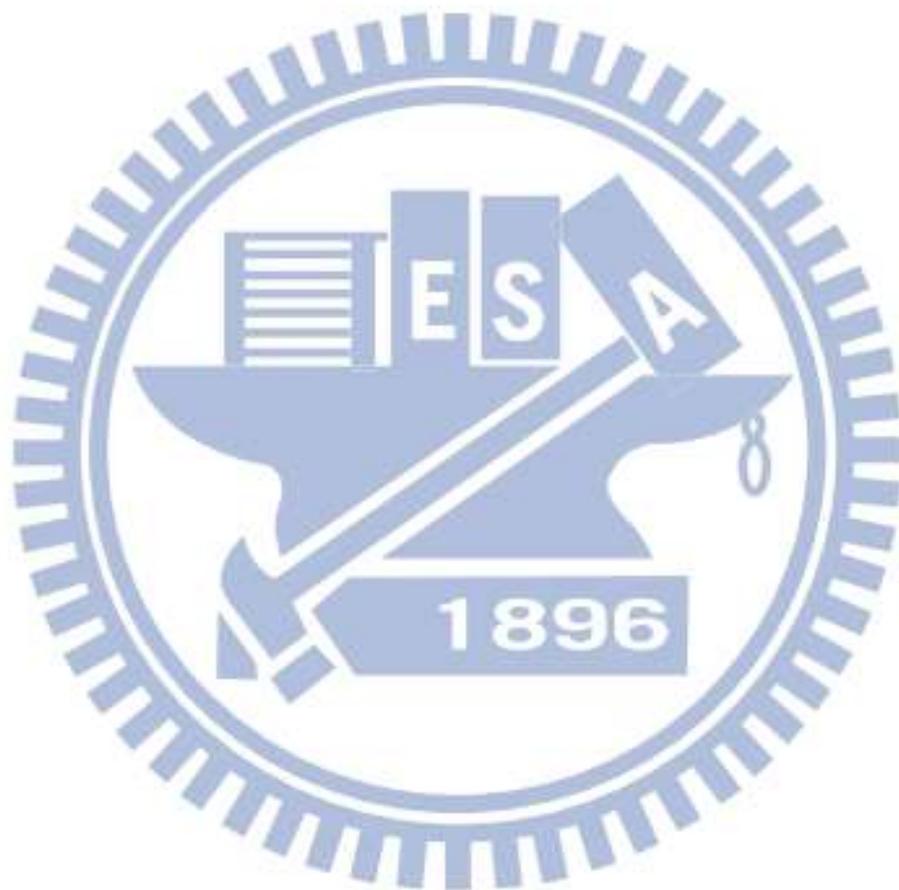


目次

1.	導論	4
1.1.	智慧空間的浮現	4
1.2.	智慧空間的設計媒材	5
1.3.	研究問題與目的	6
1.4.	研究方法與步驟	7
2.	文獻回顧	9
2.1.	智慧空間的浮現	9
2.1.1.	資訊系統的空間化	10
2.1.2.	建築空間的資訊化	14
2.2.	設計媒材	21
2.2.1.	空間設計媒材	22
2.2.2.	資訊設計媒材	26
3.	應用案例分析	36
3.1.	案例分析方式	36
3.2.	互動建築	37
3.2.1.	案例一：Hyposurface Mediafacede	38
3.2.2.	案例二：Muscle Tower II	41
3.2.3.	案例三：Dynamic Façade	44
3.2.4.	案例四：Kunsthaus Graz	46
3.3.	資訊空間	48
3.3.1.	案例一：智慧家具	49
3.3.2.	案例二：互動地板	51
3.3.3.	案例三：互動牆面	53

3.3.4.	案例四：擴增實境廚房	56
4.	工具案例分析	59
4.1.	簡化的空間設計環境	59
4.2.	視覺化的資訊設計環境	62
4.3.	空間元件與資訊元件的結合機制	64
4.4.	空間變化的視覺回饋	67
4.5.	互動功能的視覺回饋	68
4.6.	討論	70
5.	系統架構與實作	72
5.1.	系統架構	72
5.1.1.	元件庫	75
5.1.2.	整合式設計環境	80
5.2.	系統實作	86
5.2.1.	實作語言：Processing	86
5.2.2.	元件庫	88
5.2.3.	空間設計模式	89
5.2.4.	資訊設計模式	92
5.2.5.	模擬模式	96
5.3.	測試案例	96
5.3.1.	測試 1：與攝影機結合的互動牆面設計	97
5.3.2.	測試 2：結合複合感測器的智慧建築立面設計	99
5.3.3.	討論	103
6.	結論	104
	參考文獻	110



1. 導論

個人電腦宣告了資訊時代的到來，人們的日常生活與各種數位資訊產物的關聯越來越密切。不論是資訊的取得方式從實體書報進入網路搜尋，通訊與溝通的方式端賴著數位的硬體與軟體，工作與私人的訊息往來皆以個人電腦為介面。人們的生活周遭充斥著各種數位運算設備與裝置，這些裝置與設備進入人們的生活，也進入了人們生活的空間。透過資訊系統的驅動，嵌有這些數位運算裝置的空間能藉由人工智慧或是程式邏輯來回應空間使用者的需求或判斷自身的行動。這樣的新空間正不斷地出現在人們的生活周遭，與人們互動、滿足人們的需求也影響人們的生活。

1.1. 智慧空間的浮現

在這樣的發展下我們看到了智慧空間的浮現，本研究將智慧空間定義為結合了建築空間與互動資訊系統的新空間，資訊系統被隱藏在空間或是空間的元件中，而得以去介面化、直覺化與生活化 [Park et al., 2004]，讓使用者在操作時是直接與這些系統互動，例如：碰觸互動牆面或是在互動桌上放置物體，而不是透過制式的鍵盤或滑鼠等介面 [Rekimoto & Sato, 1999; Klemmer et al., 2001; Shen, Everitt & Ryall, 2003]；透過將資訊系統嵌入人們熟知的生活物件與空間中，讓使用者不需要學習便可以直覺地與資訊系統互動，讓資訊直接進入人們的生活被應用並與生活經驗相融合而不是面對與操作一部機器 [Ishii & Ullmer, 1997; Tennehouse, 2000; Patten et al., 2001; Ratti et al., 2002; Ryokai, Marti & Ishii, 2004; Yoon et al., 2004; Bonanni et al., 2005]。

這樣的新空間也讓建築物因著資訊系統的嵌入在其間的運算與控制，而得以智慧化、自動化與多樣化。智慧化的建築空間將來自感測器所收集的環境條件與狀況經過分

析與計算後自我決定對應的決策 [Schoh, 2006]、自動化建築空間讓相對關決策透過各種傳動器自動地進行環境條件(如:溫度、溼度、照明等)的控制而不需經過使用者的操作 [Strehlke, Ochsendorf & Bahr, 2004; Huang & Waldvogel, 2004; Chiu, 2006; Giselbrecht, 2010]、多樣化的建築空間透過媒體立面的變化改變建築的外觀皮層或是各種電腦控制的機械元件動態地改變結構而得以呈現多樣的風貌與空間經驗 [Oosterhuis, 2003; Bilorina & Oosterhuis, 2005; dECOi, 2011]，讓建築空間從單純的靜態的空間轉而成為複雜的動態空間 [Mitchell, 2006]。

在世界各地也有許多的研究單位都在進行相關的研究，例如：學校有卡內基美侖大學、麻省理工學院、哈佛大學、史丹佛大學、瑞士聯邦理工學院、喬治亞理工學院、香港理工大學、東京大學；研究單位有 IBM、微軟、飛利浦、新力公司電腦科學實驗室、日本電話公司等等 [Chiu, 2005]，在台灣也有許多實驗室投入這方面的研究，例如：成功大學建築系的資訊建築實驗室、台灣大學資訊網路與多媒體研究所的遍佈運算實驗室、清華大學資訊應用研究所的平行暨分散式系統實驗室以及交大建築研究所等，顯見智慧空間的研究是資訊領域與建築領域相結合的發展方向。

1.2. 智慧空間的設計媒材

設計師們在設計空間時，若由於設計媒材的不足，便會受到很大的侷限而難以發揮創造力，如同建築師 Frank Gehry 在 1989 年設計自由形體時，也由於當時的設計媒材都無法滿足設計新建築的需求，他們便透過組合與整合的方式，利用航太用的設計媒材 Catia 結合自身研發的程式而創造了新的設計媒材(後稱為 Gehry Technology)，大大地提高了實現設計的可能性也造就了新的建築空間革命。同前一節所述，我們看到了智慧空間的浮現，然而現階段發展的設計媒材並未針對智慧空間而設計，致使智慧空間的設計過程在空間設計者的眼中充滿技術問題而裹足不前，在資訊設計者眼中也難以為整體空間進行設計，因此，智慧空間的浮現除了呈現出空間概念未來的發展趨勢、帶來更具互動性與智慧化的空間型態，同時也讓我們關注到一個新的問題，也就是設計師該用什麼樣的媒材來進行這種智慧空間的設計？智慧空間既然結合了建築空間與資訊系統，那什麼樣的設計媒材可以同時滿

足空間設計者與資訊設計者的需求？

1.3. 研究問題與目的

Mitchell 說明了這樣的趨勢與問題，他認為目前持續在發展的設計媒材仍是用來設計十九世紀的建築，這些新發展的媒材並沒有針對設計智慧空間的需求而設計，未來的設計媒材必須加以擴充，除了現有的幾何構造、材料與費用等要素外，還必須再加入智慧化元件的考量。未來的模擬工具除了渲染逼真的畫面或是能源的效能模擬之外，還必須能夠「執行」設計的建築物因應各種不同外在條件與使用者需求的情境下所作的變化，就像在動畫、遊戲或是機械設計工具中所看到的那樣 [Mitchell, 2006]。目前最先進的設計媒材雖然可以在設計中放入這些智慧元件，但都沒辦法反映出互動與變化的過程，而只是凍結在一個固定的狀態，更何況，這些設計好的建築物即使是在建造後仍會持續地與使用者互動而產生變化。現有的設計媒材沒辦法滿足設計者在設計智慧空間時的需求，從本研究的文獻探討中便可以看到目前各種結合資訊與建築的智慧空間的設計與建造，多是以跨領域的大型計畫的方式來進行以整合空間設計專業與資訊設計專業。如同 Jeng 利用他先前所提出的概念架構，透過快速成型機 (Rapid Prototyping, RP) 的操作與資訊化模組 Phidgets 相結合並完成智慧空間原型 (prototype) 的實作，並將此原型視為新的設計媒材以進行智慧空間的設計，但開發這個原型的複雜度與相關技術門檻仍相當高 [Jeng, 2007]。

再者，專業訓練的差異勢必造成溝通上的落差，不同背景的人員必須藉助著更大量的溝通以相互了解，但兩方所使用的媒材卻也有著極大的差異，空間設計人員利用空間設計媒材進行空間的規劃，資訊設計人員則使用資訊設計媒材進行程式的撰寫，Kvan 便提出，設計媒材不僅是用來設計的環境，更肩負著設計者之間用以溝通的橋樑，當雙方都不熟悉對方所使用的設計媒材時，溝通便僅能透過最基本的共通媒材，如：文字、圖面、動畫等 [Kvan, 2006]，這些媒材既然難以描述新的智慧空間的特性，溝通的落差是相當明顯的。

雖然傳統的設計媒材仍然可以用來設計新的空間，從過去到現在許多的設計與計畫仍在這樣的狀況下完成，但一方面，設計過程花費的人力與溝通的複雜度讓設計這

樣的新空間的成本與門檻居高不下；另一方面，在現有的媒材無法滿足特定的需求下，設計者的創作性與創意的發揮也會受到很大的限制，空間設計者在規劃空間時無法進行完整的空間與使用者互動過程的設計，資訊設計者在撰寫程式時也難以掌握空間的概念而只能在腦中建構訊息的流動。

本研究的目的，便是要嘗試將資訊設計與空間設計所需的數位媒材特性進行整合，將資訊設計過程所需的程式建構環境與空間設計過程所需的模型建構環境進行結合，提出一個用以設計智慧空間的電腦輔助設計系統 CADIS (A Computer Aided Design system for Interactive Space) 作為整合資訊設計與空間設計的雛型系統，讓空間的設計者不需要學習複雜的程式語法便可設計智慧空間中的資訊互動部分，資訊設計者也能透過此媒材得到空間的概念。

1.4. 研究方法與步驟

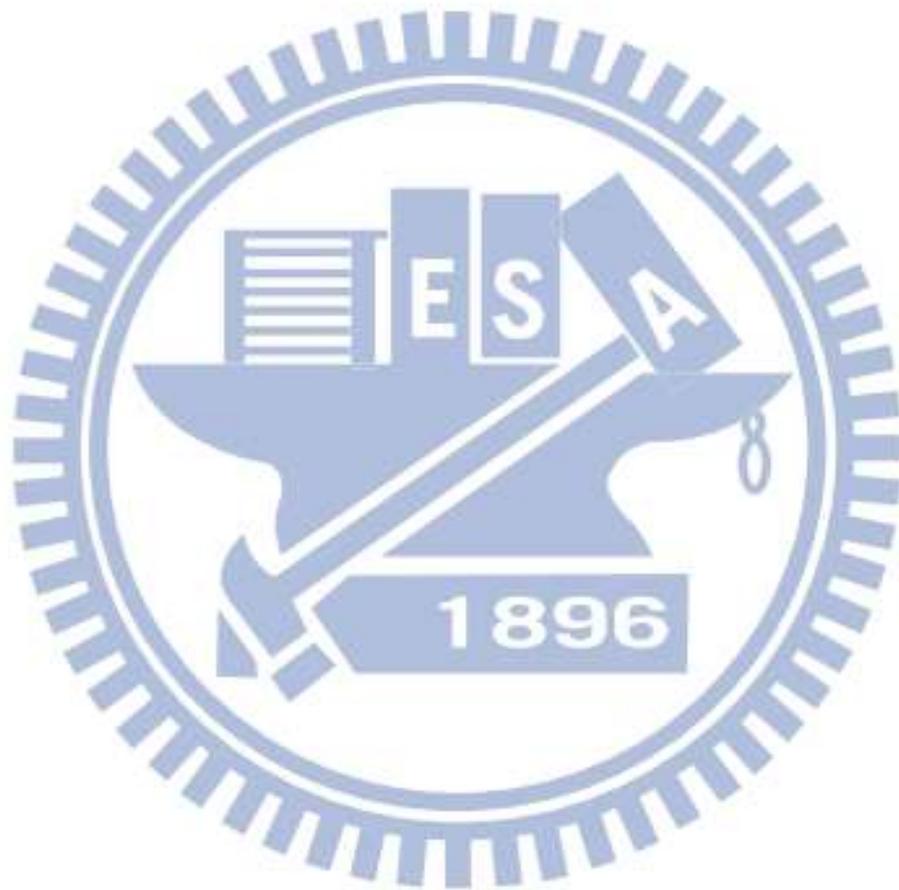
要進行智慧空間的設計需要兩種專業知識背景的設計者同時的參與，一方來自於建築或室內設計的空間設計師以及另一方來自資訊及介面設計的資訊設計者。然而我們也知道這兩方面的設計者，不僅所受的知識與訓練完全不同，使用的設計媒材也完全不同。要同時滿足雙方面的需求，CADIS 便必須是一個對於兩方面的設計者而言都是容易操作與學習的「非專業電腦輔助設計系統」。亦即，CADIS 對於空間設計師而言是一個容易學習與使用的資訊設計系統，對於資訊設計者而言則是一個簡單容易學習的空間設計系統。要達到這樣的目的，本研究必須透過案例的分析了解智慧空間的關鍵特色，以及對於兩種不同專業領域的設計媒材工具的分析以取得各自的特性，再將空間設計領域的設計媒材特性與資訊設計領域的設計媒材特性相結合，並提供兩種媒材中最關鍵的功能。

本研究透過下列方法與步驟完成 CADIS 的設計與實作：

1. 應用案例分析：透過對於數種智慧空間類型的分析，了解智慧空間在設計過程中需要具備的特徵，以了解如何讓 CADIS 提供功能支援。
2. 工具案例分析：透過對於現有的電腦輔助設計軟體以及智慧空間的設計過程的分析與歸納，了解 CADIS 該具備的設計原則與基本的功能需求。

3. 系統架構與實作：透過對各種新技術的了解與導入，配合系統分析的設計原則與功能需求，規劃出 CADIS 所需的系統架構與建構的方式以及 CADIS 各種功能的製作過程與紀錄。再以實際操作的方式，說明透過 CADIS 進行智慧空間的設計流程。

透過這些步驟，我們得以了解 CADIS 所需具備的功能特性、將其雛形系統實作出來、並用以實際進行案例的操作以說明 CADIS 將如何簡化智慧空間的設計流程。



2. 文獻回顧

本研究的文獻回顧由兩部份組成，一為透過「資訊系統空間化」與「建築空間資訊化」的趨勢來看未來的空間趨勢-智慧空間的浮現，觀察新空間的興起與其如何地影響人們的生活空間與型態，另一部份則分別探討用以設計建築空間的「空間設計媒材」與設計資訊系統的「資訊設計媒材」，藉以觀察目前設計者進行設計時所使用的設計媒材以及媒材的發展。

2.1. 智慧空間的浮現

資訊科技的大量運用，與人類之間的互動越來越頻繁，越來越多的資訊設備逐漸進入到人們的日常生活空間，本研究使用智慧空間一詞來描述這樣一個資訊設備嵌入並與人們互動的空間，相類似的概念還有「智慧環境」(Smart Environment)用來形容一個擁有許多智慧裝置的小世界，持續地讓居住者生活更舒適。[Cook & Das, 2004]；「環室智能」(Ambient Intelligent)闡述電子化的環境可以對人們與物件的存在以有感知的(sensitive)、具反應的(responsive)與可調適的(adaptive)的方式進行互動。[Ducatel et al., 2001]。

智慧空間將資訊科技與建築空間相結合，而這樣的新空間正不斷地出現在人們的生活周遭，從兩個角度可以看得出來智慧空間正逐漸在浮現：一是資訊系統的空間化，原本只存在電腦中的資訊系統，開始進入人們的真實空間與現實生活裡，人們日常生活所接觸到的物品與空間都可能隱藏著資訊系統以與人們互動；另一是建築空間的資訊化，建築空間從以往靜態的逐漸轉變為動態，這樣的動態改變除了在造型上，也反映在空間的機能上，而提供這些動態資訊或是互動行為的便是被嵌在建築各個元件裡的資訊設備。這兩個角度有著不同的切入點，但殊途同歸於空間與資訊的結合，再加上相關的文獻或計畫多以跨領域的方式進行整合，在進行討論時難免會有

相疊合而難以區分的狀況。本研究的文獻回顧以研究目標的尺度的大小作為分類的依據，從較單一而獨立的裝置的研究來看資訊系統的空間化，而從以整個空間或房間為主要思考脈絡的研究或設計案來探討建築空間的資訊化。

2.1.1. 資訊系統的空間化

自從 Weiser 提出的遍佈式運算 (Ubiquitous Computing, 也稱 Pervasive Computing) 概念，闡述二十一世紀的電腦將與傳統個人使用電腦的經驗大相逕庭，未來的運算裝置將不只是單一部電腦主機，而將以多種形式 (如：隨身音響、家庭娛樂中心、手機、視訊監控等等) 分散地存在我們的生活空間之中，並更滲透地參與人們的工作與生活 [Weiser, 1991]。這樣的想法在資訊領域引發了相當多的後續研究，Ishii 在麻省理工學院媒體實驗室所帶領的「實體位元」(Tangible Bits) 研究小組也進行著各種利用生活周遭的元素取代電腦中的圖像，讓人們透過實體空間中的物件與虛擬空間中的位元進行互動，而得以跳脫一成不變的滑鼠/鍵盤操作模式 (圖 2-1) [Ishii & Ullmer, 1997]。

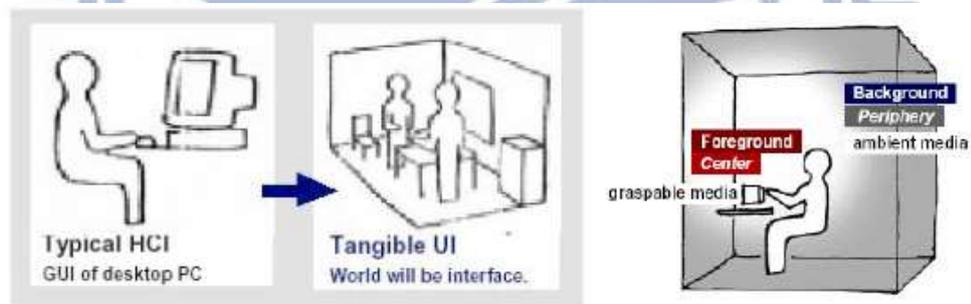


圖 2-1. 實體介面 (Tangible UI) [Ishii & Ullmer, 1997]

以下列舉幾個實體位元研究小組的研究：音樂板 (audiopad) 讓使用者操作實體的轉鈕控制電腦中播放的音樂 (圖 2.2a) [Patten et al., 2001]；沙堆 (sandscape) 讓使用者透過實體的沙，改變虛擬空間中的高程資料並影響虛擬水流方向與各種高程相關資訊的模擬 (圖 2-2b) [Ratti et al., 2002]；輸出入筆刷 (I/O Brush) 讓使用者透過攝影機取得真實空間的材質資料，並利用這些材質的影像在虛擬畫布上進行創作 (圖 2.2c) [Ryokai, Marti & Ishii, 2004]；Yoon 的繪畫 (egaku) 結合實體紙筆與虛擬數位投影，讓設計者在進行設計時能夠獲得大量數位資訊的輔助，增進創作的靈感 (圖 2-2d) [Yoon, 2004]。

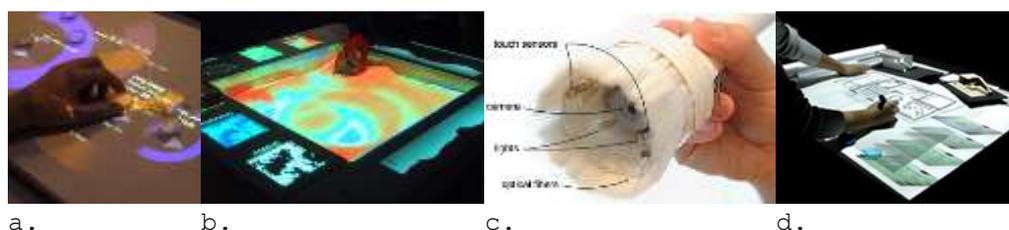


圖 2-2. 實體位元小組的研究

Rekimoto 的擴增平面 (augmented surface) 是一個多人互動的數位媒材環境，除了有共享的桌面，架在上方的掃描器也會自動掃描桌面上實體物件 (如：錄影帶) 的條碼在經過視覺辨識的運算後，將其相關的訊息投影在物體上 (圖 2-3)，除此之外，使用者也可以透過手指的點選或拖拉等動作彼此交換資料及檔案 [Rekimoto & Saitoh, 1999]。

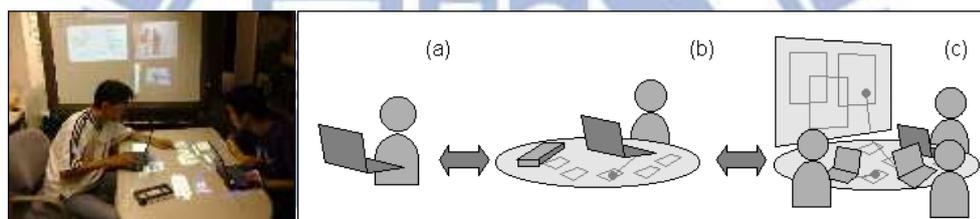


圖 2-3. Rekimoto 的擴增平面

加州大學柏克萊分校發展的整合便利貼 (post-it) 與投影系統的互動式牆面 (圖 2-4a)，實際應用在資訊建築設計上，讓多使用者同時參與互動，並結合視覺辨識的技術，由後端的電腦判讀後協助架構設計與每個設計單元的管理，每個不同的設計單元以一個便利貼呈現，電腦在判讀內容後便會依據其內容而在牆面上繪製連接線以表示各便利貼之間的相互關係並顯示在牆面上，而隨意移動便利貼後電腦也會追蹤並自動重新計算畫面 [Klemmer et al., 2001]。三菱電機研究實驗室的數位桌 (ubitable) 是一個不同使用者間透過一塊在實體環境中的互動式桌面彼此透過桌面交換數位資料，使用者各自操作自己的電腦，同時也能面對面溝通 (圖 2-4b)。系統在使用者各自的電腦上切割出共享區與私人區，使用者只要將電腦私人區裡的物件拖放置共享區，系統便會自動將物件透過投影的方式投射在桌面上以供分享，對方便可以從桌面上取得共享的物件 [Shen, Everitt & Ryall, 2003]。



圖 2-4. (a) 結合實體便利貼的互動式牆面；(b) 數位桌

與人類互動的電腦，也開始逐漸跳脫傳統電腦的形式，人機之間互動的介面的研究也從純軟體的圖形化介面 (Graphical User Interface)，開始朝向更結合真實空間的互動方式，例如前述實體位元研究小組的研究，還有同在 MIT 媒體實驗室的 Bonanni 發展的擴增實境廚房 (augmented reality kitchen) 讓人們直接使用已熟悉的廚房設備透過互動及感測裝置的嵌入提供更豐富的資訊以支援廚房中發生的活動 (圖 2-5) [Bonanni, 2005]。



圖 2-5. 擴增實境廚房

Park 在聰慧住宅 (smart home) 概念中，則將現在的生活空間中各種物件都嵌入資訊裝置後重新設計未來的家具樣貌，這些雖都是概念設計，但不難看出設計者對未來資訊融入家庭生活的企望與想像 (圖 2-6) [Park, 2004]。



圖 2-6. 聰慧住宅中提出的各種互動裝置

人與電腦如何互動(人機互動)的研究從以往偏向工程管理的方向,開始朝向如何更直覺容易使用及更貼近人類生活的方向發展,例如 Dertouzos 認為未來人與電腦將共生於一個大型的資訊市集中,人與人間、人與電腦間、電腦與電腦間彼此交換、買賣或共享資訊。接下來他更提倡人本運算的概念(Human-Centric Computing),包含五大重點:自然互動、自動化、個人化資訊存取、協同工作、個人化 [Dertouzos, 2001]。英代爾公司(Intel)的副總裁 Tennenhouse 也提出了「積極運算」(Proactive Computing)概念(圖 2-7a),認為電腦是該走入現實生活、克服實體問題、配合人們步調(get physical、get real、get out)的時候了 [Tennenhouse, 2000]。Jeng 提出的人本互動模式,以認知空間、實質空間與虛擬空間來區分互動事件發生的狀態,如圖 2-7b 所示。在認知空間中,使用者接受感知之後,發生動作。在實質空間中,人透過控制介面操控物質性介面(physical representation)進而遙控數位化界面(digital representation),讓使用者感知空間中數位資訊所帶來的空間狀態的改變。在虛擬空間中,物質性介面連結至互動事件,藉由互動事件來觸發虛擬化互動模式的進行,呈現於空間中的數位資訊為具體化的互動模式。從使用者經由控制實質化介面觸動互動事件,進而可直接操控數位資訊,產生實質與虛擬相關聯的人本互動模式 [Jeng, 2002]。

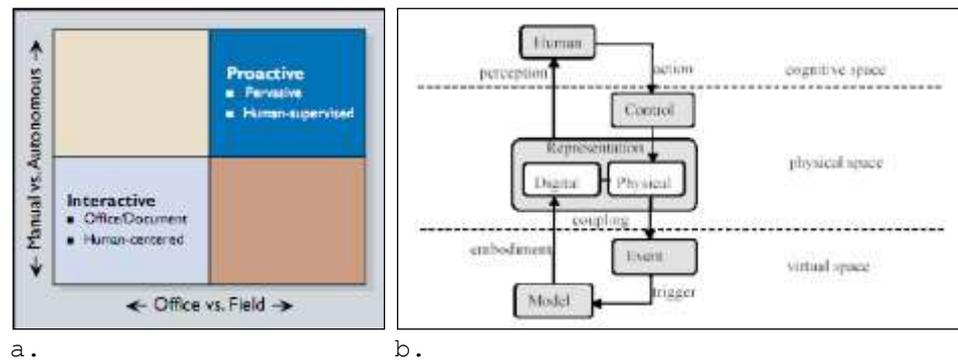


圖 2-7. (a)積極運算；(b)人本互動模式

雖然將資訊科技嵌進實體空間成為人機互動研究領域的一種趨勢，然而多數的研究仍從介面的角度專注於人與個別的互動裝置之間的互動方式，而較少從人類「活動」的角度來探討人類在進行特定活動時所需要的互動形式。Norman 提出了以人為本的互動設計 (User-Centered Design) 後，接著探討了人類日常生活常會碰到的介面問題 [Norman, 2000]，之後更提出了情感設計 (emotional design) [Norman, 2004]，近年 Norman 則提出了新的論點，認為以人為本的設計可能有害，互動的設計應以觀察人類活動並提供必要的資訊與協助，以活動為本的設計 (activity-centered design)。由此可知，為人類生活而設計的互動設備所組成的智慧空間，應從人類日常生活的「活動」為角度，透過遍佈式運算或環境感知 (context awareness) 的技術來協助使用者各項活動需求。

2.1.2. 建築空間的資訊化

隨著工業與科技的發展，建築空間的觀念也不斷地在演進。最大的改變除了外型由原本只能有直線進化而得以運用曲線，甚至是自由形體 (freeform) 的建築，另一個引發興趣的就是對於傳統建築空間是「不可變」的觀念的挑戰。十九世紀後期就有建築師們開始嘗試「可變的」(mutable) 建築，起初是一種模組化 (modular) 的概念，讓建築物利用可重複使用且規格統一的材料建構而可以隨著需求改變，如：Joseph Paxton 的水晶宮 (Crystal Palace) 便是以鋼鐵為骨架、玻璃為主要建材的建築實現了強韌、耐久、形式又簡單的建築物和伴隨著的快速的建造進度。1930 年代的 Fuller 提出了工業化的建築設計概念讓建築物具有彈性與重新設定的可能。1961 年，建築師 Cedric Price 與 Gordon Pask 在他們的合作設計

案 Fun Place 中運用了可活動的牆面、天花板與地板，噴霧設備與熱氣流，讓空間的環境能夠直接回應人們的實際需求 [Mathews, 2006]。建築師 Warren Brodey 在 1967 年提出了軟建築 (soft architecture) 的概念，有別於傳統由設計師規劃空間，軟建築的空間是由空間的居民透過運算裝置進行掌控 [Brodey, 1967]。電腦的普及與效能上的進展則讓原本需要大量運算的人工智慧與數據統計進入建築空間中參與人們的生活。將用來感應環境中溫度、溼度及聲音等等各種感應器 (sensors) 的資訊集中整合並加以管理，進而得以提供更為舒適的智慧型空間，如麻省理工學院建築所的 House_n 計畫、Huang 提出的 swisshouse [Huang, 2004]、Chiu 提出的 House Sense [Chiu, 2006] 皆屬於這類經由電腦控制空間進而改善居住者的生活。

瑞士聯邦理工學院的數位網路與媒體建築計畫 (Building IP) 除了利用各種感測器去感應溫度或溼度等等生活空間的資訊之外，還利用了「時間」的統計並分析生活在其間的人們在不同的時間的活動需求，在必要的「時候」喚醒特定的裝置運作提供協助，例如：晚上接近睡覺時間前自動預熱洗澡水以配合主人通常在睡前洗澡的習慣 [Schoch, 2006]。其下的另一個計劃「紅色地獄」(Red Hell) 則更進一步地將自動化建築的資訊與結合到空間中，讓空間除了回應使用者的操作設定改變溫度光線等環境條件，還可以依據不同的情境產生不同的方式與使用者互動 (圖 2-8) [Strehlke, Ochsendorf & Bahr, 2004]。



圖 2-8. Red Hell 結合自動化建築與媒體化建築

科裡意指整合軟體與硬體資源，並作為兩者之間橋樑的一種軟體管理系統。Arch-OS 也用了相同的概念，主要的組成構件有能源管理系統、通訊網路、空間中居民的移動與社交活動感應、背景噪音感應與環境狀態感應等，透過這些構件除了可以即時地回應居住者的需求之外，也可以將空間中的資訊以視覺化的方式呈現出來，如圖所示，Arch-OS 將居民在建築空間中的空間狀態視覺化地表達出來 (圖 2-10) [Anders, 2004]。

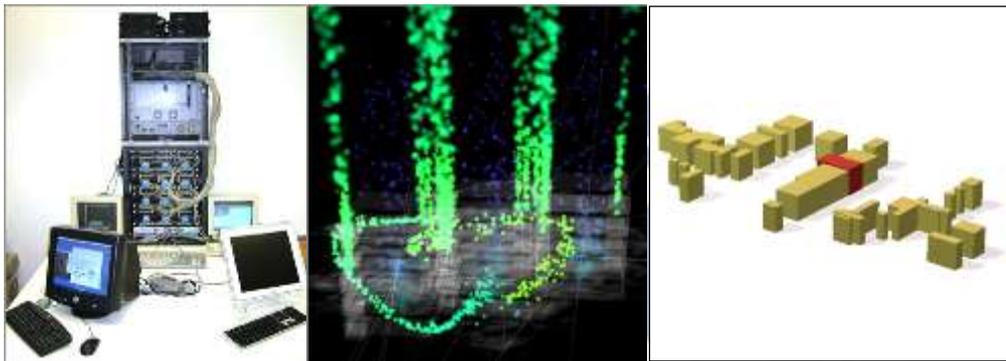


圖 2-10. Arch-OS 將居民在建築空間中的空間狀態視覺化地表達出來

Streitz 的軟房間 (Roomware) 將整個房間視為一個大型的電腦應用軟體，所有的家具包括牆面 (DynaWall)、桌子 (ConnectTable/InteracTable)、椅子 (CommChair) 其實都是一部部的電腦 (或軟體)，使用者在使用空間的同時也便是在與這些軟體互動 (圖 2-11) [Streitz, Geibler & Holmer 1998; Streitz et al., 1999; Streitz et al., 2002]。



圖 2-11. 軟房間 (Roomware)

日本 SGI 株式會社所提出的「空間機器人」(RoomRender) 是將整個房間視為一個機器人，房間中佈滿的各種感應器與控制器則為機器人的零件，運用機器人控制系統的概念參與人與空間的互動關係(圖 2-12)。



圖 2-12. RoomRender 的基本架構

RoomRender 包括三個特點，分別是若無其事的空間演出、友善的使用者介面以及聲音主控系統。若無其事的空間演出是透過感性制御技術(ST - Sensibility Technology)，將空間中人們的談話聲音語調內容加以數據化分析後判斷出氣氛為喜、怒、哀、無、笑、興奮等，再依照目的(炒熱氣氛或是讓大家冷靜等)改變牆面(FellingWall)的顏色以及氣味，例如：當 RoomRender 從「啊～好累啊」這

句話察覺出負的興奮值，便會自動將空間的燈光調暗，並啟動音樂與香精。友善的使用者介面則是提出了聲音、觸控面板與遙控器作為空間控制的介面。聲音主控系統則是將所有的控制系統整合在聲音辨識下，利用談話內容作為空間活動類別的判斷，進而與其他控制裝置連動 [日本 SGI 株式会社, 2005]。

除了環境與空間的管理，在現代的建築空間中，數位科技的運用也逐漸成為整個空間設計的其中一環。許許多多的設計概念被提出並被實現在真實的建築案例中，不論是媒體立面(media facade)、互動牆(interactive wall)、或是建築師 Rem Koolhaas 在紐約的 Prada 旗艦店也大量地運用了各種數位科技，除了利用光變玻璃作為更衣間取代傳統布簾的廉價感，還有他在期間所放置的與其他 Prada 分店連線的即時連線攝影畫面，藉此嘲諷精品店與量販店之間的差異性在哪(圖 2-13)，這些資訊系統不僅是在建築空間中的裝置，更是傳達他的設計意念的元素之一 [Bertelli et al., 2001]。



圖 2-13. Prada 在紐約旗艦店中運用了大量的資訊科技設備

ag4 mediatacture GmbH 設計工作室提出的 mediatecture [Daab, 2006]，讓人們直接與空間的媒體立面互動，透過使用者的參與改變建築的外觀表現(圖 2-14a)。ART+COM 設計工作室也將媒體立面的概念落實在一些設計案例中(圖 2-14b) [ART+COM, 2004]。



圖 2-14. (a) ag4 的 mediactecture ; (b) ART+COM 的 Vattenfall media facade

媒體立面已大量地運用在各地的設計案中：UN Studio 在首爾的拱廊百貨公司透過附屬結構上的大量經過特殊彩虹包覆處理的玻璃片白天可在外觀上產生連續的色彩變化，夜晚則成為大型的行銷廣告看板(圖 2-15a) [UNStudio, 2006]；Peter Marino 在東京的香奈兒總部 (Chanel headquarters) 在白天的玻璃立面是完全透明的而在夜晚則會發出光亮(圖 2-15b)，圖騰的設計是交由藝術總監透過電腦控制程式的編寫加以控制 [Kim, 2003]。



圖 2-15. (a) UN Studio 設計的拱廊百貨公司；(b) Peter Marino 的東京香奈兒總部

柏林的真實：聯合設計工作室 (reality:united GmbH) 在奧地利的格拉茲城 (Graz, Austria) 進行的 Kunsthaus Graz 案，仿生造型的建築外殼包覆著藍色半透明壓克力玻璃，在表層下安裝有一種稱為 BIX 的發光裝置，並能藉由特製的編輯軟體 (BIX Director) 控制來讓整個建築物的外皮變成超大的低解析度顯示幕，並可透過模擬器 (BIX Simulator) 來模擬建築在整個都市空間中的尺度以

及不同的顯示內容帶來的不同的空間經驗。透過這樣的顯示，將建築內的資訊透明似地帶到整個都市中，除了在外觀上呈現動態的皮層，也成為建築空間與都市人群溝通的工具(圖 2-16) [Lubell, 2003]。

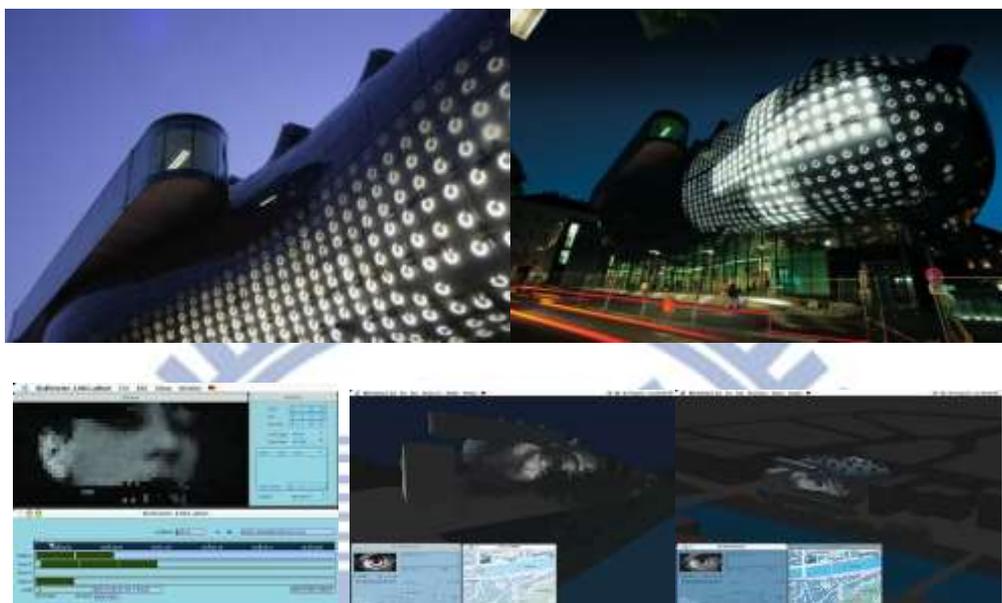


圖 2-16. Kunsthaus Graz 與 BIX Director, BIX Simulator

ART+COM 設計工作室的展覽空間牆(Wall Showroom)將大型互動牆與室內空間的設計相結合，透過電腦移動互動牆面改變空間配置以及顯示的內容，資訊設備除了在空間中顯示資訊，也成了構成空間結構的一部分(圖 2-17)。



圖 2-17. ART+COM 的 Wall Showroom

人們對於資訊系統的需求，以及數位科技滲入生活之深，在在地反映在整個空間裡，也可以看得到未來資訊與建築空間相結合的趨勢。

2.2. 設計媒材

不論是建築空間的設計或是互動資訊的設計，在設計過程都需要設計媒材(或設計系統、設計工具)將腦中構思的結果成型，以進行自我的設計檢視(design

evaluation) 或是與他人間的設計溝通 (design communication)。

由於專業領域與知識背景的區別，在傳統上，設計空間時設計者會藉助空間設計媒材，而設計互動資訊系統時則是透過資訊設計媒材，這兩種媒材的發展有各自的發展與類型。

2.2.1. 空間設計媒材

非電腦時代

從埃及時代開始，最需要設計空間的建築師便開始使用圖畫作為設計檢視與溝通的媒材，主要的設計工具便是紙筆。雖然時代的演進讓新的技法，如：透視圖法加入了設計媒材之中，但最主要的設計工具始終還是運用紙、筆與尺規來繪製出各種的草圖、平面、立面、剖面與透視圖。文藝復興早期的布魯內勒斯基首度運用模型來思考，彌補了圖畫的不足 [Liu, 1996]。

非數位時代所使用的紙筆尺規與模型，沿用了數千年。然而建築師就像藝術家一樣，在設計與創作時經常會有無限的想像，但受到設計媒材的限制，畫不出想像中的形體與空間，另一方面也受到材料與營造技術的限制，即使畫得出來也不見得建造得起來。因此，建築師一直在營造技術與設計媒材的限制下做設計，建築也就隨著營造技術與設計媒材的突破而產生時代性的變化。由埃及、古希臘羅馬、中世紀、文藝復興、直到工業時代，建築工藝與營造技術在歷史上的關鍵性發展，總會帶動建築形體與空間的一再解放 (跨距更寬、高度更高、量體更輕、曲度更大等等)，在在表現出空間的設計者冀望跳脫工具限制的強烈企圖。1989 年蓋瑞在「巴塞隆那魚」利用電腦解放建築設計與營建技術的束縛，設計工具進入了大量運用電腦的數位時代 [劉育東, 2007]。

電腦輔助設計時代

在現代的空間設計中，電腦的大量運用已成為最重要的設計工具。例如歐特克公司 (Autodesk) 於 80 年代初期開發的電腦軟體-AutoCAD，讓設計者透過指令與滑鼠的操控，簡單地構建出原本用紙筆進行的各種平面圖面，但卻能較紙筆尺規更快速而精確地進行曲線的創建與各項量測 (圖 2-18)。

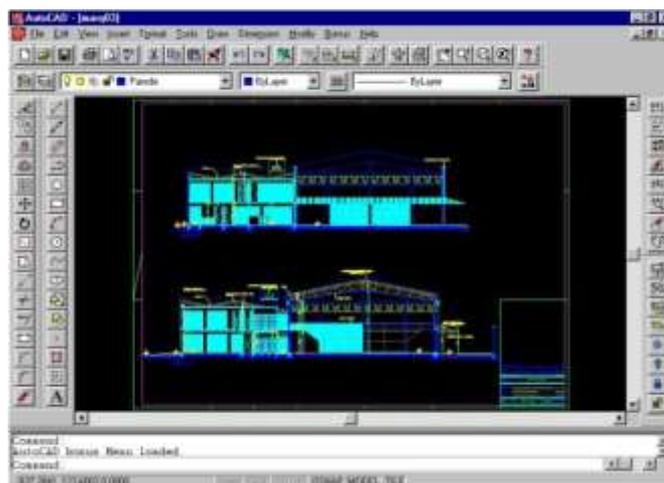


圖 2-18. AutoCAD R14 軟體

科技的進步，讓電腦的運算處理速度如日千里，軟體的能力也從二度空間的平面圖進入了立體的三度空間，設計者開始可以運用三度空間軸的概念在軟體中創建虛擬的立體模型，除了不再受到媒材限制必須在腦中將空間切割為三個視角之外，也加入了材質的計算讓虛擬模型可以被正確地著色與顯像，再透過彩現 (rendering，也稱算圖) 將光與影的效果加入而能得到更接近真實視覺的模擬圖像 (simulation)，讓圖面與實體空間之間的差距縮得更小，這類工具在空間設計上最常用的如 Autodesk 的 3D Studio MAX (圖 2-19a)，Maya (圖 2-19b)。

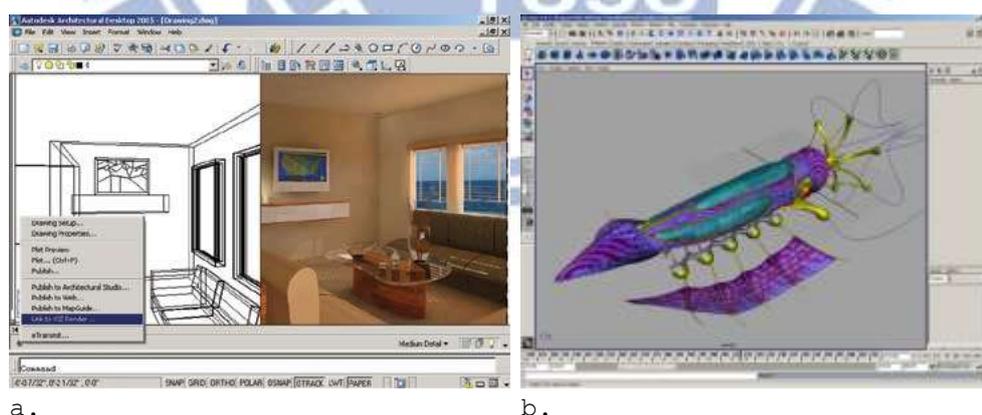


圖 2-19. (a)Autodesk 3D Studio MAX 軟體；(b)Autodesk Maya 軟體

除了形的建立與圖面的繪製，空間的設計過程還有其他的設計問題也都開始透過電腦的運用來解決，例如結構、照度分佈與管線配置的計算，透過軟體可模擬並精確

地計算出各種可能的實踐方案，讓設計構思到空間的實踐沒有落差，常用的設計工具如賓特利公司 (Bentley) 的微平台 (Microstation) (圖 2-20)。

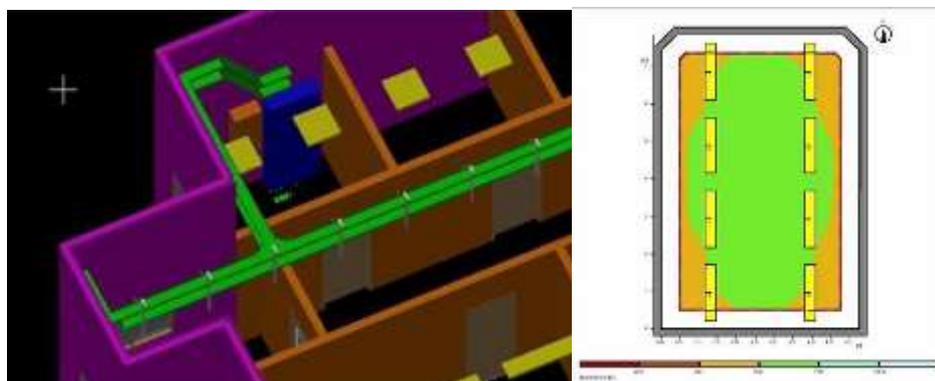


圖 2-20. Microstation 軟體

上述的數位媒材多是通用型的，也就是任何需要立體造型的設計都可以運用這些軟體媒材來輔助進行，但也因為功能上必須滿足多樣的需求，這類的軟體媒材的功能項目相當龐雜，而導致難以學習，也因此有了針對空間設計 (尤其是室內設計與建築設計) 的需求而特化的數位媒材，例如：3D Studio VIZ 便是將 3D Studio MAX 針對建築與室內設計需求而增減其功能後的版本，讓設計者可以簡單地利用一系列的元件庫 (如：窗、樓梯、牆等) 快速地設計空間 (圖 2-21a)；Abvent 公司的 Atlantis 也是有著快速的元件建立流程，而強化即時 (realtime) 瀏覽的模擬效果 (圖 2-21b)；Google 的 SketchUp (圖 2-21c) 同樣也是針對快速建立模型，讓設計者直覺地對建立的可視量體進行切割、延伸與黏合並具有真實尺度的概念，而不再是與大量的頂點、三角面、多邊形 (polygon) 等來自電腦圖學的術語奮戰，由於使用上的簡易讓學習的門檻降低，除了專業的設計者用來快速建立設計之外，也有許多非專業的使用者在使用。

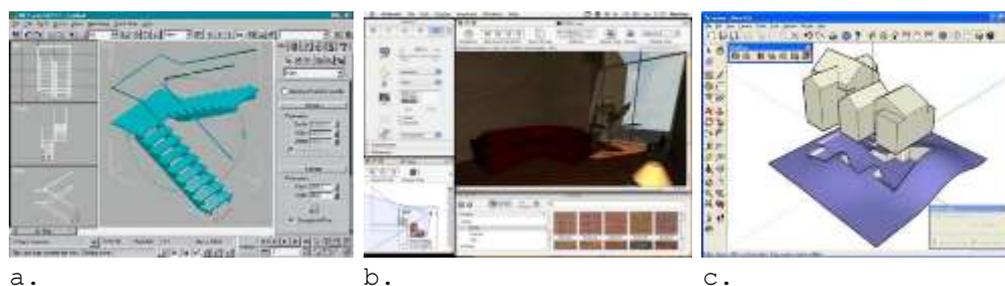


圖 2-21. (a) 3D Studio VIZ ; (b) Atlantis ; c. SketchUp

這些數位媒材的運用，固然提供了越來越完備的空間設計資訊，但都是以電腦軟體的形式存在，受限於電腦的軟硬體形式，這些工具的輸入都是經由滑鼠及鍵盤，給設計者的回應也限於僅是螢幕上的平面顯示，仍難以完整傳達設計的概念。因此，許多的研究開始進行以多模式的互動方式作為設計的工具與環境。能快速產生實體模型的電腦輔助設計製造 (CAD/CAM, computer-aided design/computer-aided manufacture) 與快速產生立體視覺的虛擬實境 (VR, virtual reality) 也在時代的發展下成為新的設計媒材。

CAD/CAM 由於能透過具備電腦數值控制 (computer numeric control, CNC) 的機具，如：雷射切割機 (laser cutter) 與快速成型機等能夠快速地產生實體的模型，成為設計師在設計階段便會加以運用的設計媒材，並進而影響了設計結果 [Lim et al., 2006]。

虛擬實境的设计媒材由於電腦顯像軟硬體的進步，已可以即時地模擬出極為逼真的視覺效果，而不需要如同以往般需耗時動輒數天。越來越多的研究便著重在運用 VR 的设计媒材 [Peng, 2005; Palmon et al., 2005; Kim & Choi, 2006; Maher et al., 2006; Tomohiro et al., 2006]。

互動設計時代

Lee 提出了以感官模式的人機互動方式，動作於傳統與數位的设计媒材 (圖說、模型等實體物件及多媒體形式之數位呈現)，得以自然而直接地進行设计說明的互動架構，稱為「多模式互動架構」。多模式互動架構以整合设计者的感官模式、動作控制行為、设计媒材以及數位資訊科技，說明在數位设计環境中，以设计者中心，與设计媒材發生各種互動行為之设计流程 (圖 2-22)。设计者可藉由手勢動作、指

揮介面進行數位資訊的控制。以直接的動作「切」出數位模型的剖面圖。以手勢控制數位模型的呈現方式，使得操控數位模型如同控制實體模型的動作。使用雷射指示器可程式化其光點位置並設定其互動內容，設計者可在光筆指示的同時，加上口頭與動態圖像的說明 [Lee, 2002]。

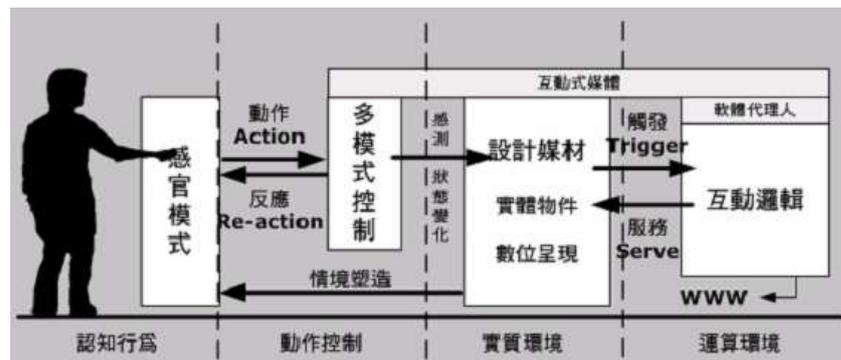


圖 2-22. 多模式互動架構

Maher 也在研究指出，使用實體介面 (tangible user interface, TUI) 進行的新互動設計媒材，相對於桌上介面 tabletop (由於傳統的電腦輔助設計媒材多是以電腦軟體的形式安裝在桌上的電腦裡，Maher 將其這類傳統的電腦輔助設計媒材稱為 tabletop)，可讓設計者更易於掌握立體物件與各物件間的空間關係，進而發掘出更多隱藏的設計線索，而有助於提昇設計的創意 [Maher, 2006]。

空間設計媒材雖然有所發展，各種新的空間設計媒材不斷出現，但多是著重在設計靜態的建築空間，空間設計媒材在 21 世紀雖然持續發展，卻仍是用在設計 19 世紀的建築，設計出來的建築物仍然是如同以往地巨大、笨重、有著骨架與皮層再加上相對簡單的機械與電力系統。但在未來，建築將會因結合複雜的感測器、處理器與通訊裝置而有著迥異的面貌。建築師將勢必開始思考建築物與這些複雜元件之間的關係 [Mitchell, 2006]。Jeng 也同樣認為具備資訊運算能力的遍佈運算媒材 (ubiquitous media) 將環繞在空間中與人們自然地互動，透過稱之為 IP++ 的計畫進行實際的設計 [Lin, Shen & Jeng, 2005]，並提出概念架構以描述這樣的新互動空間 (interactive space) 所帶來的設計議題 [Jeng, 2005]。

2.2.2. 資訊設計媒材

由人機互動介面的觀點，互動資訊系統中的機器端包括了提供人類輸入操作的硬體、

根據使用者的輸入與存在資訊系統內的知識進行運算的軟體、及輸出各種感官訊息(影像、聲音等)以回饋給使用者的硬體。因此，一個互動資訊系統涵蓋了硬體的裝置與運算處理的軟體，而由於硬體與軟體設計方式的差異，互動設計媒材大致可以分為設計硬體用的電子設計自動化(electronic design automation, EDA)與軟體的程式設計環境。

電子設計自動化

互動硬體裝置是各種電子元件以及電路組合在一塊或多塊的電路板上，電子元件經由電路的連接、電路間傳遞的訊號及元件各自的處理邏輯，共同組織工作以完成特定的功能需求。製作過程可分為三個階段，設計者先依功能與需求進行架構(schematic)設計、之後依據架構設計以及實體的尺寸限制製作出電路圖、最後再依電路圖製作成電路板並焊接上各種元件(圖 2-23)。設計者在設計階段時常會借助電子設計自動化軟體以方便地完成設計，電路板的製作階段則有兩種方式，人工進行電線連接的電路麵包板(wire wrap)方式或是透過電子印刷術將線路印刷在印刷電路板(printed circuit board, PCB)上，較小型或單純的電子裝置會透過電路麵包板來製作，裝置需要大量生產或是太過複雜會用印刷電路板方式。

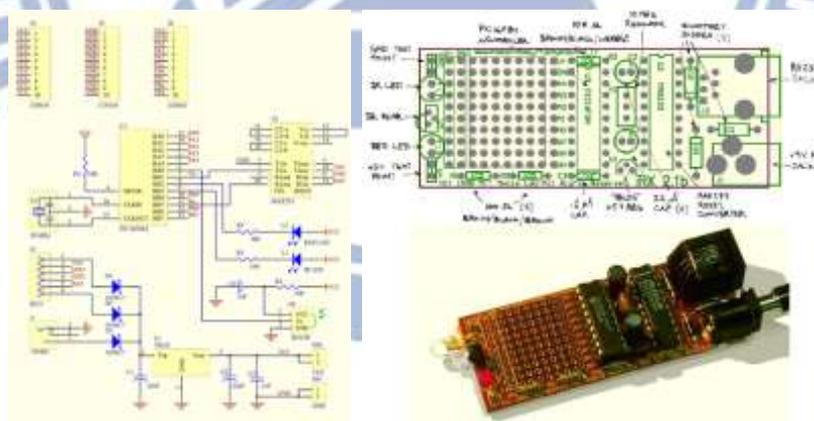


圖 2-23. 以 MIT 的 iRX 為例：(a) 架構設計圖；(b) 電路圖；(c) 電路板

電子設計自動化軟體，也稱電子電腦輔助設計(electronic computer-aided design, ECAD)，讓設計者在進行電路設計時運用各種類比或數位電子元件以及佈線進行設計，設計者在上千種的電子元件中挑選，每種元件有著各自不同單一或是複合的訊號處理功能，設計者也必須設計線路的連接以傳遞訊號，透過這些組合

來達到解決問題的目的。由於需要大量處理訊號與電子元件，電子設計自動化的系統必須能滿足幾個階段的功能需求，包括：用以進行架構設計的設計導入 (design entry)、模擬訊號的模擬器 (simulator)、分析訊號的分析器 (analysis)、計算各種訊號相互干涉的合成器 (synthesis)、模擬各種電子元件動作行為的做真器 (emulator)、將線路分佈在準備製作成电路板的範圍內 (layout)、處理線路交叉及訊號耗損的繞線 (routing) 等等。常見的 EDA 有 Cadence 公司的 OrCAD (圖 2-24a)、Xilinx 公司的 ISE (圖 2-24b) 與 Altium 公司的 Designer (圖 2-24c)。

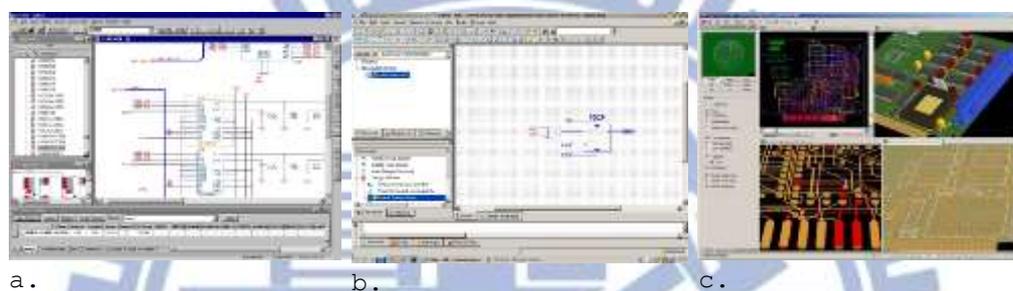


圖 2-24. (a)OrCAD ; (b)ISE ; (c)Designer

然而，單純使用電子零件 (電阻、電容及各式匝道器...等) 製作的電子裝置製作好了就難以更動，一旦發現問題或是要加入新功能就得重頭開始，架構上缺乏彈性。因此，許多的互動硬體裝置便會將處理訊號的部份交由電腦中的軟體來處理，如此便可以在不更動硬體的狀態下，只要改寫軟體就可改變裝置與人互動的機制。而個人電腦有著體積較大難以隱藏、耗電量大及價格高等缺點，有些互動裝置便會改用小型、省電、便宜及功能較為單純的微型電腦，稱為微控制器 (microcontroller) 或微處理器 (microprocessor 或 micro control unit, MCU)，這類微型電腦大致可分為：使用低階語言的 8051 系列、使用中階語言的 PIC 系列以及使用高階直譯式語言的 Basic Stamp 系列 [O'Sullivan & Igoe, 2004]。而不論個人電腦或是微型電腦，要在其中進行的設計都是透過程式語言的撰寫，在軟體的程式設計環境下達成。

軟體的程式設計環境，主要的媒材形式便是程式語言，隨著演算法與程式語言概念的演進，大致可以歸類為循序式程式設計 (procedural programming)、物件導

向式程式設計 (object-oriented programming) 、以及視覺化程式設計 (visual programming) [Preece et al., 1994; Sharp, Rogers & Preece, 2007] 。

循序式程式設計

循序式程式設計也稱第三代語言 (third-generation language, 3GL) ，主要是以各種指令讓電腦循序地執行並進而完成各種功能，由於電腦本身的硬體設計便是一行行地執行指令，這類的程式語言是最接近電腦的邏輯與效率，設計者在設計時通常會將各種的功能包裝分類而為群組般的指令集合，稱為函式 (function) ，再將各種指令以文字的形式循序地輸入電腦，最常用的有組合語言 (assembly) 、C 語言與培基 (BASIC) 語言。

組合語言是賦予設計者直接操作各種處理指令與記憶體中的資料，例如：“mov a, b”指示電腦將記憶體空間 a 中的資料搬到記憶體空間 b、“add a, b, c”指示電腦計算記憶體空間 a 中的資料與記憶體空間 b 中的資料相加後放到記憶體空間 c 中 [Carter, 2002]。由於組合語言所操作的都是電腦中的記憶體所放置的資料以及位置，對設計者而言是相當難以理解的，設計者必須一一比對資料在記憶體與處理器間搬動的指令過程才看得出來資訊的處理程序。為了更貼近人類思考的模式，具備符號表示式 (symbolic) 的語言形式，讓設計者能夠運用各種代數或變數符號代表各種資料，以更直覺地運用指令進行各種資料的處理 [Kernighan & Ritchie, 1988]。而 BASIC 語言則是更加接近人類英語語法的語言 (圖 2-25)。

<pre>dec ecx jnz NextPixels jmp LineDone DrawPixels: movq mm0,[esi] movq [edi],mm0 add esi,8 add edi,8 add ebx,4 dec ecx jnz NextPixels</pre>	<pre>FILE *infp; int ch; char str[255]; infp = fopen("test.htm","r"); while (feof(infp) == 0) { ch = getc(infp); printf("%c%i",ch,ch); } printf("Plz input a string"); gets(str); printf("%s",&str);</pre>	<pre>Sub setPquality() If mode = 6 Then Backto main ElseIf mode < 5 Then oldmode = mode set_active 6 Else Backto main set_active 6 End If End Sub</pre>
a.	b.	c.

圖 2-25. 幾種循序式語言形式：(a) 組合語言；(b) C 語言；(c) BASIC 語言

除了在個人電腦上之外，微處理器中的 8051 系列可使用組合語言或是 C 語言進行軟體的撰寫、PIC 系列可使用組合語言、PIC C (C 語言的一種) 以及 PIC BASIC (BASIC 語言的一種)、BASIC Stamp 系列則是使用 PBASIC (BASIC 語言的一種) [O'Sullivan & Igoe, 2004]。

物件導向式程式設計

由於循序式語言較為接近電腦的邏輯而非人類日常處理事務的經驗，於是以物件為概念的設計系統便被提出了。物件導向式語言是透過將整個設計的解決方案規劃為一個一個的物件，每種物件具有特定的屬性與方法，透過物件之間的互動構成一個彈性化的系統。Java 語言是最常被提到的物件導向式語言 [Eckel, 2006]。

隨著遍佈運算概念的廣為接受以及數位藝術的大量興起，越來越多非工程背景的設計領域的設計者(如：視覺設計、工業設計、產品設計等)投入互動設計的領域，讓互動資訊系統的設計媒材不再如同以往只有工程師在使用，也因此，互動系統設計媒材開始必須重視如何降低設計開發知識門檻以廣納更多不同知識背景的設計者。也因此許多研究單位進行並發明基於 Java 語言的簡易型語言，例如：麻省理工學院發展的 Processing 以及後續衍化出來的 Wiring [Hernando, 2004] 以及 DBN (design by numbers) [Maeda, 2001] 都是針對非工程背景設計者所設計的簡易型語言(圖 2-26) [Greenberg, 2007]。

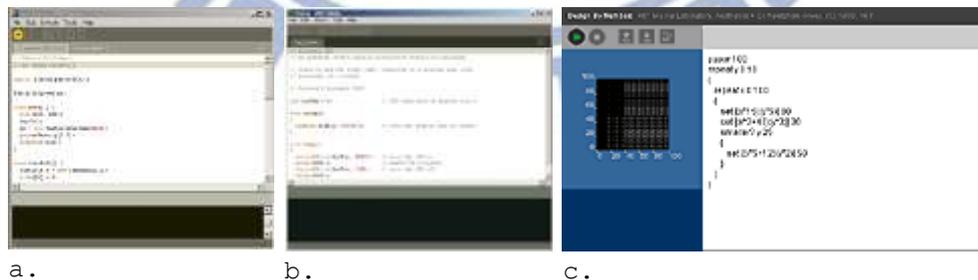


圖 2-26. 基於 Java 的簡易型語言：(a) Processing；(b) Wiring；(c) DBN

物件導向式的程式設計雖然在設計者的心智空間中已經較為直覺而容易理解，然而與循序式語言相仿仍是透過文字輸入的方式進行設計，在設計過程仍然缺乏直觀的視覺的回饋。也因此許多的程式設計系統開始走向視覺化，進而發展出了視覺化的

程式設計環境。

奧多比公司(Adobe)的 Flash 及 Director 在互動設計上是相當常見的設計環境，讓設計者可以直接看到設計的畫面與動作的結果，也可透過程式的撰寫讓畫面裡的視覺元件與使用者的輸入(滑鼠、鍵盤)產生互動(圖 2-27)，Flash 所使用的語言稱為 AS(action script)，Director 所使用的語言稱為 Logo，在許多展覽中都可以看到其運用，所有的程式都與畫面裡的物件結合在一起，是相當視覺化的物件導向程式設計過程。



圖 2-27. (a) Adobe Flash; (b) Macromedia Director

不論是物件化抑或是視覺化，對於設計者而言，都需要學習一個程式語言的語法(syntax)，程式的邏輯與運作的資料流(data flow)也非可視的，要能順利撰寫程式仍需要一段陣痛期。於是漸漸發展出視覺化程式設計(visual programming language, VPL)，能利用淺顯易懂的概念圖繪製或是設定數值等等方式來取代程式碼的鍵入，這讓許多非資訊工程背景的使用者能更易於接受，除了不再需要記得語法與語言結構之外，所有程式的邏輯都透過了可視的視覺區塊來呈現，程式運作時的資料流動狀態也可透過流程圖表而得以理解。

視覺化程式設計

視覺化程式設計讓設計者可以透過視覺化的過程，如：數值的設定、流程圖表(diagram)的繪製、區塊(block)的組合等等方式來進行程式的設計。設計者將圖表繪製完成或區塊組合好後，便可自動產生能讓電腦執行的程式檔案，整個設計過程中幾乎不需要以文字方式輸入一行行的指令，讓沒有修習過程式設計課程的人也可以很輕易地學會製作互動系統。

Max/MSP 是一套運用圖形化介面處理連接至電腦的電子裝置所發送的訊號的環境，原先的發展用於處理音樂設備數位介面 (musical instrument digital interface, MIDI) 樂器的彈奏與演出，但逐漸發展為可處理各種輸入電腦的訊號的工具 (圖 2-28a)。國家儀器公司 (National Instrument) 的 LabView 則可將電腦外部的硬體訊號接入實體的輸出入介面卡後，透過圖表設計的程式進行處理，再依據設計者的設計進行各種訊號的輸出，如：影像訊號、聲音訊號等 (圖 2-28b)。

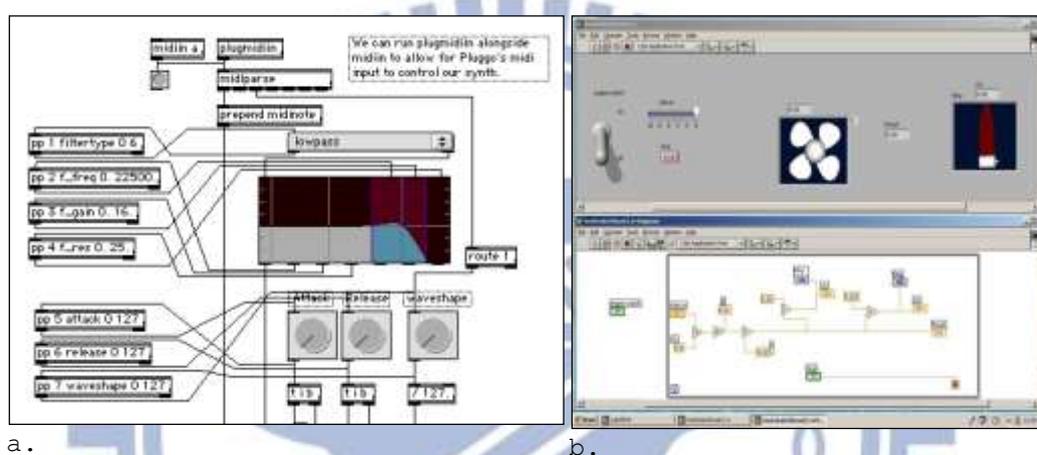


圖 2-28. (a)Max/MSP ; (b) LabView

BlueJ 是一種視覺化的程式設計環境，不同於一般使用文字輸入的方式進行程式的設計，BlueJ 運用各種的圖表來描述設計完成的系統中的各種物件，包括使用者，運算機制、事件等等，各個物件相互串連並彼此呼叫，同樣以物件導向的設計概念共同完成並解決設計問題 (圖 2-29a)。樂高公司 (LEGO) 的新腦力激盪 (MINDSTORMS NXT) 則是透過各種物件的編排與串接，控制連結在微電腦上的樂高積木或是馬達等裝置 (圖 2-29b)。

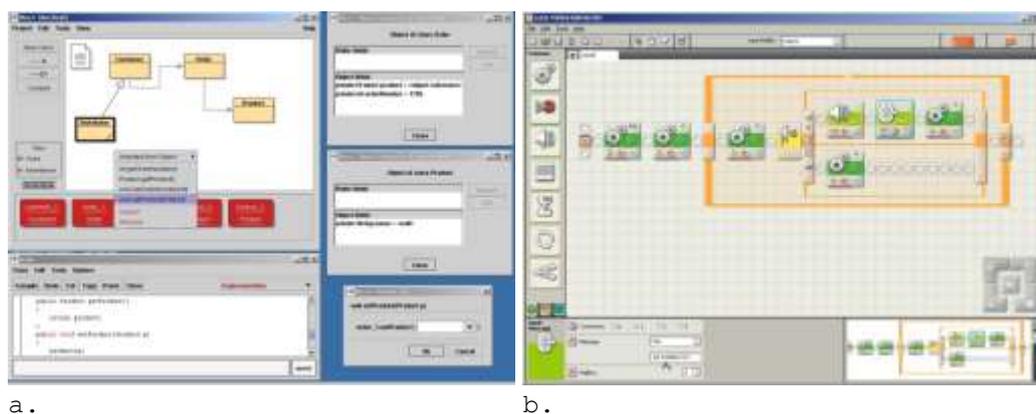


圖 2-29. (a)BlueJ；(b)MINDSTORMS NXT

相類似運用圖表繪製方式進行設計，但更著重在多媒體，尤其是即時繪製電腦影像上的還有 Quest3D(圖 2-30a)、Virtools(圖 2-30b)與 vvvv(圖 2-30c)，他們更強調在即時 3D 影像的處理上，例如匯入在其他 3D 建模軟體建立好的模型，並在 Quest3D 或是 Virtools 中設定各個元件的屬性以及運用流程圖表連結各元件以產生彼此的互動(如：碰撞、攝影機的移動軌跡等)，便可輸出讓原本靜態的立體場景變為動態的虛擬實境，vvvv 則是讓使用者透過圖表進行串接便能進行影像的處理、轉換或是各種特殊效果的呈現，這類軟體雖更著眼在即時影像上，但設計流程與一般視覺語言相較之下並沒有太大的差異。



圖 2-30. (a)Quet3D；(b)Virtools；(c)vvvv

除了以圖表來進程式設計，也有為了小朋友設計的用堆積木的概念來進行的視覺化成是設計系統。麻省理工學院媒體實驗室發展的 Scratch，則是為小朋友設計的視覺化程式設計系統。由於是為小朋友設計的，因此是以堆積木的方式進行各種互動形式的設計，設計者除了可以有視覺化的回饋之外，也可以透過這些邏輯區塊(logic blocks)的組合具體化設計的概念與想法。Scratch 主要的概念便是認為，不是每個人都會成為專業的作家，但是寫文章卻是一種必備的技能，同樣地在

程式設計也是一樣。不必然每個人都為成為一個專業的程式設計師，但是能夠運用簡單的程式邏輯以完成一些想法或傳達概念在未來的族群卻會是一個必備的溝通工具(圖 2-31a) [Resnick, 2007]。樂高公司的腦力激盪 (Mindstorm) 則是另一個例子(圖 2-31b)。



圖 2-31. (a) 麻省理工學院媒體實驗室發展的 Scratch；(b) Mindstorm

而為了讓互動系統的設計流程更為容易簡化，許多的研究單位與企業也開始進行如何簡易地整合各種軟硬體的研究或產品。例如：Phidgets [Greenberg & Fitchett, 2001] 是多種模組化的硬體裝置系統，具有感應、動作、發光二極體等各種模組，各模組間透過簡單的組合便可相互合作與傳遞訊號而不需透過複雜的電子設計(圖 2-32a)。Hernando 在愛茉莉亞互動設計學院 (Interactive Design Institute Ivrea) 發展的 Wiring [Hernando, 2004] 結合了基本的硬體輸出入介面卡以及衍化自麻省理工學院發展的 Processing 的整合式軟體設計環境(integrated development environment, IDE)，並可以與前述各種軟體設計工具相連接(圖 2-32b)。Arduino(圖 2-32c) 相類似於 Wiring 同樣也可跟許多軟體設計工具相連接。

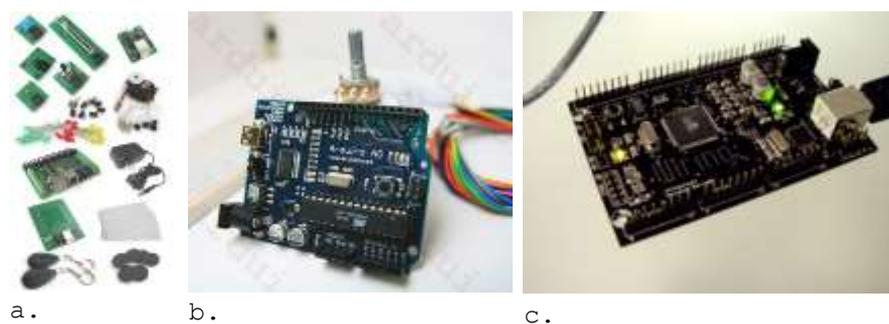


圖 2-32. (a) Phidgets 的各種模組；(b) Arduino；(c) Wiring

Microsoft 的 Robotic Studio 是用來設計機器人以及其互動的工具，可接受來自感應器 (sensor) 的資料並在處理後送至傳動器 (actuator) 以控制機器人的行動，除了運用視覺化語言來進行的程式設計環境 (圖 2-33a)，還包括了可利用 3D 場景與物理碰撞系統來即時模擬設計出來的機器人 (圖 2-33b)，並可透過網頁的方式來與電腦進行連結與監看 (圖 2-33c)。



圖 2-33. (a) Robotic Studio 設計環境；(b) 設計模擬；(c) 控制介面

3. 應用案例分析

從前章的文獻回顧中，我們看到了資訊系統逐漸被嵌入到建築空間中，也看到了建築空間裡開始充滿各種資訊視覺的互動，資訊元件與空間的實體結合共構成智慧空間。本研究要透過本章的應用案例分析，從數個智慧空間的應用案例中歸納出來幾個智慧空間的類型，並分析這些智慧空間在設計過程中有哪些特性，藉以了解智慧空間中資訊與空間是如何地結合。先由空間的尺度大小來做為分類依據，大致可以將智慧空間分為兩種類型，第一類是與建築實體產生連結進而影響實體外觀的互動建築、第二類則是將資訊透過視覺化的形式遍佈在空間中並藉以與使用者進行溝通或傳遞訊息的資訊空間。

3.1. 案例分析方式

案例分析的方式，我們採用了邱浩修在建築設計中運用「實虛共構」觀念做為互動介面的構築方法研究中，他認為數位技術的持續發展使我們預期未來建築設計的重心將不僅是空間、量體的建造，而且是介面、軟體的建造，未來建築師所要面對的主要設計挑戰將會是如何根據不同的設計條件，適切而靈活地轉換運用可觸知與不可觸知的實體（材料、物件、表皮、空間等）與虛擬（影像、聲音、網路、資訊流等）設計元素的同時，如何忠實呈現實虛共構設計在數位技術、新生活機能與形式美學結合的構築一致性，並思考透過這種一致性怎樣反映出建築在數位時代的新價值與意義，在這裡他借用建築理論家 Kenneth Frampton「構築」的觀點用來討論「數位構築」(Digital Tectonics)的方法架構，一方面在建構建築設計領域以數位（及資訊）為創作材料或結構方式，而非僅被視為為形式操作工具或隱喻性的哲學思維的設計論述基礎，另一方面探索實體空間與虛擬空間疊合時產生的可能空間構造模式與其相對應（結構、功能、美學上）的構築意義 [邱, 2008]。荷

蘭研究機構 v2_lab 出版的超都市主義 [Joke Brouwer, 2001] 一書中直指場所不再由地理及建築的邊界所界定，當像素與混凝土的融合在各個層面都無所不在的今日，城市成為由經驗、生活型態、休閒消費、對話溝通以及媒體介入所構成的事件場域，而建築則是中介這種新轉變的要角。空間的構成元素轉而變為是由虛擬的像素與實際的混凝土兩者的混成，不再僅僅是建築的實體而已，這也就是邱浩修所提的實虛共構觀念。

空間結構以及虛實對應的分析側重於案例本身的設計概念與最終的產出，然而作為一個電腦輔助設計系統，我們同時也需要分析這些案例的設計行為或是其在設計過程中對於空間設計概念上的啟發，從而得以理解本研究的系統在設計過程中需要提供的機能跟概念以輔助設計師們設計這類新型態的、混合了虛擬與實體以及智慧互動的空間。

由此，我們在本章後面就透過 1) 結構方式、2) 虛實對應以及 3) 對空間設計的啟發三個向度來做為案例分析的架構，在各案例的 1) 結構方式部分以構成空間的實體結構作為分析的對象，探討構成空間的實體結構的設計，包括單元結構的細部與整體的控制系統構成，偏重於軟硬體及機制部分；在 2) 虛實對應部分，則是以空間的實體與虛擬的像素之間的連結與互動作為分析的內容，探討該案例如何透過實體與虛擬之間的連結與互動，實虛共構地搭配出空間的機能；而在 3) 對空間設計的啟發部分，則從設計的過程與其設計行為的角度作為分析的內容，探討各案例在於設計過程中需要的概念。

3.2. 互動建築

傳統上的建築量體是一種靜態的實體，本身並沒有可變性，結構在建造後就會維持相同的形態，不會改變。然而隨著科技的進展，建築物本身不再是一種完全靜態的存在，而是可以透過形狀的改變或是各種可變結構的連動，讓自身形態產生變化，而呈現出多樣化的外觀。而這樣的改變，除了建築物本身的變形之外，也另外可透過擴增的方式將數位媒材結合到空間中，共同成為空間的構成份子。

常見的形式便是以各種視覺媒材，例如：投影機投射出的影像、大型的發光二極體

(LED) 電視牆甚或是以各種機械結構組合成的牆面，透過像素或是小單元結構元件的組合構成建築的外觀來取代傳統的建築立面，讓原本不易改變的建築立面能有動態的效果呈現。而這些動態的視覺效果呈現，除了用以呈現特定的視覺化訊息與內容之外，甚至也可以依據建築物外的路人或是建築物內的使用者進行互動，這類的空間形式也被稱為媒體立面 (Media Façade)。而像這樣以視覺媒材的形式來呈現視覺內容的原理，就是將單一的小塊元件，或許是機械的物理結構或是電子的發光元件，作為資訊系統裡各種圖像像素的對應，這些小塊元件組合而成大面積的立面或是建築體後，便如同像素組成圖像內容一般，只要調整或是變動每一個獨立的小塊元件，組合起來看就是各種不同的視覺內容，簡言之，將建築元件有如電腦像素般地操作便是互動建築的最主要概念。

在互動建築的案例上，我們參考了邱浩修所提出的四種互動建築的主題：輕逸、移形、觸發以及網絡，分別列舉了四個案例，各自對應至輕逸主題中的媒體立面項目、移形主題中的可變結構項目、觸發主題中的適境觸發項目以及網絡主題中的資料反饋項目來做為分析的對象。

- 輕逸 - 媒體立面 - Hyposurface Mediafacede (案例一)
- 移形 - 可變結構 - Muscle Tower II (案例二)
- 觸發 - 適境觸發 - Dynamic Façade (案例三)
- 網絡 - 資料反饋 - Kunsthaus Graz (案例四)

3.2.1. 案例一：Hyposurface Mediafacede

dECOi 的 Hyposurface Mediafacade [dECOi, 2011] (圖 3-1)，利用大量的金屬板作為建築的立面，以不同的高度反射光線，進而建立起深淺不一的視覺內容，外牆並會隨著使用者的移動，反映出使用者的輪廓，是一個會與使用者互動的牆面。



圖 3-1. Hyposurface Media 及其結構元件

1) 結構方式

Hyposurface Mediaface 所使用的單元元件，是由油壓系統推動的八片三角形金屬板。單一的元件，依據油壓棒推動的高低，可以讓金屬板呈現出多種不同的角度，呈現出有如花開花謝般的效果。而這些開展出來的金屬板，再透過光照的反射，便能夠呈現出不同的亮度深淺質感。將每一個單元元件排列並連結在一起後，就可以拼構成建築物的外牆，改變每一個單元元件中的油壓棒高度，就能讓牆面有著不同深淺變化，充分運用了利用可變的機械結構改變建築外觀的原理。運用油壓系統可以精確地控制金屬板高度，金屬板則因為其反光效果而呈現出色彩上的變化層次。



圖 3-2. Hyposurface Mediafacade 的互動性

2) 虛實對應

媒體立面的案例都用著相同的機制，也就是將這大量的實體元件視為電腦中的像素，像素的顏色直接對應到實體的機械性或物理性變化。單一像素的顏色以灰階的數值來看可以有 0 到 255，0 代表黑色，255 代表白色，中間的各數值就代表著不同程度深淺的灰色變化，共可以有 256 種程度上的變化。這些像素的數值可以直

接對應到油壓棒的高度，0 代表最低，255 代表最高，這樣就可以透過控制系統，判斷出各個油壓棒的高度並賦予電力產生變化。例如一面長邊有 320 個，短邊有 240 個油壓棒單元的立面牆而言，便可以對應到寬為 320 個像素，高為 240 個像素的圖像，圖像中位置在 (10, 13) 的像素點顏色值為 128，那就透過控制系統，將立面上左邊數來第 10 個與上面數來第 13 個的油壓棒調整到一半的高度，依此類推將所有的油壓棒依據圖像裡每一個像素的灰階數值進行調整，便可以讓立面呈現出該圖像的內容 (圖 3-2)。

透過單元的油壓棒高度產生深淺的變化，可以用來呈現出相對應的圖像內容，所以任何可以在電腦中繪製出來的視覺圖像，便就都可以以本案例中的立面來表現。其中便包括了攝影機的畫面，本案例將攝影機的畫面連結到立面牆上，將拍攝到的畫面，以彩色轉灰階的方式，將顏色的數值過濾掉，而只留下明暗的數值，再將這些數值對應到裝置裡的每一根油壓棒，建築物本體就可以直接地呈現人臉的畫面，而隨著攝影機拍攝到的畫面不同，建築物的立面也隨之改變，彷彿建築物看得見一般，直接地與環境中的變化做了連結，也根據這些訊息讓自體產生變化，呼應了環境，而有了與使用者互動的感覺。實體的元件與虛擬的像素，透過這樣的形式進行對應並加以結合，打破空間中實體的既定形象，而得以反映出虛擬資訊的變化。

3) 對空間設計的啟發

本案例的設計行為中，設計的概念從傳統整體的設計轉而專注於單元結構的設計以及最終型態的變動性，設計師必須運用如同於電腦數位圖像中的像素概念來進行設計的創作，並經由各單元結構的變化來思考最後的構成型態，如同像素的變化構成圖像一般。在空間設計上，本案例的建築皮層系統透過金屬板的輕薄性與高低變化，呈現出一種動態輕逸的構築型態，也因為這樣的設計概念上的轉變，虛擬的資訊變化如何與實體的空間元件之間相互對應成為建構這個案例的關鍵過程。

在這個案例中，空間元件是資訊元件的延伸，例如金屬板是空間元件，而在金屬板後面的油壓棒控制電路則是資訊元件，兩者的結合缺一不可，空間的元件讓資訊元件得以將要呈現的內容透過油壓棒的變化呈現出來，油壓棒控制電路這個資訊元件也必須具備空間資訊，才能精確地操作與控制空間元件。再者，攝影機則是另一個

資訊元件，負責將訊息傳遞到系統中進行處理，讓空間元件得以依據攝影機傳來的影像進行各種的變化，因此空間元件與資訊元件必須相連接而且互相傳遞訊號，資訊元件才得以判斷現在是該控制空間元件升起或是下降。

3.2.2. 案例二：Muscle Tower II

Hyperbody Research Group 的 Muscle Tower II [Oosterhuis, 2007]，讓傳統的建築結構成為可動的，這座塔會隨著觀賞者的位置而擺動，讓原本靜態的建築物仿如有了生命一般與使用者互動。而透過單元組合的變化，也讓這個結構體可以有許多的可能性，可以變成塔，也可以組合成床板、天花板或是各種平面立面等等。



圖 3-3. Muscle Tower II 及其結構上的攝影機

1) 結構方式

本類型的案例，是以關節作為單元元件，透過對於每個單元關節角度的控制，達到讓整個建築結構產生變形的效果。而每一個關節單元，是由一個固定的節點，連結三組油壓桿所組成，油壓桿便是用來支撐本體的主結構。而除了用以支撐本體的結構之外，在關節的節點上也額外連結了一部小型攝影機，可以用來偵測靠近的使用者或是環境上的變化，讓最後組合出來的結構具備了與環境互動的基本功能。而以關節作為單元，也讓最後的建築本體可以有更多樣化的變化，例如可以做到類似轉頭、彎下腰或甚至是扭腰擺臀的動作（圖 3-3）。

將每一個關節單元元件的節點，以油壓桿相連結後，便可以組合成立體的結構。軟體則採用了 Virtools 作為控制用的軟體系統。首先需在 Virtools 這個軟體環境中依據組合出來的實體結構，建立起相對應的 1：1 虛擬模型，每個模型中的關節就是直接對應到實體結構裡的關節單元元件，而軟體提供的物理模擬系統，則賦予這些虛擬模型最基本的物理特性，例如硬度、應力等等的計算，讓虛擬的模型能盡可能擬真地接近實體具備的特性。再透過控制每個虛擬的關節模型上的油壓桿模型長度，便可以改變模型的造型，而每一個油壓桿的模型長度，直接地去對應到實際的油壓桿長度，模型縮短，實體的油壓桿也就縮短到一樣的長度，這樣就可以透過軟體系統來控制實體的結構，藉以讓實體能夠跟著虛擬模型般地產生變化的效果（圖 3-4）。

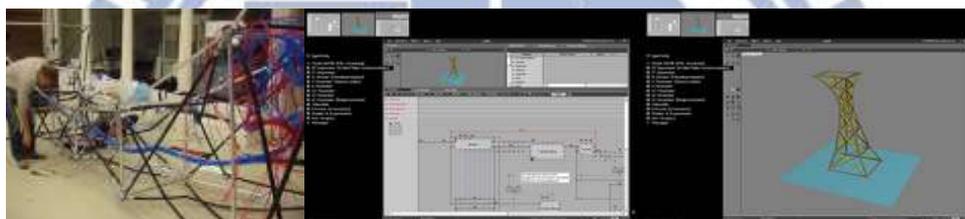


圖 3-4. Muscle Tower II 以 Virtools 作為系統的互動控制工具

2) 虛實對應

藉由控制虛擬的各個模型元件，來改變實體的建築結構，再結合攝影機拍攝的畫面，可以用來判斷附近環境中使用者的狀態，藉以表達出各種的動作，例如當某個關節攝影機拍攝的鏡頭畫面傳至電腦後，經由電腦分析判斷出來得知有使用者靠近該攝影機，便可以透過研究出來的演算法來改變各個關節的油壓桿長度，讓案例中的建築結構做出靠近該使用者或是遠離該使用者的動作，而表現出一種模仿生物的行為舉止。而由於每一個關節都有攝影機，所以讓本案例的結構可以同時地與多個使用者產生互動行為，呈現出彷彿在跟觀眾們玩耍般的感覺。

3) 對空間設計的啟發

上一個案例的媒體立面，是利用數位方式控制結構單元的變化來達到溝通或是互動的效果，每一個單元之間是一種離散式的關係，建置的成本與技術因為只需要考慮

單元結構，相較之下也更為經濟與單純。然而，就如同邱在其文中所說『...以「實體或類比」之動態皮層或結構的行變來傳遞訊息或進行互動，則賦予建築更進一步語言和形體表達上的彈性，使溝通的感知層面延伸至視覺以外（如觸覺）的可能，並在機能上發揮如動態適應環境條件等實質作用。...』，本案例透過一個大型的、可變動的立體空間結構，重新建立空間的實體與人類之間的關係。

本案例中，結構的變化不再是單一的、離散式或數位化的元件對應結果，而是一個連續的、類比式的動態線條。空間實體的結構變化來自於類似骨骼的柔性構架系統中各個節點的相對位置與連結的張力，設計概念亦不同於媒體立面的單元思考，而是需建立完整的實體形態，並以連續變化的思考模式來看待每一個節點的移動，每一個節點的變化，又拉動或牽引著與其相連接的其他節點，連續地完成整體結構上的移形。而其與人類之間的互動機制，則是其可適應性的機能展現，這種視覺上連續變化的過程，更能表現出「靠近」或「遠離」這樣的關係，豐富了設計概念的最終呈現。

在這個案例中，我們看到攝影機這樣的資訊元件，可以傳遞訊息到電腦中進行判斷與處理，再經由軟體控制系統控制空間中的結構元件，構成了完整的互動行為，宛如賦予空間元件智慧一般，資訊元件把訊息傳遞給空間元件，而空間元件成為資訊元件的延伸，嵌入資訊元件的結構可透過感應器與邏輯系統的動作，除了本身量體即具備的外型與支撐功用外，還能擴充情境判斷以及提供各種不同動作與行為的能力。而每一個攝影機的資訊元件也必須具備空間資訊，後台的系統才得以知悉訊號來自於空間的什麼位置，而能夠正確地依據對方的動作做出各種反應，才有互動的效果，而不是只有隨意擺動。

案例透過可感應人類動作的感應器，將其嵌入在建築物中用來偵測使用者的動作資訊，如：使用者的移動行為，或是建立使用者與建築物之間的相互關係，如：彼此距離的遠近等，再經由電腦進行運算處理後，控制改變建築物的機械結構進而影響建築物的外觀形態而不僅僅是立面而已，有如建築物可以察覺使用者的行為或是動作並與他們互動一般，讓建築物有著生物般的相貌。而由於結構的控制牽涉的技術層面較廣，這類的案例較少，而且進展相較於媒體立面形式的互動建築而言也慢得

多，但從下例的分析，我們仍得以看出這類仿生的互動建築，在其中的資訊元件與空間元件之間的關係其實也有著與媒體立面相類似的特性。

3.2.3. 案例三：Dynamic Façade

Giselbrecht + Partner ZT GmbH 於 2010 所設計的 Dynamic Façade，透過控制立面上的每一片金屬外牆元件，除了可以改變建築外觀之外，也可以藉此調整室內的光照與溫度，來達到節約能源的目的，這些動作可以是自動化的，也可以是被控制的（圖 3-5）。



圖 3-5. Dynamic Façade 可以改變建築物外觀也兼具功能性

1) 結構方式

本案例的單元元件是一塊可摺疊的金屬板，加上底部的滑軌結構。透過滑軌的移動，讓金屬板有不同的摺疊程度，進而影響進到室內的光照，也連帶影響了室內的溫度。設計的發想來自傳統的摺疊式窗戶，開窗與否也同樣影響了建築物的面貌，窗戶除了機能的提供，同時也影響外觀的呈現。這些單元元件是可以置換式的，而且其金屬材質也讓它們在夜晚時可以用來反射室內的燈光，提高室內的亮度與燈泡的發光效率，也具備節約能源的可能性（圖 3-6）。這類型的實體元件結構，相較於媒體立面雖然同樣有著可變化的外觀，但同時也是多種生活機能的提供者，例如：遮蔽、調節溫度、光線...等等，因此在變化的依據上，除了外觀的變化外，更有著與環境狀態產生互動的特性在，例如：白天光線過強時進行外牆的變化，讓空間中的亮度降低，外觀變化的同時，內在機能也隨之產生變化。

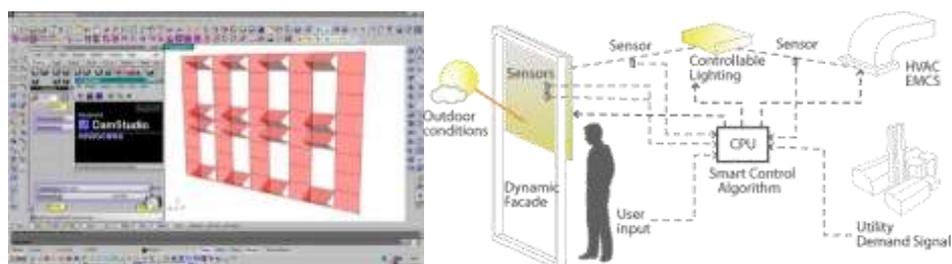


圖 3-6. Dynamic Façade 結構單元以及控制邏輯

本案例的控制系統由 1 1 2 個單元元件所構成，透過 Rhinoceros 軟體與 Grasshoper 進行模擬動作，再結合偵測器偵測太陽的光照強度與室內的溫度，將這些資訊傳到電腦控制系統後，用來調整每個單元元件的金屬板摺疊程度。這個控制系統除了改變室內亮度與溫度之外，也同時控制了建築物的外觀呈現。



圖 3-7. 視機能需求或外觀的變化需求進行立面的控制

2) 虛實對應

本案例除了使用者透過電腦系統去調整之外，系統亦會透過裝置於其上的偵測器偵測現在的太陽光照強度與室內問題自動地進行判斷與調節，而系統在改變單元元件的同時也調整室內燈光的亮度與空調的溫度，讓使用者察覺不出來這些改變。每一個房間同樣也可以自行控制，或是因應外觀上的變化需求而進行調整（圖 3-7）。看不見的環境訊息，因為智慧系統的連結，讓空間構件發生變化以對應到需求的調整上，實體的元件也因而得以與虛擬訊息相呼應，共同合作以提供更理想的機能型態，而不僅僅於改變外貌而已。

3) 對空間設計的啟發

本案例讓空間以觸發回饋的形式回應了環境的變化，是為一種適境觸發的設計概念，透過對於環境變數的感知，並做出適當的回應。空間設計的量體與其建構出來的機能並不是一種固定的狀態，而是會隨著需求、時間的週期性變化或是偶發性的變化來做適切的調整，設計的過程除了必須運用更大的結構彈性，同時也需要蒐集並歸納需要進行呼應的環境變數，如：溫度、氣溫、光照等，並在結構的設計中進行其與各變數相互對應的變化形式，例如溫度與空調強度的對應、光照與窗戶結構角度的對應等。設計的重點除了本體的形體之外，也要讓不可見的環境訊息與空間構件的變化機制進行對應式的連結。

在這個案例中，我們看到資訊元件傳遞訊息給空間元件，每一個資訊元件如滑軌的電路系統或是房間的電燈也都具備著空間資訊，讓系統可以準確地控制特定房間的亮度與溫度。我們同樣也看到資訊元件與空間元件的相連接，彼此合作，提供使用者更舒適的生活環境，同時也兼顧了能源的使用效率。

3.2.4. 案例四：Kunsthau Graz

Kunsthau Graz [Lubell, 2003]，利用燈管與像素的對應，讓建築的外觀可以呈現各種不同的圖樣變化，除了可以用來呈現影片之外，也可以即時地播放訊息或是院內正在播放的節目內容，讓空間內的觀眾與空間外的其他行人可以透過這些媒體內容將意識結合在一起。



圖 3-8. Kunsthau Graz 的整體建築外觀及其結構單元

1) 結構方式

Kunsthhaus Graz 所使用的單元元件，並不是特製的部品，而是很常見的圓形日光燈管。透過供給不同的電流，讓燈管發出不同強度的光，藉以對應到電腦圖像中的灰階數值。而由於電流的輸出與控制並沒有辦法像油壓系統那般地穩定，所以雖然圖像的灰階有著 256 種顏色的變化需求，本案例的單元元件只能呈現出大約 8 種的變化。因此每個像素還需要再做一次的轉換，把 256 種變化對應到 8 種強度，作法就是將灰階值的 0 到 31 對應到日光燈管的全暗，也就是像素的數值如果是 31 以下，那麼燈管就不亮，32 到 63 則對應到燈管的最小亮度，依此類推到像素的灰階值如果是 224 到 255，那麼該燈管就是最亮的程度。相較於機械式結構，本案例的單元元件是屬於電子式的，故障率較低也便於維修與安裝（圖 3-8）。



圖 3-9. 透過特製的 BIX Simulator 軟體來控制建築物外觀的呈現

雖然亮度的變化只有 8 種，但是由於其相同的單元元件特性，讓本案例的外牆可以用來呈現具體的影像內容或是呈現文字。而本案例最大的一個特色，便是為了讓業主可以方便控制呈現在外牆的影像，設計師額外製作了一個控制用的軟體，以方便日後的營運使用。這個軟體稱之為 BIX Simulator。BIX Simulator 可以獨立地控制每一個單元元件的亮度，也可以選擇開啟影片檔案，讓外牆直接變成電視般地播放影片內容。除此以外，也可以設定觀看的角度，具體模擬出使用者在戶外看到的建築物外觀樣貌。這套模擬軟體，讓本案例得以精確地控制建築體本身所想呈現出的形態（圖 3-9）。

2) 虛實對應

媒體立面的特性，讓 Kunsthhaus Graz 可以在立面上繪製出各種圖像或文字，例如他們可以在外牆上呈現出現在劇場裡呈現的內容或是廣告片段，也可以表現如同跑馬燈一般的文字訊息，讓建築物與所處環境中的人們產生對話，也打破了建築物

室內與室外的界線，建築本身就成為一種媒介，在意念上連結了兩個實體被區隔開來的空間。

3) 對空間設計的啟發

本案例讓內容可以隨時更換並且可以呈現出具體的訊息，而其與媒體立面不同的是，透過網路傳遞來的資料，讓外牆做即時地回饋，更是讓建築空間成為與人們生活在同一個城市底下的生命體，而不僅僅只是空間結構的一部分，而是另一個活生生的居民，建築量體的角色也隨之更趨複雜。設計的過程，在於如何打破實體與虛擬環境的界線，讓實體空間的區隔不再存在，外部的牆面成為了內部空間訊息的延伸而不是一個界線，讓空間的實體與在其間的虛擬活動或訊息相結合成為一個混成的生命體。設計的概念上不再把空間視為一個載體，而是轉而為思考其與外界對話的方式、傳遞的訊息內容、內部與外部的溝通網路等，設計師仿佛是在安排與規劃利於城市中此建築量體與周遭的人際關係般，而不僅止於本體的外型或機能。

在這個案例中，空間元件成為資訊元件的延伸，資訊元件例如燈管的控制電路以及後台的軟體系統 BIX Simulator，讓建築物外體除了本身具備的外型外，還能作為媒體呈現資訊的內容，而每一個燈管上的資訊元件必須具備空間的資訊，後台的軟體控制系統才得以將控制用的訊號傳遞到正確的資訊元件去。而空間元件與資訊元件必須相連接而且互相傳遞訊號，資訊元件才得以判斷現在是該控制空間元件點亮或是變暗。

我們可以看到在互動建築類型中，資訊元件會傳遞訊息給空間元件，例如：感應器取得人類的資訊後傳給油壓棒來改變牆面或柱等空間的結構體以與人類互動。而資訊元件也必須具備空間資訊，讓系統得以透過這些空間的資訊了解感應器或控制電路的所在位置而判斷該進行何種反應。空間元件與資訊元件相連接，讓系統得以運算與控制，而資訊元件也透過改變空間元件的形式，將自身傳遞訊息的能力延伸到空間裡，空間元件成為資訊元件的延伸。

3.3. 資訊空間

我們對資訊空間的定義，其一是在空間中呈現資訊內容，例如透過投影或螢幕、燈

光等方式，將虛擬的資訊呈現在空間中，這些媒體化的資訊亦成為空間構成的一部分，並可與使用者互動，人們操作空間中的資訊如同我們使用空間中的器物一般；另一種則是透過嵌入感測器的方式賦予空間中的家具或物品智慧，例如傳統的家具，而不論哪一種，都是透過擴增的方式，將虛擬的資訊結合進真實的空間裡，改變了空間與使用者之間的關係。

資訊空間的案例，我們以系統的整合程度作為分類，將案例分成四種類型，第一類為智慧家具，賦予智慧的方式便是將感測器嵌入到家具中，這樣的案例多數都是屬於單獨獨立的物件。第二類為互動地板，空間構成實體中的地板，除了原先已具備的支撐機能之外，能夠透過對環境觀察的資訊元件，取得使用者的訊息，再透過自身來使用者與之互動的可能性。第三類是互動立面，將空間中的各種立面，例如各種牆面、窗等視為資訊的承載介面，透過這些介面讓使用者可以直觀地與虛擬的資訊相互連接與互動，實體空間成為一部大型的虛擬電腦系統，而使用者則身在其中。第四類則是相互整合的系統，我們以擴增廚房作為探討的案例，此案例將前述各自不同的項目整合起來，從服務的角度提供互動功能，讓整個空間中的各種物件隨著廚房的工作流程與使用者的需求產生各自的訊息，共同構成完整而具動態感的廚房機能。

3.3.1. 案例一：智慧家具

透過在桌面上嵌入觸控面板，或是在家具中嵌入各種資訊元件的方式，讓傳統的家具被賦予了運算以及相互溝通通訊的能力。這些被擴增功能後的智慧家具，除了原有的功能性之外，也依照嵌入的資訊元件的不同而有了截然不同的特性(圖 3-10)。



圖 3-10. 家具嵌入資訊元件後成為智慧家具，圖為微軟的 Surface[Dietz et al., 2009]

1) 結構方式

要設計智慧家具，第一個步驟就是要讓系統得以接收到來自該家具實體的訊息，因此必須要能夠把可以跟電腦系統相互連結溝通的資訊元件嵌入到空間的實體中。例如 reacTable 就是透過貼上 AR 標籤的形式，讓隱藏在桌面下的攝影機拍攝並辨識該標籤，系統便可以識別出該空間元件的身分為何以及面對的方向，從身分以及面對的方向就可以做出許多種的變化，例如辨識出來是代表音量的元件，那就可以再依據元件的方向來判斷音量的大小，如此一來使用者就可以透過轉動該空間元件來控制環境中的音量 [Jorda, 2007] (圖 3-11a)。透過資訊元件中的加速度計，嵌入到盒子中後，便可以透過盒子的翻轉判斷目前轉向哪一面，再經由系統控制空間中的燈光明暗搭配盒子的指示訊息來呈現出溫度、濕度、天氣等等訊息。搭配多重觸碰螢幕的桌子，除了可以放杯子之外，也可以在桌面上顯示各種視覺訊息供使用者操作。在椅子裡埋入壓力感應器，就可以偵測使用者是否坐在上面，以決定是否啟動震動馬達。在房門口放入人體紅外線偵測器來判斷是否需點亮房間裡的燈光。在電視後方加上 LED 燈光，透過聲音感測器根據現場的音量與吵雜程度提供不同的燈光變化，來改變空間的氛圍。這些智慧家具已經開始逐漸充滿生活的空間裡，讓人們得以進一步與家具互動，或是這些家具之間相互連動，自動化地共同為空間的使用者服務，也就是智慧家庭的概念 [Fernandez, 2013] (圖 3-11b)。



圖 3-11. (a) reacTable 運用 AR 技術；(b) 顯示多種資訊的造型燈及控制顯示內容的遙控器

2) 虛實對應

智慧家具類型案例的互動機制是維持著原本空間元件該具備的特性，例如旋轉、堆疊、翻轉等等，在這些空間元件中嵌入感測器來偵測這些變化，再將這些感測器接收到的動作訊號與資訊元件的控制進行連接，就創造出各種不同的互動可能性。例

如使用者可能是透過旋轉空間裡桌上的花盆來調整音響資訊元件的音量，或是透過手機控制家庭裡的燈光、空調或是窗簾，甚至讓這些資訊元件相互連結，共同組成一個網路系統，彼此協調工作，在使用者進門時燈光調亮、空調啟動並播放音樂，在使用者離開時自動將這些關閉以節約能源。

3) 對空間設計的啟發

智慧家具的設計概念，讓傳統上的實體，與資訊互動機制相結合，而得以提供更豐富或多元的機能。在設計的過程中，散佈於空間裡的各個家具，都具備可以被結合與提供互動的可能性；設計的焦點已經不僅僅是這些家具所提供的基本物理特性，而是在這些家具所在的空間位置如何進行對於空間中使用者的感知，例如：椅子本身具備承受體重的物體特性，判斷重量或是壓力的感測器就很適合與其結合並進行行為訊息的偵測並進行互動以提供更多樣的功能。但這並不代表其他類型的感測器就不適合用於其上，感測器與家具的排列組合可以是無窮無盡，在這些組合中或許會看到前所未見的互動效果。在智慧空間的設計裡，滿足需求只是基本，如何透過更豐富的互動機制，創造出未知的需求更是重要的面向。

智慧家具的案例類型，資訊元件必須具備明確的空間資訊，且將其自身與空間元件做緊密的結合，讓使用者在操作傳統的空間物件的同時，其動作與行為也可以被資訊元件的感測器偵測得到，從而提供更豐富的互動回饋。

3.3.2. 案例二：互動地板

在近年的展場空間裡，互動式的地板是非常常見的一種形式，最主要的概念就是提供參觀的群眾一個與空間互動的機會，並透由參觀的人流改變了室內空間的影像，進而能夠呈現多樣化的面貌（圖 3-12）。而地板這個空間元件的基本功能就是必須要讓使用者得以站立其上，因此在結構強度的考量上，多以投影的方式作為資訊視覺化的機制，或是必須具備特別強化過的表面以保留原本空間元件的特性（圖 3-13）。



圖 3-12. 互動地板的類型的一種，以透過燈光方塊的形式進行互動 [Lightspace, 2005]

1) 結構方式

這類型的案例，是將一部具備紅外線探測的攝影機嵌在天花板上，用來大範圍地拍攝展場。經由紅外線的反射，讓攝影機攝製的畫面可以拍到各個參觀者的影像並去掉背景，再經由電腦進行電腦視覺的分析後便可以取得所有參觀者的位置，接下來就控制與每一個參觀者相對應位置的燈光來做出反應。隨著參觀者動線的改變，地板的影像也隨之改變。運用到的資訊元件僅有投影或燈光系統以及攝影機，在電腦系統端如何透過攝影機取得的資訊進行視覺畫面的變動與合成是這類案例中最重要設計。



圖 3-13. 以 LED 作為地板需考量強度 [Shenzhen Reidz Tech, 2010]

2) 虛實對應

這類型案例的互動機制可以分成幾類：

一、位置偵測：以燈光的亮暗提示參觀者的位置，這種類型由於製作上相對簡單，是最常見的一種類型，讓參觀者產生興趣，感覺被注意到。

二、動線軌跡：將使用者的移動軌跡以向量的形式記錄下來，並透過方向與速度來對影像的內容進行影響，更精確地判斷使用者的行為。

三、遊戲內容：機制上雖然相類似於位置類型，不同的地方在於讓畫面裡的虛擬物件會與使用者發生碰撞，進而變成一種可以多人進行的遊戲（圖 3-14）。



圖 3-14. 讓玩家彼此玩樂的互動地板遊戲 [Luminvision, 2009]

3) 對空間設計的啟發

本類型的案例中，空間對於使用者位置的掌握與回饋成為最重要的設計概念。空間中的使用者不再只是單純的行人，而是共同建構空間表情的元件之一，依據他們在空間中的移動，進而繪製軌跡、呈現內容或是與虛擬物件產生互動，都讓空間與人之間的關係被重組；透過位置的紀錄，使用者成為空間的一部分，環境的印象與氛圍由在其間的人們所共同決定，空間的設計不是以建立一個設計主體為標的，而是架構一個舞台，主角是使用者，最終設計則由這些特定或不特定的人群來完成。

從這類的案例我們觀察到，資訊元件要彼此傳遞訊息進行連結之外，空間元件成為資訊元件的延伸，視覺化的資訊內容必須在正確且精準的空間位置顯現，才得以構成充分的互動效果。

3.3.3. 案例三：互動牆面

相較於互動地板，互動牆面因為可以取得參觀者的外形資訊，而能夠做出更為細緻的畫面變化。且因為使用者並不像互動地板需踩在空間元件上，因此在視覺化的資訊元件的選擇就較為多樣化（圖 3-15）。



圖 3-15. uma SKIN Multitouch Display[uma, 2012]

1) 結構方式

互動牆面的案例在資訊元件的部份也是採用攝影機作為主要的偵測來源，不同於互動地板的形式僅能取得使用者的頭部，互動牆面案例的攝影機在空間中的位置較低，通常不會是從天花板往下拍而是從側面拍攝，因此能夠取得使用者的身體外型。在內容的變化上，也較常見呼應使用者身體外型的視覺內容。視覺化的資訊元件，因為不需要考慮支撐的強度，所以除了有投影的系統之外，也有由 LED 燈泡組成的牆面、多台電視或螢幕組成的電視牆[FlightPhase, 2012] (圖 3-16)。在偵測端的資訊元件上，除了攝影機之外，也有透過多重觸碰平板作為偵測的來源，這兩種偵測器原理大致相同，都是透過過濾背景物件的形式，只留下人物的外型，多重觸碰平板則是留下接觸到表面的手指頭的外型。另外也有運用距離感測器來進行與使用者之間的互動。



圖 3-16. FlightPhase 的互動牆變化

2) 虛實對應

互動牆面的機制則因為輸入來源的不同而分成兩類：一種是攝影機畫面類型，視覺上除了呈現資訊之外，也會依據人體的外型產生變化，例如在剪影上製造火花效果或是點亮人形涵蓋範圍的 LED 燈等等。另一種則是多重觸碰類型，這一類多以操作性為主，畫面提供的是可供操作的介面影像，使用者透過手指在牆面上的碰觸與虛擬的影像資訊互動，進而取得視覺化的資訊內容 [Multitouch Limited, 2009] (圖 3-17a)。由於多重觸碰類型可以讓多個使用者可以同時透過投影的牆面與在其間的資訊互動，由於不需要使用滑鼠或鍵盤，輕鬆而直覺的操作讓這樣的裝置在許多的展場都已經成為相當常見的設計 [Hard Rock, 2009] (圖 3-17b)。



圖 3-17. (a) Hard Rock 的多重觸碰牆；(b) Multitouch 的互動牆

3) 對空間設計的啟發

相較於互動地板，互動牆面類型案例在設計概念上更偏向於機能需求的滿足以及使用者主觀上的刻意操作。使用者運用肢體的活動，讓感測用的資訊元件透過使用者的行為了解其意圖，以提供其欲滿足的需求，或許是取得資訊、操作虛擬物件或是與虛擬資訊互動。在設計過程中，資訊視覺化的呈現方式是其重點，然而取得使用者的肢體行動，如：手勢、姿勢、指向或聲音指令等等以分析其意欲所在更是不可或缺；對於設計者而言，如何在空間中置放適合的感測元件取得需要的行為訊息、如何解讀這些行為以及如何與提供視覺回饋的牆面進行連結都是設計過程中必要的設計內容。

類似於互動地板，從互動牆面的案例類型我們也同樣觀察到，資訊元件除了要彼此傳遞訊息進行連結之外，更必須讓資訊元件中的位置資訊能夠精準地與虛擬空間中

的視覺物件產生互動，使用者才會有操作物件的感覺與感受到系統提供的視覺回饋。

3.3.4. 案例四：擴增實境廚房

擴增實境廚房 [Bonanni, 2005]，使用者透過冰箱上的投影得到冰箱內的資訊，從水龍頭上的 LED 燈光顏色判斷水溫，櫥櫃上的燈光以及投影在砧板上的食譜，每一種資訊都分別代表著特定的訊息（圖 3-18）。



圖 3-18. Augmented Reality Kitchen

1) 結構方式

本案例運用了廚房中大多數的空間與設備，將其與資訊元件相結合。在冰箱的內部空間嵌入一個攝影機，再以投影的方式，將攝影機的畫面投影在冰箱的正面上。流理台上則是以投影的方式，將食譜的內容畫面呈現出來，並隨著時間以及透過遠距的熱感應偵測器偵測食物的溫度，判斷目前進行到食譜的哪個階段，改變投影的食譜內容。在水龍頭裡則嵌入了溫度探測器，偵測並判斷水溫，水溫如果過高，則將至於水龍頭出水口附近的紅色發光二極體點亮，水溫如果較低，則點亮同位置的藍色 LED，讓流出來的水改變顏色。櫃子的把手裡埋入綠色的 LED，並在使用者需要找尋特定物品的時候，發光提示，例如食譜進行到某個階段需要鍋子，櫃子的手把便發光提示使用者接下來要用的物品放置的地點（圖 3-19）。



圖 3-19. Augmented Reality Kitchen 裡的各個元件

2) 虛實對應

本案例運用了多種的資訊元件，而這些也都是在智慧空間中經常被使用到的偵測器，包括了攝影機、溫度感應器、時鐘等等，而資訊的視覺化部分則使用了投影機以及多種的 LED 來做呈現。當偵測器偵測到的訊息經由電腦判斷後，再將需要讓使用者知道的訊息以視覺的形式回應到空間裡，這些資訊元件，不論是輸入用的或是輸出用的，都是散佈在整個空間中的各個角落。

3) 對空間設計的啟發

在傳統的空間設計中，空間的構件多是以靜態而固定的形式提供其物理上的特性，並由設計者置放於能夠滿足使用者需求的位置上。然而透過資訊元件的連結與對應，智慧空間中的各個元件除了傳統的物體屬性之外，還可以取得使用者的行為訊息、獲知環境的變動因素；而藉由運算系統，更讓空間得以理解使用者在其間活動的階段，並可以根據不同的活動階段提供相對應的功能以滿足需求。本案例以廚房中的活動為主題，案例中我們可以看到廚房中的各項活動的協助，都不是藉由單一元件來達成，而是透過散佈在空間中的許多元件相互溝通傳遞訊息，藉由合作的形式來共同完成。這類案例的設計上，空間的設計行為已經不只是實體構件的安排與配置，還必須包括對於各個構件之間的角色分工、運作流程與溝通機制，而這些元件之間訊息的流動與轉換，也成為整個設計過程裡至關重要的部份。

在這個案例中，我們發現，資訊元件除了嵌入在空間元件中，也必須要具備空間的資訊，例如可能有來自多個溫度感應器的訊號需要做判斷，智慧空間系統必須要能夠知道是哪一個溫度感應器才能夠做出正確的反應，是來自食物的就改變食譜內容，是來自水龍頭的就點亮水龍頭出水口的 LED 燈泡並根據溫度的高度做各種變化。

在資訊空間類型中，除了資訊元件與空間元件之間的彼此溝通、連接，資訊元件更必須具備空間資訊，以了解每個元件所在的位置並扮演好各自的角色。

由以上互動建築與資訊空間案例的分析，我們可以歸納出智慧空間設計該具備的共同特徵有以下四點：

- 資訊元件與空間元件之間會彼此溝通與傳遞訊息
- 資訊元件具備空間與位置等資訊
- 空間元件成為智慧元件的延伸
- 空間元件與資訊元件相連接與對應



4. 工具案例分析

上一章智慧空間設計案例的分析，我們分析到智慧空間的設計特性是要讓資訊元件具備空間資訊，以及讓空間元件能與資訊元件相連接，而現有的電腦輔助設計系統並無法滿足這樣的需求，因為現有的電腦輔助設計系統沒辦法同時兼顧到空間元件與資訊元件，因此空間元件與資訊元件的結合機制是 CADIS 必須要滿足的一項基本原則。而在進行智慧空間的設計過程也必須要能夠提供視覺上的回饋，包括了空間變化的視覺回饋以及資訊元件彼此連動構成的互動功能的視覺回饋。而各種智慧空間的案例，都是空間設計專業與資訊設計兩種不同專業的結合，因此 CADIS 的目標使用者，將涵蓋空間設計師與資訊設計師，如何同時滿足不熟悉空間設計軟體的資訊設計師以及不習慣資訊設計軟體的空間設計師便是 CADIS 的首要設計原則。從第二章文獻回顧中有關電腦輔助設計系統的發展，我們可以看到空間設計的軟體走向平民化，讓不是空間設計專業背景的業餘設計者，同樣可以透過這種簡化過的空間設計環境來從事簡易的空間規劃；在此同時，資訊設計系統中有關程式設計的軟體，也朝向讓非程式設計背景的設計師或藝術家，同樣可以透過簡單而且視覺化的資訊設計環境，來進行資訊或科技藝術的創作。

從以上幾點，我們可以看到 CADIS 必須要滿足的幾項需求，包括：簡化的空間設計環境、視覺化的資訊設計環境、空間元件與資訊元件的結合機制、空間變化的視覺回饋以及互動功能的視覺回饋，本章接下來便以這些需求為目標，透過對於現有的接近這些需求的工具案例進行分析比較，再得到 CADIS 身為一個工具軟體該具備的設計原則。

4.1. 簡化的空間設計環境

在空間設計系統中，伴隨著電腦的速度越來越快，電腦輔助設計軟體也能處理越來

越多的資訊或提供越來越多的功能，但也因此讓電腦輔助設計軟體的複雜度大幅上升，造成設計者的負擔，現在的空間設計師要能順利使用這些專業的空間設計系統，必須先經歷一段漫長的軟體學習時間，對於空間設計者尚且如此，更遑論資訊設計者了。因為增加的功能，也使得軟體的介面複雜很多，例如很常見的 Autodesk Maya 與同公司的 AutoCAD 系列，軟體介面上就佈滿密密麻麻的功能按鍵以及大量的參數調整畫面，這些複雜的介面與過多的功能增加了設計者的學習負擔(圖 4-1)。

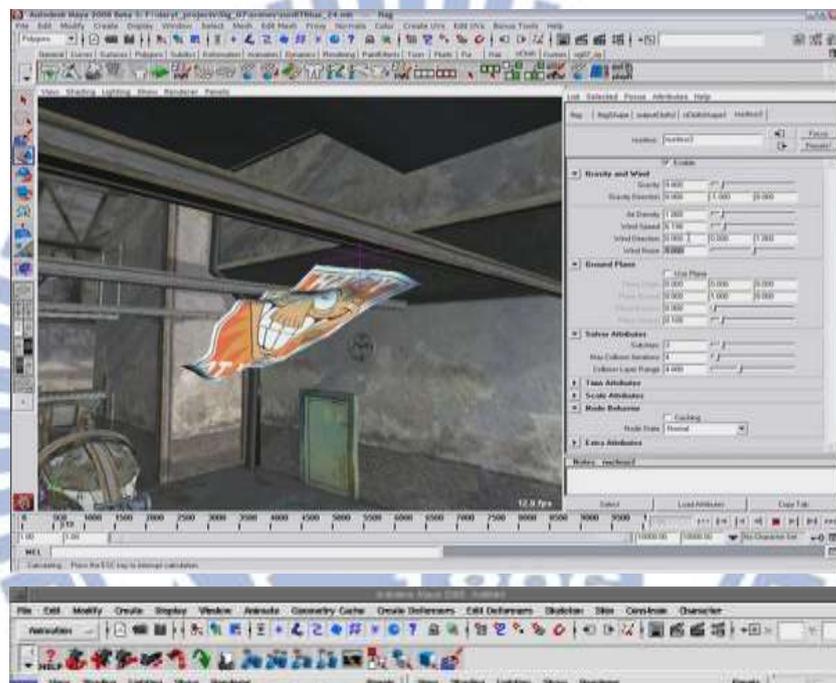


圖 4-1. Maya 2008 複雜的軟體介面

面對越來越複雜的電腦輔助設計軟體所形成的問題，在各個領域也開始提出了各種的解決方式，例如商業公司的 Google SketchUp 運用了最直覺的畫線、拖拉的概念讓使用者很快速地創建出空間的模型，而不需要學習複雜的頂點或三角面系統(圖 4-2a)；Autodesk 也針對原本電腦輔助設計軟體的複雜度過高的問題，提出了專門應用在建築設計領域的 Revit Architecture，讓使用者透過對於空間元素(如：門、窗或牆壁等)進行拖拉的動作或是設定大小等簡單的操作便可建立空間的模型，以減輕空間設計時的負擔(圖 4-2b)。



圖 4-2. (a)Google SketchUp : (b)Revit Architecture

研究領域的卡內基美濃大學的 DDDoolz 將模型的創建指令濃縮到只剩必要的三個：直線、拉昇、與曲線；喬治亞理工大學的 VR Sketchpad [Do, 2001] 則讓使用者只要用簡單的平面繪圖就可以產生空間的模型（圖 4-3）。

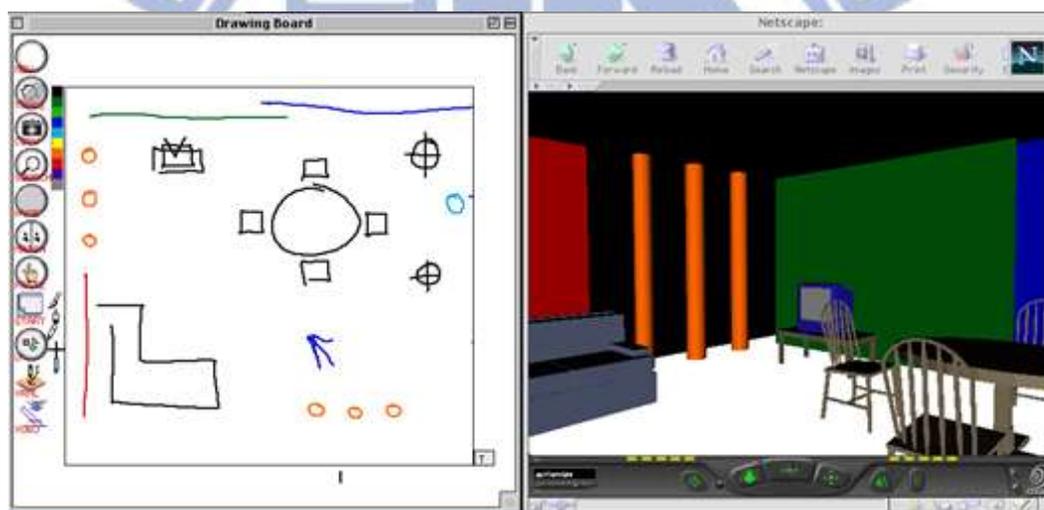


圖 4-3. VR Sketchpad

從上列工具軟體的發展可以看得出來，專業用的空間設計軟體要學習的門檻較高，學習曲線較陡峭，但如果只是要用來表達空間的樣貌，其實並不需要用到這麼複雜的軟體，只要能夠建立直線、拉出厚度，就能夠建構出空間的模型來進行溝通，更加精細的細部或是準確的圖面，仍可以交由專業的空間設計師來延伸這項工作。這些電腦輔助設計軟體，針對的對象並不是需要製作施工圖的專業空間設計師，而是針對著非空間設計專業背景的業餘者而設計，盡量簡化他們的操作，讓這些使用者不需要學習那麼複雜的指令也能夠學會表達或理解要設計的空間的概念。而資訊設

計師也就屬於這一類的人，他們不熟悉空間設計工具，沒有相關的專業背景，所以這種簡化的工具對他們而言是可以降低學習的負擔，讓他們專注於自己的專長上。

因此，要讓設計者能夠快速地學習並運用 CADIS 作為快速雛型設計的電腦輔助系統，將功能與介面簡化是一個很重要的原則。而要讓不熟悉空間設計媒材的資訊設計者也能直覺地進行空間的觀察與理解，簡化過的空間設計系統更是必須的。

4.2. 視覺化的資訊設計環境

資訊設計一般來說都是要透過程式設計的方式將資訊進行處理以及呈現，常用的方式以及工具就是程式語言及其相關的編輯器，例如：C/C++、Java 等。但這類的程式語言對於設計者而言，都需要學習一個程式語言的語法 (syntax)、程式的邏輯與運作的資料流 (data flow) 並非可視的、撰寫程式的過程也是大量的指令文字的輸入，設計過程不但枯燥也很不直覺，要能順利完成與資訊互動的設計仍需要一段學習的陣痛期 (圖 4-4)。



圖 4-4. C++的程式語法需要大量的學習才得以運用

因此，許多針對非資訊背景的設計師所設計的軟體，都希望能夠降低設計過程中必須輸入大量程式碼以及非視覺化的過程所造成的複雜度，例如：Processing 便透過簡化指令的方式，讓原本需要數百行的程式碼降低到只要十分之一就可以達到相同的互動功能；Wiring 與 DBN 同樣也都是針對非資訊背景的設計者所設計的，

只要幾行指令就可以很快速地創建視覺效果或是讓程式能夠回應使用者的輸入提供互動的功能（圖 4-5）。

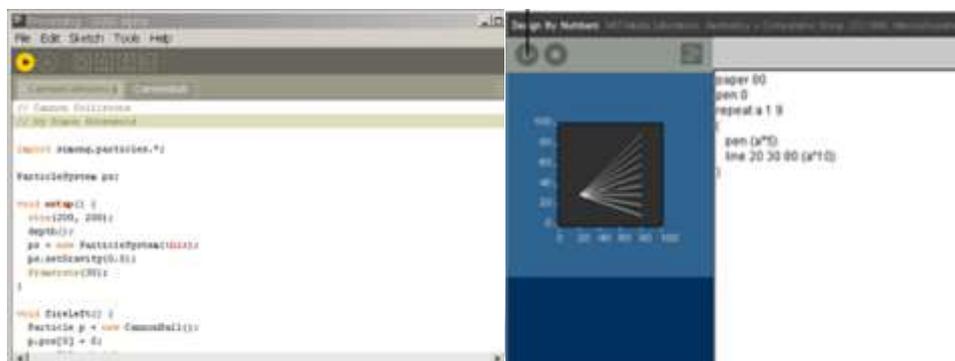


圖 4-5. Processing 與 DBN 都是透過簡單的指令便可在螢幕上繪製圖形或產生互動效果

除了簡化指令之外，也有越來越多的互動資訊設計工具，讓使用者可以透過視覺化的方式進行流程圖的編輯，視覺化地控制資料流與邏輯，例如：MAX/MSP、pure data、MINDSTORMS NXT、vvvv、Virtools 等，而麻省理工學院的 Scratch 更是針對小朋友所設計的程式設計軟體，讓即使沒有受過資訊工程訓練的小朋友也可以透過視覺化的方式撰寫程式（圖 4-6）。



圖 4-6. Scratch 以視覺化的方式呈現邏輯，取代傳統的程式碼概念

對於沒有程式背景的使用者而言，專業的程式設計工具門檻是相當高的，學習過程也枯燥乏味。因此為了讓非程式設計背景的業餘者可以享受到寫程式協助日常生活與工作的樂趣，更充分發揮電腦的運算功能，許多的工具軟體採用了簡化程式指令

的方式來降低門檻。更有許多程式設計的工具軟體，將語言與邏輯視覺化，讓設計者不再需要鍵入一大串的文字內容或是不斷地對照上下行以看懂運算的邏輯，只要看圖就能夠了解，更進一步地將資訊設計的門檻降低，讓設計者以堆疊積木的方式撰寫程式，更加地平易近人。而以視覺思考為主的空間設計師，對於這樣用視覺的方式也更能夠適應與上手。

利用淺顯易懂的概念圖繪製或是設定數值等等方式來取代程式碼的鍵入，這讓許多非資訊工程背景的使用者能更易於接受，除了不再需要記得語法與語言結構之外，所有程式的邏輯都透過了可視的視覺區塊來呈現，程式運作時的資料流動狀態也可透過流程圖表而得以理解。要讓不熟悉程式碼撰寫的空間設計者也能直覺地進行資訊設計，視覺化的資訊設計環境是很重要的設計原則。

4.3. 空間元件與資訊元件的結合機制

對於沒有學習過電腦輔助設計軟體的一般使用者而言，他們並沒有頂點 (point/vertice)、邊線 (edge)、面 (face/triangle/quad) 或是多邊形 (polygon/mesh) 等等概念，在操作一般的電腦輔助設計軟體時就會產生困難。而為了讓一般大眾也能進行室內空間的設計，都會從提供元件庫以及模組化可以自由編排的方式著手，例如：IKEA 的 Homeplanner [Crampton, 2009] 便讓使用者不需要再進行畫點、拉線之類的動作，而是利用他們比較能夠理解的概念，提供平面、門、樓梯、隔板、牆壁等等較為直覺的元件讓一般的使用者也能使用，而他們自己的設計師也同樣透過這套軟體為使用者進行規劃與設計 (圖 4-7)。

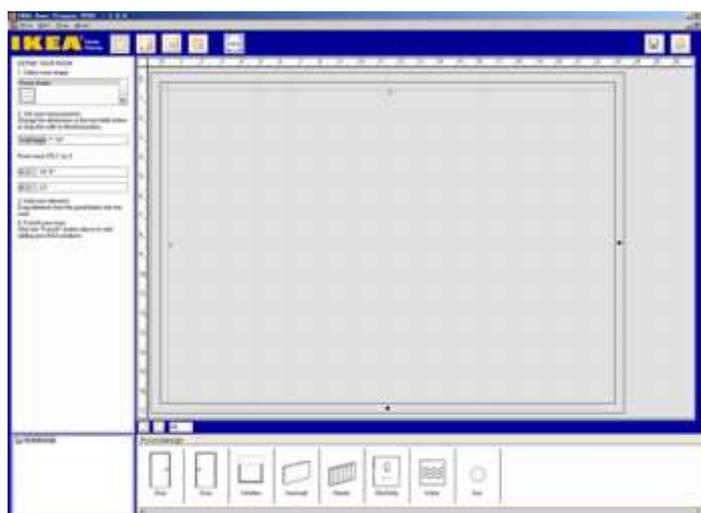


圖 4-7. IKEA 的 Homeplanner 讓一般人與設計師都可以進行室內空間的設計

同樣的概念也被用在針對一般大眾或非專業設計者的 Scenecaster 裡，使用者運用內建的家具自由地變更大小、格局、材質與佈置等進行空間設計（圖 4-8）。



圖 4-8. Scenecaster

ArchiCAD 等針對建築設計師所設計的電腦輔助設計軟體裡也同樣提供了大量的元件庫（圖 4-9），例如：牆 (Wall/Wall End)、門 (Door)、窗 (Window)、轉角 (Corner)、天光 (Skylight)、屋頂 (Roof) 等以及簡單的模型編修方式，讓設計師可以不需要再一點一線的畫出空間中的每一個元件，降低設計負擔也提高了效率。而相同的元件再透過參數的設定，也可以有多種的變化。

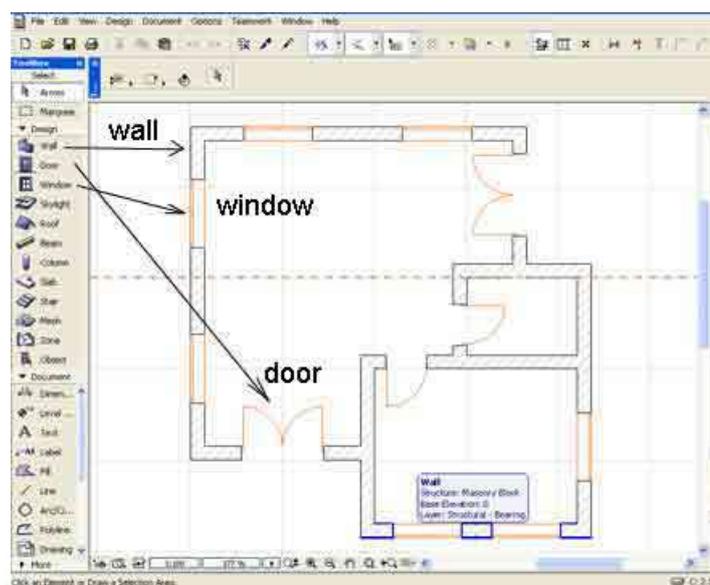


圖 4-9. ArchiCAD 的元件介面

因此，要讓資訊元件與空間元件這兩種截然不同的概念結合於同一個軟體中，將其元件化與模組化是必要的一個工作，這樣才可以將資訊元件與空間元件進行連結，設計者只需要從元件庫裡選擇需要的元件，配置於空間中，再將彼此相互連結就可以達到互動以及傳遞訊息的功能。例如設計者可以從元件庫裡找到一個攝影機的元件，放置在天花板，再將攝影機與牆上的投影螢幕做連結，將攝影機設定一下轉成紅外線模式，接著 CADIS 就自動地將攝影機拍攝的畫面經由投影螢幕投射出來，這樣的例子混合了空間元件與資訊元件，這樣的需求在智慧空間設計中是一個很常見而普遍的例子，卻已經超過了現有的電腦輔助設計軟體所能夠提供的輔助，而這兩種元件之間的無縫連結，就必須要透過元件庫的設計來達成。

而視覺化的資訊設計系統，是透過流程圖的方式進行互動程式設計，也是一個一個的元件型態透過模組化的方式構成。因此，CADIS 必須有元件庫的設計，包括構成實體空間元素的「空間元件」以及構成資訊處理元素的「資訊元件」，以讓設計者可以更輕鬆地建構出想像中的互動功能。

4.4. 空間變化的視覺回饋

傳統上，所有的電腦輔助設計軟體在空間設計階段，都無法看見場景或所設計的空間的完整面貌，而必須藉由「彩現」(render)的指令將空間中的物體轉換成投影在螢幕上的畫面、計算物體的表面材質與紋理、再加入複雜的光線模擬計算出空間中的光線對於物體所造成的色彩變化。這個彩現的過程，視場景的差異，每一個畫面可能需時從數分鐘到數小時不等，而要呈現一個空間互動的過程，大約需要數十個畫面，耗費在彩現的時間相當的長。

然而資訊設計的設計過程，則是需要修改或是撰寫新的程式邏輯後，反覆地在撰寫程式碼、執行與偵錯三個階段間循環，以測試互動的效果是否正確。程式碼撰寫完畢進入執行階段，要是發現了問題就要進入偵錯階段找尋錯誤的點，然後要再回到撰寫程式碼將錯誤修正，再進入執行階段，週而復始。一旦將兩者結合，每次加入或修改互動的功能後，就要等上相當長一段時間才能觀察到智慧空間呈現的效果，而一旦有錯誤發生或是需要增添新的功能，再次修改後又要重新等待彩現的時間，是相當耗費時間的。

單純的空間靜態圖像的視覺回饋就需要數小時不等的時間了，會產生變化的智慧空間要提供視覺回饋，以傳統的彩現方式就會把時間拉長到數小時的幾百倍數，整個設計階段會變成不斷地在等彩現，這是非常沒有效率的作法。

而在電腦圖學與遊戲相關的發展下，軟體上有渲染器(shader)技術的應用、計算光線漫射相關的演算法如：柔邊投影(Soft Shadow)、環境光漫射(Ambient Occlusion)、全域照明(Global Illumination)等新的進展，硬體上則有繪圖處理器(GPU, Graphic Processor Unit)的新概念，軟硬體的搭配在即時上已經可以處理相當程度的場景計算，可以馬上呈現相當接近彩現計算效果的光影(圖 4-10)，而不需等待完整的彩現。



圖 4-10. 即時計算的新光影效果（左）以往的即時運算效果（右）

而設計者要檢視智慧空間的運作，最重要的是在空間與使用者之間的互動機能，實際上並不需要像 Maya 或 MAX 之類的電腦輔助設計軟體計算出百分之百準確的場景彩現圖，所以只需要夠擬真的畫面便足以提供設計者對空間的概念與理解，在空間與資訊互動皆設計完仍可以匯入到其他的電腦輔助設計軟體彩現出幾可亂真的影像。

因此，CADIS 要透過繪圖處理器與渲染器的運用，能即時地模擬出空間的光影以及動態地呈現資訊與使用者之間的互動效果。這也是 William Mitchell 所說的“未來的模擬工具除了渲染逼真的畫面或是能源的效能模擬之外，還必須能夠「執行」設計的建築物因應各種不同外在條件與使用者需求的情境下所作的變化，就像在動畫、遊戲或是機械設計工具中所看到的那樣”。

4.5. 互動功能的視覺回饋

為了讓空間能夠與使用者互動，空間中通常會隱藏著各種的感應器與微處理器（MCU, Micro Control Unit），例如 Jeng 提出的動作模組就是透過微處理器進行訊號的處理與分析，而這些實體的智慧元件中便存在著邏輯程式，這些程式的

撰寫方式雖與一般的程式語言沒有太大的差異，只是在最後撰寫完成的程式，必須透過「燒錄」的方式將處理的邏輯燒錄進微處理器中。

微處理器既然是程式的負載元件，流程上便有跟撰寫程式相同的執行與偵錯的過程，而這些執行與偵錯的過程就需要在撰寫程式後、燒錄至微處理器中並接上各種的感應器後才能進行，同樣也是耗時耗工的工作，也因此有了模擬器 (simulator) 的概念，讓設計者可以在設計階段進行執行與偵錯，以預先模擬出可能的反應，確認沒問題後再進行燒錄與完整的測試。例如：柏林的真實：聯合設計工作室 (reality:united GmbH) 在奧地利的格拉茲城 (Graz, Austria) 進行的 Kunsthaus Graz 案便透過了模擬器的運用，大大降低了反覆偵錯的複雜工作，尤其在這種大型空間中會運用到非常大量的實體智慧元件，透過模擬進行測試更是必須的。另外在互動設計領域很常用的 Arduino 版，同樣也有模擬器 Fritzing (圖 4-11) 可以在燒錄之前先行模擬與各種感應器之間的連結，避免了時間的耗費以及多次燒錄造成的實體元件耗損。

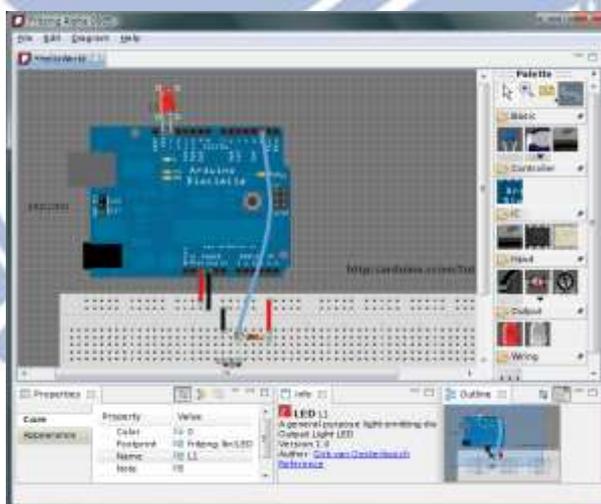


圖 4-11. Fritzing 可以先行模擬 Arduino 板子與各種電子零件之間的溝通與訊號傳遞

而智慧空間中，最多的工作就是資訊元件與空間元件之間以及資訊元件之間的大量訊息傳遞，這些訊息就影響了最後互動的呈現結果。因此，CADIS 要能夠提供互動效果的視覺回饋，除了要真實地計算元件與元件之間的訊息，更要模擬資訊元件的訊號如何地控制空間元件，進而構成一連串的反應以回應使用者的動作。

所以要能夠提供互動功能的視覺回饋，讓設計者知道自己的設計是否正確，元件之間的連動是否如自己所想像，提供元件動作訊號與訊息傳遞的模擬便是 CADIS 一個必要的設計原則。

在許多的智慧空間案例中，仍是以個人電腦取代微處理器為主要的運算單元，如：交通大學建築研究所的明日建築展、Lee 的智慧廚房、Streitz 的軟房間、瑞士聯邦理工學院的計畫等等。在這樣的空間中，如何隱藏電腦以及各種線路的排佈都是相當重要的工作項目，因此在 CADIS 中，除了要能夠模擬出家具等實體元件外，也必須要有這些智慧元件的實體，讓設計者得以在設計階段便進行思考與佈局，並據以產生設計圖面，在後來的施作階段便可以參考並更有效率地進行。

4.6. 討論

因此，從以上幾節，為了讓空間設計者能快速地進行資訊系統的設計，新設計媒材 CADIS 必須使用簡化而較容易理解的資訊設計方式，也就是視覺化程式設計，讓他們可以用視覺化的思考模式進行資訊互動部分的設計。而要讓不熟悉空間設計媒材的資訊設計者能直覺地進行空間的觀察與理解，新設計媒材也必須使用簡化過的空間設計系統，讓他們可以對於空間的結構與樣貌更有概念，了解製作的畫面實際運作後會呈現在空間中的什麼位置。而要達到讓空間元件與資訊元件相互連結，CADIS 必須要將所有的元件模組化，方便設計者取用與串接。再者，在設計過程的最後階段還必須要能夠進行能夠反映使用者動態與互動效果的視覺模擬以得到最接近實際建造完成的空間感受，透過即時的光影模擬與動作變化提供設計者有關空間變化的視覺回饋。再加上資訊設計的功能也必須要能夠在 CADIS 環境中提供視覺的回饋，讓設計者透過視覺化程式環境撰寫程式後，可以透過這樣的回饋，觀看元件之間的訊息傳遞，以及資訊元件與空間元件之間的控制效果，了解自己是否設計出想要的互動機制。

綜合以上工具軟體的案例分析與上一章的應用案例分析，我們歸納出 CADIS 必須具備的幾項特徵。

- 資訊元件與空間元件之間會彼此溝通與傳遞訊息
- 資訊元件具備空間與位置等資訊
- 空間元件成為資訊元件的延伸
- 簡化的空間設計環境
- 視覺化的資訊設計環境
- 元件與模組化
- 空間變化的即時視覺模擬
- 元件間互動效果的模擬



5. 系統架構與實作

本章將闡明 CADIS 系統的設計與建構過程，首先透過由應用案例以及工具案例的分析所得到的設計原則，建立一個可以滿足這些設計原則的系統架構，這個系統架構便是據以製作 CADIS 的骨幹。接下來透過這個架構，我們運用 Processing 程式語言來製作並撰寫出 CADIS 的基本功能，包括其元件庫系統、操作介面的設計以及整個系統的實作內容。在最後一節則是操作 CADIS 實際地設計出幾種常見的測試案例作為系統的驗證與新設計流程的測試。

5.1. 系統架構

由智慧空間應用案例的分析以及設計工具案例的分析，我們了解了要設計出 CADIS 所需要滿足的幾項設計原則，包括了：在 CADIS 中，資訊元件與空間元件之間要能夠彼此溝通與傳遞訊息、資訊元件需具備空間資訊、空間元件為資訊元件在空間中的延伸、另外也需要簡化過的空間設計環境、視覺化的資訊設計環境、並建立元件庫將元件模組化、CADIS 也必須要能夠提供模擬功能讓設計者用來觀察空間變化的即時視覺效果以及元件間的互動效果。

根據 Jeng (2007) 的研究，他們在具互動機能的空間的設計過程中，將整個設計分為三個主要的概念模組來進行：動作模組 (action modules)、使用者介面模組 (user interface modules) 以及實體運算模組 (physical computing modules)。在他的分類中，動作模組是負責處理空間的人類的行為，例如：開門、推、碰等等動作；使用者介面模組用以表示空間中的所有元件，如：門、牆、地板、天花板、家具等等；實體運算模組是用來偵測的感應器以及輸出的裝置，如：壓力感應器、伺服馬達等。Ballagas 也提出裝置的互動架構，用按鈕、滑桿、麥克風等基本裝置，在將這些裝置組合成元件，每個元件皆對應到在互動系統中執行的一

個軟體程式 [Ballagas et al., 2003]，並在後續提出各元件之間如何透過事件 (put, get) 彼此連結與互動 [Ballagas, Szybalski & Fox, 2004]。Ponnekanti 也提出遍佈式運算環境的架構，將系統架構分為使用者動作、視覺介面、後端服務 [Ponnekanti et al., 2004]。

綜合以上研究可以得知，從使用者的行為或動作作為輸入、感測器接受使用者產生的訊息、後端運算裝置接受來自感測器的訊息、產生給使用者的回應到空間的實體元件中便是構成在智慧空間中整個互動的流程。要達到這樣互動的流程，空間元件與智慧元件的建立與設計便是第一步驟，CADIS 以「元件庫」的形式將兩種元件納入進行管理，設計感測器及與實體連結的裝置稱為資訊元件，設計空間的實體元件便是空間元件。CADIS 是一個資訊設計者與空間設計者共同使用的環境，因此也必須提供簡化過的「空間設計模式」與視覺化的「資訊設計模式」。資訊設計環境讓設計者透過視覺化的程式撰寫方式設計空間中各種元件的互動機制，空間設計環境作為設計時的主要呈現環境，以及最後用以模擬互動結果的「模擬模式」，整個 CADIS 便是在資訊設計模式、空間設計模式以及模擬模式三種模式之間切換，並透過元件庫中的資訊元件以及空間元件進行各種設計或執行互動模擬，CADIS 的架構圖如圖 5-1 所示。



圖 5-1. CADIS 系統架構圖

根據上述的架構圖，設計者可在設計環境與模擬環境中分別進行設計行為與模擬觀察，在整合設計環境中，操作上類似於目前的空間設計媒材，但操作的對象除了一般的量體等空間元件之外，還有具備運算能力與輸出入能力的資訊元件，設計者透過繪製關聯架構圖表將資訊元件、空間元件與使用者資料相互連結以產生各種互動，互動的結果在切換至模擬環境時便會被自動運算以及呈現。

這樣的系統架構，可以滿足第三章與第四章分析所得到的八項 CADIS 的設計原則，透過關連架構圖表，讓空間元件與資訊元件可以相互連結與傳遞資料，資訊元件透過這個連結與模擬模式將其功能延伸到空間元件中，資訊訊號可以用來控制空間的元件。而我們在 CADIS 中所建立的元件庫裡所有的元件都將記錄下空間的性質與資料，讓資訊元件具備了空間資訊，也讓元件與模組化得以實現。關聯架構圖作為連結元件的功能，同時也是一個視覺化的程式設計環境，這些元件的連結再透過模擬環境結合了使用者資料模擬出各自之間的連動關係，將互動的功能表現出來，整合設計環境是一個簡化過的空間設計環境，讓使用者可以簡易地操作畫面觀看整個空間，而整合設計環境與模擬環境相結合，提供了擬真的空間變化模擬，系統架構與設計原則之間的關聯性如圖 5-2 所示。

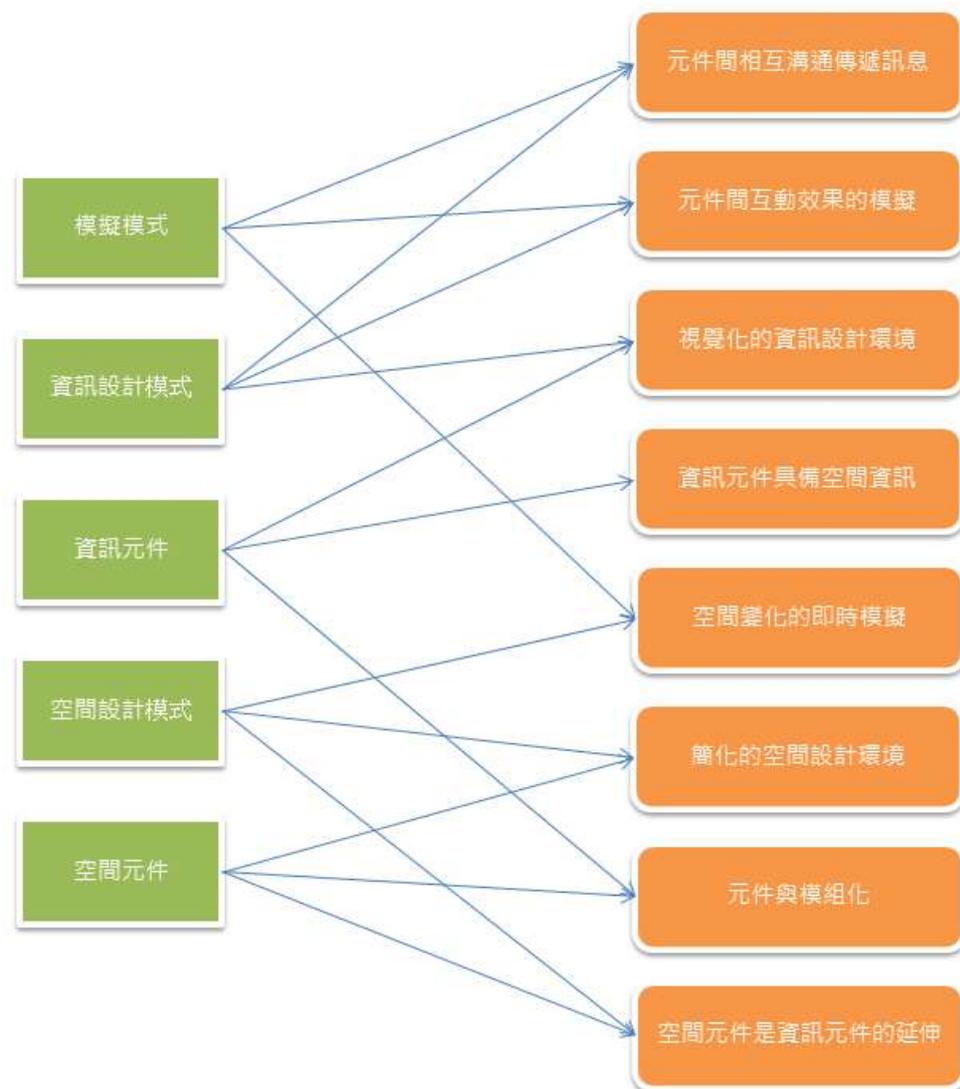


圖 5-2. 系統架構與設計原則之關聯性

需要建立的架構為兩個，一為管理資訊元件與空間元件兩種元件的元件庫，以及能夠在三種模式（資訊設計、空間設計以及模擬）中進行切換的整合式設計環境，我們在後續的兩小節中分別就以元件庫及整合式設計環境的架構加以討論。

5.1.1.1. 元件庫

CADIS 中需要建立的元件庫有兩個類型，一種為涵蓋感測器、運算處理與視覺回饋的資訊元件，一種則為組構出空間實體，涵蓋各種平面、立面、桌椅等的空間元

件，因此採用物件的方式會方便我們進行建置。每一個物件都會具備長、寬、高、位置等空間中的物理資訊，資訊元件也需要建立位置以及面向的空間資訊，因此在 CADIS 裡必須先建立基本的資料結構，接著再根據這樣的資料結構建立兩個元件庫。

首先我們先定義出資料結構所需要的變數類型，以常見的變數類型來做分類，為了讓系統簡單化，我們將這些變數類型精簡，例如在數字類型就不像一般程式語言分為整數及帶小數的浮點數，而是一律只使用浮點數。在各種智慧空間的應用案例中，圖像以及文字類型是很常見的視覺回饋型態，在 CADIS 中也列為兩種重要的資料結構。而從互動建築的案例中我們可以看到許多的案例都是透過單元的結構設計來組成大量的主體，有如以單一的像素點來構成平面的圖像一般，有些建築的立面或平面就是透過單元的結構來組成，因此我們需要一個對應到這種類型設計的資料結構，程式語言中是以無索引的集合 (Set) 以及有索引值的陣列 (Array) 作為代表，我們在 CADIS 中則一律以集合稱之以降低非程式設計背景者的學習負擔。另外我們訂立一種特殊類型稱為函式表 (Function Table)，用來代表其他不定類型，例如元件中包含的另一個元件等。以此建立 CADIS 的資料結構類型，共有數位、數字、文字、集合、向量、圖像、函式表。

- 數位 (Digital)：真或假，一或零，是或否這樣的數位型態資料
- 函式表 (Function Table)：為文字的集合，數量不定
- 數字 (Number)：整數與浮點數
- 圖像 (Picture)：為數字的集合，數量不定
- 集合 (Set)：可為任意類型但限定為相同變數類型的變數群
- 文字 (Text)：文字與字串
- 向量 (Vector)：為三組數字的集合

空間元件

許多電腦輔助設計軟體為了簡化使用上的複雜度以及針對特定的設計需求，跳脫以頂點 (vertex)、線 (polyline/b-spline)、面 (NURBS/facet) 之類的電腦繪圖概念，而改以操作量體或元件 (如：窗、門、樓梯...等) 的方式來讓設計者能快速

地學習與運用，例如：3D Studio VIZ， SketchUp 等。

CADIS 的空間元件中，依循類似的概念，除了包括了各種量體，如：牆、桌、天花板等，還應包括整個空間環境的虛體。而這些量體，為求簡化，我們將其抽象化而得到立面與平面，立面包括牆、窗、門及等，平面包括了天花板、桌、椅及地面等等；立面與平面再進一步抽象化，其實都可歸納為一樣是量體形式的空間元件，只是面向上的差異。每個量體空間元件除了外觀的模型資料外，也包括了在空間中的位置、材質顏色等等屬性，虛體空間元件則有著溫度、燈光、時間等屬性。而我們也將可影響空間呈現的元件歸類為空間元件，例如燈具、投影機及螢幕等。

空間元件的分析將著重在屬性的取得與描述，例如：量體空間元件就具備尺寸、位置、面向、材質或顏色等屬性，而環境的空間元件可偵測到溫度、亮度、時間、分貝或天氣等屬性。空間元件的另一個特性，就在於可嵌入各種元件，例如量體的空間元件可嵌入燈具的空間元件，開關的資訊元件以及攝影機的資訊元件，因此我們為量體空間元件定義了嵌入元件這樣的屬性，用來記錄這個空間元件中嵌入了哪些元件。CADIS 涵蓋的空間元件與屬性如下表所列（表 5-1）。

元件類型	屬性	資料型態
環境 Environment	溫度	數字
	亮度	數字
	時間	函式表
	分貝	數字
	天氣	文字
量體 Solid	位置	向量
	面向	向量
	尺寸	向量
	材質	文字
	嵌入元件	函式表
燈具 Light	顏色	向量
	亮度	數字
	面向	向量
	位置	向量
	耗電量	數字
螢幕 Screen	位置	向量
	尺寸	向量
	面向	向量
	影像	圖像
	耗電量	數字
投影 Projector	位置	向量
	尺寸	向量
	放大率	數字
	面向	向量
	影像	圖像
	耗電量	數字
空調 AC	強度	數字
	位置	向量
	尺寸	向量
	面向	向量

表 5-1. 空間元件庫屬性表

資訊元件

在許多的資訊設計系統中（如：OrCAD、Flash...等等）都已運用資訊元件庫的概

念讓設計者來進行設計，但是這些元件庫的方向是朝向盡可能的詳細與完整，需經過大量的學習之後才能理解各元件內容與使用方式，例如：看到元件庫中有一顆 74ALS01 的元件，若沒有相關背景便無法理解這個元件的意義。

因此在 CADIS 中，本研究將被運用在智慧空間中的資訊元件以其運作的概念進行抽象化後分出幾種類型：被動式的數位感測器、類比感測器及座標指示器、主動式的攝影機、測距儀等，例如數位感測器就包括了各種的開關、按鈕，類比感測器包括了旋鈕、調節閥或是震動感應器等，是被動地等待代理人的觸發才有訊號產生；主動式的如測距儀可以自己接收來自代理人的位置屬性，並將結果計算後以數字的形式傳遞出去，而無論採用的技術為紅外線測距或是超音波測距，運作的概念在抽象上是相同的東西。另外例如文字輸入器就包括了鍵盤、RFID 感應器，都是透過某種形式取得文字內容，因此可分類為相同的概念與類型。

資訊元件的分析著重在各種感測器的歸類以及抽象化，例如：滑鼠與搖桿透過抽象化可被歸類為被動感測器中的相對座標指示器，手寫板以及感應陣列則在抽象化後歸類為絕對座標指示器，透過抽象化的過程讓複雜的各種硬體被歸類為幾種主要的形式，以降低使用者認知上的負擔。CADIS 所涵蓋的資訊元件如下表所示（表 5-2）。

元件類型	屬性	資料型態
數位感測器 DigitalSensor	開關	數位
	位置	向量
	尺寸	向量
類比感測器 AnalogSensor	強度	數字
	位置	向量
	尺寸	向量
攝影機 Camera	影像	圖像
	位置	向量
	尺寸	向量
	面向	向量
測距儀 Ranger	距離	數字
	位置	向量
	尺寸	向量
	面向	向量
座標指示器 Pointer	座標	向量
	位置	向量
	尺寸	向量
文字輸入 Text	文字	文字
	位置	向量
語音輸入 Voice	文字	文字
	位置	向量
	音訊	圖像

表 5-2. 資訊元件庫屬性表

5.1.2. 整合式設計環境

CADIS 具備的三種模式，分別是資訊設計模式、空間設計模式以及模擬模式，由於設計上的差異性，使用者在資訊設計模式中是以視覺化的方式將各種資訊元件以及空間元件結合起來，設定傳遞的訊息與互動的方式；在空間設計模式中則是著重在觀察空間的造型配置與重新安排各種元件的位置，而在模擬模式下則是要以第一人

稱的方式來觀看各種物件模擬出來的結果，三種模式如果同時操作會讓整個系統變得非常複雜，因此 CADIS 必須是在這三種模式之下進行切換，一次僅進行一種模式的工作。資訊設計模式必須具備相關的處理能力，空間設計模式較為單純就是用來顯示模型，模擬模式則另外需要建立使用者資料以供模擬之用。

資訊設計模式

資訊設計模式中運用的架構關聯圖表是完成前述各項元件整合的最關鍵部份，除了空間中可見的可視元件（空間元件跟資訊元件）之外，也需要具備將各種資料進行各種處理的不可視元件，我們稱為邏輯元件。我們以常見的幾種運算模式作為邏輯元件，包括了進行四則運算的數值運算、回答是與否的邏輯判斷、進行重複工作的數值循環與邏輯循環、用來將集合進行拆分用的分拆元件、以及記錄常數用的固定內容元件。如同視覺化的資訊設計軟體，使用者透過對每一個元件的連接或是設定來進行互動程式的撰寫。

設計者在資訊設計模式中運用各種元件，包括資訊元件、空間元件以及邏輯元件，透過相互連結與設定數值的方式讓元件之間產生連動，也可以運用基本的邏輯區塊進行各種狀態的設計，例如：將空間元件的溫度連結設定為 -0.1 的循環邏輯便創造出溫度逐漸下降的環境，若再配合空間元件中的時間與判斷的邏輯區塊的相連接便可創造出空間環境的冬夏之別。而在資訊設計模式中所使用的邏輯元件，最主要的特性就在於具備了資料的方向，也就是輸入與輸出，輸入取得資料後，經過各種元件類型的處理，再由輸出將資料送出。在資訊設計模式中可使用的元件如下表所示（表 5-3）。

元件類別	屬性	資料型態	方向
數值計算 Arithmetic	變數 A	所有資料型態	輸入
	變數 B	所有資料型態	輸入
	計算結果	所有資料型態	輸出
邏輯判斷 TrueFalse	變數 A	所有資料型態	輸入
	變數 B	所有資料型態	輸入
	判斷結果	數位	輸出
數值循環 Accumulate	應變數	數字	輸入
	變數	數字	輸入
	結束	數位	輸出
邏輯循環 Loop	變數 A	所有資料型態	輸入
	變數 B	所有資料型態	輸入
	結束	數位	輸出
分拆 Break	來源	集合	輸入
	分拆數	集合	輸入
	輸出	所有資料形態	輸出
集合分析 Analyze	集合	集合	輸入
	數量	數字	輸出
固定內容 Fixed	常數	所有資料型態	輸出

表 5-3. 資訊設計環境中的邏輯元件

資訊設計模式透過這些邏輯元件，以及載入空間設計環境中存在的空間元件以及資訊元件，使用者再利用設定參數的方式，讓這些元件之間彼此連接，並將連接的結果記錄成架構關聯圖表。資訊設計環境的子系統架構就包括了元件載入、關聯架構圖表管理、以及一個整合操作這些功能的介面，稱為互動編輯器（Interaction Editor），資訊設計模式的子系統架構如下圖所示（圖 5-3）。



圖 5-3. 資訊設計模式的子系統架構

空間設計模式

空間設計模式提供設計者安排各種可視元件 (包括資訊元件、空間元件) 以檢視設計空間的整體狀態，介面與操作方式與多數的電腦輔助設計軟體相雷同，但需要將畫面與功能盡可能簡化。架構上除了基本的空間模型載入之外，由於需要即時的視覺顯示，CADIS 也運用了渲染器的技術，讓空間模型可以即時性地產生動態的視覺效果。在空間設計模式的部份，我們所設計的系統架構 (圖 5-4)，是以一個虛擬 3D 空間負責顯示及載入空間模型，再結合場景管理機制 (Scene Manager)，用來控制載入系統的空間模型的旋轉、平移或放大縮小等使用者觀看模式，以及各種空間元件的新增或刪除。

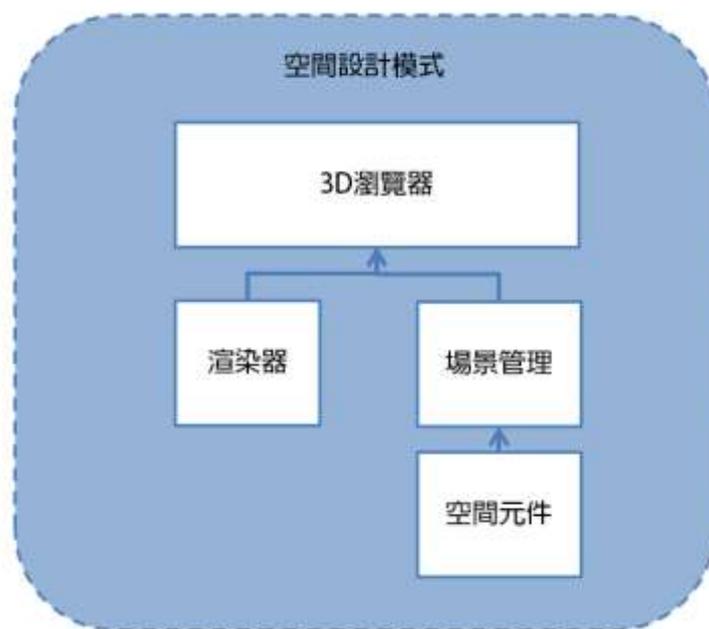


圖 5-4. 空間設計模式的子系統架構

模擬環境

不論傳統空間或是新的空間，空間的主要服務對象與使用者始終都是人類。雖然在其他的研究中有關注到以使用者動作作為輸入的概念，但並沒有涵蓋到所有的可能性，例如許多的感測器可以感測距離、人體體溫、紅外線遮斷等都可以作為輸入觸發互動。在模擬環境中，空間也需要針對使用者的動態而有所回應，因此 CADIS 裡須建立各種代表使用者的代理人 (avatar)，每個代理人以參數的方式設定各自的身高、體重、姿勢、個人識別等等外在屬性，以及各種動作如：按壓、揮動等等。各種屬性與動作都可透過關聯架構圖表與資訊元件或空間元件相連接以產生互動。在模擬環境下，系統便可透過架構關聯圖表讀取各元件間的互動行為，並依據人因知識中的規範以視覺化的方式呈現結果。使用者資料的分析將著重在人類外型資訊的分類，以及各種可被感測器擷取到的動作行為或資訊，以期在系統中建立人因知識的資料庫。在 CADIS 中，我們將使用者資訊歸納出以下類型（表 5-4）。

類型	因數屬性	資料型態
代理人 Avatar	身高	數字
	體重	數字
	識別編號	數字
	姿勢	函式表
	動作	函式表
	照片	圖像
	位置	向量
	面向	向量
	稱呼	文字

表 5-4. 代理人資料表

模擬模式是實際模擬整個空間與代理人互動的過程。設計者在整合設計環境中擺放代理人時可以設定該代理人是自動狀態或是可控制狀態，設定為自動狀態的代理人在模擬模式下會自己移動而空間也會隨著與之產生互動，可控制狀態的代理人則由設計者自行操作以實際感受互動的經驗。模擬模式在即時運算模擬的過程中，設計者可以在環境中自由地移動以觀察各個空間與代理人互動的情形以及資訊元件中各項資訊流動的過程。這部份的分析將以如何最擬真地呈現畫面，或是如何將設計好的資料輸出至具備優異彩現能力的軟體進行模擬。因此在模擬模式中，CADIS 除了標準的 3D 瀏覽環境之外，也要具備渲染器與場景管理等空間機能以及關聯圖來執行互動機能，模擬模式的子系統架構如下圖（圖 5-5）。

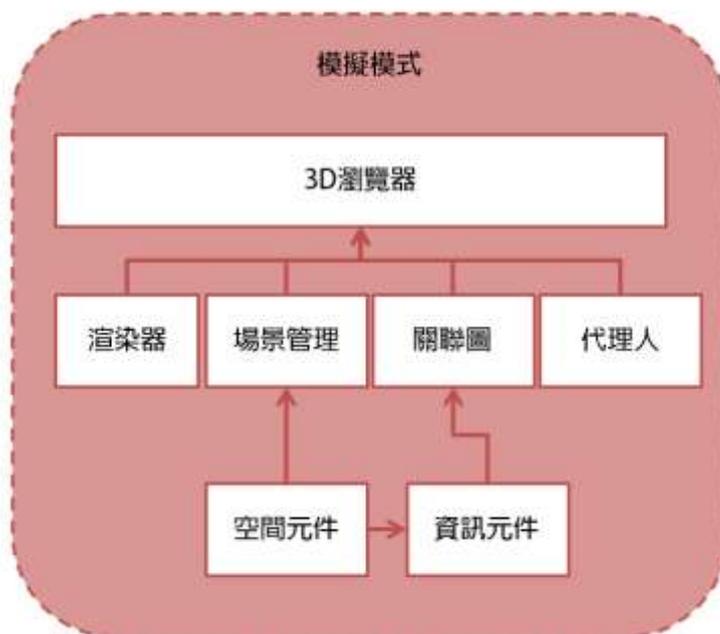


圖 5-5. 模擬模式的子系統架構

5.2. 系統實作

建立了 CADIS 的系統架構後，要透過實作的方式讓系統被建造出來並實際地運作，在後續的章節便可以透過這個系統來進行驗證測試案例的設計流程。由於 CADIS 是一種工具軟體，因此需要選擇一個程式語言來實作，我們選擇的語言是 Processing，並搭配不同的函式庫來實作出 CADIS 的各項功能，例如以 ProScene 作為場景管理器。在渲染器部分，由於 Processing 本身支援 GLSL (OpenGL Shading Language)，因此我們運用 GLSL 作為主要的渲染器語言。

5.2.1. 實作語言：Processing

Processing 是一種根基於 Java 程式語言，並將 Java 簡化過後專精在視覺與多媒體處理的高階語言，在其軟體系統的底層是透過 Java 虛擬機器 (JVM, Java Virtual Machine) 作為主要的編譯與解譯工作。Processing 具備 Java 語言的相關特色，而這些特色正是我們在實作 CADIS 時所需要的，也是 CADIS 運用 Processing 作為實作語言的原因。

物件導向

如同 Java 是一種物件導向的語言，Processing 也支援物件的機制，這在我們要建立 CADIS 的元件庫時便看得出物件導向帶來的幫助。物件導向語言將程式的內容切割為一個一個的物件，每一個物件可以涵蓋兩種性質，一種稱為屬性 (property)，一種稱為方法 (method)，若以人作為物件當做例子，身高體重等性質便是人這個物件的屬性，而跑跳說話等動作就是人的方法。在 CADIS 中，空間元件需要具備尺寸、位置等空間的屬性，我們可以透過物件的屬性來建立這些資料，而資訊元件除了這些屬性資料之外，還需要有運作或是執行等動作，這些動作就可以用方法來加以實作出來。透過 Processing 以及其物件的特色，我們可以先建立一個代表空間元件的物件，賦予其位置及尺寸等空間資料的屬性，再利用物件導向中繼承 (Inheritance) 的機制，繼承這個空間元件物件的所有屬性並建立一個新的物件用來代表資訊元件，這個資訊元件就將具備來自母代的空間資料屬性，我們也可以在這個新的物件上另外建構跟資訊互動相關的專用屬性，資訊元件便可以透過這樣的方式被賦與空間資訊並與空間元件相結合。而在物件導向語言的特性中，另一個 CADIS 需要的語言特性是多型 (Polymorphism)，多型代表的是同樣的物件用同一個 method，卻可以接受各種不同的資料結構，例如一個稱為 Logic 的物件，有一個名為 Logic.equal() 的方法，Logic.equal(13, 24) 可以用來比較兩個數字，Logic.equal("a", "b") 卻可以用來比較兩個文字。在傳統的語言中，函式對於導入的變數是固定資料結構的，傳入不同的資料結構可能會造成電腦的記憶體配置錯誤而造成當機，但在物件導向中，透過多型的方式就可以用同一個方法的名稱，用來進行不同資料結構的運算。CADIS 簡化了資料結構以及程式設計環境，因此每一個元件執行時會輸入的資料結構就很不明確，可能這次是數字，下次是文字，如果每一次都要驗證對於效能會有很大的負面影響，而因應每一種不同的資料結構來建立不同的方法又會讓元件變得多，同樣的邏輯元件，處理文字要實作一個，處理數字又要再實作一個，大量類似的物件會讓系統變得複雜，使用者也更容易混亂，因此透過物件導向的多型特色，便可以將這些動作隱藏起來，使用者只要知道哪些元件有哪些方法即可，不需要注意資料結構的對應問題，讓 CADIS 變得更單純簡易。

功能實作

將系統切割為幾個子系統：元件庫、空間設計模式、資訊設計模式以及模擬模式。每一個子系統都是各自實作完成，再透過元件庫連結空間設計模式子系統以及資訊設計模式子系統，模擬模式則是將資訊設計模式中定義好的函式做執行的動作，並將執行的結果反應到空間設計模式中的元件，作為互動效果的展現。

5.2.2. 元件庫

在採用了 Processing 作為實作的語言後，我們必須先建立最基本的物件，在 CADIS 中，最基本的元件就是空間元件，空間元件只存在屬性，雖然可以被致動器元件改變這些屬性，但空間元件本身並沒有觸發的機制，也就是說不需要互動的方法，因此我們建立的第一個基本元件就是空間元件 (SpatialObject)。空間元件除了長寬高等基本的物理屬性之外，另外也需要包含位置、面向、唯一識別號碼 (unique identity) 等等基本資料。這些空間物件，在空間設計模式下會需要被繪製出來，以 Processing 的語言規則，我們需為其建立一個 draw 函式，並在每一次繪圖的時候呼叫該函式。draw() 函式的內容就是將空間元件代表的模型繪製在空間裡，另外也包括了用來設定元件尺寸的 setSize() 以及設定位置的 setPosition()。而使用者在空間設計模式下新增物件時，會需要透過滑鼠來進行操控與點擊，所以這個骨幹的空間物件也放入了兩個動作偵測的函式，分別是 mouseClicked() 用來偵測滑鼠的點擊動作以及 mouseDragged() 用來偵測滑鼠的拖放動作，因為這兩種動作是在電腦輔助環境中最常對物件做的滑鼠指令。接下來就是利用繼承的方式，讓資訊元件繼承空間元件的物件資料，並重新定義我們需要的 draw() 方法，利用 execute() 函式讓資訊元件可以執行設定好的程式，或是由其他的資訊元件透過呼叫該資訊元件的 trigger() 函式來加以觸發。資訊元件在資訊設計模式下必須要能被拖拉移動，因此要重新定義新的 mouseDragged() 函式。而透過繼承方式得來的資訊元件，除了重新定義的函式之外，也就具備了原先空間元件裡已有的屬性，包括了長寬高 w/h/d、方向的向量 pos，唯一識別碼 uid 以及名稱，而空間元件的 setPosition() 等函式也有著相同的行為模式，並且我們可以透過物件導向中的重載方式改寫同名的函式，由此我

們所建立的最基本的兩種物件如下圖所示（圖 5-6）。

```

public class SpatialObject {
    float w, h, d;
    PVector pos;
    int uid;
    String name;

    SpatialObject(Scene _sc, int _id, String _name) {
        super(_sc);
        id = _id;
        name = _name;
        setSize();
        setPosition();
        setGrabsMouseThreshold(20);
    }

    public void draw() {
        pushMatrix();
        applyTransformation();
        if (grabsMouse()) {
            mouseEntered();
        } else {
            mouseLeaved();
        }
        box(w,h,d);
        popMatrix();
    }
    public void setSize() {
    }
    public void setSize(float myW, float myH, float myD) {
        w=myW; h=myH; d=myD;
    }
    public PVector getPosition() {
        return position();
    }
    public void setPosition() {
        setPosition(0,0,0);
    }
    void mouseClicked(Integer button) {
    }
    void mouseDragged(Point p, Camera camera) {
    }
}

```

```

public class IntelligentObject extends SpatialObject {
    SpatialObject parent;

    IntelligentObject(Scene _sc, int _id, String _name) {
        super(_sc, _id, _name);
    }

    public void draw() {
        p.draw();
        if (mode==SIMULATE_MODE) {
            execute();
        } else if (mode==INFORMATION_DESIGN) {
            if (grabsMouse()) {
                mouseEntered();
            } else {
                mouseLeaved();
            }
        }
    }

    public void triggered() {
    }

    void mouseEntered() {
    }

    void mouseLeaved() {
    }

    void mouseDragged(Point p, Camera camera) {
    }
}

```

a.

b.

圖 5-6. (a) 最基本的空間物件；(b) 最基本的資訊元件

後續的其他元件，就可以以這樣的方式持續繼承下去，例如在各種互動展場常用到的攝影機元件，就是一種資訊元件，我們就可以透過讓其繼承自資訊元件，再賦予攝影機專有的屬性跟方法，例如攝影機便會有影像這個屬性，負責儲存來自攝影機的畫面，是一種圖像的資料結構。根據上一節規劃中的各個元件的表格，將各種物件建立出來放在元件庫中以待取用。

5.2.3. 空間設計模式

空間設計模式是以 Processing 內具備的 OpenGL 繪圖模式來進行的，所有在 OpenGL 繪圖模式下的繪圖指令都具備 x、y、z 三個軸向，我們在空間設計模式納入了 ProScene 作為場景管理的函式庫，並透過渲染器來進行部分特效的繪製。結合這些功能後，Processing 預設的初始化函式 setup() 就必須在這些部分把

動作加進去，之後就要建立整個場景管理器的場景。場景的管理主要透過攝影機的設定而來，由攝影機的鏡頭設定以及操作模式的控制可以決定是哪一種模式，例如在資訊編輯模式下，攝影機是不動的；在空間設計模式下，滑鼠用來控制鏡頭；在模擬模式下滑鼠則是用來移動代理人。

```

import remixlab.proscene.*;
Scene scene;
InteractiveAvatarFrame avatar;

PShader blur, edges, ssao, toon;
PFont font;

StringList msg;
String modes = "SPACE DESIGN Mode";

void setup() {
  size(1280, 800, P3D);
  msg = new StringList();

  font = createFont("YaHei Consolas Hybrid", 48, true);
  textFont(font);

  msg.append("Computer Aided Design for Interactive Space v0.1");
  msg.append("互動空間之電腦輔助設計系統 v0.1");
  msg.append("カーティスシステム v0.1");
  msg.append("Mingly Shih (swingly@acm.org)");
  msg.append("=====");
  trace("SYSTEM: initializing");
  trace("SYSTEM: font loaded");

  smooth(8);

  trace("SYSTEM: OpenGL smoothing enabled");

  blur = loadShader("blur.glsl");
  edges = loadShader("edges.glsl");
  ssao = loadShader("custom.glsl");
  toon = loadShader("ToonFrag.glsl", "ToonVert.glsl");

  trace("SYSTEM: GLShader loaded");

  blur.set("blurSize", 64);
  blur.set("sigma", 16.0f);

  scene = new Scene(this);
  avatar = new InteractiveAvatarFrame(scene);
  avatar.setTrackingDistance(600);
  avatar.setAzimuth(PI/4);
  avatar.setInclination(avatar.inclination() - PI/16);
  WorldConstraint baseConstraint = new WorldConstraint();
  baseConstraint.setTranslationConstraint(AxisPlaneConstraint.Type.PLANE, new PVector(0.0f,1.0f,0.0f));
  baseConstraint.setRotationConstraint(AxisPlaneConstraint.Type.AXIS, new PVector(0.0f,1.0f,0.0f));

  scene.setRadius(400);
  scene.registerCameraProfile(new CameraProfile(scene, "CAD_CAM", CameraProfile.Mode.CAD));
  scene.setCurrentCameraProfile("CAD_CAM");
  scene.unregisterCameraProfile("WHEELED_ARCBALL");
  scene.setRightHanded();
  scene.camera().frame().setRotationSensitivity(1.5);
  scene.camera().frame().setSpinningFriction(0.5);
  scene.camera().frame().setTossingFriction(0.5);
  scene.camera().frame().setCADAxis(new PVector(0, 1, 0));
  scene.setAxisIsDrawn(false);
  scene.setGridIsDrawn(false);
  scene.showAll();
  scene.disableKeyboardHandling();
}

```

圖 5-7. Processing 的初始化程序以建立空間設計模式

建置好空間設計模式的基本雛形，我們可以建立一個基本的牆面跟地板作為最簡單的模型來載入，畫面如圖 5-8 所示，牆面的漸層光影是透過渲染器繪製的。

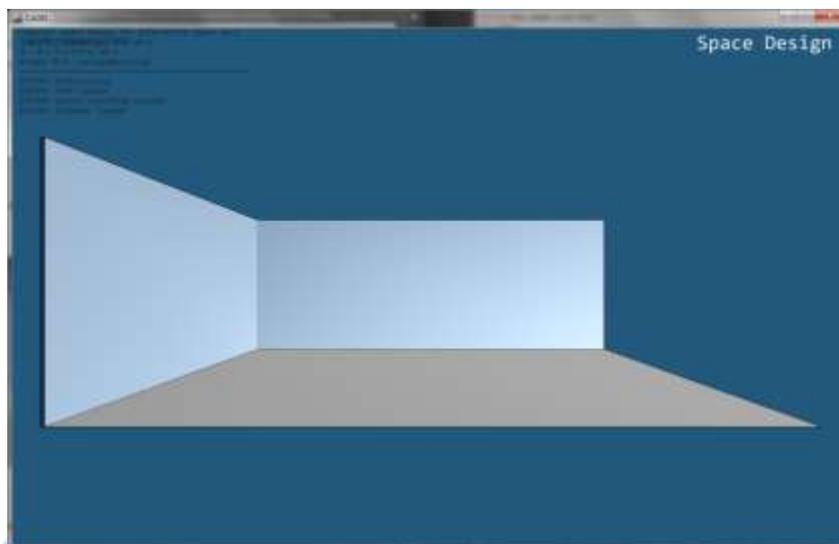


圖 5-8. 空間設計模式下載入基本的模型

另外在空間設計模式下，按照一般常見的電腦輔助設計軟體的設定，CADIS 也可以透過滑鼠的左鍵拖拉來旋轉空間，用右鍵拖拉的動作來進行空間模型的平移，以及中鍵拖拉的動作來做放大及縮小（圖 5-9）。

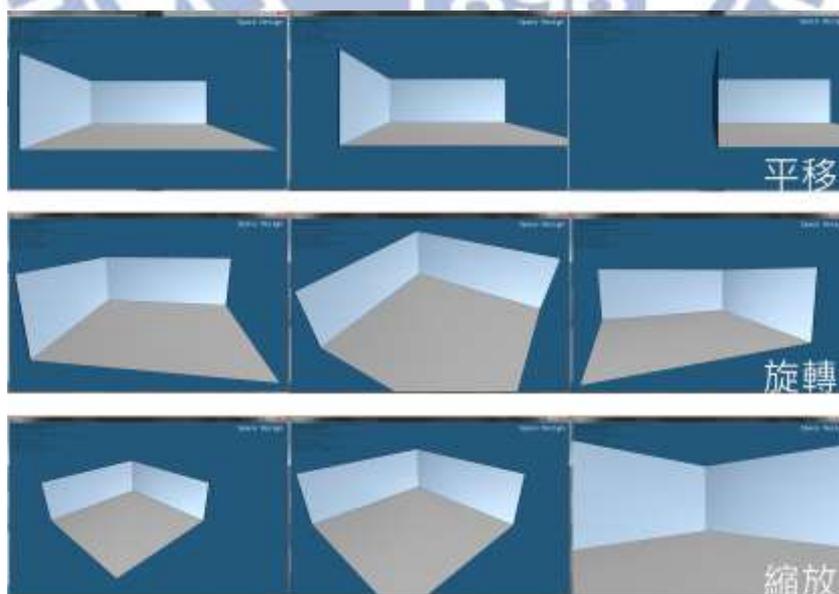


圖 5-9. 空間設計模式的滑鼠操作動作

而在滑鼠移到空間元件的中心點的時候，空間元件會呈現反白的狀態，並在畫面上顯示出設定的名稱，也就是物件屬性裡的 name 屬性（圖 5-10）。

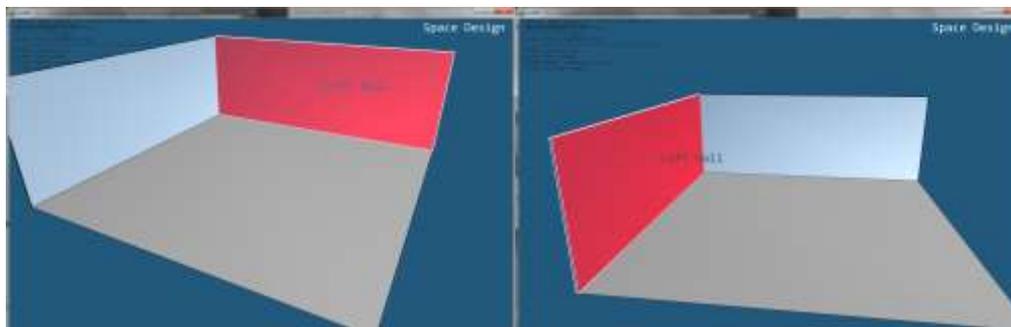


圖 5-10. 空間元件會在滑鼠移進去時呈現反白並顯示名稱讓設計者辨識

我們也可以透過選單新增一個物件到空間設計元件中，並透過鍵盤來移動該選定的物件，藉以在空間中安排配置各種空間元件，在之後的資訊設計模式下進行與資訊元件之間的連結與設定（圖 5-11）。

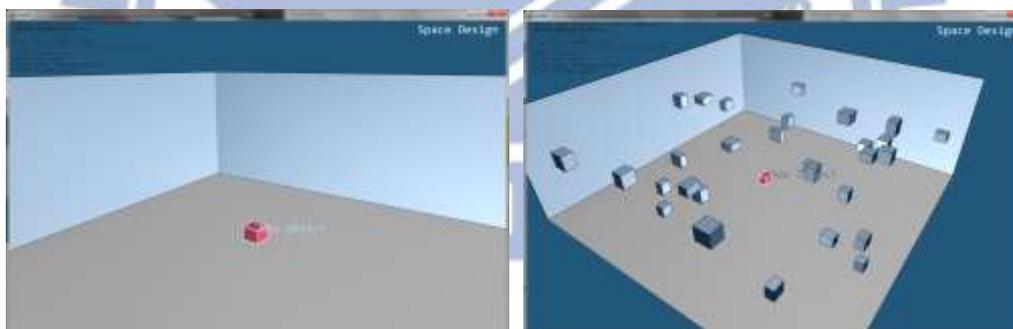


圖 5-11. 新增空間元件，CADIS 自動給予預設名稱

5.2.4. 資訊設計模式

在 CADIS 中，使用者透過按鍵的切換，從空間設計模式切換至資訊設計模式，在初始設定下，我們定義了 F1 鍵是切換到空間設計模式，F2 鍵是切換到資訊設計模式，F3 鍵是切換到模擬模式。我們先以一個場景作為例子，場景裡的物件有兩個空間實體“Left Wall”跟“Front Wall”、一片矗立在空間裡的 solid 型態元件名為“Door”（該元件的 name 屬性設為 Door）、掛在“Left Wall”上有一個 screen 型態的元件“TV”，另外有放在地面上的 projector 型態元件“Floor”

以及在 Floor 旁的一個 solid 型態元件，名稱為“Box Sofa”，另外還有一個 camera 型態的資訊元件“CC”就在鏡頭的位置（圖 5-12）。

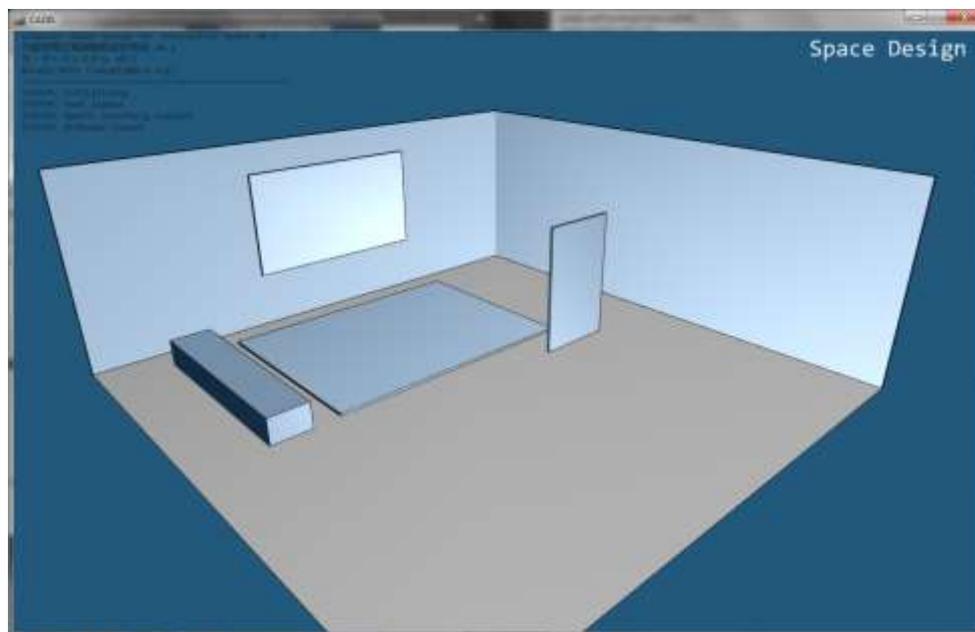


圖 5-12. 用來示範資訊設計模式的空間範例

按下 F2 鍵進入資訊設計模式後，CADIS 會自動把空間中可得到的相關元件全部放進來，並啟用渲染器將背景模糊化，保留一部分空間的樣貌可以觀察得到，但是又不致於影響資訊設計的基本介面，而像是 Front Wall 跟 Left Wall 這兩個單純的空間實體沒有特殊的互動效果就會自動被摒除掉（圖 5-13）。在圖 5-13 的介面中，每一個元件用一個方塊來表示，每個方塊上會顯示該元件的名稱（元件的 name 屬性），另外在方塊的上方位置，則會用比較小的文字來顯示該元件的形態，揭示元件的形態讓使用者比較能夠一眼了解元件的配置情形，了解目前整個空間中可運用的各式元件有幾個，另外在資訊設計模式下的介面是可以透過滑鼠的滾輪或中鍵來加以縮放以及用右鍵來拖拉畫面，因此如果畫面中的元件過多，也可以透過縮放以及拖拉的動作調整可視的範圍，方便使用者進行編輯。

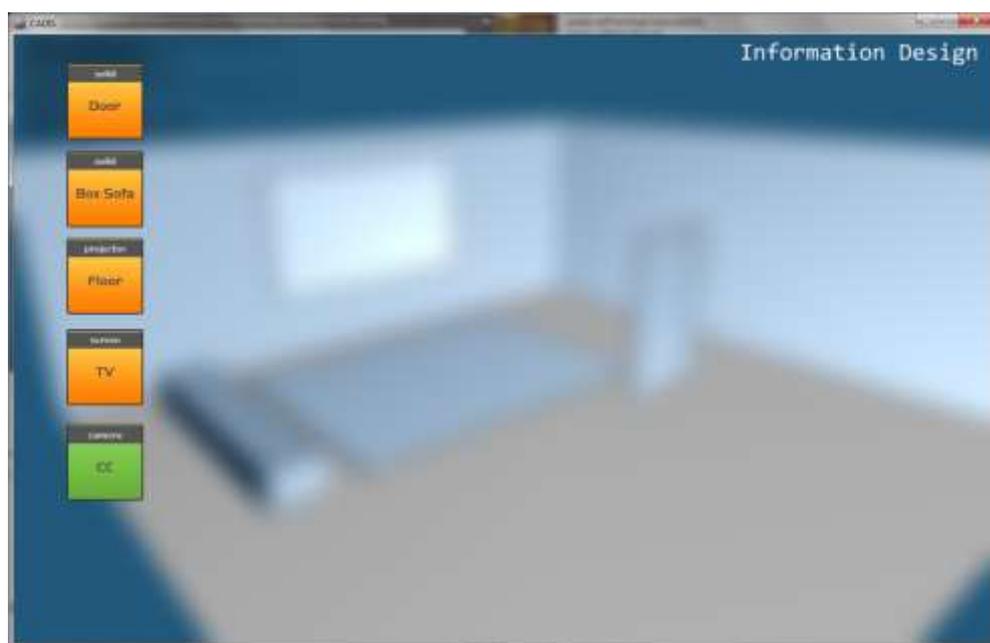


圖 5-13. 進入資訊設計模式後的介面，在介面中會列出所有可用的元件供使用者進行程式的設計
點選任一個元件，該元件可用的屬性跟方法便會被列出來，並透過字母來表示該屬性的資料型態，D 表示數位、N 是數字、T 是文字、S 代表集合、V 表示是向量、P 代表圖像、F 表示該型態為函數表型態（圖 5-14）。



圖 5-14. 點選特定元件後，列出該元件可用的屬性

選擇特定的屬性後，會在其屬性的標籤上標示隸屬的元件名稱，可以放置重複的屬性標籤，大量分布後便可以透過連接的方式將各屬性連結起來，兩個元件便可以透過這個屬性來互相連結，例如透過攝影機元件 CC 的 Image 屬性連結到投影元件 Floor 的 Image 屬性，攝影機元件的畫面便會自動地在 Floor 投影出來，所編寫出來的程式，就是關聯結構圖（圖 5-15）。



圖 5-15. 關聯架構圖的建立過程

5.2.5. 模擬模式

設計者在安排好空間元件，設定好資訊元件與空間元件的連結後，可以透過 F3 按鍵，切換至模擬模式，模擬模式是以第一人稱的形式讓設計者在空間中移動，攝影鏡頭的高度位置便是根據代理人的身高來設定的。在模擬模式中，以滑鼠的左鍵代表前進，右鍵代表後退，在前進跟後退的同時，滑鼠的方向則是用來轉動攝影鏡頭。在模擬環境中，使用者可以即時性地觀察到空間元件與資訊元件彼此連動的成果，以前一節的例子，當我們把攝影機 CC 的 Image 屬性跟 Floor 的 Image 屬性加以連結後，攝影機拍攝到的畫面就直接投射在 Floor 上，在空間導覽的同時就可以獲得視覺的回饋。

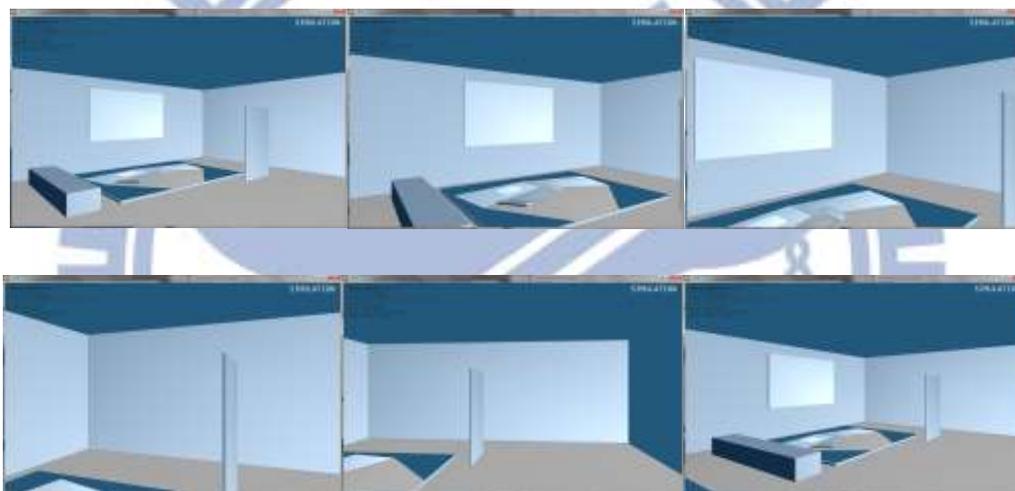


圖 5-16. 模擬環境中以第一人稱的方式在空間中移動觀看互動的模擬

5.3. 測試案例

在此節我們要透過 CADIS 的雛形系統來進行幾種案例的設計，用這些案例的設計過程來驗證 CADIS 的確容易操作，不需要複雜的程式專業背景與空間知識，而只要使用基本的滑鼠操作就可以建構出各種形式的互動空間，我們以漸進步驟的方式，並在每一個步驟附上圖解說明各項操作動作。

5.3.1. 測試 1：與攝影機結合的互動牆面設計

我們使用 CADIS 來建立一個與攝影機結合的互動牆，由攝影機偵測使用者的位置，並控制空間中的牆面自動跟隨使用者，建構出實體互動的空間，測試流程圖如圖 5-17 所示。

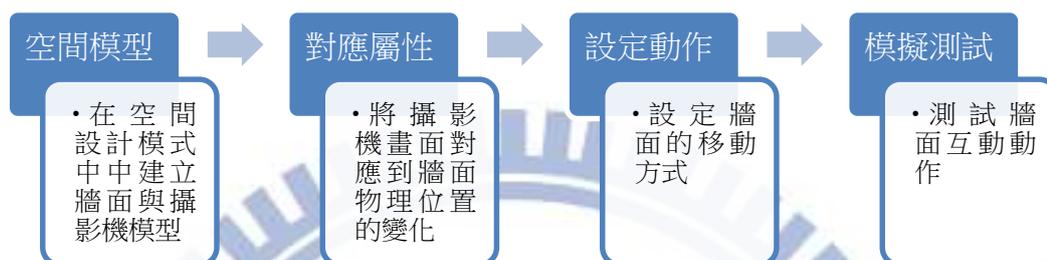


圖 5-17. 測試 1 操作流程圖

1. 空間模型：模型中事先建立好的空間元件有攝影機“C01”、可移動的互動牆“DynamicWall”（圖 5-18）。

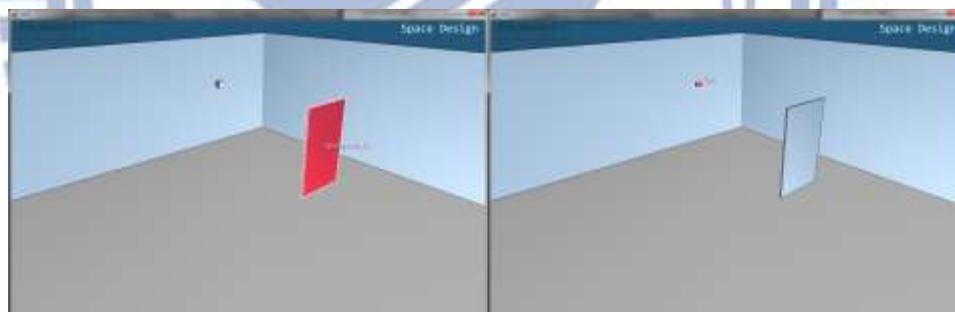


圖 5-18. 空間模型說明

2. 對應屬性：本測試案例是要將攝影機的畫面對應到牆面的物理位置，按下 F2 啟動互動編輯模式，拉出 C01 的 Image 屬性與 DynamicWall 的 Position 屬性，並拖動到適合的位置（圖 5-19）。

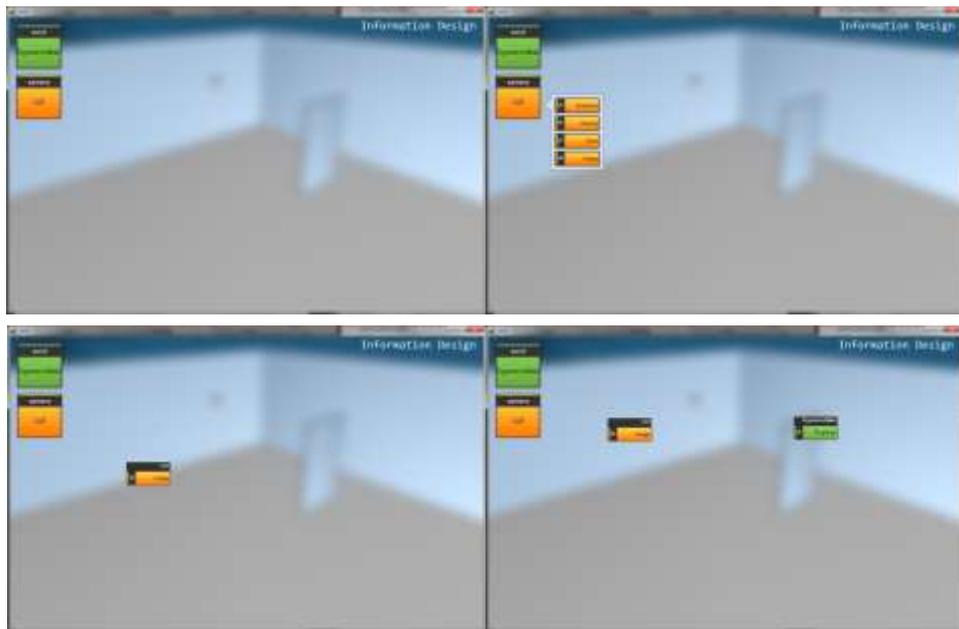


圖 5-19. 找出兩個元件之間要對應的屬性

3. 設定動作：對著 Image 屬性按下右鍵拉出預設的運算邏輯元件，影像屬性預設的對應機制只有兩種，選擇 MotionToPos，這是一個會將攝影機畫面進行處理並換算為位置座標的元件，再將他們三組連接起來（圖 5-20）。



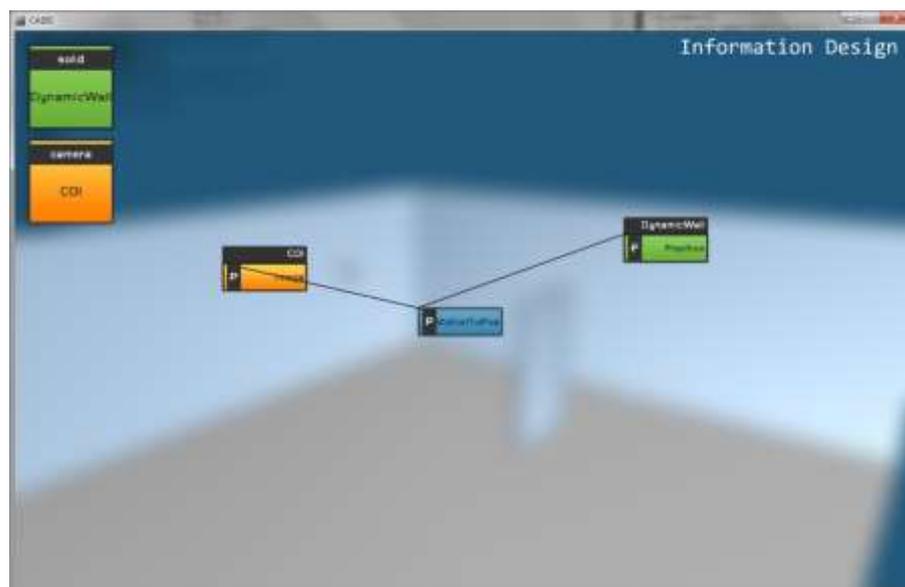


圖 5-20. 設定連結動作

4. 模擬模擬：按下 F3 鍵進入模擬環境，互動牆面會依照鏡頭拍攝到的畫面與使用者互動，使用者一有動作，牆面便會跟隨著使用者移動（圖 5-21）。



圖 5-21. 牆面依據設定的結果，跟隨著使用者移動

5.3.2. 測試 2：結合複合感測器的智慧建築立面設計

在這個測試案例，我們使用 CADIS 來嘗試建立複合式感測器的互動建築立面，這個空間會依據室內的溫度判斷是否要將門打開通風，但是又會在有人靠近時自動將門關上避免有人侵入。測試流程圖如圖 5-22 所示

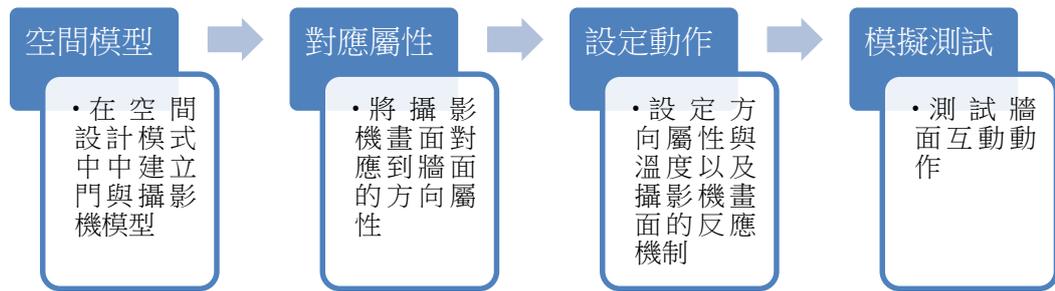


圖 5-22. 測試 2 操作流程圖

1. 空間模型：空間的模型中，有一個空間實體類型 (solid) 的物件，名為 "SmartDoor"，另外在門的上方放置了一個攝影機，名為 "CCTV"，剩下的部份則是外牆 (圖 5-23)。

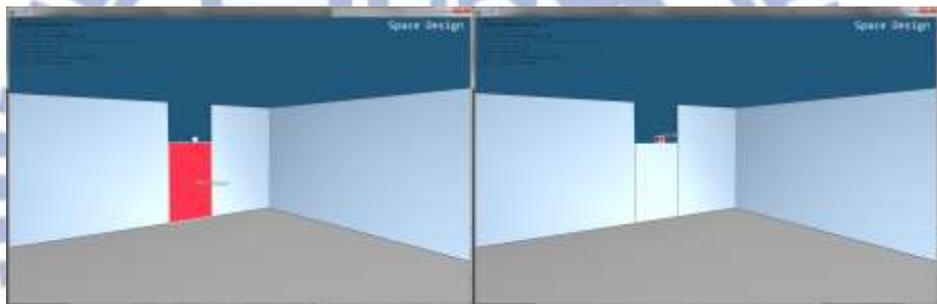


圖 5-23. 空間模型說明，這次利用兩個物件的多重屬性

2. 對應屬性：這個測試中要對應的屬性，除了有環境物件 Environment 的溫度 (Temperature 屬性) 之外，還有攝影機物件 CCTV 的 Image 屬性，以及 SmartDoor 的 Facing (方向) 屬性，也就是控制門的轉向方向。我們需要複數的 Facing 屬性，因為 CCTV 跟 Environment 都會跟這個 Facing 屬性產生互動 (圖 5-24)。



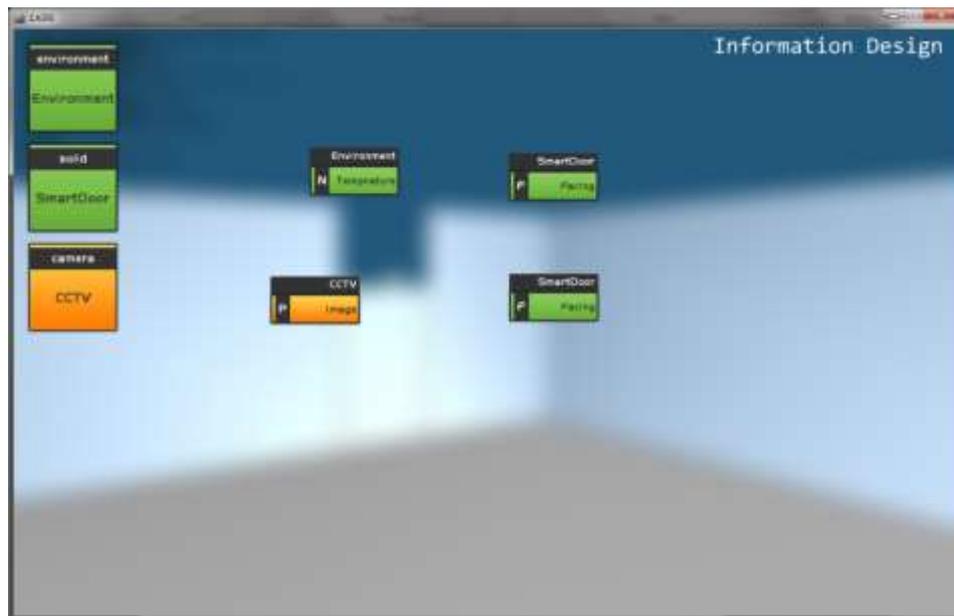


圖 5-24. 找出需要對應的屬性

3. 設定動作：這次需要比較多個邏輯元件來進行連結，首先我們需要判斷溫度，因此我們需要從系統選單裡拉來一個邏輯判斷元件 (TrueFalse)，再拖進來一個固定元件 (Fixed)，用來設定溫度的臨界值，固定元件設定為“>24”，再從邏輯判斷元件將連結拉至另一個固定元件設定=90，最後連結到 SmartDoor 的 Facing 屬性，就能用來改變 SmartDoor 的面向，轉 90 度表示開門。再來在 CCTV 的 Image 屬性上把運算邏輯元件的 MotionToPos 拉進來，同樣需要一個固定元件，設定攝影機拍攝到的畫面的臨界值，設定為“>80”，同樣再透過另一個邏輯判斷元件，連結到一個設定為 0 的固定元件，最後連結到另一個 SmartDoor 的 Facing 屬性，就可以讓 SmartDoor 面向轉回來為 0 度，變成關門 (圖 5-25)。

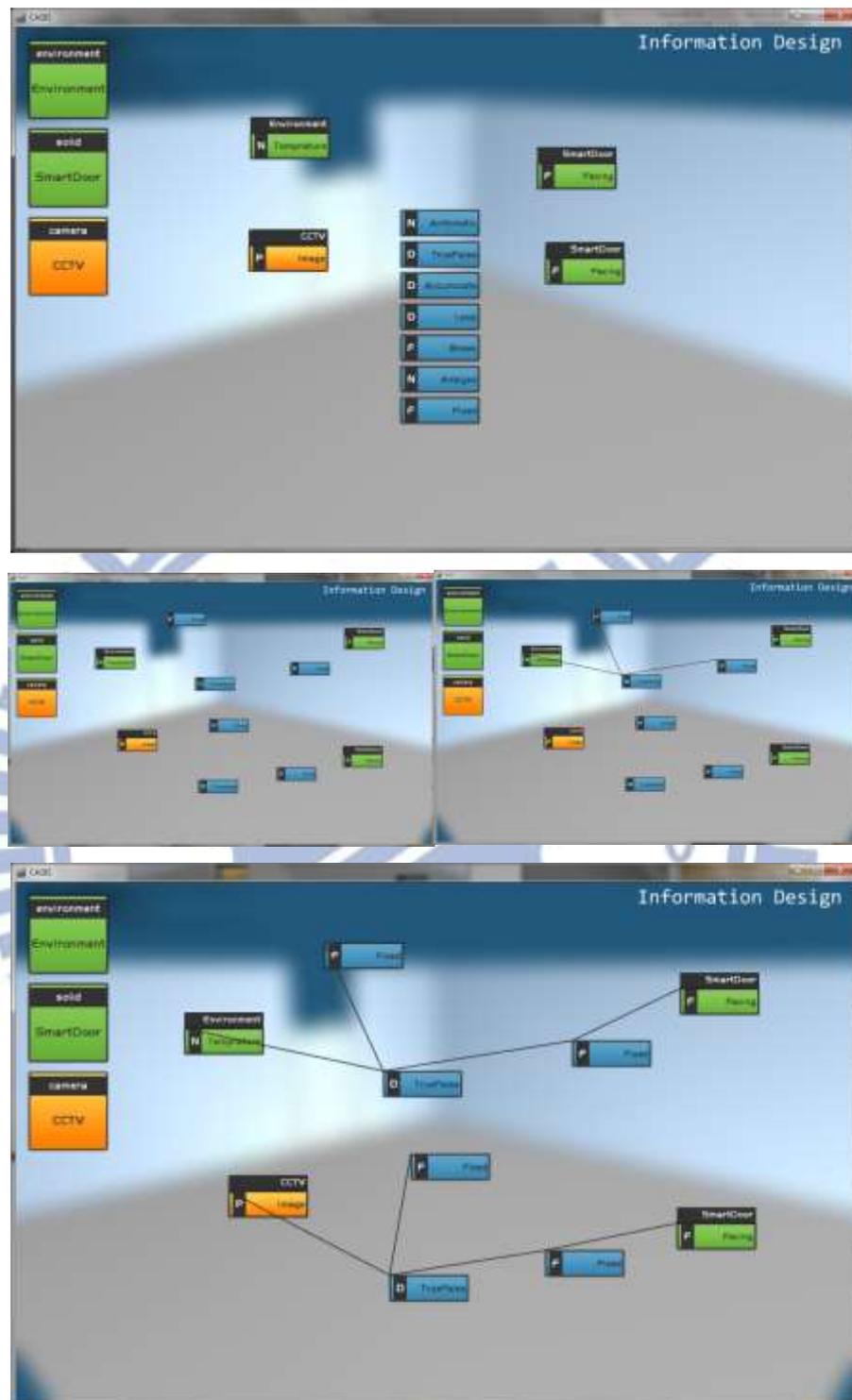


圖 5-25. 將空間元件、資訊元件與邏輯元件串接起來

4. 模擬測試：進入模擬環境進行測試，當人遠離的時候，溫度又超過 24 度，智慧門就會打開以降低室內溫度，而當鏡頭發現有人靠近的時候，門就會

自動關上，的確達到我們設定的動作目標（圖 5-26）。

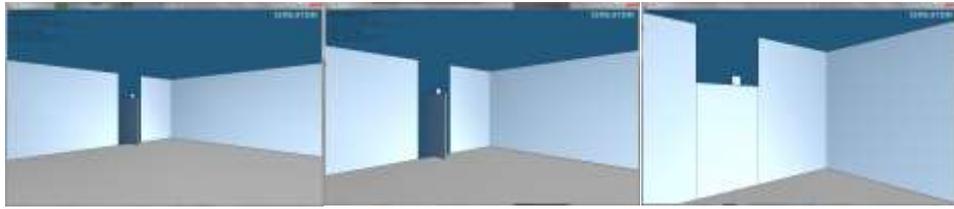


圖 5-26. 慢慢靠近智慧門，則自動關上

5.3.3. 討論

從我們的測試，我們發現，運用 CADIS 提供的方式，的確可以快速地撰寫出互動的機制，只需要透過幾項的拖拉動作，就可以讓空間具備智慧，或是讓空間與人們互動。從流程圖也可以看得出來步驟的相似性，步驟一先建立空間模型，步驟二連結空間與資訊元件、步驟三設定動作方式以及最後一個步驟進行模擬測試。相類似的操作流程可以降低使用者的學習負擔，這也達到了我們希望這是一個針對非專業者所設計的電腦輔助設計系統的目的；空間設計模式只使用簡單的量體讓資訊設計者可以不用學習太多專業的知識就可以了解空間的配置；資訊設計模式透過視覺化的互動撰寫方式，讓空間設計者不需要學會任何程式語言就可以設計出一些空間元件之間的互動機制；模擬環境則讓設計者可以以第一人稱的方式在場景中瀏覽，直接觀察到執行的結果，並可以隨時在設計與模擬模式之間進行切換。

然而從我們的測試例子，也看得出來整個流程較缺乏說明的細節，對於第一次用 CADIS 的使用者而言，容易有不知道現在要做什麼的感覺；另外，視覺化的方式雖然降低了學習程式的負擔，但是一旦使用的元件多了，也比較容易發生畫面非常混亂的情況，對於程式邏輯的判讀上會造成一些困擾，不過這一點對於所有的視覺化程式這個領域而言都還是一個需要持續去進行的研究方向。

6. 結論

由於科技的進步，電腦以及各種小型處理器的運算速度越來越快，運算單元的結構越來越小，使得將這些處理器嵌入到實體空間的各種器物中變成了是一個越來越容易的事情，也因此我們的空間開始有著越來越普遍的具有智慧的物件，小從小盒子、文具、燈具、桌椅，大至牆面、地板甚至到建築物的立面。這些智慧物件可以感應我們的存在與否、可以了解天氣現況、探知我們的行為舉止，也可以讓建築物產生變形、讓立面變成介面、或是運用外牆傳達訊息。我們在這過程之中，看到了智慧空間的浮現。

一個在實體跟虛擬的層面都設計得當的智慧空間，除了在實體層面由空間設計師設計一個兼具美感與物理機能的建築空間，讓人們生活在其中感受空間的規劃帶來的舒適之外，虛擬的層面可以提供人們動態的機能，它們可以主動關懷空間中的使用者，它們可以整合實體虛擬的所有訊息做出精確的判斷來節約能源、提供更貼心的服務，讓使用者可以不用花費心思在控制各式各樣的設備或裝置。在智慧空間中我們看到空間設計師與資訊設計師的彼此合作，共同賦予空間生命。然而，專業背景的差異性，讓空間設計師與資訊設計師之間難以擁有共同的溝通平台，電腦輔助設計軟體對於資訊設計者而言太複雜，程式語言對於空間設計師而言也是難以理解。他們需要大量的溝通以消弭知識上的落差來確定彼此的想法，他們也沒辦法針對對方的領域提出不同的觀點相互討論，因為現有的工具媒材都太過專注在各自的範圍裡，雖然有部分的工具發展方向朝向給非專業者使用，但並沒有一個整合的工具。所以，雖然新的空間浮現，卻帶來了一個重要的問題，也就是現有的設計媒材都還沒辦法滿足設計這類型空間的需求。

研究貢獻

有鑑於此，本研究的目的便是在於嘗試整合資訊設計媒材與空間設計媒材，將兩種不同媒材的關鍵特點提出並加以整合，最後實作了一套雛形工具系統，讓兩個領域的設計者都能夠運用，稱為智慧互動空間的電腦輔助設計系統 CADIS。

本研究在分析空間案例時整理出以下幾項重點，可以作為智慧空間設計時的基本概念：

- 從智慧空間的案例分析中我們發現，許多智慧空間的設計都是運用了像素與實體元件之間的對應，每一個像素的變化對應到每一個實體元件的變化，這樣的一對一關係，讓虛擬的訊息得以轉換型態後透過空間實體展現出來，例如媒體立面或是資料的回饋等等形式。
- 有些類型的智慧空間，則是透過偵測或感應器將空間中的人因及環境變數數值化，透過計算及判斷這些數值的變化，設計師則需規劃空間中的元件應該採取的反應以滿足使用情境的需求。
- 嵌入在空間中的資訊元件或裝置，必須要具備基本的物理資訊，尤其是位置與面向，這有助於進一步的互動以及活動的判斷，例如攝影機有了位置與面向的資訊，負責計算的數位系統便能得知畫面與空間的對應關係。

而在分析工具案例時，我們也整理出來幾個有關於電腦輔助設計系統的關鍵重點，可作為觀察與分析電腦輔助設計系統時的參考：

- 在空間設計媒材的工具系統中，最關鍵的重點是在於三維擬真的瀏覽環境以從各種角度觀察空間中各個元件的配置，建模以及貼圖等等功能，則被簡化功能以降低系統複雜度，近期的 SketchUp 以及 AutoCAD123 都具有這樣簡化的特徵。建化後的空間設計環境，可以讓資訊設計師更容易調整與理解空間進而得以更合理與精確地處理反應的部份。
- 視覺化的資訊設計環境、簡化的邏輯與記憶體操作，是許多針對非專業背景者的程式語言工具的關鍵特徵，如 Pure Data 與 Scratch。透過簡單的視覺化區塊堆疊與連結便可以撰寫程式，不需要記憶語法、詞彙或規則，降低了非程式背景的設計者設計互動行為的學習門檻。

- 元件庫的使用與經營，則已是現在許多電腦輔助設計系統的基本功能，例如 ArchiCAD 內建了許多跟室內設計相關的元件（植栽、牆、窗、家具等等），Sketchup 可以從網路資料庫下載大量元件來應用，Unity 的 Assets Store 也讓設計者付費或免費下載元件運用在自己的設計中。
- 即時的模擬不論在資訊設計媒材或是空間設計媒材都是非常重要的基本元素，讓設計者在進行設計時，可以邊設計邊觀看模擬的結果以反覆地調整自己的設計。在空間設計媒材中，發展的趨勢在減少運算時間可以迅速反應的視覺模擬；在資訊設計媒材中，發展的方向則是往多樣化輸出的模擬，例如模擬燈光效果、控制的線路模擬等。

在實作 CADIS 的過程，我們也發現了一些重點，在發展電腦輔助設計媒材時也可以作為參考原則：

- 由於發展方向上的差異，CADIS 並沒有取代專業媒材的能力。必須跟現有的媒材進行搭配，因此觀察到了互通機制的問題。傳統的互通機制就是透過檔案交換的方式來達成，然而不同的檔案格式就有著不同的優劣效益，在發展任何電腦輔助設計媒材時，交換檔案的相容性會是非常重要的而必須重視的特點。本研究焦點在於整合兩種不同的設計媒材，因此並沒有針對這個問題進行更深入的探究，而僅採用 obj 作為通用的檔案格式，也因此受到 obj 格式的限制，在尺寸或是軸向上並沒有辦法做到精確的控制，成為研究上的限制之一。
- 電腦輔助設計媒材的環境必須能夠同時提供第一人稱與第三人稱的鏡頭變化。設計者慣於運用第三人稱視角進行設計以獲得全面的空間訊息，然而在模擬時，必須要能夠模擬人眼的視角，以更擬真地觀察空間的氣氛與細節。
- 環境中的元件必須要有獨一無二的識別或是由設計者命名，這對於空間設計媒材與資訊設計媒材都是非常重要的功能。資訊設計者或是空間設計者，都必須要透過這個識別名稱來判定配置的位置與互動的對象。

最後實作完成的 CADIS，雖然是一個雛形系統，但是已經能夠運作並進行測試。

- CADIS 有讓空間設計師可以簡單撰寫程式的資訊設計環境，讓他們可以透過這個機制表達自己對於互動設計部分的想法，從而期望讓空間與資訊之間的連結更為緊密。
- 透過 CADIS 設計者可以簡易地配置空間的實體，再為這些實體創造虛擬的連結讓各個空間構件之間可以產生互動性，由這些互動所建構出來的行為與空間的本體相結合形成了智慧空間。
- 我們在測試案例 1 測試了建構一個投影系統結合攝影機畫面的簡單互動，也在測試案例二創作一個「天氣太熱而且附近沒有人就自動打開房間通風讓室內溫度下降，一有人靠近就自動把門關上」的新型態空間，從測試案例可以得知，運用這樣整合了空間與資訊設計的系統除了整體設計的過程簡化之外，也可以發展新的空間設計。

本研究的貢獻，最主要在於結合空間設計媒材與資訊設計媒材而提出新型態媒材的嘗試，而從測試結果的案例可以得知這是具備可行性而且重要的方向，使用者可以透過這個系統探索更多可能的智慧空間形式與其互動設計，而不需要事先學習大量的專業知能。這樣的一個新媒材工具的方向，就如同 Mitchell 所說未來的模擬工具除了渲染逼真的畫面或是能源的效能模擬之外，還必須能夠「執行」設計的建築物因應各種不同外在條件與使用者需求的情境下所作的變化。

研究限制

本研究所提出的 CADIS 系統是一個雛形系統，有著研究上的限制。第一個限制是在系統元件庫的完整度上，目前的系統僅內建了一些現下常見與常用的資訊與空間元件。由於未來不知道會出現什麼樣子的互動裝置可以用來嵌入空間中，所以雖然本系統運用了物件的方式來建立元件庫，在未來有新的資訊元件時可以透過繼承的方式建立更大的元件庫，而有著更大的元件庫，也代表著可以創造出更多的可能性，但這有待持續地擴充。

第二個限制，是在輸出的機能上並不完備。使用者雖然可以運用 CADIS 進行設計，但是設計好的內容除了空間模型是通用的檔案格式可以在不同軟體間轉移之外，其

他的部分並沒有辦法匯出或是轉移到其他的軟體進行更細部的設計。例如規劃好的關聯架構圖表，就沒有辦法直接轉換成相對應的程式碼，而僅止於於模擬的階段觀看而已。

未來方向

本研究提出了一個新的方向，也就是一個整合了空間設計與資訊設計的電腦輔助設計工具，也透過 CADIS 這個雛形系統的實作，試圖看到未來的電腦輔助設計工具該有的樣貌。我們針對下列幾個層面來討論 CADIS 未來可以發展的方向：

1. 設計教育者：對於設計教育者而言，他們未來可以透過發展更完備的 CADIS，來進行設計課程的教學。這樣的教學融合了互動設計與空間設計，在空間設計時需考量資訊元件的配置，在資訊互動設計時需注意其與空間的對應關係。在這塊領域的發展上，將會朝著整合軟硬體的方面發展。例如將設計好的智慧空間中戶動的部份寫入到 Arduino，空間的部份則以模型或是 3D 列印的形式印出，再經過組裝後，便可以有著可以實際運作的模型，在教學上將有著相當大的助益，相關的課程可以結合兩種領域被境的學生，也可以是完全的初學者，因為 CADIS 的基本特徵就是一個針對非專業設計者的電腦輔助設計媒材。
2. 學生：由於易學易用的特性，學生可以透過 CADIS 進行設計的學習，從 CADIS 的發展中我們觀察到，更加精確的視覺模擬是未來的方向之一，這包括了正確來自光追蹤計算的光影效果以及透過這些光影帶出來的空間氛圍，這將會在渲染器更進一步發展後，未來的電腦輔助設計工具將不再需要經過彩現的動作，直接而即時地顯現出來。學生在學習時，不會因為專業知識上的落差而無法理解設計的成果，而能藉由更精確的模擬效果了解光影效果、空間型態、結構變化以及到人與空間之間的互動關係，這有賴於更精準的模擬技術的發展才能達成。
3. 專業設計者：未來發展的方向是在設計溝通上，因為在專業設計者之間的溝通將會是發生在兩種不同專業背景者之間。未來需要建立將 CADIS 作為溝通平台的機能性擴充，例如：元件庫的擴充共享、同時的多人操作、文件

的交換、設計概念的快速傳達、同時進行多種互動效果的模擬以讓參與的設計者能夠方便進行討論、分享想法或是討論進度，這些都是未來針對設計溝通可以結合的方向。像這樣即時而且多人共同的創作模式，已被運用在資訊設計環境如 Hackpad 可以讓多人同時撰寫程式並驗證語法正確性，以及文件共筆環境的 Google Drive 中讓參與者可以在同一份文件、試算表或是簡報中各自編寫不同的內容，顯見這是一個有助於共同創作與進行溝通的機制。

4. 產業：與產業相關的未來的研究，將會是電腦輔助設計與控制系統的結合，就如同電腦輔助設計系統與電腦輔助製造系統（CAM）的整合一般，整個產業的設計流程將會有很大的變化，未來從初期的設計階段到後期的製造階段，再連接到最終端的使用者將會整合到同一套系統中。在發展 CADIS 的過程中我們發現，這些元件之間的相互串連背後所使用的機制，與控制系統的是相同的；再加上嵌入到空間元件中的處理器越來越強大，在未來將有足夠的運算能力來進行自我程式的改寫，因此我們認為，在未來電腦輔助設計系統將與控制系統相結合，而變成一個可程式化的控制系統，設計者透過這個系統進行設計與配置，透過輸出功能將互動的機制燒錄為晶片，同樣的輸出功能會將空間的模型結構元件透過快速成形等機器建置為實體，最後使用者也會運用同一套系統進行互動機制的改寫與調整，以更符合他的需求，這將會是一個很重要的未來方向與設計流程上的變革。

本研究鑑於新型智慧空間的浮現以及現有設計媒材的不足，嘗試將兩種不同的設計媒材進行整合而希望提出新的設計創作媒材。在研究的過程中，我們分析了設計的案例找出智慧空間的設計特色、也分析了各種媒材工具尋找設計媒材的關鍵重點、而最後實作的系統 CADIS 雖然僅是一個可運作的雛形系統，但已經達到本研究設定的目的，讓 CADIS 可以同時讓空間設計者與資訊設計者都可以簡易地學習操作。從智慧空間的設計媒材角度出發，本研究是一個新型態設計媒材研究的起點，也是電腦輔助設計研究的新分歧點。未來的設計媒材，當如 Mitchell 所預言的除了渲染逼真的畫面或是能源的效能模擬之外，還必須能夠「執行」設計的建築物因應各種不同外在條件與使用者需求的情境下所作的變化。

參考文獻

- Anders, P., (2004). Arch-OS: An Implementation of Cybrid Strategies, in proceedings of Annual Conference of the Association for Computer Aided Design in Architecture, Cambridge, Ontario, pp. 282-293.
- ART+COM, (2004). digital media.interface, in *iF yearbook communication* Birkhauser.
- Ballagas, R., M. Ringel, M. Stone & J. Borchers, (2003). iStuff: a physical user interface toolkit for ubiquitous computing environments, in proceedings of SIGCHI conference on Human factors in computing systems, Ft. Lauderdale, Florida, United States, pp. 537-544.
- Ballagas, R., A. Szybalski & A. Fox, (2004). Patch Panel: Enabling Control-Flow Interoperability in UbiComp Environments, in proceedings of Second IEEE Annual Conference on Pervasive Computing and Communications, pp. 241-252.
- Bertelli, P., R. Koolhaas, J. Hommert & M. Kubo, (2001). *Projects for Prada*, Fondazione Prada.
- Biloria, N. & K. Oosterhuis, (2005). Design Informatics, in proceedings of 2005 Annual Conference of the Association for Computer Aided Design in Architecture, Savannah, Georgia, United States, pp. 226-235.

- Bonanni, L., E. Arroyo, C.-H. Lee & T. Selker, (2005). Exploring feedback and persuasive techniques at the sink. *interactions*, 12(4), 25-28.
- Brouwer, J., P. Brookman & A. Mulder, (2001). *Transurbanism*. NAI Publishers
- Brodey, W. M., (1967). Soft architecture: The Design of Intelligent Environments. *Landscape*, 17(1), 8-12.
- Carter, P., (2002). *PC Assembly Language*.
- Chiu, M.-L. (ed.) (2005). *Insights of Smart Environments*, Taipei, Taiwan: Archidata.
- Chiu, M.-L., B. Chiang, G.-C. Lee & H. Tseng, (2006). House sense, in proceedings of CAADRIA 2006, Kumamoto, Japan, pp. 313-322.
- Cook, D. & S. Das, (2004). *Smart Environments: Technology, Protocols and Applications*, Wiley.
- Crampton, N. (2009). SIMS Furniture An interactive, user-centric system for furniture design-Preliminary Research Document.
- Daab, (2006). *ag4 Media Facades*, Daab.
- dECOi Architects (2011). Hyposurface. Retrieved 18 June, 2013, from <http://www.decoi-architects.org/2011/10/hyposurface/>
- Dertouzos, M. L., (2001). *The Unfinished Revolution: Human Centered Computers and What They Can Do for Us*, Collins.
- Dertouzos, M. L. & B. Gates, (1997). *What Will Be: How the New World of Informatin Will Change Our Lives*, Harper, San Francisco.

- Dietz, P. H., & Eidelson, B. D. (2009). SurfaceWare: dynamic tagging for Microsoft Surface. In Proceedings of the 3rd International Conference on Tangible and Embedded Interaction pp. 249-254. ACM.
- Do, E. Y. L. (2001). VR sketchpad. In Computer Aided Architectural Design Futures 2001 pp. 161-172. Springer Netherlands.
- Ducatel, K., M. Bogdanovicz, F. Scapolo, J. Leijten & J.-C. Burgelman, (2001). *Scenario for Ambient Intelligence in 2010*, Seville.
- Eckel, B., (2006). *Thinking in Java (4th Edition)*, Prentice Hall.
- Fernandez, A. (2013). Wall Sconces with Hidden Weather Display and Tangible User Interface, retrieved 24 June, 2013, from <http://www.instructables.com/id/Wall-Sconces-with-Hidden-Weather-Display-and-Tangi/>
- Flightphase, (2012). Interactive Wall at University of Dayton, retrieved 18 June, 2013 from <http://www.flightphase.com/case-studies/ud-interactive-wall>
- Giselbrecht + Partner ZT GmbH, (2010). Dynamic Façade. Retrieved 18 June, 2013, from <http://www.dailytonic.com/dynamic-facade-kiefer-technic-showroom-by-ernst-giselbrecht-partner-at/>
- Greenberg, I., (2007). *Processing: Creative Coding and Computational Art (Foundation)*.
- Greenberg, S. & C. Fitchett, (2001). Phidgets: Easy Development of Physical Interfaces through Physical Widgets, in proceedings of UIST, pp. 209-218.

- Gross, M., S. Wurmlin, M. Naef, E. Lamboray, C. Spagno, A. Kunz, E. Kolle-Meier, T. Svoboda, L. V. Gool, S. Lang, K. Strehlke, A. V. Moere & O. Staadt, (2003). blue-c: a spatially immersive display and 3D video portal for telepresence, in proceedings of SIGGRAPH 2003, San Diego, California, pp. 819-827.
- Hernando, B., (2004). Wiring: Prototyping Physical Interaction Design. Programming and prototyping with electronics for designers, Interaction Design Institute Ivrea.
- Huang, J. & M. Waldvogel, (2004). The swisshouse: an inhabitable interface for connecting nations, in proceedings of Symposium on Designing Interactive Systems, ACM Press, Cambridge, MA, United States, pp. 195-204.
- Ishii, H. & B. Ullmer, (1997). Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms, in proceedings of CHI'97, Atlanta, Georgia, United States, pp. 234-241.
- Jeng, T., (2002). Interaction and Social Issues in Human-Centered Reactive Environment, in proceedings of CAADRIA, Cyberjaya, Malaysia, pp. 285-292.
- Jeng, T., (2005). Advanced ubiquitous media for interactive space, in proceedings of Computer Aided Architectural Design Futures 2005, Springer, Vienna, Austria, pp. 341-350.
- Jeng, T., H.-C. Lin, Y.-T. Shen, C.-A. Pan & C.-I. Chen, (2007). Physical prototyping of interactive space, in proceedings of CAADRIA 2007, Nanjing, China, pp. 369-376.

- Jordà, S., Geiger, G., Alonso, M., & Kaltenbrunner, M. (2007). The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces. In Proceedings of the 1st international conference on Tangible and embedded interaction pp. 139-146. ACM.
- Kelly, T., (2001). *The Art of Innovation*, Doubleday, New York.
- Kernighan, B. W. & D. M. Ritchie, (1988). *The C Programming Language*, Prentice Hall.
- Kim, M.-Y. & J.-W. Choi, (2006). Developing a digital storytelling model for virtual test-bed to simulate the smart office, in proceedings of CAADRIA 2006, Kumamoto, Japan, pp. 331-340.
- Kim, S., (2003). Chanel store on the Rue Faubourg designed by Peter Marino + Assoc Architects, in *Interior Design*.
- Klemmer, S. R., M. W. Newman, R. Farrell, M. Bilezikjian & J. A. Landay, (2001). The designers' outpost: a tangible interface for collaborative web site, in proceedings of the 14th annual ACM symposium on User interface software and technology, ACM Press, Orlando, Florida, pp. 1-10.
- Kvan, T., (2006). Design ergo sum, in proceedings of CAADRIA 2006, Kumamoto, Japan, pp. 19-26.
- Lee, C.-H. & T. Jeng, (2002). A Context Manager for Multimedia Presentation in Intelligent Corners: a Human-Centric Interaction Approach, in proceedings of 7th International Conference on Computer Aided Architectural Design and Research in Asia (CAADRIA 2002), Malaysia.
- Lightspace, (2007). Lightspace, retrieved 20 July, 2011 from <http://www.lightspacecorporation.com>

- Lim, C.-K., C.-S. Tang, W.-Y. Hsao, J.-H. Hou & Y.-T. Liu, (2006). NEW MEDIA IN DIGITAL DESIGN PROCESS: Towards a standardize procedure of CAD/CAM fabrication, in proceedings of 11th International Conference on Computer Aided Architectural Design Research in Asia, Kumamoto, Japan, pp. 597-599.
- Lin, H.-C., Y.-T. Shen & T. Jeng, (2005). IP++:Computer-augmented information portal in place, in proceedings of CAADRIA 2005, Vol. 2, New Dehli, India, pp. 185-192.
- Liu, Y.-T., (1996). *Understanding of Architecture in the Computer Era*, Hu's, Taipei.
- Lubell, S., (2003). In Graz, Austria, a new arts center will speak its own digital, in *Architectural Record*.
- Luminvision Ltd., (2009). Interactive Floor Projection, retrieved 20 July, 2010 from <http://www.luminvision.co.uk/>
- Maeda, J., (2001). *Design By Numbers*, MIT Press.
- Maher, M. L. & M. Kim, (2006). The effects of tangible user interfaces on designers' cognitive actions, in proceedings of CAADRIA 2006, Kumamoto, Japan, pp. 119-124.
- Maher, M. L., M. Rosenman, K. Merrick, O. Macindoe & D. Marchant, (2006). Designworld: an augmented 3d virtual world for multidisciplinary, collaborative design, in proceedings of CAADRIA 2006, Kumamoto, Japan, pp. 133-142.

- Marentette, L., (2010). New Hollywood Hard Rock Cafe Sparkles with Interactive Multi-touch Wall and Microsoft Surface Booths! Retrieved 18 June, 2013 from <http://interactivemultimediatechnology.blogspot.tw/2010/08/new-hollywood-hard-rock-cafe-sparkles.html>
- Mathews, S., (2006). The Fun Palace as Virtual Architecture: Cedric Price and the Practices of Indeterminacy. *Journal of Architectural Education*, 39-48.
- Mitchell, W. J., (2006). From sketch to city of bits, in proceedings of CAADRIA 2006, Kumamoto, Japan, pp. 1-5.
- Moggridge, B., (2006). *Designing Interactions*, MIT Press.
- Multitouch Ltd., (2013). MultiTouch and USC School of Cinematic Arts Unveil Largest University Interactive Display Installation in North America, retrieved 18 June, 2013 from <http://www.multitouch.com/2013/06/13/multitouch-and-usc-school-of-cinematic-arts-unveil-largest-university-interactive-display-installation-in-north-america/>
- Norman, D. A., (2002). *The design of everyday things*.
- Norman, D. A., (2004). *Emotional Design: Why We Love (or Hate) Everyday Things*, NetLibrary, Inc.
- Norman, D. A., (2005). Human-centered design considered harmful, in *interactions*, 14-19.
- Norman, D. A. & S. W. Draper, (1986). *User Centered System Design; New Perspectives on Human-Computer Interaction*, Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA.

- Oosterhuis, K., (2003). *Hyperbodies, Towards an E-motive Architecture*, Birkhauser.
- O'Sullivan, D. & T. Igoe, (2004). *Physical Computing: Sensing and Controlling the Physical World with Computers*, Course Technology.
- Palmon, O., R. Oxman, M. Shahar & P. L. Weiss, (2005). Virtual environments in design and evaluation: an aid for accessible design of home and work settings for people with physical disabilities, in proceedings of Computer Aided Architectural Design Futures 2005, Springer, pp. 145-154.
- Park, S. H., S. H. Won, J. B. Lee & S. W. Kim, (2004). Smart home-digitally engineered domestic life. *Personal and Ubiquitous Computing*, 7(3), 189-196.
- Patten, J., H. Ishii, J. Hines & G. Pangaro, (2001). Sensetable: a wireless object tracking platform for tangible user interfaces, in proceedings of SIGCHI, ACM Press, Seattle, Washinton, U.S., pp. 253-260.
- Peng, C., (2005). Townscaping: development of dynamic virtual city augmented 3d sketch design tools, in proceedings of Computer Aided Architectural Design Futures 2005, Springer, pp. 105-114.
- Ponnekanti, S. R., B. Lee, A. Fox, P. Hanarahan & T. Winograd, (2001). ICrafter: A Service Framework for Ubiquitous Computing Environments, in proceedings of UbiComp 2001, Springer-Verlag, pp. 56-75.
- Preece, J., Y. Rogers, H. Sharp, D. Benyon, S. Holland & T. Carey, (1994). *Human Computer Interaction (ICS)*, Addison Wesley.

- Ratti, C., Y. Wang, H. Ishii, B. Piper & D. Frenchman, (2004). Tangible User Interfaces (TUIs): A Novel Paradigm for GIS. *Transactions in GIS*, 8(4), 407-421.
- Rekimoto, J. & M. Saitoh, (1999). Augmented surfaces: a spatially continuous work space for hybrid computing environments, in proceedings of SIGCHI, ACM Press, Pittsburgh, Pennsylvania, United States, pp. 378-385.
- Resnick, M., (2007). All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten, in proceedings of SIGCHI, ACM Press, Washington, DC, United States, pp. 1-6.
- Ryokai, K., S. Marti & H. Ishii, (2004). I/O brush: drawing with everyday objects as ink, in proceedings of SIGCHI, ACM Press, Vienna, Austria, pp. 303-310.
- Scenecaster, (2009). Scenecaster, retrieved 20 July, 2010 from <http://scenecaster.wikispaces.com/>
- Schoch, O., (2006). Applying digital smart technologies for a sustainable architecture, in proceedings of CAADRIA 2006, Kumamoto, Japan, pp. 261-268.
- Sharp, H., Y. Rogers & J. Preece, (2007). *Interaction Design: Beyond Human-computer Interaction*, John Wiley and Sons Ltd.
- Shen, C., K. Everitt & K. Ryall, (2003). UbiTable: Impromptu Face-to-Face Collaboration on Horizontal Interactive Surfaces, in proceedings of UbiComp, Springer Berlin/Heidelberg, Seattle, Washinton, pp. 281-288.

Shenzhen Reidz Tech, (2010). LED Dance Floor, retrieved 20 July, 2010 from <http://www.reidz.com/>

Strehlke, K., M. Ochsendorf & U. Bahr, (2004). Generative Interfaces and Scenarios for Intelligent Architecture: A Framework for Computer Integrated Buildings, in proceedings of 7th Generative Art Conference, Milan, Italy.

Streitz, N. A., J. Geibler & T. Holmer, (1998). Roomware for Cooperative Buildings: Integrated Design of Architectural Spaces and Information Spaces, in *Cooperative Buildings. Integrating Information, Organization, and Architecture* Springer.

Streitz, N. A., J. Geißler, T. Holmer, S. i. Konomi, C. Müller-Tomfelde, W. Reischl, P. Rexroth, P. Seitz & R. Steinmetz, (1999). i-LAND: an interactive landscape for creativity and innovation, in proceedings of SIGCHI, ACM Press, Pittsburgh, Pennsylvania, United States, pp. 120-127.

Streitz, N. A., T. Prante, C. Muller-Tomfelde, P. Tandler & C. Magerkurth, (2002). Roomware: the second generation, in proceedings of SIGCHI, ACM Press, Minneapolis, Minnesota, United States, pp. 506-507.

Tennenhouse, D., (2000). Proactive computing, in *Communications of the ACM*, 43-50.

Tomohiro, F., K. Sakata, W. Yeo & A. Kaga, (2006). Development and Evaluation of a Close-range View Representation Method of Natural Elements in a Real-time Simulation for Environmental Design - Shadow, Grass, and Water Surface, in proceedings of eCAADe, Volos, Greece, pp. 58-65.

Uma Information Technology GmbH, (2012). uma SKIN™ Multitouch Display, retrieved 18 June, 2013 from <http://www.uma.at/en/press-center/press-releases/bally/>

UNStudio, (2006). *Design Models: Architecture Urbanism Infrastructure*, Thames & Hudson.

Weiser, M., (1991). The Computer for the Twenty-First Century. *Science American*, September 1991, 94-104.

Yoon, J., K. Ryokai, C. Dyner, J. Alonso & H. Ishii, (2004). egaku: enhancing the sketching process, in proceedings of SIGGRAPH, ACM press, Los Angeles, California, p. 42.

劉育東, (2007). *數位建築與東方實驗*, 天下遠見出版股份有限公司, 台北, 台灣.

邱浩修, (2008). 建築空間運用「實虛共構」觀念做為互動或溝通介面的設計與構築方法研究, 國科會專題研究 NSC96-2221-E029-028

日本 SGI 株式会社 (2005). 空間ロボット RoomRender. Retrieved 25 June, 2007, from <http://www.sgi.co.jp/robot/roomrender/>

