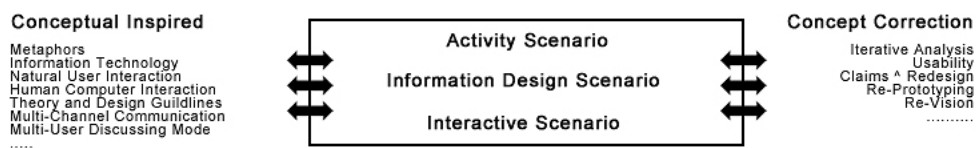


第四章 系統設計與實作

我們在上一章已完成第一階段的問題分析，也導出問題劇本來描述我們所關注的使用需求，同時提出一個系統的運作核心，也就是整合同步與非同步特性的多頻道溝通架構環境。緊接著，根據第三章有關劇本法的步驟描述，我們在此章節進入第二階段，也就是系統設計階段。這個階段將透過系統的概念發展、功能的設定等步驟逐一實現多頻道溝通模式架構之下的群組軟體。發展階段將以下列幾個大步驟來進行系統概念發展跟細部設計：



(圖 49) 系統概念與設計發展(可逆流程)

依照附錄(APPENDIX I) 所描述的劇本種類跟流程，本章節的寫作將步驟簡單分為系統概念發展與原型設計實作兩部分。系統概念發展這一節是對關鍵議題提出一個可行的解決方案並繪製出概念劇本，解決方案則包括上一章所提的多頻道溝通架構與虛擬紙張的隱喻。此時概念劇本呈現的是系統完成後的可能使用情境，讓系統真正開始進入實作之前，可以讓系統規劃者先針對可能需要處理的項目先行通盤的了解跟計畫，也就是讓本研究在進如實作之前可以先有部分的小測試跟實驗，規劃出藍圖，不僅對於系統規劃者本身可以概略性的掌握未來系統的面貌，也可以在這個階段描述系統的樣貌給專家學者是先知知道並擷取多方的意見，事先修正某些小問題點。最後再透過這一本概略的系統規劃藍圖，也就是未來的活動劇本來引導下一節的原型設計實作。原型設計階段則依照此藍圖所條列的項目，以資訊劇本跟互動劇本的所提的資訊架構、繪製腳本與版面配置等開始進行系統的實際設計跟實作，最後製作出能夠運作的系統原型以供最後的測試跟回頭修正。

4.1 系統概念發展 (功能藍圖)

系統概念發展也就是在系統真正進如實作之前的藍圖規劃(功能藍圖)，透過測試與撰寫概念劇本的方式可以用來描述一個理想中的設計藍圖，並解釋一個期望中的理想互動過程該怎麼進行接著對一些系統功能上的特性做清楚地界定。系統設計者便可以透過該劇本的最後的文字描述與重點工作項目進一步引導設計原型的介面細部設計。如同先前所提，Tang (1991)對於傳統設計環境的觀察已有提出一些基本原則，他認為一個合作式環境必須：

- (1) 允許所有的參與者都擁有共同的視角可以查看討論桌上的狀態，並且同時的可以直覺的參與討論桌上的活動，就像移動身子靠近討論桌或是白板這樣的直覺。
- (2) 支援使用者的手勢溝通與傳達。彼此的手勢必須是顯而易見的，並且伴隨著手勢的相關物件和語音必須被適當的提供。
- (3) 允許使用者本能地傳達，藉由透過建立設計物件如草圖、速寫來詮釋與溝通想法。
- (4) 允許無縫隙的混合工作檯面的行為(條列、草圖、手勢) 和功能(保存資訊、表現想法、媒介的互動)

而以自然的使用者介面(Natural User Interface) 所標榜的設計原則來解釋:

- (1) 分享空間(Idea shared space) 就像是一個開放式的書架或是白板這種媒介，設計者被鼓勵隨時地分享他們片斷的想法，並將想法用文字或是草圖記錄在這一類的空間上。並且同時也可透過媒介取得的其他人的想法來刺激自己進一步的靈感。
- (2) 系統必須允許設計者本能地、直覺地、不費力的、愉快地地將想法分享出去，通常對於一個慣用筆紙來輔助思考的建築設計人員來說，筆跟紙的介面是最快速直覺得可以將抽象的概念視覺化的表現工具，是最自然的使用者介面。
- (3) 好的系統應該讓設計者在靈感驅使下瞬間地去萃取出具體的想法，並在上面進行透明的編輯，以上一個想法為依據持續擴散的發想。也就是要能夠因應傳統設計者慣用描圖紙思考、註解和繪畫的需求。這個原始的且不斷循環的項目是本研究重視的部份，這個項目鼓勵設計者去試圖思考、聯想、想法連結(linking idea)並和由其他設計者所發想出來的設計想法(design artifacts) 產生刺激互動。
- (4) 當有具有決定性的想法藉由分享空間的靈感觸發而被衍生出來，系統應該提供一個頻道或是介面轉換可以順暢地去切換連結主、副頻道之間的情境，而不唐突的中斷檯面上的討論。

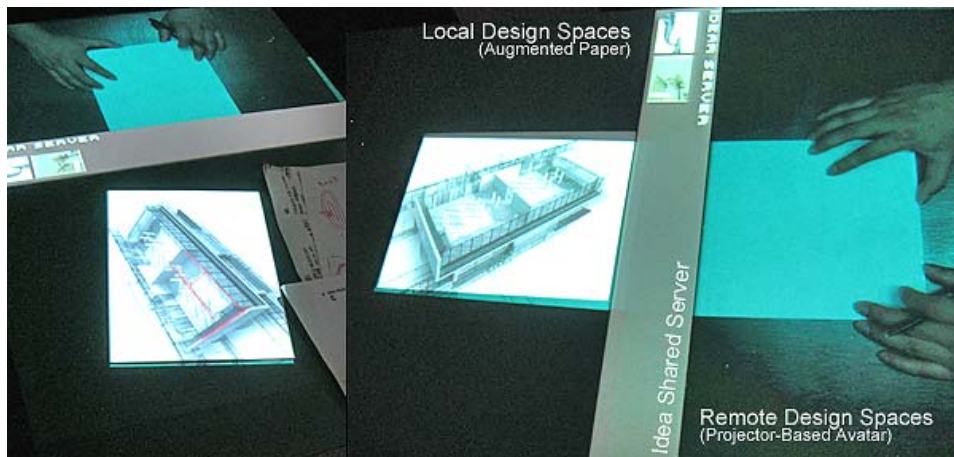
綜合這些基礎概念，在系統尚未明朗化與進行實作之前，我們參考相關案例中所使用的遠端視訊的技巧所設計的遠端合作討論桌，搭配一個簡單的實體紙板與擴增投影的裝置，透過劇情引導的方式邀請建築所設計組同學實際體驗系統可能的實體終端介面形式。並透過他們給予實值上的意見回饋作為未來系統設計中斷介面時的參考。比較詳細的方式是將紙張的虛擬影像(Dzmitry Aliakseyeu, 2003) 對應投影在實體的紙板並將桌子中央處設置一個代表共享空間的物件作為白板的隱喻，來置入遠端合作式設計環境中所缺乏的白板功能，設計師可以透過簡單的操作將想法傳遞到共有的空間上，而測試的主要目的是想知道這類的介面對於底下幾個現象的可靠性。

(1)想法的分享(Idea Shared Spaces):

我們以“push” 為隱喻來詮釋傳統設計環境的“分享”這個動作並跟設計組同學介紹 Pick-and-Drop 的操作方式讓他們試著實際操作。傳統上設計師臨時要與人分享或是討論一個自己的想法時，都會直覺的將自己手中的畫有概念紙張往外推到大家都可以注意到的地方。然而不然發現在現有的遠端合作式設計環境中並沒有所謂公私等區域劃分的概念，導致設計者除了視訊溝通的管道之外並

- 以多頻道溝通模式為架構的群組軟體設計

沒有一個可以即時分享的機制。而這個在遠端環境中所做的實驗就是要測試如果有這麼一個物件可以去支援這一類的活動，設計師在合作討論過程會如何透過這個物件來分享想法。這個測試讓設計者可以在討論進行的任何時刻將他私人的紙板推到代表共享空間的物件邊緣上，這裡以一個置放於中心中央的實體橫板代表這個空間，設計者並透過數位筆的點選(pick)想法草圖拖拉並置放到桌中代表共享的物件上方，此時不同地另一端的設計桌上的共享物件也會同步顯示。



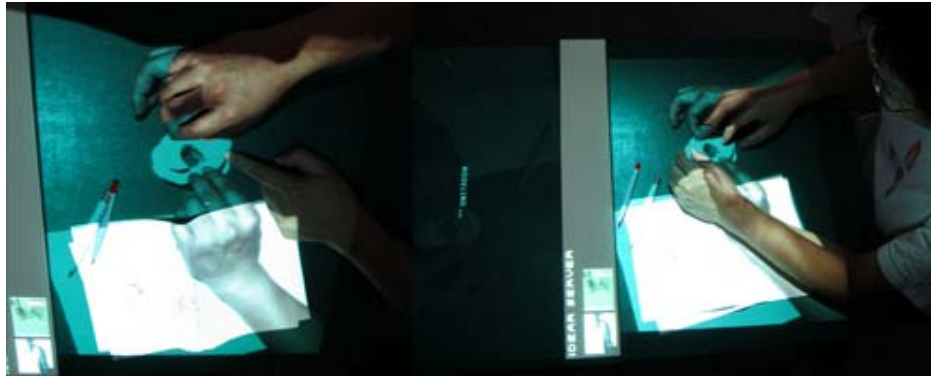
(圖 50) 非同步的想法分享伺服器概念

(2)反覆的靈感刺激(Inspiration repeatedly):

延續上方對於測試的描述，放在桌子中央代表共享空間的物件在我們的設定中它就是一個所謂的創意伺服器(Idea shared server)。這樣的伺服器是以背景頻道(Back-channel) 的概念去介入現有的遠端設計環境，透過非同步通道的方式讓設計者可以與其他設計者所貢獻的想法有所互動。當有這樣的機制之後，設計者會期待從這些分享的想法中孕育(Fostering)出一個新的聯想或是結合兩個之前隱藏看不出有直接關聯的想法成爲一個新的想法。然後重新將這個新的想法透過上一段描述的互動方式重新將新的想法推回到共享空間也就是創意伺服器上。過程中系統也允許設計者去重新思考並透過透明疊層的草圖編輯，類似透明的手繪草圖被廣泛地使用在概念發想的過程。

(3)黏土模型的表現(Modeling):

這個以視訊傳輸爲主的測試環境，方便讓我們透過簡單的影像疊合作進一步的測試。我們知道在設計早期概念發展階段，黏土模型是除了手繪草圖以外最被廣泛運用的概念呈現工具，當僅能表現二度空間資訊的草圖或是文字不足以充分及完整的表現腦中的抽象的想法，設計者往往會使用黏土模型或是簡單的模型板直覺的、清楚的去表現三度空間的一些資訊。而這個測試便是利用視訊的特性，將另一端的影像以同樣的視角直接正投影在本地端的设计桌面上，讓本地端的设计者可以透過同樣的視覺角度來看另一端解釋一個空間概念。這樣的介面允許設計者改變自己的視點，以對方的視點來觀看，並帶領設計者去移動雙手去對對方的手勢，雙方的手勢影像是重疊的。



(圖 51) 遠端擴增實境中的黏土表現工具測試

(4)遠端察覺(Awareness in Remote Design Environment):

透過視訊聯結遠地兩端的影像並直接疊影在桌面上的情況之下，手勢本身就同時具有察覺(Awareness)與指引(Pointing)的功用，不管是手部移動的意圖察覺或是指頭引導對方手勢，都可以直接被傳遞，就跟以往同桌面對面交談一樣。

從這個簡單的測試跟上面幾個對於合作式設計環境的概念描述，我們對於整個系統的樣貌開始比較有點雛形。從這個測試的回饋意見中，我們了解到設計者很習慣於這一類採用擴增實境的手法所設計的筆跟紙板介面，虛擬紙張的影像被投影在對應的紙板上，而紙板此時不僅提供實體碰觸的熟悉感又能兼顧虛擬紙張即時編輯跟傳輸的特性。這樣的終端介面的設計桌形式很適合被用在遠端設計環境的規劃藍圖中。而將伺服器不同步分享概念整合在現有的遠端視訊環境中，不僅可以提供一個原本純視訊溝通所缺乏的類似白板的共享空間，透過簡單的介面讓設計者直覺的使用這個伺服器就像在使用白板一樣的自然。

以筆跟紙為主的實體終端介面的型式可以有很多種，上面的步驟我們選擇性的採用擴增實境跟虛擬紙張的方式進行介面的案例練習，主要的測試項目除了實體介面本身的適用性之外，也是第一次嘗試將不同步運作的想法伺服器(ISS:Idea Shared Server)的概念整合在同步的視訊設計環境中，透過對虛擬紙張跟紙板的簡單直接移動，就可以把虛擬圖層上的某個想法拖拉到想法伺服器上，設計者的感受是它可以在不被干擾的狀態之下迅速的且直覺的將想法分散出去，也就是分享到創意伺服器上。這個簡單的測試讓我們了解到只要透過實體介面的改善，要在現有的同步溝通全部廣播的溝通架構之下去整合一個不同步的溝通環境是可行的，而且是可以不互相干擾的。

也因此，緊接著我們進一步開始設定一些所謂多頻道溝通架構之下所該有的軟體運作流程，因為我們希望進一步去嘗試不同的創意伺服器整合的型式，並且開始思考要如何在現有顯性的主頻道之外去提供額外的多組隱性的討論頻道為目的。我們這次改用混合式的搭配的實體介面進行軟體

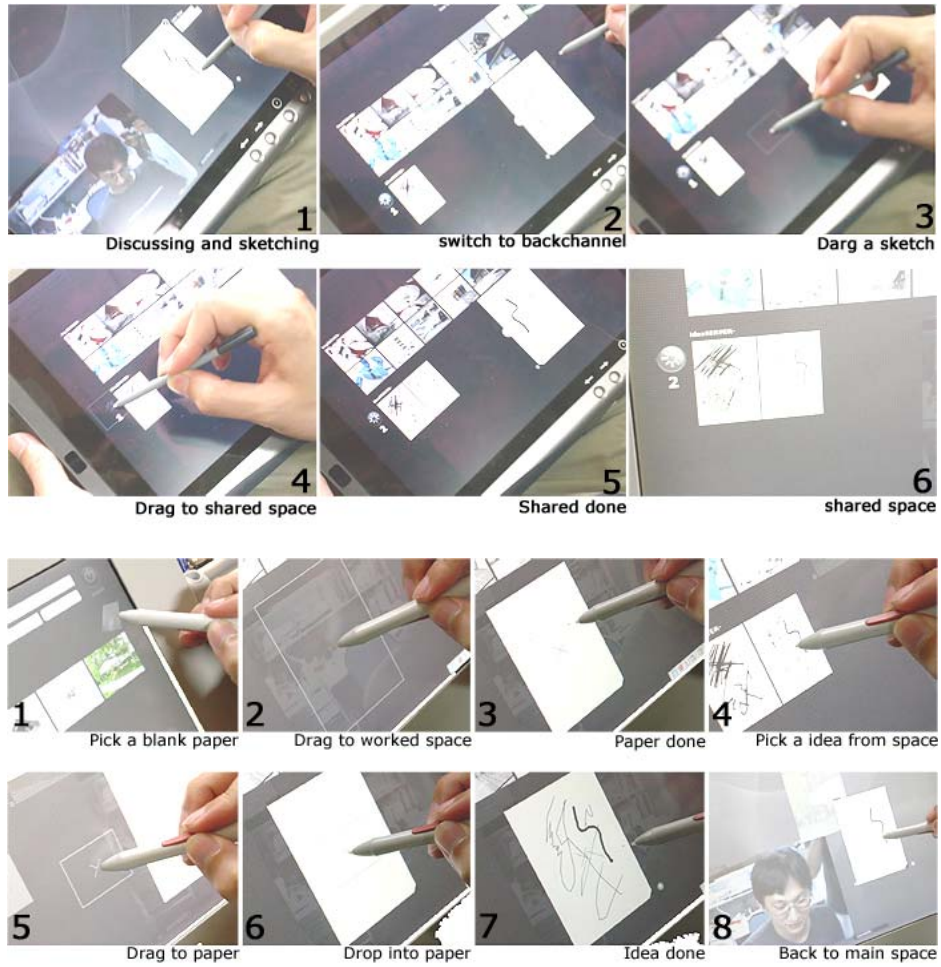
- 以多頻道溝通模式為架構的群組軟體設計的

測試：A 端採用 Tablet PC；同時不同地的 B 端則採用繪圖板、桌上型電腦的型式跟 Tablet PC 的整合環境。當 A 端透過 Tablet PC 向 B 端提出設計討論的要求時，B 端的 Tablet PC 就像是一個 Ambient Display，以不影響不干擾的方式提醒 B 設計者 A 這個要求。當 B 拿起 Tablet PC 並且進行登入，彼此的連線建立。這樣的流程看似比虛擬紙張的操作方式更顯得繁瑣，然而所使用的硬體設備是以現今設計環境隨手可得的硬體終端當介面測試。而此時的虛擬紙張的概念不是以擴增實境的投影方式，而是透過軟體的直接撰寫，以類似無框視窗的範圍來替代許擬紙張的概念，透過筆的直接點選、拖拉跟拖曳幾種模式來設計虛擬紙張的切換模式以及圖層編輯的效果。使用者可以本研究所預設的幾種模式進入編輯、切換代表不同頻道的紙板、進入想法伺服器。而測試項目的結果，就虛擬紙張的軟體介面來說，使用者本身就已習慣筆跟紙的介面，對於長期使用類似視窗介面的方式也很熟悉因此顯得容易上手；而切換模式的觸發方式本研究則試圖將現實經驗中所代表的使用紙張的行為直接隱喻例如：翻轉紙張代表切換背景頻道、抽掉紙張代表移除虛擬紙張等操作模式，也就是如何透過筆的操作介面來達成對虛擬紙張的操作。而主要的測試項目以多頻道溝通模式來說，此一版本針對軟體的環境測試中，使用者的回饋是紙張對應頻道的概念沒這麼強烈；就他們所能感受的來說他們可以新增不同的虛擬紙張的板並且可以在上面做簡單的草圖編輯，然後當他需要將某想法分享出去，也就是放到白板上時卻需要暫時離開目前的畫面，翻轉紙張切換到屬於想法伺服器的後台去，才能分享與查閱別人丟出來的想法概念，紙板之間的連結性也反而沒採用擴增實境時這麼的強烈，即便是本研究事先向他們描述虛擬紙板代表一個頻道時。而這也開始讓我們思考，兩種不同的型式之下所呈現的媒材特性，是哪一個環節讓受測者有這樣不同的體驗跟感受？就系統該有的功能設定上，是不是會因為使用的實體終端介面不同而必須跟著改變此功能對外的呈現機制？如何在純軟體介面中去強化不同使用者虛擬紙板之間的連結性？重新定義這個創意伺服器的意義跟方式？

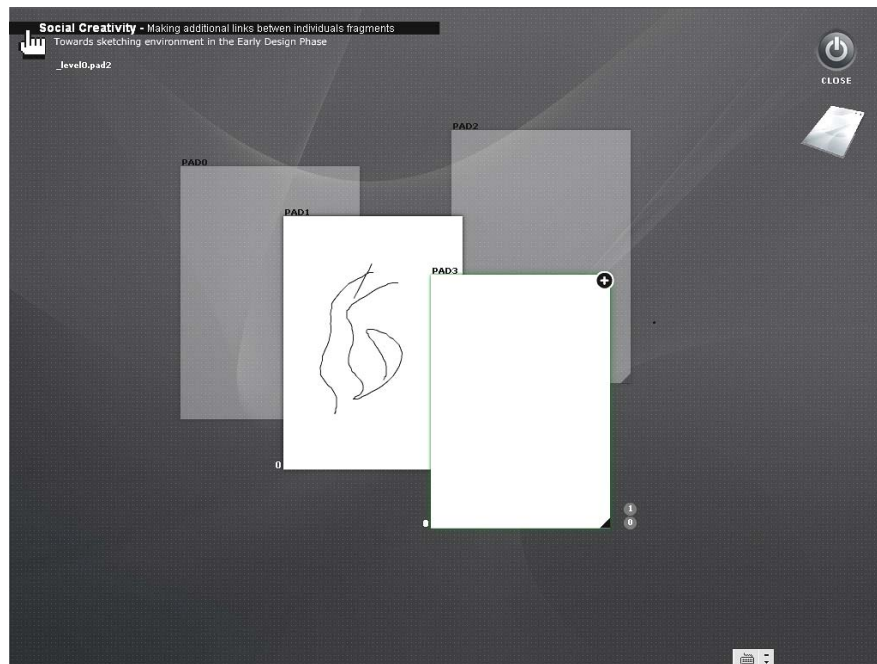


(圖 52) 先期針對實體介面與想法伺服器的實驗

- 以多頻道溝通模式為架構的群組軟體設計



(圖 53) 在 Tablet PC 上實驗 Idea Shared Spaces 的介面操作流程



(圖 54) 紙板與想法伺服器的資訊交換

有了這些先期測試所留下來且可以供思考的素材，我們著手開始進行多頻道溝通架構下的軟體功能設定，而對應的實體終端則先以一般容易取得的繪圖板設備為本研究主要的設定環境，目的則是檢驗多頻道溝通架構的運作方式：

多頻道的溝通架構：

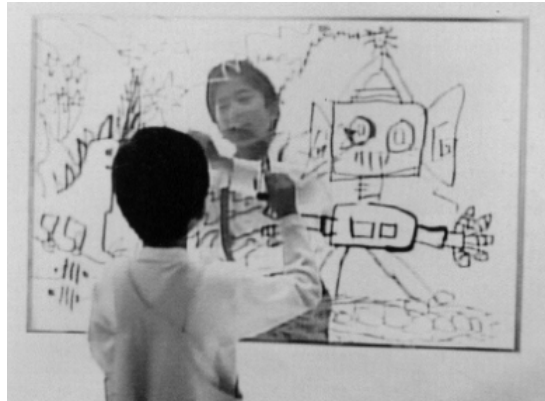
在系統的功能與核心架構基本上必須得提供兩組頻道類型之間轉換的流暢性，設計者使用系統時可以在共享空間的視訊討論、想法分享、想法表達與私人區域的個人發想、草圖編輯等不同的使用機制之間順暢地且自然的切換又能維持察覺頻道之間狀態的能力，縮減現有遠端合作式各種不同媒材之間的轉換所造成的短暫中斷。

繪圖板為基礎的實體終端：

我們知道採用擴增實境搭配虛擬紙張投影的實體終端，在使用認知上可以強化設計者對於紙板等同於頻道；也就是對紙板之間的連結性有比較深的感受。然而因為這樣的硬體設置往往需要更多的準備工作，包括透過識別標誌用來判斷實體紙板的位置、轉向，並且還得經過影像的運算虛擬紙張該投影的相對方位大小等，才有辦法對應的投影在紙板上。就理想中的狀態本研究似乎應該選擇採用這樣的機制。然而，本研究是以提出一個新的多頻道溝通架構並試著嘗試它的可行性為主。對於實體終端所帶來的不同感受或是設計方式，是未來以此多頻道架構概念所衍生的相關對於實體介面的研究。因此本研究決定採用目前設計環境中常用的繪圖板設備跟軟體環境的提供，作為主要的功能設定，也就是去開發一個群組軟體介面的方式來達成。

強化虛擬紙板的連結性：

使用電腦的圖形式介面來模擬虛擬紙張的方式，最大的限制就是必須牽就於螢幕的範圍大小限制，這跟實體空間中使用紙板的移動性相比之下是最大的缺陷。加上使用者往往已經習慣於視窗式的介面，對於電腦視窗背後遠端連結的感受反而沒實體介面這麼深，就好比是一個駕駛開車開久了，他認知中只知道只要一轉動方向盤車就會跟著移動，而對於背後運作的機械的運作感受沒這麼深刻。同樣的，這對於我們想重新灌輸使用者，紙板等同於頻道的觀念時有很大的阻礙，經過初步測試的結果，使用者仍然只會將虛擬紙張當作只是一張可以進行數位編輯的紙張。然而在我們對多頻道溝通的定義中，每一塊紙板所代表的就是一個通道、一個頻道、一個具有連結性的溝通紙板，使用者在編輯某一塊紙板的同時，只要是他授權使用該頻道給其他使用者，該使用者就可同時對該頻道的紙板進行共同編輯與閱覽，這就是我們所謂的紙板等同於頻道的觀念。我們歸納之後發現照成使用者對於虛擬紙張缺乏頻道觀念的最大的主因，在於電腦上虛擬紙張的即時更新性就是 *What you see is What I see* 的環節上，當共同編輯的一方有了更動，頻道另一方相對應的紙板應同時更新，對於多人同時在同一張紙張上進行討論的感受會比較深刻(圖 55)。



(圖 55) 虛擬紙板有關連結性的隱喻 (ClearBoard, Hiroshi Ishii, 1994)

想法伺服器的定義：

以紙板等同頻道的觀念，加上多頻道溝通架構是由一組單一主頻道與多組副頻道所構築而成的系統架構。這樣的架構之下某方面主頻道所對應的屬於討論群體的共有紙板，在遠端設計環境中它所扮演的就是一個共享白板的角色。在新的系統功能設定中，它可以取代原先幾種版本中所設定的想法伺服器等機制。也就是透過一個新開的虛擬紙板，並將該板的使用授權開放給所有的使用者，大家都可以隨時在這個板子上進行想法的分享跟討論等。透過這樣變通的方式，可以減輕設計者在前一個版本的測試中所遇到的共享與私人工作檯面之間切換的頻率太過於繁複或是整各畫面轉換所造成的短暫中斷等。

4.2 系統原型設計 (介面細部設計)

這裡我們將概念劇本所引導出的功能定義與設計藍圖轉換為一個能夠實際運作的原型設計，進一步將系統的基本介面、資訊配置、互動流程等細部給勾勒出來。包括採用隱喻的設計手法或是整合自然的使用者介面(Natural User Interface)概念等等，都將在這裡做更詳盡的描述。一個良好的使用者介面，應該符合使用者對自己、他人、環境和一些他們已接觸過的事物並經由經驗或訓練所產生之心智模型。一個系統的使用心智模式大都來自於使用者認為他能做什麼與可見到的部份，也就是介面應該要有一些提示可以誘發使用者可以怎麼去操作，這個提示包括使用者本身認知可以找到的到一個熟悉的對應方式或是進一步適應的方式。而以筆式系統所開發的介面來說，跟一般傳統滑鼠的操作介面有某方面的相似度包括點選、拖曳、釋放等(Pick&Drop) 的基本操作，配合虛擬紙板本身多處不同預設的感應區塊，點選的區塊的不同對應不同的功能觸發。系統則將以這些觀念為主進行介面的配置工作：

引喻的設計手段(Metaphor)：Norman(1988) 認為系統的設計，應該使用真實世界與使用者腦中的知識，以減少使用者的負擔，使用隱喻是讓電腦等系統與使用者的認知產生直接對應(mapping)的一種方式，直接將現有的使用認知模式透過引喻移轉的方式讓使用者用既有的習慣去熟悉一個新的操作介面。因此適當的隱喻能夠讓系統與使用者之間的陌生間距縮小。

筆式的繪圖環境：本系統的介面，是以筆式數位環境為概念劇本發展的主要使用場景，因次我們嘗試移轉設計師對於紙跟筆的既有認知模式到本研究系統的介面設計上，透過類似虛擬紙張的圖形式介面來讓設計者自然的操作，透過幾個常用的動作例如翻轉、抽掉作為引喻，以設計者熟悉的使用心智去規劃系統的編輯介面，底下描述系統介面的一些設計基本原則跟操作模式：

(1)拿紙的隱喻

我們捨棄選單點選新增紙板的方式，透過一個拖拉的方式，讓使用者從一個代表性的描圖紙的符號直接拖拉虛擬的紙張到工作檯面上。在傳統的討論情境裡，使用者臨時有想法或是欲和身旁的設計夥伴表達一些想法時，習慣性地從視線前方附近隨手拿起一捲描圖紙筒並撕下一張紙開始透過草圖來描述自己的想法。介面的細部設計中直接將這個動作透過點選拖拉的動作讓使用者迅速地且隨心所欲得新增任何的虛擬紙板。

(2)翻轉的隱喻

虛擬紙張必須允許設計者迅速且直覺的方式翻轉切換私人的工作空間跟共享桌面，並且是在以不干擾主頻道討論的情況下進行。這在前一個測試版本的介面設計中，透過紙板的翻轉直接將畫面半透明反轉到想法伺服器的畫面中，半透明是為讓設計者仍然能夠保持對主頻道的察覺能力。然而這樣的方式經過初步的測試仍然會照成使用者的短暫中斷，加上新的系統可以透過紙板對頻道的觀念，直接作為主副頻道之間的區別。因此，在新的系統所規劃的介面設計上，這個翻轉的意義成為只有當設計者需要參考圖庫時的一個私人切換功能，透過筆在紙板的邊緣感應區內向外拖拉，畫面半透明轉到共同的圖庫參考區，同樣拖拉選取可以參考的設計圖稿到該紙板上，參考圖自動變成該板新增的圖層。

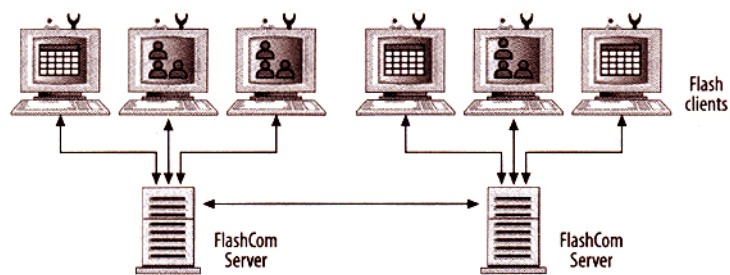
(2)抽掉的隱喻

透過筆點在紙板上下緣的感應區內向下拖拉(內縮)，便可把紙板從工作台區域內刪除。這是從設計師在跟夥伴討論時往往會有的一個習慣性動作，當一張描圖紙空白處不夠使用時，他會把紙張移到往旁邊放，或是直接從下方抽掉丟掉，這是一個很習慣的自然反應。

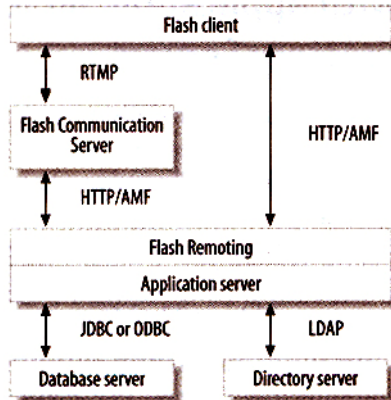
4.2.1 系統開發環境與平台

一開始考量系統可能需要即時的遠地兩端進行影像、聲音與草圖資料的傳輸，我們可以方便安裝設定的 Adobe (formerly Macromedia™) Flash Communication Server 作為系統核心的影音傳輸伺服器，負責傳遞多個終端應用程式之間所需要的串流。終端的軟體環境則使用 Macromedia™ ActionScript2.0 的物件導向語言與 Flash 作為系統介面開發的環境，比使用來實作多頻道溝通架構之下的系統可能的操作方式。而實體的終端介面，本研究以現有的設計環境中容易取得的繪圖板、平板電腦等裝置進行系統的實作開發。

本研究是採用之前的 Flash Communication Server 1.5 (現已改名為 Flash Media Server) 提供的免費版本進行測試。它的核心元件需要獨立安裝伺服器在有對外開放連結能力的電腦上做為影音串流的傳輸中心，並透過 Adobe (formerly Macromedia)自訂的 RTMP: Real Time Messaging Protocol 協定與客製化的應用程式端口(Flash Client) 進行溝通，只要透過簡單的核心類別函數便可以簡單的向伺服器提出要求，方便達到影音串流、共享物件所需要的資料傳遞等一些基本功能。從它的運作架構來看，他是以單頻道廣播的方式向所有登記的終端主動更新畫面，不是一個多頻道溝通架構發展而擁有可以分辨頻道差異或個別通知更新概念的系統。雖然即便不管任何一個終端有所更動 FCS 都會全面呼叫所有終端對應的函數進行同步更新，然而我們仍然可以採用這現成的溝通架構去加入本研究寫在終端程式上的判別程式作為頻道管理的核心，透過隱藏與顯示、封鎖跟開放的方式來解決紙板授權的機制，讓終端使用者可以僅對開放給自己使用的某些紙板通道進行編輯，而對於尚未獲取授權的紙板則無法閱覽跟編輯，我們將進一步描述如何達到這樣的多頻道溝通架構。



(圖 56-1) Flash Communication Server 現有的溝通架構



(圖 56-2) Flash Communication Server 現有的溝通架構

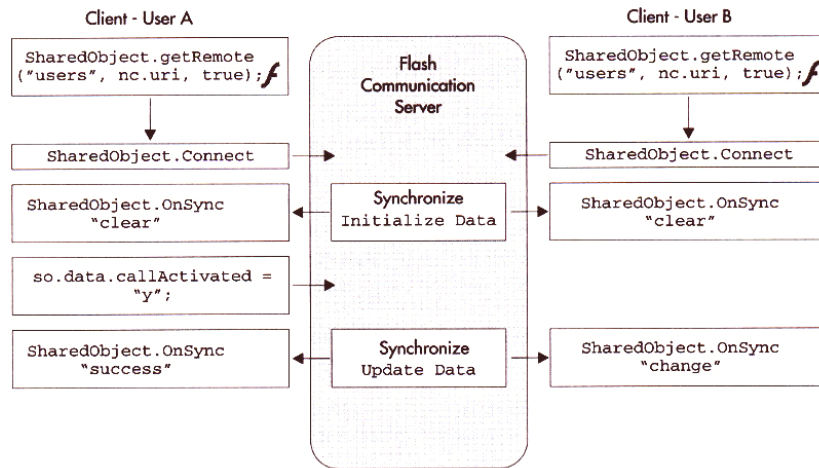
4.2.2 系統架構

本系統的架構主要是要實現多頻道溝通架構的可能性，至於有關本研究提出的這個架構，已在上面第三章多頻道溝通架構的段落作了很詳細的定義跟描述，主要的特色則可以歸納如下：

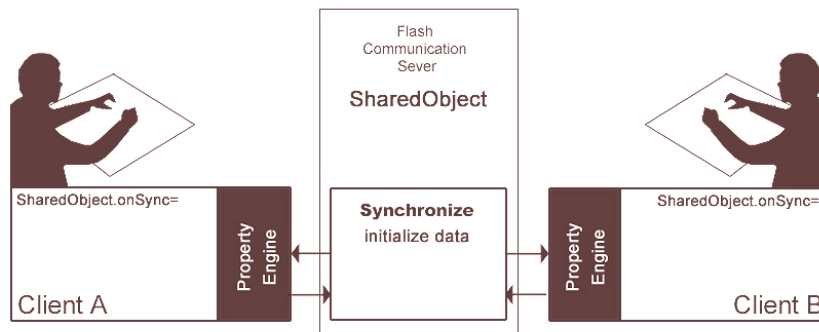
1. 一組單一的同步主頻道空間跟多組可以同時運行的非同步副頻道空間所組成。
2. 每一個紙板都有屬於自己頻道的屬性，像是獨立的閱覽權限名單、草圖的筆跡紀錄等。
3. 該架構之下的伺服器端會依據紙板該頻道的個別屬性、授權的名單各別通知更新。
3. 每個終端都可以新增屬於自己的副頻道。

接著我們將描述實作階段要如何在 FlashCommunicationServer 現有的同步架構上去實現多頻道溝通架構。因為我們從 FlashCommunicationServer 現有的架構圖可以知道，他是靠伺服器端的一個同步機制在接收跟通知所有終端應用程式並進行同步更新。這樣的方式之下，每個終端都被視為對等，且沒有所謂個別屬性差異的觀念，因此不管任何一個終端有所改變，所有其他的終端都會收到更新的要求。這對於我們要以此現有的系統去實現多頻道溝通的模式似乎有點衝突也很難著手進行，因為它的開發環境並不是一個開放式的架構，僅能透過設定好的函式類別來與伺服器端溝通。而這有兩個方式可以達成，一個是透過編寫存在伺服器端的描述語言 (Server-Side ActionScript) 副檔名是 *.asc 的程式或是直接使用終端的 (Client-Side) 的程式撰寫來達成屬性控制。而本研究的應變方式則是用一種類似匝道控制的邏輯與辨別屬性的方法使終端收到同樣的通知但卻做出不同的反應，依照終端本身登入時所指定的 id 與操作過程由伺服器端對於每個開通頻道指定的屬性與識別 id 作為辨別的依據作個別的回應(圖 57)。

- 以多頻道溝通模式為架構的群組軟體設計



(圖 57) Flash Communication Server 的同步溝通架構



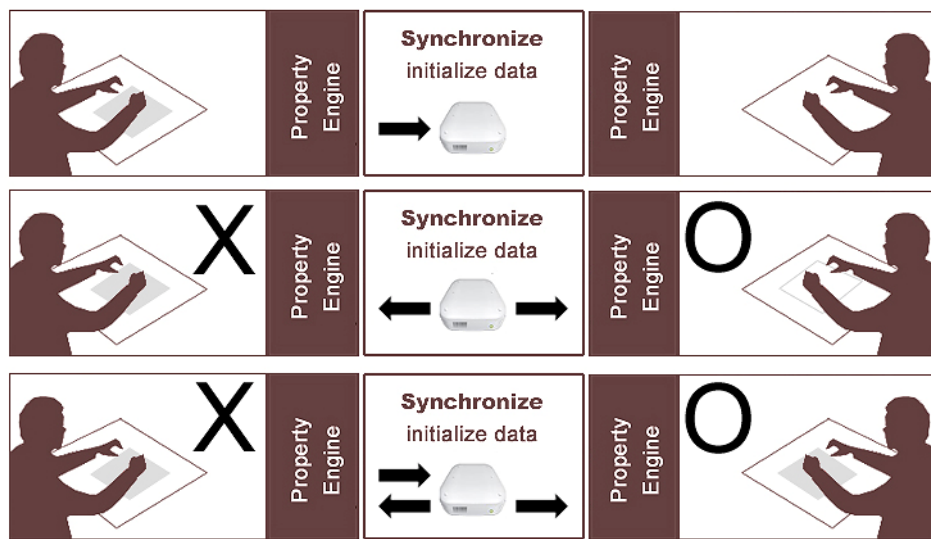
(圖 58) 一個基於同步架構所因應的非同步多頻道設計

4.2.3 系統實作

整個運作系統包含寫在伺服器端的程式(main.asc) 與終端介面內部的紙板編輯程式跟頻道管理程式等。伺服器端的程式主要用以管理登入的資訊、分配指定使用者 ID 跟建立存在伺服器端的共享物件(Shared-Object)，所有相關所需要的參數都會被初始化並存放在在這個共享物件中，透過屬性指定的物件方法 Object.data 可以直接進行指定而透過 SharedObject.getRemote() 的方法進行遠端存取。這個共享物件事後會以 *****.fso** 的檔案格式自動被存放在伺服器端的指定目錄中。而終端的應用程式除了有處理紙板相關編輯例如新增圖層、圖層管理、筆跡紀錄、介面切換等主要基本功能之外，主要就是跟伺服器端程式相呼應的頻道管理核心、共享物件的傳遞核心兩大項：

(1)頻道管理核心：

包括跟伺服器建立連線、初始化遠端共享物件、視訊串流傳輸連線、同步判讀的核心程式等。當終端應用程式第一次被開啓的時候，系統的連線元件首先會主動要求使用者輸入伺服器連線管理所需要的識別名稱，並將使用者輸入的名稱送到伺服器提出登入系統的要求，當伺服器視登入人數狀況同意該終端登入時便會配給該使用者一個獨立的 Channel ID，並通知該終端可以開始進行溝通所需要的參數初始化與設定、建立遠端物件的連結以及初始化視訊連線等。而整個頻道管理核心最重要的機制就是同步判讀的核心程式，也就是系統架構中所提到的應變方式，同步但依照各自的屬性作出不同的反應的部份。這個部分的判斷被放在終端準備接收伺服器呼叫的 onSync() 函式段落中(圖 59)：



(圖 59) 同步判讀核心流程圖

整個判讀的邏輯簡單的敘述可以分為三大主要方塊：

新頻道的註冊 當有任何一個終端向伺服器要求註冊一組新的代表頻道的紙板，此時的存在於伺服器的共享物件的 registerMode 模式會被開啓，並向所有的終端呼叫 onSync() 的同步更新程式，此時分屬不同終端的程式會依據自己的身分做 idCheckString() 的判讀而有兩種迥然不同的反應：一種是提出新頻道註冊的該終端將會判讀這是來自於自己的更動訊息而主動跳過；其他的終端則會判讀是屬於別人的更動而跟著執行 addFollowPad() 函式來向伺服器要求該頻道相關的註冊資訊，接著主動新增一個相對應的頻道。此時跟著新增的頻道紙板並不會直接顯示在終端，原因是該頻道目前是私人的頻道，提出該頻道註冊的終端使用者尚未授權予任何人對該頻道擁有閱覽跟編輯的權限，因此頻道是以隱藏的方式存在在所有的終端。

紙板的更動更新 當有任何一個終端對某個頻道的紙板進行編輯產生更動，這個訊息同樣會透過終端主動告知伺服器這個狀態的改變，此時伺服端的共享物件的 `updateLayerMode` 會被啟動，並向所有的終端呼叫 `onSync()` 的同步更新程式。每個終端同樣有兩種不同的反應：“是不是來自於本身的更動通知?”。如果是自己這端提出的更動訊息則忽略不做任何反應，如果不是就向伺服端的共享物件提取相關的更動參數，透過 `updateThePad()` 函式將某個指定的且已被更動的紙板頻道做同步更新的動作。不管是有允許授權看的道的紙板或是沒有權限隱藏中的紙板，都有可能接到通知而進行同步的動作。

頻道的閱覽權限 到判讀程式的最後一個步驟，每個終端都會向伺服端要求目前已被註冊並開啓的紙板頻道陣列 - `padArrar`，該陣列同樣被存在遠端的共享物件中，裡頭並記錄著每一個頻道的屬性包括是由誰開啓註冊、目前的授權閱覽的名單陣列 - `padArray[i].registerUserArray` 等資訊。透過這個屬性，終端取得所有紙板的資料跟使用者名單陣列之後，便會依這個字的情況屬性分別做不同的判讀，決定該終端的使用者對於某些頻道的使用權限是否被允許，進一步調整該頻道的顯示公開與否。

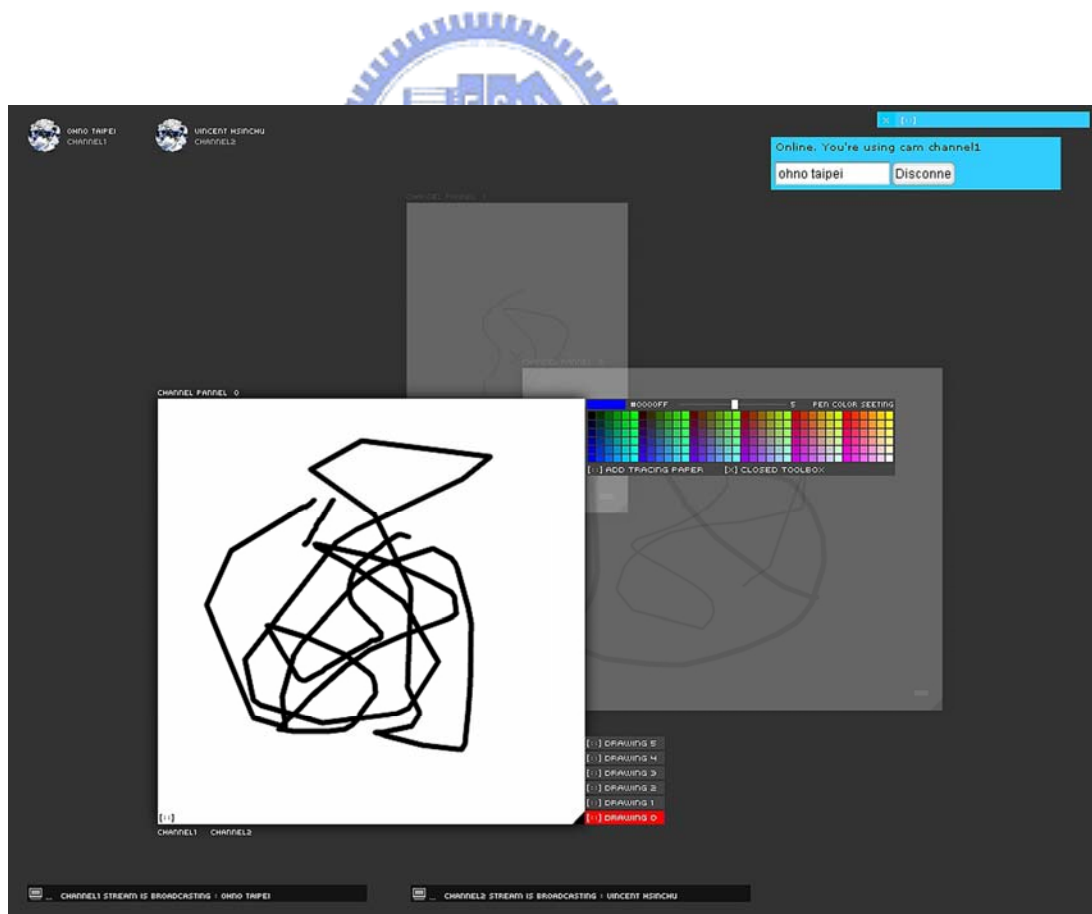
(2)共享物件的傳遞：

所有的終端跟伺服器來往的依據都是根據連線之後就已初始化的共享物件(Shared-Objects: `**.fso`) 所有之後需要交換的資訊都會不斷透過這個物件以 `draw_so =SharedObject.getRemote()` 的方式去建立跟取得，包括幾種預設的模式切換狀態、已註冊頻道陣列、紙板筆記陣列傳遞、使用者資訊等等。因此這個負責物件傳遞的元件對於整個背後運作的機制佔有很大的使用頻率，它的初始也是透過 RTMP 協定去與伺服端連繫，透過 `client_nc.connect("rtmp://Server IP/scfso")` 的方式在伺服端的應用程式資料夾 `scfso` 中建立連線，再使用 `SharedObject.getRemote()` 的方法去指定該物件。

所以整個系統的實作部份便是以這些原則去撰寫細部的程式部份(APPENDIX II)，在 `FlashCommunicationServer` 所提供的現有同步架構之下去應變實作一個整合非同步溝通的系統。當然理想中的多頻道溝通實作方式應該是採用比較資訊工程的作法，直接在伺服端的部份就可以針對每個不同的終端應用程式的屬性作判讀來選擇廣播同步與否的動作，而不用先同步到終端之後再由終端反向跟伺服端要求相關資料來判讀。這點是因為本研究主要是以測試溝通架構的可靠性，因此初步先選用方便但非開放開發環境的 **FCS** 作為主要實作環境。

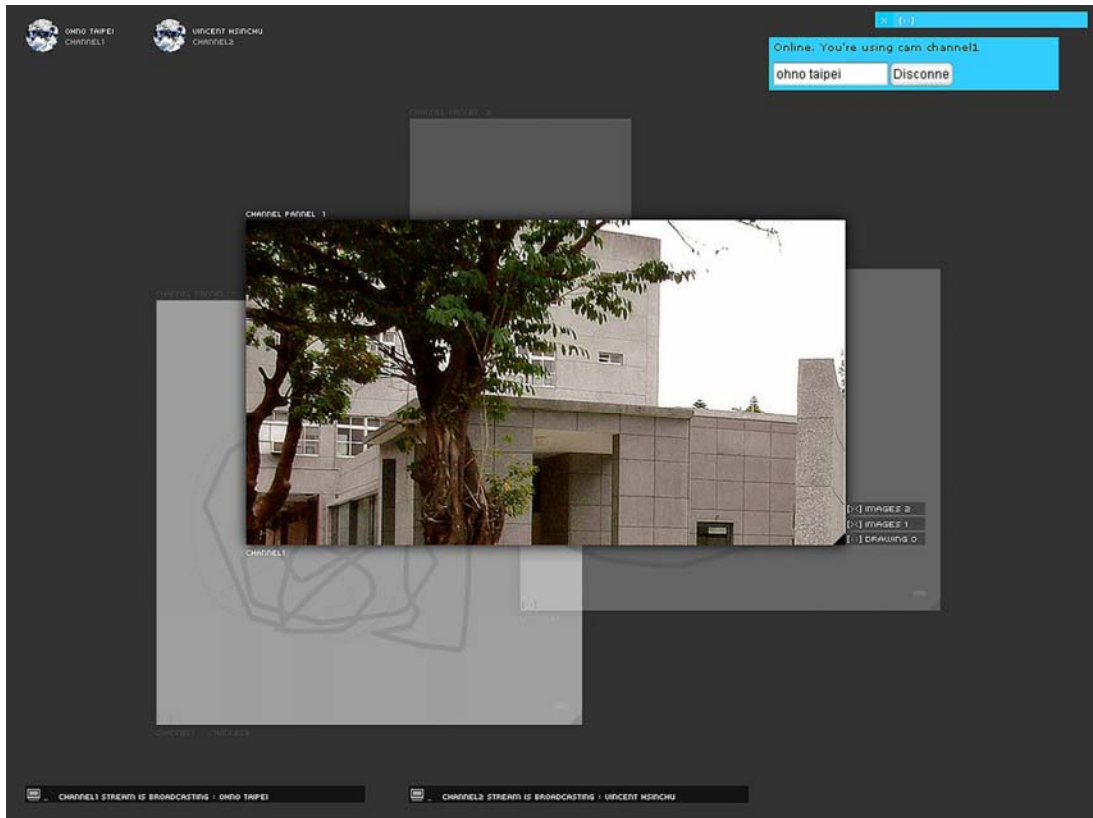
4.3 系統介面說明

在 4.2 節描述整個系統的原型跟背後的運作流程之後，接著在第三節我們透過劇本描述的方式來解釋系統前端的介面部分如何操作(圖 58)。當使用者透過終端應用程式登入系統之後，就會進入終端的主畫面，大致可以分為右上角的工具列、左上角有參予討論的使用者名單、下方的視訊窗口以及中間的工作檯面區。使用者可以透過工具列任意地從 [::] 符號所代表的描圖紙筒進行拖拉一張虛擬的空白紙板到中央工作檯面區的任何地方。就像下圖所呈現的，一個紙板代表一個頻道、一個通道，使用者可以在上方進行書寫編輯跟對紙板縮放，並可以透過拖拉左上方的其他使用者符號到欲分享的紙板頻道上，該紙板便會透過伺服器授權給予對方進行閱覽並共同編輯的權限。



(圖 60) 系統終端主要操作介面

- 以多頻道溝通模式為架構的群組軟體設計



(圖 61) 系統終端主要圖層編輯介面

紙板的縮放：

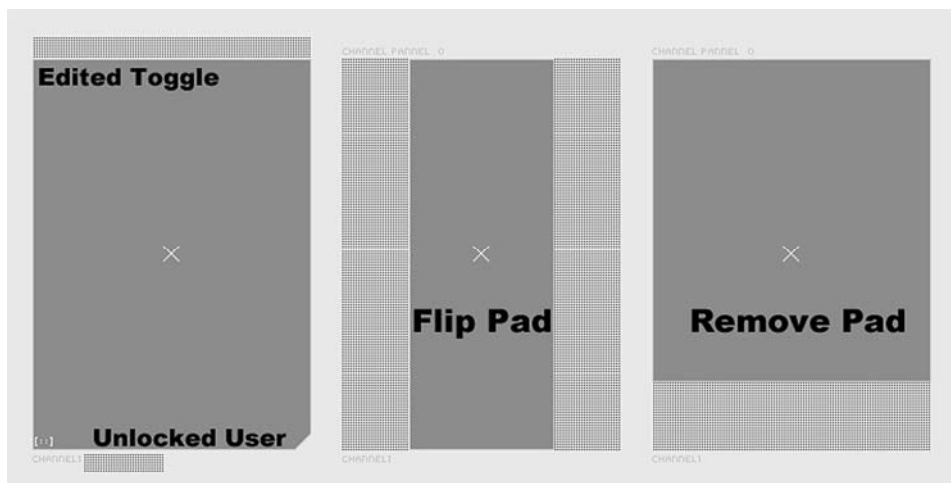
每一張紙板都可以像視窗系統一樣的進行縮放，這是虛擬紙張可以勝過實體紙板加擴增投影等實體介面的優點之一。因為它允許使用者針對各自的需要即時直覺的調整紙板的不同大小。



(圖 62) 紙板的縮放

紙板的感應對應操作切換：

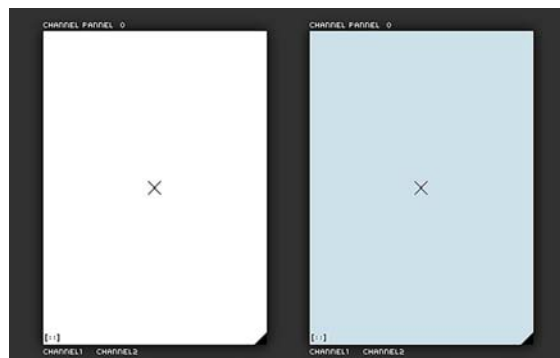
Edited Toggle 允許使用者將筆在點這個感應區塊內，向外拖拉(DragOut) 可以切換該板的狀態是處於編輯或是移動的模式。**Unlocked User** 則可以讓使用者將之前開放授權給其他設計夥伴編輯的名單，直接透過點選名稱便可以將他的權限關掉。**Flip Pad** 在紙板的左右兩側有相對應的感應區域，只要將筆尖在這些區域向外拖曳到紙板範圍之外就可以將板直接切換到參考圖片區。最後，**Remove Pad** 的感應區域，同樣是拖曳會將紙板從工作檯面上整個移除，這個狀態會被通知到伺服器端再透過伺服器端去同步所有其他的終端將該板移除(圖 63)。



(圖 63) 紙板的所預設的感應區域

紙板的授權與差異：

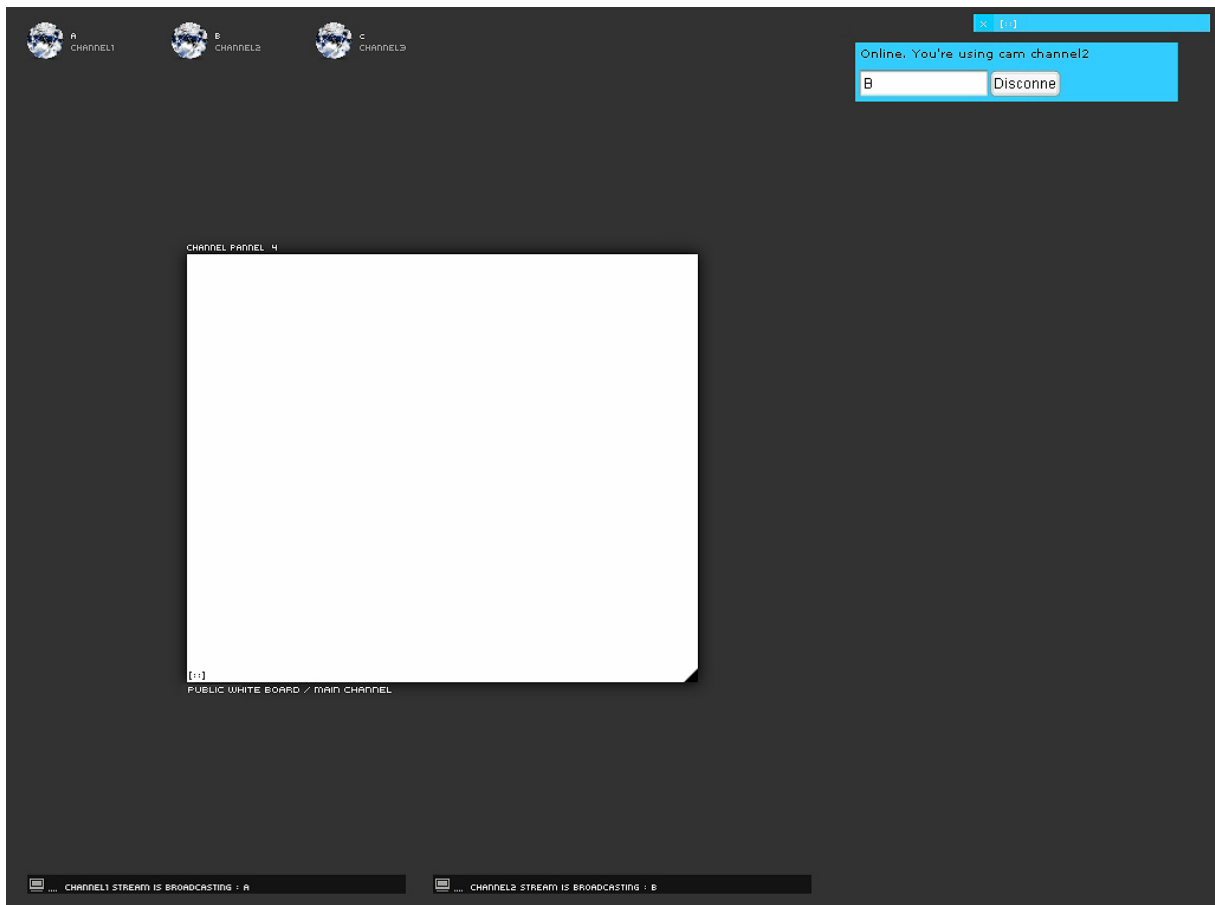
當使用者 A 在他的桌面新增一塊紙板並向伺服器端註冊這個頻道時，伺服器端會同步通知所有的終端包括 B，此時的終端 B 判讀該頻道並非是自己註冊的，因此會自動新增對應該頻道的板子但不將它顯示給使用者 B。當使用者 A 想跟 B 討論想法，並透過它工作檯面左上方的使用者名單將 B 加入某個頻道來授權他可以對該板進行共同編輯時，此時 B 端相對應的紙板便會主動顯示給使用者 B。然而為了讓板子有所區別，自己註冊的板子本研究以白色的代替，而因為其他使用者授權而獲得分享的紙板則是以淡藍色的底來區分(圖 64)。



(圖 64) (左圖) 擁有主控權的頻道 (右圖) 被授權使用的頻道

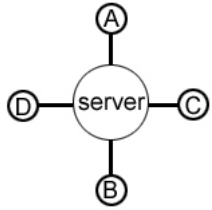
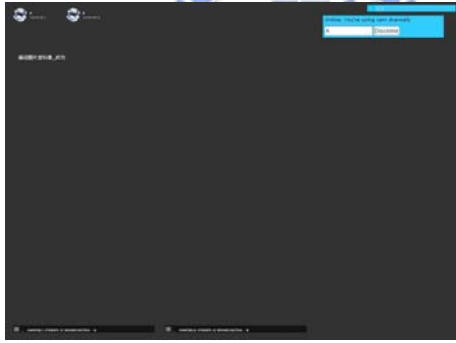
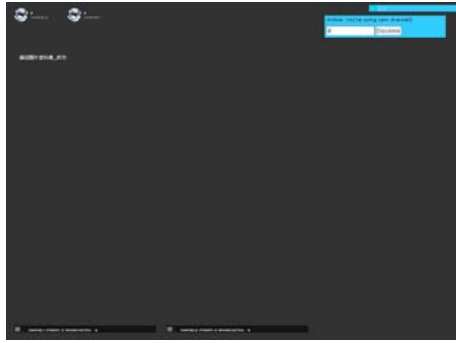
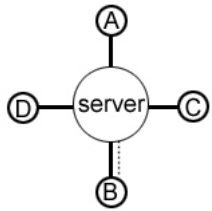
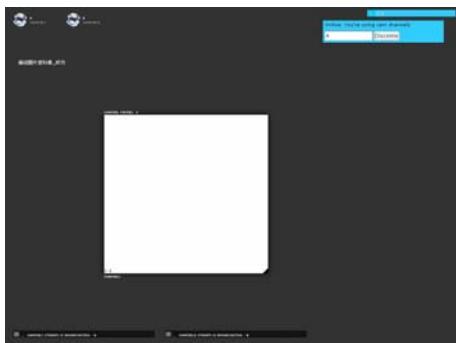
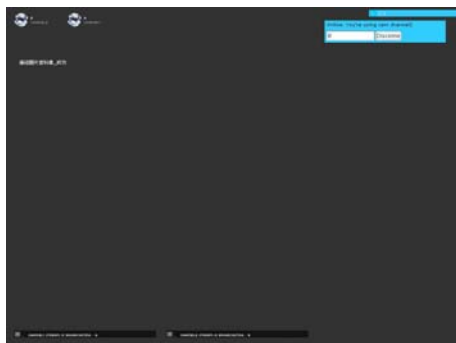
4.4 系統演示劇本

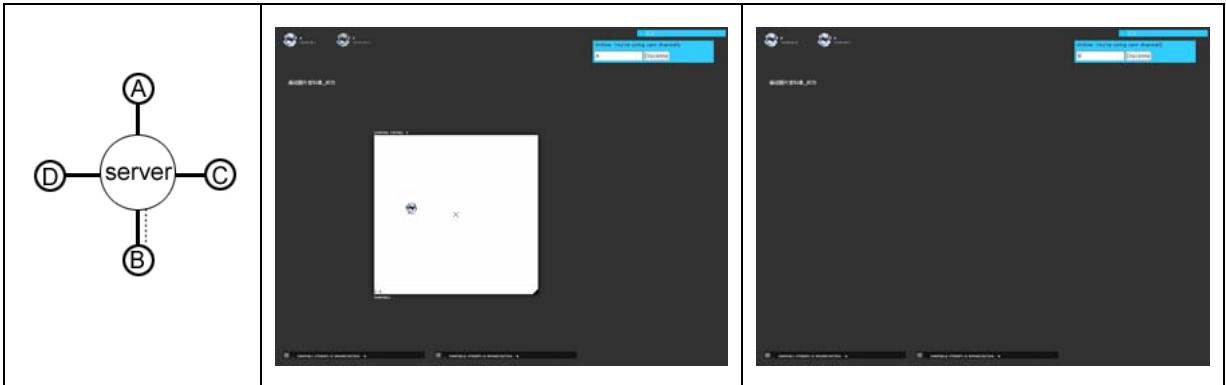
系統演示劇本用在系統實作之後的一個操作過程解說，可以快速透過同樣是劇本(Scenario)的描述方式，以實際的操作過程紀錄按部就班的一一對每個動作做可能的解說。除了可以讓使用者進行實際上機測試之前的手冊，也可以讓整個系統運作被後可以有比較完善的紀錄。接著，我們透過實機拍攝的畫面，將對系統有關如何在遠端合作環境中去支援小組討論的行為做詳細的描述。首先，系統必須讓使用者進行登入的動作。當所有終端使用者完成登入之後，畫面會出現一個代表主頻道討論空間的紙板，並且在左上角顯示目前在系統中的所有使用者清單，主頻道紙板的左下方也會出現該頻道已被授權編輯與進行討論的使用者名單。每個終端只有被授權使用的紙板頻道才會出現在終端的畫面上，而這個一開始系統主動註冊用來代表主頻道的紙板是共同編輯的頻道，也是遠端環境中用來隱喻電子白板共有的角色。所以首次登入後，這個頻道所代表的紙板會自動出現在所有終端的畫面上(圖 65)。



(圖 65) 系統首次登入時的畫面，系統主動配給每個終端一個共有的主頻道空間

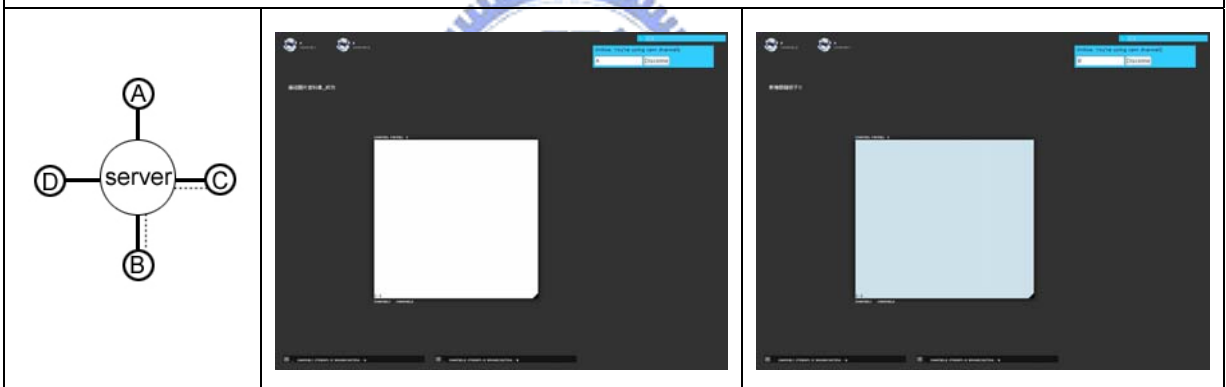
此時，一個基本的遠端合作筆式環境的基本操作就和一般的筆式環境一樣，使用者可以透過筆式硬體或是繪圖板在任何已打開的紙板上進行編輯，同時透過視訊跟聲音對談來與遠端進行討論。特色在於，這個代表頻道的紙板是具有本研究期間強調的連結性，任何一端的終端在紙板上作了什麼更動，其他的終端也會跟著同步更新。也就是終端的兩邊的感受其實是在同一張紙上進行編輯。而設計者除了這個唯一的共有的主頻道紙張之外，可以自行透過代表描圖紙筒的[::]符號拖拉出任何空白的紙板開始進行想法的發想、草寫紀錄等。這些都是現有的遠端合作式系統透過不同的技術可以達成的，操過過程大同小異。最大的不同在於紙板的頻道劃分，因此接著這裡將直接進入這個系統實作的最大特色，頻道之間的劃分是如何操作的，以及他的背後邏輯運作過程的相對應流程：(以其中兩組終端畫面為例)

<p>所有終端第一次登入系統之後，會顯示由系統主動註冊代表主頻道共同編輯的紙板，這一塊紙板是唯一的，代表傳統設計環境中的電子白板或是大面積紙張的隱喻，所有使用者都可以在上面進行編輯與討論跟註解。(這裡為了方便解釋使用者如何在之後新增屬於自己的私人頻道紙板，特將唯一的主頻道紙板畫面去除，僅顯示使用者如何使用 BackChannel 空間的互動過程。)</p>		
<p>Server - Client</p> 	<p>Client B</p> 	<p>Client C</p> 
<p>(上圖) 一開始的工作平台(Platform)，使用者可以在平台上任意地方新增頻道，頻道透過紙板來表現。(左圖以粗線來代表 終端與伺服器之間永遠有一條唯一的主頻道空間。)</p>		
		
<p>(上圖) B 使用者在自己的工作檯面上新開一個紙板，這個紙板並主動向伺服器註冊一組頻道使用，並將該板的屬性回報給伺服器。(左圖虛線代表新建立的副頻道 BackChannel)</p>		



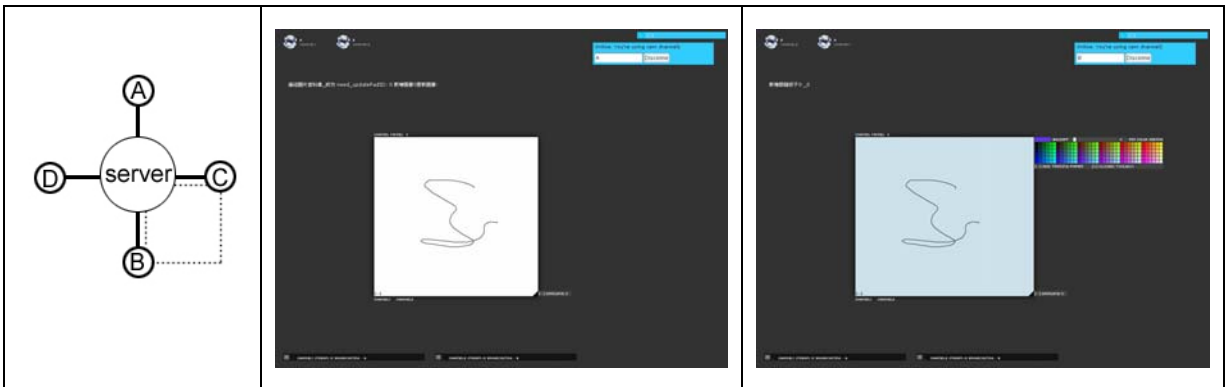
(上圖) 目前由 B 所註冊的頻道只有 B 可以進行編輯與使用，終端處理 B 端的相關資訊但不加以廣播。而 B 持續的在自己的私人頻道上進行發想，並透過草圖紀錄的一些想法。此時 B 使用者突然想到一個不錯的想法，但想跟比較熟悉的使用者 C 先行詢問意見並將想法精煉，也就是類似小組討論的需求。所以 B 透過左上角的使用者清單將 C 的代表符號，拖曳到自己所開啓的紙板上頭，將該板的頻道使用權開放給使用者 C。

(本研究實作的環境限制，因此讓伺服器端還是廣播，由 C 終端的屬性引擎代替伺服器端決定要不要顯示來因應。)



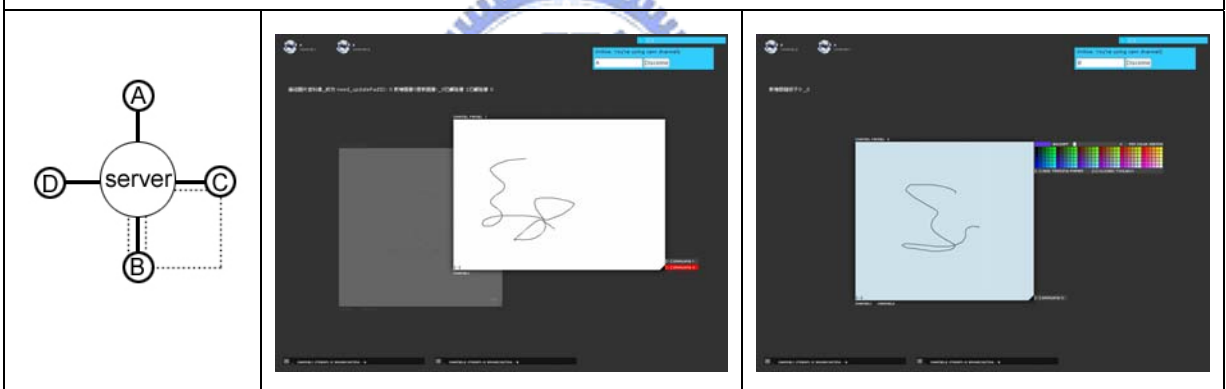
(上圖) 此時的 SERVER 收到終端 B 的授權要求，將之前僅有 B 可以使用的頻道授權給予終端 C。此時的 C 終端的畫面便會將該板的狀態顯現出來，並將之前 B 在該頻道上所做的動作，完全呈現。

(本研究實作的環境限制，因此寫在伺服器端的屬性引擎這時候收到通知，將之前隱藏的頻道顯示，並開放編輯)



(上圖) 此時由 B 註冊並開放給 C 同時編輯的紙板所代表的副頻道空間對於其他使用者 A D 來說就是一個看不見的頻道空間(註 1)，但對於 B C 來講就是一個背景頻道空間(Back Channel)。而對於 B 或 C 來說，原本代表主頻道的紙板，在這一短暫的時候對她們來說，就是不用刻意專注但仍然可以察覺的紙板空間或說氛圍頻道(Ambient Channel)。

(註 1) 原因是 SERVER 並沒有收到註冊者 B 的授權，給予 A D 權限。然後這是本研究的設計，如果有其他議題，可以在介面上顯示這條連結線給討論會議主持者知道，但不一定要顯示溝通內容。



(上圖) 系統允許每一位終端使用者任意新增紙板並向 SERVER 註冊頻道來使用，並且隨時可以決定其它使用者對於某些紙板的共同編輯權限，只要透過簡單的拖拉代表性使用者的符號就可以輕易授權與關閉授權。以終端 B 來說，他在與 C 討論的同時，臨時又有一個概略的想法，但可能還無法解釋給 C 了解，因此又在平台上新增了紙板，紙板並主動向 SERVER 端註冊一組新頻道使用，而這樣的往覆的過程在一個正常的設計討論環境是不斷發生跟終止的，因此這樣的遠端設計環境才能有效的支援本研究所謂的小組討論。