DIM 的系統環境中，這些代理人的啟始與終結和他們執行行為的序列，主要關係於代理人在訊息傳遞過程中的溝通機制。基本上，DIM 的溝通機制為非同步性的訊息傳遞 (asynchronous message passing)，因此，每一個代理人有一個信箱，或稱代理人訊息序列 (the agent message queue)，此信箱為放置其它代理人所寄出的訊息，當一個訊息被投遞在一個代理人的訊息序列時，此接收的代理人會被通知而進行此訊息內容的相關行為 (圖 45)。

另外，DIM 元素在訊息傳遞中所使用的連結溝通語言 (LCL)，在此系統環境中，必須建立在 FIPA 所定義的代理人溝通語言 (ACL) 的標準。基本上，ACL 主要包括下列的空格 (field): 1) 訊息的發送者 (sender) 2) 訊息的接收者 (receiver) 3) 溝通意圖 (communicative-act) 4) 訊息內容 (content) 5) 訊息內容語言

(*language*) 與 6)知識實體論 (*ontology*), 而這些空格可以分別對應 DIM 之 LCL 元素 (詳見 4.4.3 小節), 例如: LCL 的語言行動對應於 ACL 的溝通意圖。

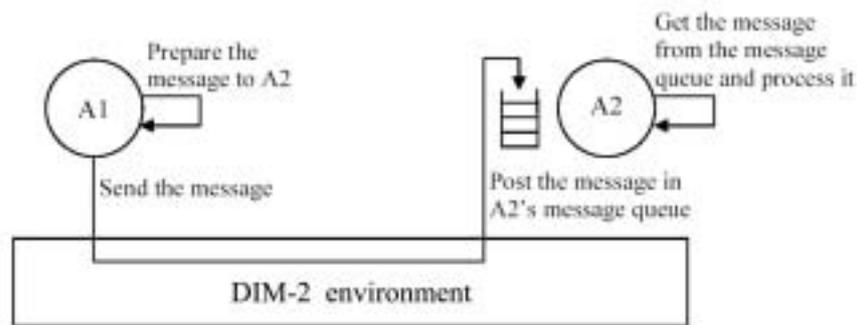


圖 45. DIM 系統代理人之溝通機制

因此, 一個在 DIM 系統的訊息必須包含相關的 ACL 欄位 (slot), 並藉由 `jade.lang.acl.ACLMessage` 的 class 執行成一個物件 (objects), 予以提供代理人進行訊息傳遞的相關行為 (包括訊息的接收與發送) 架構在 JESS 的機制中, 這些 ACL 的空格可以用 JESS 的樣板 (template) 加以定義, 此樣板名稱是 `ACLMessage`, 其包含了上述 ACL 空格的不同欄位, 這些欄位主要提供 DIM 代理人在訊息傳遞的知識基礎。(有關 JESS 機制的樣板名稱、欄位在 5.3 章節再予以說明)

```
(deftemplate ACLMessage
  (slot communicative-act) (slot sender) (multislot receiver)
  (slot language) (slot ontology) (slot content)
)
```

除了上述欄位外, `ACLMessage` 樣板也包含其他的欄位 (如 `protocol`、`envelope`、`reply-with`、`conversation-id` 等), 有助於 DIM 系統的延伸, 並處理 DIM 代理人進行複雜的溝通機制, 包括同時間進行許多對話的控制, 或有關超時設定的訊息接收等問題, 有關這些溝通機制與分散式想法連結的關係將為未來的後續研究。

5.2.4 代理人 Yellow Pages 服務

為了提供代理人之間在進行分散式的想法連結過程中, 可以找尋到具有相關需求的代理人進行想法連結 (例如: 場景代理人找尋具有相關設計技能的角色代理人), DIM 提供 Yellow Pages 的服務, 允許代理人將它們的服務內容公佈, 以便讓其他代理人發現它們。基本上, Yellow Pages 服務包含公佈代理人提供的服務內容與搜尋所需服務內容的代理人。

在 DIM 系統中, 服務內容係指角色代理人的設計技能。角色代理人藉由發佈自己

提供設計技能的服務，提供場景代理人根據不同的議題，搜尋到具有相關設計技能的角色代理人，以進行此分散式的想法連結（圖 46）。上述的公佈與搜尋服務，主要藉由使用者代理人與導演代理人之代理人平台的 DF 代理人負責執行。

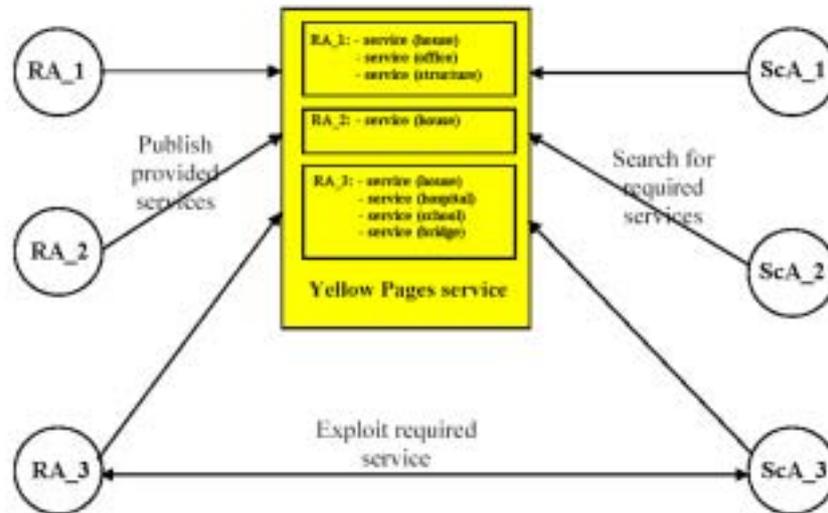


圖 46. DIM 的 Yellow Pages 服務

為了公佈服務內容，代理人必須提供適當服務內容的描述（如服務類型、服務名稱），並藉由 DFService class 中的 register() 的方法予以執行。例如：一個角色代理人具有住宅型態之設計技能，並從事想法元件連結的設計工作，此角色代理人的服務類型為“house”，服務名稱為“linking-ideas”，此語法的定義如下：

```
protected void setup() {
    ...
    DFAgentDescription dfd = new DFAgentDescription();
    dfd.setName(getAID());
    ServiceDescription sd = new ServiceDescription();
    sd.setType("house");
    sd.setName("linking-ideas");
    dfd.addServices(sd);
    try {
        DFService.register(this, dfd);
    }
    ...
}
```

另外，一個代理人如果要想找尋相關服務內容的其他代理人，此代理人必須向 DF 提

供相關服務內容的描述（如服務類型），藉由與上述代理人公佈服務內容的描述相互比對，進行具有此服務內容代理人的搜尋，此搜尋機制由 DFService class 中的 search()的方法予以執行。例如：一位場景代理人希望在不同代理人平台中，尋找具有設計技能和設計工作分別是“house”和“linking-ideas”的角色代理人，以進行分散式想法連結，其語法定義如下：

```
protected void setup() {  
    ...  
    DFAgentDescription template = new DFAgentDescription( );  
    ServiceDescription sd = new ServiceDescription( );  
    sd.setType("house");  
    sd.setType("linking-ideas");  
    template.addServices(sd);  
    try {  
        DFAgentDescription[ ] result = DFService.search(myAgent, template);  
        roleAgents = new AID[result.length];  
        for (int i = 0; i < result.length; ++i) {  
            roleAgents[i] = result.getName( );  
        }  
        ...  
    }  
}
```

5.3 代理人執行

為了有效結合以 JAVA 為基礎之 JADE 代理人環境，DIM 代理人（包括 RA、RAS、ScA 和 StA）的知識基礎與溝通機制，都是建構在以 JESS 為基礎的專家系統語言中，並分別儲存在不同檔案名稱的 CLP 格式中。而這些代理人的產生，主要經由 JADE 的 addBehaviour（），將這些代理人的 CLP 檔置放進來，以進行分散式的想法連結。例如：一個角色代理人（RA1）的知識基礎放置在 c:/DIM/RA1.clp 的檔案中，RA1 可以藉由繼承 jade.core.Agent class 中的 addBehaviour（）功能，創造 RA1 並賦予它相關行為，其語法如下：

```
public class RA1 extends Agent {  
    protected void setup() {  
        addBehaviour(new BasicJessBehaviour (this,"c:/DIM/RA1.clp",1));  
    }  
}
```

在 JESS 的機制中，資料結構是由許多不同的事實 (facts) 所組成，每個事實則是由樣板 (template) 所呈現，而一個樣板主要由一個樣版名稱與許多不同的欄位 (slots) 所組成，這些樣板與欄位構成 DIM 代理人主要的知識基礎。此外，JESS 是一種以規則基礎 (rule-based) 之系統架構的程式語言[157]，其系統架構主要包括工作記憶、規則基礎與推理引擎，工作記憶是將正在執行的相關資訊放置在推理引擎中，規則基礎則儲存推理引擎使用的規則，而推理引擎是經由控制使用的規則與這些規則在工作記憶的執行程序，予以產生系統的輸出。架構在上述這些 JESS 的相關機制，這些代理人之工作記憶、規則基礎與推理引擎，在下列章節有詳細的說明。

5.3.1 角色代理人

角色代理人 (RA) 為 DIM 系統主要的代理人。在內在設計情境層中，一個 RA 經由與 RAS、UA (或 DA) 和其他 RA 進行內部相互作用，反射性的連結並產生具關聯性的想法元件。架構在上述 JESS 的機制，RA 的知識基礎主要包括：1) 工作記憶 2) 規則基礎 3) IE 產生器(IE Generator)與 4) RA 知識庫(Role Repository) (圖 47)。

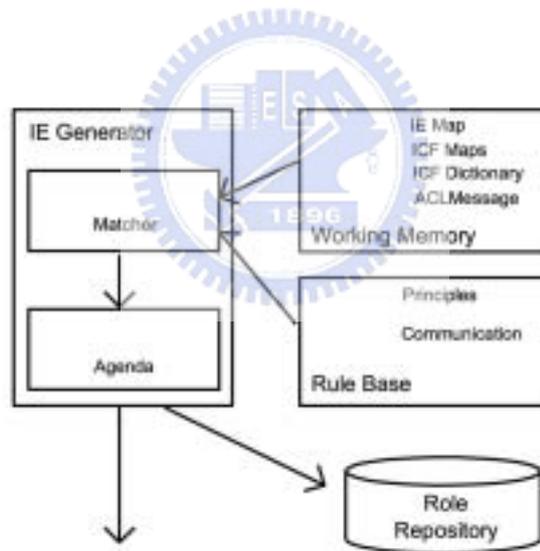


圖 47. 角色代理人的知識基礎元素

基本上，RA 的工作記憶主要放置 RA 正在執行想法連結的相關資訊 (包括如想法元件、ICF 地圖、ICF 詞典與角色 IE 地圖) 與傳遞訊息的相關資訊，規則基礎則是儲存 RA 連結想法的相關規則 (不同類型的事件)，且這些規則被適當的組織並區劃，而 IE 產生器 (包括 Matcher 和 Agenda) 控制整個規則基礎與工作記憶資訊的執行流程，另外，RA 知識庫則儲存所有 RA 的設計知識。

5.3.1.1 工作記憶

RA 的工作記憶主要包括角色 IE 地圖、三種不同的 ICF 地圖、ICF 詞典與想法元件，還有代理人溝通語言的相關資訊。除了 ICF 詞典是由目錄 (*list*) 所構成外，其它 RA 的工作記憶皆由不同種類的樣板所構成，每一種樣板根據其所屬的名稱與包含不同欄位而呈現 RA 的工作記憶，包括想法元件樣板、IE 地圖樣板、三種類型的 ICF 地圖樣板與代理人溝通語言樣板。

ICF 詞典目錄

ICF 詞典主要由許多詞典目錄所組成。基本上，詞典目錄由二個 tokens 所構成，第一個 token 是 RA 所能了解包含在 ICF 詞典之外部領域概念語彙的關鍵字，第二個 token 是 RA 所能了解包含在 ICF 詞典之內部領域概念語彙的關鍵字，同時此關鍵字必須存在於此 RA 的角色 IE 地圖，而這些外部與內部的關鍵字對於 RA 而言，彼此不相同但具有相類似的意義。這些詞典目錄的語法定義如下：

(movement circulation)

在上述這個目錄的例子中，“circulation”和“movement”是儲存在一位 RA 之 ICF 字典的詞典目錄中的關鍵字，對此 RA 而言，“circulation”和“movement”具有相同的意義，且能被此 RA 了解的領域概念語彙，而其中“circulation”的關鍵字須存在於此 RA 的角色 IE 地圖。

想法元件樣板

想法元件的樣板名稱是 ie，主要包含議題 (issue)、概念 (concept)、形式 (form) 與喜好 (preference) 四個欄位。議題、概念與形式的欄位值是關鍵字，而其中的概念與形式是多重欄位 (multislot)，可以放置一個以上的關鍵字，另外，喜好欄位值是 1 到 5 的整數值 (內定值為 1)。因此，想法元件樣板的語法定義如下：

```
(deftemplate ie
  (slot issue) (multislot concept) (multislot form) (slot preference (default 1))
)
```

例如：在一個 RA 的想法元件中，其議題值是“light”，概念值是“opening”和“centralization”，而形式值是“courtyard”，且形式值的設計案例被放置在 http://lai.cyut.edu.tw/DIM/RA1_1 的 URL，另外，RA 對於此想法元件的喜好值是 3。因此，此想法元件可以由下列的想法元件樣板加以定義。

```
(ie (issue light) (concept opening centralization)
  (form courtyard http://lai.cyut.edu.tw/DIM/RA1\_1) (preference 3)
)
```

IE 地圖樣板

IE 地圖的樣板名稱是 map，主要包含兩組想法元件的 ICF 欄位（issue、concept、form）和想法元件之間的元件連結欄位，因此，IE 地圖樣板包括七個欄位，這些欄位值皆為關鍵字，而其中的概念與形式是多重欄位，分別放置一個以上的關鍵字和形式的 URL 位址。因此，IE 地圖樣板的語法定義如下：

```
(deftemplate map
  (slot issue_1) (multislot concept_1) (multislot form_1)
  (slot link_type)
  (slot issue_2) (multislot concept_2) (multislot form_2)
)
```

例如：在一個包括二個想法元件的 IE 地圖中，一個想法元件的議題值是“light”，概念值是“opening”和“centralization”，而形式值是“courtyard”與http://lai.cyut.edu.tw/DIM/RA1_1的 URL，而另外一個想法元件的議題值是“light”，概念值是“opening”和“centralization”，而“garden”與http://lai.cyut.edu.tw/DIM/RA1_12的 URL 是此想法元件的形式值，且這兩個想法元件間存在相似元件連結。因此，此 IE 地圖可以由下列的 IE 地圖樣板加以定義。

```
(map (issue_1 light) (concept_1 opening centralization)
  (form_1 courtyard http://lai.cyut.edu.tw/DIM/RA1\_1)
  (link_type similarity)
  (issue_2 light) (concept_2 opening centralization)
  (form_2 garden http://lai.cyut.edu.tw/DIM/RA1\_12)
)
```

ICF 地圖樣板

ICF 地圖樣板主要由三種不同類型的樣版所構成：議題地圖樣板、概念地圖樣板與形式地圖樣板，而這些樣版的名稱分別是 issue_map、concept_map 和 form_map。基本上，每一種類型的樣板皆包含三個欄位值：二個相同類型的屬性欄位值，與它們之間的屬性連結欄位值（similarity, contrast 或 contiguity），這些欄位值皆是關鍵字。這三種 ICF 地圖樣板的語法定義如下：

```

(deftemplate issue_map
  (slot issue_1)(slot link_type)(slot issue_2)
)

(deftemplate concept_map
  (multislot concept_1)(slot link_type)(multislot concept_2)
)

(deftemplate form_map
  (multislot form_1)(slot link_type)(multislot form_2)
)

```

例如：在一個形式地圖中，其中一個形式的欄位值是“courtyard”與 http://lai.cyut.edu.tw/DIM/RA1_1 的 URL，而另外一個形式的欄位值是“garden”與 http://lai.cyut.edu.tw/DIM/RA1_12 的 URL，且屬性連結的欄位值則是相似性（similarity）。因此，此形式地圖可以由下列的議題樣板加以定義。

```

(form_map (form_1 courtyard http://lai.cyut.edu.tw/DIM/RA1\_1)
  (link_type similarity)
  (form_2 terrace http://lai.cyut.edu.tw/DIM/RA1\_12)
)

```

ACL 訊息樣板

ACL 訊息樣板的名稱是 ACLMessage，其主要功能是提供 RA 與 RAS 之間的訊息傳遞以連結與產生想法元件。根據之前 JESS 語法定義的 ACL 訊息樣板（詳見 5.2.3 章節），例如：在解決有關空間組織（spatial organization）的設計問題時，RA_1 將產生的元件連結“similarity”與想法元件“light, opening, courtyard”，以“PROPOSE”的語言意圖向 RAS 提出計畫，此訊息內容可以用下列的 ACLMessage 樣板加以定義。

```

(ACLMessage (communicative-act PROPOSE) (sender RA_1) (receiver RAS)
  (content “similarity light opening courtyard”)
  (language JESS) (ontology spatial_organization)
)

```

除了上面論述的樣板外，為了執行 DIM 之原則有關想法元件的解譯、連結語選擇等機制，RA 的資料結構還包含其他規則基礎在運算過程中所需要的樣板，包括解譯規則在搜尋 ICF 關鍵字的樣板（search_issue template、search_concept template

和 search_form template) 解譯完成後的樣板 (temporary_ie template) 與經由想法元件的連結而產生輸出想法元件候選者的樣板 (selected_ie template) 等，這些樣板在不同的規則基礎中再予以說明。

5.3.1.2 規則基礎

RA 的規則基礎主要包括連結原則與溝通規則。連結原則提供 RA 在進行想法元件連結時，可以在不同的事件類型中使用不同的規則進行 DIM 原則的相關機制，包括想法元件的解譯、連結與選擇，以連結與產生多樣性的想法元件。而溝通規則是提供 RA 與 RAS 在進行相互作用中，有關進行訊息傳遞的發送規則與接收規則。

在 DIM 系統中，上述的規則皆建構在 JESS 的向前鏈接 (forward chaining) 之邏輯機制中，此邏輯機制是以左右手定則 (分別稱為 LHS 和 RHS) 的假如...然後 (if...then) 之語言陳述 (statement)，予以進行這些規則的執行[157]。有關這些規則的邏輯機制與其語法在下面章節有詳細說明。

想法元件的解譯規則

想法元件的解譯規則主要將一個 RA 接收到的輸入想法元件 (input_ie) 中三個屬性值 (議題、概念與形式)，根據 RA 進行解譯的容許限制，經由 RA 內部 ICF 詞典與不同的解譯事件進行這三個屬性值的解譯，其目的除了產生 RA 角色 IE 地圖包含的屬性值外，同時，也產生臨時性的想法元件。

基本上，RA 須先將接收到的 input_ie 之 ICF 屬性值分別放置在搜尋字彙樣板 (包括 search-issue、search-concept 和 search-form templates) 後，才可以進行此 input_ie 的解譯。因此，在解譯規則的邏輯機制中，當其 LHS 中的比對條件滿足 RA 的容許限制 (tolerance 樣板中的欄位值)，同時，上述 input_ie 在搜尋字彙樣板中的欄位值與 RA 之 ICF 詞典中詞典目錄的第一個 token 值或第二個 token 值相同時，在此規則的 RHS 中，則宣稱 (assert) 一個臨時想法元件樣板 (樣板名稱是 temporary_ie) 的事實。此解譯規則 (容許限制為 3) 的語法定義如下：

```
(defrule search-key-word-3
  (tolerance (number 3))
  (search-issue (word ?i))(search-concept (word ?c))(search-form (word ?f))
  (?my_i1 ?my_i2)( ?my_c1 ?my_c2)( ?my_f1 ?my_f2)
  (test ( and (or (eq ?my_i2 ?i)(eq ?my_i1 ?i))
              (or (eq ?my_c2 ?c)(eq ?my_c1 ?c))
              (or (eq ?my_f2 ?f)(eq ?my_f1 ?f))))
  =>
```

```
(assert (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2)))
)
```

想法元件的連結規則

想法元件的連結規則主要把上述經過解議的 *temp_ie*，經由 RA 不同類型的連結形式（包括相似性連結、對比性連結與相鄰性連結），而連結並產生所有可能的想法元件候選者。而在不同類型的連結形式，每一個經過解議的 *temp_ie* 可以經由不同事件類型而連結到相關的想法元件並產生 *cand_ie*，由於，這些不同類型的連結形式具有相類似的邏輯機制（詳見 4.3.2 章節），因此，作者以相似性連結為例進行說明相關事件類型（*Event_{t-1}*、*Event_{t-2}*、*Event_{t-3}*和 *Event_{t-4}*）的規則。

基本上，*Event_{t-1}*是藉由連結 RA 之角色 IE 地圖而產生 *cand_ie* 的事件類型。在 *Event_{t-1}*規則的邏輯機制中，當其 LHS 中一個臨時想法元件樣板之事實的三個欄位值與 IE 地圖樣板之事實的其中一組 ICF 的三個欄位值相同時，且元件連結是相似性連結（similarity）時，在此規則的 RHS 中，則宣稱（assert）一個 *map_ie* 樣板的事實，此事實包含上述 IE 地圖樣板中另外一組 ICF 的三個欄位值。*Event_{t-1}*之連結規則的語法定義如下：

```
(defrule similarity_principle_1
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2))
  ?fact <- (map(issue_1 ?i&:(eq ?i ?my_i2))
            (concept_1 ?c&:(eq ?c ?my_c2))
            (form_1 ?f&:(eq ?f ?my_f2))
            (link_type similarity)
            (issue_2 ?i2)
            (concept_2 ?c2)
            (form_2 ?f2))
  =>
  (assert (map_ie(issue ?i2)(concept ?c2)(form ?f2)))
)
```

此外，RA 的知識基礎須提供搜尋想法元件之喜好欄位值的規則。在此規則的 RHS 中，當上述這些 *map_ie* 樣板事實的 ICF 欄位值，與 RA 知識庫中 *ie* 樣板事實的 ICF 欄位值相同時，在此規則的 RHS 中，則宣稱（assert）一個 *selected_ie* 樣板的事實，此事實包含上述比對成功之想法元件樣板事實的喜好值。此規則的語法定義如下：

```
(defrule si_convert_selected_ie
  (map_ie(issue ?i_map)(concept ?c_map)(form ?f_map))
```

```

?fact <- (ie (issue ?i&:(eq ?i ?i_map)) (concept ?c&:(eq ?c ?c_map))
          (form ?f&:(eq ?f ?f_map)))
=>
(bind ?p (fact-slot-value ?fact preference))
(assert (selected_ie (principle similarity)
                    (issue ?i) (concept ?c) (form ?f) (preference ?p)))
)

```

Event_{t-2} 是直接經由連結 RA 知識庫中的想元件而產生 *cand_ie* 的事件類型。在 *Event_{t-2}* 規則的邏輯機制中，當其 LHS 中一個 *temporary_ie* 樣板事實中二個欄位值 (issue 和 concept，或 issue 和 form) 與 *ie* 樣板事實中相對應的二個欄位值相同時，則此規則的 RHS 除了宣稱 (assert) 一個 *selected_ie* 樣板的事實 (包含上述比對成功之想法元件樣板事實的喜好值) 外，同時，此規則也宣稱一個新的 *ie* 樣板的事實 (包含上述 *temporary_ie* 樣板事實的欄位值) 與一個 *map* 樣板的事實 (包括上述 *temporary_ie* 樣板事實、*ie* 樣板事實和相似元件連結)，這兩種不同樣板的事實被儲存在此 RA 的 JESS 知識庫 (RA_2.clp)。此規則的語法定義如下：

```

(defrule similarity_principle_3
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2))
  ?fact <-(ie (issue ?i&:(eq ?i ?my_i2)) (concept ?c&:(eq ?c ?my_c2))
            (form ?f&:(neq ?f ?my_f2))(preference ?p))
=>
  (assert (selected_ie(principle similarity) (issue ?i) (concept ?c) (form ?f)
                    (preference ?p)))
  (assert (ie(issue ?my_i2)(concept ?my_c2)(form ?my_f2)))
  (assert (map (issue_1 ?i)(concept_1 ?c)(form_1 ?f)(link_type similarity)
              (issue_2 ?my_i2)(concept_2 ?my_c2)(form_2 ?my_f2)))
  (save-facts c:/DIM/RA_2.clp MAIN::ie MAIN::map)
)

```

Event_{t-3} 和 *Event_{t-4}* 是藉由與 ICF 地圖的置換而產生 *cand_ie* 的事件類型，因此，這兩個事件類型須先經由關鍵字的置換規則而發生。在此置換規則的邏輯機制 (以置換概念屬性值為例) 中，當其 LHS 中 *temporary_ie* 樣板事實中之概念欄位值與 *concept_map* 樣板的其中一個欄位值相同時，在此規則的 RHS 中，則原來概念欄位值會被 *concept_map* 樣板事實的另一個欄位值所取代，並宣稱 (assert) 一個已置換的 *similarity_concept* 樣板事實。此規則的語法定義如下：

```

(defrule matching_similarity_concept_map_r

```

```

(temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2))
?fact <- (concept_map(link_type similarity)(concept_1 ?i&:(eq ?i ?my_i2)))
=>
(bind ?similarity_concept (fact-slot-value ?fact concept_2))
(assert (similarity_concept (icf ?similarity_concept)))
)

```

在產生 *cand_ie* 的規則中，當其 LHS 中一個 *temp_ie* 經由置換的 *similarity_concept* 樣板事實的欄位值與 *ie* 樣板事實的概念欄位值相同時，在此規則的 RHS 中，則宣稱 (assert) 一個 *selected_ie* 樣板事實 (包含上述比對成功之想法元件樣板事實的喜好值)。同時，此規則也宣稱一個新的 *ie* 樣板的事實 (包含上述 *temporary_ie* 樣板事實的欄位值) 與一個 *map* 樣板的事實 (包括上述 *temporary_ie* 樣板事實、*ie* 樣板事實和相似元件連結)，這兩種不同樣板的事實被儲存在此 RA 的 JESS 知識庫 (RA_2.clp)。此規則 (*Event1_3*) 的語法定義如下：

```

(defrule similarity_principle_5
(temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2))
(similarity_concept (concept ?similarity_concept))
?fact <- (ie (issue ?i&:(eq ?i ?my_i2)) (concept ?c&:(eq ?c ?similarity_concept))
(form ?f&:(neq ?f ?my_f2))(preference ?p))
=>
(assert (selected_ie (principle similarity)
(issue ?i)(concept ?c)(form ?f)(preference ?p)))
(assert (ie(issue ?my_i2)(concept ?my_c2)(form ?my_f2)))
(assert (map (issue_1 ?i)(concept_1 ?c)(form_1 ?f)(link_type similarity)
(issue_2 ?my_i2)(concept_2 ?my_c2)(form_2 ?my_f2)))
(save-facts c:/DIM/RA_2.clp MAIN::ie MAIN::map)
)

```

有關相似原則與其他原則 (對比原則與相鄰原則) 的事件類型與其規則詳見於本論文的附錄 A。

想法元件的選擇規則

想法元件的選擇規則是將產生的候選想法元件 (*cand_ie*) 根據 RA 喜好的連結原則，提供 RA 選擇一個具有最大喜好值的想法元件為最後輸出的想法元件 (詳見 4.3.3 章節)。因此，選擇規則主要包括 1) 產生二個具有最大喜好值之 *selected_ie* 樣板事實的目錄集合，一組集合是放置與使用者輸入的連結原則類型相同之

selected_ie 樣板事實的目錄，而另一組集合則放置其他二種連結原則類型之 selected_ie 樣板事實的目錄，接著 2) 根據 RA 喜好的連結原則類型，從這兩個集合的目錄中選擇一個事實為輸出想法元件 (*output_ie*)

在產生具有最大喜好值事實的目錄規則中，作者以相同於輸入的連結原則類型的 selected_ie 樣板事實之目錄為例進行說明。在此規則的邏輯機制中，當其 LHS 中一個 selected_ie 樣板事實中的 principle 欄位值與 principle 樣板事實的欄位值相同時，且這些 selected_ie 樣板事實中的喜好值相互比較後，在此規則的 RHS 中，則具有最大喜好值之事實的相關欄位值（包括原則類型、議題、概念、形式、URL）被放置在一組目錄集合中。此規則的語法定義如下：

```
(defrule selected_prefered_ie
  (principle (type ?my_type))
  ?fact <- (selected_ie (principle ?t&:(eq ?t ?my_type)) (issue ?i) (concept ?c)
            (form ?f ?url) (preference ?p))
  (not (selected_ie (preference ?p2&:(> ?p2 ?p))))
=>
  (bind ?*num* (+ ?*num* 1))
  (bind ?*sim_list* (insert$ ?*sim_list* (+ (length$ ?*sim_list*) 1)
                                   (create$ ?t ?i ?c ?f ?url)))
)
```

有關另一組 selected_ie 樣板事實目錄的建立，其規則也相似於上述 selected_prefered_ie 規則的邏輯機制，其語法定義可以詳見附錄 A。在選擇一個事實為輸出想法元件的規則中，當其 LHS 中一個 ACLMessage 樣板事實的 content 欄位值與 RA 接收到的訊息內容不相同時，則此規則的 RHS 將根據不同的條件，在其中一組目錄集合中隨機的選擇一組相關的目錄值（包括原則類型、議題、概念、形式、URL）以取代原有 ACLMessage 樣板事實的 content 值。然而，假如這兩組目錄集合都沒有目錄的放置時，則此規則無法產生一個事實為輸出想法元件，並告知相關代理人（RAS）任務失敗。此規則的語法定義如下：

```
(defrule select_a_random_ie
  ?m <- (ACLMessage (communicative-act PROPOSE) (sender ?s)
          (content ?c&:(neq ?c ?*new_c*)) (receiver ?r))
=>
  (if (> ?*num* 0) then
    (bind ?rd (* (mod (random) ?*num*) 5))
    (bind ?*new_c* (implode$ (create$ (nth$ (+ ?rd 1) ?*sim_list*)))
  )
```

```

        (nth$ (+ ?rd 2) ?*sim_list*)
        (nth$ (+ ?rd 3) ?*sim_list*)
        (nth$ (+ ?rd 4) ?*sim_list*)
        (nth$ (+ ?rd 5) ?*sim_list*))))
    (modify ?m (content ?*new_c*))
  else
    (if (> ?*num_1* 0) then
      (bind ?rd (* (mod (random) ?*num_1*) 4))
      (bind ?*new_c* (implode$ (create$
        (nth$ (+ ?rd 1) ?*sim_list_1*)
        (nth$ (+ ?rd 2) ?*sim_list_1*)
        (nth$ (+ ?rd 3) ?*sim_list_1*)
        (nth$ (+ ?rd 4) ?*sim_list_1*)
        (nth$ (+ ?rd 5) ?*sim_list_1*))))
      (modify ?m (content ?*new_c*))
    else
      (if (= ?*num_1* 0) then
        (modify ?m (communicative-act FAILURE)
          (content "I have no design ideas")))))
  )

```

溝通規則

角色代理人的溝通規則主要建立在 RAS 與 RA 之間，有關產生想法元件的訊息傳遞規則，主要包括接收規則與發送規則。接收規則是 RA 從 RAS 接收到要求產生想法元件計劃的訊息，並予以轉換成進行想法元件連結的相關資訊，而發送規則是將產生的輸出想法元件之訊息提出計劃給 RAS。

在接收規則的邏輯機制，當其 LHS 中一個 ACLMessgae 樣板事實中溝通意圖（communicative-act）欄位值為（CFP）產生時，在此規則的 RHS 中，則根據此事實中的訊息內容宣稱（assert）成不同樣板名稱的事實，同時，進行修正（modify）原有 ACLMessgae 樣板事實中的相關欄位值。此規則的語法定義如下：

```

(defrule proposal
  ?m <- (ACLMessage (communicative-act CFP) (sender ?s) (content ?c) (receiver ?r))
=>
  (assert (search-issue (word (nth$ 1 (explode$ ?c))))))
  (assert (search-concept (word (nth$ 2 (explode$ ?c))))))

```

```

(assert (search-form (word (nth$ 3 (explode$ ?c))))))
(assert (principle (type (nth$ 4 (explode$ ?c))))))
(assert (tolerance (number (nth$ 5 (explode$ ?c))))))
(bind ?*type* (nth$ 4 (explode$ ?c)))
(modify ?m (communicative-act PROPOSE) (sender ?r) (receiver ?s) (content ?c))
)

```

在發送規則的邏輯機制，當其 LHS 之上述 ACLMessage 樣板事實中相關欄位值被修正完成後，如溝通意圖（communicative-act）欄位值修正為（PROPOSE）（詳見上述接收規則的語法定義），則此規則的 RHS 將修正過後的 ACLMessage 樣板事實發送給 RAS。此規則的語法定義如下：

```

(defrule send-a-message
  (MyAgent (name ?n))
  ?m <- (ACLMessage (sender ?n))
  =>
  (send ?m)
)

```

5.3.1.3 規則組織與區劃

為了避免 DIM 系統在執行原則中不同類型的事件規則，而導致規則之間的衝突（rule confliction），同時，提供此系統能有效的控制想法元件的連結過程，根據本研究在第四章建立的連結機制（詳見 4.3.3 章節），這些事件規則必須依據 RA 在連結想法元件的步驟而予以適當的區劃（partition）。

根據 RA 的溝通規則與原則的步驟，上述規則基礎的規則被組織成下列的區劃與步驟，包括 1) 訊息接收規則、2) 想法元件的解譯規則、3) 三種連結形式（相似性連結、對比性連結及相鄰性連結）規則、4) 想法元件的選擇規則與 5) 訊息發送規則。而架構在 JESS 的機制中，這些規則組織的區劃方式可以藉由不同模組（module）的建立，而將上述不同步驟所包含的規則予以區劃。例如：想法元件的連結規則在相似性連結中，其所包含不同事件類型（ $Event_{1-1}$ 、 $Event_{1-2}$ 、 $Event_{1-3}$ 和 $Event_{1-4}$ ）的規則可以藉由 SI 模組的定義和其它事件規則區劃開來，其語法定義如下：

```

(defmodule SI)

```

為了避免在連結規則中，ICF 地圖轉換關鍵字之規則在不同類型連結原則的重複與衝突，DIM 系統將連結規則區劃成四個模組，包括轉換關鍵字規則的模組與三種

連結形式規則的模組 (SI、CR、CI)。因此，RA 的規則基礎被區劃成八個模組，這八個模組依序為 Proposal、Interpret、Convert、SI、CR、CI、Select 和 Send，而其中 SI、CR 與 CI 的順序可以透過變數 (RA 喜好的連結原則類型) 的設定，而有六種不同的模組序列。這六種不同的模組序列經由 run-system 的功能，予以集中 (focus) 這八個模組中不同規則的啟動順序。有關此功能的語法定義如下：

```
(deffunction run-system (?*type)
  (if (eq ?link SI) then
    (bind ?module_list (create$ SI CR CI SI CI CR))
    (bind ?rd (* (mod (random) 2) 3))
    (focus Proposal Interpret Convert (nth$ (+ ?rd 1) ?module_list)
      (nth$ (+ ?rd 2) ?module_list)
      (nth$ (+ ?rd 3) ?module_list))
    Select Send)
  else
    (if (eq ?link CR) then
      (bind ?module_list (create$ CR SI CI CR CI SI))
      (bind ?rd (* (mod (random) 2) 3))
      (focus Proposal Interpret Convert (nth$ (+ ?rd 1) ?module_list)
        (nth$ (+ ?rd 2) ?module_list)
        (nth$ (+ ?rd 3) ?module_list))
      Select Send)
    else
      (if (eq ?link CI) then
        (bind ?module_list (create$ CI SI CR CI CR SI))
        (bind ?rd (* (mod (random) 2) 3))
        (focus Proposal Interpret Convert (nth$ (+ ?rd 1) ?module_list)
          (nth$ (+ ?rd 2) ?module_list)
          (nth$ (+ ?rd 3) ?module_list))
        Select Send))))))
)
```

例如：當 RA 選擇對比性的連結原則時，RA 可以經由 run-system 的功能在兩個模組區劃中隨機的選擇，包括 Proposal、Interpret、Convert、CR、SI、CI、Select 和 Send 的模組序列，與 Proposal、Interpret、Convert、CR、CI、SI、Select 和 Send 的模組序列。進而，根據 run-system 功能，IE 產生器中的 Matcher 藉由模式比對的方式，決定此 RA 對目前的工作記憶內容所採取的規則，並經由 Agenda 控制所有 DIM 原則之規則執行與啟動的優先順序後，最後，RA 開始執行想法元件的連

結與產生。

5.3.2 場記代理人

在 DIM 系統的內在設計情境層中，場記代理人 (RAS) 透過與不同 RA 和 UA (或 DA) 之間的相互作用，而動態性的儲存並視覺化此相互作用的設計結果 (內部 IE 地圖)，同時，RAS 也會在此內部 IE 地圖中自動的搜尋一個想法等級最大的想法元件，以提供使用者輸入想法元件的參考依據。因此，RAS 的知識基礎主要包括：1) 工作記憶、2) 規則基礎、3) 內部 IE 地圖產生器及 4) RAS 知識庫 (圖 48)。

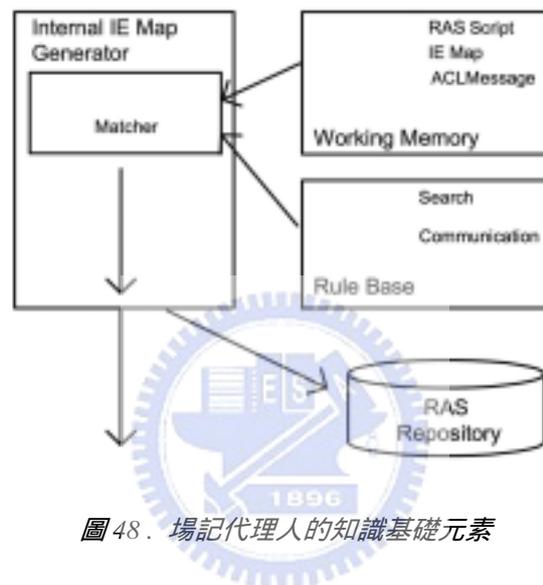


圖 48. 場記代理人的知識基礎元素

基本上，RAS 的工作記憶主要放置場記的腳本、IE 地圖與訊息傳遞的相關資訊，規則基礎則儲存 RAS 進行想法連結的相關規則，而內部 IE 地圖產生器控制整個產生地圖的規則與資訊的流程，另外，RAS 知識庫則儲存不同角色代理人共同產生的想法元件與這些想法元件之間的元件連結。

5.3.2.1 工作記憶

RAS 的工作記憶主要包括內部相互作用的腳本內容、想法元件、內部 IE 地圖與代理人溝通語言的相關資訊，而這些資訊分別由場記樣板、想法元件樣板、IE 地圖樣板與 ACL 訊息樣板所構成。

場記樣板

場記樣板的名稱是 script-holder，其功能是描述 RAS 在某一特定場景中內部相互作用的腳本內容，而此內容提供 RAS 與不同 RA 進行內部相互作用的依據。基本上，場記樣板包括“issue”、“time”、“role_number”、“role_id”、“principle”與“tolerance”六個欄位。

所謂 issue 欄位是指 RA 要解決的設計問題(議題), time 欄位是不同 RA 連結想法元件的時間, role-number 欄位是使用者(UA 或 DA)可以扮演角色的最大數量, rolei-id 欄位是不同 RA 的 AID, principle 欄位是不同 RA 喜好的連結原則, 而 tolerance 欄位則是不同 RA 解議的容許限制。因此, 場記樣板的語法定義如下:

```
(deftemplate script-holder (slot issue) (slot time) (slot role-number) (multislot role-id)
(multislot principle)(multislot tolerance)
)
```

例如: 在一個有關 circulation 議題的場景中, 一位 UA (以 UA_{lai} 為例) 選擇三位角色代理人 (RA1、RA2 與 RA3) 進行角色扮演, 而這三位 RA 喜好的連結原則依序是 similarity、contrast 與 contiguity, 且它們在解譯的容許限制分別是 3、2 和 3, 另外, 此內部相互作用的參與時間是 2 分鐘, 因此, 上述的腳本內容可以用下列 script-holder 的樣板事實予以呈現:

```
(script-holder (issue circulation) (time 2) (role_number 3)
(role_id RA1@lai:1099/jade RA2@lai:1099/jade RA3@lai:1099/jade)
(principle similarity contrast contiguity) (tolerance 3 2 3)
)
```

除了上述的場記樣板外, 想法元件樣板是構成內部 IE 地圖的主要元素外, 同時, 此樣板也提供 RAS 在內在相互作用中, 搜尋最大想法等級之想法元件相關規則的主要資料(如 search_link_using_ie 規則), 而 IE 地圖樣板提供 RAS 將不同 RA 產生的想法元件與元件連結, 予以建立內部 IE 地圖的資料, 另外, ACL 訊息樣板則是提供 RAS 與其他代理人進行內部相互作用時的相關訊息資料。有關這些 RAS 樣板的機制與語法相同於上述 RA 工作記憶的樣板, 其內容詳見 5.3.1 章節。

除了上述的樣板資料外, DIM 系統為了計算內部 IE 地圖中想法元件的想法等級, RAS 的資料結構還包含其規則基礎在運算過程中所需要的臨時性樣板, 如計算最大想法等級之想法元件的 link_count 樣板。這些臨時性樣板會在描述 RAS 的規則基礎中再予以說明。

5.3.2.2 規則基礎

RAS 的規則基礎主要包括搜尋規則與溝通規則。搜尋規則是 RAS 對於不同 RA 共同產生的內部 IE 地圖, 藉由計算出具有最多想法等級的想法元件, 提供 UA (或 DA) 作為輸出想法元件的參考。而溝通規則為提供角色代理人, UA 或 DA 在進行相互作用中, 有關於相關訊息的發送規則與接收規則。

搜尋規則

為了搜尋內部 IE 地圖中具有最多想法等級的想法元件，搜尋規則主要包括在內部 IE 地圖中計算每一個想法元件之想法等級（一個想法元件之元件連結的總數量）的規則，以及產生一個具有最多想法等級的想法元件。在計算每一個想法元件的想法等級的規則中，RAS 首先必須在其工作記憶中搜尋 map 樣板事實中的兩組 ICF 欄位值（issue、concept 和 form），如在搜尋第一組的 ICF 欄位值的語法定義如下：

```
(defquery search-link-assignments-left
  (declare (variables ?i ?c ?f ?url))
  (map (issue_1 ?i) (concept_1 ?c)(form_1 ?f ? url))
)
```

然後，藉由 search_link_number 功能計算所有 map 樣板事實中同一組 ICF 的想法等級之數值，假如想法等級數值不等於 0 時，則宣稱（assert）一個描述此組 ICF 想法等級的 link-count 樣板事實，而此事實 count 欄位值包含想法等級的數值。此功能的語法定義如下：

```
(deffunction search_link_number (?i ?c ?f ?url)
  (bind ?count_l (count-query-results count-link-assignments-left ?i ?c ?f ?url))
  (bind ?count_r (count-query-results count-link-assignments-right ?i ?c ?f ?url))
  (bind ?count (+ ?count_l ?count_r))
  (if (neq ?count 0) then
    (assert (link-count (ie ?i ?c ?f ?url)(count ?count))))
)
```

而上述 search_link_number 功能必須藉由下列的規則予以執行。

```
(defrule search_link_using_ie
  (ie (issue ?i)(concept ?c)(form ?f ?url))
  (not (link-count (ie ?i ?c ?f ?url))))
=>
  (search_link_number ?i ?c ?f ?url)
)
```

在產生一個具有最多想法等級的想法元件規則中，包括產生具有最大想法等級之 link-count 樣板事實的目錄，和從此目錄中隨機的選擇一組事實之值，以作為內部相互作用的輸出想法元件（output_ie）。在產生具有最大想法等級之 link-count 樣

板事實的目錄中，在此規則的 LHS 中，這些 link_count 樣板事實中之 count 欄位數值相互比較後，在此規則的 RHS 中，則將具有最大數值之事實的 ie 欄位值（包括議題、概念、形式與形式的 URL）放置在一個目錄中。此規則的語法定義如下：

```
(defrule print-highest-link-assignment
  (link-count (ie ?i ?c ?f ?url)(count ?count))
  (not (link-count (count ?count2&:(> ?count2 ?count))))
=>
  (bind ?*num* (+ ?*num* 1))
  (bind ?*ie_list* (insert$ ?*ie_list* (+ (length$ ?*ie_list*) 1) (create$ ?i ?c ?f ?url)))
)
```

在此目錄中隨機的選擇一組事實以作為輸出想法元件，在其規則的 LHS 中，當 ACLMessage 樣板事實的 communicative-act 欄位值是 PROPOSE 時，在此規則的 RHS 中，則在上述規則產生的目錄中隨機選擇一組（包括四個 token）為輸出想法元件之內容，同時，此內容透過一個 ACLMessage 樣板事實的宣稱（assert）而傳遞給使用者（UA 或 DA）。此規則的語法定義如下：

```
(defrule propose_highest_link_ie
  (link-count (ie ?i ?c ?f ?url)(count ?count))
  ?m <- (ACLMessage (communicative-act PROPOSE) (sender ?s) (content ?c)
          (receiver ?r))
=>
  (bind ?rd (* (mod (random) ?*num*) 4))
  (bind ?output_ie (implode$ (create$
                              (nth$ (+ ?rd 1) ?*ie_list*)
                              (nth$ (+ ?rd 2) ?*ie_list*)
                              (nth$ (+ ?rd 3) ?*ie_list*)
                              (nth$ (+ ?rd 4) ?*ie_list*))))
  (retract ?m)
  (assert (ACLMessage (communicative-act PROPOSE) (sender ?r) (receiver ?s)
                      (content ?output_ie)))
  (time_save)
)
```

而為了動態性的紀錄此內部 IE 地圖在不同時間的變化，在上述規則的 RHS 中，time_save 功能提供 RAS 在內部相互作用的不同時間範圍內（起始時間與中止時間的時間差），能將不同 RA 共同產生的內部 IE 地圖與想法元件，予以動態性的儲存

在 RAS 的 JESS 知識庫中。此功能的語法定義如下：

```
(deffunction time_save ( )
  (bind ?t2 (- (time) ?*time-begin*))
  (bind ?file (implode$ (create$ c:/DIM/RAS/I_IE_Map ?t2 .clp)))
  (save-facts ?file script-holder map ie)
)
```

而上述的規則也是 UA (或 DA) 要求 RAS 產生一個最多想法等級之想法元件的溝通規則。另外，RAS 也會把此內部相互作用的訊息流程 (message queue) 儲存在 RAS 的內部 IE 地圖知識庫，此訊息流程主要包括內部相互作用中的所有訊息，而這些訊息的格式 (format) 架構在 JADE 的機制，除了紀錄每筆資訊的時間外，這些資訊內容由 FIPA 之 ACL 的元素所構成。

2005/9/1 下午 3:33

```
(CFP :sender ( agent-identifier :name RAS@lai:1099/JADE :addresses (sequence
  http://lai:7778/acc ))
  :receiver (set ( agent-identifier :name RA1@lai:1099/JADE ))
  :content "circulation opening court similarity 3"
  :language jess
  :ontology spatial_organization
)
```

在 DIM 系統中，此訊息流程的紀錄可以提供 UA (包括 DA) 藉由載入此訊息流程的檔案，而重新使用這些訊息內容並進行內部相互作用的想法連結。

溝通規則

RAS 的溝通規則包括四個階段的訊息接收與發送規則，第一階段是 UA (或 DA) 要求 RAS 開始進行內部相互作用，而第二階段是 RAS 要求 RA 提出產生想法元件與元件連結的計畫，第三階段則是 RA 要求 RAS 將產生的內部 IE 地圖放置在 RAS 的知識庫，第四階段是 UA (或 DA) 要求 RAS 產生最多想法等級之想法元件的計畫，同時將產生的 IE 地圖動態性的儲存在 RAS 的知識庫。

在第一個階段中，此規則的 LHS 是當 RAS 接收到 ACLMessage 樣板事實之 communicative-act 欄位值是 REQUEST_WHEN，且 script-holder 樣板事實的 role-number 欄位值是一個整數值 (以 RA 的參與數量為 2 為例) 時，則此規則的 RHS 將 ACLMessage 樣板事實中 content 欄位之字串值轉換成目錄的 tokens，而這些 tokens 提供此規則宣稱 (assert) 一個 script-holder 樣板事實，且控制此內部相

互作用的時間，而 RAS 也經由一個新宣稱的 ACLMessage 樣板事實將訊息回傳給 UA (或 DA)。此規則的語法定義如下：

```
(defrule request_when_internal_interplay_2
  ?m <- (ACLMessage (communicative-act REQUEST_WHEN)
          (sender ?s) (content ?c) (receiver ?r))
  (role-script (role-number 2))
=>
  (assert (script-holder (issue (nth$ 1 (explode$ ?c))) (time (nth$ 2 (explode$ ?c)))
                    (role-number (nth$ 3 (explode$ ?c)))
                    (role_id (nth$ 4 (explode$ ?c))(nth$ 5 (explode$ ?c)))
                    (role_principle (nth$ 6 (explode$ ?c) (nth$ 7 (explode$ ?c)))
                    (role_tolerance (nth$ 8 (explode$ ?c))(nth$ 9 (explode$ ?c)))))
  (bind ?*time-duration* (* (nth$ 2 (explode$ ?c)) 60 )
  (assert (ACLMessage (communicative-act INFORM) (sender ?s) (receiver ?r))
          (content "I will start the idea linking"))
  (retract ?m)
)
```

此外，上述規則藉由 script-holder 樣板事實之 role-number 欄位值來控制 ACLMessage 樣板事實中有關字串轉換成 token 的數量，進而影響 script-holder 樣板事實相關的欄位值。而在控制內部相互作用的時間，主要經由下面的規則予以執行，此規則係指當起始時間 (?*time-begin*) 與終結時間等於參與時間 (?*time-duration*) 時，則 RAS 將停止所有的規則啟動，並宣稱 (assert) 一個告知 UA (或 DA) 結束此內部相互作用的 ACL 訊息樣板事實。此規則的語法定義如下：

```
(while (> ?*time-duration* (- (time) ?*time-begin*)) do
  (assert (ACLMessage (communicative-act INFORM) (sender ?s) (receiver ?r))
          (content "Time out !"))
  (halt)
)
```

有關第二個階段的規則可以詳見上一章節 RA 的溝通規則。而在 RAS 第三階段的規則中，當其 LHS 中的一個 ACL 訊息樣板事實之 communicative-act 欄位值是 REQUEST 時，在此規則的 RHS 中，則將 content 欄位之字串值轉換成目錄的 tokens，並根據目錄 token 的順序分別宣稱 (assert) map 樣板事實與 ie 樣板事實，同時，RAS 也經由一個新宣稱的 ACLMessage 樣板事實將訊息回傳給 RA。此規則

的語法定義如下：

```
(defrule request_ie_map
  ?m <- (ACLMessage (communicative-act REQUEST)(sender ?s) (content ?c)
        (receiver ?r))
  =>
  (assert (map (issue_1 (nth$ 1 (explode$ ?c))) (concept_1 (nth$ 2 (explode$ ?c)))
              (form_1 (nth$ 3 (explode$ ?c))(nth$ 4 (explode$ ?c)))
              (link_type (nth$ 5 (explode$ ?c)))
              (issue_2 (nth$ 6 (explode$ ?c))(concept_2 (nth$ 7 (explode$ ?c)))
              (form_2 (nth$ 8 (explode$ ?c))(nth$ 9 (explode$ ?c))))))
  (assert (ie (issue (nth$ 1 (explode$ ?c))) (concept (nth$ 2 (explode$ ?c)))
              (form (nth$ 3 (explode$ ?c))(nth$ 4 (explode$ ?c))))))
  (assert (ie (issue (nth$ 6 (explode$ ?c))) (concept (nth$ 7 (explode$ ?c)))
              (form (nth$ 8 (explode$ ?c))(nth$ 9 (explode$ ?c))))))
  (assert (ACLMessage (communicative-act INFORM) (sender ?r) (receiver ?s)
                    (content ?c)))
  (retract ?m)
)
```

RAS 在第四個階段的規則可以詳見上述 RAS 之 propose_highest_link_ie 規則與 time_save 功能，此外，這些產生的內部 IE 地圖被儲存在 RAS 的內部 IE 地圖知識庫，並藉由圖像視覺化介面予以呈現。而經由這些接收規則而產生新的訊息（事實），皆由下面的發送規則將相關訊息傳遞給指定的接收者，此規則的語法定義如下：

```
(defrule send-a-message
  (MyAgent (name ?n))
  ?m <- (ACLMessage (sender ?n))
  =>
  (send ?m)
)
```

最後，在 RAS 之知識基礎中，其內部 IE 地圖產生器的 Matcher，將藉由相關訊息的模式比對方式，決定此 RAS 對目前的工作記憶內容所採取的規則，以採取適當的行為進行內部相互作用的互動。

5.3.3 場景代理人

在 DIM 系統的外在設計情境層中，場景代理人(ScA)經由和不同 UA、 DA 和 StA 的外部相互作用（包括如想法元件與其他腳本內容的訊息傳遞），而動態性的儲存並視覺化此相互作用的設計結果（外部 IE 地圖），同時，ScA 也會在此外部 IE 地圖中，自動的搜尋一個想法等級最大的想法元件，以提供使用者輸入想法元件的參考依據。因此，ScA 的知識基礎主要包括：1) 工作記憶、2) 規則基礎、3) 外部 IE 地圖產生器與 4) ScA 知識庫（圖 49）。

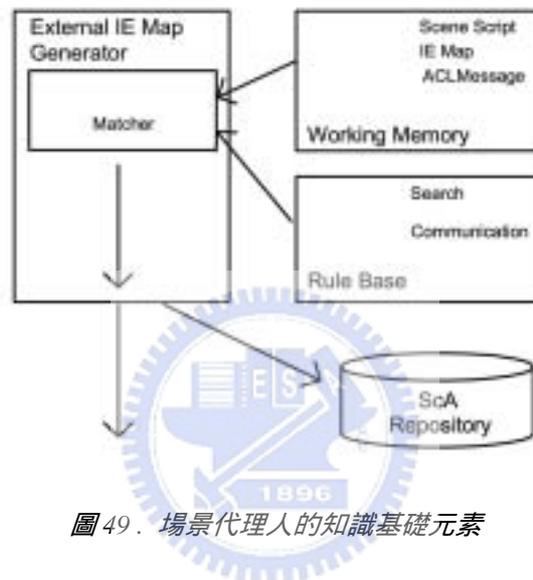


圖 49. 場景代理人的知識基礎元素

基本上，ScA 的工作記憶主要放置場景腳本、外部 IE 地圖與進行訊息傳遞的樣板資料，規則基礎則是儲存 ScA 進行想法連結的相關規則，而外部 IE 地圖產生器是控制整個產生地圖的規則與其資訊的流程，另外，外部 IE 地圖知識庫則儲存不同 UA 與 DA 共同產生的想法元件與其之間的元件連結。

5.3.3.1 工作記憶

ScA 的工作記憶主要包括場景腳本、想法元件、外部 IE 地圖與代理人溝通語言的相關資訊，而這些資訊分別由場景樣板、想法元件樣板、IE 地圖樣板與 ACL 訊息樣板所構成。

場景樣板

場景樣板的名稱是 scene，其功能是描述 ScA 在某一特定場景的外部相互作用中之腳本內容，而此腳本內容主要提供 ScA 與其他代理人進行相互作用的依據。基本上，場景樣板包括“issue”、“time”、“role-number”、“role-skill”與“role-id”五個欄位。

所謂 issue 欄位係指使用者 (UA 或 DA) 在一個場景中欲解決的一個特定設計問題 (議題) , 而 time 欄位關係此場景所需要的時間 , role-number 欄位是使用者可以扮演角色的最大數量 , role-skill 欄位是此場景所需要 RA 的設計技能 , 另外 , role-id 欄位則是使用者在其各自的內部相互作用中 , 他們選擇之不同 RA 的 AID。因此 , 場景樣板的語法定義如下 :

```
(deftemplate scene (slot issue)(slot time)(slot role-number)(multislot role-skill)
(multislot role-id)
)
```

例如 : 在一個有關 circulation 議題的場景中 , 此場景需要有住宅設計之設計技能的角色代理人 , 同時 , 每一位使用者 (UA 和 DA) 在此場景最多只能扮演兩位角色 , 因此 , 當使用者是 DA_{lai} 和 UA_{tengwen} , 而其中 DA_{lai} 扮演的角色是 RA1 和 RA2 , 而 UA_{tengwen} 扮演的角色是 RA3 和 RA4 , 另外 , 此場景進行的時間是十分鐘。因此 , 上述的腳本內容可以用下列場景樣板事實予以呈現 :

```
(scene (issue circulation) (time 10) (role-number 2)( role-skill house)
(role-id RA1@lai:1099/jade RA2@lai:1099/jade RA3@tengwen:1099/jade
RA4@tengwen:1099/jade )
)
```

除了上述的場景樣板外 , 想法元件樣板是構成外部 IE 地圖的主要元素 , 並提供 ScA 能在外在相互作用中 , 搜尋到最大想法等級之想法元件相關規則的主要資料 , 而 IE 地圖樣板提供 ScA 將不同 UA 或 DA 產生的輸出想法元件與元件連結 , 予以建立外部 IE 地圖的資料 , 另外 , ACL 訊息樣板則是提供 ScA 與其他代理人進行外部相互作用時的相關訊息資料。有關這些 ScA 樣板的機制與語法相同於上述 RA 工作記憶的樣板 , 其內容詳見 5.3.1 章節。

5.3.3.2 規則基礎

ScA 的規則基礎主要包括搜尋規則與溝通規則。搜尋規則是對於使用者 (UA 與 DA) 所共同產生的外部 IE 地圖 , 經由計算出具有最多想法等級的想法元件 , 以提供這些使用者選擇共同設計想法的參考依據。另外 , 溝通規則是提供 ScA 與 StA 和使用者在進行外部相互作用中 , 有關訊息傳遞的發送規則與接收規則。

基本上 , ScA 與 RAS 有相似的搜尋規則 , 其差別在於 ScA 的搜尋規則是在其外部 IE 地圖的知識庫中計算並搜尋最多想法等級的想法元件 (有關 ScA 的搜尋規則可以參考 RAS 在 propose_highest_link_ie 規則的語法) , 同時 , 將此場景的腳本內容、

產生的 IE 地圖與想法元件儲存在此 ScA 的 JESS 知識庫中。其語法定義如下：

```
(deffunction time_save ()
  (bind ?t2 (- (time) ?*time-begin*))
  (bind ?file (implode$ (create$ c:/DIM/ScA1/E_IE_Map ?t2 .clp)))
  (save-facts ?file scene map ie)
)
```

最後，儲存在此 ScA 知識庫的外部 IE 地圖，可以藉由圖像視覺化介面予以呈現。另外，ScA 也會把此外部相互作用的訊息流程 (message queue) 予以儲存在 StA 的知識庫，此訊息流程提供 DA 藉由重新載入訊息流程的檔案，重新加以利用這些訊息進行外部相互作用的想法連結，而此訊息流程的機制與語法可以詳見於 161 頁。

溝通規則

ScA 的溝通規則主要包括四個階段的訊息接收與發送規則。第一階段是 StA 要求 ScA 開始進行此場景的外部相互作用，而第二階段是 ScA 告知使用者 (UA 或 DA) 開始進行外部相互作用的想法連結，第三階段則是使用者要求 RAS 將產生的 IE 地圖放置在 ScA 的知識庫，最後，DA 在第四階段要求 ScA 產生最多想法等級之想法元件的計畫，同時，ScA 將產生的外部 IE 地圖動態性儲存在它的外部 IE 地圖知識庫。

在第一階段中，此接收規則的 LHS 是當 ScA 接收到 ACL 訊息樣板事實之 communicative-act 欄位值是 REQUEST_WHEN，且 scene 樣板事實的 role-number 欄位值是一個整數值 (以 RA 的參與數量為 2 為例) 時，在此規則的 RHS 中，則宣稱 (assert) 一個 scene 樣板事實和 ACLMessage 樣板事實，且控制此外部相互作用的時間，而 ScA 也經由一個新宣稱的 ACLMessage 樣板事實將訊息回傳給 StA。此規則的語法定義如下：

```
(defrule request_when_external_interplay
  ?m <- (ACLMessage (communicative-act REQUEST_WHEN)
          (sender ?s) (content ?c) (receiver ?r))
  (scene (role-number 2))
=>
  (assert (scene (issue (nth$ 1 (explode$ ?c))) (time (nth$ 2 (explode$ ?c)))
              (role-number (nth$ 3 (explode$ ?c)))
              (role-skill (nth$ 4 (explode$ ?c)))
              (role-id (nth$ 5 (explode$ ?c)) (nth$ 6 (explode$ ?c))))))
```

```

(bind ?*time-duration* (* (nth$ 2 (explode$ ?c)) 60 )
(assert (ACLMessage (communicative-act INFORM) (sender ?s) (receiver ?r)
          (content ?c))
(retract ?m)
)

```

此外，上述規則藉由 scene 樣板事實之 role-number 欄位值來控制 ACLMessage 樣板事實中有關字串轉換成 token 的數量，進而影響 scene 樣板事實相關的欄位值。

在第二階段中，ScA 告知使用者開始進行外部相互作用的想法連結，同時，使用者會根據 StA 之 play 樣板事實中 user-order 的欄位值，依序在 ScA 的外部 IE 地圖中選擇一個想法元件，並要求他們自己的 RAS 開始進行內部相互作用，此規則詳見於 RAS 的規則基礎章節。

由於，ScA 與 RAS 在第三階段與第四階段的溝通規則是相似（差別在於接收者身分的不同），因此，有關這二個階段的溝通規則可以詳見 RAS 的 request_ie_map 與 propose_ie_map 規則。而經由這些接收規則產生新的訊息（事實），皆由下面的發送規則將相關訊息傳遞給指定的接收者。

```

(defrule send-a-message
(MyAgent (name ?n))
?m <- (ACLMessage (sender ?n))
=>
(send ?m)
)

```

最後，在 ScA 的知識基礎中，其外部 IE 地圖產生器中的 Matcher，將藉由相關訊息的模式比對方式，決定此 ScA 對於目前知識庫內容所採取的規則，並採取適當的行為進行外部相互作用的互動。

5.3.4 舞台代理人

在 DIM 系統的外在設計情境層中，舞台代理人（StA）經由與使用者（DA 和 UA）以及 ScA 進行外部相互作用，將 DA 外在相互作用的腳本內容傳遞給所有 ScA，並根據腳本內容中場景的順序，依序要求每一位 ScA 在一定時間內，完成一個指定議題的想法連結工作，而當這些場景的想法連結時間終止時，則 StA 將每一位 ScA 產生的設計結果（外部 IE 地圖），與其訊息傳遞過程紀錄儲存在 StA 的知識庫。因此，StA 的知識基礎主要包括：1) 工作記憶、2) 規則基礎、3) 推理引擎 4) StA 知識庫（圖 50）。

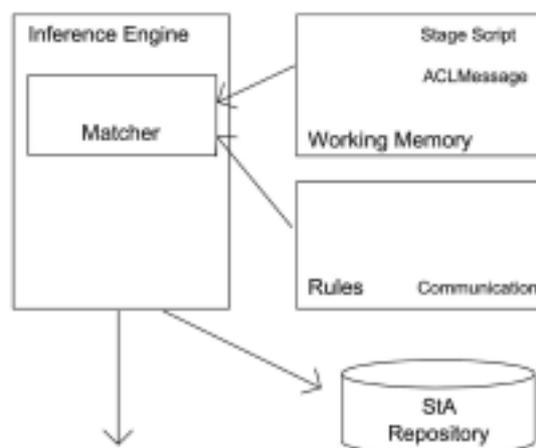


圖 50. 舞台代理人的知識基礎元素

基本上，StA 的工作記憶主要放置整個舞台劇本（play，或稱遊戲）的腳本內容與訊息傳遞的相關資訊，規則基礎則是儲存 StA 進行想法連結的溝通規則，而推理引擎是控制整個溝通規則的流程，另外，StA 知識庫則儲存一個完整想法連結工作中，所有場景的設計結果（外部 IE 地圖）與其整個連結過程的訊息流程。

5.3.4.1 工作記憶

StA 的工作記憶由劇本樣板與 ACL 訊息樣板所構成，而這兩種樣版是 StA 在外部相互作用中，進行想法連結的主要資料。

劇本樣板

劇本樣板的名稱是 play，其主要功能是 StA 根據 DA 所編輯的外部相互作用之腳本內容，提供 DIM 的代理人進行相互作用的依據。基本上，劇本樣板包括“play-name”、“user-number”、“user-order”、“scene-number”、“scene-order”、“time”、“role-number”和“role-skill”八個欄位。

所謂 play-name 欄位係指劇本的名稱，而 user-number 欄位是使用者（UA 與 DA）的數量，user-order 欄位是這些使用者的身分與順序，scene-number 欄位是場景的數量，scene-order 欄位是所有指定議題的名稱與其順序，time 欄位是不同場景進行的時間，role-skill 欄位是不同場景所需要 RA 的設計技能，role-number 欄位是指每一位使用者在不同場景中扮演角色的最大數量。此 play 樣板的語法定義如下：

```

(deftemplate play (slot paly-name) (slot user-number) (multislot user-order)
                  (slot scene-number) (multislot scenes) (multislot time)
                  (multislot role-skill) (multislot role-number)
  
```

)

而在 play 樣板中，這些不同欄位名稱所包含內容的順序，彼此之間有相互對應的關係。例如：在一個名稱為 cyut-studio 的劇本中，使用者包括 DA (DA_{lai}) 與二位 UA (UA_{tengwen} 和 UA_{jesse})，而進行外部相互作用的順序為 UA_{jesse} DA_{lai} UA_{tengwen}，而劇本內容主要包括二個指定議題的場景 (light 和 circulation)，其順序是先 light 而後 circulation。

在 light 場景中，其進行想法連結的時間限制是 20 分鐘，而每一位 UA (包括 DA) 最多只能扮演兩位角色，同時，此場景需有住宅設計 (house) 之設計技能的 RA。而在 circulation 場景中，其進行想法連結的時間限制是 30 分鐘，而每一位 UA (包括 DA) 最多只能扮演三位角色，且此場景需要住宅設計之設計技能的 RA。因此，此劇本的內容可以藉由下列的 play 樣板事實而呈現：

```
(play (paly-name cyut_studio) (user-number 3) (user-order UAjesse DAlai UAtengwen)
(scene-number light circulation) (scene-order light circulation) (time 20 30)
(role-skill house house) (role-number 2 3)
)
```

除了上述的場景樣板外，想法元件樣板也是構成 StA 知識庫的主要資料，另外，ACL 訊息樣板提供 StA 在外部相互作用與 ScA 和 DA 進行訊息傳遞的相關資料。有關這些 StA 樣板的機制與語法相同於上述 RA 工作記憶的樣板，其內容詳見 5.3.1 章節。

5.3.4.2 規則基礎

在 DIM 係統中，StA 負責將 DA 編輯的劇本內容傳遞給不同的 ScA，同時，StA 儲存每一個場景進行相互作用的過程，因此，StA 的規則基礎主要以溝通規則為主。

溝通規則

基本上，StA 的溝通規則主要發生在兩個階段，第一階段是 DA 要求 StA 開始進行外部相互作用，第二階段中則是 DA 要求 StA 將 ScA 產生的想法元件告知所有的 UA。

在第一階段規則的 LHS 中，是當 StA 接收到 ACLMessage 樣板事實之 communicative-act 欄位值是 REQUEST，且 play 樣板事實中 scene-number 欄位值和 user-number 欄位值是某指定之整數值時 (如場景的數量是 2，使用者的數量是 2)，在此規則的 RHS 中，則宣稱 (assert) 一個 play 樣板事實和 ACLMessage 樣

板事實。

在此 play 的樣板事實中，主要根據 play 樣板事實的欄位順序，將原先 ACLMessage 樣板事實 content 欄位之字串值轉換成目錄的 tokens，並放置在此事實的相關欄位中。而在 ACLMessage 樣板事實中，主要將 communicative-act 欄位值與 content 欄位值分別由“INFORM”和“I will start the play”所取代，同時將此訊息內容回傳給 DA。因此，此階段之規則的語法定義如下：

```
(defrule request_when_external_interplay_2_2
  ?m <- (ACLMessage (communicative-act REQUEST_WHEN)
           (sender ?s) (content ?c) (receiver ?r))
  (scene (scene-number 2) (user-number 2))
=>
  (assert (play (play-name (nth$ 1 (explode$ ?c)))
               (user-number (nth$ 2 (explode$ ?c)))
               (time (nth$ 3 (explode$ ?c)) (nth$ 4 (explode$ ?c)))
               (user-order (nth$ 5 (explode$ ?c))(nth$ 6 (explode$ ?c)))
               (scene-number (nth$ 7 (explode$ ?c)) (nth$ 8 (explode$ ?c)))
               (scene-order (nth$ 9 (explode$ ?c))(nth$ 10 (explode$ ?c)))
               (role-skill (nth$ 11 (explode$ ?c))(nth$ 12 (explode$ ?c)))
               (role-number (nth$ 13 (explode$ ?c))(nth$ 14 (explode$ ?c))))))
  (assert (ACLMessage (communicative-act INFORM) (sender ?s) (receiver ?r))
           (content "I will start the play"))
  (retract ?m)
)
```

在上述的規則中，藉由 play 樣板事實之 scene-number 欄位值和 user-number 欄位值來控制 ACLMessage 樣板事實中有關字串轉換成 token 的數量，進而影響 play 樣板事實相關欄位值。在第二階段中，StA 根據 DA 的要求把產生的想法元件傳遞給所有參與的 UA，此規則的語法定義如下：

```
(defrule request_ie_users
  ?m <- (ACLMessage (communicative-act REQUEST) (sender ?s) (content ?c)
           (receiver ?r))
=>
  (assert (ACLMessage (communicative-act INFORM) (sender ?s) (receiver ?r))
           (content "?c"))
  (retract ?m)
)
```

這些接收規則皆經由之前論述的發送規則將相關訊息傳遞給指定的接收者。最後，StA 推理引擎中的 Matcher 藉由相關訊息的模式比對方式，決定此 StA 對目前的工作記憶內容所採取的溝通規則，予以採取適當的行為互動於外部相互作用。有關上述 DIM 系統代理人的知識基礎之主要程式碼詳見附錄 A。

5.4 系統實例

DIM 的系統實例說明主要包括兩個部分：1) 代理人知識獲得與建立，與 2) 角色如何扮演、想法如何連結。由於 RAS、StA 和 ScA 的知識庫是經由 DIM 系統自動產生，因此，代理人的知識獲得與建立，主要強調一套標準化之 RA 知識庫的建立方式，此方式可以提供 DIM 系統資料庫的建立、修正與擴增。此外，作者透過 DIM 系統的實際模擬，予以說明多重參與者在分散式的設計環境中，如何在 DIM 之系統環境中進行角色的扮演與想法的連結。

5.4.1 代理人的知識獲得與建立

基本上，設計知識的獲得主要來自語言與文字的敘述[28]。因此，一個 RA 知識的獲得與建立，主要根據設計工作以找尋相關的早期設計案例，並進行這些案例在早期設計階段中，有關概念性設計知識之文字敘述分析，以建立 DIM 知識庫的基礎。而為了避免過多不同的領域概念語彙描述相同的想法，而造成資料庫建立與維護的困難，進而影響 DIM 系統搜尋與想法連結的流暢性，DIM 必須提供參與者選擇相關建築領域概念語彙的標準，因此，一些相關於設計題目且具專業性建築設計之書籍的領域概念語彙，將提供 DIM 系統建立領域概念語彙的參考依據。例如：以住宅設計為例，一系列 GA HOUSES[158]是有關住宅設計的專業書籍，其包括的早期設計案例可以提供參與者進行案例分析，並建立想法元件中的領域概念語彙。

根據 DIM 的知識呈現概要（議題、概念、形式與喜好），每位參與者對於自己選取之早期設計案例，進行個人對設計案例的解譯、分析與分解，且根據議題、概念與形式的分類，選擇每個屬性值適當的領域概念語彙為關鍵字，以建立這些不同類型屬性之值的關聯性。例如：在住宅設計中，一位參與者為解決有關「動線」(circulation) 設計議題，他藉由上述的設計案例分析，建立了「線型」(linear) 與「連接」(connect) 的設計概念，最後，他使用 Roto 建築師在 Malibu 住宅設計中的「橋」(bridge) 為解決「動線」議題的想法元件之形式值（圖 51）。此圖表將提供此參與者建立一個具關聯性案例知識之 RA 知識庫的基礎。

Issue	Concept	Form
● circulation	● linear, connect a linear bridge to connect the house's living units	● corridor Michael Grecco Hanselmann House
● view	● enclosure, connection, continuous a courtyard is enclosure by creating a continuous connection corridor around a court	● bridge Rolo Architects House in Malibu
● lighting	● public, open creating a public and open space between entrance and urban landscape	● courtyard Alexander Gerkin House of the Glass Spike
● in_between	● opening a big opening for seeing distant mountain view	● garden Tadon Atda Sudakayama House
	● cutting, hole cutting various holes within thick walls for inducing sunlight into interior space	● window Chen-Kun Li DK, G. House
	● private a garden is designed for a private zone between the house and natural landscape	● skylight Tadon Atda Twin House
	● scattering scattering the building into several units for layout	

圖 51. 屬性值之關鍵字的關聯性建立

想法元件建立

依據圖 51 中不同屬性之間的連結，一組有關聯性的三個不同屬性之值與喜好值，共同形成一個想法元件(表 10)，而其中，每一個想法元件中形式值的多媒體資訊，被儲存在共同的伺服器中。

表 10. 一位參與者建立的想法元件

RA1_1	
Issue	view
Concept	public, open
Form	courtyard  http://lai.yuntech.edu.tw/DIM_2/RA1_1
Preference	3

例如：在表 10 中，一個想法元件（稱之為 RA1_1）的議題值為「view」，概念值

為「public」和「open」，形式值為「courtyard」，喜好值為 3。此外，此想法元件形式值的多媒體資訊（一張影像照片）被放置在 http://lai.cyut.edu.tw/DIM/RA1_1 的 URL 中。

ICF 詞典建立

為建立此 RA 的 ICF 詞典，參與者根據這些相關書籍所提供在有關議題、概念與形式的領域概念語彙，進行此 RA 之 ICF 詞典的建立。根據書籍提供的領域概念語彙，參與者各自選擇他們的想法元件所用到的關鍵字（如表 11 中每個屬性類別的右邊）。此外，參與者也可以根據自己的設計經驗，增加其它相同意義的領域概念語彙為關鍵字（如表 11 中每個屬性類別的左邊）。這些成對的關鍵字（詞典目錄）被放置在此 RA 之 ICF 詞典中（有關 RA 之 ICF 詞典和詞典目錄詳見於 5.3.1.1）

表 11. 一位參與者建立的 ICF 詞典

ICF 詞典					
sightseeing	landscape	nil	sunshade	stage	platform
view	landscape	hole	opening	veranda	cloister
sunlight	light	square	open-space	backyard	garden
drying	ventilation	link	connection	patio	courtyard
buffer	in-between	division	partition	hall	corridor
buffer	transition	dispersion	scattering	nil	view-extension

想法地圖建立

在建立 RA 的想法地圖中，參與者則根據自己對於關鍵字（領域概念語彙）的不同解譯，進行三種類型的 ICF 地圖的建構。基本上，參與者在每一種類型的 ICF 地圖中，經由三種不同的屬性連結（屬性相似連結、屬性對比性連結或屬性相鄰連結），建構這些不同關鍵字之間的關聯性（圖 52）。例如在圖 52 的概念地圖中，一個關鍵字“connection”與另一個關鍵字“union”之間存在屬性相似連結，在議題地圖中，一個關鍵字“lighting”與另一個關鍵字“view”之間存在屬性相鄰連結。

上述這些想法元件、ICF 詞典與不同種類的想法地圖建立 RA 主要的知識基礎，而這些知識基礎根據之前論述 JESS 樣板與欄位的建構方式，經由每位參與者自行建立並儲存在一個指定的 JESS 知識庫（一個 clp 的檔案）中。在這些知識庫中，每一個想法元件中形式值的多媒體資訊，必須給予其 URL 位址並存取在一個集中的伺服器（server），經由網路超連結這些 URL 位址，而連結到一個想法元件中形式值的多媒體資訊。當上述的想法元件、ICF 詞典與 ICF 地圖儲存在 JESS 的知識庫

後，RA 的角色 IE 地圖可以經由 DIM 系統自動產生，並儲存在相同的 JESS 知識庫中。此外，參與者可以使用上述相同的方式，建構其它 RA 的知識基礎（有關建構 RA 知識庫 clp 檔之實例詳見附錄 B）。

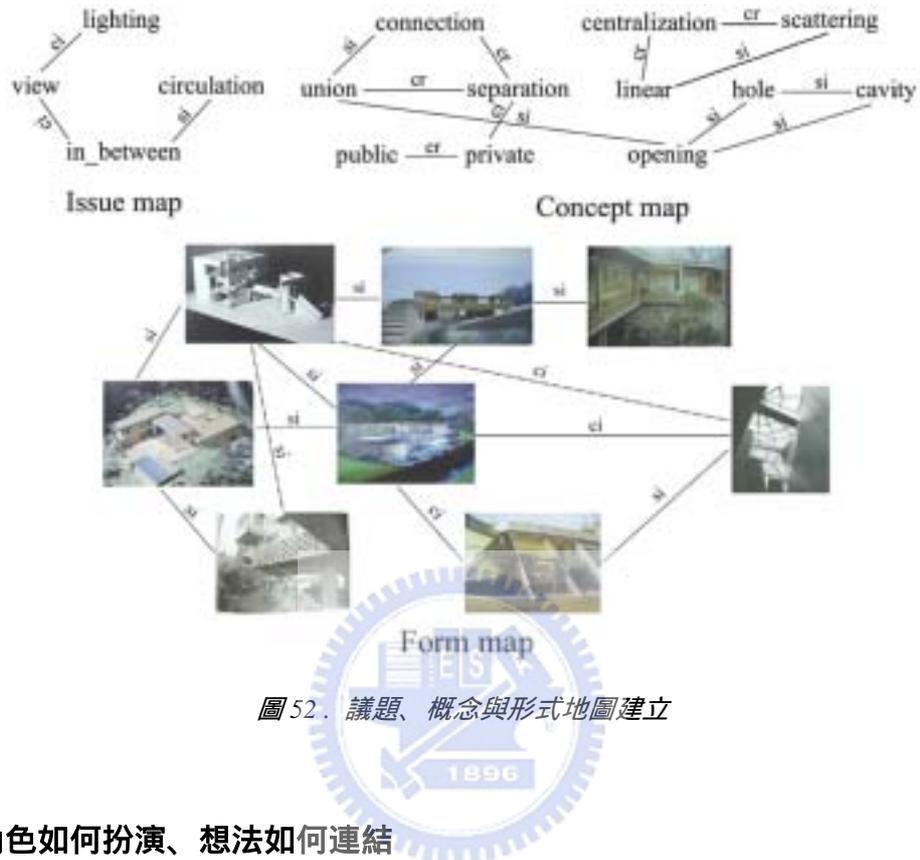


圖 52. 議題、概念與形式地圖建立

5.4.2 角色如何扮演、想法如何連結

在 DIM 的系統環境中，多重知識實體為了一個特定設計工作而進行的分散式想法連結，可以被視為一場角色扮演的遊戲（以下簡稱遊戲）。為了解釋 DIM 系統如何輔助參與者在一個分散式的設計環境中進行想法聯想，本研究透過一個實際的案例模擬，以說明角色如何扮演？又想法如何連結？。

在此實際案例中，設計工作是一個長條街屋的住宅設計，其所需要的設計技能是住宅設計（house design），而採光（daylight）與動線（circulation）依序是指定的設計議題，且分別進行 10 分鐘與 15 分鐘的想法連結。參與者分別是 UA_{lai} 、 UA_{chang} 和 UA_{joe} ，這三位設計者分別在不同的地理位置，使用各自電腦中的 DIM 系統，並依照 UA_{joe} 、 UA_{chang} 和 UA_{lai} 產生想法的順序，進行分散式想法連結遊戲。

此外，腳本內容的編輯包括兩個部分：外在相互作用與內部相互作用。內在相互作用的腳本內容經由 DA 與 UA 各自進行編輯，但只有 DA 可以編輯外在相互作用的腳本內容。在此想法連結的遊戲中， UA_{lai} 為導演代理人（DA），而 UA_{chang} 和 UA_{joe} 為使用者代理人（UA），因此， UA_{lai} 負責編輯外在相互作用的腳本內容，同

時， UA_{lai} 、 UA_{chang} 和 UA_{joe} 編輯他們各自內在相互作用的腳本內容。

基本上，每個分散式想法連結的遊戲必須經由四個步驟，包括 1) 啟動遊戲、2) 編輯腳本、3) 執行遊戲、4) 終結遊戲。而每一個步驟皆需參與者，經由 JADE 代理人環境中的使用者介面[156]輸入必要的訊息內容，以利此想法連結遊戲的進行。有關此設計實例在這四個步驟的詳細說明如下：

1. 啟動遊戲：啟動遊戲主要包含載入代理人與連結代理人平台兩個步驟。在載入代理人的步驟中，每位設計師在各自 JADE 代理人平台中的載入代理人視窗（圖 53），將所有參加此遊戲的相關代理人（以 UA_{lai} 為例， UA_{lai} 必須載入 StA、二個 ScA、RAS 與欲參與遊戲的演員），分別輸入這些代理人的名稱與其 Java class 的路徑而載入。例如：當 UA_{lai} 載入 StA 時， UA_{lai} 須經由此視窗載入一個 StA 的代理人名稱，與此代理人 Java class 的路徑（DIM_2.StAgent）



圖 53. 載入代理人視窗

在連結代理人平台步驟中， UA_{lai} 、 UA_{chang} 和 UA_{joe} 可以透過 HTTP 的通訊協定，在各自的代理人平台中增加其他設計者的遠端代理人平台，以進行分散式環境中不同代理人平台的訊息傳遞。例如， UA_{lai} 藉由新增代理人平台視窗（圖 54）載入 UA_{joe} 之代理人平台中 AMS 的名字（ams@joe:1099/Jade）與其住址（http://joe:7778/acc）（7778 是預先設定的伺服器連接埠），相同的， UA_{lai} 可以使用相同的方法連結 UA_{chang} 之代理人平台。

最後， UA_{lai} 可以在 JADE 代理人管理視窗看到本身載入的代理人，與其他設計者之代理人平台的名稱與其載入的代理人，例如：在 UA_{lai} 的代理人管理視窗中，除了在 Main-Container 有 UA_{lai} 自己載入代理人外，（包括 StA@lai:1099/JADE、ScA1@lai:1099/JADE、ScA2@lai:1099/JADE、RAS@lai:1099/JADE、RA1@lai:1099/JADE 等），同時，也看到 UA_{chang} 和 UA_{joe} 的代理人平台的名稱（chang@1099/JADE 和 joe@1099/JADE）（圖 55），與各自代理人平台載入的代理人，例如在 joe@1099/JADE 代理人平台中的 RAS@joe:1099/JADE、RA1@joe:1099/JADE、RA2@joe:1099/JADE。



圖 54. 新增代理人平台視窗

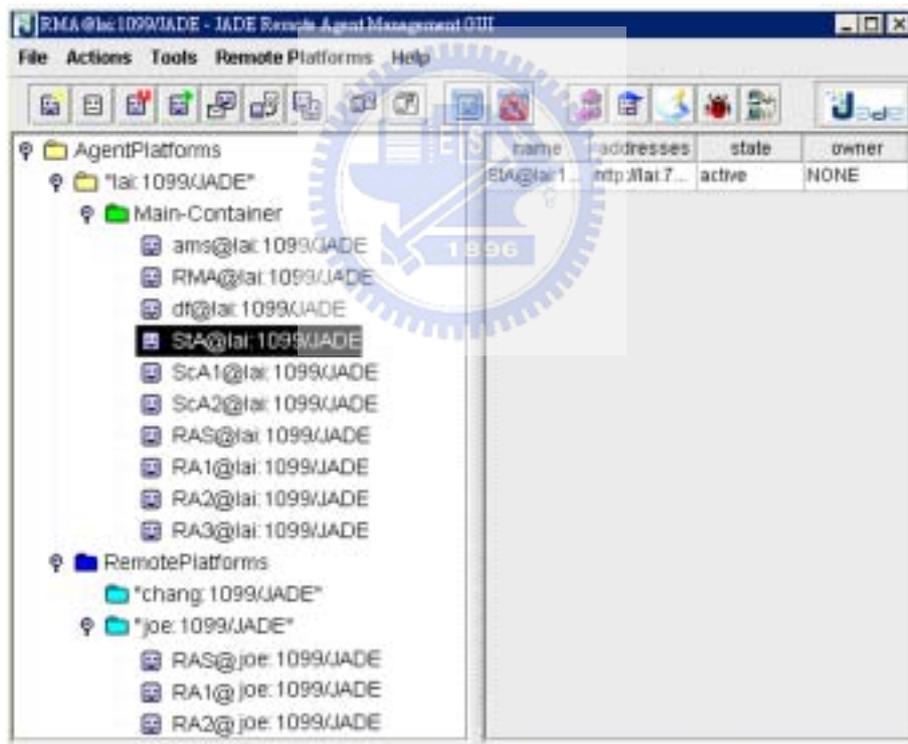


圖 55. 代理人管理視窗

- 編輯腳本：當載入代理人與連結代理人平台完成後， UA_{lai} 必須在 JADE 編輯腳本視窗（圖 56）中編輯外部相互作用的腳本內容，包括 StA 與所有 ScA 相關資訊內容的輸入。在輸入 StA 的資訊中， UA_{lai} 首先在編輯腳本視窗的接收者表格（Receiver）中，輸入 StA 的 AID（ $StA@lai:1099/JADE$ ），並在訊息內容（Content）表格輸入 StA 的訊息內容（依序為遊戲名稱、所有場景議題與

演出時間)。在圖 56 的例子中，遊戲名稱是 row_house，場景議題是 daylight 和 circulation，演出時間是 10 分鐘與 15 分鐘，當 UA_{lai} 輸入上述資訊完成後，UA_{lai} 按下傳遞訊息指令並將此訊息內容儲存至 StA 的知識庫後，UA_{lai} 繼續編輯二個 ScA (ScA₁ 和 ScA₂) 的資訊內容。

在輸入 ScA 的資訊中 (以 daylight 的場景 ScA₁ 為例)，UA_{lai} 同樣在 JADE 編輯腳本視窗的接收者表格中輸入 ScA₁ 的 AID，並在訊息內容表格輸入 ScA₁ 的訊息內容 (依次為議題名稱、演出時間、角色數量、設計技能)。此外，經由 UA_{lai} 代理人平台中 DF 的 Yellow Pages 服務，ScA₁ 能獲知 UA_{lai}、UA_{chang} 和 UA_{joe} 代理人平台中具有“house design”設計技能之演員的 AID，以提供使用者選擇並輸入適合演員的身分。當輸入演員身分完成後，所有設計者則在各自的 JADE 代理人平台中進行 ScA₁ 的遊戲。

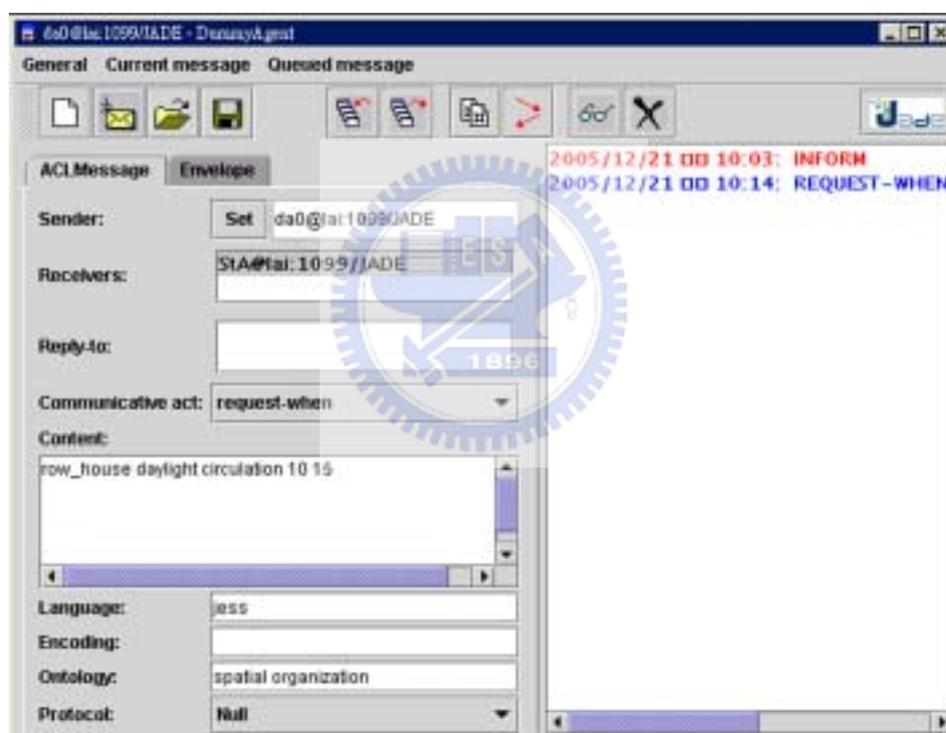


圖 56. 編輯 StA 腳本視窗

3. 執行遊戲：根據採光 (daylight) 的議題，排序第一位的設計者 UA_{joe} 首先在 JADE 編輯腳本視窗 (圖 56) 輸入 RAS 的 AID 與其它相關訊息內容 (包括議題、概念、形式、時間、角色數量、AID、連結原則、容許限制)，當 UA_{joe} 輸入完成且按下傳遞訊息的指令，則存活在 UA_{joe} 代理人平台中的不同 RA 開始進行內部相互作用，並經由它們之間的訊息傳遞而產生內部 IE 地圖 (如圖 57)。

在上述 RAS 的輸入時間內， UA_{joe} 可以在產生的內部 IE 地圖中選擇一個想法元件為輸出，或當時間超過時則由電腦自動的選擇一個想法元件為輸出，這些輸出的想法元件（包括 URL）與元件連結會結合原有的外部 IE 地圖。而下一位設計者 UA_{chang} 須在此外部 IE 地圖自行選擇一個想法元件為輸入值，或當時間超過時則由電腦自動的選擇一個想法元件為輸入值，並進行上述相同於 UA_{joe} 之內部相互作用的操作模式。

4. 終結遊戲：當 ScA_1 場景時間結束時， UA_{lai} 、 UA_{chang} 和 UA_{joe} 以相同於步驟三的操作過程，進行 ScA_2 在動線（circulation）議題的想法連結。當最後一個場景的時間終止時，此想法連結遊戲會自動終結。除此之外，想法連結遊戲的終結也可以經由導演代理人 UA_{lai} 強迫終止而達成。

這些產生的 IE 地圖（包括外部與內部）可以藉由 GraphVIZ 的圖像視覺軟體予以呈現，而存在於這些 IE 地圖中之想法元件的多媒體資訊（包括如早期設計案例的影像圖片、3D 模型、平面圖或動畫等），可以經由這些想法元件的 URL 而超連結到網路伺服器而獲得（圖 57）。

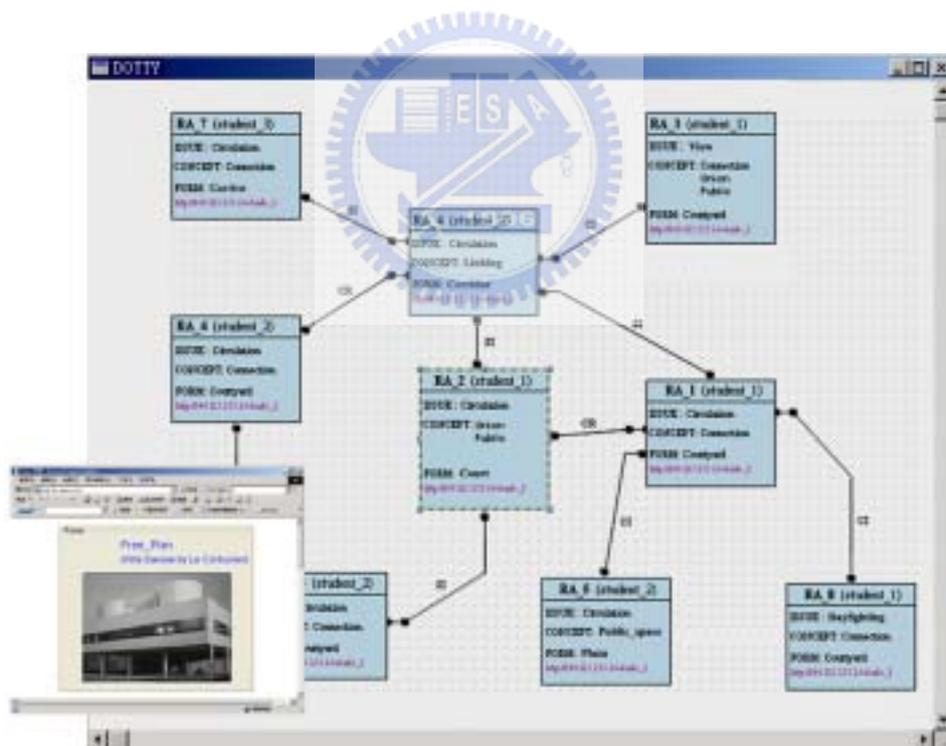


圖 57. 一個 IE 地圖（外部或內部）與想法元件的多媒體資訊

這些產生的外部 IE 地圖，都被儲存在遊戲名稱為 row_house 的檔案目錄中（ StA 的知識庫），而透過載入此 StA 知識庫中單一或全部場景（Scene1 和 Scene2）的連結過程（圖 58），可以提供 UA_{lai} 重新利用 StA 的知識庫，予以繼續沿用並重新啟動一個新的想法連結遊戲。此外， UA_{lai} 、 UA_{chang} 和 UA_{joe} 也可以在各自的 RAS 知

識庫中，重新利用不同時間下所儲存的內部相互作用之想法連結過程，而這些內部相互作用之想法連結過程，可以經由一個稱為 Sniffer 的使用者介面而顯示（圖 59）。

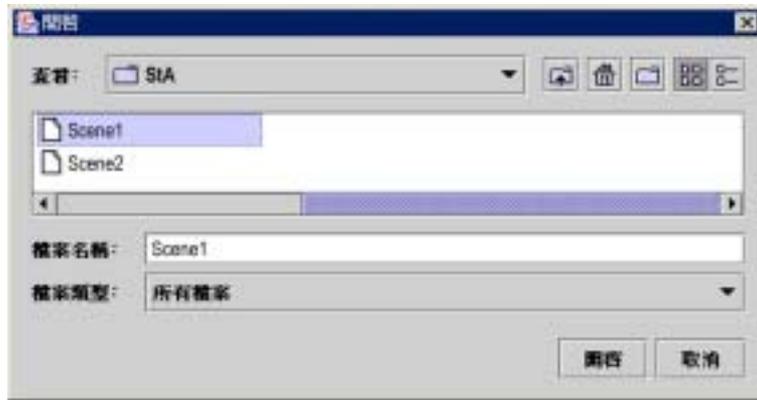


圖 58. 載入連結過程視窗

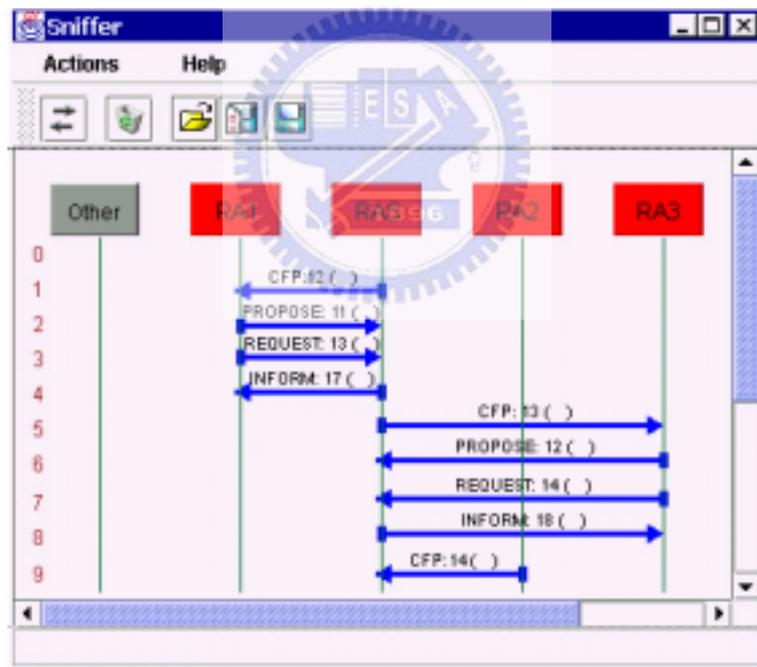


圖 59. 內部相互作用連結過程顯示介面

六、DIM 系統之連結有效性評估

研究的假設必須經由適當的方式而予以驗證。因此，除了本研究的第一個假設「想法聯想的分散式連結機制，可以經由運算化的機制進行了解並建構」，已經由 DIM 系統的實作而獲得驗證外（詳見第五章），在本章中，主要針對本研究的第二個假設「在分散式的設計環境中，發展的數位工具可以輔助多重參與者產生有效性的連結」進行驗證。為了驗證 DIM 系統可以提供連結有效性，以利參與者進行分散式想法連結，本研究經由建立一個適當的評估方式，並實際執行一個設計實驗，進行 DIM 系統之連結有效性評估。

根據第一章有關想法聯想之連結問題特質的論述，立即性強調此連結問題在視覺回饋的介面需求，反射性與關聯性強調連結問題在想法與其關係的連結建立，衝突性則強調多重知識實體在知識改變之運算機制。因此，本研究的評估重點在於有效性連結在反射性與關聯性之特質的評估，而有關立即性特質，本研究會在實驗過程中特別注意，以提供未來介面設計需求的依據，而在設計實驗中的發現，也將提供本研究進一步精練衝突性在相關於角色學習之運算機制。除此之外，此設計實驗將提供本研究更進一步了解 DIM 系統，如何輔助參與者進行分散式想法連結，又如何使用連結過程中產生的想法地圖。

6.1 評估方式與內容

在進行 DIM 系統的設計實驗之前，首先就評估方式與評估內容進行說明，經由評估方式與評估內容的探討，建立本研究的設計實驗方式，以利進行 DIM 系統在連結有效性的評估。

6.1.1 評估方式

Osborn[11]認為想法聯想是「量產生質」的行為，此行為是一種強調藉由生產大量的想法，進而提高解決問題方法的品質，在設計領域中，這種強調生產數量的設計活動，Goldschmidt[14,46]稱之為設計生產（design productive），而有效性（effectiveness）為評估設計生產過程的主要方式。為了評估設計生產過程之有效性，Goldschmidt[46]使用連結圖像（linkography）理論，在個人和團隊基礎¹（team-based）的設計生產過程中，進行有關不同設計想法與設計想法之間關聯性建立的研究。

基本上，連結圖像理論主要在 *移動*（move）（或稱為想法）與 *連結*（link）所構成

¹。團隊基礎（team-based）為結合二人以上的參與者，共同參與集體腦力激盪會議的團隊組織（Goldschmidt, 1995）。

的知識網路結構（或稱想法網路結構），紀錄設計移動（design moves）之間的連結，以了解設計生產過程中移動與連結的相互共存關係（correlates）[14,46,159]。在連結圖像理論中，一個設計移動為“...一個轉變（transforms）設計情境的步驟、且行動或操作，此設計情境相對於它之前移動的狀態”，而連結是產生這些設計移動的推理模式，經由反射性的連結機制改變設計移動之狀態，並建立狀態之間的關聯性。因此，設計生產過程強調有效性的連結建立，而有效性的連結必須建構在設計想法與連結的相互共存關係。

架構在連結圖像理論的基礎下，Goldschmidt[14]提出連結指標（Link Index，簡稱LI）的量化分析方式，以進行上述相互共存關係在設計生產過程中之有效性評估。在此量化分析方式中，LI 數值與設計生產過程的有效性成正比關係，換言之，當LI 數值越高，設計生產過程的有效性越高，相對的，當LI 數值越低，設計生產過程的有效性越低。由於，本研究的評估重點是在連結的有效性，同時，此連結的有效性是建立在一個分散式想法連結的設計生產過程中，因此，連結指標（LI）提供本研究評估 DIM 系統在連結有效性的主要方式。

此外，經由想法聯想所產生的想法必須「建立在其他想法的基礎上」[11]，van der Lugt[13]認為連結指標的數值，同時也反應參與者在一個關聯性的想法網路結構中，尋求和其他設計想法進行組合的評估指標，簡言之，一個想法網路結構之連結指標數值越高，參與者越容易進行設計想法的組合。因此，van der Lugt[13]除了使用連結密度²（Linking Density）的術語取代連結指標外，他也提出連結形式指標（Link Type Indexes，簡稱LTI），此指標主要是應用在經由想法組合後的想法網路結構中，其透過量化分析想法之間不同關聯性的連結分佈，而了解想法網路結構的功能。

綜合以上的論述，連結指標（LI）除了提供本研究進行連結有效性的評估方式外，同時，連結指標也提供本研究判斷並選擇合適的設計結果（外部IE地圖），以利想法組合的進行。此外，在一個經由想法組合的外部IE地圖範圍中（圖60），連結形式指標（LTI）量化三種不同形式的元件連結（相似元件連結、對比元件連結與相鄰元件連結）在此地圖範圍的分佈，此指標的數值有助於本研究了解這些連結形式在想法組合的功能。

根據設計共鳴現象[104]，在每一個場景中，其所產生的關鍵想法元件與其他想法元件進行組合的方式，由於牽涉到想法元件在選擇頻率的關係，而有助於本研究了解 DIM 系統在衝突性的學習機制。除了上述的連結形式指標外，設計共鳴現象也提供本研究了解關鍵想法元件，如何藉由不同連結形式進行想法組合，為了解

²根據 van der Lugt（2002）的定義，連結密度係指在想法地圖的區域範圍中，連結的所有數量除以（divide）想法的所有數量之值。

想法組合的方式，關鍵想法元件與其他想法元件之間的連結路徑長度 (Linking Path Length, 簡稱 LPH) 扮演重要的角色。有關連結指標、連結形式指標與連結路徑長度的定義與相關說明，在下面章節有更進一步的說明。

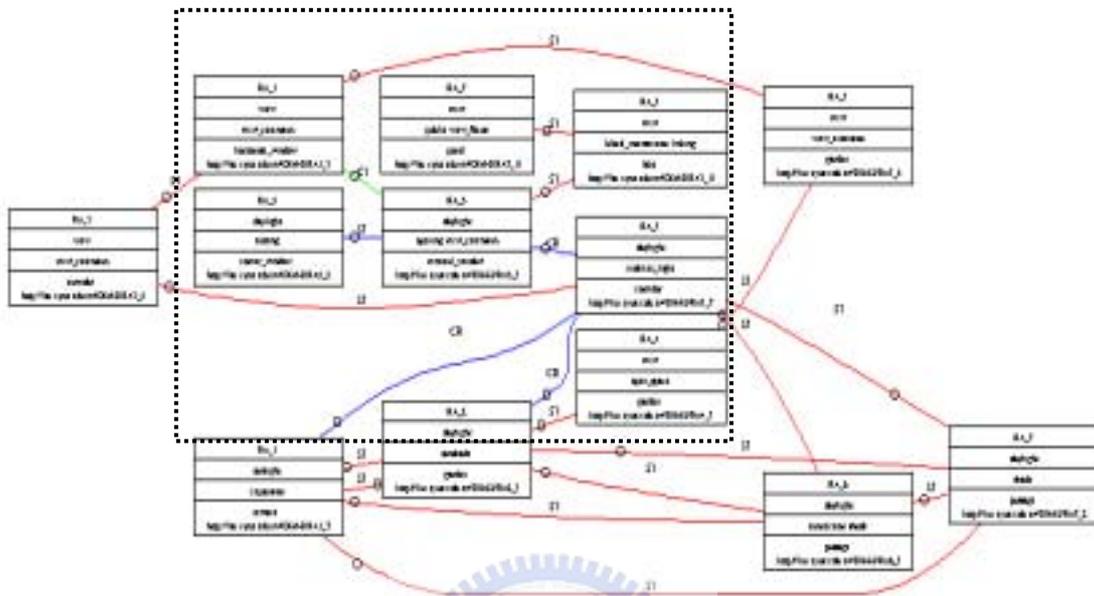


圖 60. 一個外部 IE 地圖與想法組合的範圍

6.1.1.1 連結指標

連結指標係指在想法網路結構中，所有連結 (link) 的數量除以 (divide) 所有設計移動 (move) 的數量之值[14]。在本研究的評估中，其連結指標是指在IE地圖中 (主要為外部IE地圖)，三種不同形式之元件連結的總數量除以想法元件的總數量，其數學算式如下：

$$LI = \frac{\text{元件連結總數量}}{\text{想法元件總數量}}$$

例如：在圖60的IE地圖 (以IE₁為例) 中，當連結的總數量是22，而想法元件的總數量是13時，則IE₁的LI之值為1.69 ($\frac{22}{13} = 1.69$)。根據之前對於LI數值的定義，當一個IE (稱之為IE₂) 地圖之LI數值大於1.69時，則IE₂的連結有效性高於IE₁的連結有效性，比較於IE₁，IE₂較有利於參與者進行設計想法的組合；然而，當IE₂的LI數值小於1.69時，則IE₂的連結有效性低於IE₁的連結有效性，比較於IE₁，IE₂較不

易於參與者進行設計想法的組合。

6.1.1.2 連結形式指標

連結形式指標 (LTI) 係指在一個想法網路結構的範圍中，某一種類型之連結的數量除以 (divide) 所有類型連結數量之值[13]。在本研究的評估中，連結形式指標是指一個經由想法組合而被選取的外部 IE 地圖範圍，某一種形式的元件連結數量除以 (divide) 所有形式元件連結的總數量之值。基本上，DIM 系統主要包括三種不同形式的連結形式指標：相似連結指標(簡稱 LTI_{si})、對比連結指標(簡稱 LTI_{cr}) 與相鄰連結指標(簡稱 LTI_{ci})，其數學算式如下：

$$LTI_{si} = \frac{\text{相似元件連結數量}}{\text{元件連結總數量}}$$

$$LTI_{cr} = \frac{\text{對比元件連結數量}}{\text{元件連結總數量}}$$

$$LTI_{ci} = \frac{\text{相鄰元件連結數量}}{\text{元件連結總數量}}$$

例如：在圖 60 之外部 IE 地圖的範圍中（虛線範圍），其關鍵想法元件是 RA1 產生的想法元件 (daylight, indirect_light, corridor)，而想法組合的想法元件數量是 7，相似元件連結數量是 2，對比元件連結數量是 2，相鄰元件連結數量是 1，元件連結的總數量是 5，因此，此地圖範圍的 LTI_{si} 、 LTI_{cr} 和 LTI_{ci} 之值分別為 0.4 ($\frac{2}{5}=0.4$)、0.4 ($\frac{2}{5}=0.4$) 與 0.2 ($\frac{1}{5}=0.2$)。在這三種不同形式之 LTI 值中，當 LTI_{si} 越高時，則代表參與者使用較多的相似元件連結進行想法組合，而當 LTI_{cr} 越高時，則代表參與者使用較多的對比元件連結進行想法組合，另外，當 LTI_{ci} 越高時，則代表參與者使用較多的相鄰元件連結進行想法組合。

6.1.1.3 連結路徑長度

架構在設計共鳴現象，連結路徑長度 (LPL) 係指經由想法組合而選取的外部 IE 地圖範圍中，一個想法元件與關鍵想法元件之間最短的路徑長度，也就是這二個想法元件之間最少的元件連結數量。因此，在一個因想法組合而選取的地圖範圍中，當一個想法元件直接與關鍵想法元件相連結時，此想法元件之 LPL 數值為 1，當一個想法元件必須經由另一個想法元件，才能連結到關鍵想法元件時，此想法元件之 LPL 數值為 2，以此類推。例如：在圖 60 IE 地圖框選的範圍中，當關鍵想法元件是 RA1 產生的想法元件 (daylight, indirect_light, corridor)，則 RA3 產生想法元件 (RA3_1) 的 LPL 是 1，而 RA3 產生想法元件 (RA3_3) 的 LPL 是 2。

因此，當 LPL 數值越高時，代表想法組合之想法元件與關鍵想法元件的連結路徑長度越遠，相對的，兩個想法元件的關聯性越差，當 LPL 數值越低時，代表想法組合之想法元件與關鍵想法元件的連結路徑長度越近，相對的，兩個想法元件的關聯性越好。經由 LPL 數值的分析，本研究可以了解參與者在外部 IE 地圖進行想法組合的方式，與 ICF 地圖轉化的角色學習機制，以提供 DIM 系統在未來的後續研究中，能自動化的產生想法組合的地圖範圍，並進行角色代理人之間的學習。

6.1.2 評估內容

為了進行連結有效性之評估，本研究使用兩組以團隊基礎的受測者，在相同的設計條件下（包括如相同的設計議題、參與人數、參與時間等）進行此設計實驗。在這兩組團隊中，第一組團隊以面對面的方式進行想法連結，而另一組團隊則是使用 DIM 系統進行分散式的想法連結。當實驗完成後，本研究將這兩組團隊所產生的外部 IE 地圖進行連結指標之比較，當其產生的外部 IE 地圖之連結指標數值越高時，則代表此團隊的連結有效性也越高。

此外，為了解參與者使用 DIM 系統進行想法組合的應用與影響，在有關想法組合的分析中，主要透過觀察使用 DIM 系統之團隊的受測者，如何在他們產生的外部 IE 地圖中，進行有關設計共鳴現象中之關鍵想法元件的想法組合，並藉由連結形式指標（LTI）與連結路徑深度（LPL）予以分析，而此分析中的發現，將提供本研究精練 DIM 系統在分散式想法連結中有關衝突性的學習機制。

6.2 設計實驗：設計工作室的集體腦力激盪會議

設計工作室學習是強調設計導向且小組教學的學習方式，在建築設計教育扮演重要的角色。在早期設計階段中，設計工作室學習鼓勵學生產生多樣性的設計想法，予以創造性的解決設計問題，而為了產生多樣性的設計想法，使用集體腦力激盪會議並結合想法聯想的行為，經常被應用在設計工作室學習中[27,101,109]。在此學習過程中，設計工作室的老師通常會使用早期設計案例，啟發學生進行設計創造，同時，經由學生彼此之間的集體腦力激盪式的討論產生多樣性設計想法，且這些設計想法相互連結後，發展成可行的設計替選方案。綜合上述，設計工作室學習的方式與過程反映本研究之連結問題，因此，本研究選擇建築系之設計工作室學習為本設計實驗的對象。

6.2.1 設計工作

在這個設計實驗中，設計題目是長條街屋的住宅設計，其基地位置坐落於一台灣某歷史廟埕區的傳統老街之 4.5m x 30m 之長條街屋基地，各面臨 8 米與 6 米之街道；此基地面臨兩側街道，包括面對一條 15 米的河溝（6 米側）與 3.6m「亭仔腳」

之傳統老街（8 米側），其餘兩側與鄰房相連接（圖 61）。

此設計工作主要在處理有關於空間組織的設計問題（如動線、採光與景觀），並強調發展早期概念階段的設計想法。在進行設計實驗之前，我們準備一個有關住宅設計的案例資料庫提供給參與者使用，參與者必須先對資料庫的案例進行了解，同時，在想法聯想的過程中，他們被強迫使用這些資料庫的案例產生設計想法。

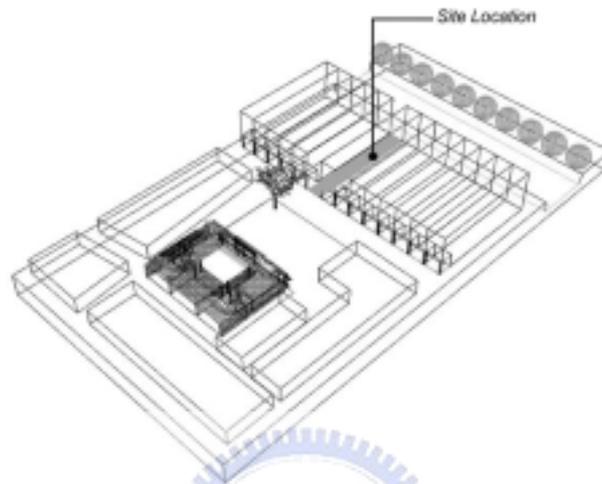


圖 61. 設計實驗的設計工作：單一家庭的長條街屋住宅設計

6.2.2 實驗受測者

DIM 系統主要輔助具有建築領域專業知識的設計者進行分散式的想法連結，因此，在選擇設計工作室的學生條件，除了需具有二年以上設計工作室的經驗外，同時必須學習過建築史與近代建築史等相關理論的課程，以具備熟悉早期設計案例的能力與其相關的背景。為了使設計實驗的過程簡單且容易觀察並紀錄，以及確保想法連結過程的流暢性，這些學生曾經在相同的設計工作室進行設計學習，具有良好的默契且熟悉腦力激盪會議的技術，且皆都擁有相似的建築設計領域知識、設計能力與繪圖的熟練技法。

根據上述的條件，本研究在朝陽科技大學建築系三年級的學生中，選擇同一個設計工作室的學生參與此設計實驗。此設計實驗包括六位學生（包括 A、B、C、D、E 與 F），每位學生具有上述論述的相同設計背景，包括如先前在一二年級具有相同的建築設計訓練，並曾是作者在二年級之設計工作室學生，同時，這些學生具有相類似的設計領域知識，且熟悉先前論述之早期設計案例的分析方式，並經常使用建築領域概念語彙去描述設計想法。

為了評估 DIM 系統在設計生產之連結有效性，本研究將這六位學生分成二組（Team

1 和 Team 2), Team 1 (學生 A、B、C) 為學生面對面的參與想法聯想, Team 2 (學生 D、E、F) 是使用 DIM 系統進行分散式的想法連結。每一組各有三位學生, 除了反應分散式想法連結機制外, 同時, 有利於本研究在實驗過程中的觀察。

6.2.3 實驗過程

為確保此設計實驗的流暢性與正確性, 參與實驗的學生必須熟悉想法元件、想法地圖、ICF 地圖與 ICF 詞典的分析與建立, 與 DIM 系統的使用操作 (討論在第五章節)。因此, 在進行此實驗之前, 對於參與實驗的學生進行 DIM 模型的案例分析方法講解, 且確認每位學生都能熟悉使用 DIM 模型的知識呈現方式進行設計案例的分析後, 進行為期一個星期的住宅設計之早期設計案例收集與分析。

在實驗前的階段中, 設計工作室的老師 (熟悉 DIM 的知識呈現方式) 須指導這六位學生進行早期設計案例分析, 這些學生整理出九組不同的知識基礎 (包括角色 IE 地圖、ICF 地圖等), 分別給予九位角色代理人的記憶組織 (詳見 5.4.1 章節)。為使用 DIM 系統的流暢性, Team 2 的學生們必須熟悉 DIM 系統的資料輸入與輸出, 與 JADE 代理人平台相關的使用者介面與操作步驟。

除此之外, 這兩組學生的工作環境、集體腦力激盪會議的流程、與角色扮演遊戲的腳本內容必須先進行說明, 其目的是兩組學生可以在相同的設計條件中進行想法連結, 以利本研究對於這兩組學生產生的設計結果, 進行連結有效性的評估與比較。

6.2.3.1 工作環境

為求設計實驗過程的流暢與容易觀察, 這兩組學生依據產生設計想法的設計工具, 被安排在不同的工作環境進行設計實驗。在 Team 1 的工作環境中, 學生 A、B 和 C 在同一間製圖教室的一張大型繪圖桌進行面對面的想法聯想 (圖 62, 左側) 根據腳本內容, 每位學生各自有不同角色代理人的想法元件卡 (表 10), 提供他們扮演不同角色進行想法連結, 同時, 根據不同的相互作用, 將他們產生的想法元件卡與連結紀錄在 A4 與 A0 的紙張, 以建立內部與外部相互作用產生的 IE 地圖 (圖 57)。在實驗過程中, 除了全程有攝影機紀錄外, 作者也在旁邊指導並進行觀察。

在 Team 2 的工作環境中, 學生 D、E 和 F 在同一間電腦教室, 分別使用三台安裝 DIM 系統的電腦進行分散式想法連結, 為避免學生相互干擾與容易進行實驗的觀察, 每位學生相隔一部電腦, 並依據 L 型的配置使用桌上的電腦 (圖 62, 右側)。在實驗過程中, 除了全程有攝影機紀錄外, 同時作者扮演一個協助的角色, 隨時在不同參與學生的工作空間, 進行學生在操作 DIM 系統的觀察與輔助。



圖 62. 設計實驗的工作環境：Team 1（左側）和 Team 2（右側）

6.2.3.2 會議流程

在進行設計實驗之前，實驗的腳本內容（包括如設計議題、參與時間、產生想法順序等）必須事先向學生清楚說明與界定，同時，這兩組學生都必須進行暖身實驗。在完成暖身實驗後，這二組學生的設計實驗分別進行 25 分鐘的集體腦力激盪會議，而為了能在有限的時間內產生大量的設計想法，參與的學生被鼓勵產生設計想法，而不對任何產生的設計想法進行批評。

當限定的時間結束時，二組的每位學生必須在每一個場景（指定議題）所產生的外部 IE 地圖，選擇不同的想法元件進行尋求想法組合，並發展他們各自的替選方案，因此，整個會議的過程所包含的步驟詳見表 12。這些產生的想法地圖與替選方案都必須被收集起來，以作為未來分析與系統評估之用。

表 12. 集體腦力激盪會議的流程

圖 回	1. 暖身實驗
	2. 腳本內容說明，包括設計時間、設計議題、產生想法順序等
	3. 會議時間設定（25 分鐘）
	4. 產生設計想法，且不對任何產生的想法進行批評
	5. 選擇相關的先前想法或設計案例
	6. 產生設計想法，且不對任何產生的想法進行批評
	7. 當限定的時間終了，收集兩組學生產生的外部 IE 地圖，進行連結有效性的評估分析。

6.2.3.3 腳本內容

為了進行設計實驗的比較和評估，這兩組學生的腳本內容都必須相同。此設計實

驗的舞台是解決住宅設計有關空間組構的設計問題，且為了解在想法連結過程，不同設計議題之間的關係，此舞台包括二個場景，其場景的議題之先後順序分別為採光（daylight）與動線（circulation）。每一個場景的設計議題（採光和動線）必須在一定的時間限制內完成，分別為採光 15 分鐘與動線 10 分鐘，此外，在這兩組學生中，每一組都有一位學生扮演導演的身分，除了必須負責控制舞台中每一個場景的過程外，也必須與其他兩位參與者進行想法連結（表 13）。

以表 13 為例說明，舞台為住宅設計之空間組構，場景的順序為光線（daylight）動線（circulation），其時間分別為 15 分鐘和 10 分鐘，同時，Team 1 的導演（DA）為 A, B 和 C 為使用者（UA）；Team 2 的導演（DA）為 D, E 和 F 為使用者（UA）。Team 2 在光線場景之外部相互作用中，產生想法的順序為 D、E、F，而在學生 E 的內部相互作用中，其扮演角色之產生想法順序為 Role4、Role5、Role6，這三位角色同時被分別賦予相似原則、對比原則與相鄰原則。

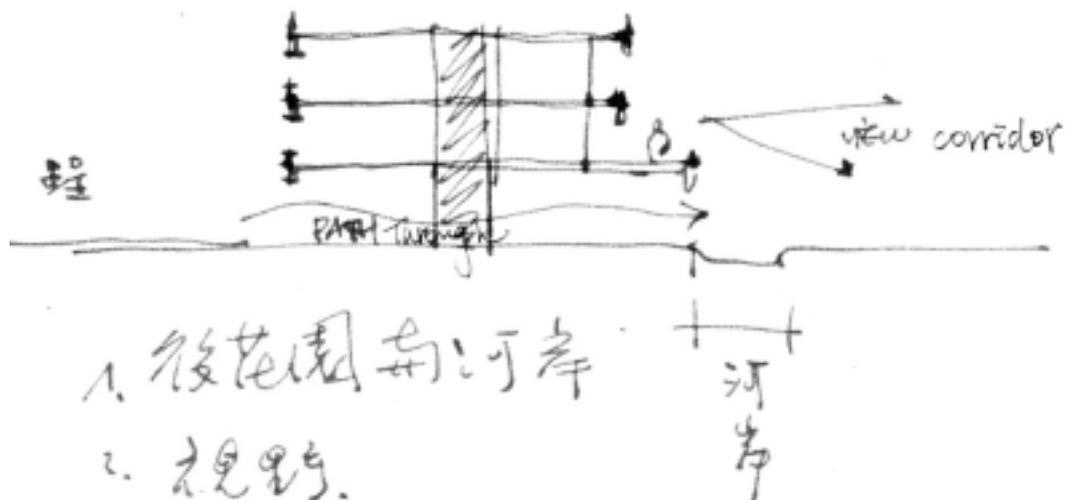
表 13. 設計實驗的腳本內容

舞台	住宅設計之空間組構				
場景	光線（daylight）		動線（circulation）		
時間	15 分鐘		10 分鐘		
相互作用	外部	內部	外部	內部	
參與演員與角色 和產生想法順序	Team 1: A (DA)	Role1 (相似原則)	Team 1: C (UA)	Role9 (對比原則)	
	Team 2: D (DA)	Role2 (對比原則)	Team 2: F (UA)	Role8 (相鄰原則)	
		Role3 (相鄰原則)		Role7 (相似原則)	
	Team 1: B (UA)	Role4 (相似原則)	Team 1: A (DA)	Role3 (相似原則)	
	Team 2: E (UA)	Role5 (對比原則)		Team 2: D (DA)	Role1 (對比原則)
		Role6 (相鄰原則)		Role2 (相鄰原則)	
	Team 1: C (UA)	Role7 (相似原則)	Team 1: B (UA)	Role5 (對比原則)	
	Team 2: F (UA)	Role8 (對比原則)		Team 2: E (UA)	Role4 (對比原則)
		Role9 (對比原則)			Role6 (相似原則)

6.3 評估與分析

為了進行連結有效性的分析與評估，在 Team 1 的實驗過程中，每位學生根據產生想法的順序，在每一次各自的內部相互作用中，將聯想到的想法元件卡固定在 A4 的紙張（內部 IE 地圖），在限定的時間內，選擇的一個想法元件放置在固定在 A0 的紙張（外部 IE 地圖），同時，每位學生在這些 IE 地圖中產生的想法元件間，根據 DIM 之知識呈現概要，使用三種不同顏色的麥克筆在這些想法元件之間，繪製三種不同類型的元件連結，並標示此連結的連結形式。在 Team 2 的實驗過程中，每位學生在各自電腦中 DIM 系統的使用者介面，依序輸入相關的訊息內容，而 DIM 系統會自動的產生不同的 IE 地圖、想法元件與元件間的元件連結，而這些資訊皆可以從參與學生各自電腦中的檔案而獲得。

此外，為了解 DIM 系統如何輔助學生在想法組合之應用及影響，與連結問題之衝突性在學習上的運算機制，Team 2 的學生在每一個場景產生的外部 IE 地圖中進行想法組合。根據設計共鳴現象，每位學生使用產生的外部 IE 地圖進行想法組合，並經由想法草圖的呈現（圖 63），各自發展整組共同的設計方案，同時，這些學生必須在此外部 IE 地圖中，框選此設計方案所使用到的地圖範圍（圖 63），並標示使用到的元件連結與想法元件，以利本研究進行後續的分析。例如：在圖 63 中，學生 E 根據景觀議題之場景的外部 IE 地圖，所繪製有關解決景觀設計問題的想法草圖，與其使用到的外部 IE 地圖範圍，和使用到的元件連結與想法元件。此外，Team 2 的每一位學生在一定的時間限制下（20 分鐘），被要求發展兩個以上不同的設計替選方案，以求設計實驗樣本分析的客觀性。



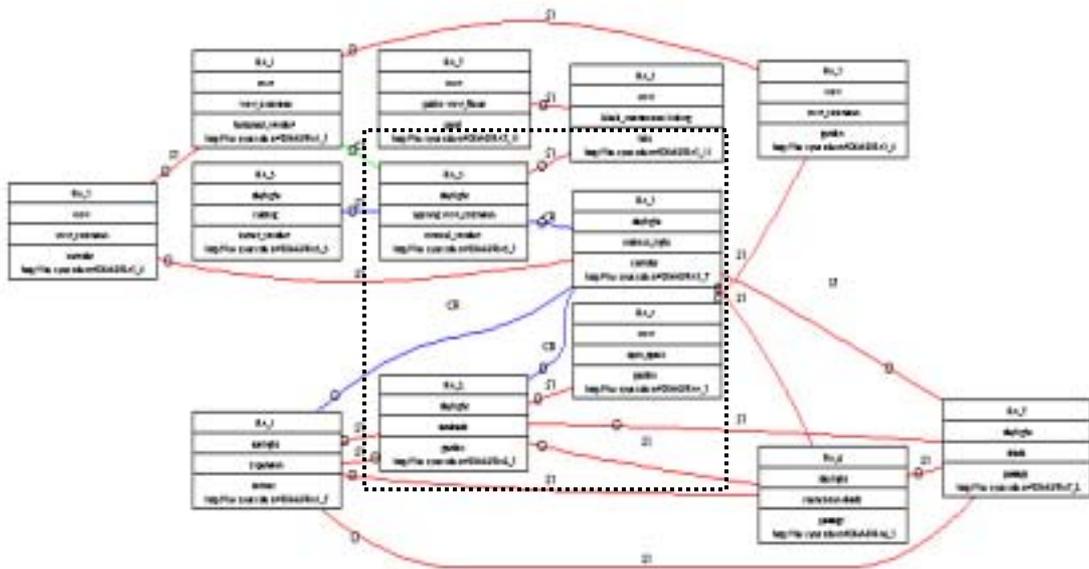


圖 63. 學生 E 的想法草圖與其想法組合的地圖範圍

6.3.1 連結有效性評估

在這個設計實驗中，Team 1 與 Team 2 在採光與動線場景所產生的外部 IE 地圖，分別見圖 64 與 65 (Team 1)，和 66 與 67 (Team 2)。Team 1 在採光 (daylight) 場景所產生的外部 IE 地圖 (圖 64)，想法元件數量為 12，連結數量為 10，而在動線 (circulation) 場景產生的外部 IE 地圖 (圖 65)，Team 1 產生的想法元件數量為 9，連結數量為 8。

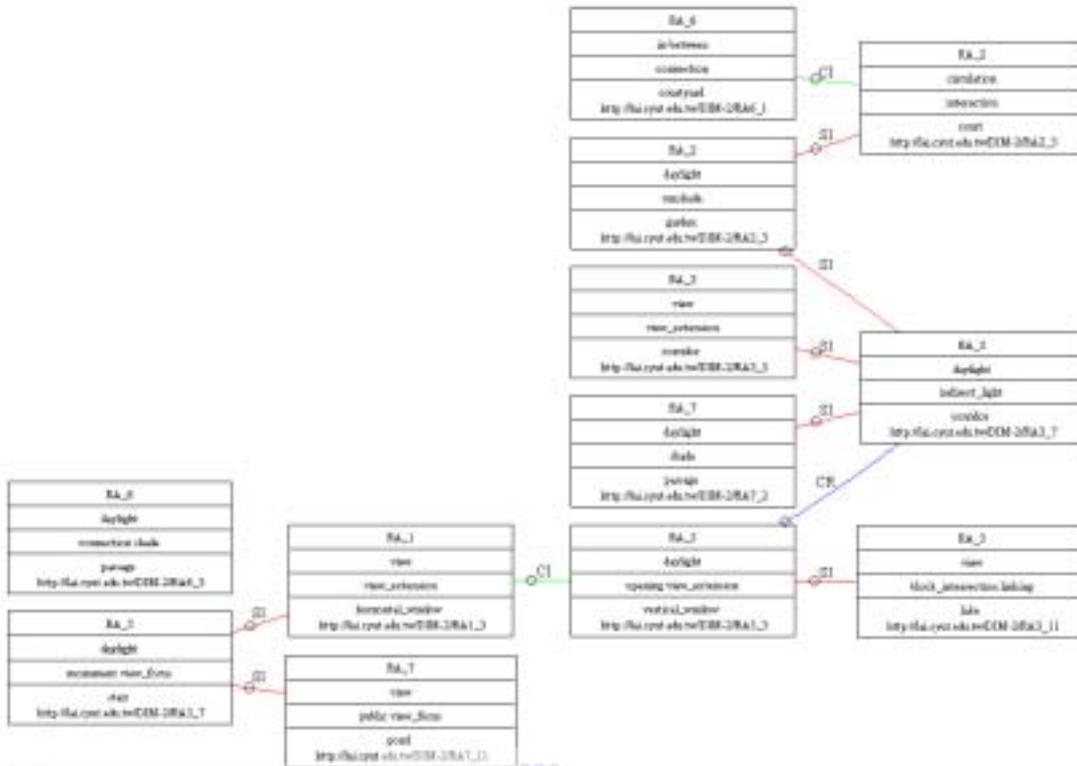


圖 64. Team 1 在採光 (daylight) 場景產生的外部 IE 地圖

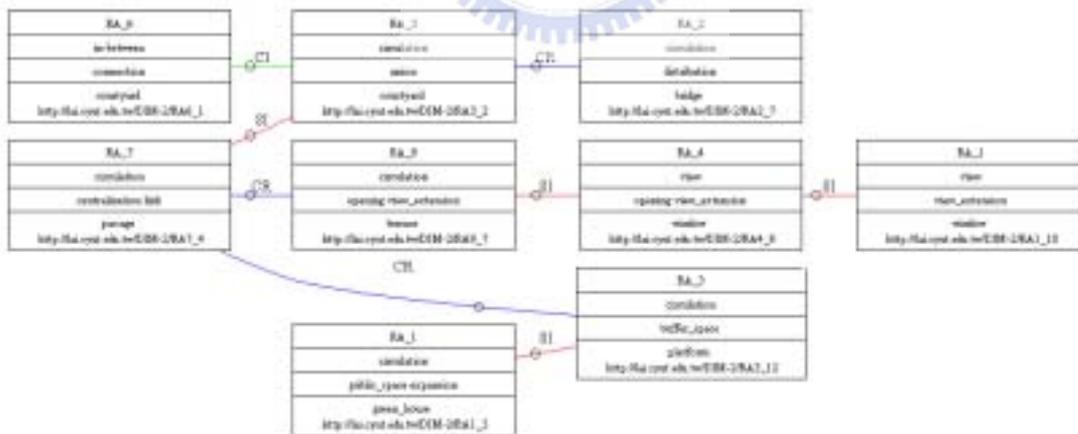


圖 65. Team 1 在動線 (circulation) 場景產生的外部 IE 地圖

另一方面，使用 DIM 系統的 Team 2，其在採光 (daylight) 場景所產生的外部 IE 地圖 (圖 66) 中，想法元件數量為 21，連結數量為 34，而在動線 (circulation) 場景產生的外部 IE 地圖 (圖 67)，Team 2 產生的想法元件數量為 14，連結數量為 22。

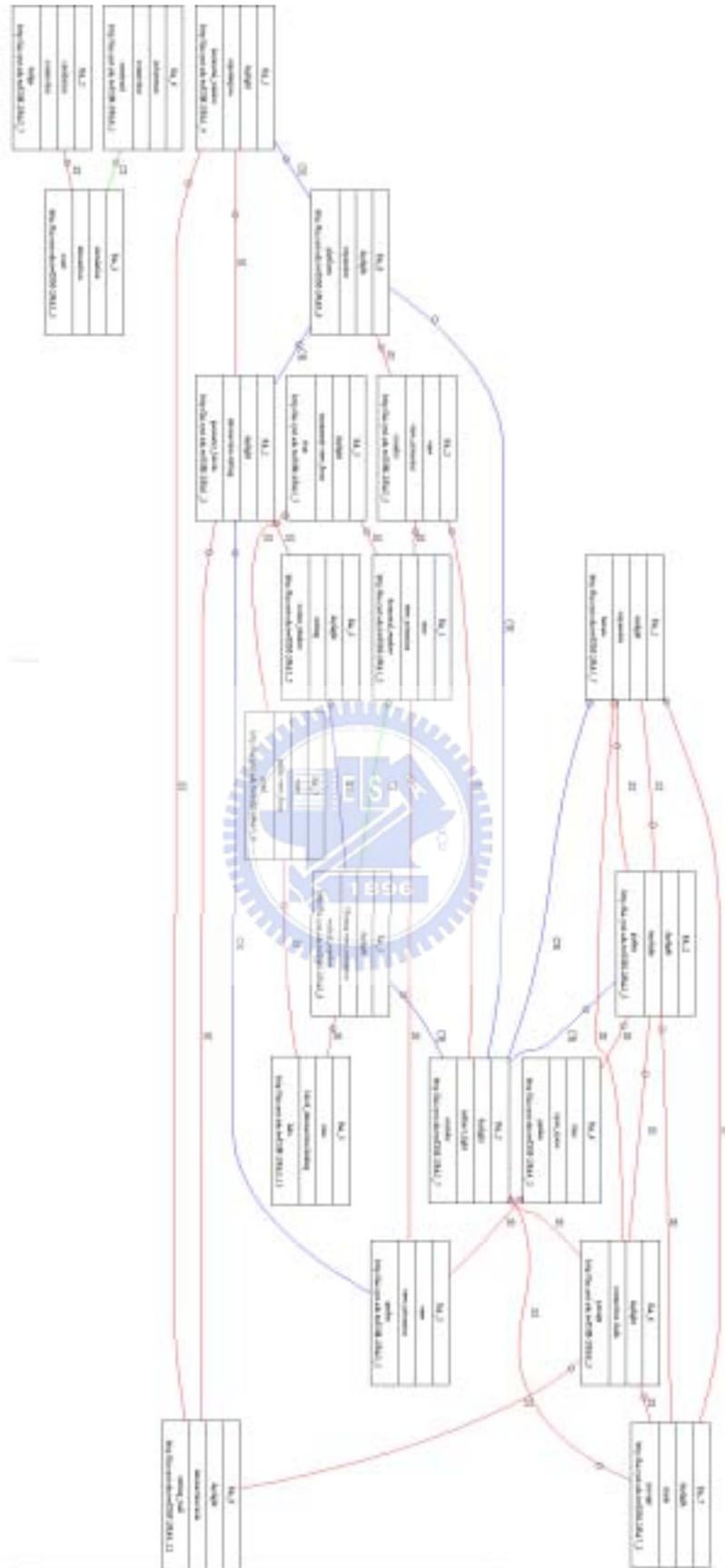


圖 66. Team 2 在採光 (daylight) 場景產生的外部 IE 地圖

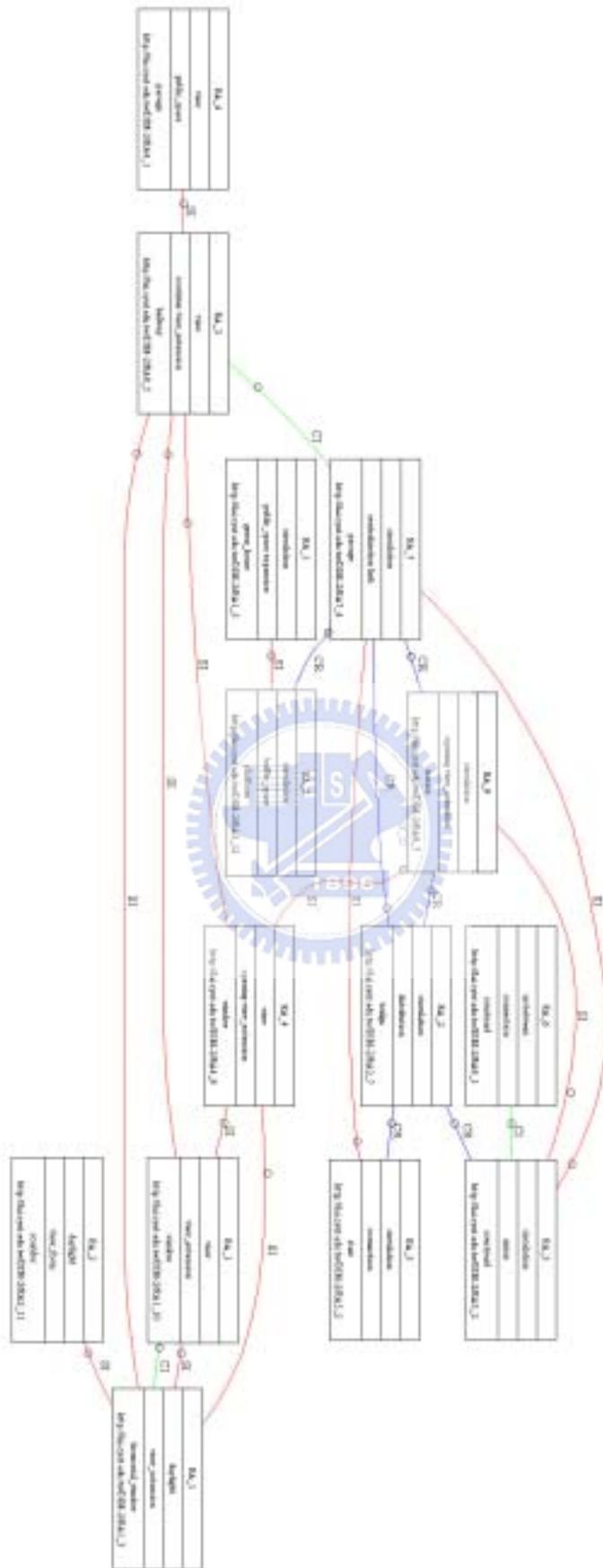


圖 67. Team 2 在動線場景產生的外部 IE 地圖

上述兩組學生在兩個場景產生的想法元件與元件連結數量被分別紀錄在表 14 中。同時，根據上述連結指標 (LI) 的定義 (詳見 6.1.1.1)，Team 1 在採光與動線的 LI 數值分別為 0.83 和 0.89，而 Team 2 在採光與動線的 LI 數值分別為 1.62 和 1.57。

表 14. Team 1 與 Team 2 產生外部 IE 地圖連結有效性比較圖表

場景 (議題)	Team 1 (A、B、C)		Team 2 (D、E、F)	
	採光 (daylight)	動線 (circulation)	採光 (daylight)	動線 (circulation)
想法元件數量	12	9	21	14
元件連結數量	10	8	34	22
LI	0.83	0.89	1.62	1.57

6.3.2 想法組合與連結分析

為了解參與者如何使用上述的外部 IE 地圖進行想法組合，主要經由 Team 2 學生產生之替選方案所選取的地圖範圍，進行有關連結形式指標與連結路徑長度的分析，在此想法組合的實驗中，Team 2 的學生 D、E 和 F 分別發展 3 個、2 個、3 個的設計替選方案，表 15 為 Team 2 學生在想法組合所選取的地圖範圍中，所包含的想法元件數量 (包括關鍵想法元件)、元件連結數量與三種連結形式指標的數值。例如：在表 15 中，學生 D 在其替選方案 Case2 的想法元件數量為 5，元件連結數量為 4， LTI_{si} 為 0.75， LTI_{cr} 為 0.25， LTI_{ci} 為 0。

表 15. Team 2 參與者產生替選方案之連結形式指標圖表

參與者	D			E		F		
	Case1	Case2	Case3	Case1	Case2	Case1	Case2	Case3
想法元件	5	5	5	4	5	7	6	7
元件連結	5	4	4	4	4	6	6	7
LTI_{si}	0.6	0.75	0.5	0.5	0.5	0.83	0.83	0.86
LTI_{cr}	0.2	0.25	0.5	0.5	0.5	0.17	0.17	0.14
LTI_{ci}	0.2	0	0	0	0	0	0	0

表 16 是 Team 2 學生所選取的地圖範圍中，除了關鍵想法元件之外，在不同連結路徑長度中所包含之想法元件的數量。例如：在表 16 中，學生 D 在其替選方案 Case2 中，使用 4 個想法元件與關鍵想法元件進行組合，在此地圖範圍中，其 LPL1 的想法元件數量為 1，LPL2 的想法元件數量為 1，LPL3 的想法元件數量為 2。

表 16. Team 2 參與者產生替選方案之連結路徑長度圖表

參與者	D			E		F		
	Case1	Case2	Case3	Case1	Case2	Case1	Case2	Case3
LPL1	2	1	4	2	1	3	2	4
LPL2	2	1	0	1	1	3	1	3
LPL3	1	2	0	1	1	1	2	0
LPL4	0	0	0	0	0	0	1	0
LPL5	0	0	0	0	0	0	0	0

6.4 發現與討論

經由上述設計實驗分析中發現，相較於面對面想法連結的 Team 1，使用 DIM 系統進行分散式想法連結的 Team 2，除了產生較多的想法元件與元件連結數量外，連結指標數值也較高（詳見表 14）。依據表 14 中的數值顯示，DIM 系統在早期設計階段中，經由反射性與關聯性的運算機制，可以提供多重參與者在分散式的設計環境中進行連結有效性之建立，進而，輔助 Team2 的學生在「建立在其他想法的基礎上」，尋求想法組合以發展可行的設計替選方案。

在尋求想法組合的過程中，在外部 IE 地圖中之三種不同的元件連結，提供不同的功能進行想法組合。此外，在設計共鳴現象中，關鍵想法元件與其他想法元件之間的關聯性（包括元件連結的數量、連結路徑長度）扮演關鍵的因素。為了更進一步了解 DIM 系統如何輔助設計者進行分散式想法連結，本研究主要從輔助設計結果（想法地圖應用）與設計過程（連結過程自動化）進行說明，同時，有關連結問題在立即性之介面需求與衝突性的學習機制，也在下列章節進行討論。

想法地圖的應用

有關想法地圖（主要指外部 IE 地圖）的應用，本研究主要根據 1) 想法元件的組

合方式，2) 元件連結的連結應用，與 3) 連結過程的重新使用進行討論。

1. 想法元件的組合方式：在想法組合的過程中，根據設計共鳴現象[104]，參與者以關鍵想法元件為中心向外延伸，並根據想法元件的組合數量，想法元件之連結路徑長度與其元件連結的數量等條件，找尋其它的想法元件進行組合。在本實驗中，本研究發現與關鍵想法元件組合的想法元件數量不超過 7 個(表 15)，此現象似乎反應人類在短期記憶中，最多只能同時處理 7 個知識區塊[111]的現象；同時，一個想法元件的元件連結數量，提供參與者選擇此想法元件進行想法組合的另一個重要因素，換言之，具有較多元件連結的想法元件，比較容易被選擇與關鍵想法元件進行組合。

在表 16 的數值中，本研究發現學生在組合外部 IE 地圖，主要可以透過廣度基礎或深度基礎，進行設計想法的組合。在廣度基礎的組合方式中，主要使用 LPL 較小的想法元件進行想法組合，例如：在表 16 中，學生 D 的 Case3 或學生 F 的 Case3；在深度基礎的組合方式中，主要經由不同路徑深度之 LPL 值進行想法組合，例如：在表 16 中，學生 D 的 Case2 或學生 E 的 Case2。然而，當一個想法元件的 LPL 超過 4 以上或無連結時，此想法元件較難與關鍵想法元件進行想法組合，這反應了設計者在處理關聯性的知識建構有其範圍性。

2. 元件連結的連結應用：三種不同型態的元件連結（相似、對比與相鄰）除了提供學生找尋不同設計想法之間的關聯性外，同時，也提供學生進行設計想法組合的依據。在想法組合之連結形式指標中，這些設計替選方案之地圖範圍中， LTI_{si} 與 LTI_{cr} 的數值普遍較 LTI_{ci} 高，反應參與者容易使用相似元件連結與對比形式連結進行想法組合，以強調問題解決方法（也就是概念與形式）的組合。此外，本研究發現經由相鄰元件連結所連結之想法元件的議題值，當此議題值出現在此場景 IE 地圖的頻率越少，且具有此議題值之想法元件擁有元件連結數量較多時，此想法元件提供參與者選擇下一個場景的議題值，此議題值也是此場景的起始想法元件。

然而，DIM 的場景順序是由 DA 預先設定的，因此，這些場景的議題值順序與上述相鄰元件連結所決定的議題值是有所差異。例如：在此實驗中，原先設定的第二個議題值為動線 (circulation)，擁有相鄰元件連結且頻率出現最少之想法元件的議題值為中介 (in-between) (圖 66)。因此，DIM 如何經由相鄰元件連結，自動化產生議題值並動態性的改變場景的順序，進而，選擇適當的想法元件為下一個場景的輸入值，以建立不同場景之間的關聯性，將是未來的後續研究。

3. 連結過程的重新利用：在分散式的想法連結過程中，內部相互作用與外部相互作用的連結過程（包括腳本內容、溝通機制、時間等），被動態性的儲存在每一位使用者（包括導演）的 RAS 知識庫，和導演的 ScA 知識庫中。因此，參與者可以在連結想法的過程中，重新利用這些連結過程並進行編輯，例如：經由下載在儲存在 ScA 知識庫中的連結過程，多重參與者可以延續原本特定場景的外部相互作用進行想法連結，此外，每一個參與者也可以下載儲存在 RAS 知識庫中的連結過程，延續其內部相互作用並進行想法連結。因此，在有限的時間限制之下，連結過程的重新利用有助於提升連結有效性之建立。

除了外部 IE 地圖外，在內部 IE 地圖中，設計共鳴現象同樣影響著參與者在個人的內在相互作用中，選取為輸出設計想法的依據，同時，參與者對個別想法元件的喜好程度，也為重要的選擇依據。雖然在產生的想法元件與連結時，DIM 提供立即性的視覺回饋，然而，當產生的 IE 地圖（包括外部與內部）所包含的想法元件或連結過多時，參與者不容易在有限的視窗看到所有想法元件與其相關的連結原則，導致地圖閱讀上的問題，同時，IE 地圖過於龐大的問題也會造成參與者在選擇與組合想法的困擾。因此，在 DIM 中，地圖的資訊視覺化與其範圍控制策略的介面需求，在未來研究（詳見第七章）是必須加以考慮的。

設計過程自動化

DIM 系統產生的內在與外在 IE 地圖，可以提供參與者產生個人與共同設計想法的依據，有助於個人腦力激盪與集體腦力激盪的進行，另外，DIM 系統的自動化機制，可以有效的輔助參與者進行分散式想法連結，在想法連結過程之不同的相互作用中，相關的代理人能動態性的改變連結原則，並予以自動化的連結與產生設計想法，同時設計過程自動化還能提供參與者下列的幫助：

1. 自動連接中斷的連結過程：在此實驗過程中，當參與的學生（UA 或 DA）臨時不在電腦桌時，則相關的代理人（RAS 和 ScA）分別在內外相互作用中，能自動的產生想法元件與元件連結，以連結其他學生產生的設計想法；或當其中一位學生沒有適當的設計知識產生設計想法時，則其他代理人（UA 或 DA）能自動的替補並產生設計想法。因此，對於學生進行面對面想法連結時，因想法連結過程的中斷而影響設計生產過程的結果，但就 DIM 系統而言，不會有想法中斷的問題。

此外，上述連結過程也會因為在不同的相互作用中，參與者或角色之間傳遞訊息之網路拓樸關係³的差異而改變最後的設計結果，此現象提供另一種連接中斷

³不同知識實體進行相互作用時，彼此之間的連結關係稱之為網路拓樸（network topology）關係（Chang, 2004）。

之連結過程的可能方式。有關網路拓樸關係如何影響連結過程與其運算機制，在第八章的未來研究有更進一步說明。

2. 自動化解決連結過程之衝突性：在外部相互作用中，學生之間因個人主觀性而導致的設計衝突，可以經由每位學生在其內部相互作用中，改變角色代理人的設計知識（包括容許限制和原則型式），或扮演之角色的替換而獲得解決，進而提升想法連結之設計生產過程。除了想法元件之間的衝突性外，學生在其內部相互作用中，也會發生連結之間的衝突性，所謂**連結衝突性**為一個角色代理人所產生的兩個不同想法元件，會有二種以上不同形式的元件連結，換言之，也就是會有兩種以上的關聯性建立。

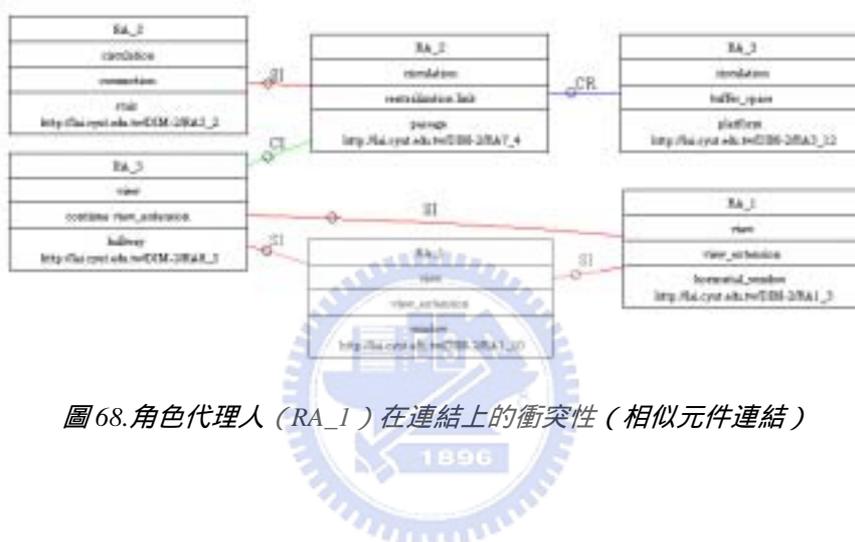


圖 68. 角色代理人 (RA_1) 在連結上的衝突性 (相似元件連結)

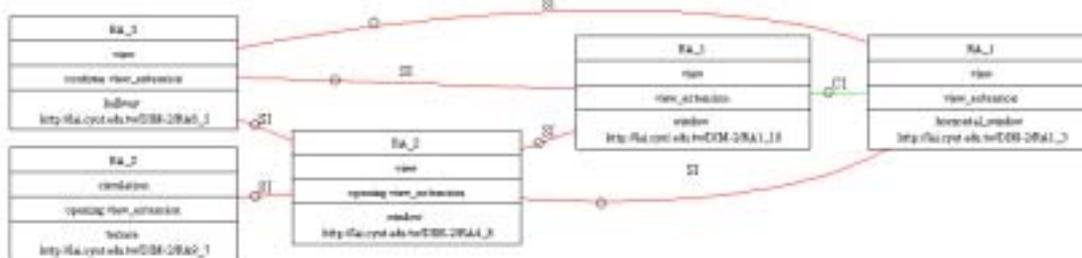


圖 69. 角色代理人 (RA_1) 在連結上的衝突性 (相鄰元件連結)

例如：在圖 68 與 69 中，學生 D 所產生的二個內部 IE 地圖中，RA_1 產生的兩個想法元件，在不同時間所儲存的 RAS 知識庫中，具有不同形式的元件連結，包括相似元件連結（圖 68）與相鄰元件連結（圖 69），進而影響 Team 2 在外在相互作用的設計結果（圖 67）。此現象反應了設計者在不同的設計情境下，對於設計想法之間的關聯性會採取不同的反射性行為，而導致內部知識的衝突性。此衝突性提供設計者在相同的想法元件間，產生二組以上不同關聯性的外部 IE 地圖，並使用這些不同型式的連結進行設計想法的組合。相較於面對面

進行想法連結的方式對於連結過程的衝突無法立即解決，DIM 系統則以節省時間的方式迅速解決該衝突。

3. 自動化增加連結之設計知識：JESS 的推理引擎可以動態性增加或改變 RA 的設計知識（包括角色 IE 地圖、想法元件與元件連結），並儲存在此 RA 的知識庫中，此現象可以被視為代理人的學習機制。此學習機制會根據容許限制的差異而有所不同（容許限制越低，角色代理人可以增加較多的設計知識在其知識庫中；容許限制越高，角色代理人在設計知識的增加較少），此現象說明了設計者對於外在設計知識的包容度越高，越能增加其長期記憶中的設計知識。在此設計實驗中，此學習機制提供學生在連結想法的過程中，可以連結到角色代理人原先知識庫中沒有的想法元件，有助於學生產生較多樣性與非預期性的設計想法。有關學習機制在第七章中的未來研究，將有更進一步說明。

在 DIM 系統中，內外相互作用之連結過程的資訊（包括時間、語言行動、訊息內容、接收者、發送者等），與其產生的不同 IE 地圖（內部與外部），皆可以被自動化的儲存在每位學生各自的資料庫中，除了提供學生重新利用這些連結過程資訊進行想法連結外，同時，這些資訊有助於在未來研究中，進行了解多重參與者在不同的相互作用中，所產生設計想法之間的建構過程與關係。



七、結論與建議

設計本身是一種設計者與想法之間的互動性行為，而想法聯想的連結與分散性問題是了解互動性行為的關鍵。本論文主要透過連結的觀點，對想法聯想中有關於分散式想法連結的機制進行探索，而為了解想法聯想之分散式想法連結機制，本研究架構在 Woodbury[26]之設計的「探測」概念，進行連結機制與其相關於搜尋在設計運算理論的探討。基本上，此分散式想法連結機制主要包括三個階層的運算機制：呈現、回憶與溝通，而這三個階層的運算機制，提供本研究建立一個輔助分散式想法連結模型（動態想法地圖，簡稱 DIM）的基礎，且經由 DIM 的模型建構與其系統實作，本研究對於分散式想法連結的連結機制與連結有效性提出下列的結論：

1. 分散式想法連結的三階層運算機制，是由不同互動性元素（角色、演員、舞台、場景和腳本），並結合三種不同類型的設計知識（知識、原則與連結過程）而建構並予以運算化。而這些元素與設計知識反應想法聯想在連結問題的特質：立即性、關聯性、反射性與衝突性。
2. 上述這些組成元素與設計知識，藉由結合案例基礎推理（CBR）與多重代理人（Multi-Agent）的運算技術予以具體化，並發展成一個輔助分散式想法連結的設計系統。因此，一個設計工作可經由案例基礎推理的機制，建立連結知識呈現和回憶行為，並透過多重代理人的分散性機制，在分散式的設計環境中，進行自動化的溝通以連結設計想法。
3. 分散式想法連結機制的系統建立，驗證此分散式想法連結機制運算化的可行性，並有利於連結有效性的建立。這些優勢有助於多重參與者在分散式的設計環境中，可以不受地理位置與時間條件的限制下，進行設計想法的連結、產生與組合，進而提供他們發展早期設計階段的設計替選方案。

基本上，DIM 在系統實作上與其他系統一樣受限於使用的軟體架構與程式語言。此外，DIM 系統缺乏普遍認可之領域概念語彙的建立，而造成此分散式想法連結機制在概念語彙解譯上的限制，而此限制可以藉由建立及連結一個普遍性（universal）建築領域概念語彙的資料庫（例如：普林斯敦大學認知科學研究室建立的語彙資料庫[160]）而改善，進而強化本研究建立之分散式想法連結機制，並提升 DIM 系統在產生多樣性想法與有效性連結的效能。

總之，本研究建立的分散式想法連結機制，將提供另一向度的「設計搜尋」策略，並建立「分散性設計搜尋」的可能方向，同時，此機制有助於多重參與者在分散

式的設計環境中，搜尋到關聯性、多元性且不同面向的設計知識。因此，架構在此分散式想法連結機制的基礎下，本研究有助於不同的建築設計領域（如學界的設計工作室、業界的建築師事務所），在走向國際化且分散式設計的趨勢中，建立一個智慧化且自動化之創造性問題解決會議（集體腦力激盪）的設計環境。除此之外，也由於 DIM 模型與其系統的建立，提供本研究在相關設計運算領域的研究貢獻與未來研究的期望。

7.1 研究貢獻

有關想法聯想過程中不同參與者進行分散式想法連結，與分散式想法連結模型 DIM 的實際執行，本研究提供詳細且完整的論述，也由於此分散式想法連結模型與其系統的建立，本論文的研究貢獻包括：

1. 建立想法聯想被運算化與分散化的機制：經由案例基礎推理與多重代理人的數位技術，想法聯想中的連結機制與三個原則（相似性、對比性與相鄰性）可以被運算化與分散化，並實際將連結機制發展成一個輔助設計系統，其透過如自動連結關聯性的想法、提供想法組合的依據、重新利用連結過程等助益，而輔助參與者在分散式的設計環境中進行想法聯想。
2. 形成想法組合的連結知識建構模式：在作者提出之 DIM 系統下，作者察覺到參與者如何尋求想法組合的連結知識建構，與三種不同想法聯想原則在想法組合所扮演的功能（詳見第六章節）。這些發現，將有助於未來相關想法組合與連結的認知研究，同時，也提供本研究發展控制 DIM 系統產生最佳化的想法地圖，以利設計者進行想法組合的基礎。
3. 解決分散式知識之間的設計衝突：在 DIM 不同的相互作用過程中，由於角色代理人各自擁有不同的設計知識，這些分散式設計知識所產生的設計衝突，透過三種不同的連結原則（相似原則、對比原則與相鄰原則）角色替換與其容許限制的改變，而建立出不同設計知識的關聯性，以解決設計衝突。另外，解決設計衝突也可以透過角色代理人彼此之間的學習（競爭與演化）而達成，有關代理人學習將是未來的後續研究。
4. 促使設計知識的自動化成長：在 DIM 中，擁有 JESS 推理引擎的角色代理人在想法連結的過程，不但具有動態性，而且自動化的增加這些角色代理人之內部記憶組織的設計知識，例如：增加想法元件與連結於角色代理人之記憶組織，進而擴大此角色代理人的角色 IE 地圖。另外，角色代理人的設計知識成長可以被視為一種代理人的學習機制，同時也是一種案例調整（adaptation）的過程。

5. 動態性搜尋設計案例知識：早期應用在輔助建築設計之 CBR 系統（例如：PRECEDENTS、Archie-II 等）的搜尋機制，由於受限於單一知識實體之集中式的知識基礎，無法動態性搜尋分散式之案例資料庫的相關資訊，而缺乏不同面向與多樣性的設計想法。然而，在 DIM 中，設計者可以其內部與外部相互作用，藉由 DIM 的分散式想法連結機制，動態性的搜尋分散式的案例知識，除了產生多樣性的設計想法外，也建立這些想法之間的連結有效性。

7.2 未來研究

未來研究主要根據本研究與其設計實驗的發現而提出，包括解決設計問題程序的自動化、相互作用網路拓樸關係、代理人的學習機制與想法地圖區域控制與視覺化。另外，有關角色代理人在內部記憶組織的案例知識調整，也是未來研究的重點。

1. 解決設計問題程序的自動化：在 DIM 中，解決設計問題程序是固定且預先設定的，然而根據第六章節中設計實驗的發現，此解決問題的程序主要根據具有相鄰連結原則之想法元件的連結數量而決定。因此，如何自動化的產生議題屬性值，並進行動態性的調整解決設計問題的程序，在 DIM 的架構下，未來必須加以考慮。

架構在角色扮演的理論，此未來研究可以經由了解 StA 的推理能力（包括計算與判斷外部 IE 地圖中，不同想法元件之連結種類與數量），與此 StA 與不同 ScA 之間在外部相互作用中之語言行動等代理人溝通語言機制的研究而達成。

2. 相互作用網路拓樸關係：根據本研究之設計實驗的發現，在分散式想法連結的過程中，不同的網路拓樸關係（例如：環狀、星狀、點對點與簇群狀，見圖 70）建立不同知識實體之間的連結關係[120]，進而影響一個分散式想法連結的設計結果。

因此，DIM 提供不同相互作用之網路拓樸關係，提供設計者在內部與外部相互作用中動態的選擇不同的網路拓樸關係，除了增加可以增加產生想法地圖的多樣性外，同時可以進行研究不同網路拓樸關係、連結過程與設計結果的關係。這些網路拓樸關係可以結合不同類型的語言行動與代理人溝通語言機制，在不同的 JADE 代理人平台之間，經由組織區劃這些平台所包含的不同容器，並建立這些容器與 DIM 系統代理人之間的連結而達成。

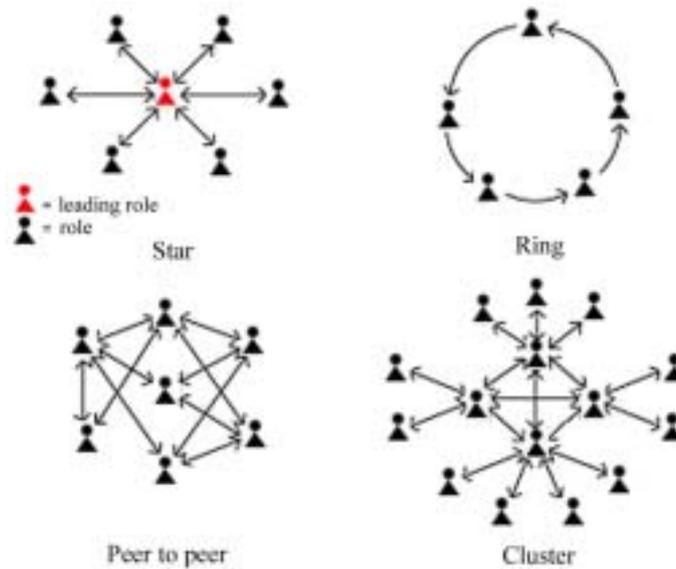


圖 70. 相互作用之網路拓樸關係：環狀、星狀、點對點與簇群狀

3. 代理人的學習機制：為增加產生設計想法的多樣性，DIM 需加強角色在 ICF 地圖轉換與 ICF 詞典解譯的功能。ICF 地圖轉換功能提供角色在內部交互作用連結想法元件的差異性，ICF 詞典除了可以藉由容許程度增加其變化性外，同時可以藉由增加 ICF 詞典中的領域概念語彙而達成，而上述這些功能的進行，皆須透過角色之間的學習而達成。

架構在角色扮演的理論下，有關 DIM 的連結過程，代理人的學習機制主要包括競爭學習與演化學習。競爭學習是不同角色代理人在連結過程中，它們所產生的想法元件被選擇為共同想法之競爭，為這些角色代理人爭取存活的依據。由於角色代理人彼此之間的競爭學習，演化學習是改變因相互競爭而失敗的角色代理人之內部記憶組織，包括演化不同類型的 ICF 地圖與 ICF 詞典的解譯能力，另外，想法元件中的喜好值也會因為演化學習而改變（喜好值的增加或減少）。

4. 想法地圖區域控制與視覺化：在第六章的設計實驗中發現，當產生的 IE 地圖過於龐大時，設計者很容易在這些地圖中迷失，而造成選擇與組合設計想法的障礙。Ulrich 和 Eppinger[161]認為雖然有成千上萬的想法可以在發散連結步驟中連結一起，但其中僅有 5 至 20 個可以被採用為問題解決的參考。為了形成一個可以理解的想法地圖，因此，在想法地圖中的想法元件與連結應被審慎的減少，以有效的提供設計者進行想法組合的依據。對於前述，在未來研究中，可以經由連結密度、連結形式指標與連結深度指標，提供 DIM 系統控制地圖範圍的運算機制，包括如深度控制、連結控制與元件控制等，以提升設計者進行想法組合與設計生產的有效性。

除此之外，使用資訊視覺化（information visualization）的相關技術與理論 [162]，有助於使用者清楚的閱讀想法地圖的資訊以進行設計想法的連結與組合，以免 IE 地圖所產生的想法元件過多，而造成設計者不易在有限的視窗看到所有想法元件與其元件連結。

5. 案例調整：改變舊有的、或經過擷取的解決問題方法，此過程稱之為案例調整。Dave 等人 [87] 認為此過程又可細分為兩種方式：調整與組合（combination）；基本上，案例組合採用類似於案例調整的技術，其主要差別在於必須有兩個或兩個以上的案例，從案例基礎中被選取，然後再進行案例之間的組合。

在 DIM 系統中，案例調整經由增加角色代理人內部記憶組織的想法元件與連結，與原有角色 IE 地圖中的想法元件進行組合，進而產生新的角色 IE 地圖。惟受限於角色本身 ICF 詞典解譯詞彙的能力，角色 IE 地圖的組合只能增加個別角色代理人所認識詞彙的想法元件，而未能增加其他角色代理人不同詞彙的想法元件，在未來研究中，DIM 可以藉由動態性的增加角色代理人之 ICF 詞典的詞彙而予以達成。除了角色 IE 地圖組合性的調整外，想法元件本身的調整可以藉由想法元件中的概念屬性詞彙數量的增加而達成，這些都將提供本研究在未來有關案例調整的研究方向。

參考文獻

- [1] Giedion, S., Space, Time and Architecture- the growth of a new tradition, Harvard University Press, Cambridge, Massachusetts, 1967.
- [2] Rapaport, D., The History of the Concept of Association of Ideas, International Universities Press, Inc., 1974.
- [3] Leibniz, G. W., New Essays Concerning the Human Understanding, La Salle, vol 3. 1703.
- [4] Bacon, F., De Augmentis Scientiarum, Dutton, New York, 1623.
- [5] Locke, J., An Essay Concerning Human Understanding in A. C. Fraser, Ed., Dover, New York, 1959.
- [6] Hume, D., "An Enquiry Concerning Human Understanding", in The English Philosophers from Bacon to Mill, E. A. Burt, Ed., pp. 585-589, Modern Library, New York, 1748.
- [7] Kant, I., Critique of Practical Reason, Bobbs-Merrill, New York, 1788.
- [8] Lewin, K., "Das Problem der Willensmessung und der Assoziation", Psychologische Forschung, vol. 2, pp. 65-140, 1922.
- [9] Kohler, W., "Zur Theorie des Successivver-gleiches und der Zeitfehler," Psychologische Forschung, vol. 5, pp. 115-175, 1923.
- [10] Freud, S., "The Interpretation of Dreams", Standard Edition of the Complete Psychological Works of Sigmund Freud, vol. 4 & 5., Hogarth Press, London, 1990.
- [11] Osborn, A. F., Applied Imagination: Principles and Procedures of creativity Thinking, Charles Scribner's Son, New York, 1963.
- [12] De Bono, E., Lateral Thinking, Penguin Books, New York, 1970.
- [13] van der Lugt, R., "Developing a graphic tool for creative problem solving in design groups", Design Studies, vol. 21, no. 5, pp. 505-522, 2000.
- [14] Goldschmidt, G. "The designer as a team of one", Design Studies, vol. 16, no. 2, pp. 189-209, 1995.
- [15] Haymaker, J., Keel, P., Ackermann, E., and Porter, W. "Filter mediated design: generating

- coherence in collaborative design", Design Studies, vol. 21, no. 2, pp. 205-220, 2000.
- [16] van der Lugt, R. and Buijs, J. A. "Creative problem solving in product development: an exploration into the use of CPS in design practice", in Creative Thinking: Towards Broader Horizons, S. Dingli, Ed., Malta University, Msida, Malta, 1998.
- [17] Goldschmidt, G. and Weil, M., "Contents and Structure in Design Reasoning", Design Issues, vol. 14, no. 3, Massachusetts Institute of Technology, 1998.
- [18] Chang, T. W. and Lai, I. C., Implementing Distributed Interaction with Dynamic Agent Role Interplay System (DARIS), Technical Report by Graduate School of Computational Design, NYUST, 2005.
- [19] Simon, H. A., The Sciences of the Artificial, Second Edition, The MIT Press, 1981.
- [20] Archea, J. "Puzzle-Making: What Architects Do When No One is Looking", in Computability of Design, Y. Kalay, Ed. Wiley-Interscience Publication, New York, 1987.
- [21] Schank, R. C., Dynamic Memory, Cambridge University Press, 1982.
- [22] Schank, R. C., Dynamic Memory Revisited, Cambridge University Press, 1999.
- [23] Coyne, R. D., Rosenman, M. A., Radford, A. D. and Gero, J. S., Knowledge-Based Design Systems, Addison-Wesley Publishing Company, 1990.
- [24] Jackson, P., Expert Systems, Third Edition, Addison-Wesley, 1999.
- [25] Kolodner, J. L., Case-Based Reasoning, Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1993.
- [26] Woodbury, R. F., "Searching for design; Paradigm and practice", Building and Environment, vol. 26, pp. 61-73, 1991.
- [27] Lai, I. C., "Dynamic Idea Maps: A Framework for Linking Ideas with Cases during Brainstorming," International Journal of Architectural Computing, vol. 3, no. 4, pp. 429-447, 2005.
- [28] Oxman, R., "Think-Maps: teaching design thinking in design education", Design Studies, vol. 25, no. 1, pp. 63-91, 2004.
- [29] Russell, S. J. and Norvig, P., Artificial Intelligence: A Modern Approach, Prentice Hall, 1995.
- [30] Gero, J. S. and Fujii, H., "A computational framework for concept formation for a situated design agent", The Second International Workshop on Strategic Knowledge and Concept

Formation, pp. 59-71, 1999.

- [31] Anumba, C. J., Ugwu, O. O., Newnham, L. and Thorpe, A., "Collaborative design of structures using intelligent agents", Automation in Construction, vol. 11, no. 1, pp. 89-103, 2002.
- [32] Liu, H., Tang, M. and Frazer, J. H., "A Knowledge Based Collaborative Design Environment", in Agents in Design, J. S. Gero and F. M. T. Brazier Eds., pp. 233-246, 2002.
- [33] Chang, T. W. and Lai, I. C. Implementing Distributed Interaction with Dynamic Agent Role Interplay System (DARIS), Technical Report by Graduate Institute of Architecture, NCTU, 2004.
- [34] 劉育東, "Protocol Analysis 在一般媒材, 電腦媒材與網路環境下的比較研究", 國科會技術報告 NSC902211-E009-047, 台灣, 2001.
- [35] Jones, J. C., Design Methods, John Wiley & Sons Inc. Press, New York, 1970.
- [36] Manolya, K., Stephen, A. R., and Linden, J., "Structure in idea sketching behaviour", Design Studies, vol. 19, no. 4, pp. 485-517, 1998.
- [37] Verstijnen, I. M., Hennessey, J. M., Leeuwen, C., Hamel, R. and Goldschmidt, G., "Sketching and creative discovery", Design Studies, vol. 19, no. 4, pp. 519-546, 1998.
- [38] Akin, O. "How do architects design? ", in Artificial Intelligence and Pattern Recognition in Computer Aided Design, J. Latombe, Ed., pp. 65-119, North Holland Publishing Company, Amsterdam, Holland, 1978.
- [39] Akin, O., "Architects' reasoning with structures and functions", Environment and Planning B: Planning and Design, vol. 20, pp. 273-294, 1993.
- [40] Schon, D. A. and Wiggins, G., "Kinds of seeing and their functions in designing", Design Studies, vol. 13, no. 2, pp. 135-156, 1992.
- [41] Akin, O. "An exploration of design process", Design Methods and Theory, vol. 13, no. 3-4, pp. 115-119, 1979.
- [42] Ericsson, K. A. and Simon, H. A., Protocol Analysis: Verbal Reports as Data, Revised Edition, The MIT Press, Cambridge, Massachusetts, 1993.
- [43] Eastman, C. M. "On the analysis of intuitive design process," in Emerging methods in environmental design and planning, G. T. N. Moore, Ed., The MIT Press, Cambridge, Massachusetts, 1970.

- [44] Elsas, P. A. and Vergeest, J. S. M., "New functionality for computer-aided conceptual design: the displacement feature", Design Studies, vol. 19, no. 1, pp. 81-102, 1998.
- [45] Gross, M. D., "The electronic Cocktail napkin: a computational environment for working with design diagrams", Design Studies, vol. 17, no. 1, pp. 53-69, 1996.
- [46] Goldschmidt, G., "Linkography: Assessing design productivity", in Cybernetics and systems'90, R. Trappl, Ed., pp. 291-298, World Scientific, Singapore, 1990.
- [47] Newell, A., Human Problem Solving. Prentice-Hall, Englewood Cliffs, N.J., 1972.
- [48] Alexander, C., Notes on the Synthesis of Form, Harvard University Press, Cambridge, Massachusetts, 1964.
- [49] Pena, W., and Parshall, S. A., Problem seeking: an architectural programming primer, Fourth Edition, John Wiley & Sons Inc., New York, 2001.
- [50] Arsimow, W., Introduction to Design, Prentice-Hall, Englewood Cliffs, N.J., 1962.
- [51] Akin, O., Models of Architectural Knowledge: An Information Processing View of Architectural Design, Department of Architecture in Carnegie Mellon University, Pittsburgh, 1979.
- [52] Anderson, J. R., Cognitive psychology and its implications, Third Edition, WH Freeman, New York, 1990.
- [53] Anderson, J. R. "Acquisition of cognitive skill", Psychological Review, vol. 89, no. 4, pp. 369-406, 1982.
- [54] Anderson, J. R., Greeno, J. G., Kline, P. J. and Neves, D. M. "Acquisition of problem solving skill", in Cognitive Skills and their Acquisition, J. R. Anderson, Ed., pp. 191-230, Lawrence Erlbaum, Hillsdale, N.J., 1981.
- [55] Norman, D. A., The Psychology of Everyday Things, Basic Books, 1988.
- [56] Woodbury, R. F. and Chang, T. W., "Building enclosures using SEED-Config", The Sixth CAAD Futures Conference, pp. 49-54, Singapore, 1995.
- [57] Woodbury, R. F. and Chang, T. W. "Massing and enclosure design with SEED-Config", ASCE Journal of Architectural Engineering, vol. 1, no. 4, pp. 170-178, 1995b.
- [58] Dave, B. G., "CDT: A Computer-Assisted Diagramming Tool," The Fifth CAAD Futures 1993 Conference, pp. 91-109, Pittsburgh, USA, 1993.

- [59] Woodbury, R. F., "Exploring large design spaces of building designs", Australian Research Council proposal, 1996.
- [60] Gero, J. S. and Kazakov, V., "An exploration-based evolutionary model of a generative design process", *Microcomputers in Civil Engineering*, 1996.
- [61] Flemming, U., "More on the representation and generation of loosely packed arrangements of rectangles", *Environment and Planning B: Planning and Design*, vol. 16, pp. 327-359., 1989.
- [62] Chang, T. W., *Geometric Typed Feature Structures: Toward Design Space Exploration*, School of Architecture, Landscape Architecture and Urban Design in University of Adelaide, 1999.
- [63] Novak, J. D., "Clarify with Concept Maps", *The Science Teacher*, vol. 58, no. 7, pp. 45-49, 1991.
- [64] Alexander, C., *The Timeless Way of Building*, Oxford University Press, New York, 1979.
- [65] Mitchell, W. J., *The Logic of Architecture*, The MIT Press, Cambridge, Massachusetts, 1990.
- [66] Oxman, R., "Precedents in Design: a computational model for the organization of precedent knowledge", *Design Studies*, vol. 15, no. 2, pp. 117-134, 1994.
- [67] Rhodes, M., "An analysis of creativity", *Phi Delta Kappan*, vol. 42, pp. 305-310, 1961.
- [68] Quillian, M. R., "Semantic memory", in *Semantic Information Processing*, M. Minsky, Ed., pp. 216-270, The MIT Press, Cambridge, Massachusetts, 1968.
- [69] Minsky, M. "Frame-system theory", in *Thinking: Readings in Cognitive Science*, P. C. Wason, Ed., Cambridge University Press, Cambridge, Massachusetts, 1977.
- [70] Sowa, J. F., *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks/Cole, Pacific Grove, CA, 2000.
- [71] Novak, J. D., *Learning, Creating and Using Knowledge: Concept Maps as Facilitative Tools in Schools and Corporations*, Lawrence Erlbaum Associates, London, 1998.
- [72] Novak, J. D. and Gowin, D. B., *Learning how to learn*, Cambridge University Press, New York, 1984.
- [73] Ausubel, D. P., "A subsumption theory of meaningful verbal learning and retention", *Journal of General Psychology*, vol. 66, pp. 213-224, 1962.

- [74] Canas, A. J., Hill, G. and Lott, J., Support for Constructing Knowledge Models in CmapTools, Technical Reprint by Institute for Human and Machine Cognition, Pensacola, 2004.
- [75] Gaines, B. R. and Shaw, M. L. G. "Concept maps as hypermedia components", International Journal of Human Computer Studies, vol. 43, pp. 323-361, 1995.
- [76] Ruiz-Primo, M. A., Schultz, S. E., Li. M., and Shavelson, R. J., "Comparison of the Reliability and Validity of Scores from Two Concept-Mapping Techniques", Journal of Research in Science Teaching, vol. 38, pp. 260-278, 2000.
- [77] McClure, J. R., Sonak, B. and Suen, H. K., "Concept Map Assessment of Classroom Learning: Reliability, Validity, and Logistical Practicality", Journal of Research in Science Teaching, vol. 36, pp. 475-492, 1999.
- [78] Wilson, R. J., Introduction to Graph Theory, Fourth Edition, Addison Wesley, Edinburgh Gate, England, 1996.
- [79] Carre, B., Graphs and Networks, Clarendon Press, Oxford, 1979.
- [80] Maher, M. L., Balachandran, M. B. and Zhang, D. M., Cased-Based Reasoning in Design, Lawrence Erlbaum Associates, Mahwah, N.J., 1995.
- [81] Akin, O. "Case-based instruction strategies in architecture", Design Studies, vol. 23, no. 4, pp. 407-431, 2002.
- [82] Clark, R. H. and Pause, M., Precedents in Architecture. Van Nostrand Reinhold, New York, 1985.
- [83] Rowe, C. and Kotter, F., Collage City, The MIT Press, Cambridge, Massachusetts, 1978.
- [84] Dennis, M., Court & Garden: From the French Hotel to the City of Modern Architecture, The MIT Press, Cambridge, Massachusetts, 1992.
- [85] Gero, J. S., "Design Prototypes : A Knowledge Representation Schema for Design", AI Magazine, vol. 11, pp. 26-36, 1990.
- [86] Rosenman, M., Gero, J., and Oxman, R. E., "What's in a Case? " The Fourth CAAD Futures Conference, pp. 230-285, 1991.
- [87] Dave, B. G., Schmitt, B. F. and Smith, I., "Case Based Design in Architecture", in Artificial Intelligence in Design, J. Gero. and F. Sudweeks, Eds. pp. 145-162, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

- [88] Domeshek, E. A. and Kolodner, J. L., "A case-based design aid for architecture", in Artificial Intelligence in Design, J. S. Gero, Ed., pp. 497-516, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1992.
- [89] Tsatsoulis, C. and Williams, A. B., "Case-Based Reasoning", in Knowledge-Based System, vol. 3, pp. 807-837, Academic Press, 2000.
- [90] Oxman, R. and Shabo, A. "The web as a visual design medium", The International Conference on Information Visualization, 1999.
- [91] Maher, M. L. and Pu, P., Issues and Applications of Case-Based Reasoning in Design, Lawrence Erlbaum Associates, Mahwah, N.J., 1997.
- [92] Domeshek, E. A. and Kolodner, J. L. "The Designers' Muse: Experience to aid conceptual design of complex artifacts", in Issues and Applications of Case-Based Reasoning in Design, M. L. Maher and P. Pu, Eds., pp. 11-38, Lawrence Erlbaum Associates, Mahwah, N.J., 1997.
- [93] Voss, A. "Case Design Specialists in Fabel", in Issues and Applications of Case-Based Reasoning in Design, M. L. Maher and P. Pu, Eds. pp. 301-335, Lawrence Erlbaum Associates, Mahwah, N.J., 1997.
- [94] Flemming, U., " Case-Based Design in the SEED System", in Knowledge-Based Computer-Aided Architectural Design, G. K. Carrara, and Y. E. Kalay, Eds., pp. 69-91, Elsevier Science, Amsterdam, The Netherlands, 1994.
- [95] Segers, N. and Vries, B., "The Idea Space System: Words as Handles to a Comprehensive Data Structure", The Tenth CAAD Futures Conference, pp. 31-40, 2003.
- [96] Oxman, R. and Oxman, R., " Precedents: Memory Structure in Design Case Libraries", The Fifth CAAD Futures Conference, pp. 273-287, 1993.
- [97] Heylighen, A., In case of architectural design: Critique and praise of Case-Based Design in archit, Department of Architecture in Katholieke University, Leuven Belgium, 2000.
- [98] Coyne, R. and Snodgrass, A., Interpretation in Architecture: Design As a Way of Thinking, First Edition, Routledge, 2005.
- [99] Dewey, J. How We Think, Dover Publications, New York, 1997.
- [100] Lewin, K., "Das Problem der Willensmessung und der Assoziation", Psychologische Forschung, vol. 1, pp. 191-302, 1922.
- [101] Lai, I. C., "Mapmaker - Linking Play among Ideas, Cases and Links", The 10th Conference

of Computer Aided Architectural Design Research in Asia, pp. 222-227, New Delhi, India, 2005.

[102] Anderson, J. R. and Bower, G. H., Human as-associative memory, Winston-Wiley, Washington and New York, 1973.

[103] Lai, I. C., Kao, K. T. and Chang, T. W., "Role Playing for Linking Ideas in the Idea Association Process", The 10th Conference of Computer Aided Architectural Design Research in Asia, pp. 119-129, New Delhi, India, 2005.

[104] Lai, I. C. and Chang, T. W., "A Distributed Linking System for Supporting Idea Association in the Conceptual Design Stage", Design Studies, in press, 2006.

[105] Jakobson, R., "Closing Statement: Linguistics and Poetics", in Style in Language, T. A. Sebeok, Ed., pp. 350-377, John Wiley & Sons Inc., New York, 1960.

[106] Searle, J. R., Speech Acts: an Essay in the physical of Language, Cambridge University Press, Cambridge, Massachusetts, 1969.

[107] Wooldridge, M., An Introduction to MultiAgent Systems, John Wiley & Sons Inc., London, 2002.

[108] Austin, J. L., How to Do Things with Words, J. o. Urmson, Ed., Oxford University Press, Oxford, 1965.

[109] Lai, I. C., "Agent Communication for Role Playing in the Idea Association Process", The 11th Conference on Computer Aided Architectural Design Research Conference, pp. 151-160, Kumamoto, Japan, 2006.

[110] Petrovic, I. K. and Svetel, I. "Conversation on Design Action: By Men or by Machines? ", The 12th European Conference on Education in Computer Aided Architectural Design, pp. 15-23, 1994.

[111] Miller, G. A., "The magical number seven, plus or minus two: Some limits on our capacity for processing information", The Psychological Review, vol. 63, pp. 81-97, 1956.

[112] Carbonell, J. G., "A computational model of analogical problem solving", The Seventh International Joint Conference on Artificial Intelligence, pp. 147-152, 1981.

[113] Carbonell, J. G. "Learning by analogy: Formulating and generalizing plans from experience", Machine Learning, vol. 1, pp. 137-161, 1983.

[114] Tan, M. "Emergent Relations," The Tenth CAAD Futures Conference, pp. 73-81, 2003.

- [115] Leake, D. B., Case-Based Reasoning: Experiences, Lessons, and Future Directions, The MIT Press, Cambridge, Massachusetts, 1996.
- [116] Watson, I. D., Applying Case-Based Reasoning: Techniques for Enterprise Systems, Morgan Kaufmann Publishers, San Francisco, CA, 1997.
- [117] Gero, J. S., "Conceptual designing as a sequence of situated acts", in Artificial Intelligence in Structural Engineering, I. Smith, Ed., pp. 165-177, Springer-Verlag Press, Berlin, 1998.
- [118] Clancey, W. J., Situated Cognition: On Human Knowledge and Computer Representations, Cambridge University Press, New York, 1997.
- [119] Yardley-Matwiejczuk, K. M., Role Play: Theory and Practice, SAGE Publications Ltd., London, 1997.
- [120] Chang, T. W., "Modeling Generative Interplay Using Acting Role Model. From distributed collaboration to generative interplay", CoDesign, vol. 2, no. 1, pp. 35-48, 2006.
- [121] Schank, R., Virtual Learning: A revolutionary approach to building a highly skilled workforce, McGraw-Hill, New York, 1997.
- [122] Chang, T. W. and Huang, J. H. "A pilot study of role-interplay in Web-based learning environment", Educational Media International, vol. 39, no. 1, pp. 75-85, 2002.
- [123] Radford, A. D. and Stenvens, G. "Role-Play in Education: A Case Study from Architectural Computing", Journal of Architectural Education, vol. 42, no. 1, 1988.
- [124] Moreno, J. L., Psychodrama, Beacon House, New York, 1946.
- [125] Andrianoff, S. K. and Levine, D. B. "Role Playing in an Object-Oriented World", The 33rd SIGCSE Technical Symposium on Computer Science Education Conference, pp. 121-125, 2002.
- [126] Radford, A. D. "Games and learning about form in architecture", Automation in Construction, vol. 9, no. 4, pp. 379-385, 2000.
- [127] Stanislavski, C., Creating a Role, Reprint Edition, Theatre Arts Books, London, 1968.
- [128] Levin, I. and Levin, I., Working on The Play and The Role--the Stanislavsky Method for Analyzing the Characters in a Drama, Ivan R Dee, Inc., Chicago, 1992.
- [129] Schank, R. C. and Abelson, R. P., Scripts, Plans, Goals and Understanding, Lawrence Erlbaum and Associates, Hillsdale, N.J., 1977.

- [130] Petrovic, I. K., "Computer Design Agents in a Brainstorming Session", The 15th Conference of Education and research in Computer Aided Architectural Design in Europe, Vienna, 1997.
- [131] Wooldridge, M. and Jennings, N. R., "Agent Theories, Architectures, and Languages", in Intelligent Agents, M. Wooldridge and N. R. Jennings, Eds., Springer-Verlag Press, 1995.
- [132] Maes, P., "Artificial Life Meets Entertainment: Life like Autonomous Agents", Communications of the ACM, vol. 11, pp. 108-114, 1995.
- [133] Hayes-Roth, B., An Architecture for Adaptive Intelligent Systems, Technical Report by Knowledge Systems Laboratory in Stanford University, Palo Alto, CA, 1995.
- [134] Kannengiesser, U. and Gero, J. S., "Situated agent communication for design", in Agents in Design 2002, J. S. Gero and F. Brazier, Eds., pp. 85-94, Key Centre of Design Computation and Cognition, University of Sydney, Australia, 2002.
- [135] Saunders, R. and Gero, J. S., "Curious Agents and Situated Design Evaluations", in Agents in Design 2002, J. S. Gero and F. Brazier, Eds., pp. 133-149, Key Centre of Design Computation and Cognition, University of Sydney, Australia, 2002.
- [136] Ishida, T., Ishiguro, H., and Nakanishi, H., "Connecting Digital and Physical Cities", in Revised Papers from the Second Kyoto Workshop on Digital Cities II, Computational and Sociological Approaches, M. Tanabe, P. van den Besselaar, T. Ishida, Eds., pp. 247-256, Springer-Verlag Press, 2002.
- [137] Akahani, J. I., Hiramatsu, K. and Furukawa, Y., "Agent-Based Coordination of Regional", in Revised Papers from the Second Kyoto Workshop on Digital Cities II, Computational and Sociological Approaches, M. Tanabe, P. van den Besselaar, T. Ishida, Eds., pp. 283-291, Springer-Verlag Press, 2002.
- [138] Maher, M. L. and GU, N., "Design Agents in Virtual Worlds: A User-Centred Virtual Architecture Agent", in Agents in Design 2002, G. J. S. and Brazier, F. M. T., Eds., pp. 23-38, Key Centre of Design Computing and Cognition, University of Sydney, Australia, 2002.
- [139] Ligtenberg, A. A., Breget, K. and Van Lammeren, R. "Multi-actor-based land use modeling: spatial planning using agents", Landscape and Urban Planning, vol. 56, no. 1-2, pp. 21-33, 2001.
- [140] Aly, S. and Krishnamurti, R. "Can Doors and Windows Become Design Team Player? ", in Agents in Design 2002, G. J. S. and Brazier, F. M. T., Eds., pp. 3-22, Key Centre of Design Computing and Cognition, University of Sydney, Australia, 2002.

- [141] Muller, J. P., The Design of Intelligent Agents : A Layered Approach, Springer-Verlag Press, London, 1996.
- [142] Hoare, C. A. R., "Communicating sequential processes", Communications of the ACM, vol. 21, no. 8, pp. 666-677, 1978.
- [143] Milner, R., Communication and Concurrency, Prentice-Hall, Englewood Cliffs, N.J., 1989.
- [144] Finin, T., Labrou, Y. and Mayfield, J., "KQML as an agent communication language," in In software agents, J. M. Bradshaw, Ed., pp. 291-316, The MIT Press, Cambridge, Massachusetts, 1997.
- [145] FIPA, Specification part 2 - Agent communication language, FIPA Organization, 1999.
- [146] FIPA, <http://www.fipa.org/>, 2005.
- [147] MASIF, <http://www.omg.org>, 2005.
- [148] O'Brien, P. D. and Nicol, R. C., "FIPA-towards a standard for software agents", BT Technolgy Journal, vol. 16, no. 3, pp. 51-59, 1998.
- [149] Poslad, S., Buckle, P. and Hadingham, R., The FIPA-OS agent platform: Open Source for Open Standards, Technical Report by Technology and Medicine Department in Imperial College of Science, London, 2002.
- [150] Domeshek, E. A. and Kolodner, J. L., "Using the points of large cases", Artificial intelligence for engineering design, analysis and manufacturing, vol. 7, no.2, pp. 87-96, 1993.
- [151] Smith, I., Lottaz, C. and Faltings, B., "Spatial composition using cases: IDIOM", in CBR Research and Development, pp. 88-97, Springer-Verlag, Berlin, Germany, 1995.
- [152] Ching, F. D. F., Architecture: Form, Space and Order, Second Edition, Van Nostrand Reinhold, New York, 1996.
- [153] Grecu D. L., and Brown, D. C., "Guiding Agent Learning in Design", in Knowledge Intensive Computer Aided Design, S. Finger, Tomiyama, T. and M. Mantyla, Eds., pp. 276-293, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.
- [154] Prasad, M. V. N., Lander, S.E., and Lesser, V.R., "On Retrieval and reasoning in distributed case bases", The 1995 IEEE International Conference, pp. 351-356, Vancouver, Canada 1995.
- [155] Lander, S. E., Negotiated search in heterogeneous multi-agent systems, Department of

Computer Science in University of Massachusetts, Amherst, 1993.

[156] JADE, <http://jade.tilab.com/>, 2005.

[157] Friedman-Hill, E. JESS in Action: Rule-Based System in Java, Manning Publications Co., Greenwich, CT, 2003.

[158] Global Architecture, GA HOUSES, A.D.A. EDITA, Tokyo.

[159] Goldschmidt, G., "How good are good ideas? Correlates of design creativity", Design Studies, vol. 26, no. 6, pp. 593-611, 2005.

[160] Princeton University, <http://wordnet.princeton.edu/>, 2004.

[161] Ulrich, K. T. and Eppinger, S. D., Product Design and Development, McGraw-Hill, New York, 1995.

[162] Ware, C., Information Visualization-Perception for Design, Morgan Kaufmann Publishers, San Diego, CA, 2000.



附 錄



附錄 A : DIM 代理人主要程式碼

A.1 角色代理人(RA)

```
.....  
: Defining RA's template  
  
// Idea entity  
(deftemplate ie(slot issue)(multislot concept)(multislot form) (slot preference (default 1)))  
  
// IE_Map  
(deftemplate map(slot issue_1) (multislot concept_1) (multislot form_1)(slot link_type)(slot issue_2) (multislot concept_2)  
(multislot form_2))  
  
// ICF_Map  
(deftemplate issue_map (slot issue_1)(slot link_type)(slot issue_2) )  
(deftemplate concept_map (multislot concept_1)(slot link_type)(multislot concept_2) )  
(deftemplate form_map (slot form_1)(slot link_type)(slot form_2) )  
  
// ACLMessage  
(deftemplate ACLMessage  
  (slot communicative-act)  
  (slot sender)  
  (multislot receiver)  
  (slot reply-with)  
  (slot in-reply-to)  
  (slot envelope)  
  (slot conversation-id) (slot protocol)  
  (slot language) (slot ontology) (slot content)  
  (slot encoding) (multislot reply-to) (slot reply-by)  
)  
  
// Temporary_IE and ICF  
(deftemplate temporary_ie(slot issue)(multislot concept)(multislot form))  
(deftemplate tolerance(slot number (default 3)))  
(deftemplate search-issue (slot word))  
(deftemplate search-concept (slot word))  
(deftemplate search-form (slot word))  
(deftemplate principle(slot type))  
(deftemplate map_ie(slot issue)(multislot concept)(multislot form))  
  
// Selected IE  
(deftemplate selected_ie(slot principle)(slot issue)(multislot concept)(multislot form) (slot preference (default 1)))  
  
.....  
: Defining RA's global value  
  
(defglobal ?*link* = "")  
(defglobal ?*new_c* = "")  
(defglobal ?*sim_list* = (create$ ))  
(defglobal ?*num* = 0)
```

```

.....
,,,,,,,,, Loading RA1's working memory

(reset)
(load-facts c:/jess/test/R1_map.clp)

.....
,,,,,,,,, Receiving ACLMessage from Dummy Agent

(defmodule Proposal)

(defrule proposal
  (declare (auto-focus TRUE))
  ?m <- (ACLMessage (communicative-act CFP) (sender ?s) (content ?c) (receiver ?r))
=>
  // 將輸入的 string 改為 list, 並改成 IE fact
  (assert (search-issue (word (nth$ 1 (explode$ ?c))))))
  (assert (search-concept (word (nth$ 2 (explode$ ?c))))))
  (assert (search-form (word (nth$ 3 (explode$ ?c))))))
  (assert (principle (type (nth$ 4 (explode$ ?c))))))
  (assert (tolerance (number (nth$ 5 (explode$ ?c))))))
  (modify ?m (communicative-act PROPOSE) (sender ?r) (receiver ?s) (content ?c))
  (save-facts c:/jess/DIM2.clp)
  (run-system )
)

.....
,,,,,,,,, Loading RA's principle: interpreting, converting and three types of linking

(batch c:/jess/test/Interpret.clp)
(batch c:/jess/test/Convert.clp)
(batch c:/jess/test/Link_SI.clp)
(batch c:/jess/test/Link_CR.clp)
(batch c:/jess/test/Link_CI.clp)

.....
,,,,,,,,, Selecting an IE with the highest number of preference within Similarity principle

// select IEs with the most number of preference
(defmodule Select)

(defrule selected_ie
  // identify the selected IE's preference with the highest number of preference
  ?fact <- (selected_ie (issue ?x)(concept ?y)(form ?z)(preference ?p))
  // Comparing selected IEs' preference number, and then matching the IEs with highest preference number
  (not (selected_ie (preference ?p2&:(> ?p2 ?p))))
=>
  (bind ?issue (fact-slot-value ?fact issue))
  (bind ?concept (fact-slot-value ?fact concept))
  (bind ?form (fact-slot-value ?fact form))
  (bind ?preference (fact-slot-value ?fact preference))
  (bind ?*num* (+ ?*num* 1))
  (bind ?*sim_list* (insert$ ?*sim_list* (+ (length$ ?*sim_list*) 1) (create$ ?issue ?concept ?form)))
  // insert new elements to end of global list and bind new list to ?sim_list
  (printout t "flag2 -----> ?sim_list= " ?*sim_list* crlf)
  (printout t "flag1 -----> length of list= " (length$ ?*sim_list*) ". " crlf)
  (printout t ?issue ?concept ?form ?preference crlf )

```

```

)

// select random similarity link and modify ACLMessage by selection data

(defrule select_random_similarity
  ?m <- (ACLMessage (communicative-act PROPOSE) (sender ?s) (content ?c&: (neq ?c ?*new_c*)) (receiver ?r))
=>
  (if (> ?*num* 0) then
    (bind ?rd (* (mod (random) ?*num*) 3))
    (bind ?*new_c*
      (implode$
        (create$
          (nth$ (+ ?rd 1) ?*sim_list*)
          (nth$ (+ ?rd 2) ?*sim_list*)
          (nth$ (+ ?rd 3) ?*sim_list*)
        )
      )
    )
    (modify ?m (content ?*new_c*))
  else
    (if (= ?*num* 0) then
      (modify ?m (communicative-act FAILURE)(content "I have no design ideas"))
    )
  )
)

)

.....
,,,,,,Sending ACLMessage to Dummy Agent

// When a message is asserted whose sender is this agent, the message is sent and then retracted from the knowledge base.

(defrule send-a-message
  (MyAgent (name ?n))
  ?m <- (ACLMessage (sender ?n))
=>
  (send ?m)
)

(deffunction run-system ()
  (focus Proposal Interpret Convert SI Select)
)

```

A.1.1 RA principle: Interpretation

```

// The following codes are for interpreting an input IE into several temporary IEs through DIM dictionary
// Rule for interpreting an input_ie into temp_ie based on tolerance 3
// While DIM finds words in right hand, and then DIM stops to find words in left hand within dictionary

```

```

(defmodule Interpret)

(defrule search-key-word-3
  (tolerance (number 3))
  (search-issue (word ?i))
  (search-concept (word ?c))
  (search-form (word ?f))
  (issue ?my_i1 ?my_i2)
  (concept ?my_c1 ?my_c2)
)

```

```

(form ?my_f1 ?my_f2)
(test
  (and
    (or (eq ?my_i2 ?i)(eq ?my_i1 ?i))
    (or (eq ?my_c2 ?c)(eq ?my_c1 ?c))
    (or (eq ?my_f2 ?f)(eq ?my_f1 ?f))
  )
)
(not (temporary_ie
  (issue ?ii&:(or (eq ?ii ?my_i1)(eq ?ii ?my_i2)))
  (concept ?cc&:(or (eq ?cc ?my_c1)(eq ?cc ?my_c2)))
  (form ?ff&:(or (eq ?ff ?my_f1)(eq ?ff ?my_f2)))
)
)
=>
(assert (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?*url*)))
(save-facts c:/DIM/temp.clp MAIN::temporary_ie)
)

// Rule for interpreting an input_ie into temp_ies based on tolerance 2
// While DIM finds words in right hand, and then DIM stops to find words in left hand within dictionary

```

```

(defrule search-key-word-2
  (tolerance (number 2))
  (search-issue (word ?i))
  (search-concept (word ?c))
  (search-form (word ?f))
  (issue ?my_i1 ?my_i2)
  (concept ?my_c1 ?my_c2)
  (form ?my_f1 ?my_f2)
  (test
    (or
      (and (or (eq ?my_i2 ?i)(eq ?my_i1 ?i))(or (eq ?my_c2 ?c)(eq ?my_c1 ?c)))
      (and (or (eq ?my_c2 ?c)(eq ?my_c1 ?c))(or (eq ?my_f2 ?f)(eq ?my_f1 ?f)))
      (and (or (eq ?my_f2 ?f)(eq ?my_f1 ?f))(or (eq ?my_i2 ?i)(eq ?my_i1 ?i)))
    )
  )
  (not (temporary_ie
    (issue ?ii&:(or (eq ?ii ?my_i1)(eq ?ii ?my_i2)))
    (concept ?cc&:(or (eq ?cc ?my_c1)(eq ?cc ?my_c2)))
    (form ?ff&:(or (eq ?ff ?my_f1)(eq ?ff ?my_f2)))
  )
)
=>
(assert (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?*url*)))
)

```

```

// Rule for interpreting an input_ie into temp_ies based on tolerance 1
// While DIM finds words in right hand, and then DIM stops to find words in left hand within dictionary

```

```

(defrule search-key-word-1
  (tolerance (number 1))
  (search-issue (word ?i))
  (search-concept (word ?c))
  (search-form (word ?f))
  (issue ?my_i1 ?my_i2)
  (concept ?my_c1 ?my_c2)
  (form ?my_f1 ?my_f2)
)

```

```

(test
  (or
    (or (eq ?my_i2 ?i)(eq ?my_i1 ?i))
    (or (eq ?my_c2 ?c)(eq ?my_c1 ?c))
    (or (eq ?my_f2 ?f)(eq ?my_f1 ?f))
  )
)
(not (temporary_ie
  (issue ?ii&:(or (eq ?ii ?my_i1)(eq ?ii ?my_i2)))
  (concept ?cc&:(or (eq ?cc ?my_c1)(eq ?cc ?my_c2)))
  (form ?ff&:(or (eq ?ff ?my_f1)(eq ?ff ?my_f2)))
)
)
=>
(assert (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?*url*)))
)

```

A.1.2 RA principle: Converting

// Converting interpreted words into three global values as inputs to match left-hand side of linking principles
 // 此 rule 為將 temporary ie 的值轉成三個 global 變數，以做為 linking principle rule's left-hend side 的 ie 輸入值

```

(defmodule Convert)

// Converting the three global values into different linking principle words from ICF maps
// issue
// 相似性 issue in issue_map (左邊和右邊)
(defrule matching_similarity_issue_map_l
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  ?fact <- (issue_map(issue_1 ?i&:(eq ?i ?my_i2))(link_type similarity))
=>
  (bind ?similarity_issue (fact-slot-value ?fact issue_2))
  (assert (similarity_issue (icf ?similarity_issue)))
)

(defrule matching_similarity_issue_map_r
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  ?fact <- (issue_map(link_type similarity)(issue_2 ?i&:(eq ?i ?my_i2)))
=>
  (bind ?similarity_issue (fact-slot-value ?fact issue_1))
  (assert (similarity_issue (icf ?similarity_issue)))
)

// 相鄰性 issue in issue_map (左邊和右邊)
(defrule matching_contiguity_issue_map_l
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  ?fact <- (issue_map(issue_1 ?i&:(eq ?i ?my_i2))(link_type contiguity))
=>
  (bind ?contiguity_issue (fact-slot-value ?fact issue_2))
  (assert (contiguity_issue (icf ?contiguity_issue)))
)

(defrule matching_contiguity_issue_map_r
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  ?fact <- (issue_map(link_type contiguity)(issue_2 ?i&:(eq ?i ?my_i2)))
=>
  (bind ?contiguity_issue (fact-slot-value ?fact issue_1))
)

```

```

    (assert (contiguity_issue (icf ?contiguity_issue)))
  )

// concept
// 相似性 concept in concept_map (左邊和右邊)
(defrule matching_similarity_concept_map_l
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  ?fact <- (concept_map(concept_1 ?c&:(eq ?c ?my_c2))(link_type similarity))
=>
  (bind ?similarity_concept (fact-slot-value ?fact concept_2))
  (assert (similarity_concept (icf ?similarity_concept)))
)

(defrule matching_similarity_concept_map_r
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  ?fact <- (concept_map(link_type similarity)(concept_2 ?c&:(eq ?c ?my_c2)))
=>
  (bind ?similarity_concept (fact-slot-value ?fact concept_1))
  (assert (similarity_concept (icf ?similarity_concept)))
)

// 對比性 concept in concept_map (左邊和右邊)
(defrule matching_contrast_concept_map_l
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  ?fact <- (concept_map(concept_1 ?c&:(eq ?c ?my_c2))(link_type contrast))
=>
  (bind ?contrast_concept (fact-slot-value ?fact concept_2))
  (assert (contrast_concept (icf ?contrast_concept)))
)

(defrule matching_contrast_concept_map_r
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  ?fact <- (concept_map(link_type contrast)(concept_2 ?c&:(eq ?c ?my_c2)))
=>
  (bind ?contrast_concept (fact-slot-value ?fact concept_1))
  (assert (contrast_concept (icf ?contrast_concept)))
)

// form
// 相似性 form in form_map (左邊和右邊)
(defrule matching_similarity_form_map_l
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  ?fact <- (form_map(form_1 ?f&:(eq ?f ?my_f2))(link_type similarity))
=>
  (bind ?similarity_form (fact-slot-value ?fact form_2))
  (assert (similarity_form (icf ?similarity_form)))
)

(defrule matching_similarity_form_map_r
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  ?fact <- (form_map(link_type similarity)(form_2 ?f&:(eq ?f ?my_f2)))
=>
  (bind ?similarity_form (fact-slot-value ?fact form_1))
  (assert (similarity_form (icf ?similarity_form)))
)

// 對比性 form in form_map (左邊和右邊)
(defrule matching_contrast_form_map_l

```

```

(temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
?fact <- (form_map(form_1 ?f&:(eq ?f ?my_f2))(link_type contrast))
=>
(bind ?contrast_form (fact-slot-value ?fact form_2))
(assert (contrast_form (icf ?contrast_form)))
)

(defrule matching_contrast_form_map_r
(temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
?fact <- (form_map(link_type contrast)(form_2 ?f&:(eq ?f ?my_f2)))
=>
(bind ?contrast_form (fact-slot-value ?fact form_1))
(assert (contrast_form (icf ?contrast_form)))
)

```

A.1.3 RA principle: Linking

Similarity

```

.....
..... Similarity linking principle (rules)
.....

(defmodule SI)

// rule for matching IE with three same slot values in left side of IE_Map to fetch right side IE and link
// rule for matching IE with three same slot values in right side of IE_Map to fetch left side IE and link

(defrule similarity_principle_1
(temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
?fact <- (map(issue_1 ?i&:(eq ?i ?my_i2))
          (concept_1 ?c&:(eq ?c ?my_c2))
          (form_1 ?f&:(eq ?f ?my_f2) ?x)
          (link_type similarity)
          (issue_2 ?i2)
          (concept_2 ?c2)
          (form_2 ?f2 ?y))
=>
(assert (map_ie(issue ?i2)(concept ?c2)(form ?f2 ?y)))
)

(defrule similarity_principle_2
(temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
?fact <- (map(issue_1 ?i)
          (concept_1 ?c)
          (form_1 ?f ?x)
          (link_type similarity)
          (issue_2 ?i2&:(eq ?i ?my_i2))
          (concept_2 ?c2&:(eq ?c ?my_c2))
          (form_2 ?f2&:(eq ?f ?my_f2) ?y))
=>
(assert (map_ie(issue ?i)(concept ?c)(form ?f ?x)))
)

// rule for converting map_ie facts into selected_ie facts (three same ICF slot value)

(defrule si_convert_selected_ie
(map_ie(issue ?i_map)(concept ?c_map)(form ?f_map ?url))

```

```

?fact <- (ie(issue ?i&:(eq ?i ?i_map))(concept ?c&:(eq ?c ?c_map))(form ?f&:(eq ?f ?f_map) ?x))
=>
(bind ?p (fact-slot-value ?fact preference))
(assert (selected_ie(principle similarity)(issue ?i)(concept ?c)(form ?f ?x)(preference ?p)))
)

// rule for matching ie with two same slot values(issue and form, issue and concept)
// to link new form's or concept's slot value.The temp_ies and IEs will be saved in RA's KB
// 當 issue 值一樣, 概念值一樣(?my_i2, ?my_c2, ~?my_f2)

(defrule similarity_principle_3
(temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
?fact <-(ie (issue ?i&:(eq ?i ?my_i2))(concept ?c&:(eq ?c ?my_c2))(form ?f&:(neq ?f ?my_f2) ?x)(preference ?p))
=>
(assert (ie(issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url)))
(assert (map (issue_1 ?i)(concept_1 ?c)(form_1 ?f ?x)(link_type similarity)
(issue_2 ?my_i2)(concept_2 ?my_c2)(form_2 ?my_f2 ?url)))
(assert (selected_ie(principle similarity)(issue ?i)(concept ?c)(form ?f ?x)(preference ?p)))
(save-facts c:/DIM/RA_1.clp MAIN::ie MAIN::map)
)

// 當 issue 值一樣, 形式值一樣(?my_i2, ~?my_c2, ?my_f2)

(defrule similarity_principle_4
(temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
?fact <- (ie (issue ?i&:(eq ?i ?my_i2))(concept ?c&:(neq ?c ?my_c2))(form ?f&:(eq ?f ?my_f2) ?x)(preference ?p))
=>
(assert (ie(issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url)))
(assert (map (issue_1 ?i)(concept_1 ?c)(form_1 ?f ?x)(link_type similarity)
(issue_2 ?my_i2)(concept_2 ?my_c2)(form_2 ?my_f2 ?url)))
(assert (selected_ie(principle similarity)(issue ?i)(concept ?c)(form ?f ?x)(preference ?p)))
(save-facts c:/DIM/RA_1.clp MAIN::ie MAIN::map)
)

// 當 IE 中只有一個相同 slots, 此 Rule 藉由 ICF_Maps 找尋其它相似的 slot 值後, 再產生新 form 或 concept 的 slot 值
// 當 issue 值一樣, 概念值不一樣(?my_i2, ?similarity_concept, ~?my_f2)

(defrule similarity_principle_5
(temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
(similarity_concept (icf ?similarity_concept))
?fact <- (ie (issue ?i&:(eq ?i ?my_i2))(concept ?c&:(eq ?c ?similarity_concept))(form ?f&:(neq ?f ?my_f2) ?x)
(preference ?p))
=>
(assert (ie(issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url)))
(assert (map (issue_1 ?i)(concept_1 ?c)(form_1 ?f ?x)(link_type similarity)
(issue_2 ?my_i2)(concept_2 ?my_c2)(form_2 ?my_f2 ?url)))
(assert (selected_ie(principle similarity)(issue ?i)(concept ?c)(form ?f ?x)(preference ?p)))
(save-facts c:/DIM/RA_1.clp MAIN::ie MAIN::map)
)

// 當 issue 值一樣, 形式值不一樣(?my_i2, ~?my_c2, ?similarity_form)

(defrule similarity_principle_6
(temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
(similarity_form (icf ?similarity_form))
?fact <- (ie (issue ?i&:(eq ?i ?my_i2))(concept ?c&:(neq ?c ?my_c2))(form ?f&:(eq ?f ?similarity_form) ?x)
(preference ?p))
=>

```

```

(assert (ie(issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url)))
(assert (map (issue_1 ?i)(concept_1 ?c)(form_1 ?f ?x)(link_type similarity)
            (issue_2 ?my_i2)(concept_2 ?my_c2)(form_2 ?my_f2 ?url)))
(assert (selected_ie(principle similarity)(issue ?i)(concept ?c)(form ?f ?x)(preference ?p)))
(save-facts c:/DIM/RA_1.clp MAIN::ie MAIN::map)
)

// 當 issue 值不一樣, 概念值一樣(?similarity_issue, ?my_c2, ~?my_f2)

(defrule similarity_principle_7
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  (similarity_issue (icf ?similarity_issue))
  ?fact <- (ie (issue ?i&:(eq ?i ?similarity_issue))(concept ?c&:(eq ?c ?my_c2))(form ?f&:(neq ?f ?my_f2) ?x)
            (preference ?p))
=>
  (assert (ie(issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url)))
  (assert (map (issue_1 ?i)(concept_1 ?c)(form_1 ?f ?x)(link_type similarity)
            (issue_2 ?my_i2)(concept_2 ?my_c2)(form_2 ?my_f2 ?url)))
  (assert (selected_ie(principle similarity)(issue ?i)(concept ?c)(form ?f ?x)(preference ?p)))
  (save-facts c:/DIM/RA_1.clp MAIN::ie MAIN::map)
)

// 當 issue 值不一樣, 形式值一樣 (?similarity_issue, ~?my_c2, ?my_f2)

(defrule similarity_principle_8
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  (similarity_issue (icf ?similarity_issue))
  ?fact <- (ie (issue ?i&:(eq ?i ?similarity_issue))(concept ?c&:(neq ?c ?my_c2))(form ?f&:(eq ?f ?my_f2) ?x)
            (preference ?p))
=>
  (assert (ie(issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url)))
  (assert (map (issue_1 ?i)(concept_1 ?c)(form_1 ?f ?x)(link_type similarity)
            (issue_2 ?my_i2)(concept_2 ?my_c2)(form_2 ?my_f2 ?url)))
  (assert (selected_ie(principle similarity)(issue ?i)(concept ?c)(form ?f ?x)(preference ?p)))
  (save-facts c:/DIM/RA_1.clp MAIN::ie MAIN::map)
)

// 當 IE 中沒有相同的 slots, 此 Rule 為藉由 ICF_Maps 找尋其它相似的 slot 值後, 再產生新 form 或 concept 的 slot 值
// 當 issue 值不一樣, 概念值不一樣(?similarity_issue, ?similarity_concept, ~?my_f2)

(defrule similarity_principle_9
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  (similarity_issue (icf ?similarity_issue))
  (similarity_concept (icf ?similarity_concept))
  ?fact <- (ie (issue ?i&:(eq ?i ?similarity_issue))(concept ?c&:(eq ?c ?similarity_concept))
            (form ?f&:(neq ?f ?my_f2)?url)(preference ?p))
=>
  (assert (ie(issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url)))
  (assert (map (issue_1 ?i)(concept_1 ?c)(form_1 ?f ?x)(link_type similarity)
            (issue_2 ?my_i2)(concept_2 ?my_c2)(form_2 ?my_f2 ?url)))
  (assert (selected_ie(principle similarity)(issue ?i)(concept ?c)(form ?f ?x)(preference ?p)))
  (save-facts c:/DIM/RA_1.clp MAIN::ie MAIN::map)
)

// 當 issue 值不一樣, 形式值不一樣(?similarity_issuep, ~?my_c2, ?similarity_form)

(defrule similarity_principle_10
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))

```

```

(similarity_issue (icf ?similarity_issue))
(similarity_form (icf ?similarity_form))
?fact <- (ie (issue ?i&:(eq ?i ?similarity_issue))(concept ?c&:(neq ?c ?my_c2))
          (form ?f&:(eq ?f ?similarity_form) ?x)(preference ?p))
=>
(assert (ie(issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url)))
(assert (map (issue_1 ?i)(concept_1 ?c)(form_1 ?f ?x)(link_type similarity)
            (issue_2 ?my_i2)(concept_2 ?my_c2)(form_2 ?my_f2 ?url)))
(assert (selected_ie(principle similarity)(issue ?i)(concept ?c)(form ?f ?x)(preference ?p)))
(save-facts c:/DIM/RA_1.clp MAIN::ie MAIN::map)
)

```

Contrast

```

.....
..... Contrast linking principle (rules)
.....

(defmodule CR)

// rule for matching IE with three same slot values in left side of IE_Map to fetch right side IE and link
// rule for matching IE with three same slot values in right side of IE_Map to fetch left side IE and link

(defrule contrast_principle_1
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  ?fact <- (map (issue_1 ?i&:(eq ?i ?my_i2))
              (concept_1 ?c&:(eq ?c ?my_c2))
              (form_1 ?f&:(eq ?f ?my_f2)?x)
              (link_type contrast)
              (issue_2 ?i2)
              (concept_2 ?c2)
              (form_2 ?f2 ?y))
=>
  (assert (map_ie(issue ?i2)(concept ?c2)(form ?f2 ?y)))
)

(defrule contrast_principle_2
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  ?fact <- (map(issue_1 ?i)
            (concept_1 ?c)
            (form_1 ?f ?x)
            (link_type contrast)
            (issue_2 ?i2&:(eq ?i ?my_i2))
            (concept_2 ?c2&:(eq ?c ?my_c2))
            (form_2 ?f2&:(eq ?f ?my_f2) ?y))
=>
  (assert (map_ie(issue ?i)(concept ?c)(form ?f ?x)))
)

// rule for converting map_ie facts into selected_ie facts (three same ICF slot value)

(defrule cr_convert_selected_ie
  (map_ie(issue ?i_map)(concept ?c_map)(form ?f_map ?url))
  ?fact <- (ie(issue ?i&:(eq ?i ?i_map))(concept ?c&:(eq ?c ?c_map))(form ?f&:(eq ?f ?f_map)?x))
=>
  (bind ?p (fact-slot-value ?fact preference))
  (assert (selected_ie(principle contrast)(issue ?i)(concept ?c)(form ?f ?x)(preference ?p)))
)

```

// 當 IE 中只有一個相同的 slots,此 Rule 為藉由 ICF_Maps 找尋概念與形式對比的語彙, 並使用兩個 slots 值(issue 和
 // 對比 form, issue 和對比 concept), 予以找尋取新的對比性 form 或 concept 的 slot 值; 同時, 將此 temporary ie 和一
 // 個 IE_Map 儲存於 RA's 知識庫

// 當 issue 值一樣, 對比概念值一樣(?my_i2, ?contrast_concept, ~?my_f2)

```
(defrule contrast_principle_3
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  (contrast_concept (icf ?contrast_concept))
  ?fact <- (ie (issue ?i&:(eq ?i ?my_i2))(concept ?c&:(eq ?c ?contrast_concept))(form ?f&:(neq ?f ?my_f2)?x)
    (preference ?p))
=>
  (assert (ie(issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url)))
  (assert (map (issue_1 ?i)(concept_1 ?c)(form_1 ?f ?x)(link_type contrast)
    (issue_2 ?my_i2)(concept_2 ?my_c2)(form_2 ?my_f2 ?url)))
  (assert (selected_ie(principle contrast)(issue ?i)(concept ?c)(form ?f ?x)(preference ?p)))
  (save-facts c:/DIM/RA_2.clp MAIN::ie MAIN::map)
)
```

// 當 issue 值一樣, 對比形式值一樣(?my_i2, ~?my_c2, ?contrast_form)

```
(defrule contrast_principle_4
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  (contrast_form (icf ?contrast_form))
  ?fact <- (ie (issue ?i&:(eq ?i ?my_i2))(concept ?c&:(neq ?c ?my_c2))(form ?f&:(eq ?f ?contrast_form)?x)
    (preference ?p))
=>
  (assert (ie(issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url)))
  (assert (map (issue_1 ?i)(concept_1 ?c)(form_1 ?f ?x)(link_type contrast)
    (issue_2 ?my_i2)(concept_2 ?my_c2)(form_2 ?my_f2 ?url)))
  (assert (selected_ie(principle contrast)(issue ?i)(concept ?c)(form ?f ?x)(preference ?p)))
  (save-facts c:/DIM/RA_2.clp MAIN::ie MAIN::map)
)
```

// 當 IE 中只有一個相同的 slots,此 Rule 為藉由 ICF_Maps 找尋概念與形式對比的語彙, 並使用兩個 slots 值(相似 issue
 // 和對比 form, 相似 issue 和對比 concept), 予以找尋取新的對比性 form 或 concept 的 slot 值
 // 當 issue 值不一樣, 對比概念值一樣(?similarity_issue, ?contrast_concept, ~?my_f2)

```
(defrule contrast_principle_5
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  (similarity_issue (icf ?similarity_issue))
  (contrast_concept (icf ?contrast_concept))
  ?fact <- (ie (issue ?i&:(eq ?i ?similarity_issue))(concept ?c&:(eq ?c ?contrast_concept))
    (form ?f&:(neq ?f ?my_f2)?x)(preference ?p))
=>
  (assert (ie(issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url)))
  (assert (map (issue_1 ?i)(concept_1 ?c)(form_1 ?f ?x)(link_type contrast)
    (issue_2 ?my_i2)(concept_2 ?my_c2)(form_2 ?my_f2 ?url)))
  (assert (selected_ie (principle contrast)(issue ?i)(concept ?c)(form ?f ?x)(preference ?p)))
  (save-facts c:/DIM/RA_2.clp MAIN::ie MAIN::map)
)
```

// 當 issue 值不一樣, 對比形式值一樣(?similarity_issue, ~?my_c2, ?contrast_form)

```
(defrule contrast_principle_6
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  (similarity_issue (icf ?similarity_issue))
```

```

(contrast_form (icf ?contrast_form))
?fact <- (ie (issue ?i&:(eq ?i ?similarity_issue))(concept ?c&:(neq ?c ?my_c2))
          (form ?f&:(eq ?f ?contrast_form)?x)(preference ?p))
=>
(assert (ie(issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url)))
(assert (map (issue_1 ?i)(concept_1 ?c)(form_1 ?f ?x)(link_type contrast)
            (issue_2 ?my_i2)(concept_2 ?my_c2)(form_2 ?my_f2 ?url)))
(assert (selected_ie (principle contrast)(issue ?i)(concept ?c)(form ?f ?x)(preference ?p)))
(save-facts c:/DIM/RA_2.clp MAIN::ie MAIN::map)
)

```

Contiguity

```

.....
..... Contiguity linking principle (rules)
.....

(defmodule CI)

// rule for matching IE with three same slot values in left side of IE_Map to fetch right side IE and link
// rule for matching IE with three same slot values in right side of IE_Map to fetch left side IE and link

(defrule contiguity_principle_1
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  ?fact <- (map(issue_1 ?i&:(eq ?i ?my_i2))
            (concept_1 ?c&:(eq ?c ?my_c2))
            (form_1 ?f&:(eq ?f ?my_f2) ?x)
            (link_type contiguity)
            (issue_2 ?i2)
            (concept_2 ?c2)
            (form_2 ?f2 ?y))
=>
  (assert (map_ie(issue ?i2)(concept ?c2)(form ?f2 ?y)))
)

(defrule contiguity_principle_2
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  ?fact <- (map(issue_1 ?i)
            (concept_1 ?c)
            (form_1 ?f ?x)
            (link_type contiguity)
            (issue_2 ?i2&:(eq ?i ?my_i2))
            (concept_2 ?c2&:(eq ?c ?my_c2))
            (form_2 ?f2&:(eq ?f ?my_f2) ?y))
=>
  (assert (map_ie(issue ?i)(concept ?c)(form ?f ?x)))
)

// rule for converting map_ie facts into selected_ie facts (three same ICF slot value)

(defrule ci_convert_selected_ie
  (map_ie(issue ?i_map)(concept ?c_map)(form ?f_map ?url))
  ?fact <- (ie(issue ?i&:(eq ?i ?i_map))(concept ?c&:(eq ?c ?c_map))(form ?f&:(eq ?f ?f_map)?x))
=>
  (bind ?p (fact-slot-value ?fact preference))
  (assert (selected_ie(principle contiguity)(issue ?i)(concept ?c)(form ?f ?x)(preference ?p)))
)

```

```

// rule for matching ie 中的兩個相同的 slots 值(concept 和 form), 予以找尋取新 issue 的 slot 值;
// 同時, 將此 temporary ie 和一個 IE_Map 儲存於 RA's 知識庫

// 當 concept 值一樣, form 值一樣(?my_c2 and ?my_f2)

(defrule contiguity_principle_3
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  ?fact <- (ie (issue ?i&:(neq ?i ?my_i2))(concept ?c&:(eq ?c ?my_c2))(form ?f&:(eq ?f ?my_f2) ?x)(preference ?p))
=>
  (assert (ie(issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url)))
  (assert (map (issue_1 ?i)(concept_1 ?c)(form_1 ?f ?x)(link_type contiguity)
              (issue_2 ?my_i2)(concept_2 ?my_c2)(form_2 ?my_f2 ?url)))
  (assert (selected_ie(principle contiguity)(issue ?i)(concept ?c)(form ?f ?x)(preference ?p)))
  (save-facts c:/DIM/RA_3.clp MAIN::ie MAIN::map)
)

// 當 IE 中只有一個相同的 slots, 此 Rule 為藉由 ICF_Maps 去找尋其它相似的 slot 值後, 再產生新 issue 的 slot 值
// 當 concept 值一樣, form 值不一樣(?my_c2 and ?similarity_form)

(defrule contiguity_principle_4
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  (similarity_form (icf ?similarity_form))
  ?fact <- (ie (issue ?i&:(neq ?i ?my_i2))(concept ?c&:(eq ?c ?my_c2))(form ?f&:(eq ?f ?similarity_form)?x)
            (preference ?p))
=>
  (assert (ie(issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url)))
  (assert (map (issue_1 ?i)(concept_1 ?c)(form_1 ?f ?x)(link_type contiguity)
              (issue_2 ?my_i2)(concept_2 ?my_c2)(form_2 ?my_f2 ?url)))
  (assert (selected_ie(principle contiguity)(issue ?i)(concept ?c)(form ?f ?x)(preference ?p)))
  (save-facts c:/DIM/RA_3.clp MAIN::ie MAIN::map)
)

// 當 concept 值不一樣, form 值一樣(?similarity_concept and ?my_f2)

(defrule contiguity_principle_5
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  (similarity_concept (icf ?similarity_concept))
  ?fact <- (ie (issue ?i&:(neq ?i ?my_i2))(concept ?c&:(eq ?c ?similarity_concept))
            (form ?f&:(eq ?f ?my_f2)?x)(preference ?p))
=>
  (assert (ie(issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url)))
  (assert (map (issue_1 ?i)(concept_1 ?c)(form_1 ?f ?x)(link_type contiguity)
              (issue_2 ?my_i2)(concept_2 ?my_c2)(form_2 ?my_f2 ?url)))
  (assert (selected_ie(principle contiguity)(issue ?i)(concept ?c)(form ?f ?x)(preference ?p)))
  (save-facts c:/DIM/RA_3.clp MAIN::ie MAIN::map)
)

// 當 IE 中沒有相同的 slots, 此 Rule 為藉由 ICF_Maps 去找尋其它相似的 slot 值後, 再產生新 issue 的 slot 值
// 當 concept 值不一樣, form 值不一樣(?similarity_concept and ?similarity_form)

(defrule contiguity_principle_6
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  (similarity_concept (icf ?similarity_concept))
  (similarity_form (icf ?similarity_form))
  ?fact <- (ie (issue ?i&:(neq ?i ?my_i2))(concept ?c&:(eq ?c ?similarity_concept))
            (form ?f&:(eq ?f ?similarity_form)?x)(preference ?p))
=>
  (assert (ie(issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url)))

```

```

(assert (map (issue_1 ?i)(concept_1 ?c)(form_1 ?f ?x)(link_type contiguity)
            (issue_2 ?my_i2)(concept_2 ?my_c2)(form_2 ?my_f2 ?url)))
(assert (selected_ie(principle contiguity)(issue ?i)(concept ?c)(form ?f ?x)(preference ?p)))
(save-facts c:/DIM/RA_3.clp MAIN::ie MAIN::map)
)

// 此 Rule 為藉由 ICF_Maps 去找尋其它相鄰性的 issue 值後, 再產生新的 IE
// 當相鄰性 issue 值一樣, form 值與 concept 值一樣(?contiguity_issue, ?my_c2, ?my_f2)

(defrule contiguity_principle_7
  (temporary_ie (issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url))
  (contiguity_issue (icf ?contiguity_issue))
  ?fact <- (ie(issue ?i&:(eq ?i ?contiguity_issue))(concept ?c&:(eq ?c ?my_c2)) (form ?f&:(eq ?f ?my_f2)?x)
            (preference ?p))
=>
  (assert (ie(issue ?my_i2)(concept ?my_c2)(form ?my_f2 ?url)))
  (assert (map (issue_1 ?i)(concept_1 ?c)(form_1 ?f ?x)(link_type contiguity)
              (issue_2 ?my_i2)(concept_2 ?my_c2)(form_2 ?my_f2 ?url)))
  (assert (selected_ie(principle contiguity)(issue ?i)(concept ?c)(form ?f ?x)(preference ?p)))
  (save-facts c:/DIM/RA_3.clp MAIN::ie MAIN::map)
)

```



A.2 場記代理人 (RAS)

```
.....
,,,,,,,,, Defining RAS's template

// script-holder
(deftemplate script-holder (slot issue) (slot time) (slot role-number) (multislot role-id) (multislot principle)
                          (multislot tolerance))

// Idea entity
(deftemplate ie(slot issue)(multislot concept)(multislot form) (slot preference (default 1)))

// IE_Map
(deftemplate map(slot issue_1) (multislot concept_1) (multislot form_1)(slot link_type)(slot issue_2) (multislot concept_2)
              (multislot form_2))

// link_count
(deftemplate link-count (multislot ie)(slot count))

// link_ie
(deftemplate link-ie (multislot ie))

// ACLMessage
(deftemplate ACLMessage
  (slot communicative-act)
  (slot sender)
  (multislot receiver)
  (slot reply-with)
  (slot in-reply-to)
  (slot envelope)
  (slot conversation-id) (slot protocol)
  (slot language) (slot ontology) (slot content)
  (slot encoding) (multislot reply-to) (slot reply-by)
)

.....
,,,,,,,,, Defining RAS's global value

(defglobal ?*ie_list* = (create$ ))
(defglobal ?*num* = 0)
(defglobal ?*t1* = (time))

.....
,,,,,,,,, Requesting RAS to save an IE_Map into I_IE_Map from RAS

(defrule request_ie_map
  ?m <- (ACLMessage (communicative-act REQUEST) (sender ?s) (content ?c) (receiver ?r))
=>
  // 將輸入的 string 改為 list, 並改成 IE fact
  (assert (map (issue_1 (nth$ 1 (explode$ ?c)))
              (concept_1 (nth$ 2 (explode$ ?c)))
              (form_1 (nth$ 3 (explode$ ?c))(nth$ 4 (explode$ ?c)))
              (link_type (nth$ 5 (explode$ ?c)))
              (issue_2 (nth$ 6 (explode$ ?c)))
              (concept_2 (nth$ 7 (explode$ ?c)))
              (form_2 (nth$ 8 (explode$ ?c))(nth$ 9 (explode$ ?c)))
              )
  )
)
```

```

(assert (ie (issue (nth$ 1 (explode$ ?c)))
           (concept (nth$ 2 (explode$ ?c)))
           (form (nth$ 3 (explode$ ?c))(nth$ 4 (explode$ ?c)))
          )
)

(assert (ie (issue (nth$ 6 (explode$ ?c)))
           (concept (nth$ 7 (explode$ ?c)))
           (form (nth$ 8 (explode$ ?c))(nth$ 9 (explode$ ?c)))
          )
)

(assert (ACLMMessage (communicative-act INFORM) (sender ?r) (receiver ?s) (content ?c) )
        (retract ?m)
        (save-facts C:/DIM/RAS/I_IE_Map.clp map ie)
)

.....
;;;;;;; Proposing an IE which has the highest-number links

// define a query for counting IE_Map's right IE's and left IE's link_type

(defquery count-link-assignments-left
  (declare (variables ?issue ?concept ?form ?form_url)
           (map (issue_1 ?issue) (concept_1 ?concept)(form_1 ?form ?form_url)))

(defquery count-link-assignments-right
  (declare (variables ?issue ?concept ?form ?form_url)
           (map (issue_2 ?issue) (concept_2 ?concept)(form_2 ?form ?form_url)))

// Defining the function for counting the link number of left IE and right IE in IE_Maps

(deffunction search_link_number (?issue ?concept ?form ?form_url)
  (bind ?count_l (count-query-results count-link-assignments-left ?issue ?concept ?form ?form_url))
  (bind ?count_r (count-query-results count-link-assignments-right ?issue ?concept ?form ?form_url))
  (bind ?count (+ ?count_l ?count_r))
  (if (neq ?count 0) then
      (assert (link-count (ie ?issue ?concept ?form ?form_url)(count ?count)))
      (printout t ?issue " " ?concept " " ?form " " ?form_url " is " ?count crlf)
    )
)

// Searching the link number of IEs in all IE_Maps

(defrule search_link_using_ie
  ?m <- (ACLMMessage (communicative-act CFP) (sender ?s) (content ?c) (receiver ?r))
  (ie (issue ?x)(concept ?y)(form ?z ?url))
  (not (link-count (ie ?x ?y ?z ?url)))
=>
  (search_link_number ?x ?y ?z ?url)
)

// Searching the highest-number of link number of IEs in all IE_Maps

```

```

(defrule print-highest-link-assignment
  (declare (salience -10))
  ?m <- (ACLMessage (communicative-act CFP) (sender ?s) (content ?c) (receiver ?r))
  (link-count (ie ?issue ?concept ?form ?form_url)(count ?count))
  // Comparing IE's link number, and then matching the IEs with highest-link number
  (not (link-count (count ?count2&:(> ?count2 ?count))))
=>
  (printout t "Most active link found " ?issue"_" ?concept"_" ?form "_" ?form_url crlf)
  (printout t "The number is " ?count crlf)
  (bind ?*num* (+ ?*num* 1))
  // Inserting ?issue ?concept ?form within link-count into ?ie_list
  (bind ?*ie_list* (insert$ ?*ie_list* (+ (length$ ?*ie_list*) 1) (create$ ?issue ?concept ?form ?form_url)))
)

// Selecting the only highest-link IE randomly
(deffunction time_save ()
  (bind ?t2 (- (time) ?*t1*))
  (bind ?file (implode$ (create$ c:/DIM/RAS/ ?t2 .clp)))
  (save-facts ?file map ie))

// Selecting the only highest-link IE randomly
(defrule select_most_link_IE
  (declare (salience -100))
  (link-count (ie ?issue ?concept ?form ?form_url)(count ?count))
  ?m <- (ACLMessage (communicative-act CFP) (sender ?s) (content ?c) (receiver ?r))
=>
  (printout t ?*num* crlf)
  (bind ?rd (* (mod (random) ?*num*) 4))
  (bind ?output_ie
    (implode$
      (create$ (nth$ (+ ?rd 1) ?*ie_list*)(nth$ (+ ?rd 2) ?*ie_list*)(nth$ (+ ?rd 3) ?*ie_list*)(nth$ (+ ?rd
        4) ?*ie_list*))
    )
  )
  (retract ?m)
  (assert (ACLMessage (communicative-act PROPOSE) (sender ?r) (receiver ?s) (content ?output_ie)))
  (time_save)
  (printout t "The selected IE is " "(" ?output_ie ")" crlf)
)

.....
,,,,,,,,, Sending message to Dummy agent

(defrule send-a-message
// When a message is asserted whose sender is this agent, the message is sent and then retracted from the knowledge base

(MyAgent (name ?n))
  ?m <- (ACLMessage (sender ?n))
=>
  (send ?m)
)

(reset)

(run)

```



```

(assert (ie (issue (nth$ 1 (explode$ ?c)))
           (concept (nth$ 2 (explode$ ?c)))
           (form (nth$ 3 (explode$ ?c))(nth$ 4 (explode$ ?c)))
          )
)
)
(assert (ie (issue (nth$ 6 (explode$ ?c)))
           (concept (nth$ 7 (explode$ ?c)))
           (form (nth$ 8 (explode$ ?c))(nth$ 9 (explode$ ?c)))
          )
)
)

(assert (ACLMMessage (communicative-act INFORM) (sender ?r) (receiver ?s) (content ?c) )
(retract ?m)
(save-facts C:/DIM/ScA/E_IE_Map.clp map ie)
)

.....
,,,,,, Proposing an IE which has the highest-number links

(load-facts c:/DIM/ScA/E_IE_Map.clp)

// Defining a query for counting IE_Map's right IE's and left IE's link_type

(defquery count-link-assignments-left
  (declare (variables ?issue ?concept ?form ?form_url))
  (map (issue_1 ?issue) (concept_1 ?concept)(form_1 ?form ?form_url)))

(defquery count-link-assignments-right
  (declare (variables ?issue ?concept ?form ?form_url))
  (map (issue_2 ?issue) (concept_2 ?concept)(form_2 ?form ?form_url)))

// Defining the function for counting the link number of left IE and right IE in IE_Maps

(deffunction search_link_number (?issue ?concept ?form ?form_url)
  (bind ?count_l (count-query-results count-link-assignments-left ?issue ?concept ?form ?form_url))
  (bind ?count_r (count-query-results count-link-assignments-right ?issue ?concept ?form ?form_url))
  (bind ?count (+ ?count_l ?count_r))
  (if (neq ?count 0) then
      (assert (link-count (ie ?issue ?concept ?form )(count ?count)))
      (printout t ?issue " " ?concept " " ?form " " ?form_url " is " ?count crlf)
    )
)

// Searching the link number of IEs in all IE_Maps

(defrule search_link_using_ie
  (ie (issue ?x)(concept ?y)(form ?z ?url))
  (not (link-count (ie ?x ?y ?z)))
=>
  (search_link_number ?x ?y ?z ?url)
)

// Searching the highest-number of link number of IEs in all IE_Maps

(defrule print-highest-link-assignment
  (declare (salience -10))
  (link-count (ie ?issue ?concept ?form)(count ?count))

```

```

// Comparing IE's link number, and then matching the IEs with highest-link number
(not (link-count (count ?count2&:(> ?count2 ?count))))
=>
(printout t "Most active link found " ?issue"_" ?concept"_" ?form crlf)
(printout t "The number is " ?count crlf)
(bind ?*num* (+ ?*num* 1))
//Inserting ?issue ?concept ?form within link-count into ?ie_list
(bind ?*ie_list* (insert$ ?*ie_list* (+ (length$ ?*ie_list*) 1) (create$ ?issue ?concept ?form)))
)

// Selecting the only highest-link IE randomly

(defrule select_most_link_IE
  (declare (salience 100))
  (link-count (ie ?issue ?concept ?form)(count ?count))
  ?m <- (ACLMessage (communicative-act CFP) (sender ?s) (content ?c) (receiver ?r))
=>
  (printout t ?*num* crlf)
  (bind ?rd (* (mod (random) ?*num*) 3))
  (bind ?output_ie
    (implode$
      (create$ (nth$ (+ ?rd 1) ?*ie_list*)(nth$ (+ ?rd 2) ?*ie_list*)(nth$ (+ ?rd 3) ?*ie_list*))
    )
  )
  (retract ?m)
  (assert (ACLMessage (communicative-act PROPOSE) (sender ?r) (receiver ?s) (content ?output_ie)))
  (printout t "The selected IE is " "(" ?output_ie ")" crlf)
)

.....
,,,,,,; Sending message to Dummy agent

(defrule send-a-message
// When a message is asserted whose sender is this agent, the message is sent and then retracted from the knowledge base

(MyAgent (name ?n))
  ?m <- (ACLMessage (sender ?n))
=>
  (send ?m)
)

(reset)

(run)

```

A.4 舞台代理人 (StA)

```
.....
;; Defining StA's template

//Play
(deftemplate play (slot paly-name) (slot time-duration) (slot user-number) (multislot user-order) (slot scene-number)
  (multislot scenes) (multislot time) (multislot role-skill) (multislot role-number))

// Idea entity
(deftemplate ie(slot issue)(multislot concept)(multislot form) (slot preference (default 1)))

// IE_Map
(deftemplate map(slot issue_1) (multislot concept_1) (multislot form_1)(slot link_type)(slot issue_2) (multislot concept_2)
  (multislot form_2))

// time -duration
(deftemplate duration (slot my_time))

// issue
(deftemplate issue (slot my_issue))

// ACLMessage
(deftemplate ACLMessage
  (slot communicative-act)
  (slot sender)
  (multislot receiver)
  (slot reply-with)
  (slot in-reply-to)
  (slot envelope)
  (slot conversation-id) (slot protocol)
  (slot language) (slot ontology) (slot content)
  (slot encoding) (multislot reply-to) (slot reply-by)
)

.....
;; Defining StA's global value

(defglobal ?*time-begin* = (time))

.....
;;Requesting StA to start the play and save the outcome to the StA's clip file based on the time duration

(defrule request_ie_map
  ?m <- (ACLMessage (communicative-act REQUEST) (sender ?s) (content ?c) (receiver ?r))
=>
  .....
  .....
  // 將輸入的 string 改為 list, 並改成不同樣版的 fact
  (assert (time-duration (my_time (nth$ 1 (explode$ ?c))))))
  (assert (issue (my_issue (nth$ 2 (explode$ ?c))))))
  (assert (ACLMessage (communicative-act INFORM) (sender ?r) (receiver ?s) (content ?c) ))
  (retract ?m)
  (save-facts C:/DIM/StA/E_IE_Map.clp map ie)
)
```

```
.....  
.....  
..... Send message to Dummy agent
```

```
(defrule send-a-message  
// When a message is asserted whose sender is this agent, the message is sent and then retracted from the knowledge base  
  (MyAgent (name ?n))  
  ?m <- (ACLMMessage (sender ?n))  
=>  
  (send ?m)  
)  
  
(reset)  
  
(run)
```



附錄 B：角色代理人工作記憶案例

角色代理人(RA)

// IE template

(ie (issue landscape)(concept connection)(form yard http://lai.cyut.edu.tw/DIM/RA1_1)(preference 4))
(ie (issue landscape)(concept view-extension)(form greenhouse http:// lai.cyut.edu.tw/DIM/RA1_2)(preference 5))
(ie (issue landscape)(concept view-extension)(form terrace http:// lai.cyut.edu.tw/DIM/RA1_3)(preference 4))
(ie (issue landscape)(concept view-extension)(form window http:// lai.cyut.edu.tw/DIM/RA1_4)(preference 3))
(ie (issue landscape)(concept view-extension)(form window http:// lai.cyut.edu.tw/DIM/RA1_15)(preference 3))
(ie (issue light)(concept opening)(form terrace http:// lai.cyut.edu.tw/DIM/RA1_3))
(ie (issue light)(concept view-focus)(form window http:// lai.cyut.edu.tw/DIM/RA1_6))
(ie (issue light)(concept superimpose)(form window http:// lai.cyut.edu.tw/DIM/RA1_4))
(ie (issue light)(concept sunshade)(form terrace http:// lai.cyut.edu.tw/DIM/RA1_7)(preference 2))
(ie (issue circulation)(concept connection)(form lobby http:// lai.cyut.edu.tw/DIM/RA1_8)(preference 3))
(ie (issue circulation)(concept union)(form window http:// lai.cyut.edu.tw/DIM/RA1_5)(preference 2))
(ie (issue circulation)(concept public-space)(form window http:// lai.cyut.edu.tw/DIM/RA1_5)(preference 5))
(ie (issue circulation)(concept expansion)(form greenhouse http:// lai.cyut.edu.tw/DIM/RA1_2)(preference 4))
(ie (issue circulation)(concept public-space)(form stair http:// lai.cyut.edu.tw/DIM/RA1_11)(preference 5))
(ie (issue circulation)(concept union)(form bridge http:// lai.cyut.edu.tw/DIM/RA1_4)(preference 4))
(ie (issue circulation)(concept public-space)(form yard http:// lai.cyut.edu.tw/DIM/RA1_12)(preference 3))
(ie (issue circulation)(concept public-space)(form platform http:// lai.cyut.edu.tw/DIM/RA1_18)(preference 2))
(ie (issue landscape)(concept view-extension)(form platform http:// lai.cyut.edu.tw/DIM/RA1_23)(preference 5))
(ie (issue landscape)(concept view-focus)(form yard http:// lai.cyut.edu.tw/DIM/RA1_13)(preference 4))
(ie (issue landscape)(concept view-focus)(form garden http://RA1_14))
(ie (issue light)(concept opening)(form window http:// lai.cyut.edu.tw/DIM/RA1_6)(preference 2))
(ie (issue light)(concept opening)(form skylight http:// lai.cyut.edu.tw/DIM/RA1_5)(preference 3))
(ie (issue light)(concept opening)(form skylight http:// lai.cyut.edu.tw/DIM/RA1_17))

// ICF Map

// Issue_Map template

(issue_map (issue_1 ventilation)(link_type contiguity)(issue_2 light))
(issue_map (issue_1 light)(link_type contiguity)(issue_2 landscape))
(issue_map (issue_1 landscape)(link_type similarity)(issue_2 view))
(issue_map (issue_1 in-between)(link_type similarity)(issue_2 transition))

// Concept_Map template

(concept_map (concept_1 private-space)(link_type similarity)(concept_2 personal-space))
(concept_map (concept_1 connection)(link_type similarity)(concept_2 union))
(concept_map (concept_1 interaction)(link_type similarity)(concept_2 integration))
(concept_map (concept_1 view-focus)(link_type similarity)(concept_2 landmark))
(concept_map (concept_1 partition)(link_type similarity)(concept_2 union))
(concept_map (concept_1 connection)(link_type contrast)(concept_2 partition))
(concept_map (concept_1 partition)(link_type contrast)(concept_2 union))
(concept_map (concept_1 private-space)(link_type contrast)(concept_2 public-space))
(concept_map (concept_1 personal-space)(link_type contrast)(concept_2 public-space))
(concept_map (concept_1 view-extension)(link_type contrast)(concept_2 view-focus))
(concept_map (concept_1 scattering)(link_type contrast)(concept_2 centralization))

```
// Form_Map template
(form_map (form_1 platform)(link_type similarity)(form_2 terrace))
(form_map (form_1 courtyard)(link_type similarity)(form_2 garden))
(form_map (form_1 garden)(link_type similarity)(form_2 bamboo-grove))
(form_map (form_1 courtyard)(link_type similarity)(form_2 yard))
(form_map (form_1 garden)(link_type similarity)(form_2 yard))
(form_map (form_1 platform)(link_type similarity)(form_2 terrace))
```

```
// ICF Dictionary list
(sightseeing landscape)
(view landscape)
(sunlight light)
(wind ventilation)
(buffer in-between)
(buffer transition)
(movement circulation)
(nil sunshade)
(hole opening)
(cavity opening)
(link connection)
(nil view-extension)
(nil view-focus)
(extension expansion)
(combination union)
(open-space public-space)
(overlay superimpose)
(backyard yard)
(hothouse greenhouse)
(glass window)
(veranda terrace)
(hall lobby)
```



簡 歷

作者1988年畢業於東海大學建築系，1994畢業於美國康乃爾大學建築與都市設計研究所，於2002年考取交通大學土木工程研究所（庚組：數位建築）博士班。目前任教於朝陽科技大學建築系專任講師，與淡江大學建築系兼任講師，且曾經在新竹市、嘉義縣和宜蘭縣擔任環境景觀、都市設計顧問或校園規劃小組顧問等。此外，作者也受邀參與國際性學術組織的評審委員，包括2005 Far Eastern International Digital Architectural Design (2005年國際遠東數位建築獎) 第一階段的評審委員，和The 10th International Congress Society of Digital Graphics (Sigradi 2006) 論文評審委員。

就讀博士期間，作者對於如何利用數位化技術的機制（設計運算與數位媒材），以啟發建築設計的創意與思考有高度的興趣，並在國際性的研討會與期刊發表許多相關的研究論文。在2006年6月，作者順利完成博士學位，並獲得交大建築研究所的「交大建築研究獎」首獎。至今，作者發表的學術論文與著作如下：

期刊論文

1. Lai, Ih-Cheng (2005), Dynamic Idea Maps: A Framework for Linking Ideas with Cases during Brainstorming, *International Journal of Architectural Computing (IJAC)*, Vol 3 (4), pp. 229-247
2. 賴怡成 (2005), 探討想法聯想過程中想法與設計案例之間的動態連結, *朝陽設計學報第六期*, pp. 33-47
3. Lai, Ih-Cheng and Cheng, Teng-Wen (2006), A Distributed Linking System for Supporting Idea Association in the Conceptual Design Stage, *Design Studies*, accepted in 02/14/2006, in press (SCI)

研討會論文

1. 賴怡成 (2002), 探索電腦輔助建築設計的教學方法 - 真實與虛擬的設計空間的重疊與互動, 中華民國設計學會 2002 年設計學術研究成果研討會
2. Lai, Ih-Cheng (2002), Merging Design Thinking with Digital Media to Overlap Real and Virtual Design Worlds, Connecting the Real and the Virtual - design e-ducation. *20th eCAADe Conference Proceedings*. Warsaw (Poland), pp. 114-117
3. Lai, Ih-Cheng and Chang, Teng-Wen (2003). Companing Physical Space with Virtual Space—A Co-existence Approach, *the Proceedings of 8th Conference on Computer Aided Architectural Design Research (CAADRRIA)*, Thailand. pp. 359-370. NSC-91-2422-H-009
4. Chang, Teng-Wen and Lai, Ih-Cheng (2003). Reflective Navigation in Information Landscape—A 3D Museum for Civic Buildings, *the Proceedings of 8th Conference on Computer Aided Architectural Design Research (CAADRRIA)*, Thailand. pp. 641-650. NSC-91-2422-H-009

5. Lai, Ih-Cheng and Lin, Jun-Hong (2003). *e-Space: An Introductory CAAD Course*, the conference poster brochure on *10th CAAD Future*, Taiwan.(十五分鐘口頭報告短篇論文)
6. Lai, Ih-Cheng (2004). Framework for Case-Based Reasoning to Support Idea Association in a Brainstorming Session, *the Proceedings of 9th Conference on Computer Aided Architectural Design Research (CAADRIA)*, Seoul (Korea). pp. 209-222 (**Best paper in 2004 CAADRIA Conference**)
7. Lai, Ih-Cheng (2004). A Teaching Model for Integrating Conventional Design Curriculum with Digital Media, *the Proceedings of 9th Conference on Computer Aided Architectural Design Research (CAADRIA)*, Seoul (Korea). pp. 467-476
8. Lai, Ih-Cheng (2004). Dynamic Linkages between Ideas and Cases, *the Proceedings of 7th International Conference on Design & Decision Support Systems in Architecture and Urban Planning*, Netherlands (Ruwenberg), pp. 225-238
9. 賴怡成, 張登文, 林俊宏, 呂凱慈, 簡兆芝, 曹世緯, 邵唯晏, 李思廣 (2004), 一個實體與虛體的共存策略, 中華民國建築學會第十六屆第一次建築研究成果發表會
10. 張登文, 賴怡成, 施勝誠, 黃英修 (2004), 合作式設計之動態角色代理者系統, 中華民國建築學會第十六屆第一次建築研究成果發表會
11. Lai, Ih-Cheng and Chang, Teng-Wen (2004). Dynamic Interactions Between Navigators and Information. *eCAADe 2004 Conference Proceedings*. Copenhagen(Denmark) 15-18 September 2004, pp. 58-65
12. Lai, Ih-Cheng, (2004), Interactive Patterns for Associating Ideas during Brainstorming, *eCAADe 2004 Conference Proceedings*. Copenhagen(Denmark) 15-18 September 2004, pp. 256-261
13. 賴怡成, 林俊宏 (2004), 探索數位媒材結合傳統建築設計教學的教學模型, 中華民國建築學會第十六屆第二次建築研究成果發表會
14. Lai, Ih-Cheng (2005). Mapmaker - Linking Play among Ideas, Cases and Links, *the Proceedings of 10th Conference on Computer Aided Architectural Design Research (CAADRIA)*, New Delhi (India), pp. 222-227
15. Lai, Ih-Cheng, Kao, Kun-Tai and Chang, Teng-Wen (2005). Role Playing for Linking Ideas in the Idea Association Process, *the Proceedings of 10th Conference on Computer Aided Architectural Design Research (CAADRIA)*, New Delhi (India). pp. 119-129
16. Kao, Kun-Tai, Chang, Teng-Wen and Lai, Ih-Cheng (2005). Learning Construction Decision Making with Arbitratot - Competing and Evolving in Dynamic Role Interplay, *the Proceeding of 22nd CIB W78 Conference on Information Technology in Construction*, Dresden (Germany)
17. 賴怡成 (2005), 探討動態性的想法連結輔助設計工作室教學, 中華民國建築學會第十七屆第一次建築研究成果發表會
18. 賴怡成 (2005), 以數位媒材探索運動狀態在建築空間與形式上演化, 中華民國建築學會第十七屆第一次建築研究成果發表會
19. Lai, Ih-Cheng, (2005), Infilling Time into Space-A Pedagogical Approach for Evolving Space Using Digital Media, *eCAADe 2005 Conference Proceedings*. Lisbon (Portugal), pp.147-154
20. Chang, Teng-Wen and Lai, Ih-Cheng (2005). A Design Tool for Supporting Idea Association at Small Design Offices, *International Design Congress (IDC) - IASDR 2005*, full paper review,

Yunlin (Taiwan)

21. 賴怡成 (2005), 探討代理人在想法聯想過程中角色扮演的溝通模式, 中華民國建築學會第十七屆第二次建築研究成果發表會
22. Lai, Ih-Cheng (2006). Agent Communication for Role Playing in the Idea Association Process, *the Proceedings of 11th Conference on Computer Aided Architectural Design Research (CAADRIA)*, Kumamoto (Japan) , pp. 151-160
23. 賴怡成 (2006), 從仿生學到數位演化：一個使用數位媒材的設計工作室教學法, 中華民國建築學會第十八屆第一次建築研究成果發表會
24. Lai, Ih-Cheng and Chang, Teng-Wen (2006). Linking Interaction for Supporting Idea Association, *eCAADe 2006 Conference Proceedings*. Volos (Greece) (accept as the long paper)

書籍文章

1. 賴怡成 (2005), Piranesi 與 Bach 的對話 - 數位建築國外案例介紹—**台灣建築雜誌** 11/2005, pp. 76-79
2. 賴怡成 (2003), 尋找數位時代城市空間的特質與現象-Space, Body and Perception —**台灣建築雜誌** 03/2003, pp. 54-57
3. Liu, Yu-Tung, Haung, Ying-Hsiu, Lai, Ih-Cheng and Gao, Wan-Ping (eds.) (2003), *Developing Digital Architecture -2002 FEIDAD Award*, BIRKHAUSER press
4. 劉育東、黃英修、賴怡成、高宛屏 (2003), *數位建築發展-2002 遠東國際數位建築獎* (共同編輯), 上博國際圖書有限公司出版
5. Lai, Ih-Cheng and Lin, Jun-Hong (2003), *e-Space: An Introductory CAAD Course*, *數位建築教育*, pp. 168-169, 邱茂林編, 田園城市出版
6. Liu, Yu-Tung, Lai, Ih-Cheng, Wu, Pei-Ling and Wu, Yi-Ching (eds.)(2004), *Diversifying Digital Architecture -2003 FEIDAD Award*, BIRKHAUSER press
7. 劉育東、賴怡成、吳佩玲、吳逸菁 (2004), *數位建築多樣性-2003 遠東國際數位建築獎* (共同編輯), 田園城市出版

技術報告

1. 張登文、賴怡成, 2002-2003, 台灣城市建築 3D 數位博物館—子計畫三：台灣公共建築時空館, 國科會專題研究計畫報告, NSC-91-2422-H-009-3004-, 2003 年 3 月
2. 張登文、賴怡成, 2002-2003 子計畫四：建築設計協同作業之角色模式與代理者系統研究, 國科會專題研究計畫報告, (NSC-91-2211-E-009-054-) 2003 年 10 月
3. 張登文、陳冠燁、賴怡成、張粧亭, 2003-2004, 三維幾何示意圖系統之運算性及可能性之研究 - 以 Eisenman 為例, 國科會專題研究計畫報告 (NSC92-2211-E-009-060)
4. 張登文、賴怡成 (共同主持人), 2004-2005, 分散式案例式推理在早期設計階段之研究--以小型設計事務所為例, 國科會專題研究計畫報告 (NSC-93-2211-E-224-028-)

除了上述的學術著作外，作者自1994年從美國康乃爾大學回國後，曾經在國內不同的建築師事務所擔任設計經理、設計總監和設計顧問，在這些實務經驗的過程中，作者參加許多國內重要的建築設計競圖，並獲得許多建築設計獎項，而這些建築設計大部份已被實際完工，並刊登在國內相關的建築設計雜誌，這些建築作品與得獎紀錄詳見如下：

建築作品與雜誌刊登

1. 國立高雄第一科技大學外語學院教學大樓 — 登於 *TA-台灣建築雜誌* 2005/06 (Team member)
2. 台中市中山地政辦公大樓—登於 *建築師雜誌* 2003/03 (Design director)
3. 台中市惠文國民小學, 2001 (Design director)
4. 國立高雄科技大學學生宿舍大樓—登於 *建築師雜誌* 2001/01 (Team member)
5. 台北縣立清水中學科學館及圖書館—登於 *建築師雜誌* 2000/06 (Team member)
6. 淡江大學舊化學館重建工程—登於 *TA-台灣建築雜誌* 1999/11 (Team member)
7. 國立高雄科技大學教學行政大樓—登於 *建築師雜誌* 1997/03 (ps: 此作品獲台灣省 87 年優良建築物獎) (Project manager)
8. 宜蘭厝—登於 *空間雜誌* 1997 (Project manager)

建築設計獲獎作品紀錄

1. 台灣大學化學館大樓興建工程第二名(負責整體規劃與建築設計), 2002
2. 宜蘭縣政府文化局第二館區興建工程第二名(負責整體規劃與建築設計), 2002—登於 *TA-台灣建築雜誌* 09/2002
3. 台中市慎齋佛教小學興建工程競圖第一名(負責整體規劃與建築設計), 2001—登於 *建築師雜誌* 07/2002
4. 國立高雄科技大學游泳池興建工程競圖第一名(負責整體規劃與建築設計), 2000
5. 台中市文小 60 小學興建工程競圖第一名(負責整體規劃與建築設計), 1999
6. 台中市中山地政辦公大樓興建工程競圖第一名(負責建築造型與立面設計), 1999—登於 *建築師雜誌* 12/1999 與 2003/03
7. 台中市惠文國民小學興建工程競圖第一名(負責整體規劃與建築設計), 1998—登於 *建築師雜誌* 04/2000
8. 宏國建設桃園生活館興建工程競圖第一名(負責整體規劃與建築設計), 1998
9. 交通部,電信局大樓新建工程競圖第三名(負責建築造型與立面設計), 1997
10. 國立台灣大學醫學院附設醫院興建會議中心暨研究大樓競圖第二名(負責建築造型與立面設計), 1997
11. 淡江大學舊化學館重建工程競圖第一名(負責建築造型與立面設計), 1996
12. 國立傳統藝術中心第一期建築工程行政中心及宿舍區設計競圖第三名(負責整體規劃與建築設計), 1996—登於 *建築師雜誌*
13. 台灣省立桃園醫院新屋分院工程競圖第二名(負責建築造型與立面設計), 1996

14. 高雄市立楠梓特殊學校新建工程競圖第一名(負責建築造型與立面設計), 1996
15. 高雄市政府警察局消防大樓, 鼎金派出所, 及里鄰活動中心新建工程競圖第一名(負責建築造型與立面設計), 1992
16. 台灣省建築師公會競圖大學組第二名(第一名從缺), 1987—登於**建築師雜誌** 02/1988

此外，從 1995 到迄今，作者也陸續受邀參加台灣各大學建築系與建築研究所的評圖，包括交通大學建築研究所、東海大學建築研究所、東海大學建築系、淡江大學建築系、中原大學建築系、銘傳大學建築系、雲林科技大學空間設計系與蘭陽技術學院建築系等。同時，在 2003 與 2004 年也受邀參加國際性的建築設計工作營與評審，分別包括淡江大學建築系的 Archigram 國際工作營，與韓國 Dongseo University 建築系舉辦的 International Architectural Design Workshop in International Architectural Cultural Festival (BIACF)。

