

國立臺灣大學工學院機械工程學系

博士論文

Department of Mechanical Engineering

College of Engineering

National Taiwan University

Doctoral Dissertation

協同環境下CAD模型接力設計之設變傳遞

Design Change Propagation of CAD Sequential Design in
Collaboration



趙甦農

Chao, Su-Nung

指導教授：尤春風 博士

Advisor: You, Chun-Fong, Ph.D.

中華民國 98 年 7 月

July, 2009

協同環境下CAD模型接力設計之設變傳遞

摘要

近年來，CAD產業日益成熟，而協同設計的觀念更被廣泛地應用在許多CAD系統上，以上游工業設計與下游機構設計之間的接力設計而言，可以提供兩端遠距離的共同設計與討論。接力設計通常使用中介格式作檔案交換，當過程中產生設計變更時，便會遭遇困難，工業設計端的設計變更，易導致機構設計的努力付諸東流。針對此設變傳遞之問題，本研究以三個步驟來進行。

首先，本研究提出一種彈性協同架構，將CAD的結構做分層，使得CAD系統可以輕易地從單機操作轉換成各種協同設計的版本。其次，應用彈性架構，整合CAD與PDM系統成為一個網站平台，使用者可省去安裝步驟，直覺操作，並且解決不同單位跨越設計平台的問題。最後，將彈性架構中的傳遞資料紀錄為設計步驟，藉由保存傳遞資料，達到同步或非同步協同的功能。利用「模型相似性比對」技術，讓協同兩端比較設計變更前後的檔案版本差異，並修改設計步驟以傳遞設計變更，使得機構設計在工業設計設變後得以延續。

關鍵字： CAD、接力設計、設變傳遞、彈性協同架構、模型相似性比對

Design Change Propagation of CAD Sequential Design in Collaboration

Chao, Su-Nung

Abstract

Recently, some products have been concentrated on developing distributed CAD systems with collaborative methodologies. Collaboration could provide discussion and co-design between the two upstream industrial design and downstream mechanical design teams. The sequential design starts from industrial design using neutral file as exchange format. When the design change occurs during the process, the following mechanical design can't access the design contents. According to the problem about design propagation, three steps are proposed in this research.

First of all, this research proposes flexible collaboration architecture, separating the CAD structures into some layers, and make CAD system easier to change its working style from standalone to any other collaborative edition. Second, CAD and PDM system can integrate to from a web platform using the flexible architecture. It is unnecessary to install the CAD system, and users can manipulate models directly, which can solve the problem when design stage is across two different sites. Final, take the recording data passed in the flexible architecture as design steps, and by reserving these data, the design procedures can be recorded and replayed in real-time or afterward. As a result, the synchronous collaboration and asynchronous collaboration can be achieved. System will compare the later input neutral file with the former one by using "similarity comparison" of 3D model, and modify the design stages on former neutral file and apply them into the later neutral file. Thus, the design change can propagate in the mechanical design if it happens in the industrial design.

Keywords: CAD, sequential design, design change, flexible collaboration architecture, similarity comparison

目錄

摘要	i
Abstract.....	ii
目錄	iii
圖目錄	vi
表目錄	ix
第一章 導論	1
1.1 研究背景	1
1.2 研究目的	2
1.3 研究範圍	3
1.4 研究方法與步驟	4
1.4.1 多層次彈性架構與CAD實作	4
1.4.2 協同平台與動態協同紀錄	5
1.4.3 動態協同紀錄於設計變更傳遞之運用	5
1.5 論文架構	6
第二章 文獻回顧	7
2.1 CAD建模技術、協同設計及傳遞資料格式	7
2.1.1 實體模型建模技術	7
2.1.2 CAD協同設計	9
2.1.3 巨集指令	10
2.2 協同平台與動態協同紀錄	12
2.2.1 協同平台	12
2.2.2 協同紀錄	13
2.3 中介格式為媒介之接力設計與設變傳遞	13
2.3.1 中介格式	14
2.3.2 設計分析	16
2.3.3 中介格式模型間的相異比較	17
2.3.4 設計步驟複製	18

2.4 參考文獻之整理比較	19
第三章 彈性協同架構與CAD系統實作	20
3.1 階層間傳遞的資料格式	22
3.1.1 巨集型態傳輸資料 (MTPD)	23
3.1.2 巢狀樹資料 (NTD)	24
3.1.3 巢狀圖形資料 (NGD)	25
3.2 可調整的操作型態	25
3.2.1 單機版	26
3.2.2 對等式協同	28
3.2.3 主從式協同—輕客戶端與重伺服器端	29
3.3 實作CAD系統	31
3.3.1 實作單機版	32
3.3.2 實作對等式協同	33
3.3.3 實作主從式協同—輕客戶端與重伺服器端	35
3.3.4 實際案例執行時間比較	35
3.4 市面上現行CAD系統之比較	45
3.4.1 對等式協同	45
3.4.2 主從式協同	47
3.4.3 協同比較	50
第四章 協同平台與動態協同紀錄	52
4.1 網頁為基的架構	52
4.2 作傳送用途的資料格式	53
4.3 非同步專案設計及動態回顧	54
4.3.1 代理伺服器	54
4.3.2 動態回顧	55
4.3.3 非同步設計	56
4.4 實作協同平台與動態協同紀錄	57
4.4.1 系統架構	57
4.4.2 實作協同平台	57

4.4.3 動態協同紀錄	63
4.4.4 重現協同	65
第五章 動態協同紀錄於設計變更傳遞之運用	68
5.1 設計分析	69
5.2 模型相異性比較	69
5.2.1 中介格式檔案的面比較	71
5.2.2 邊緣與端點的比較	79
5.2.3 2D圖形的比較	83
5.3 設計步驟複製	84
5.4 協同討論	87
5.5 實作設計變更傳遞	87
5.5.1 系統架構	87
5.5.2 範例研究1: 手機機構設計	88
5.5.3 範例研究2: 滑鼠機構設計	92
5.5.4 成果討論	96
第六章 結論與未來展望	97
6.1 論文成果與結論	97
6.1.1 論文成果	97
6.1.2 論文結論	99
6.2 未來展望	99
參考文獻	101
作者簡歷	107
歷年發表文獻	108

圖目錄

圖2.1 邊界表示法之資料結構	8
圖2.2 布林運算示意圖	9
圖2.3 巨集指令於協同系統的應用[27]	12
圖2.4 不同CAD系統的檔案傳遞方式	15
圖2.5 Part42拓撲與幾何的資料結構	16
圖3.1 在CAD系統中七個層級之互動情況	21
圖3.2 CAD客戶端到伺服器端之間資料流	23
圖3.3 MTPD的資料格式	24
圖3.4 NTD的資料格式	24
圖3.5 修改過之OBJ 格式	26
圖3.6 單機版本的資料流動示意圖	27
圖3.7 單機版的架構	27
圖3.8 對等式協同的資料流動情況	28
圖3.9 對等式協同的資料流動示意圖	29
圖3.10 精簡主從式架構的資料流動順序示意圖	30
圖3.11 模擬在精簡主從式架構中伺服器與多客戶端的互動情況	31
圖3.12 SpringSolid CAD單機版本	32
圖3.13 對等式協同中，使用者開啟伺服器並建立連結	33
圖3.14 對等式協同之協同過程	34
圖3.15 簡單的伺服器端使用者介面	35
圖3.16 客戶端的網頁畫面	36
圖3.17 不同協同狀態下，使用者操作之流程	37
圖3.18 系統開啟檔案流程圖	38
圖3.19 以對等式協同開啟測試檔案	39
圖3.20 主從式協同一重客戶端與輕伺服器端 開啟受測檔案	43
圖3.21 CATIA中開啟對等式協同	46
圖3.22 使用Net Meeting來進行對等式協同	46

圖3.23 CATIA協同會議控制視窗	47
圖3.24 CATIA兩邊可以同時開啟同一檔案，進行協同討論	47
圖3.25 AutoVue伺服器端程式	48
圖3.26 直接在使用者端開啟網頁讀取CAD模型	48
圖3.27 AutoVue進行協同前，必須先設定會議主題並開啟會議	49
圖3.28 兩台電腦的使用者加入同一個會議，進行協同討論	49
圖3.29 提供3D標註方便使用者進行協同討論	50
圖4.1 三層式架構一般的元件佈局	53
圖4.2 多層次彈性架構與三層式架構之組合架構圖	54
圖4.3 PDM伺服器扮演CAD代理伺服器的角色	55
圖4.4 使用者透過PDM客戶端瀏覽專案協同過程	56
圖4.5 從PDM伺服器要求協同紀錄的MTPD便可進行非同步設計	56
圖4.6 協同設計平台的系統架構	58
圖4.7 架設協同平台之伺服器端	59
圖4.8 使用者透過網頁瀏覽器登入網站	59
圖4.9 登入網站後的三種選項	60
圖4.10 專案討論區	61
圖4.11 選擇專案成員的畫面	61
圖4.12 協同開始，專案中的每個使用者都會進入此畫面	63
圖4.13 PDM系統之檔案操作歷史頁面	64
圖4.14 兩個使用者協同設計的流程圖	65
圖4.15 使用者非同步檢視協同紀錄之流程	67
圖5.1 簡單流程圖說明設計變更之法則	68
圖5.2 產生ID2M1之詳細流程圖	70
圖5.3 參考面分類之流程圖	72
圖5.4 幾何比較之虛擬程式碼	73
圖5.5 決定相同面的流程	74
圖5.6 決定相嵌面的流程	75
圖5.7 從實體模型之結構中取得面邊關係	76

圖 5.8 模型的 3D 圖形及其面邊關係矩陣	77
圖 5.9 利用面邊關係及已知面來尋找未知對應面	78
圖 5.11 邊的比較方式	80
圖 5.13 端點的比較方式	82
圖 5.15 處理某些步驟時須參考更新之 ID1	85
圖 5.16 後續步驟複製必須參考更新後的 ID1 與 ID2	85
圖 5.17 MTPD 與 MTPD' 之間的關係	86
圖 5.18 ID1、ID2 與 ID1M1 之間的關係	88
圖 5.19 利用 MTPD Viewer 檢視與分析	89
圖 5.20 手動選取 MTPD 項目並進行修改	90
圖 5.21 系統建議 ID1、ID2 更新後再執行比較	90
圖 5.22 更新的 ID1 與 ID2 比較	91
圖 5.23 應用 MTPD'S 於 ID2 所產生之結果	92
圖 5.24 ID1、ID2 與 ID1M1 之間的關係	93
圖 5.25 MTPD viewer 顯示 MTPDS 中的每個 MTPD	94
圖 5.26 ID1 與 ID2 之比較	94
圖 5.27 修改 MTPD 項目時，系統提供對應面選擇	95
圖 5.28 為找到參考物件及其對應物件，系統建議執行再比較	95
圖 5.29 比較更新過後的 ID1 與 ID2	95
圖 5.30 將 MTPD'S 應用於 ID2 而得到之結果	96

表目錄

表2.1 分散建模機制的分類表	11
表2.2 參考文獻之整理比較.....	19
表3.1 網路速度1024KB/s之對等式協同開啟比較時間表	40
表3.2 網路速度256KB/s之對等式協同開啟比較時間表	41
表3.3 網路速度48KB/s之對等式協同開啟比較時間表	41
表3.4 網路速度1024KB/s之主從式協同開啟比較時間表	44
表3.5 網路速度256KB/s之主從式協同開啟比較時間表	44
表3.6 網路速度48KB/s之主從式協同開啟比較時間表	45
表3.7 CAD比較表	51



第一章 導論

1.1 研究背景

電腦輔助設計(Computer Aided Design , CAD)能透過電腦及其圖形設備幫助設計人員進行設計工作。在編輯方面，平移、旋轉、縮放等工作皆由電腦計算處理。系統亦可以在設計時幫助設計人員進行繪圖判斷，不但能減輕設計人員的負擔，對於提高設計品質與縮短設計周期也有一定幫助。

CAD的發展歷史可以回溯到20世紀中，首先推出CAD系統為2D幾何建模的概念，接著出現3D曲面造型的功能。80年代出現實體模型的概念，模型必須是封閉的實體。接下來出現特徵為基底的建模方式、參數化的建模歷程紀錄，慢慢發展成現在的CAD系統。

近年來，協同設計在CAD的發展史中成為重要的里程碑，主要是透過網路建立一個虛擬的設計團隊，並整合分散各地的系統。有些工作可以透過企業內部網路與其他部門的人員作協同設計，而有一些協同系統更可以讓設計者透過網際網路與其他企業的員工做溝通。

產品資料管理(Product Data Management, PDM)在協同設計與CAD之間也扮演重要角色。從產品的概念設計到製造、生產、改版一連串過程中，PDM系統收集產品的相關資訊，其中包含陸續增加的CAD設計資料。設計資料依檔案相關性加以整理後，提供資料保存位置並與其屬性作連結，並紀錄改版與維持資料單一性，在協同

設計中提供檔案交換的功能。

協同工程按照產業結構來分，可以分成水平協同與垂直協同。水平協同可以提升企業內部的設計效率，而垂直協同則能讓中心與協力廠商之間的分工合作更密切。水平協同與垂直協同在功能上是互補，利用垂直協同可無縫隙地整合上游設計與下游製造廠商，水平協同則可協助完成設計工作。若系統能整合此兩種協同模式，則此整合的系統可以提升效率與分享相關知識，進而提高設計品質。

若按照協同模式來分，則可以分為主從式協同(Client-Server)以及對等式協同(Peer-to-Peer)，其中擁有客戶端與伺服器端的主從式協同又可按照客戶端是否擁有核心計算能力而分成主從式協同一輕客戶端與重伺服器端 (Thin Client-Thick Server)與主從式協同一重客戶端與輕伺服器(Thick Client-Thin Server)。如果CAD系統可以在單機版、主從式協同與對等式協同之間做狀態的更換，則將更有利於水平協同與垂直協同之間的整合。設計人員的複雜繪圖工作使用單機版(Stand Alone)，而在企業內不同部門的協同討論則使用主從式協同，與企業外部的協力廠商則可以使用對等式協同。

1.2 研究目的

協同的應用廣泛，以工業設計與機構設計之間的接力設計來說，其可以提供兩端遠距離的共同設計與討論。工業設計端設計完工業設計模型，機構設計端接著設計內部機構，當兩者設計端使用不同的CAD系統，中介格式的檔案，如STEP(Standard for Exchange of Product Model Data)、IGES(Initial Graphics Exchange Specification)

檔案，常被使用來傳遞設計結果。由於中介格式檔並無法攜帶CAD系統內的原始特徵資料，只記錄幾何及拓撲資料，接續設計的機構設計端也就只能以幾何與拓撲資料開始做機構設計。當工業設計端產生設計變更時，會產生新的中介格式檔案，以將新的外型傳遞至機構設計端。雖然工業設計端也許只更改某些特徵參數，但卻因為中介格式檔無法紀錄特徵參數，使得新的中介格式檔與原本的中介格式檔沒有關聯，機構設計端拿到新的中介格式檔，也只能靠使用者肉眼辨識先後檔案的差異，而機構設計的特徵可能無法套用在新的檔案上。由於設計變更的傳遞無法跨越中介檔案，往往使得接力設計中延續設計的一方，因為設計變更發生就前功盡棄，設計必須從新的中介格式檔案重新開始。

因此，如果能有一個CAD系統，可在協同環境下以傳遞中介格式的方式延續跨CAD設計，並且當先前CAD系統所設計的模型產生設計變更時，可以將此CAD設計的紀錄先提出，然後套用在新的中介格式檔上，便可快速地將新的變更傳遞至現行的設計上，進而解決目前存在的問題。

1.3 研究範圍

論文中將提出一個多層次彈性架構，實作CAD系統，並成功執行各種操作模式：包含單機版、主從式協同一輕客戶端與重伺服器端、主從式協同一重客戶端與輕伺服器以及對等式協同。提出的架構，應用在PDM系統上，進而將CAD與PDM整合成網頁的協同平台，並且透過本架構之傳輸機制，將CAD的操作流程條理分明的被記錄，利用特定的機制重現設計流程，達到動態的協同紀錄，最後再藉此

動態協同紀錄搭配模型相似性比對法則，進行接力設計的變更傳遞。

論文的主要目標包括：

- 1、多層次彈性架構與CAD實作
- 2、協同平台與動態協同紀錄
- 3、動態協同紀錄運用於中介格式為媒介之設計變更傳遞

1.4 研究方法與步驟

基於研究的三個主要目標，分別分章節簡要說明論文所提之研究方法與步驟，詳細地分別敘述於第三、四、五章。

1.4.1 多層次彈性架構與CAD實作

本論文提到多層次彈性架構的建置方式，將CAD以主要的七個層級來劃分，分別是控制層、使用者介面層、資料樹層、圖形層、通訊層、橋接層與核心層，各層分別執行CAD不同的部份。此彈性架構可以藉由資料傳遞的方式改變而產生不同的運作模式，詳細內容，將在第三章詳述。

介於架構中的資料傳輸形態也扮演相當重要的角色，傳輸資料型態參考巨集指令(Marco)，並根據不同CAD模組所需要的功能逐一做指令的定義。針對CAD各種不同的運作型態，做資料流的規劃，並且實際以JAVA、JAVA Script、JSP等語言，搭配尤春風博士所發展之SpringSolid CAD核心，實作單機版、精簡客戶端主從式協同以及對等式協同CAD系統，以驗證此多層次彈性架構可以提高CAD系

統調整的彈性。

1.4.2 協同平台與動態協同紀錄

CAD系統搭配先前所提的彈性架構，可以將客戶端輕量化成網頁元件。PDM系統利用同樣的架構，可成功轉換為輕量化的網頁使用者介面，在網頁端PDM使用者介面可以與CAD使用者介面整合成入口網站，而後端的伺服器也可以完成整合。兩者搭配成為一個協同平台，可線上設計，還可以透過網頁介面來管理產品相關設計資料。

透過網頁化的機制，CAD的操作流程可以被歸類為產品設計的相關資料，被記錄存放於PDM系統內。紀錄存放的資料可分為兩種，一種是指令與輸入資料，有利網路傳輸，卻需要核心支援大量計算；另外一種則是計算結果回傳，無須再多做計算，但也因資料量較大而易被網路的傳輸速度影響。不過上述兩種記錄資料都可透過對應的處理方式而動態重現協同的過程。

1.4.3 動態協同紀錄於設計變更傳遞之運用

本論文提出一個在接力設計時設計變更傳遞的機制，針對先後輸入不同的中介格式檔案進行差異比較，並在本論文提出的彈性架構所建置的CAD中進行第一次機構設計，然後分析設計內容，再將彈性架構中的傳輸資料進行複製，並以先前輸入檔案的差異比較為根據來調整設計步驟，將第一次機構設計中的大部分設計內容延伸到第二次輸入的中介格式檔案上，以達到設計變更傳遞的目的。

1.5 論文架構

本文共分六章：在本章說明論文中的研究背景、目標以及系統架構；第二章將對相關的論文作一詳實的回顧與討論；第三章則介紹論文所提出的彈性協同架構，並且利用此一架構直接實作可單機、可網頁操控、可協同又能動態改變的CAD系統；第四章提到實作網頁上的協同平台，並利用紀錄讓協同過程重現；第五章則衍生先前所提到之彈性協同架構，實作接力設計之設變傳遞；最後於第六章提出結論與未來展望。



第二章 文獻回顧

在協同工程中，水平協同與垂直協同在功能上互補。為了能使CAD同時進行此兩種協同並整合其優點，參考諸多協同模式與實作方式後，論文中提出一個可以彈性變化的架構，並藉此實作CAD系統與網路協同平台，再利用此平台為基礎，探討在不同CAD間以中介格式檔案為傳遞媒介進行設變傳遞的情況。

本章將針對論文相關文獻做回顧，其中主要分三個部份：2.1節回顧有關CAD建模技術及協同設計之相關文獻。2.2節則整理CAD協同平台實作、協同紀錄等相關文獻。2.3節則是條列設變傳遞相關的研究。

2.1 CAD建模技術、協同設計及傳遞資料格式

要針對CAD系統提出協同架構的改良並加以實作，就須先研究CAD系統本身。了解實體模型建模技術以及協同模式之後，參考傳遞巨集指令的方式，便可以定義多層次架構及其中傳遞的資料型態。本節將介紹實體模型建模技術、CAD協同模式及傳遞巨集指令（Marco）的參考文獻。

2.1.1 實體模型建模技術

現今一般CAD系統核心通常使用實體模型[1]來定義與展現物體，包含幾何與拓樸架構。實體模型有二種表示法：（1）邊界表示法（Boundary Representation, B-Reps），（2）構成實體幾何法

(Constructive Solid Geometry , CSG) 。

邊界表示法所建構的實體模型是透過點 (point)、曲線 (curve)、曲面 (surface) 等幾何資料，及頂點 (vertex)、稜邊 (edge)、迴路 (loop)、面 (face) 等拓樸資料表現物體的形狀與位置，資料結構如圖 2.1 所示。

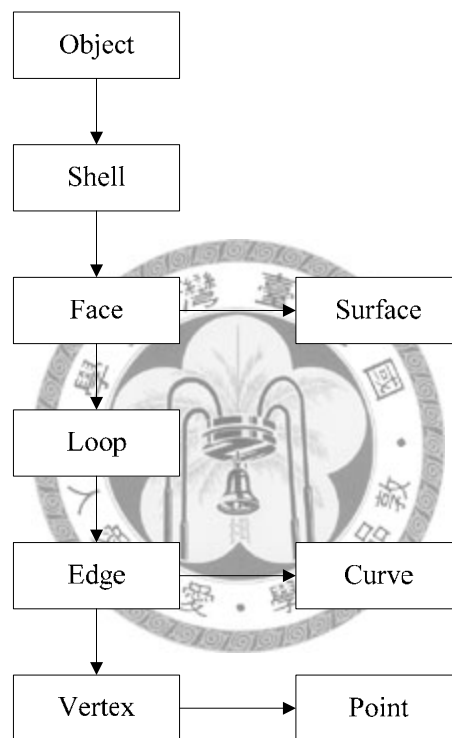


圖 2.1 邊界表示法之資料結構

構成實體幾何就是將運算元根據布林運算組合在一起，這些運算包括：聯集、交集以及差集，示意圖如圖 2.2。

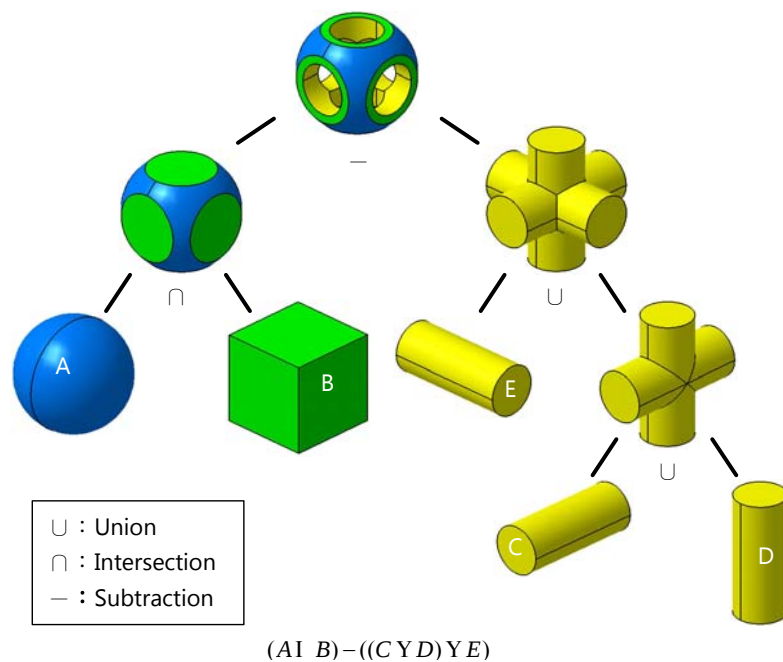


圖 2.2 布林運算示意圖

邊界表示法的形狀資料以數值提供，除顯示容易以外，還方便將資料加以應用。構成實體幾何法具有定義形狀時較不易出錯及資料簡潔的優點。雖然邊界資料的輸入方式較為複雜，但卻應用方便，而構成實體幾何法的資料需要布林運算才能獲得實際的幾何形狀，故應用上較麻煩，一般實體模型技術都將此兩種法則綜合運用，截長補短。

2.1.2 CAD協同設計

從協同機制與系統架構的觀點來看，相關的研究與實作可以簡單地分成兩種類別：

- (1) 協同檢視、標註、檢測3D CAD 模型
- (2) 能實作協同設計的共同建構3D CAD模型

第一個研究方向主要提供視覺化、註記以及檢測功能，並沒有修改與共同創造等功能。CyberView[2]使用 VRML 的格式來檢視 STEP 資料，而 CollabView[3]則提供一個視覺的呈現並且在網路上與別人分享 CAD 模型，在網路的環境上透過一些組合的三角形網格顯示模型，並且提供旋轉、平移、縮放等操作功能。有許多新技術的提出讓資料更有效率地在網際網路之間傳輸。以三角形網格簡化為例，Hoppe[4]根據邊折疊法則（edge collapse method）發展一個多精細度模型法（level-of-detail, LOD），Taubin 等人[5]則提出一套漸進森林分裂（Progressive Forest Split, PFS）法則，可以將前一個步驟簡化成低細節程度（low-detail）的模型，再轉換成高細節解析度。

上述系統可以支援一群人針對某特定的產品一起工作，但是簡化過的三角形網格模型與原本的模型資料差異相當大，加上不能支援多使用者同時一起創造或修改細部設計，因此只能當作是支援的工具。

第二種研究方向提供真正共同建構的功能，而其中有三種常見的分散建模機制，在表 2.1 中分類列舉。雖然許多軟體都可以協同，但是因為商業的限制，很難取得這些系統的詳細技術及內部機制。

2.1.3 巨集指令

在協同環境當中，資料的傳遞扮演重要的角色。Qiang 等人[8]使用巨集指令作為傳送協同資料，藉由此方法來減低傳送時間，並

表 2.1 分散建模機制的分類表

架構	描述與特性	研究結果
主從式協同— 重客戶端與輕 伺服器端	◆ 客戶端有整個CAD系統以及額外的通訊模組。 ◆ 伺服器端有通訊模組提供客戶與客戶之間的資訊交換。 ◆ 單獨的CAD系統可以藉此機制變成重客戶端，但很難改變成網頁應用程式。	CollabCAD[6], Nam and Wright[7] , Qiang et al.[8], Pahng et al.[9] , Chang HC et al.[10], N.Shyamsundar et al. [11], IX Design[12], Qiuli Sun et al.[13]
主從式協同— 輕客戶端與重 伺服器端	◆ 客戶端的結構是輕量化，主要支援視覺化，而主要的實體建模則是在伺服器中執行。 ◆ 以應用服務的方式，模組可分別提供使用者。 ◆ 因為主要的模型及操作功能都保留在伺服器端，資料的一致性比較好維持。	AlibreDesign[14], GS-Design[15], Li et al.[16], OneSpace[17], Li Chen et al.[18], Karthik Ramani et al.[19] Chan et al. [20], Bidarra et al.[21], Foster[22]
對等式協同	◆ 與重客戶端相似，但是伺服器已經嵌在同一個應用程式裡頭 ◆ 每個元件都相同，每個使用者都可以開啟伺服器功能與他人做整合。 ◆ 不需要額外的伺服器。	CATIA[23], Pro/E[24], Stephen C. F. Chan et al.[25]

建立一個群組同步操作的方法。Choi 等人[26] 也提出用巨集指令參數交換CAD模型。Li等人[27]則將巨集指令定義為系統模型操作

(system modeling operations, SMOs)，並提出中性的模型指令 (neutral modeling commands, NMCs)，使得所有的CAD系統可以藉由SMOs與NMCs的交換而得到及時的同步設計，如圖2.3所示。

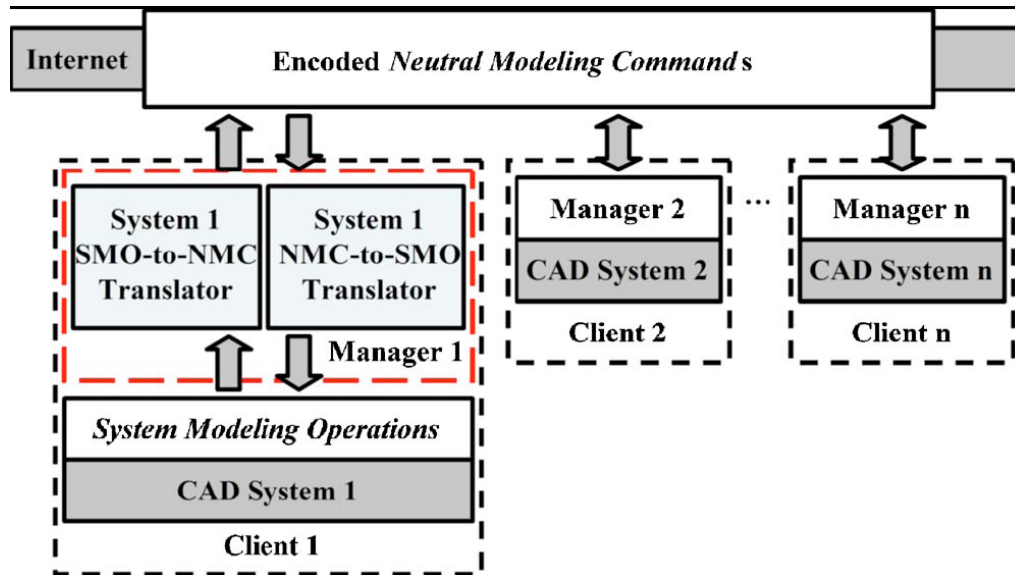


圖 2.3 巨集指令於協同系統的應用[27]

2.2 協同平台與動態協同紀錄

以多層次架構實作的CAD系統可以任意變換協同的型態，然而要建立協同平台，仍需要決定較適合的協同型態。而在多層次架構下所使用的傳輸資料乃參考巨集資料的型態，要紀錄協同的每個動作，也需要將巨集資料紀錄整理。以下針對協同平台以及協同記錄做文獻介紹。

2.2.1 協同平台

Qiang等人[8]利用巨集指令傳送協同資料，使用主從式協同一重客戶端與輕伺服器的架構。Shyamsundar 等人[11]提出智慧伺服器

(Intelligent Server, IS) 的概念來提供協同平台的控制，實作主從式協同一輕客戶端與重伺服器之模式且僅提供介面，而讓後端伺服器作CAD核心運算。 Chan等人[25]認為使用主從式協同一輕客戶端與重伺服器之協同模式，可以降低CAD資料的安全性問題，而在伺服器上控管也可以使客戶端不需要知道其他客戶端的位置。 Li等人[28]同樣提出一套架構，採取主從式協同一輕客戶端與重伺服器模式，除可以共同設計外，還可以提供網頁的視覺化模型瀏覽、電腦輔助製程規劃。參考諸多文獻，得知若要實作網路協同平台，並將CAD系統朝網頁化發展，使用主從式協同一輕客戶端與重伺服器模式較為適合。

2.2.2 協同紀錄

Li[27]曾提及在不同CAD系統之間傳遞使用者指定的動作(SMO)，並將SMO轉譯為中性建模命令(NMC)，再傳給其他CAD系統，而所有傳遞的中性建模命令將按照時間分類儲存在伺服器中。

2.3 中介格式為媒介之接力設計與設變傳遞

設計變更可能發生在初始設計、製造或生產，甚至產品已經上市之後。在初始階段的工業設計與機構設計可能發生在不同的公司與部門，如果工業設計與機構設計是利用不同的CAD軟體來設計，當設計變更產生時，就會發生設計變更傳遞的問題[29]。

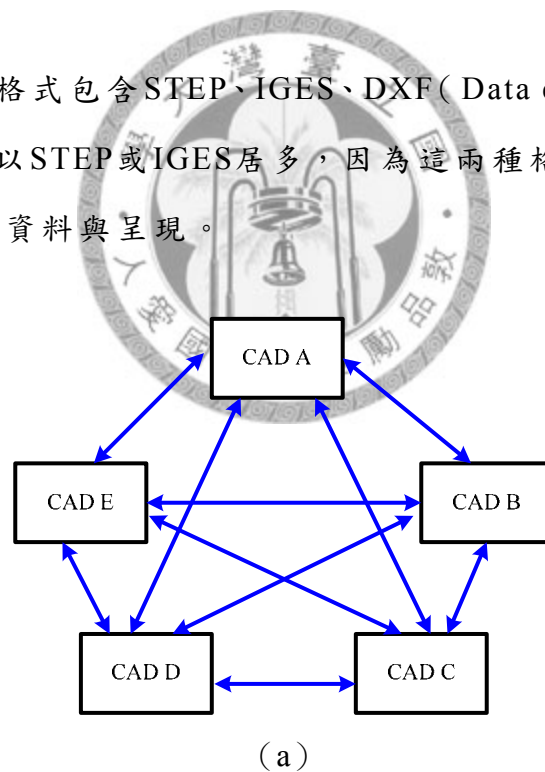
當工業設計修改過後，機構設計一般來說需要重新建置。本研究第五章可針對舊設計進行分析、比較不同的輸入中介格式檔案、

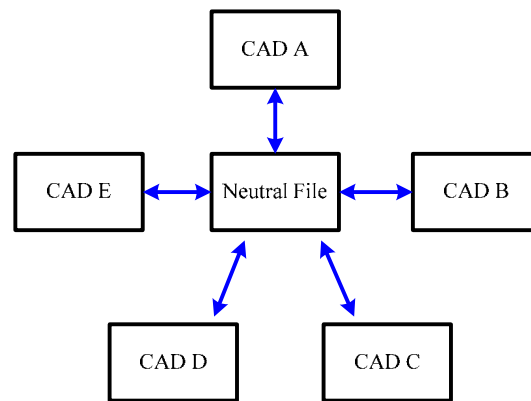
複製設計步驟到新的中介格式檔上。如果問題發生在設計步驟複製時，或者改變範圍太廣，則可能機構設計端必須與工業設計端進行溝通。以下則針對這幾點提出相關研究回顧。

2.3.1 中介格式

在不同CAD系統間，不同的檔案格式或資料結構導致檔案交換傳遞時的不相容，因而需要一個中繼轉換機制來協助溝通。圖2.4為不同CAD系統間檔案傳遞的兩種方式。a為檔案在不同系統中採用直接傳遞的方式，b為檔案透過中性檔案而傳遞至各個系統。

中性檔案的格式包含STEP、IGES、DXF (Data eXchange Format) 等，而一般選擇以STEP或IGES居多，因為這兩種格式可以確切的紀錄3D幾何模型的資料與呈現。





(b)

圖 2.4 不同 CAD 系統的檔案傳遞方式

(a) 不同系統中資訊直接傳遞

(b) 不同系統中透過中性檔傳遞資訊

國際標準組織（ISO）於西元1994年發佈ISO-10303，即STEP標準，提供以產品為導向的產品資訊標準化表示法，確保產品生命週期中資料表示與交換得以一致。STEP標準藉由指定產品生命週期中，所產生各種產品資料的表示與交換形式，進行資料交換，因此形式與CAD系統無關，故可在各種系統上，以不同的方法執行、儲存與使用產品資料，且具有相當之精確性、一致性及時效性。STEP標準依其功能性的不同分成若干Parts，以便能完整地描述產品資料表示標準的運作。圖2.5所示，Part 42[30]採用實體模型之邊界表示法的資料結構，定義產品的幾何資料、拓樸資料及幾何模式。Part 203/214（AP203/214）[31][32]等標準採用Part 42定義幾何、拓樸、產品與產品結構之標準模型，運用於CAD系統中的應用協定。

IGES[33]是美國國家標準局（National Bureau of Standards）在西元1980年所公布的第一個CAD資料交換的規格，並在西元1981年，美國國家標準（ANSI）制訂為標準，代號ANSI Y14.26M為IGES

版本1.0。IGES一開始是被發展用來做工程圖的資料交換，舉凡2D/3D的線架構、文字、尺寸與曲面的資料等都包含，但是在初期階段資料交換的情況並不符合預期，經過不斷地發展與改善，IGES所涵蓋的內容已經不單單只是1.0版本所提到的範圍而已，取而代之是考慮CAD系統所支援的單元、語法格式與資料的一致性。目前最新的版本為5.3，此一版本已經涵蓋幾何與拓樸跟其他附加資訊。

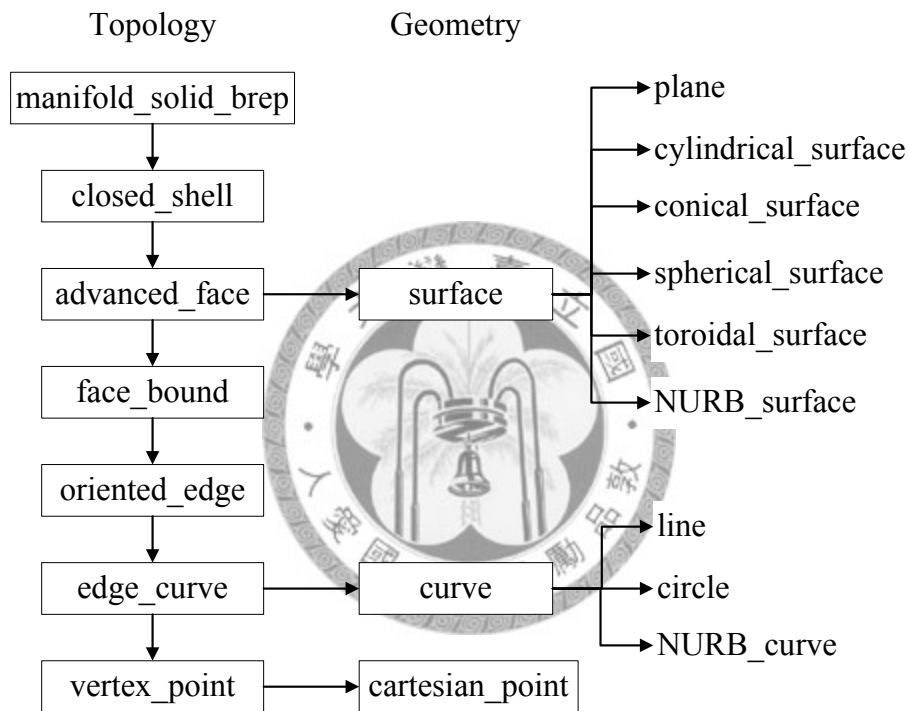


圖 2.5 Part42 拓樸與幾何的資料結構

2.3.2 設計分析

許多CAD系統採取B-reps及CSG來建模。利用布林運算以及構造實體幾何法來產生整個實體模型外型。所有使用過的特徵都被記錄。如果設計變更產生，這些特徵都可以用來再產生部分的實體模型。

通常在不同CAD系統之間要傳遞一個設計檔案必須透過中介檔案格式。以STEP為例，STEP的Part42支援邊界表示法[34]，但是卻無法紀錄參數尺寸。此外，Part42標準雖然有包含布林運算及其他一些簡單的功能，如延伸實體、旋轉實體等，但是並沒有提供其他大部分的功能，如薄殼或導角。因此，若STEP被用來做CAD模型交換的格式，則幾乎整個設計變更不能夠被傳遞[26]。

當設計在不同CAD系統間做轉換，倘若設計變更發生在先前的CAD系統（CAD1），則之後的CAD系統（CAD2）將會因為沒有紀錄特徵參數而不能透過CSG重新產生實體模型。為了能夠調整特徵參數，發展中介格式的特徵辨識[35-37]。

Choi[26]提出使用巨集取代STEP來達到設計傳遞的功能。Silva[29]發現使用參數設計可以預防在設計變更時產生無效的幾何外型。Sheyun[38]提到將參數設計與知識管理做結合來達到設計變更管理。

如果不能取得CAD輸入模型上的特徵資料，除針對中介格式輸入模型做特徵辨識以外，還可以針對正在CAD2設計的CAD模型進行分析。在列出新設計特徵與輸入中介檔案之間的所有參考關係後，新設計特徵可以被轉移到新的輸入中介檔案來達到不同CAD間的設計傳遞。下一節將介紹一些研究有關判斷中介格式中輸入模型的不同處。

2.3.3 中介格式模型間的相異比較

透過CAD模型之間的差異比較，可以分辨對應元件在不同CAD

模型內的差異。可從幾何或拓撲方面，進行比較模型。

從幾何面來看，Marcin[39]提出一個方法，計算受檢查模型的平行切片之2D距離區域。Silva [29]用幾何方式比較簡化後的模型。Zhang[40]結合以距離為基及角度為基的法則來達成相似零件的比較。Cardone[41]提倡形狀簽章的概念。

從拓撲面來看，Iyer[42]提出簡化的骨架圖形 (skeletal graph) 保存模型的幾何跟拓撲資料。Cicirello[43]根據製造特徵發展一個模型相依圖。El-Mehalawi [44-45]提出比較目標模型的拓撲結構圖。

2.3.4 設計步驟複製

複製一個3D模型可以從複製構造實體幾何的結構做起，但是如果複製設計步驟是根據不同的起始輸入模型，不同模型上的元件相異性就顯得相當重要[46]。以STEP為基礎的目標模型上之元件編號 (ID number) 必須重新建立關係，因為在不同的模型上，雖然有很多是相同的面或是稜邊，但這些元件的編號有可能是不同的。

Choi等人[26] 也提出用巨集指令來交換CAD模型，該巨集指令檔案包含使用者在建模時之指定指令。先針對數個商用CAD系統進行建模命令的分析，再依此分析建立一標準之模型命令。接著便可將模型轉存成巨集指令檔案，藉由巨集指令與事先定義好的標準模型命令之映射，達到CAD之間交換模型的目的。



2.4 參考文獻之整理比較

表 2.2 參考文獻之整理比較

項目	參考內容	說明
實體模型	[1]	採用 SpringSolid 作為實體模型運算之核心，包含邊界表示法與構成實體幾何法。
協同設計	[14-21]	先參考主從式協同一輕客戶端與重伺服器端，再提出彈性協同架構，將主從式協同、對等式協同以及單機版做整合。
巨集指令	[8][26][27]	定義 CAD 建模之命令，參考巨集指令之參數交換方式，將本 CAD 建模命令包裝成傳送資料。
協同平台	[25] [27]	將主從式協同一輕客戶端與重伺服器端轉換成網頁協同平台。
協同紀錄	[27]	儲存本研究 CAD 之建模命令。
中介格式之相異比較	[44-45]	利用兩模型的拓樸結構圖，搭配已知的對應面，得到未知的對應面。
設計步驟複製	[26]	複製巨集指令，但由於指令作用之目標物已經改變，必須搭配相異比較，修改對應面之資料。

第三章 彈性協同架構與CAD系統實作

本節提出一個多層次且有彈性的架構，用以建構CAD系統。此多層次架構中，每個層級的功能獨立，且實作此架構之CAD系統，可以在不改變使用者操作習慣的情況下，由單機版彈性地改變，進而使其具有協同的功能。CAD系統實作此架構後可以容易地在「單機版」、「對等式協同」及「主從式協同」等建模方式之間做轉換。主從式協同可根據計算核心的位置分為「重客戶端與輕伺服器」與「輕客戶端與重伺服器」，但「主從式協同—重客戶端與輕伺服器」因與「對等式協同」相似而省略。

資料在各層之間傳送時，資料型態也扮演很重要的角色。在架構中的資料傳遞採用巨集的概念，藉著此種資料型態，不但溝通的方式可以輕易的變換型態，協同的紀錄與步驟的複製亦可容易地被執行，如此水平的協同與垂直協同就可以更成功的整合。

為使CAD系統可以在單機版、對等式協同及主從式協同之間任意變換，此一架構可以分成七個層級，包含控制層(Control Layer)、介面層(UI Layer)、資料樹層(Data Tree Layer)、圖形層(Graphic Layer)、通訊層(Communication Layer)、橋接層(Bridge Layer)以及核心層(Kernel Layer)，如圖3.1所示。

使用者端可以接觸者為介面層、資料樹層與圖形層。介面層的成員像是系統框架、指令選單、功能表、對話視窗等使用者會直接接觸的輸入介面。資料樹層從組裝件的角度出發，因此在組裝模式

時可以直接新增零件或是組件，若針對其中零件做零件設計，則可以看到目前的零件設計的特徵做樹狀資料呈現。圖形層負責呈現2D與3D的資料。資料樹層與圖形層不僅能呈現結果，使用者在圖形層與資料樹層也可以觸發事件要求轉為輸入命令。控制層負責與介面層、資料樹層與圖形層對應，接收來自此三層的輸入，轉化成可以傳遞的資料，透過通訊層從客戶端傳送到伺服器端，而伺服器端將經過計算後的資料透過通訊層傳到客戶端，也可以在控制層分配給介面層、資料樹層或圖形層以呈現資訊。在伺服器端除負責溝通的通訊層外，還有橋階層與核心層。橋接層負責將使用者端傳入的要求及夾帶資訊進行分析後，呼叫核心層的對應函式，進而得到計算結果，然後再進行資料封裝，以便傳送到客戶端給接觸使用者的三層級作資料呈現。

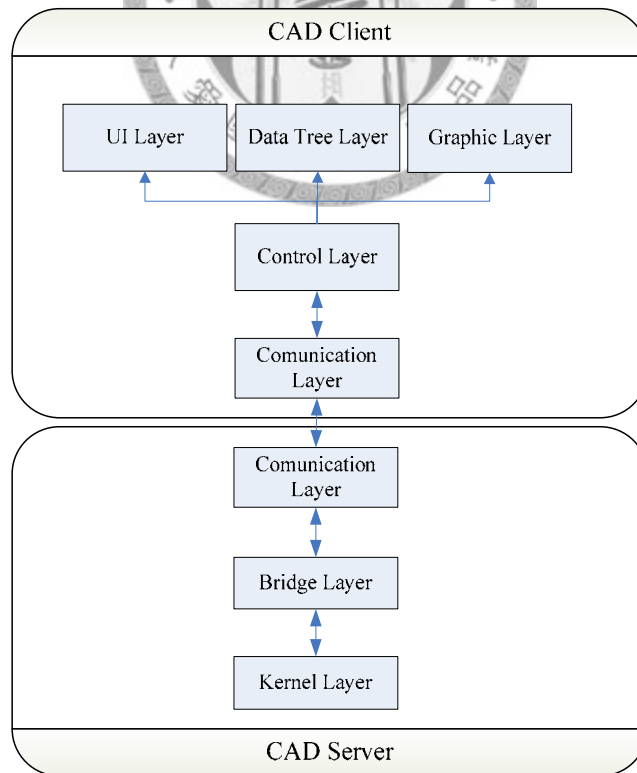


圖 3.1 在 CAD 系統中七個層級之互動情況

3.1 階層間傳遞的資料格式

為讓使用者能夠在網頁上操作CAD系統，就必需設計一個完整的資料儲存模型，來記錄傳輸資料。本論文提出之資料格式，主要記錄在客戶端所指定的指令並被傳送至伺服器端，而伺服器上計算完之成果亦可轉換成此資料格式以傳回客戶端，流程圖如圖3.2所示。其中從客戶端傳至伺服器端的資料格式主要是參考巨集型態，而從伺服器端回傳至客戶端，則針對客戶端的建構樹狀結構、3D圖形所需資料或者是對話框參數、系統自動操作客戶端的步驟...等，進行資料格式分類。資料格式分為下列三種：巨集型態傳輸資料 (Marco Type Passing Data, MTPD)、巢狀樹資料 (Nested Tree Data, NTD)和巢狀圖形資料 (Nested Graphic Data, NGD)。

使用者從客戶端輸入指令及相關資訊，然後客戶端控制層便會將指令轉換成MTPD。透過共同的定義檔案，客戶端與伺服器端可以清楚地知道所對應代號的指令定義及其夾帶的使用者輸入數值。伺服器端在接收到MTPD指定後，經由橋接層分析MTPD並呼叫核心層提供函式，經過實體核心計算結果，可以轉換成NTD、NGD和MTPD，同樣再透過共同的定義檔，讓客戶端在接收後可以分析並將資料顯示。以下將針對MTPD、NTD以及NGD做詳細的說明。

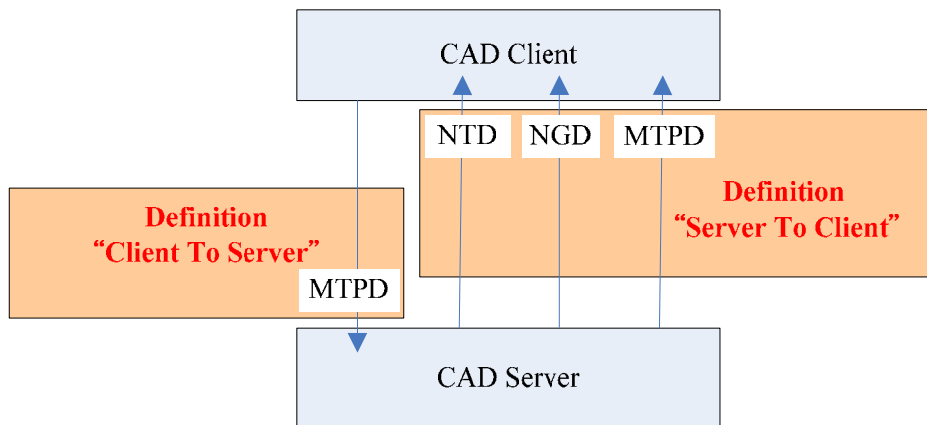


圖 3.2 CAD 客戶端到伺服器端之間資料流

3.1.1 巨集型態傳輸資料 (MTPD)

巨集型態傳輸資料，以MTPD代表之。當使用者下達的指令與CAD核心內部運作有關時，這些資訊會從客戶端傳送到伺服器端。一旦伺服器端接收資訊，會先分析資訊內容，其資料型態如圖3.3所示。operationIndex代表該指令的代號，如新增、移除、更改名稱、修改外型...等，assemblyId是指選擇的物件在組裝層級上之索引值，partId則代表該物件在零件設計層級上之索引值，valueArray則能夠包含一連串動作的指定值。MTPD的定義檔案則紀錄operationIndex的對應步驟，每個不同的功能就會對應一個不同的operationIndex。

在伺服器端分析使用者端傳來的MTPD並執行完後，當系統需要使用者端再傳入其他參數、呈現使用者端的對話框內參數或控制使用者端的一些操作步驟時，也可能再以MTPD方式傳送出去。因此，MTPD的傳遞不僅是從客戶端傳遞至伺服器端，也可以從伺服器端送往客戶端。

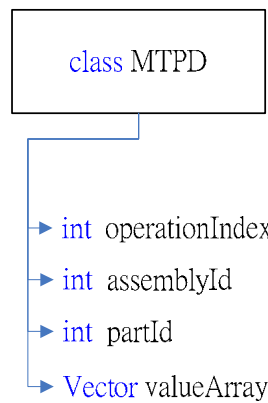


圖 3.3 MTPD 的資料格式

3.1.2 巢狀樹資料 (NTD)

巢狀樹資料，以NTD縮寫代表之。當建立模型特徵時應用到CAD核心技術，該特徵相關的資訊會被記錄下來，如特徵所在之零件實體層級、零件所在之組裝層級、特徵在該零件實體之順序、特徵名稱...等，其資料之結構如圖3.4所示。在特徵建立後，此一結構資料會被送到客戶端並且建立樹狀資料。

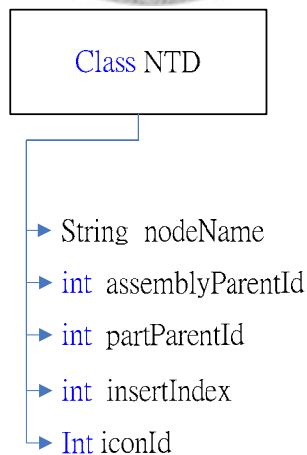


圖 3.4 NTD 的資料格式

3.1.3 巢狀圖形資料 (NGD)

巢狀圖形資料，以NGD縮寫代表之。橋接層呼叫完核心層提供的函式並建構實體模型後，便可將其轉換成三角網格並顯示在螢幕上。首先參考Wavefront公司所制訂的OBJ模型[47]，此格式不但在電腦繪圖的領域應用相當廣泛，並且資料的記錄重複性低，更有利於作為傳遞格式。NGD即為修改過的OBJ的檔案格式，有效提升與實體模型的連結性。為建立NGD與NTD之間的關連，NTD通常先被核心計算後送出，在使用者端建立好拓樸樹節點後，然後NGD才被建立。

NGD的格式內容以實體(Body)為主體，一個實體對應一個NGD檔案，而一個零件(Part)底下可能會包含至少一個以上的實體。實體底下包含面(Face)、邊(Edge)與頂點(Vertex)等資料。當面進行三角形化後，先列出每個頂點及其法線單位向量，然後再定義每個三角形面所包含的頂點編號。NGD中的每一個圖型都有編號，並與實體模型當中STEP元件的編號一致。其規格如圖3.5所示。

3.2 可調整的操作型態

本論文提出三種可變化之操作型態，分別為「單機版」、「主從式協同－客戶端與重伺服端」及「對等式協同」。

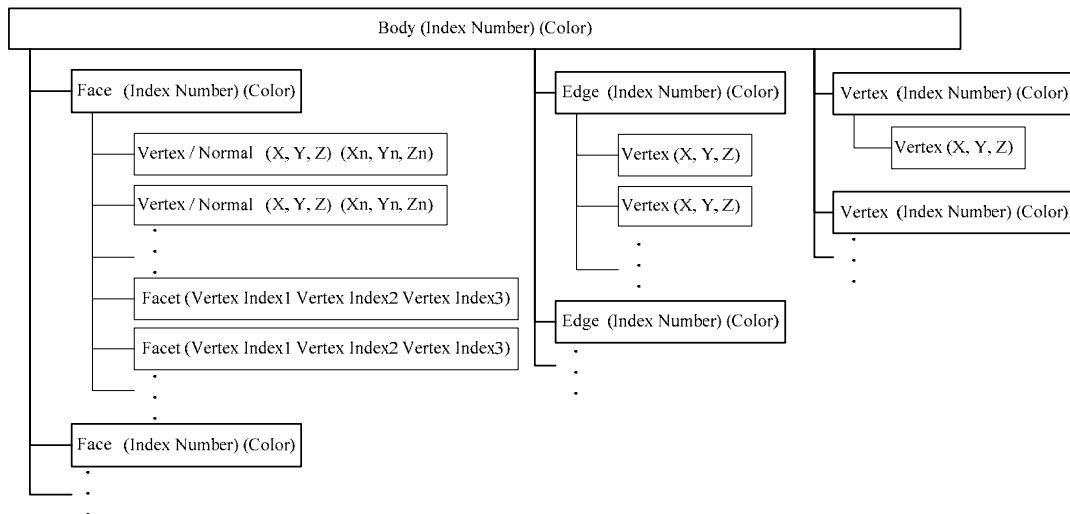


圖 3.5 修改過之 OBJ 格式

3.2.1 單機版

單機版是CAD系統最常見的類型，如CATIA、Pro/E、UG...等等。其操作環境可以無須網路，而檔案存放在使用者端。為達成此操作類型，在多層次架構中的通訊層必須被放在同一應用程式內，並且直接在程式記憶體中傳遞資料。此資料流動順序如圖3.6所示。使用者在介面層、圖形層、資料樹層下達指令後，控制層會先將指令轉換成MTPD，由於在記憶體內部傳遞的緣故，通訊層的功能已被傳遞記憶體位置所取代，橋接層則可直接拿到MTPD，在進行分析並呼叫核心層內函式支援之後，計算的結果轉換成NTD與NGD兩種格式，再藉由記憶體傳遞至控制層，並由控制層分配給圖形層與資料樹層。圖3.7顯示通訊層由伺服端與客戶端整合的架構。

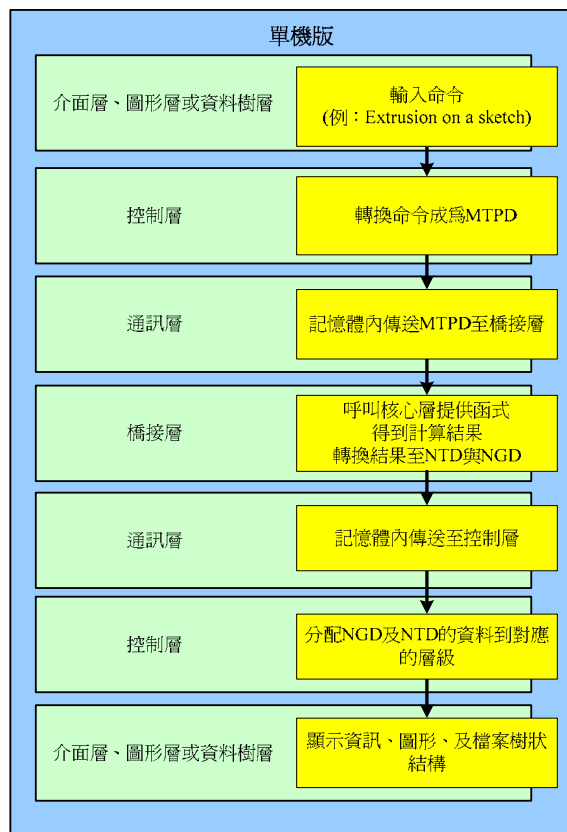


圖 3.6 單機版本的資料流動示意圖

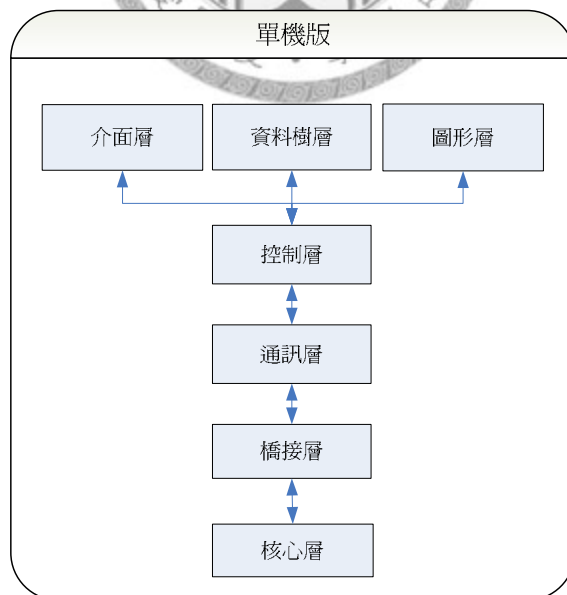


圖 3.7 單機版的架構

3.2.2 對等式協同

在此類型下之操作，每個使用者面對的是一個類似單機版的程式，但有協同功能的程式。其實仍可將此對等式協同拆開成客戶端與伺服器端來看，每個客戶端有自己對應的伺服器端，而每一個「客戶端－伺服器端」皆可組成一類似單機版的對等式協同系統。此種協同可有效減少網路傳輸上的負擔，只有MTPD是在網路上被授權的使用者發送給其餘的人。圖3.8則是模擬對等式協同中每一個「客戶端－伺服器端」互動情況的示意圖，虛線為授權的使用者所發送的MTPD，實線則為每個伺服器端計算過後回傳的結果。

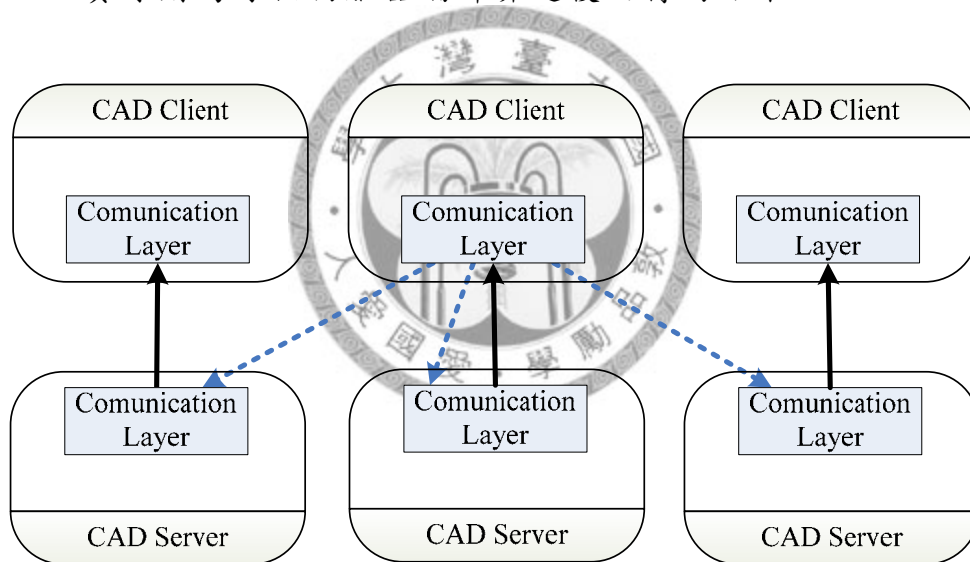


圖 3.8 對等式協同的資料流動情況

有授權的使用者可以透過自己系統的介面層、圖形層及資料樹層下達指令，透過控制層整合成MTPD後，利用通訊層發送MTPD至包含他自己本身在內的每個伺服器端來進行運算，在伺服器端的橋接層收到MTPD之後，每個系統會利用本身的核心層所提供運算能力來進行計算，而各伺服器端計算完成之結果經過轉換變成NTD與

NGD，將會直接內部回傳到各客戶端，建立其結果圖形及樹狀結構資料，此資料流動順序如圖3.9所示。

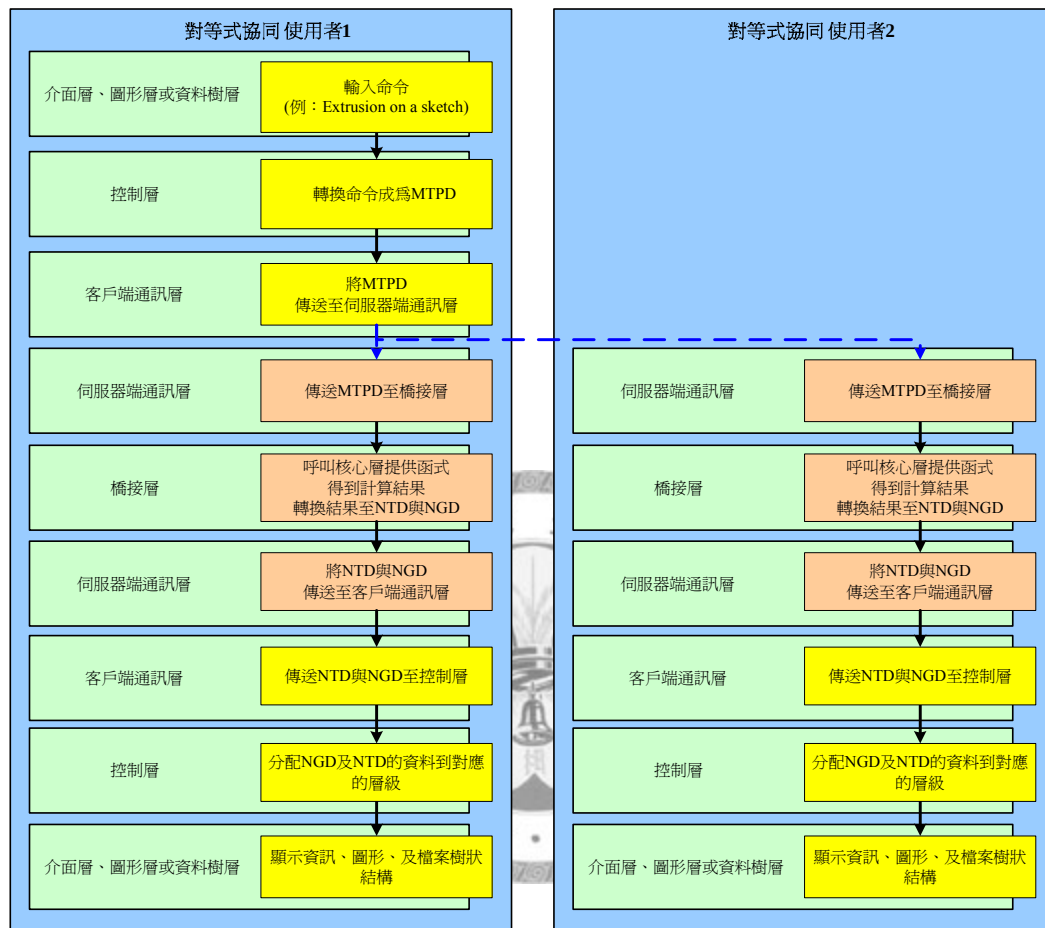


圖 3.9 對等式協同的資料流動示意圖

3.2.3 主從式協同－輕客戶端與重伺服器端

在此模式下，客戶端的程式被縮到最小，而一個伺服器端可以連接到許多客戶端。唯有授權的客戶端可以要求伺服器端執行某些特定動作，而這些指令由一個MTPD或數個MTPD組成，從客戶端發送後，經過通訊層到伺服器端。在伺服器端的橋接層可以收到指令

並且跟核心層溝通並請求計算結果，得到結果之後橋接層可以產生出一些NTD與NGD。這些資料會透過伺服器端通訊層被送回到客戶端。客戶端通訊層將資料傳給控制層，並且在此分配資料至資料樹層和圖形層。核心相關的拓撲及幾何計算僅在伺服器端發生，藉由此動作，所有的客戶端可以接收到同樣的資料，並且顯示同樣的圖形外觀。圖3.10顯示此運作模式的資料流動順序。

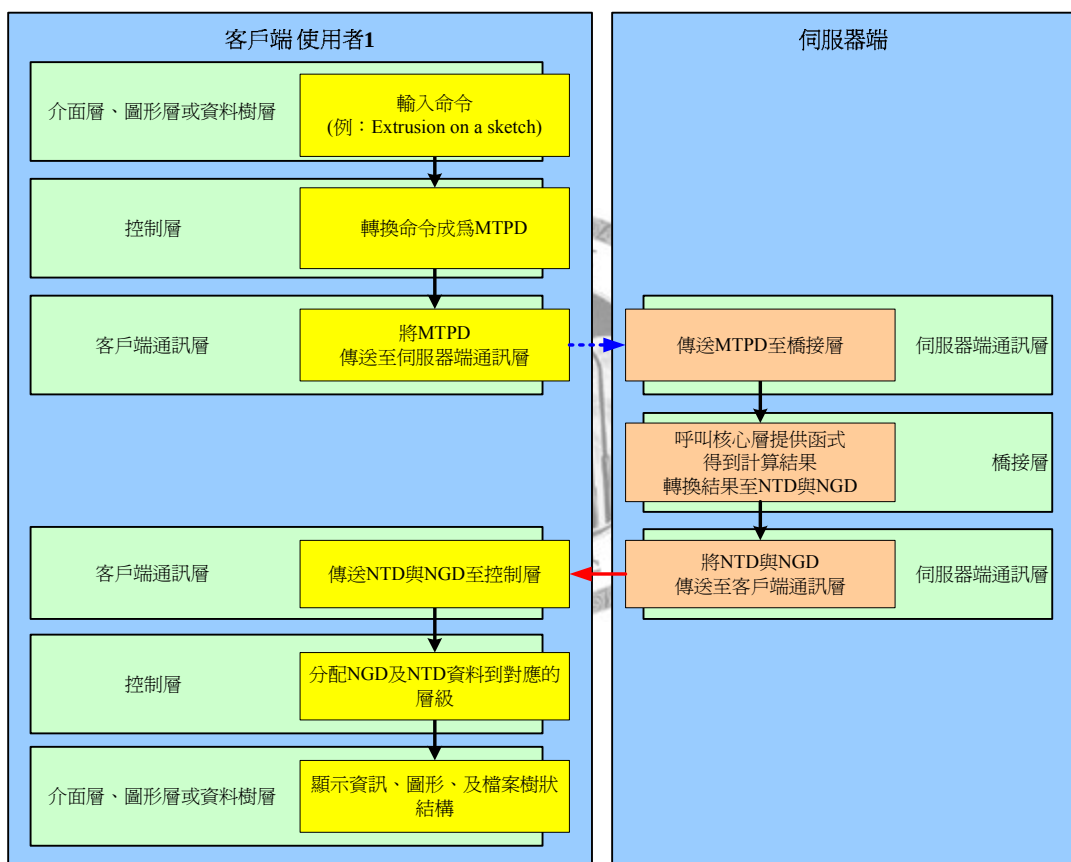


圖 3.10 精簡主從式架構的資料流動順序示意圖

客戶端與伺服器端透過通訊層進行溝通，無論是客戶端發出的MTPD或者是伺服器端回傳的NTD、NGD，都可以進行複製，而從伺服器回傳的NTD、NGD可以在複製數份之後，分別傳回至與伺服器通訊層有建立連結的客戶端，則這些客戶端將可以在沒有發出任何

MTPD的情況下得到計算的結果。不過NTD與NGD的資料量遠大於MTPD，使用此種模式需要考慮網路環境。模擬伺服器與多客戶端的互動情況顯示如圖3.11所示。

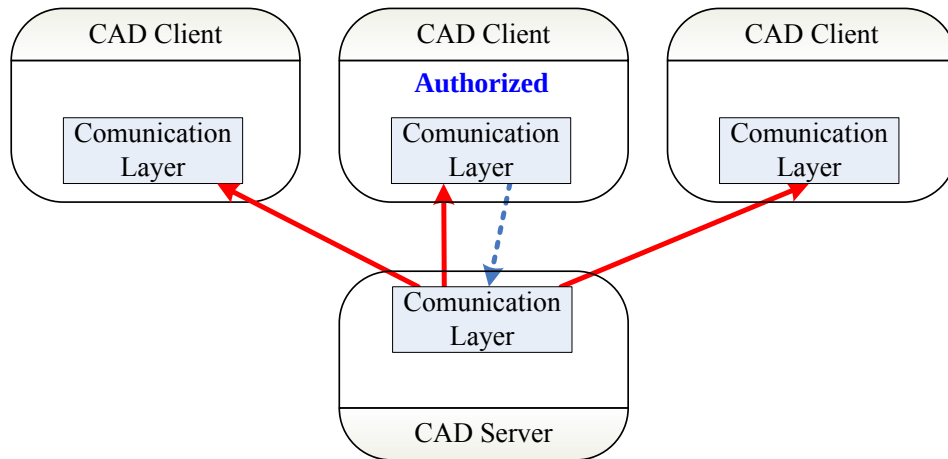


圖 3.11 模擬在精簡主從式架構中伺服器與多客戶端的互動情況

3.3 實作CAD系統

本節將討論使用本論文所提出的架構建構一個CAD系統。首先，必須確定此系統能夠在最少改變的情況下，根據使用者的指定輕易地變換其操作型態。

以JAVA為主要程式語言，撰寫介面層、控制層、資料樹層及橋接層，並利用JAVA 3D實作圖形層，而通訊層則採用JSDT (Java Share Data Toolkit)來實作，最後利用尤春風博士所開發的SpringSolid CAD核心JAVA版本來實作核心層。

3.3.1 實作單機版

為建立一個單機版本，可以將架構內的所有層級放在同一個應用程式內，資料也可以直接在記憶體內傳遞而不必透過網路。市面上商用CAD系統大多為此種模式。SpringSolid CAD系統框架如圖3.12所示。使用者僅能看到介面層的整體框架、選單、工具列以及左手邊的資料樹層與右手邊的圖形層，除此三個層級是使用者可接觸到外，其餘包含控制層、通訊層、橋階層及核心層則都是在程式端運作。

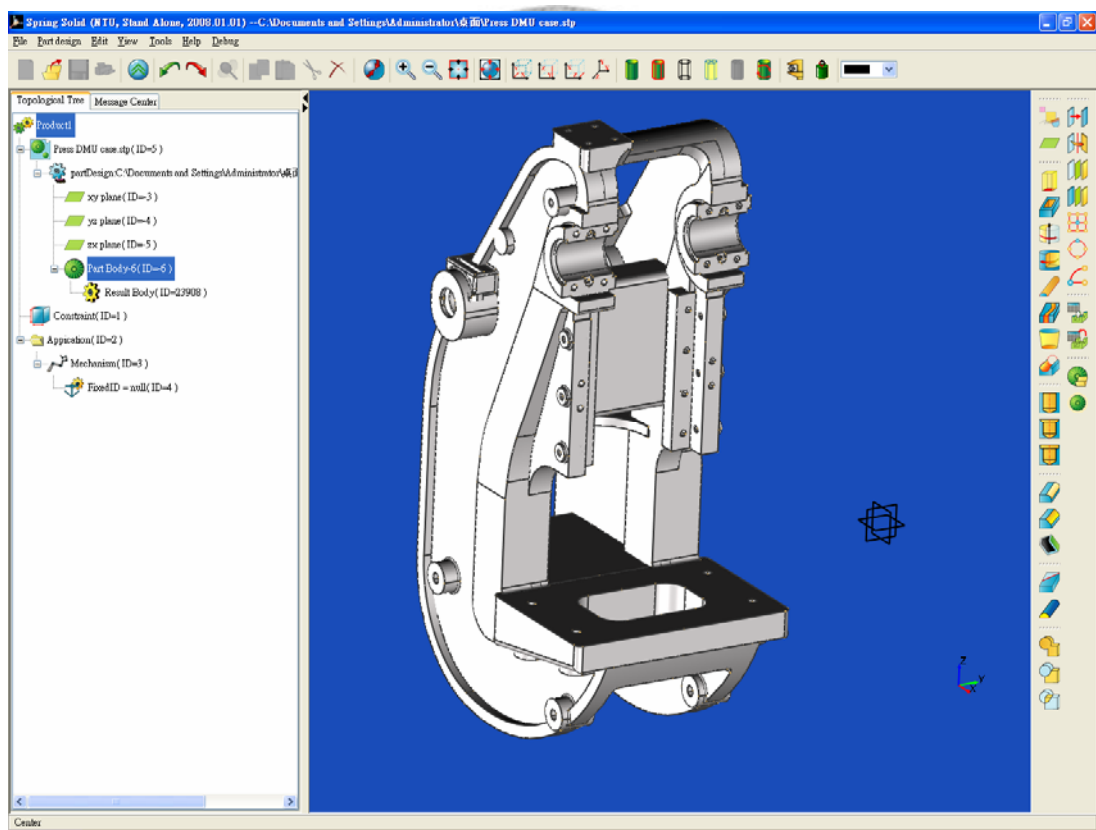


圖 3.12 SpringSolid CAD 單機版本

3.3.2 實作對等式協同

對等式協同可以節省網路的資源，是一種非常經濟的協同方式。此種協同僅有MTPD在網路間被傳送，而NTD與NGD的傳送則是透過應用程式內部的記憶體共用。在協同過程中，首先其中一方必須先開啟伺服器程式，如圖3.13(a)所示，接著另一端再進行連接，使得兩邊的使用者名單一致，如圖3.13(b)(c)所示。其協同過程如圖3.14所示，在圖3.14(a)當中兩端同時開啟一個檔案，而在圖3.14(b)中，其中一人下達指令做草圖之新增。

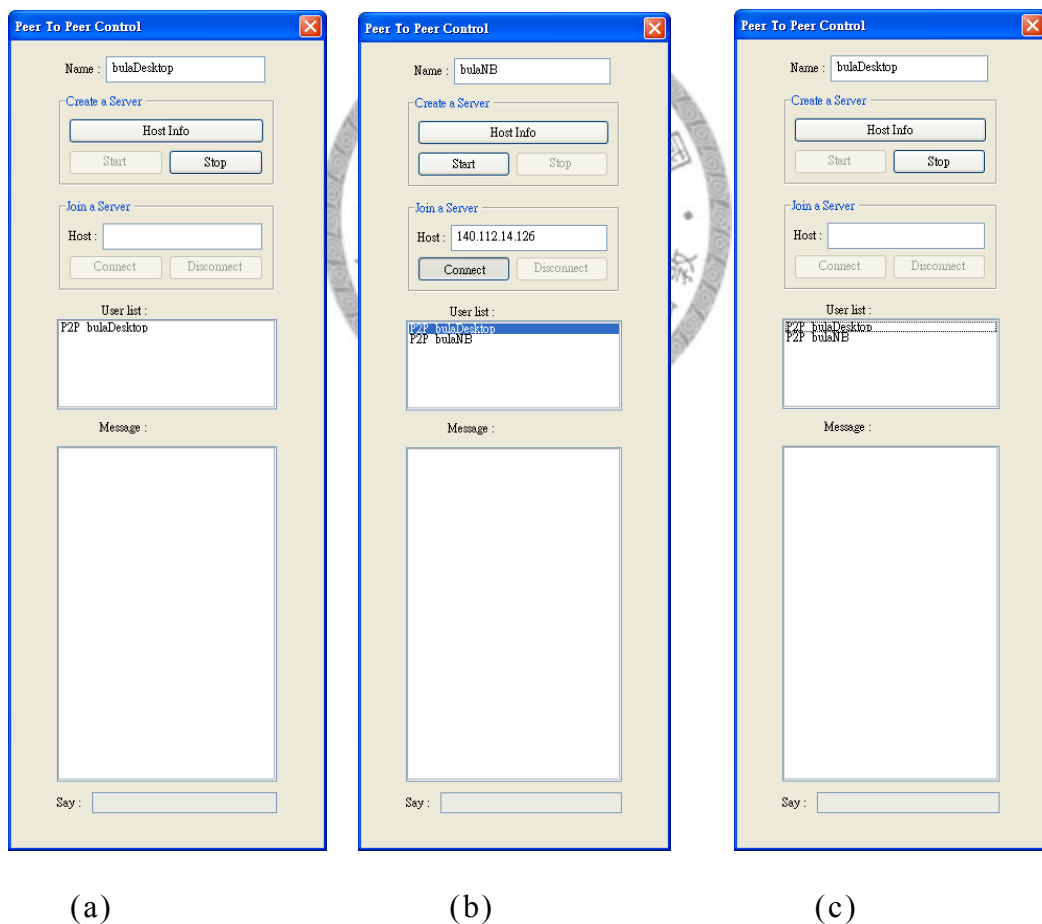
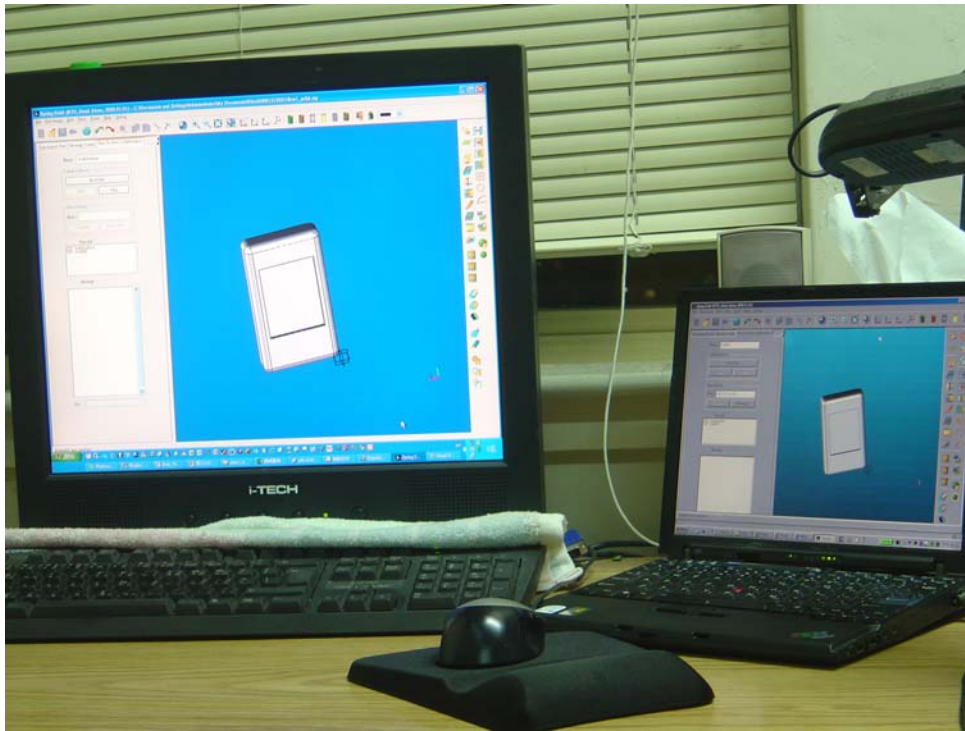
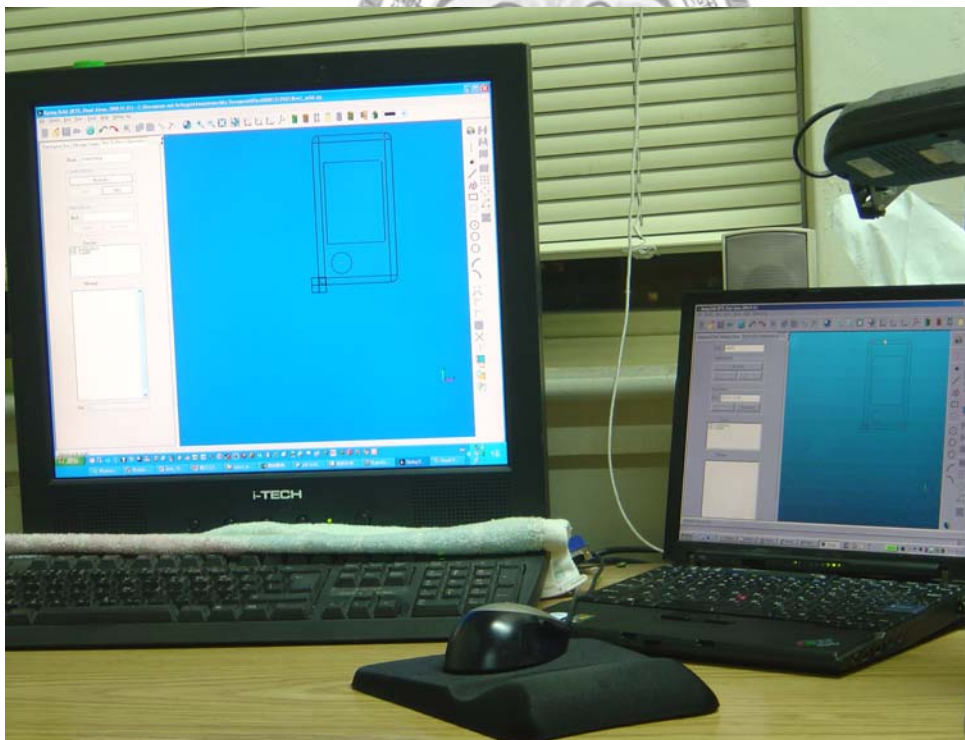


圖 3.13 對等式協同中，使用者開啟伺服器並建立連結



(a)



(b)

圖 3.14 對等式協同之協同過程

3.3.3 實作主從式協同一輕客戶端與重伺服器端

伺服器端包含下述功能：接收MTPD，實體模型計算，將結果轉換成NGD與NTD並發送至使用者端。使用者無須太多的操作技巧，因此介面設計如圖3.15所示。

為了讓每個使用者方便操作，客戶端系統用Applet的格式呈現，被置放在網頁瀏覽器的框架中，使用者只要點擊工具列、指令列或是在對話視窗填入適當資訊，接著就可以得到計算結果，程式顯示如圖3.16所示。



圖 3.15 簡單的伺服器端使用者介面

3.3.4 實際案例執行時間比較

為了進行對等式協同與主從式協同的比較，首先繪製兩者的主要計算順序，如圖3.17所示。兩者同樣都是由操作者開始，進行步驟如下：（A）進行讀取檔案傳遞（B）計算並產出三角形網格資料（C）將三角形網格資料放到使用者螢幕上。

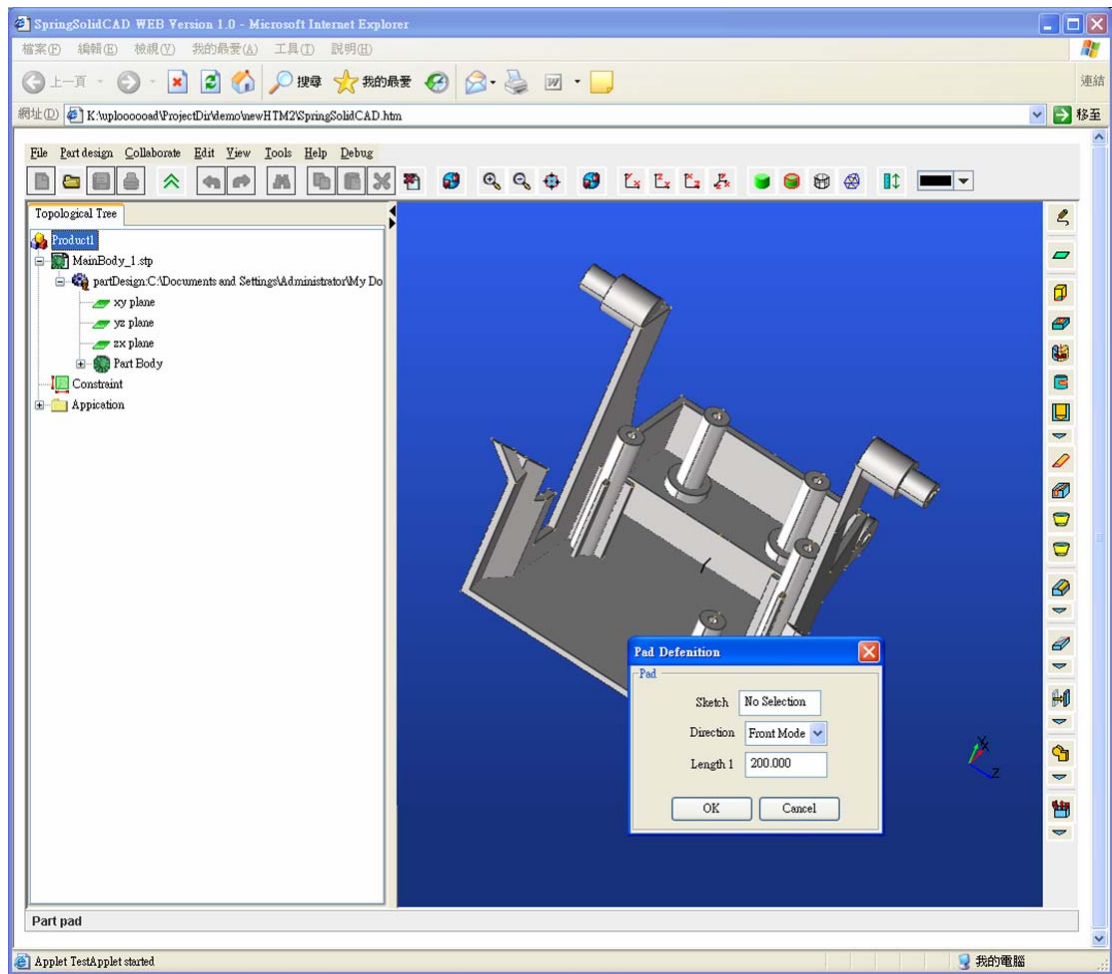


圖 3.16 客戶端的網頁畫面

假設協同的使用者數量為 N ，主從式協同的使用者從開啟檔案到看到檔案所花費的時間是 $A1+B1+N*C1$ 。相同動作在對等式協同所花費的時間則是 $A2*(N-1)+B2+C2$ 。若使用的電腦運算速度皆相同，則 $B1$ 與 $B2$ 相同。另外，若讀取檔案之大小與網路傳遞速度相同，則可以認定 $A1$ 與 $A2$ 相同。當使用者數量為兩人時，此兩種協同方式的時間差約為 $2*C1-C2$ ，其中 $C2$ 代表在電腦內部傳送的時間，可以不列入考慮，因此產出的三角形網格的精細度與網路傳遞的速度，決定主從式協同與對等式協同之間的時間差異。

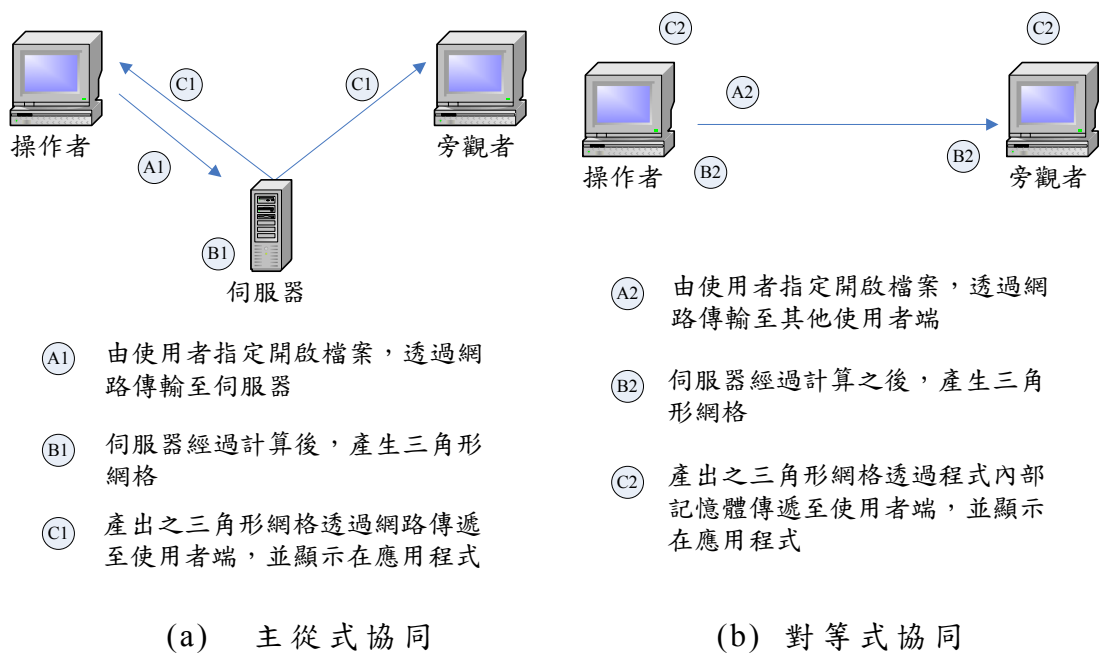


圖 3.17 不同協同狀態下，使用者操作之流程

以下本研究針對開啟檔案來比較其執行時間，對等式協同利用兩台電腦來執行，而主從式協同則須用三台電腦來執行。受測檔案為一沖床機台基座之零件STEP檔案。在網路速度方面，分別以1024KB/s、256KB/s及48KB/s來測試協同的流暢度，其中1024KB/s模擬10M網路卡的極限值(10Mb網卡極限 80% $10240\text{bps} / 8 * 0.8 = 1024\text{KB/s}$)，256KB/s則為光纖提供上傳速度2M，48KB/s則是3G手機傳輸的384Kbps。

紀錄的過程主要從使用者指定讀取檔案，計算核心開始讀取到產生三角形網格資料，到最後系統將三角形網格呈現到螢幕上，如流程圖3.18所示。

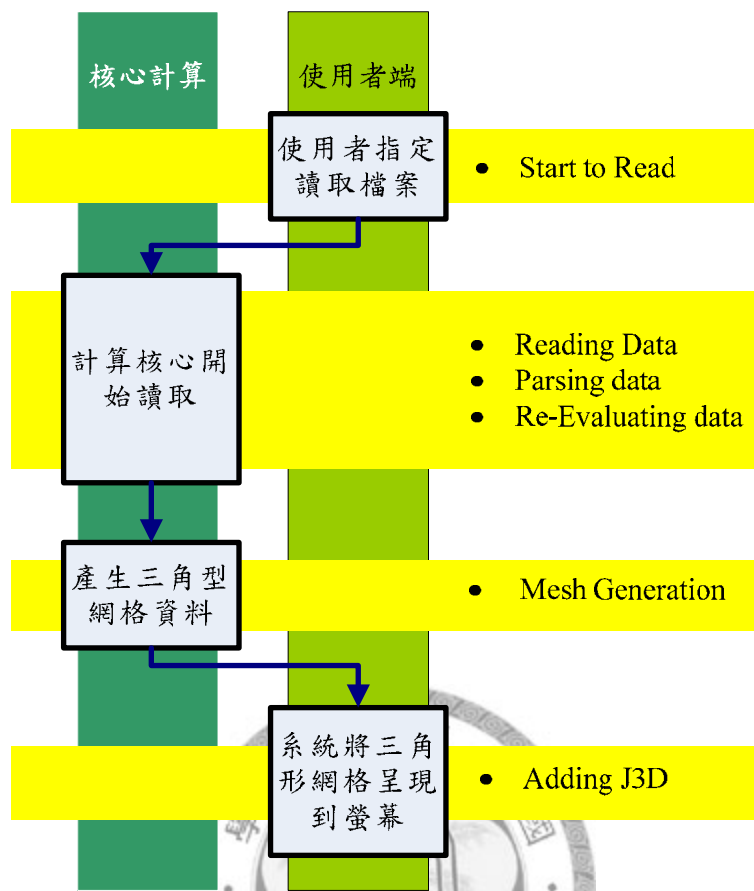
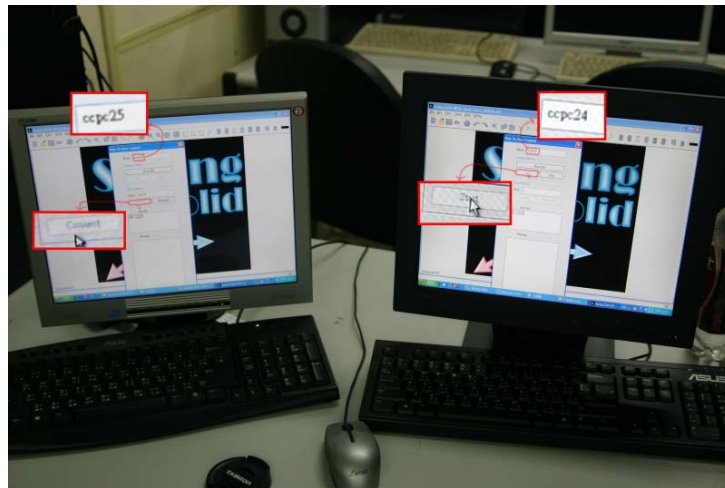


圖 3.18 系統開啟檔案流程圖

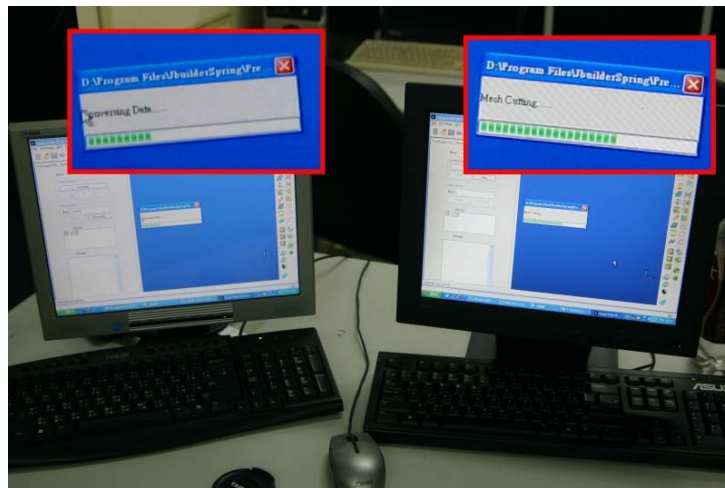
此次測試共使用三台電腦，配備皆為: Intel Core 2 Duo CPU E8300 @ 2.83GHz，2GB RAM，Windows XP 作業系統。主要測試檔案為 Press DMU case.stp，檔案大小為 1354KB。

3.3.4.1. 對等式協同

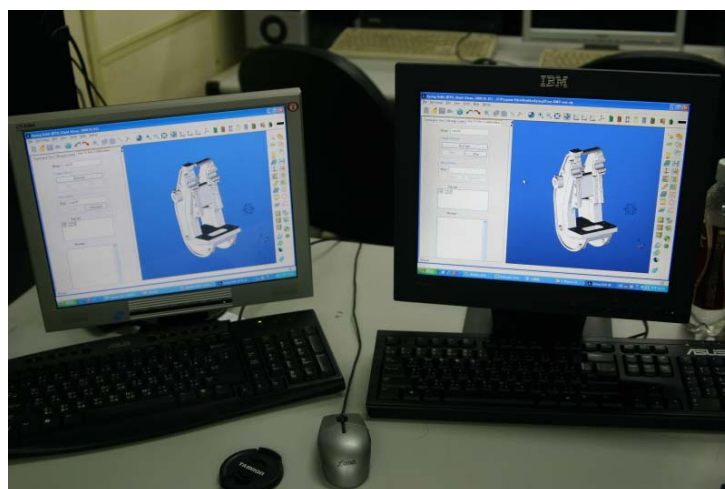
首先利用對等式協同 CAD 同時開啟檔案，如圖 3.19 所示。



(a) 兩邊使用者登入



(b) 開啟檔案時因網路環境而有速度不同



(c) 檔案開啟完成

圖 3.19 以對等式協同開啟測試檔案

表3.1~3.3分別紀錄在不同的網路速度環境下，對等式協同開啟檔案的比較時間表。

表3.1網路速度1024KB/s之對等式協同開啟比較時間表

電腦		PC1 (sec)	PC2 (sec)
過程紀錄	Disposing File	3.907	4.016
	Reading Data	0.546	0.625
	Parsing Data	1.219	1.312
	ReEvaluating Data	2.813	2.954
	Generating Mesh	17.828	17.609
	Sending NGD and NTD	0.094	0.078
	Analyzing NTD	1.25	1.438
	Adding NTD to J3D	5.703	3.437
第一次 時間長度		33.36	31.469
第二次 時間長度		33.453	31.172
第三次 時間長度		33.583	31.375
時間長度 三次平均		33.583	31.375

表 3.2 網路速度 256KB/s 之對等式協同開啟比較時間表

電腦		PC1 (sec)	PC2 (sec)
過程紀錄	Disposing File	5.344	5.563
	Reading Data	0.531	0.625
	Parsing Data	1.281	1.562
	ReEvaluating Data	2.891	3.031
	Generating Mesh	17.187	17.61
	Sending NGD and NTD	0	0.078
	Analyzing NTD	1.329	1.422
	Adding NTD to J3D	5.734	3.437
第一次 時間長度		34.297	33.328
第二次 時間長度		34.546	33.125
第三次 時間長度		34.438	32.125
時間長度 三次平均		34.427	32.859

表 3.3 網路速度 48KB/s 之對等式協同開啟比較時間表

電腦		PC1 (sec)	PC2 (sec)
過程紀錄	Disposing File	15.391	28.328
	Reading Data	0.547	0.641
	Parsing Data	1.281	1.312
	ReEvaluating Data	3.047	3.204
	Generating Mesh	17.906	17.015
	Sending NGD and NTD	0.094	0.078
	Analyzing NTD	1.265	1.407
	Adding NTD to J3D	5.782	3.453
第一次 時間長度		45.313	55.438
第二次 時間長度		44.390	54.984
第三次 時間長度		44.594	55.813
時間長度 三次平均		44.766	55.412

從表3.1到表3.3的比較，可以看出使用者開啟檔案(Start to Read)到系統讀取檔案(Start Reading Data)之間的時間由4秒、5.5秒到28.3秒，隨著網路速度下降，此階段因為用到網路傳輸而使花費時間慢慢增加。PC1在本機作開啟動作，理論上開啟速度不受網路環境影響，但是因為要將檔案傳遞至PC2，PC1亦要花不同時間將檔案傳出，也使得PC1速度稍微變慢。三角形網格呈現在螢幕的速度PC2也略大於PC1，所以使得在網路速度較快的情況下，PC2甚至在接收完檔案後，PC2還比PC1還快顯示結果。

3.3.4.2.主從式協同一重客戶端與輕伺服端

接著利用主從式協同來開啟檔案，如圖3.20所示。



(a)



(b)



(c)



(d)

圖 3.20 主從式協同一重客戶端與輕伺服器端 開啟受測檔案

(a)中間電腦開啟伺服器端，兩旁電腦開啟客戶端網頁

(b) 客戶端網頁與伺服器端連結，使用者可看到 CAD 系統畫面

(c)開啟檔案過程中，伺服器端計算中，而使用者端可看到進度表

(d)客戶端檔案開啟完成

表 3.4~3.6 分別紀錄在不同的網路速度環境下，主從式協同開啟檔案的比較時間表。

表 3.4 網路速度 1024KB/s 之主從式協同開啟比較時間表

電腦		PC2 (sec)	PC1 (sec)	PC3 (sec)
過程紀錄	Disposing File			1.984
	Reading Data		0.641	
	Parsing Data		1.078	
	ReEvaluating Data		2.703	
	Generating Mesh		20.797	
	Sending NGD and NTD		0.094	
	Analyzing NTD	47.899		23.453
	Adding NTD to J3D	3.679		5.953
第一次 時間長度		78.875		56.703
第二次 時間長度		77.704		54.422
第三次 時間長度		78.906		57.422
時間長度 三次平均		78.495		56.182

表 3.5 網路速度 256KB/s 之主從式協同開啟比較時間表

電腦		PC2 (sec)	PC1 (sec)	PC3 (sec)
過程紀錄	Disposing File			5.859
	Reading Data		0.641	
	Parsing Data		1.062	
	ReEvaluating Data		2.672	
	Generating Mesh		20.578	
	Sending NGD and NTD		0.078	
	Analyzing NTD	64.141		32.61
	Adding NTD to J3D	3.797		7.953
第一次 時間長度		98.828		71.453
第二次 時間長度		98.719		69.375
第三次 時間長度		98.351		69.421
時間長度 三次平均		98.633		70.083

表 3.6 網路速度 48KB/s 之主從式協同開啟比較時間表

電腦		PC2 (sec)	PC1 (sec)	PC3 (sec)
過程紀錄	Disposing File			28.75
	Reading Data		0.64	
	Parsing Data		1.094	
	ReEvaluating Data		2.688	
	Generating Mesh		20.531	
	Sending NGD and NTD		0.062	
	Analyzing NTD	321.438		170.625
	Adding NTD to J3D	20.297		6.094
第一次 時間長度		395.50		230.484
第二次 時間長度		379.187		245.828
第三次 時間長度		395.312		230.203
時間長度 三次平均		390.016		235.505

由結果可知，主從式協同雖然可以讓使用者不安裝即可直接操作，但需要消耗較大網路資源來做額外的三角形網格傳遞，如果在網路速度較低的環境下，使用此種協同模式可能會耗時較久。

3.4 市面上現行CAD系統之比較

由於市面上也有許多CAD系統提供協同功能，本節針對對等式協同與主從式協同，各找出一套商業應用軟體來進行功能之研究。

3.4.1 對等式協同

以CATIA為例，使用Net Meeting來進行對等式協同。使用者可以從選單Tools→Conferencing，點選host以建立會議，或點選Guest

以加入會議，如圖3.21所示。

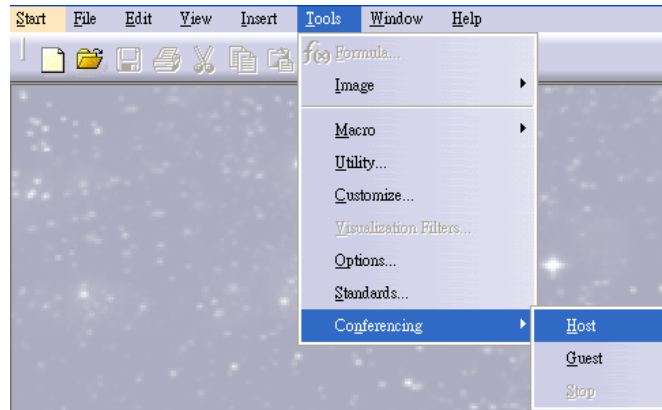


圖 3.21 CATIA 中開啟對等式協同

接著可以從Net Meeting來呼叫對方，如圖3.22所示，一方要求對方回應後即可建立連線。

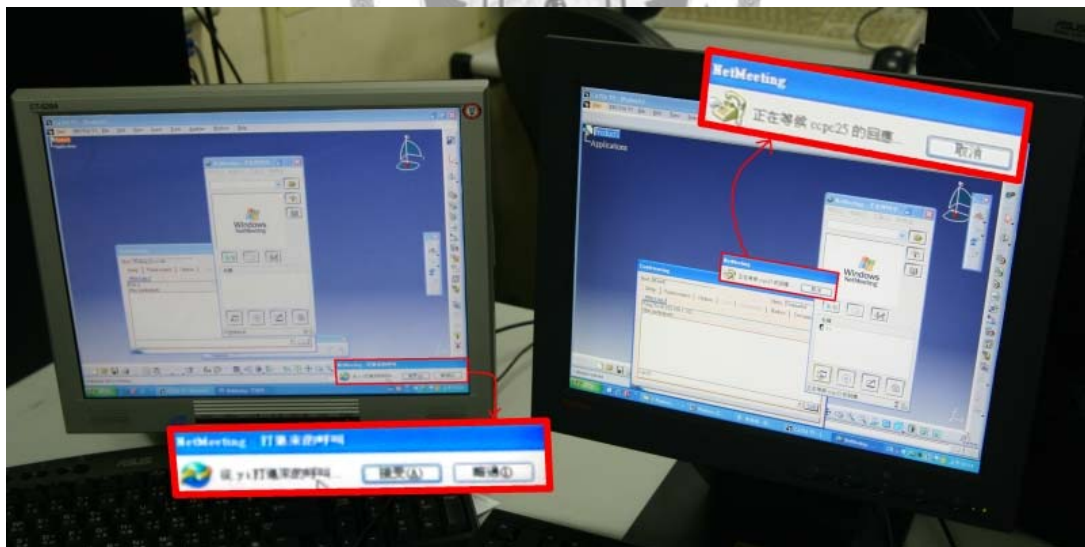


圖 3.22 使用 Net Meeting 來進行對等式協同

建立連線之後，調整完會議設定後，可按下開始視訊會議的按鈕，開始會議。共同開啟檔案只能透過檔案傳輸，如圖3.23所示，

當兩邊開啟相同檔案之後，就可以共同瀏覽、設計，如圖3.24所示。

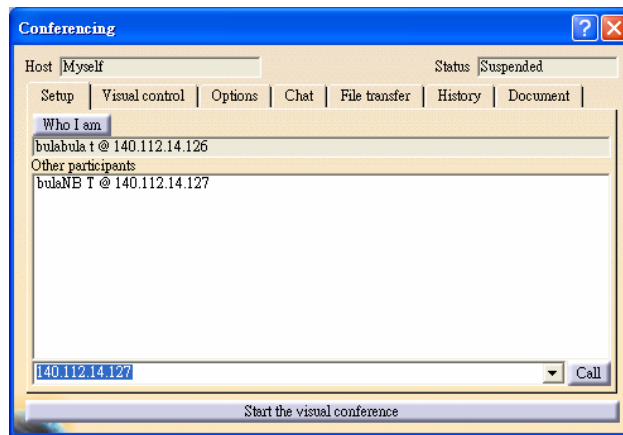


圖 3.23 CATIA 協同會議控制視窗

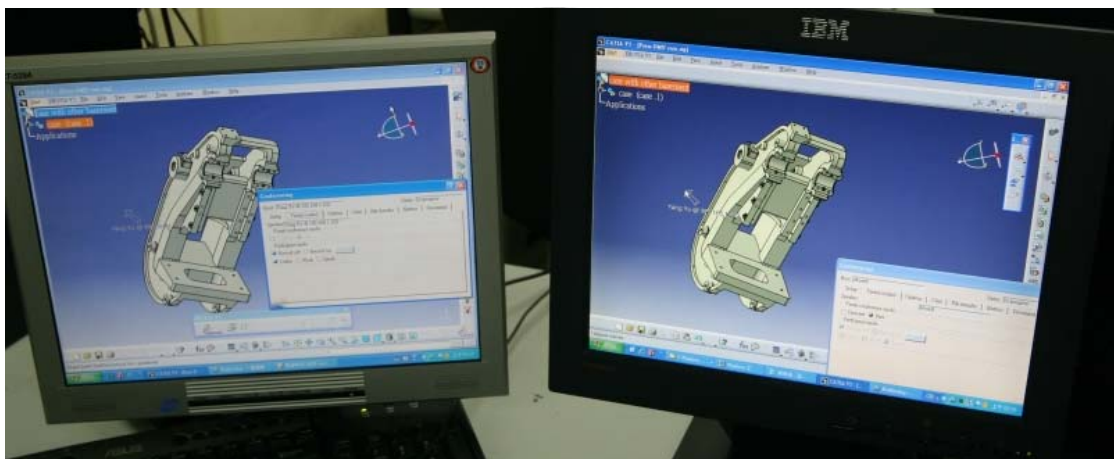


圖 3.24 CATIA 兩邊可以同時開啟同一檔案，進行協同討論

3.4.2 主從式協同

主從式協同，以AutoVue為例。首先在伺服器端開啟伺服主程式如圖3.25之後，在客戶端再開啟網頁。使用者便可以透過客戶端的網頁直接操作AutoVue，如圖3.26所示，網頁端只有使用者介面，真正的計算核心仍在伺服器端。

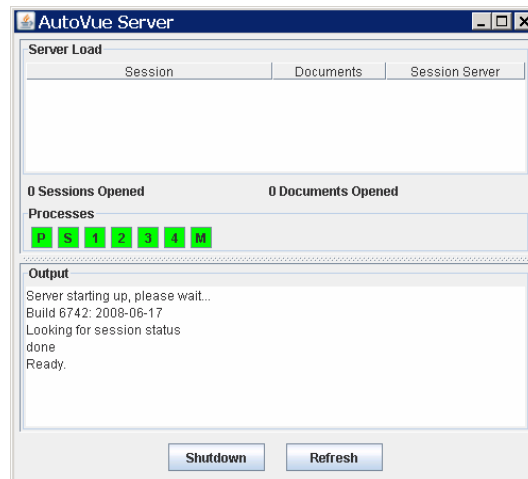


圖 3.25 AutoVue 伺服器端程式

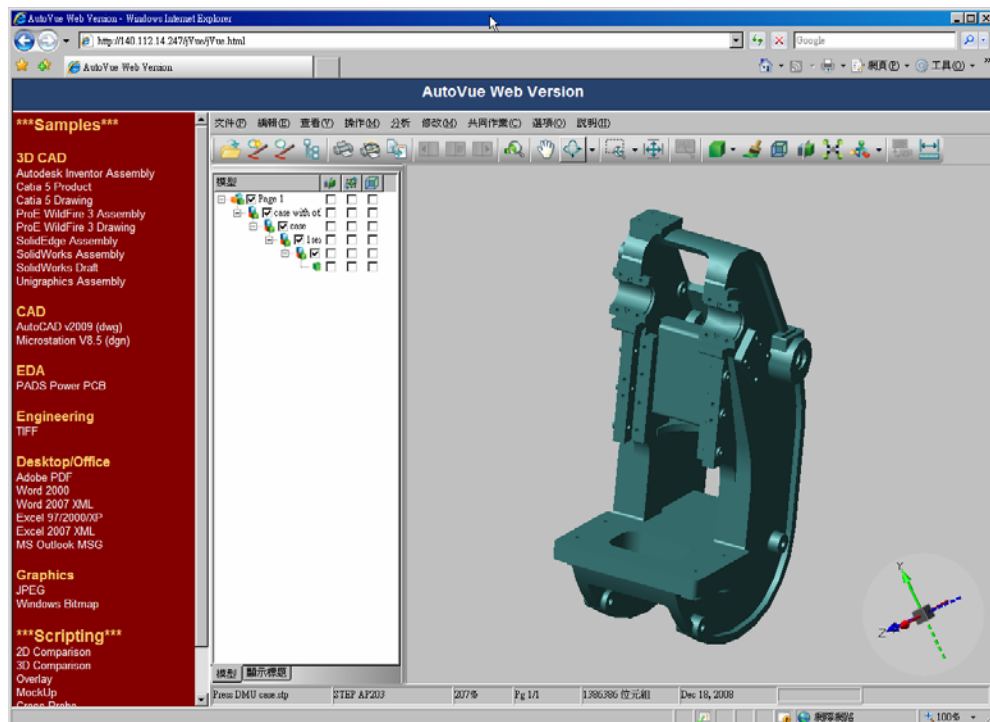


圖 3.26 直接在使用者端開啟網頁讀取 CAD 模型

除可以使用網頁開啟CAD模型以外，使用者更可以利用其主從式協同的架構，透過網頁進行協同討論。使用者首先開啟會議，輸入會議相關資訊，如圖3.27所示。接下來兩端的使用者便可透過加入同一會議而進行網路同步瀏覽模型，如圖3.28所示，另外AutoVue

還提供使用者標註的功能，如圖 3.29，方便使用者作協同討論。

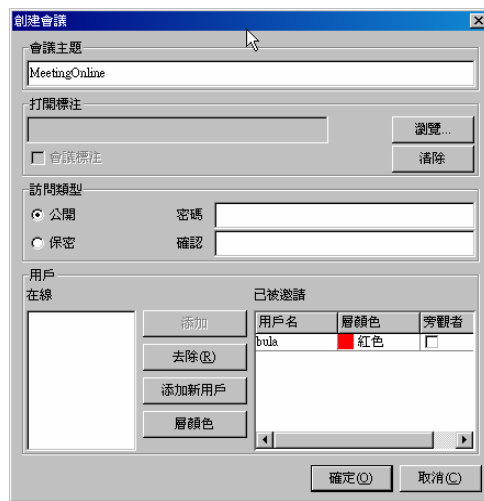


圖 3.27 AutoVue 進行協同前，必須先設定會議主題並開啟會議

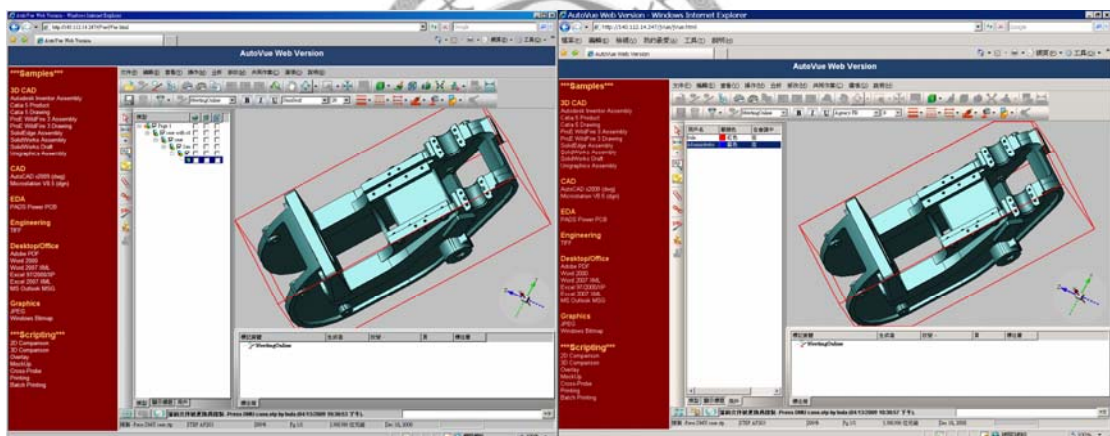


圖 3.28 兩台電腦的使用者加入同一個會議，進行協同討論

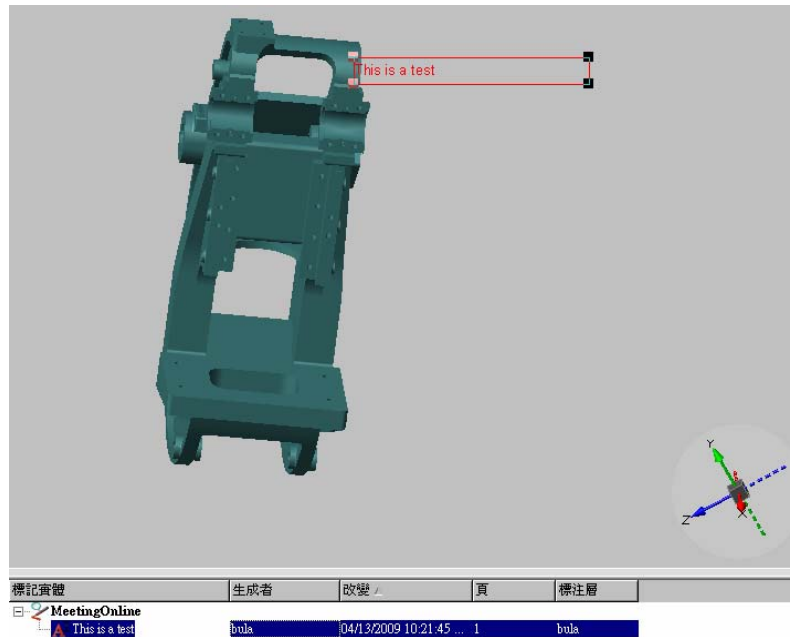


圖 3.29 提供 3D 標註方便使用者進行協同討論

3.4.3 協同比較

以CATIA為代表的對等式協同，為一般的商業CAD系統主要提供之協同功能，使用者除自行在本機之單機版軟體進行繪圖之外，還可以透過協同功能與其他使用者連結，進行討論。使用者可以使用與繪圖時相似的操作介面並可以直接與他人互動，但缺點是兩邊的使用者必須都要安裝相同的軟體，才可以進行協同。

以AutoVue為代表的主從式協同，將精簡客戶端放置在網頁上，透過與伺服器端的連結，使用者便可以瀏覽3D模型，不需要額外安裝軟體。另外還可以開啟會議，使會議中的使用者可以同步地瀏覽、註記CAD模型，但AutoVue無法直接針對CAD模型作編修，僅能算是功能強大的協同CAD模型瀏覽器。此外，因為採用網頁當作精簡客戶端，在協同的過程當中，系統需傳遞大量的圖形資料，因此協同

的反應相較於先前提及的對等式協同來說，有些遲緩。

兩種截然不同的協同模式，各有利弊，本章所提出之彈性協同架構綜合此兩種協同模式，既能夠讓都有安裝CAD的使用者，進行對等式協同，又可以讓沒有安裝此CAD系統的人，如上、下游廠商可以透過網頁達到共同設計的成果。

以下簡單比較CATIA、AutoVue以及本研究所提出之架構。

表3.7 CAD比較表

	CATIA	AutoVue	本文
客戶端是否需要安裝	O	X	X
使用者可否編修模型	O	X	O

O 代表『是』；X 代表『否』

第四章 協同平台與動態協同紀錄

協同設計在不同的群組之間執行時，系統若能自動記錄整個協同過程，對於事後想要重新瀏覽設計的使用者來講相當有意義。許多以網頁為基的CAD 研究並沒有提出共同設計的過程紀錄，而本章將以本文第三章中所提出之多層次彈性架構建構一個協同平台，並且在網頁上提供協同過程之動態回顧。PDM系統被放置在協同平台資料傳輸要徑上，所有資料傳遞都會經過PDM系統，因此可過濾來自使用者端對於伺服器端的資料，避免重複或無意義的指令。並且PDM系統也可以記錄產品設計時每個不同版本的資訊，提供使用者每個步驟的細節紀錄。這些資料可以在任何時間被瀏覽，還可以重新建構為模型，以供非同步協同設計使用。

在網頁為基的系統中，檔案傳輸格式扮演相當重要的角色，將使用3.1節所提出的資料格式。另外利用PDM系統來管理傳輸檔案，除有紀錄功能外，並可動態播放共同設計的記錄資料，詳細的內容將在第4.3節做說明。

4.1 網頁為基的架構

本章提出之協同平台採取三層式架構，可以概略分為使用者介面、伺服器介面以及資料庫，如圖4.1所示。本研究採取JSP/Servlet與Applet來建立使用者介面，其中JSP/Servlet負責接收使用者下達的命令以及表示伺服器端的計算結果，而Applet則可以彩現3D模型、幾何拘束、工程註記等。伺服器端除包含SpringSolidCAD的實體模

型計算核心以外，另外利用JSDT技術來管理資料交換與網路溝通，負責回應客戶端的網頁。資料庫端則採用 Microsoft 的 SQL server 當作資料庫系統。

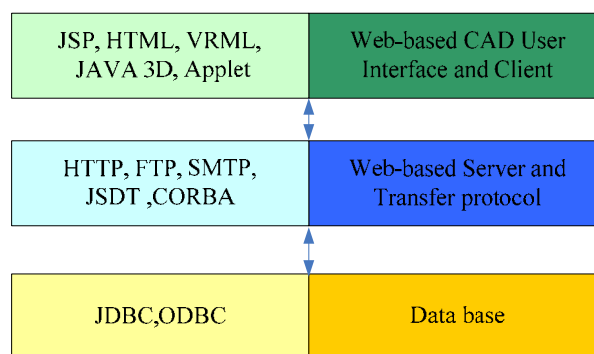


圖 4.1 三層式架構一般的元件佈局

當協同平台將CAD與PDM系統做結合，PDM系統與CAD系統需要互相傳遞資料，而此時引用第三章所提及之多層次彈性架構，如圖3.1，利用其中的通訊層做資料交換。另外新增CAD—PDM通訊伺服器來連結兩系統間之資料傳遞。圖4.2即為多層次彈性架構與三層式架構之組合架構圖。

4.2 作傳送用途的資料格式

藉由一致的參數定義，指令、計算結果等資料不但可以最小化，還可以簡單地傳送。輸出的資料與巨集資料相似，包含使用者輸入的相關資訊及指令，以及目前工作的指標。此部分已在3.2節詳細說明。

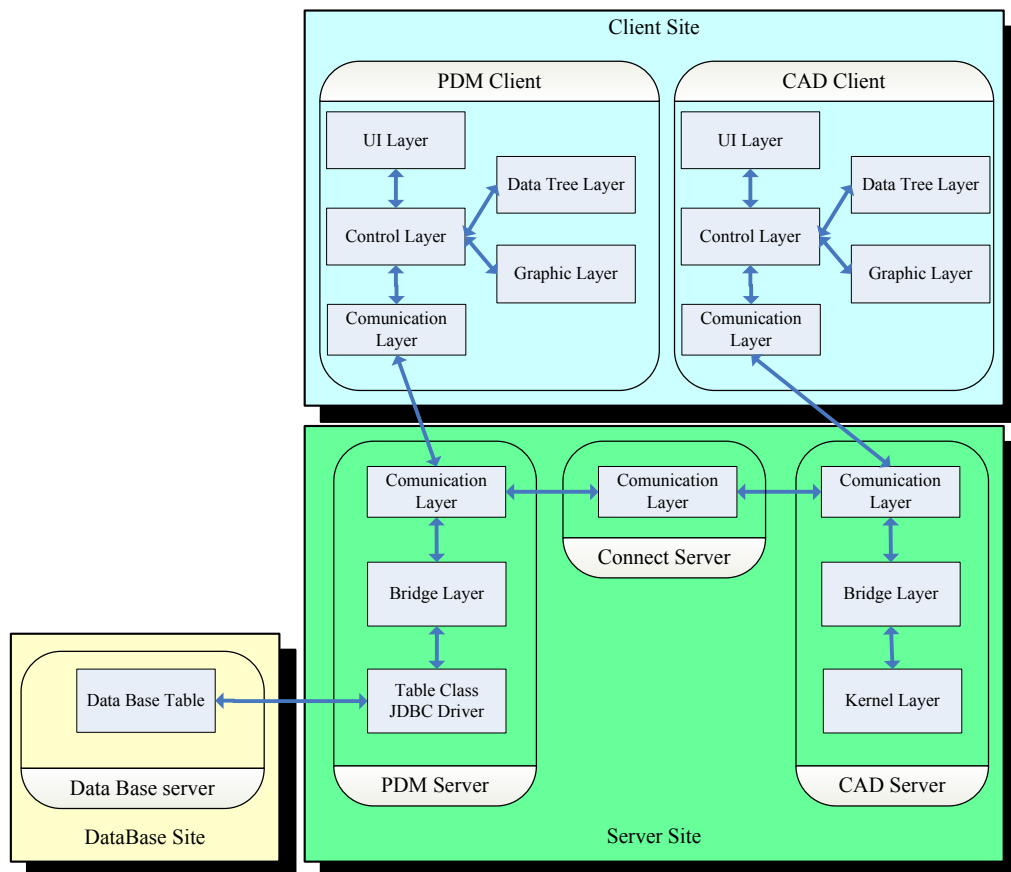


圖 4.2 多層次彈性架構與三層式架構之組合架構圖

4.3 非同步專案設計及動態回顧

4.3.1 代理伺服器

延伸代理伺服器的概念並與協同平台結合，本研究將PDM系統放置在協同設計時資料傳輸要徑上，過濾使用者端或伺服器端送往另外一端的資料，消除重覆或不必要的指令資料，如圖4.3所示。經常使用的資料、網頁的輸入、伺服器計算後的輸出都會儲存在PDM伺服器上以回應使用者的需求。只要需要，隨時可觀看資訊，而這些資訊可以動態播放共同設計的歷史，甚至重建模型。

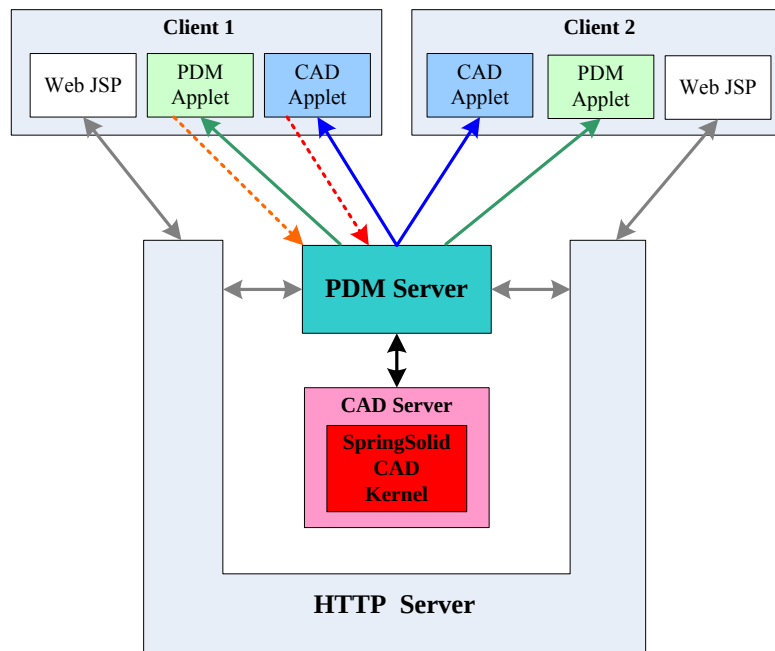


圖 4.3 PDM 伺服器扮演 CAD 代理伺服器的角色

在CAD設計過程中，若能在代理伺服器建立某些共用元件，可以減少伺服器的負載，便可增加設計效率與改善系統。

4.3.2 動態回顧

在3.1節中曾提及MTPD、NTD、NGD在多層次彈性架構中傳遞之資料格式。伺服器端與客戶端必須分別收到對方所送出的資料才得以進行下一步驟。利用此特性，置放於協同網站資料傳輸要徑上的PDM系統可以收集兩方送出的資料。而當使用者提出要在PDM系統上瀏覽協同紀錄時，便可將收集的資料按順序送出。如圖4.4所示，使用者透過PDM 客戶端，可以向PDM伺服器端要求瀏覽某次專案的協同過程，只要伺服器按照順序傳遞NGD至PDM Client，則使用者便可透過PDM Client瀏覽流程中的NGD，並且看到該專案的協同過程中建構3D模型的變化。

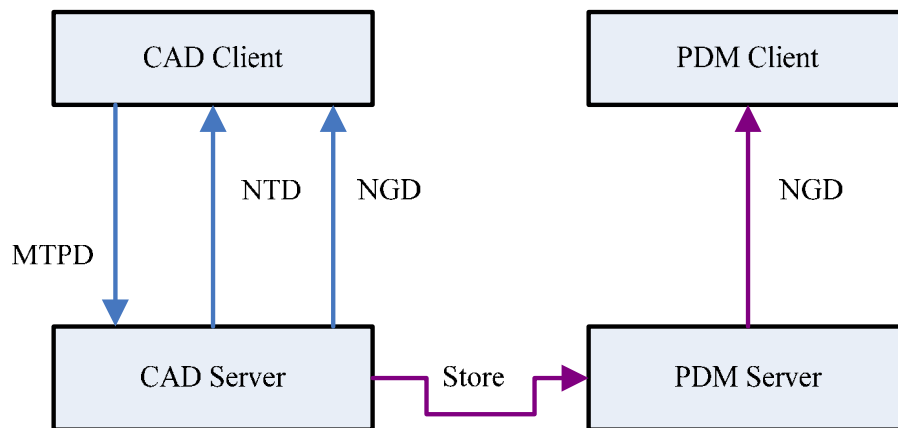


圖 4.4 使用者透過 PDM 客戶端瀏覽專案協同過程

4.3.3 非同步設計

前一節中，提到利用瀏覽NGD功能的PDM Client來做設計流程的動態回顧。如果使用者在協同專案已經完全結束後，想要進行設計成果的再修改，便可以透過CAD Client跟PDM伺服器要求先前協同專案之設計步驟資料，如圖4.5。先前協同專案所記錄的步驟可能很多，使用者的後續再設計不一定要從上次結束開始接續。

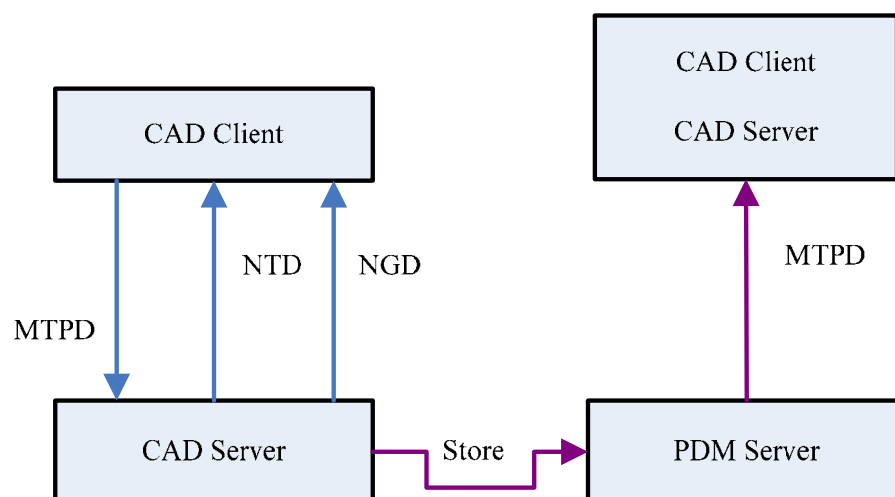


圖 4.5 從 PDM 伺服器要求協同紀錄的 MTPD 便可進行非同步設計

4.4 實作協同平台與動態協同紀錄

本研究網頁為基的設計系統上之動態歷史紀錄，用 SpringSolid 模型與 STEP 為基的 PDM 系統，以上兩者都是由台灣大學機械工程研究所之實體模型系統實驗室所開發。

4.4.1 系統架構

針對協同平台入口網站，系統包含 HTTP 伺服器 (Apache Tomcat[48])、PDM 伺服器、CAD 伺服器管理員以及網頁論壇伺服器。此外還包含儲存資料的資料庫。圖 4.6 展現協同設計平台的系統架構。

同一協同群組可以建立新的專案工作空間或者針對舊的專案進行修改。建立新專案或修改專案時，CAD 伺服器管理員會產生一個對應的 CAD 伺服器，提供實體模型的核心計算。

4.4.2 實作協同平台

藉由本文所提出的架構，建構一個網頁為基的協同設計平台，運用在 4.2 節所提出的資料型態，幫助此系統達成非同步設計的功能。而實作此系統的成果將如下所述。

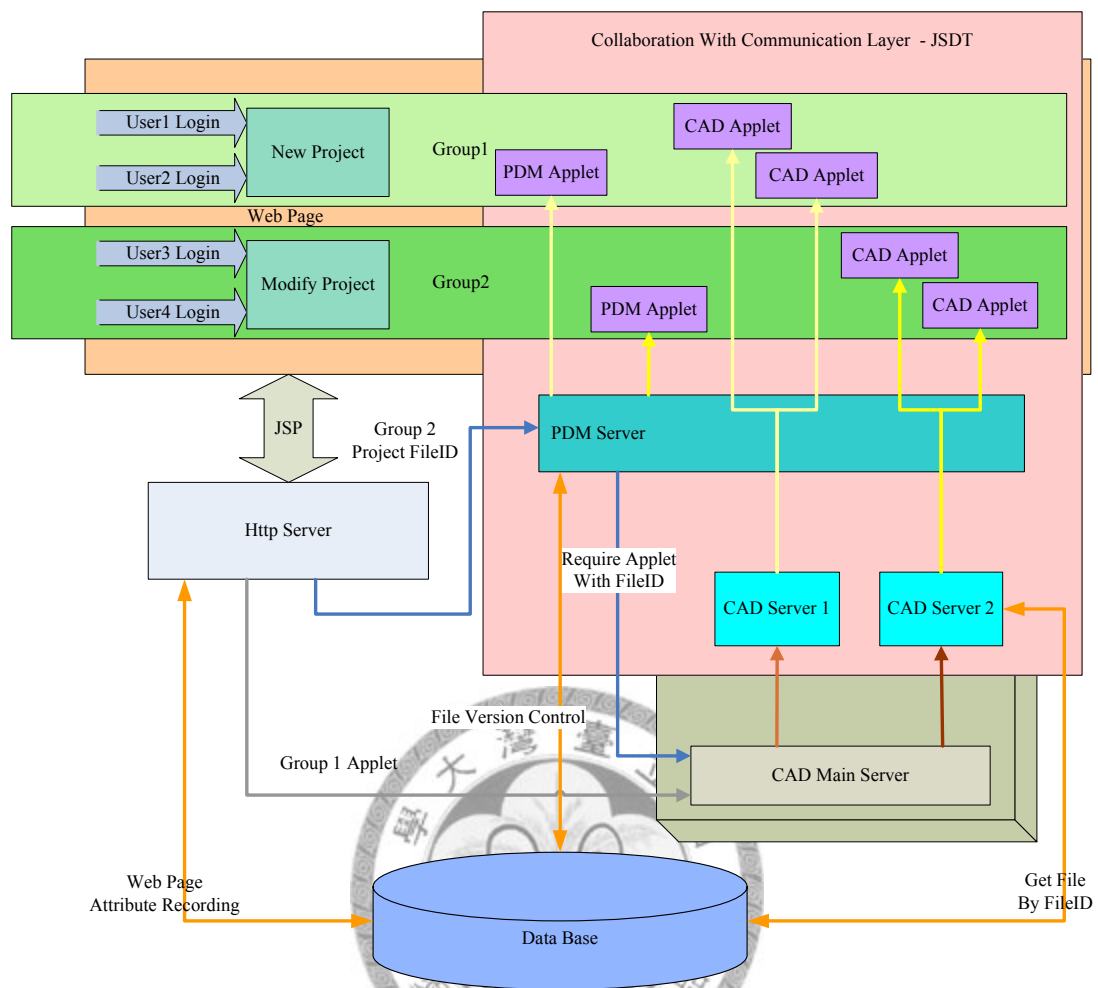


圖 4.6 協同設計平台的系統架構

在伺服器端，首先把用來建立CAD與PDM連結的Collaborative CAD PDM connection server啟動，然後再啟動PDM server 跟CAD server，如圖4.7所示。

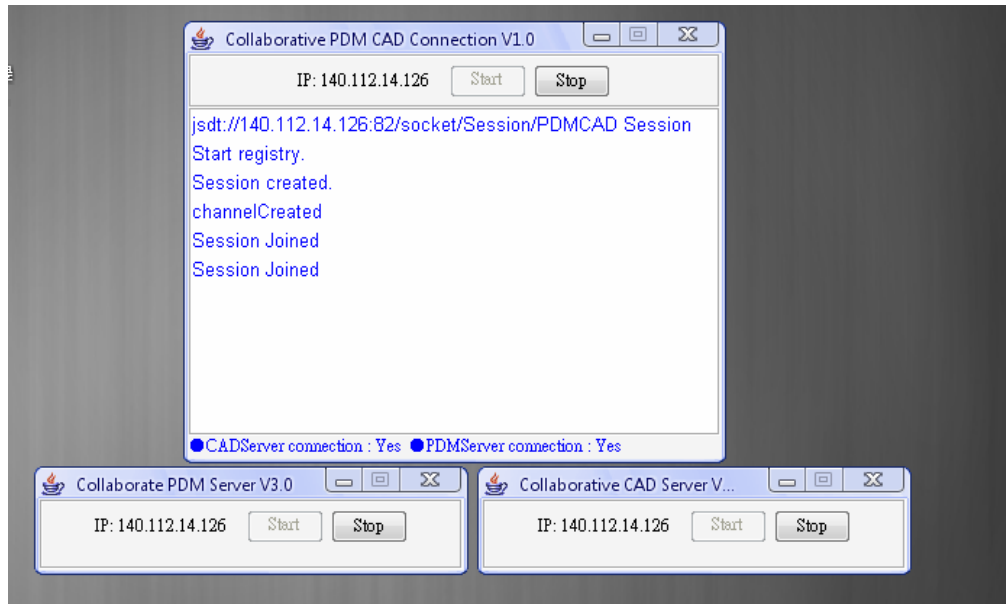


圖 4.7 架設協同平台之伺服器端

使用者再利用網頁瀏覽器連結到SpringSolid論壇之後，輸入帳號密碼即可登入，如圖4.8。

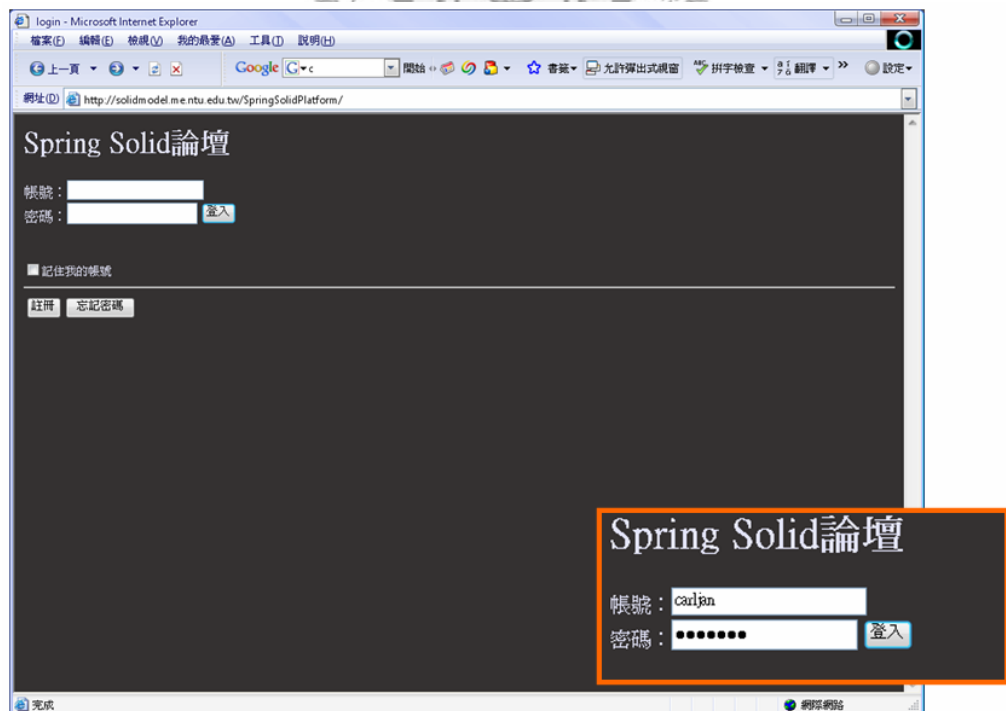


圖 4.8 使用者透過網頁瀏覽器登入網站

使用者在輸入使用者名稱與密碼後即可以進入協同平台。登入後有三個選擇，分別是：執行／參加協同專案會議、進入線上討論或是瀏覽PDM系統，如圖4.9所示。



圖 4.9 登入網站後的三種選項

點選進入專案討論區後，出現畫面如圖4.10所示，在此可以新增專案或者是修改專案。下方表格會顯示進行中的專案或是等待中即將進行的專案。此表格會定時更新，另外也提供按鈕強制更新表單。

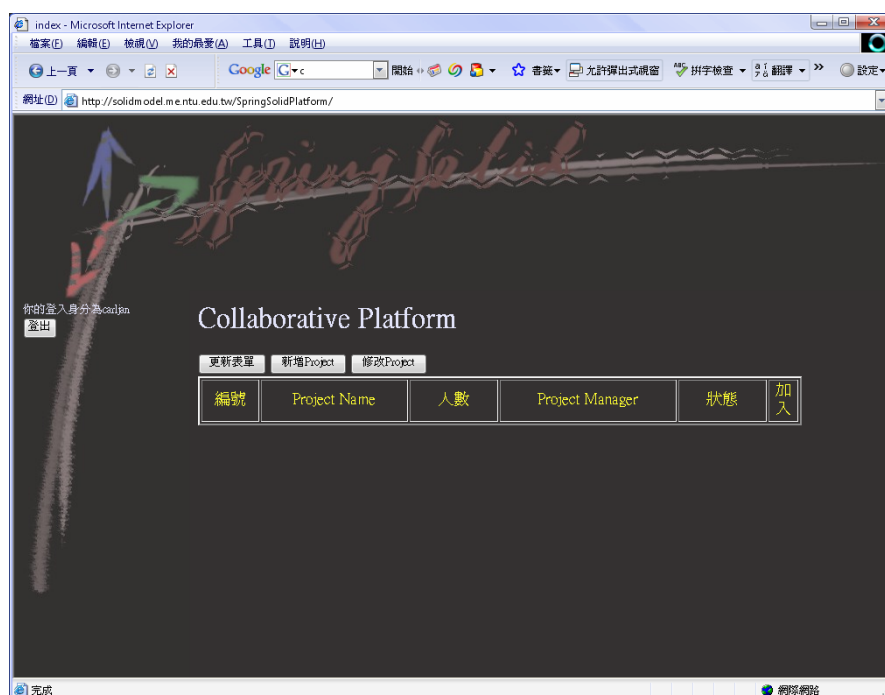


圖 4.10 專案討論區

不管選擇新增專案或是修改專案，都要選擇參與的人員，如圖 4.11所示。在選擇確定後會進入一個等待的畫面，只要等所有人員到齊，便可以開始專案。

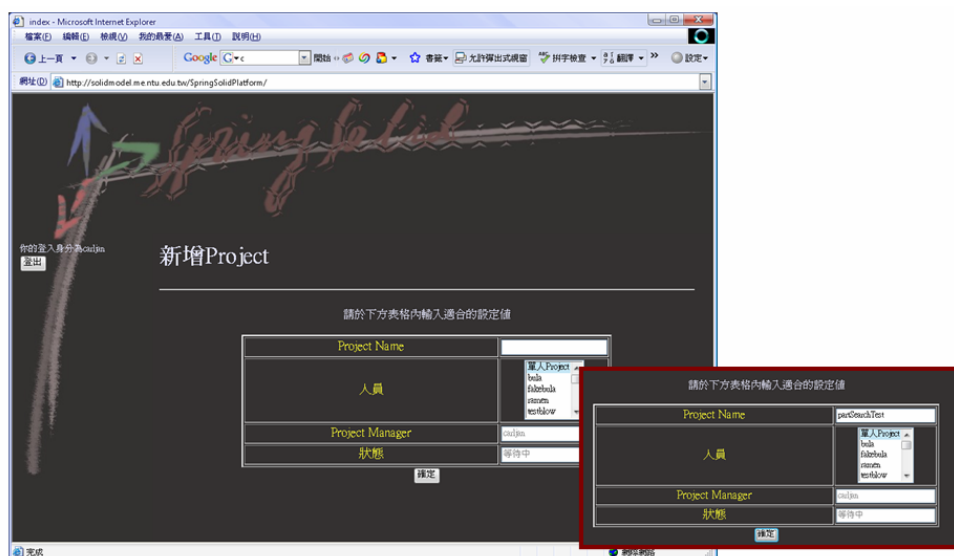


圖 4.11 選擇專案成員的畫面

當協同專案啟動，專案中每一個參與者都會被導向到CAD頁面，如圖4.12所示。CAD客戶端的Applet被嵌入在此網頁中，而另一方面CAD伺服器管理員則會常駐開啟。每當一個新的專案啟動時，CAD伺服器管理員就會自動產生一個相對應的CAD伺服器。此CAD伺服器會接收相對應的Applet傳來的傳輸資訊，並且負責其專案的拓撲與幾何計算。

介於CAD客戶端與CAD伺服器之間的PDM伺服器，藉由JDST技術記錄所有在4.2節中提及的傳輸資料。CAD客戶端、CAD伺服器以及PDM伺服器中的通訊層都是JDST的節點，每個節點可以發送與接收訊息。當CAD客戶端Applet送出訊息到CAD伺服器，PDM伺服器也可以收到一樣的訊息。在CAD伺服器計算完成後送出計算結果到CAD客戶端時，PDM伺服器也同樣記錄協同過程的每個計算結果。

在一個協同專案執行的過程中，其他使用者仍然可以透過網頁來建立其他不同主題的專案以及其專案所屬的CAD模型，CAD伺服器管理員也會自動產生另一個相對應的CAD伺服器，PDM 伺服器仍然會記錄所有使用者與CAD伺服器間傳遞的資料。

在結束協同線上設計專案後，所有的檔案都會被存放在PDM系統裡面。所有的專案成員可以讀取專案資訊，而使用者可根據其權限作專案修改或是瀏覽。

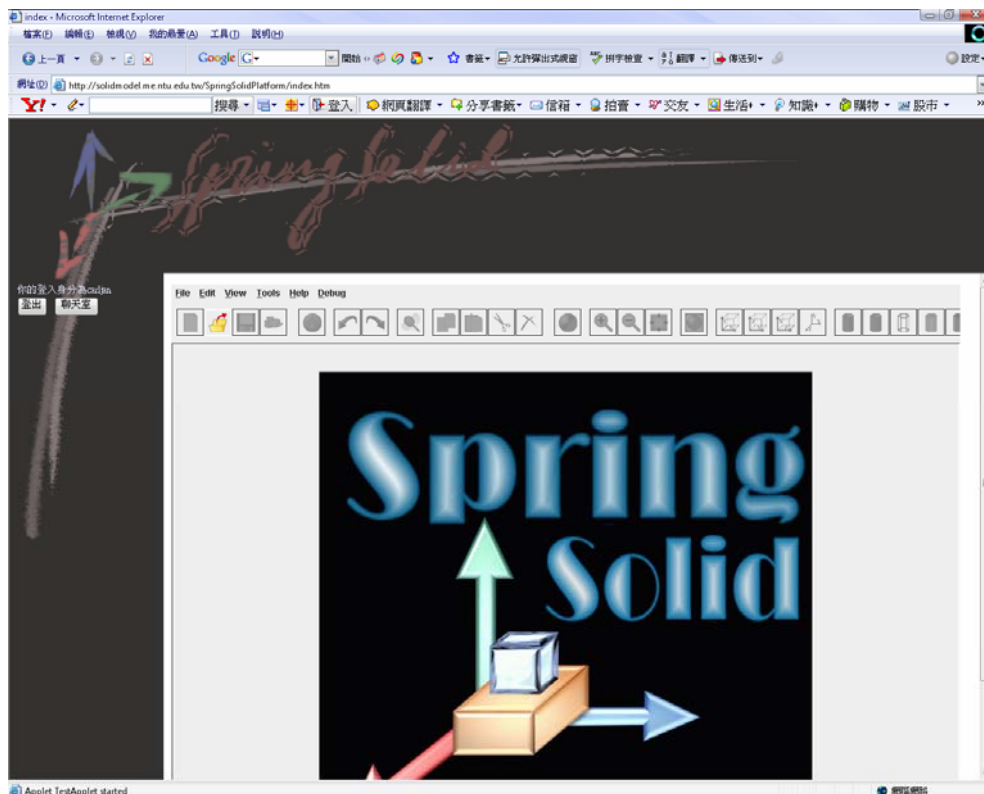


圖 4.12 協同開始，專案中的每個使用者都會進入此畫面

4.4.3 動態協同紀錄

本文所提出的PDM系統記錄產品版本的所有資訊，包含檔案名稱、檔案型態、檔案尺寸、3D模型外型跟操作步驟等，所有的操作步驟都會顯示在一個分頁-歷史紀錄，如圖4.13。

此歷史紀錄詳細列出每一操作步驟，並註明其來源。操作步驟的來源分類為兩種－網頁輸入跟計算輸出。網頁輸入包含所有客戶端Applet所做的步驟，包含開啟檔案，對2D草圖做凸塊或是增加一個布林運算等，即MTPD的集合。計算輸出則包含所有從CAD伺服器端傳送的計算結果，像是三角形的網格切割，建構歷史樹的節點等，即NTD與NGD的集合。

當協同終止時，PDM可以提供使用者在設計過程中每個步驟的詳細紀錄。使用者可選擇網頁瀏覽模式或是再計算模式。當選擇網頁瀏覽模式時，所有的計算結果NTD與NGD會送到使用者端並且在網頁上展開；當選則再計算模式時，只有網頁輸入的MTPD指令會被送到使用者端，而使用者端必須有計算核心的支援，才能夠將MTPD經過計算轉換成NTD及NGD，並顯示出來。前者比較節省CPU的計算時間，卻需要使用網路傳輸大量資料。後者有效減少網路頻寬的要求，但是對於客戶端電腦的計算能力要有一定程度的要求。CAD系統透過再計算模式將操作步驟重現有助於非同步協同的執行。

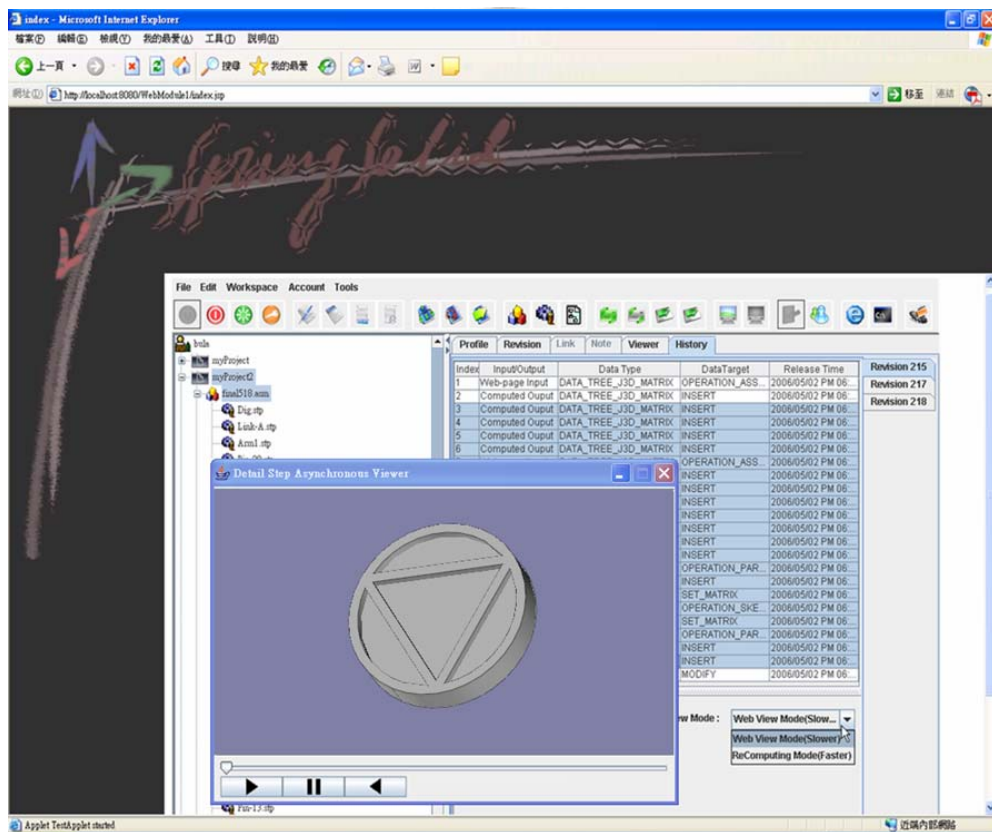


圖 4.13 PDM 系統之檔案操作歷史頁面

4.4.4 重現協同

協同設計的流程圖如圖4.14所示。

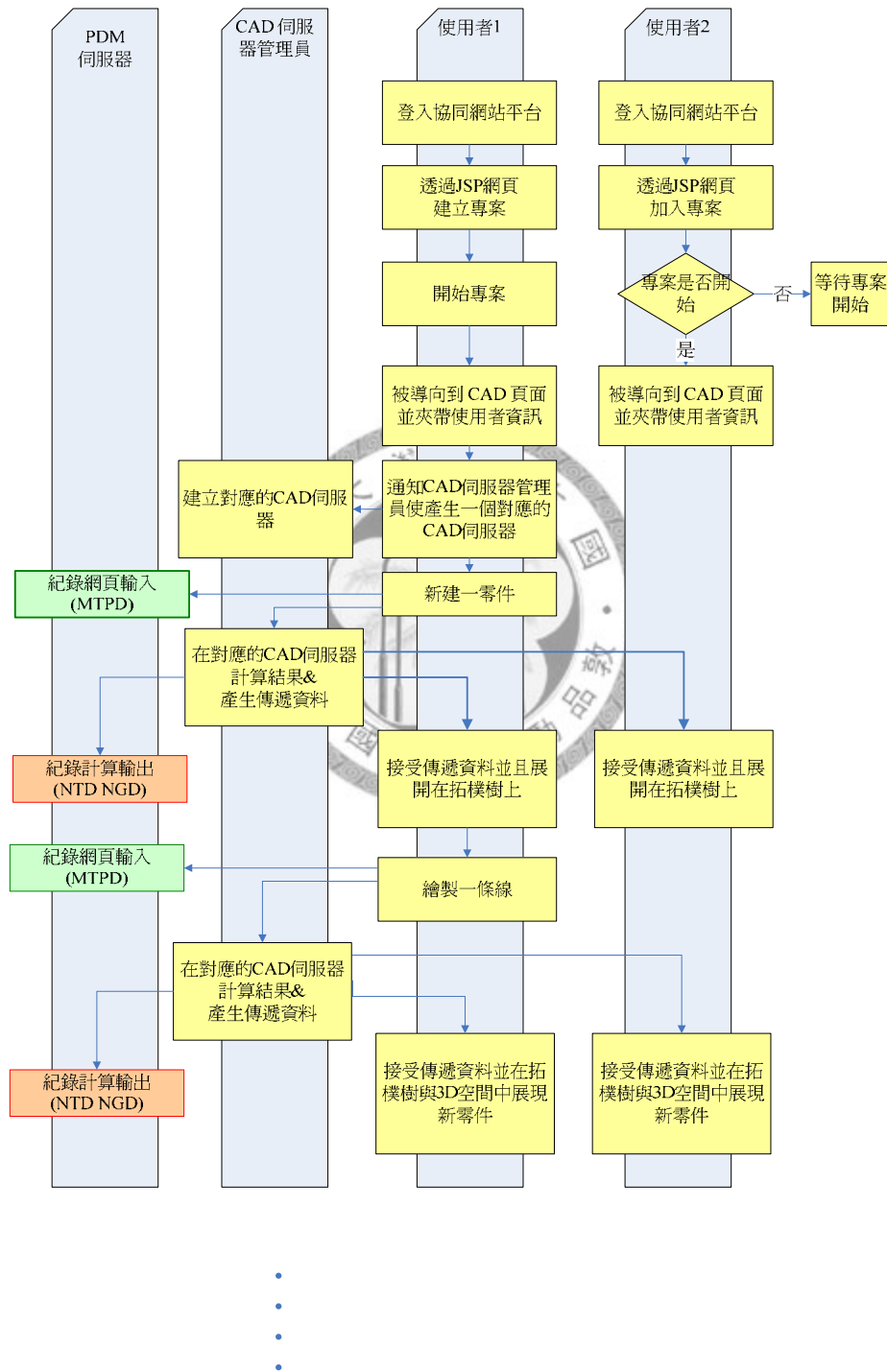


圖 4.14 兩個使用者協同設計的流程圖

使用者1與2登入此網站後，有權限的使用者1可以新增專案，使用者2則可以加入專案。有權限的使用者1可以決定何時專案開始。當專案開始時，使用者1與2的網頁都會被導向CAD頁面，如圖4.12。使用者的個人資訊，如登入名稱、群組名稱、專案名稱等都會被JSP頁面一併傳送。CAD頁面的顯示代表背後的CAD伺服器管理員接收到此訊息，並且自動在背後產生對應的CAD伺服器。

使用者1和2可以在相同的視角下協同並繪製一些簡單的CAD模型，譬如畫2D草圖，拉昇、選擇特定邊以指定導角。在客戶端所有的操作步驟會被記錄成MTPD的型態，並且在PDM系統內記錄，貼上網頁輸入的標籤，而在伺服器端的步驟則會被記錄成NGD以及NTD被在PDM系統內記錄歸類為計算輸出。

PDM系統紀錄所有的細節。有權限的使用者可以登入PDM系統並查看協同紀錄。一個簡單的步驟檢視器可以檢視幾個不同的步驟，如圖4.13所示。圖4.15則顯示使用者回顧共同設計紀錄時的流程。使用者3登入網站後，可透過JSP網頁進入PDM系統客戶端，在選擇專案後，可指定版本及要執行的步驟進行逐步瀏覽。當使用者3採用網頁瀏覽模式，則PDM伺服器會在收到使用者3要求後，依序送出當初專案進行時所記錄的NTD與NGD，使用者3便可看到每個步驟新增的圖形與建構歷史。使用者4同樣在登入平台後，透過JSP網頁進入PDM系統客戶端，選擇專案後，指定版本及要執行的步驟進行逐步瀏覽。由於使用者4的使用者端有CAD核心運算支援，所以採用再計算模式，則PDM伺服器會在收到使用者4的要求後，依序送出MTPD到使用者4端作計算，經過核心計算，便可在使用者4端依序看到每個步驟新增的圖形與建構歷史。

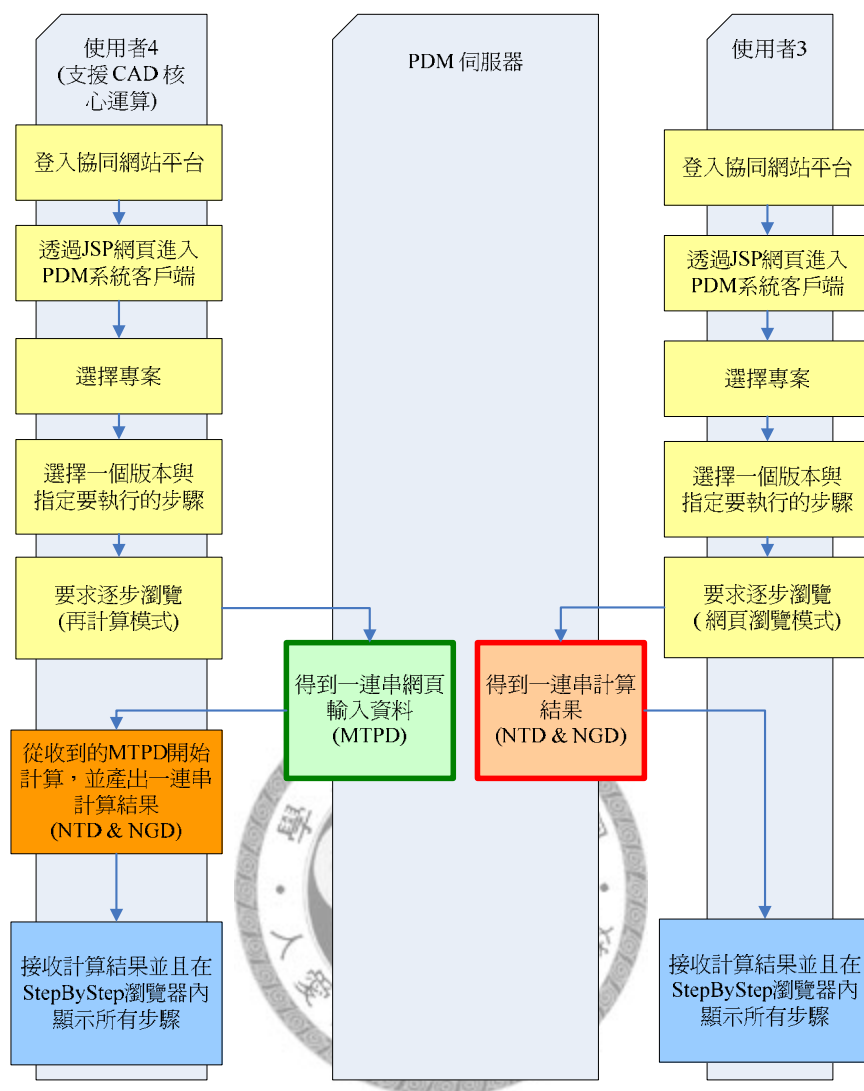


圖 4.15 使用者非同步檢視協同紀錄之流程

第五章 動態協同紀錄於設計變更傳遞之運用

本章根據巨集型態之傳遞資料(MTPD)，提出設變傳遞的流程，如圖5.1所示。工業設計團隊使用某些CAD/CAID軟體來繪製工業設計(ID)，產生第一版本的工業設計(ID1)，之後再將此模型傳遞至機構設計團隊。以ID1為基礎，機構設計團隊使用SpringSolidCAD來設計機構(ID1M1)。此時當工業設計團隊欲改變ID之局部設計而產生第二版本工業設計(ID2)，ID1M1將無法繼續設計。本章所提出之法則藉由複製ID1M1之MTPD項目(MTPDS)，並將之應用於ID2上，以便產生ID2M1，據此針對不同CAD系統間傳遞檔案時設計變更之問題，提出解決方案。

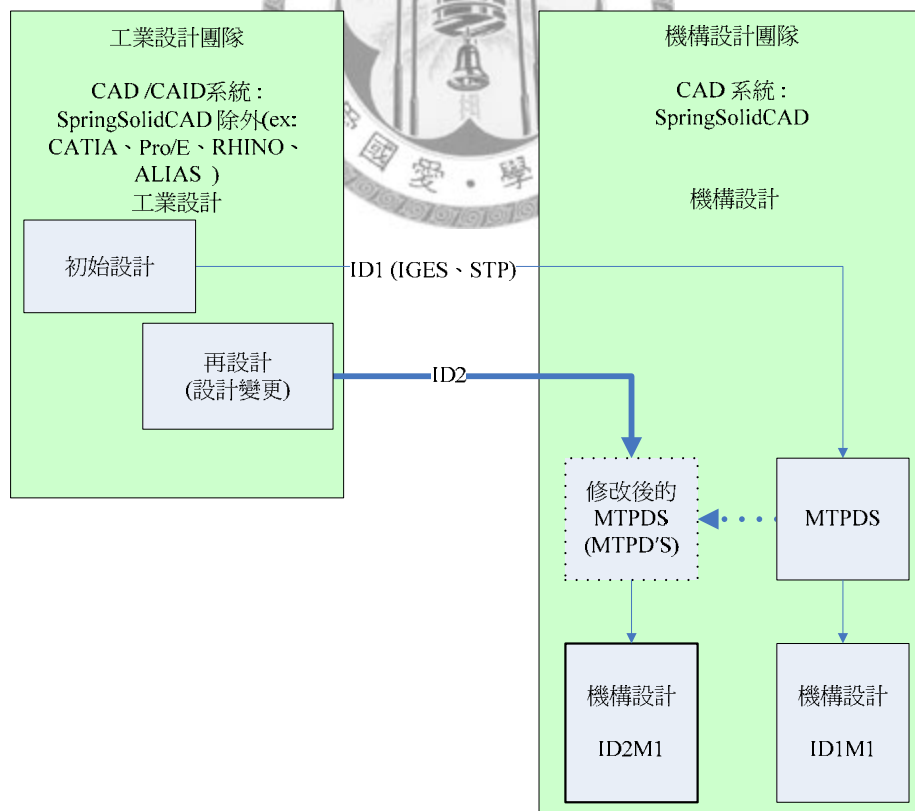


圖 5.1 簡單流程圖說明設計變更之法則

圖5.2則為詳細的流程圖。當ID2產生時，先針對原先設計到一半的ID1M1進行設計步驟的分析，釋出MTPD的集合（MTPDS）。並在ID1中取得MTPDS所參考的元件（如面、邊、點），再進一步比對這些元件與ID2中相對應之元件，可得到比較結果。如果一個MTPD沒有包含任何參考元件，或者其在ID1中的參考元件與ID2中的相對應元件為完全相同，則MTPD可直接被收錄至新的MTPD集合（MTPD'S）。如果MTPD包含某些參考元件，而這些在ID1的參考元件與在ID2中的元素並不相同，則會彈出視窗詢問使用者是否要更改MTPD的參考元件。只要能成功建構MTPD'S，就可以將其應用在ID2上產生ID2M1。

5.1 設計分析

首先從ID1M1中釋出MTPDS後再對其進行分析。MTPDS的每一個MTPD都符合一個設計步驟，而數個設計步驟能夠對應特定的特徵。MTPD紀錄所有操作步驟的細節內容，包含指令索引、參考元件索引及動作值...等等。藉由此分析，將MTPD使用的參考元件按照順序一一列出，並進一步針對這些參考元件進行模型相異性的比較。

5.2 模型相異性比較

一個模型包含數種元件的型態。本節針對面、邊、點以及2D草圖，描述如何在中介格式的檔案與更新後的模型中進行比較。

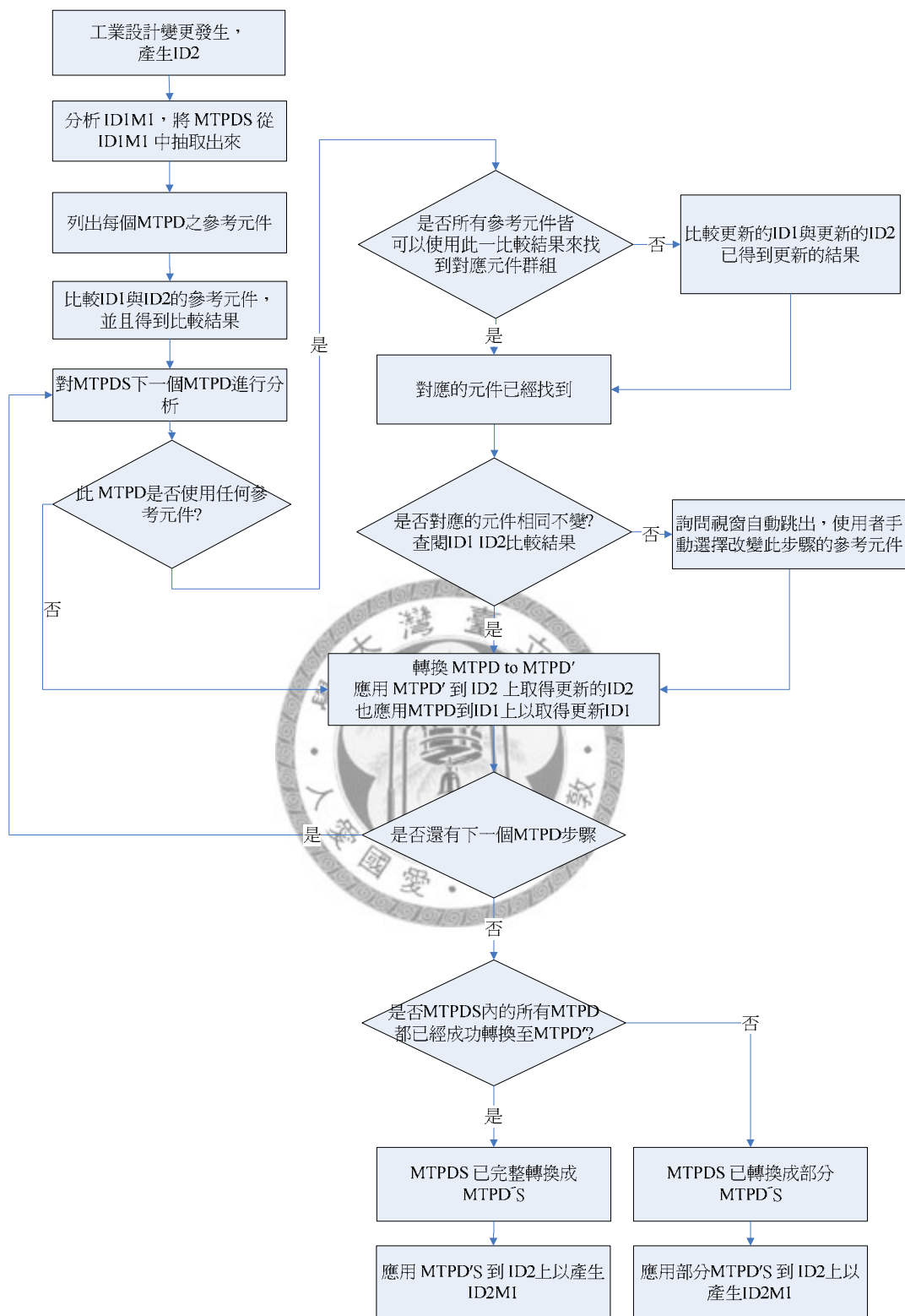


圖 5.2 產生 ID2M1 之詳細流程圖

5.2.1 中介格式檔案的面比較

本段討論使用幾何及拓樸的方法來比較ID1與ID2模型的面，進行順序如下：

(1) 將不同的面分類成下列三種型態，其中第三種型態又再細分四項，圖5.3則說明分類這些面的流程。

(a) 相同：

當ID1面的拓樸、幾何形狀參數與位置皆與ID2的某個面相同時，可將該面定義成相同面。

(b) 相嵌：

當ID1面的型態、與ID2的某個面相同，兩者之間有部份是完全重合，可將該面定義成相嵌面。

(c) 相異：

當ID1面無法符合上述兩種型態之檢查，即定義該面為相異面。

由於相異面無法找到在ID2上之對應面，因此需要再搭配拓樸圖比較將相異面找到ID2之對應面。對應面找到後，便可比較兩對應面的型態，並分類成下面四種：

(c1) 不相關的面

(c2) 不同的幾何型態

(c3) 相同幾何型態，相同形狀

(c4) 相同幾何型態，不同形狀

(2) 輸入ID1的參考面，搜尋在ID2的對應面，並且將其分類至適當層級。

使用幾何比較可以將ID1的面分類成相同、相嵌或是相異的面。

當ID2有許多改變的面與ID1不相同時，會有較多的相異面。幾何比較僅能夠找到ID1與ID2之間的相同面與相嵌面之對應關係，而無法找出ID1與ID2之間的相異面對應關係，因此採用拓樸圖形比較法則來進行補強，拓樸圖形比較法則必須根據幾何比較之後的結果，再進一步將相異面的對應關係找出。在拓樸圖形比較法則中，ID1中的三個參考面被用來當作輸入面，其中兩個面已經在ID2中有對應面，而第三個面則是用幾何比較在ID2中無法找出對應面的面，藉此方法，可以找到第三個面在ID2的對應面。這兩個法則將會在下面詳述。

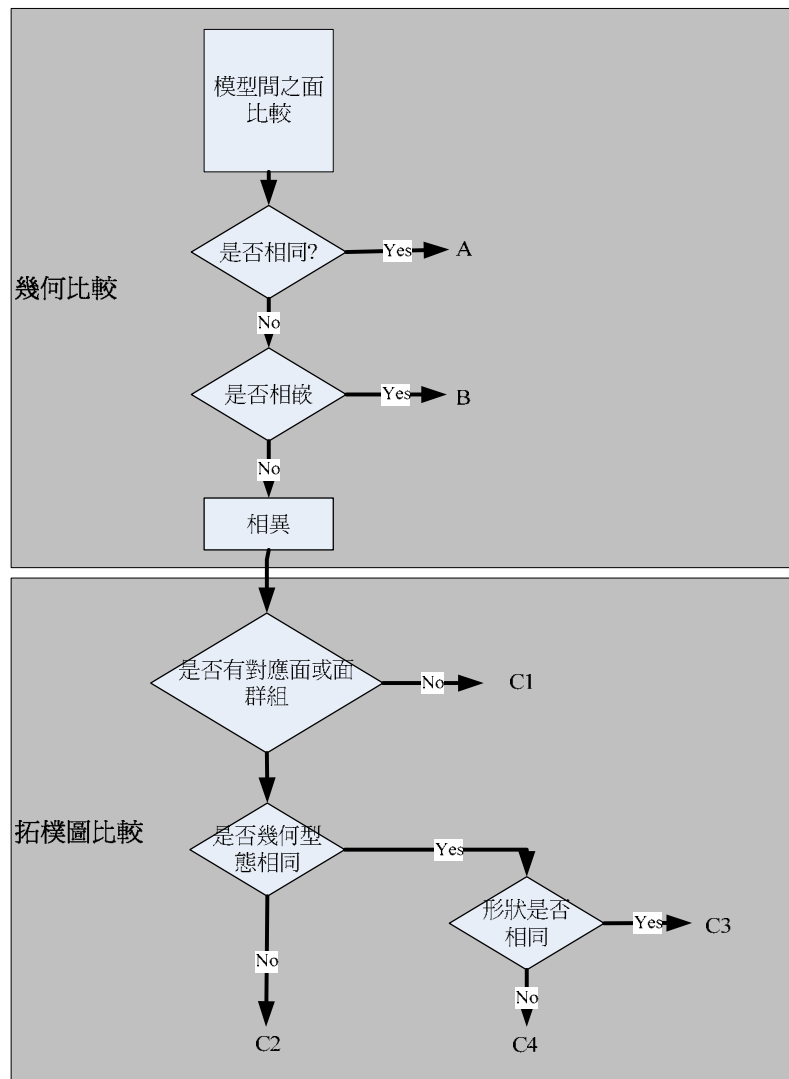


圖 5.3 參考面分類之流程圖

5.2.1.1 幾何比較

利用貼標籤的方式，將所有的面進行比對後貼上標籤，其虛擬程式碼如圖 5.4 所示，ID1 與 ID2 上所有的面會先透過 CompareIsSame 來比較，當兩個面相同，則透過 SetFaceAttributeSame 將這兩個面的性質設為相同，並且貼上定義為相同的標籤。當全部都比對完，則透過 CompareIsEmbedded 進行相嵌面比較，當兩個面符合相嵌條件，則會透過 SetFaceAttributeEmbedded 將兩個面的性質設為相嵌，並貼上定義為相嵌的標籤。在做完相同面與相嵌面的搜尋後，剩下的面都貼上定義為相異。

GeometryComparison(faceID1, faceID2)

```
m = faceID1.length
n = faceID2.length

for i = 0 to m
    for j = 0 to n
        if faceID2[j].GetLabel() = NONE then
            isSame = CompareIsSame( faceID1[i], faceID2[j] )
            if isSame = True
                SetFaceAttributeSame( faceID1[i], faceID2[j] )

for i = 0 to m
    for j = 0 to n
        if faceID2[j].GetLabel() = NONE then
            isEmbedded = CompareIsEmbedded( faceID1[i], faceID2[j] )
            if isEmbedded = True then
                SetFaceAttributeEmbedded( faceID1[i], faceID2[j] )

for i = 0 to m
    if faceID1[j].GetLabel() = NONE then
        faceID1[i].SetLabel( DIFFERENT )
```

圖 5.4 幾何比較之虛擬程式碼

搜尋對應面的第一階段是做相同面的比較。首先比較原始面與目標模型上的其他面之端點數量，如果端點數量在某些面上相同，再比較該面上的端點的座標，接著比較面上的所有邊的型態，如果型態相同則再進一步比較邊的參數，在參數都相同的情況下，可以認定這些面為相同。圖5.5描述決定相同面的流程。

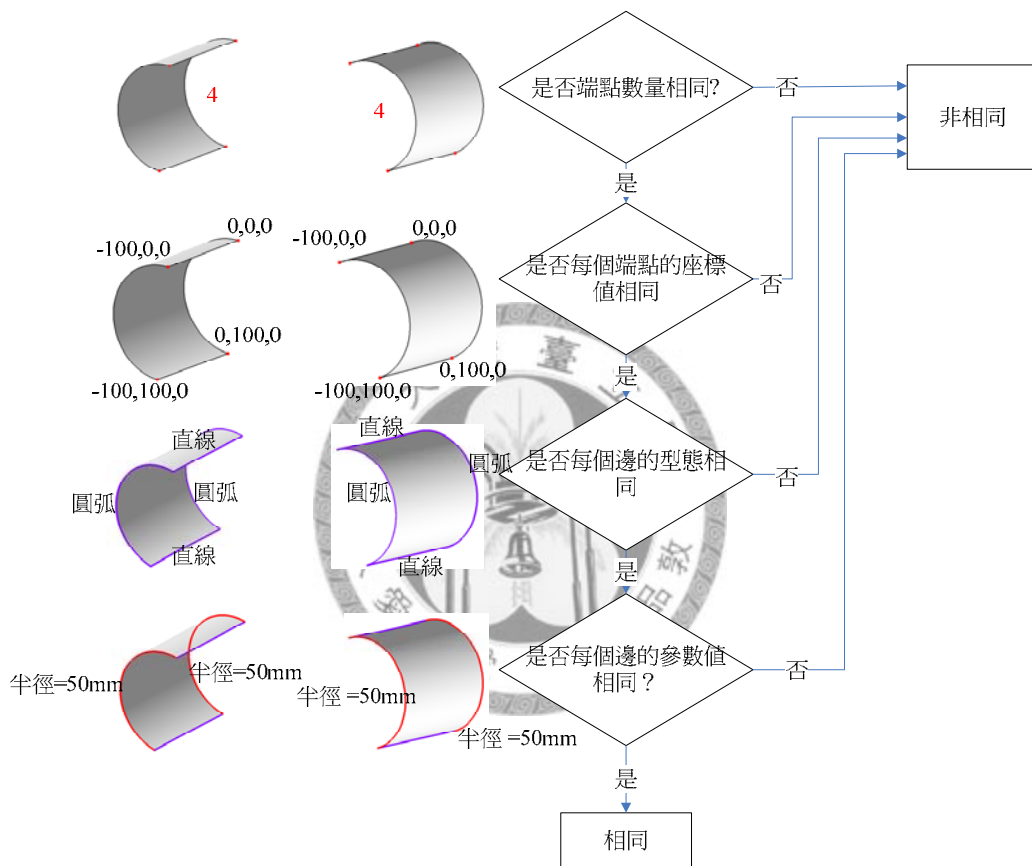


圖 5.5 決定相同面的流程

如果在目標模型上找不到相同面，下一步就是要判斷是否有對應相嵌的面。圖5.6說明辨識對應相嵌面的流程。在立體直角座標系統內，面的包圍盒（Bounding Box）為一有長寬高之立體空間，其長寬高皆大於面的長寬高，因此可將面包圍起來。用此來檢查是否面有重疊，如果包圍盒有重疊則將原始面上所有端點複製到目標面

上，並判斷這些端點是否在其參數範圍中。如果有，則此兩面可判讀為相嵌。使用此方法也可以找到完全相同的面，所以在使用此法則之前，通常先用上述的相同面決定法則來比對兩模型，將相同面先剔除。

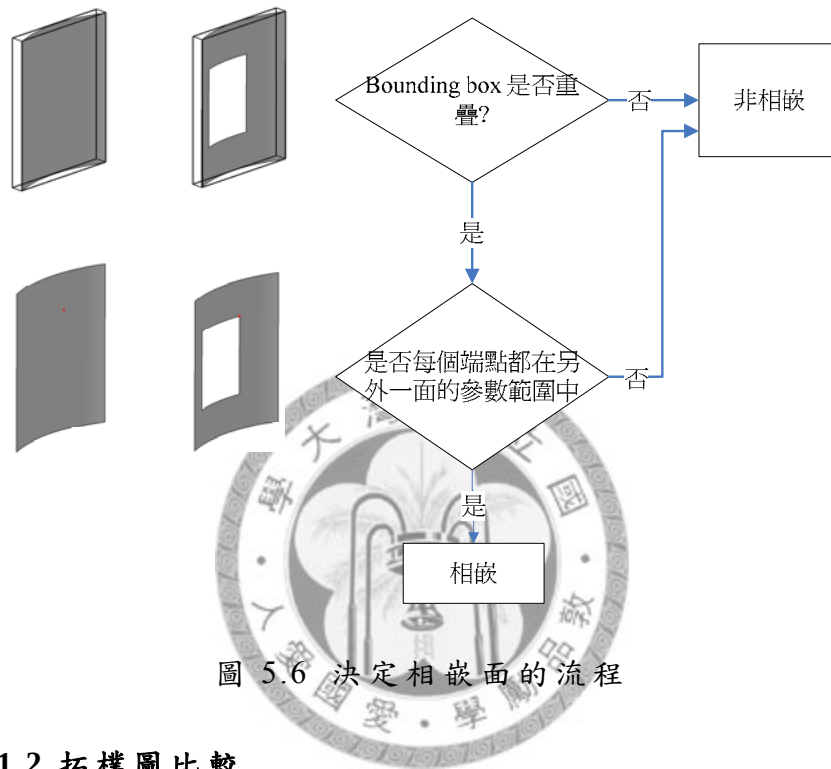


圖 5.6 決定相嵌面的流程

5.2.1.2 拓撲圖比較

幾何比較可以在ID2找到一些對應的面，但當幾何比較不能辨識對應的面時，本研究提出拓撲法則來進行補強。利用El-Mehalawi[44-45]所提出的法則，可以將模型轉換成拓撲圖，再針對拓撲圖進行比較。系統在讀取ID1之後得到實體模型以及其面矩陣（face_array），進一步再獲得每個面及面迴路（face loop）裡的定向邊矩陣（oriented_edge_array）。每一個定向邊都可以找到其對應的邊及其型態，而邊可以在找到另一個定向邊，進而找到另外一個面迴路與相鄰面，利用此特性，可以取得模型中所有面與邊的關係。圖5.7展示在實體模型結構中如何找到面與邊關係。一個STEP檔案的

面與邊關係可以被整合成一個數學矩陣，兩個面中若無夾邊則以數字0表示，若有夾邊則以數字1-7表示，不同的數字代表不同的夾邊型態，圖5.8則顯示模型中的面與邊的關係如何以面邊關係矩陣來描述。

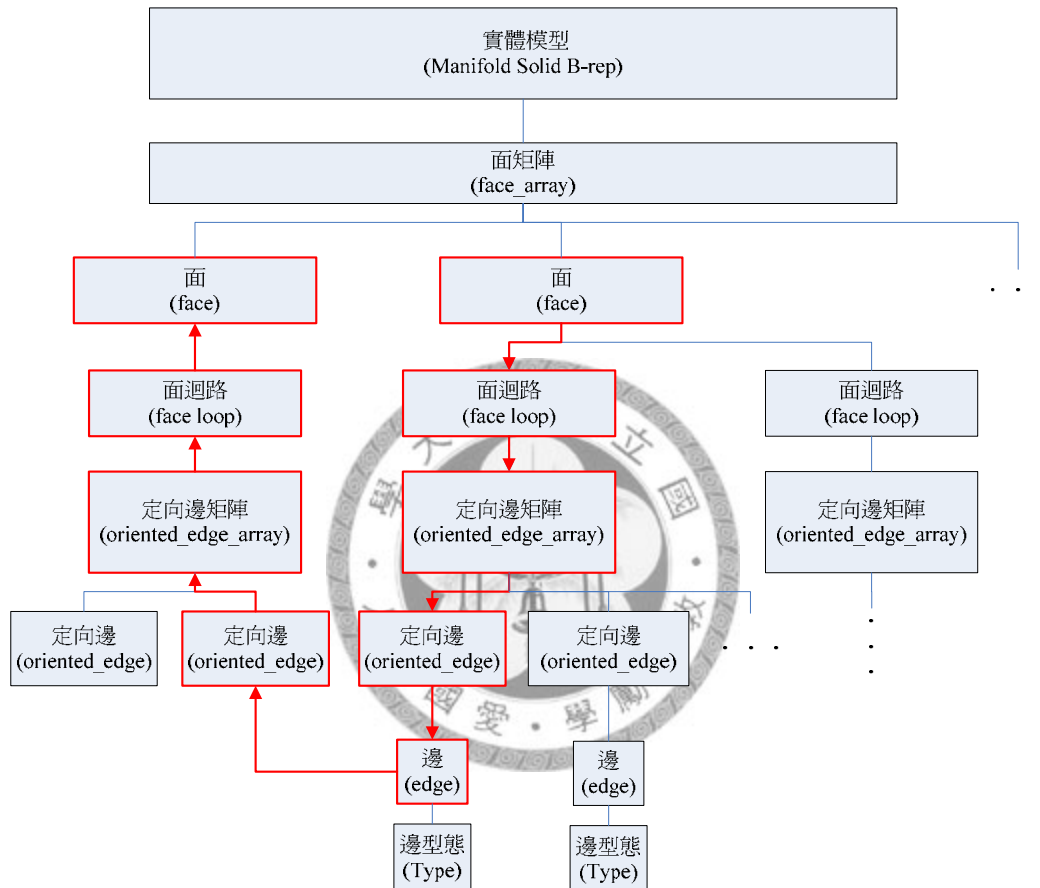
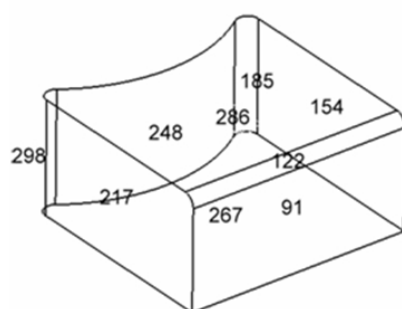


圖 5.7 從實體模型之結構中取得面與邊關係

faces	91	122	154	185	217	248	267	286	298
91	0	1	1	0	1	0	1	0	0
122	1	0	2	0	2	0	0	1	0
154	1	2	0	1	0	0	1	1	0
185	0	0	1	0	0	1	2	2	0
217	1	2	0	0	0	0	1	1	1
248	0	0	0	1	0	0	2	2	1
267	1	0	1	2	1	2	0	0	2
286	0	1	1	2	1	2	0	0	2
298	0	0	0	0	1	1	2	2	0



夾邊類型	代號
LINE	1
CIRCLE	2
ELLIPSE	3
PARABOLA	4
HYPERBOLA	5
BEZIER CURVE	6
B-SPLINE CURVE	7

圖 5.8 模型的 3D 圖形及其面邊關係矩陣

使用幾何比較過後，在ID1與ID2當中相同與相嵌的面皆可以找到，剩下的相異面則無法在另一模型中找到對應面。若按照有無對應面，可以將所有面分成兩個集合，集合A包含在另外的模型中有對應的面，而剩下的面則構成另一個集合B。在圖5.8中，系統在B1集合中找出FaceB1_1，其有兩個相鄰面在FaceA1_1、FaceA1_2在A1集合。利用這兩個相鄰面在A2集合中找尋到對應面FaceA2_1、FaceA2_2。利用面邊關係搭配兩個已知面FaceA2_1、FaceA2_2在B2集合中找尋目標面，而FaceB1_1的對應面則可以順利被找到。此法則可以協助系統將B1中的面在B2中找到對應面。

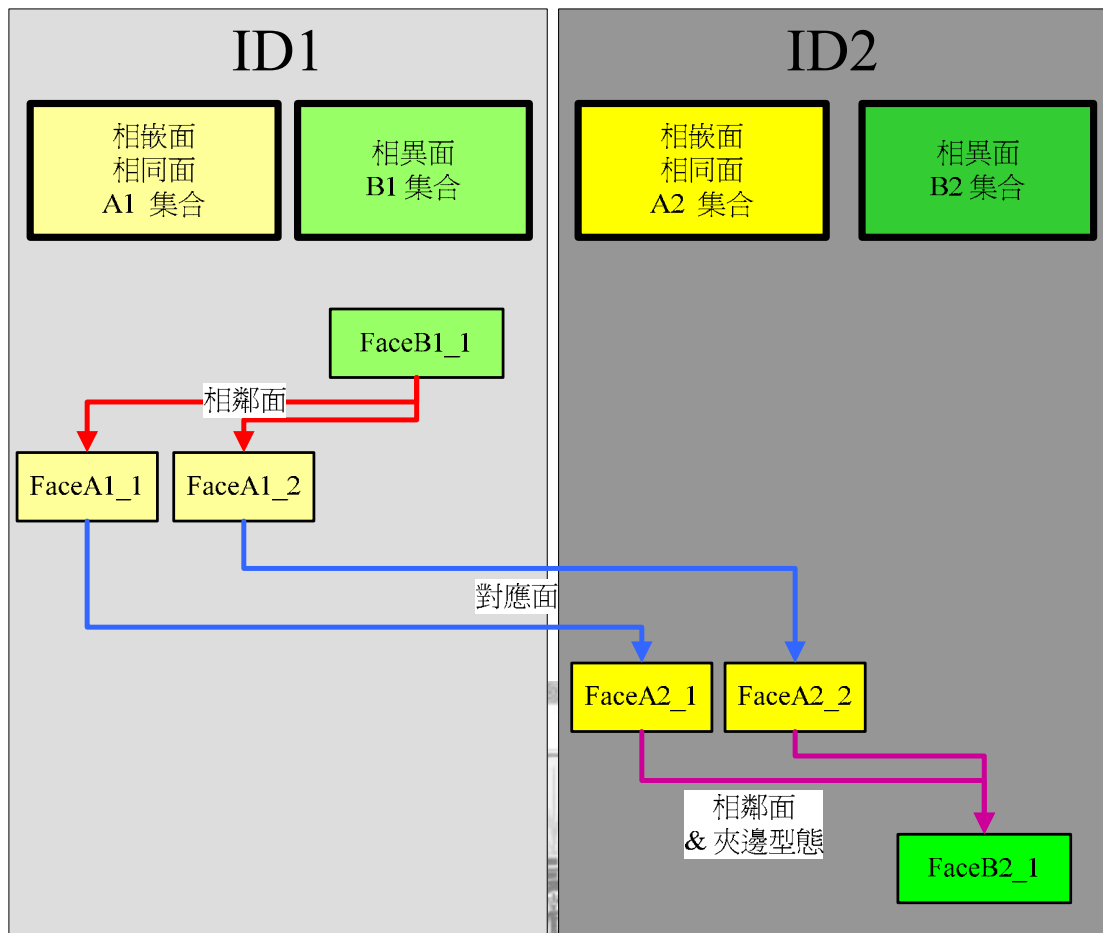


圖 5.9 利用面邊關係及已知面來尋找未知對應面

對應面關係建立之後，在B2集合的面可以被分為四個群組，如圖5.3下方之拓撲圖比較所示。

虛擬程式碼如圖5.10所示，先分別將兩個ID產生面邊關係，並在ID1尋找相異面，利用FindAdjacentFacesWithCorresponding找有對應面的相鄰面。如果找到相鄰面，則會先取得與輸入之相異面之夾邊型態，再利用FindCorrespondingFace，輸入相鄰面之對應面與夾邊型態，搭配ID2的面邊關係，推測出輸入相異面之對應面。如果找不到相鄰面，則會跳過此輸入相異面，先進行下一個相異面的對應面搜尋。ID1當中的相異面全部搜尋完一遍後，部分的相異面會被設

定為有對應面，接著會再從頭開始搜尋，ID1當中之相異面但已有對應面者，再搜尋時會被排除。搜尋的動作將會一直重複，直到搜尋找不到相異面之對應面。

```

TopologyComparison(faceID1, faceID2)
ID1Relation = SetFaceEdgeRelation(faceID1)
ID2Relation = SetFaceEdgeRelation(faceID2)
foundCount = 1
m = faceID1.length
n = faceID2.length

while foundCount != 0
    foundCount = 0
    for i to m
        if faceID1[i].GetLabel() = DIFFERENT & faceID1[i].GetCorrespondingFaceID != -1 then
            IDB1_1 = faceID1[i].IDNumber()
            faceAdjacents = FindAdjacentFacesWithCorresponding(faceID1[i])
            if faceAdjacents != null then
                IDA1_1 = faceAdjacents[0].IDNumber()
                IDA2_1 = faceAdjacents[0].GetCorrespondingFaceID()
                IDA1_2 = faceAdjacents[1].IDNumber()
                IDA2_2 = faceAdjacents[1].GetCorrespondingFaceID()
                typeA1_1 = GetFaceRelation(IDB1_1, IDA1_1, ID1Relation)
                typeA1_2 = GetFaceRelation(IDB1_1, IDA1_2, ID1Relation)
                IDB2_1 = FindCorrespondingFace(IDA2_1, IDA2_2, typeA1_1, typeA1_2, ID2Relation)
                if IDB2_1 != 0 then
                    faceID1[i].SetCorrespondingFaceID(IDB2_1)
                    foundCount++

```

圖 5.10 拓樸圖比較之虛擬程式碼

5.2.2 邊緣與端點的比較

在兩個模型中若要比較邊緣和端點並找到其關連性，可以先建立一個面的對應表。系統可以從指定的邊去原始模型中搜尋兩個相鄰的面，並進一步在新模型中找到其對應面。一旦對應面在新模型中找到，便可以將指定邊與兩個對應面上的夾邊進行幾何參數的比較，如圖5.11(a)所示。但當新模型中找到的兩個對應面沒有相鄰之夾邊，則代表該邊在新模型中已經分成兩個以上的邊，此時必須將相鄰的兩個面分開搜尋對應面，並從該對應面中的邊中，找到幾何

位置、向量最接近的邊做為對應邊，如圖5.11(b)所示。

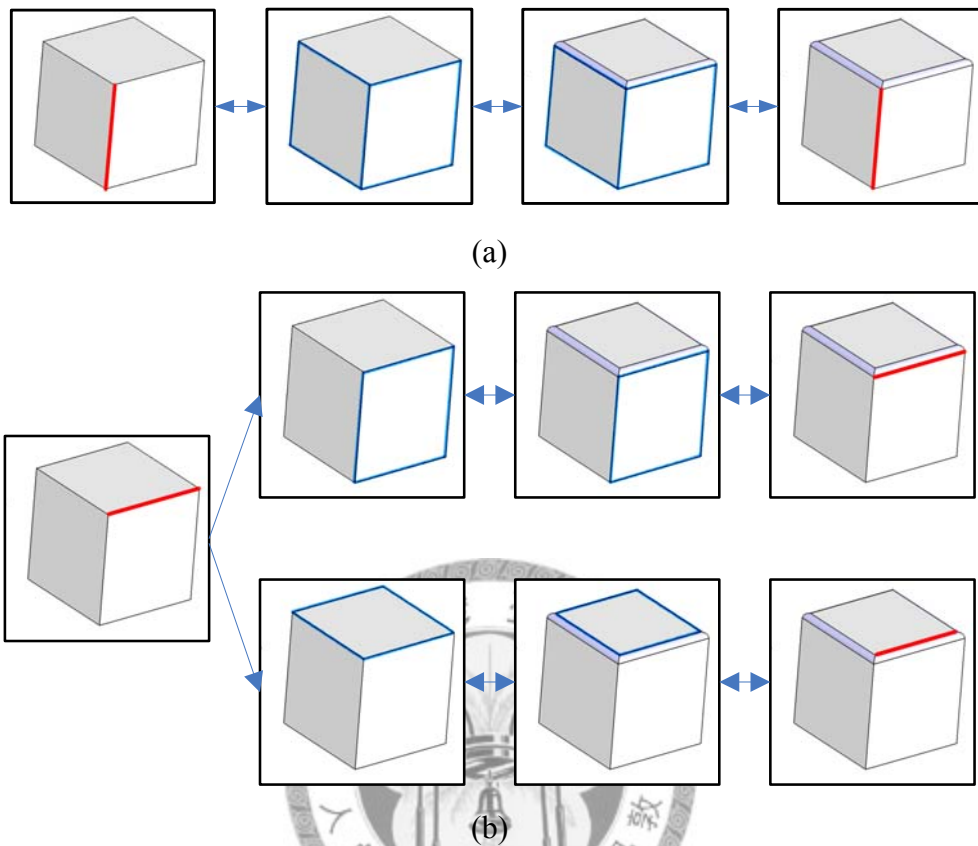


圖 5.11 邊的比較方式(a)搜尋兩對應面的夾邊 (b)搜尋單一對應面的最接近邊

搜尋的虛擬程式碼如圖5.12所示，先利用GetUsedFace找到目標搜尋邊所使用到的面，再找到這些面的對應面。得到對應面之後，先判斷是否這些對應面有共邊，若有共邊則為所求。當無法求得共邊時，則利用FindNearestEdge將目標搜尋邊與每個對應面進行比對，找到在該面上最接近的邊，而這些邊的集合即為對應邊。

FindCorrespondingEdge(EdgeInput)

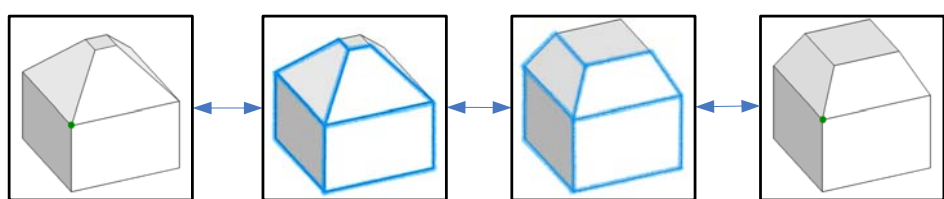
```
faceInput = GetUsedFace(EdgeInput)
faceOutput = FindCorrespondingFace(faceInput)
EdgeOutput = FindCommonEdge(faceOutput)

if EdgeOutput = null then
    m = length[faceOutput]
    for i = 0 to m
        EdgeOutput[i] = FindNearestEdge(faceOutput[i], EdgeInput)

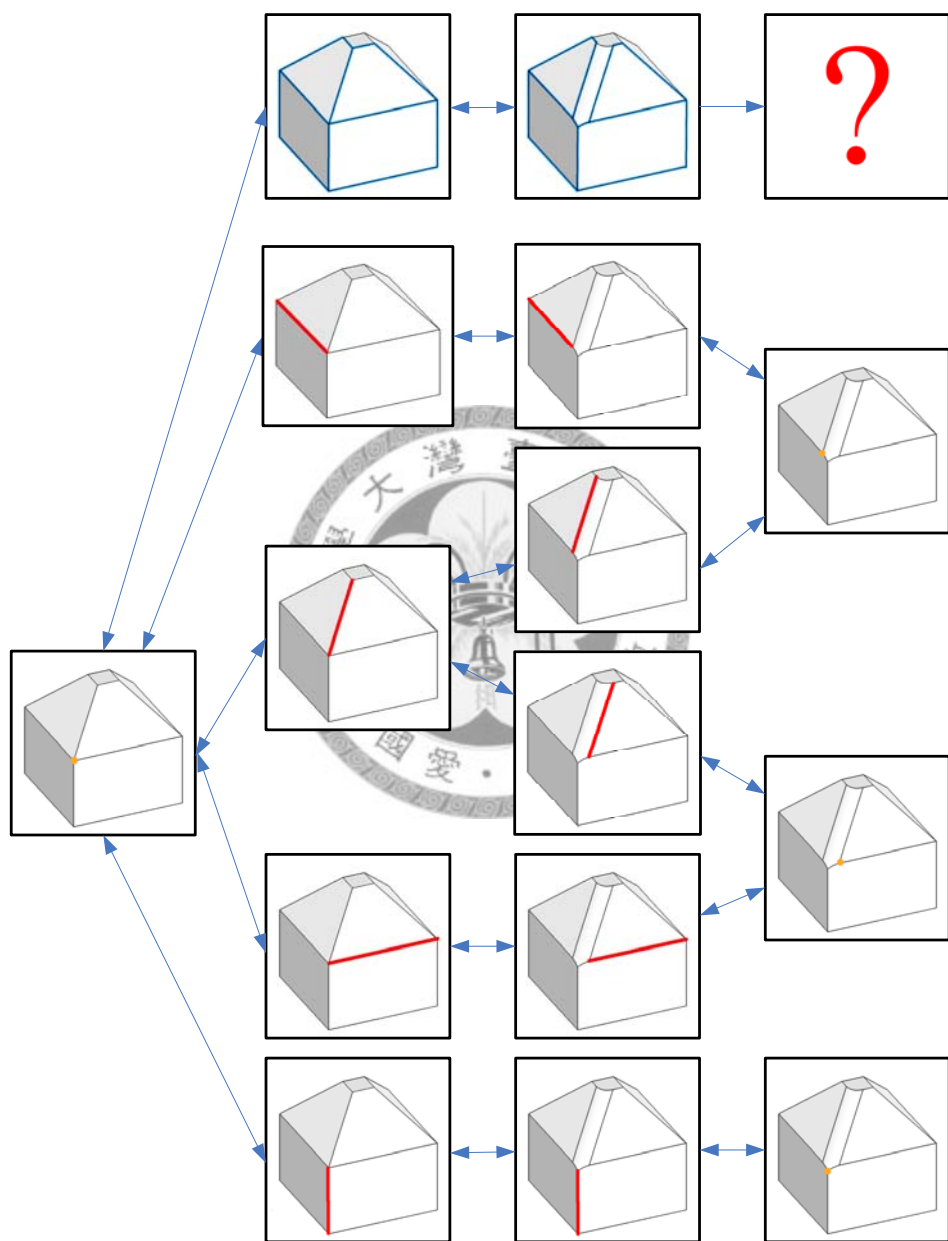
return EdgeOutput
```

圖 5.12 搜尋對應邊之虛擬程式碼

比對端點也與比對邊緣相似。可以先從端點相鄰面開始尋找，如果相鄰面的對應面一樣有同一點交點，則可以判定該交點為此端點的對應點，如圖 5.13(a) 所示。但若相鄰面的對應面無法相交於一點，則代表可能有些相鄰邊已經在新模型中分成兩個邊。則可以再從相關聯的邊開始，從關聯的邊可以找到在目標模型中對應的邊，然後再將對應邊的端點與輸入端點之幾何參數進行比對，找到幾何位置較為接近的對應端點，如圖 5.13(b) 所示。



(a)



(b)

圖 5.13 端點的比較方式(a)搜尋對應面的交點(b)搜尋單一對應邊的端點

搜尋的虛擬程式碼如圖 5.14 所示，先利用 GetUsedFace 找到目標搜尋點所使用到的面，再找到這些面的對應面。得到對應面之後，先判斷是否這些對應面有共點，若有共點則為目標搜尋點之對應點。當無共點時，則利用 GetUsedEdge 找到目標搜尋點所使用到的邊，再找到這些邊的對應邊，最後再利用 FindNearestVertex 將目標搜尋點與每個對應邊進行比對，找到在該邊上最接近的點，而這些點的集合即為對應點。

```
FindCorrespondingVertex( VertexInput )  
  
faceInput = GetUsedFace( VertexInput )  
faceOutput = FindCorrespondingFace( faceInput )  
VertexOutput = FindCommonVertex( faceOutput )  
  
if VertexOutput = null then  
    edgeInput = GetUsedEdge( VertexInput )  
    edgeOutput = FindCorrespondingEdge( edgeInput )  
    m = length[ edgeInput ]  
    for i = 0 to m  
        VertexOutput [i] = FindNearestVertex( edgeOutput[i], VertexInput )  
  
return VertexOutput
```

圖 5.14 搜尋對應點之虛擬程式碼

5.2.3 2D圖形的比較

2D草圖由於是SpringSolidCAD的內部元件，通常是MTPD應用在ID模型時產生，也只會更新的模型上找到。因為是透過MTPD在兩邊的模型上建立，所以兩邊的模型一定會有對應關係。2D草圖是平面的，也很容易比較。先比較草圖上點與曲線的數量，如果數量相同，系統會再比對每個點的座標。

5.3 設計步驟複製

參考元件的相異性經過比較之後，根據MTPDS的順序，便開始複製步驟。如果MTPD只是指定模組間的切換，沒有包含任何參考元件，會被直接轉換成MTPD'。當MTPD包含的參考元件在ID1與ID2中相同也會直接被轉換成MTPD'。使用者只有當參考元件在ID1與ID2是不相同的情況下，才需要半自動的進行調整MTPD。

藉由更改MTPD中的紀錄，將ID1中的參考元件變成ID2中對應的元件，MTPD便可轉換成MTPD'。經過比對之後，系統應可提供要更改的對應元件。如果MTPD中的參考元件不能在原始的ID1中找到，則此參考元件應該是在前面的某些MTPD執行之後才被創造出來。在這個情況下，系統列出的對應元件已經不夠參考。

MTPDS中的每一個步驟在複製時，利用上節所提到模型相異性比較，將步驟中所使用到ID1的參考元件，轉換成在ID2的對應元件。但是若步驟使用到的參考元件是在某些步驟之後才形成，便無法在ID1中找到。如圖5.15所示，在第四步驟結合面長出厚度後，由於第五步驟的繪製草圖必須參考第四步驟所新生長出來的面，因此第五步驟所使用到的參考元件，無法在初始的ID1當中找到，也就無法複製此步驟。

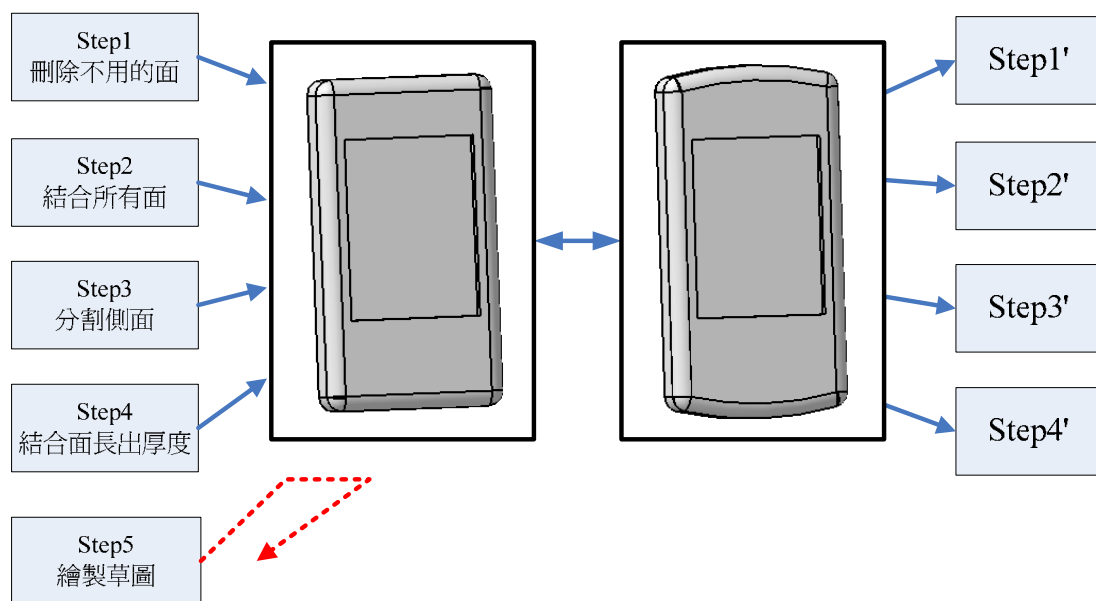


圖 5.15 處理某些步驟時須參考更新之 ID1

此時若能夠先將第一步驟到第四步驟先套用在初始的ID1上，則第五步驟的參考面可以在更新過後的ID1上找到，而更新過後的ID1與ID2也可以再次藉由模型相異性比較，得到對應關係，順利將步驟複製，如圖5.16所示。

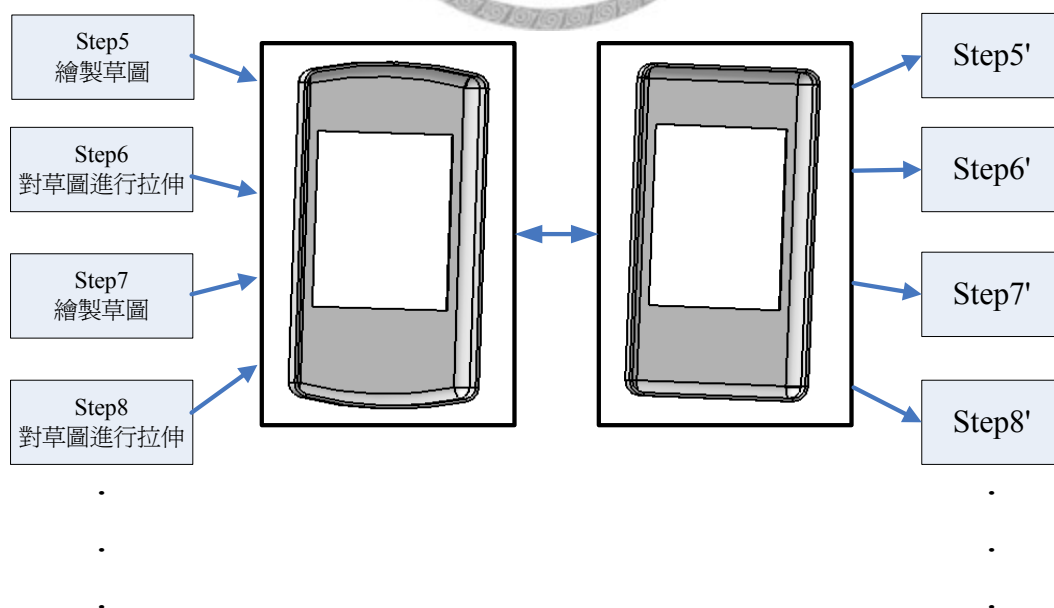


圖 5.16 後續步驟複製必須參考更新後的 ID1 與 ID2

MTPDS裡所有的MTPD都被轉換成MTPD'，則MTPD也算是成功的轉換成MTPD'S。原本的ID1+MTPDS可以產生ID1M1，而現在ID2+MTPD'S則可以產生ID2M1。圖 5.17 展示當 MTPDS 轉換成 MTPD'S後，ID1M1如何變成ID2M1。在MTPDS當中，可以分解成許多MTPD步驟，當MTPD完全相同或是沒有使用到參考元件時，便可直接複製成為MTPD'，如圖 5.17 中的MTPD1'、MTPD3'及MTPD5'。當MTPD所使用的參考元件在ID1與ID2是不相同時，則如圖 5.17 中MTPD2'、MTPD4'、MTPD6'及MTPD7'，判斷參考元件相異的情況並尋求對應元件，然後將MTPD更改成為新的MTPD'。當所有的MTPD'都已建置，便可以組成新的MTPD'S，並可直接套用在ID2上以得到ID2M1。

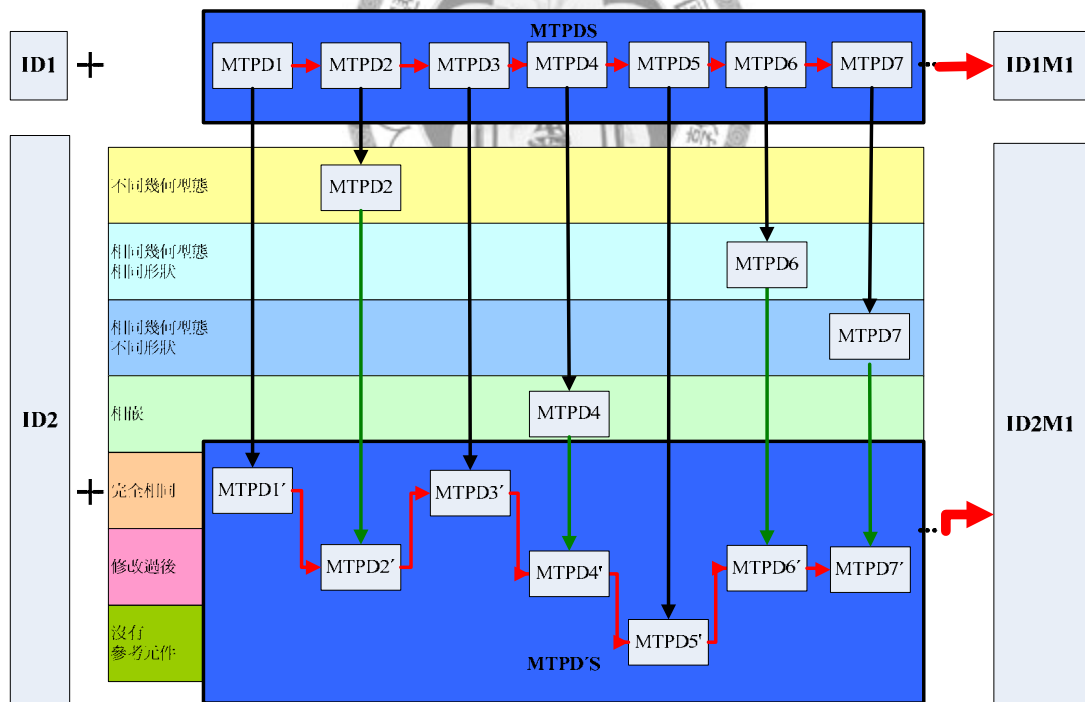


圖 5.17 MTPD 與 MTPD'之間的關係

5.4 協同討論

如果每個需要轉換的MTPD上之參考元件都是相同的，便不會有問題發生。然而當元件有變換時，使用者需要來回調整MTPD，並確定複製過程可以繼續。當機構設計端設法將MTPD轉換成MTPD'時，需要與其他人協同討論。

MTPDS的複製過程是在CAD上進行，所以協同討論必須有效的運用原本CAD的架構。原先是為了在不同CAD之間傳遞設計檔案，才使用中介格式檔案，若為了進行協同討論卻又安裝相同CAD，未免本末倒置。本節提出使用網頁協同以進行設計步驟複製，可以避免安裝，也可以達到分散協同的目的。

5.5 實作設計變更傳遞

本研究希望在不同CAD系統之間作設計變更的傳遞，使用SpringSolidCAD及其多層次架構以當做平台與機構設計軟體。使工業設計端的設計變更可以平順地傳遞至機構設計，並可利用網頁的特性，在步驟複製時進行協同討論。

5.5.1 系統架構

在多層次架構中，設計特徵可以用MTPD的型態來記錄，而整個模型最後以一個新的格式儲存。

在機構設計端，雖然當中介格式檔案的輸入從ID1變成ID2時，

有些幾何元件改變，只要MTPD用到的相關參考元件不變，系統仍可根據先前設計的MTPD執行核心運算。然而一旦對應參考元件在ID2已改變，MTPD就應該做些調整以符合新的輸入模型。

5.5.2 範例研究1: 手機機構設計

此輸入的工業設計STEP檔案是手機模型。原始面上已經開始進行一些機構設計。此模型增加一個分模線，並分離上下蓋，加厚下蓋。當機構持續進行設計，新的工業設計產生。圖5.18顯示這些模型間的關係。

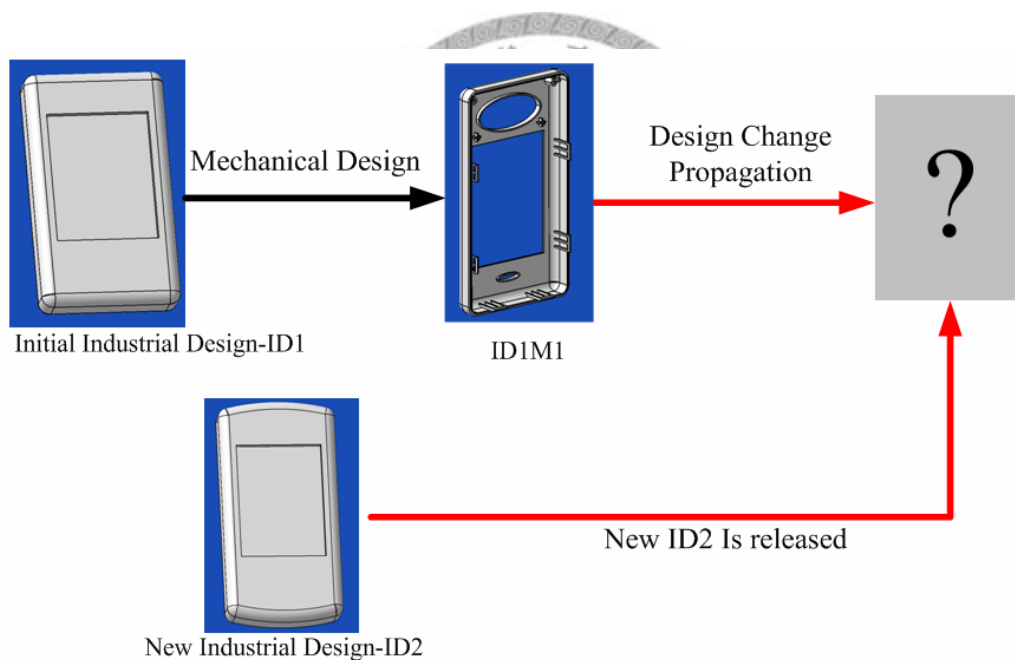
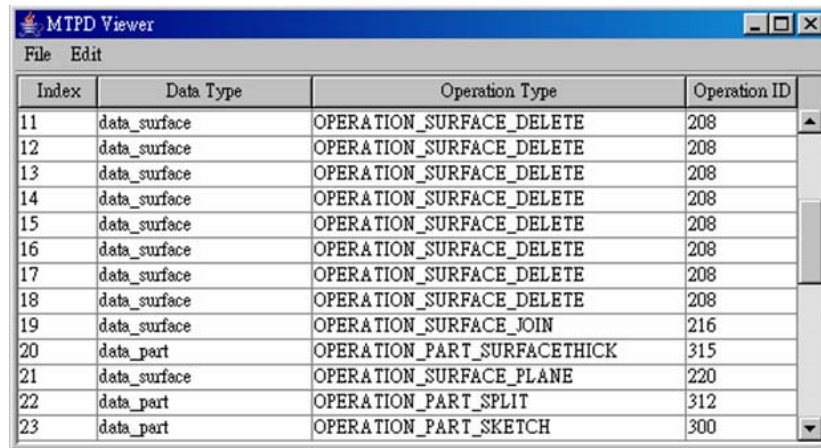


圖 5.18 ID1、ID2 與 ID1M1 之間的關係

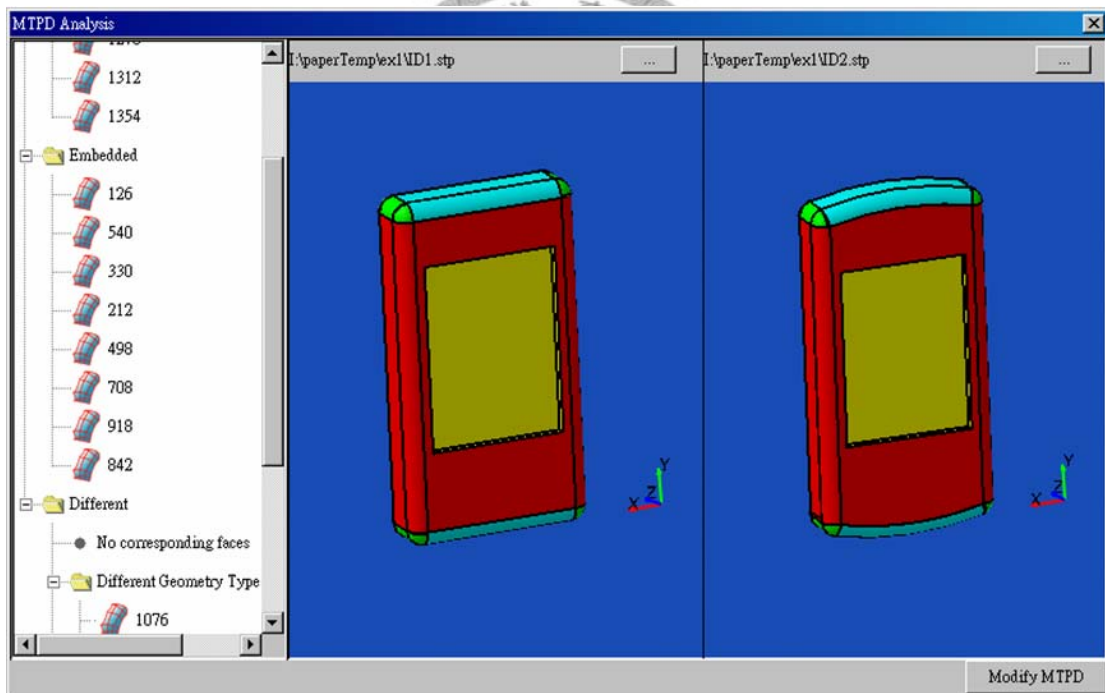
一旦設計變更發生，這些MTPDS會先從ID1M1被輸出。輸出MTPDS之後，系統會開始比較ID1與ID2，並且將ID1與ID2的面分類為三類。圖5.19顯示步驟順序。在圖5.19(a)，MTPD Viewer可以用來讀取MTPDS，並且將其中每個MTPD列出。(b)系統開啟ID1.stp與

ID2.stp並進行面比較。這些面會被分成三類，並且右手邊圖形會顯示，左手邊也有樹狀結構。



Index	Data Type	Operation Type	Operation ID
11	data_surface	OPERATION_SURFACE_DELETE	208
12	data_surface	OPERATION_SURFACE_DELETE	208
13	data_surface	OPERATION_SURFACE_DELETE	208
14	data_surface	OPERATION_SURFACE_DELETE	208
15	data_surface	OPERATION_SURFACE_DELETE	208
16	data_surface	OPERATION_SURFACE_DELETE	208
17	data_surface	OPERATION_SURFACE_DELETE	208
18	data_surface	OPERATION_SURFACE_DELETE	208
19	data_surface	OPERATION_SURFACE_JOIN	216
20	data_part	OPERATION_PART_SURFACE_THICK	315
21	data_surface	OPERATION_SURFACE_PLANE	220
22	data_part	OPERATION_PART_SPLIT	312
23	data_part	OPERATION_PART_SKETCH	300

(a)



(b)

圖 5.19 利用 MTPD Viewer 檢視與分析 (a) MTPD viewer 顯示 MTPDS 中的每個 MTPD (b) 讀取 ID1 跟 ID2 後顯示比對結果

分析完ID檔案跟MTPD過後，MTPD如果有需要修改者會被列出來。如果參考元件是不相同，才需要修改。使用者只要按選擇MTPD的欄位，按下右鍵，系統會彈出右鍵視窗，提供可變更的對應元件指標，如圖5.20。

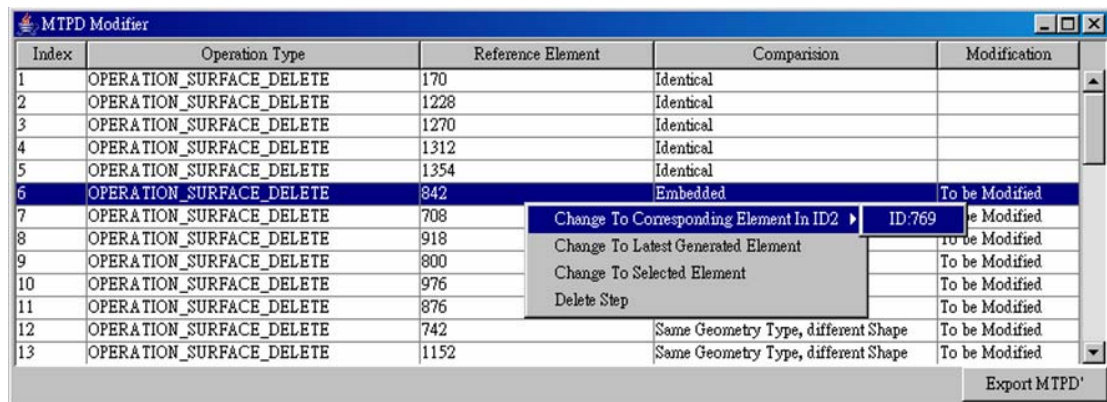


圖 5.20 手動選取 MTPD 項目並進行修改

在MTPD中有些參考元件，不能在ID1中找到，如圖5.21所示。因為這些參考元件是在某些MTPD執行之後才被產生的，應該要更新ID1跟ID2到最新狀態，而比較法則可以再執行一次便可得到新的比較表單。圖5.22顯示更新後的ID1與ID2之間的比較結果。

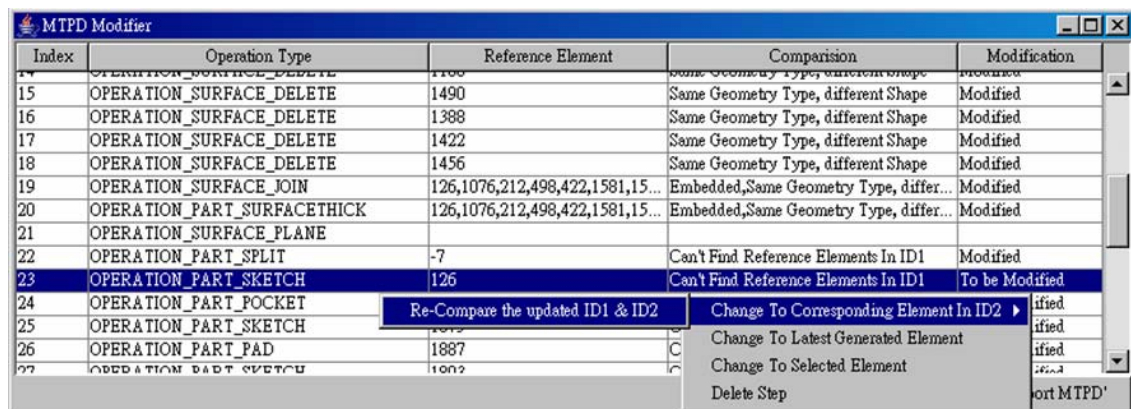


圖 5.21 系統建議 ID1、ID2 更新後再執行比較

當MTPD中參考元件不能在更新的ID1上找到時，使用者可以反覆地執行比較。最後，MTPDS中的MTPD都被修改，而使用便可藉由按下按鈕輸出MTPD'S。此MTPD'S可應用於ID2上已產生ID2M1。圖5.23展示MTPD'S應用於ID2的結果。

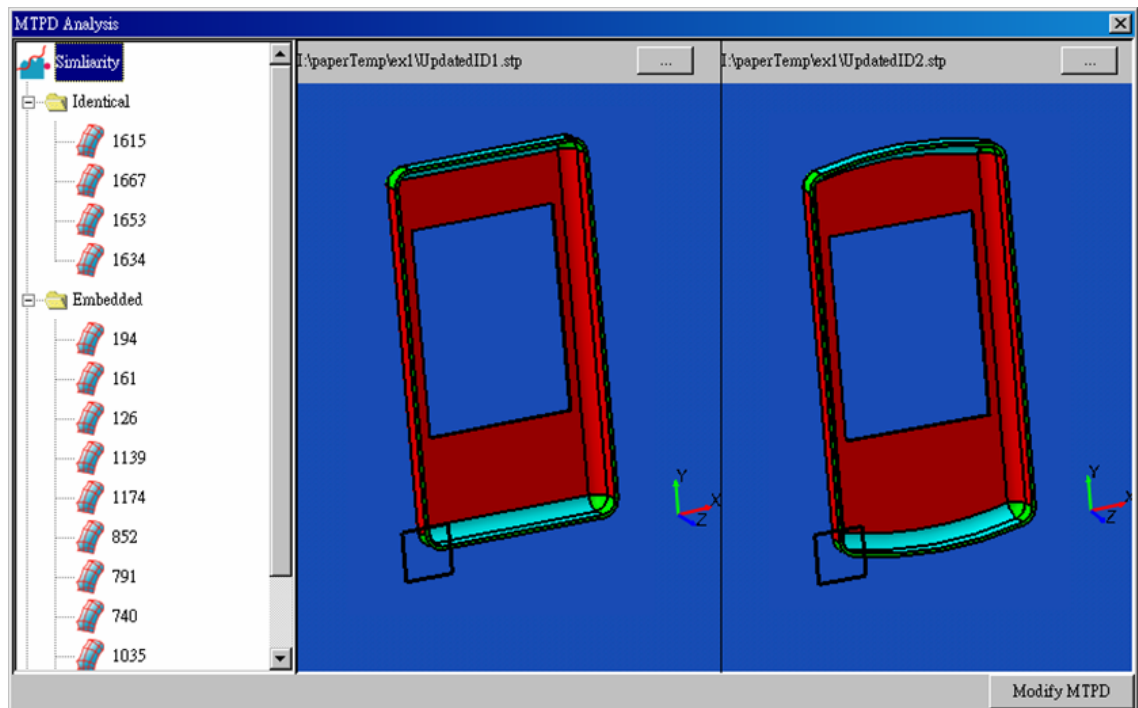


圖 5.22 更新的 ID1 與 ID2 比較

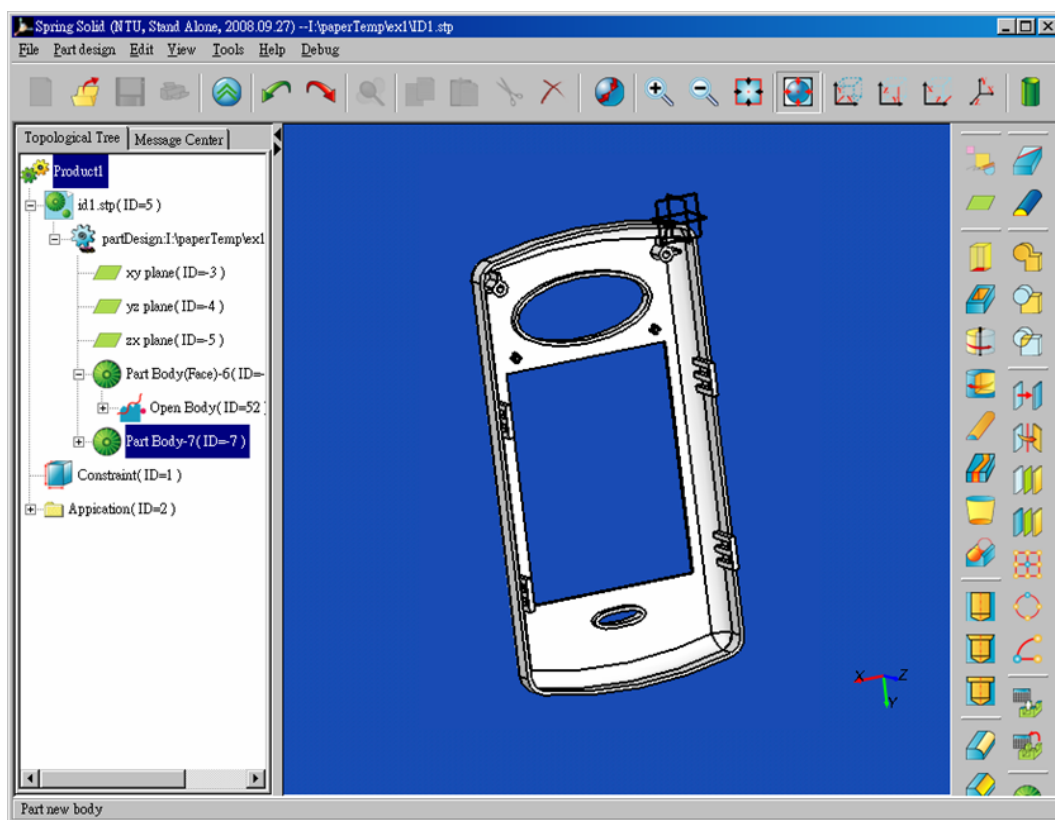


圖 5.23 應用 MTPD'S 於 ID2 所產生之結果

5.5.3 範例研究2：滑鼠機構設計

此輸入的工業設計STEP檔案是滑鼠模型。ID1M1是基礎機構設計。ID2中滑鼠的高度跟某些邊被調整。圖5.24顯示模型間的關係。

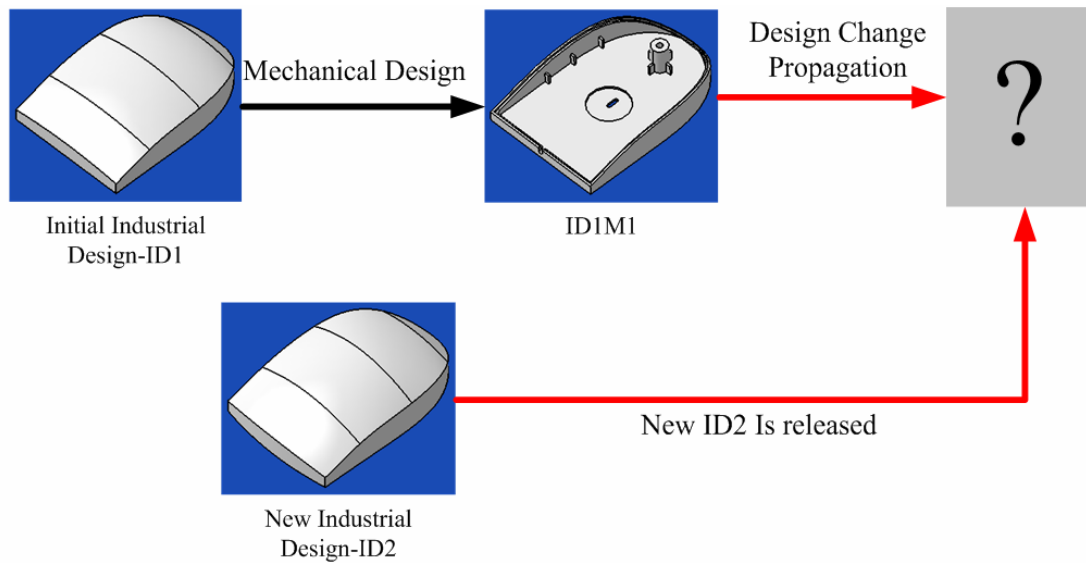


圖 5.24 ID1、ID2 與 ID1M1 之間的關係

系統會建議使用者執行下列的步驟：

- (1) 自 ID1M1 分析 MTPDS，並且全部在 MTPD Viewer 上展現，如圖 5.25。
- (2) 比較 ID1 與 ID2，並且在圖形與結構樹上顯示差異處，如圖 5.26。
- (3) 提供對應的面讓使用者可以在修改 MTPD 時選擇，如圖 5.27。
- (4) 如果參考元件在 ID1 無法找到，則再次執行比較，如圖 5.28。
- (5) 比較更新過後的 ID1 與 ID2，並且在圖形與結構樹上顯示差異處，如圖 5.29。
- (6) 應用輸出的 MTPD'S，將 ID2 改變為 ID2M1，如圖 5.30。

MTPD Viewer			
File Edit			
Index	Data Type	Operation Type	Operation ID
1	data_surface	OPERATION_SURFACE_DELETE	208
2	data_surface	OPERATION_SURFACE_DELETE	208
3	data_surface	OPERATION_SURFACE_DELETE	208
4	data_surface	OPERATION_SURFACE_DELETE	208
5	data_surface	OPERATION_SURFACE_JOIN	216
6	data_part	OPERATION_PART_SURFACETHICK	315
7	data_part	OPERATION_PART_SKETCH	300
8	data_surface	OPERATION_SURFACE_EXTRUSION	213
9	data_part	OPERATION_PART_SPLIT	312
10	data_part	OPERATION_PART_SKETCH	300
11	data_part	OPERATION_PART_POCKET	307
12	data_part	OPERATION_PART_SKETCH	300
13	data_part	OPERATION_PART_POCKET	307

圖 5.25 MTPD viewer 顯示 MTPDS 中的每個 MTPD

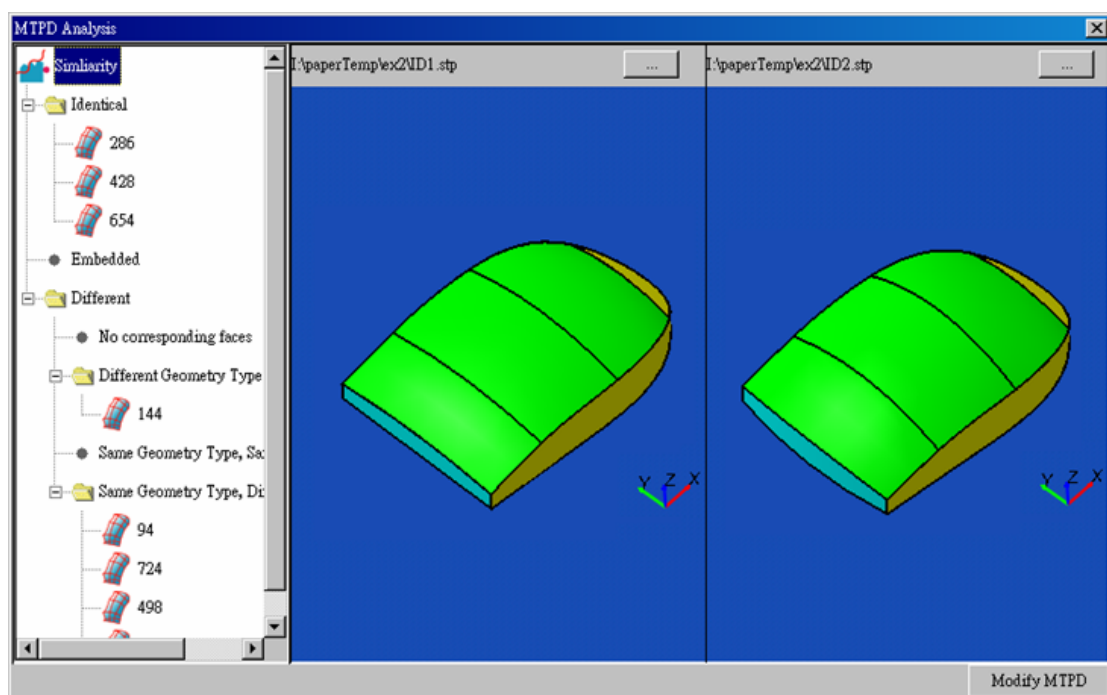


圖 5.26 ID1 與 ID2 之比較

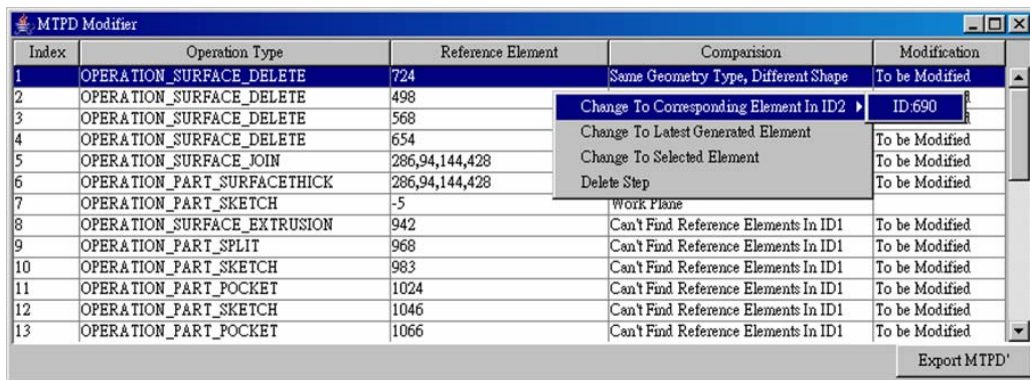


圖 5.27 修改 MTPD 項目時，系統提供對應面選擇

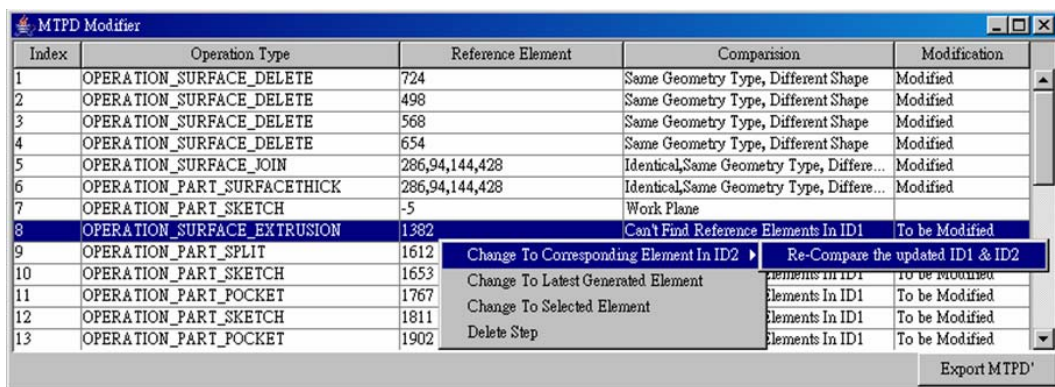


圖 5.28 為找到參考物件及其對應物件，系統建議執行再比較

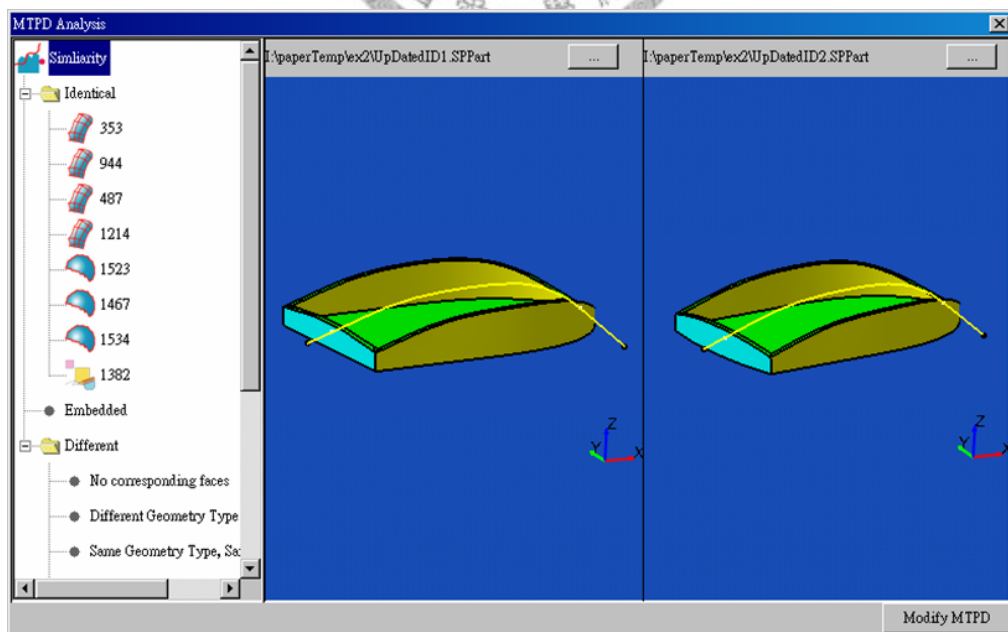


圖 5.29 比較更新過後的 ID1 與 ID2

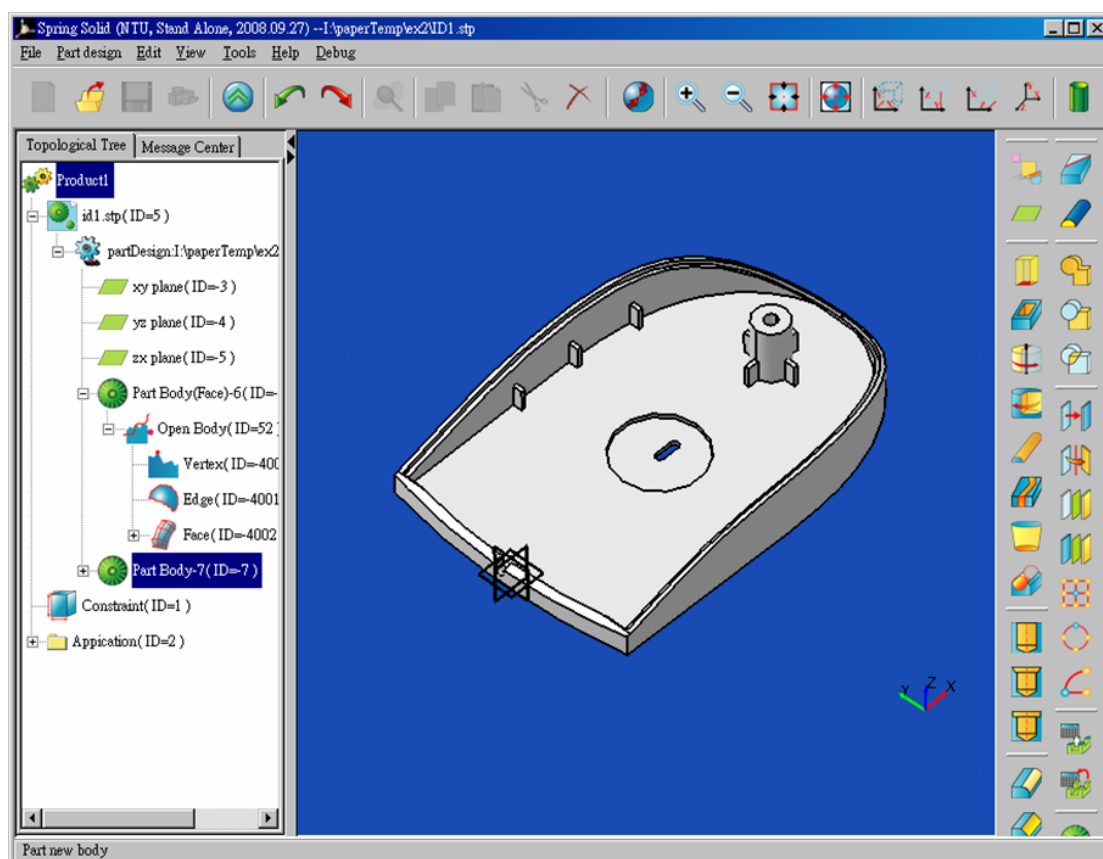


圖 5.30 將 MTPD'S 應用於 ID2 而得到之結果

5.5.4 成果討論

本章提出修改MTPDS來達成設變傳遞的目的，實作後可以得到新MTPD'S。在協同討論方面，只要使用者參加同一個專案，藉由在SpringSolidCAD的網頁平台上應用MTPD'S在ID2上，使用者便可以看到由ID2經過MTPD'S中的每一步驟，產生ID2M1。而ID2M1在網頁平台上產生後，使用者可以瀏覽、針對模型討論並做同步的縮放、旋轉及平移等動作。應用在單機版本或是對等式協同上，MTPD'S同樣也可以使ID2逐步轉變成ID2M1。修改或刪除MTPD'S中的步驟則會影響ID2M1最後的造型。

第六章 結論與未來展望

協同設計的觀念在CAD產業被廣泛的應用，本研究主要研究在上游工業設計與下游機構設計之間的接力設計中，大多使用中介格式檔案作設計延續，當其過程中工業設計端產生設計變更，將如何透過中介格式檔案傳遞至機構設計端。本研究提出一個CAD多層次彈性架構，應用於CAD系統，並利用此架構建置協同平台，達成在協同環境下以傳遞中介格式的方式延續跨不同CAD系統作設計，並利用傳遞資料格式進行動態紀錄，當先前CAD系統所設計的模型產生設計變更時，可以先將此CAD設計的紀錄提出，然後套用在新的中介格式檔上，便可快速地將新的變更傳遞至現行的設計上，因此可以解決現有的問題。本章中提出結論與未來展望。

6.1 論文成果與結論

6.1.1 論文成果

本論文主要成果分列如下：

A. 多層次彈性架構與CAD實作

- (1) 提出一個多層次且有彈性的架構，此架構可以分成七個層級，包含控制層、介面層、拓樸樹層、圖形層、通訊層、橋接層以及核心層。
- (2) 在架構中的資料傳遞採用巨集的概念，定義MTPD、NTD及NGD等資料型態。

- (3) 為配合協同產業的靈活運用，規劃三種可調整操作型態：點對點模式、精簡客戶端主從式架構及單機板。

B. 協同平台與動態協同紀錄

- (1) 協同平台採取三層式架構，分為使用者介面、伺服器介面以及資料庫，搭配本論文所提出多層次彈性架構，新增CAD/CAM 連結伺服器，並利用其通訊層作為PDM與CAD之間的資料交換。
- (2) 延伸代理伺服器的概念，將PDM系統放置在協同平台內設計資料傳輸要徑上，以消除重覆或不必要的要求
- (3) PDM系統收集客戶端與伺服器端送過的MTPD、NTD、NGD，加以分類紀錄。之後再依照使用者的指定方式，依序傳出MTPD或是NTD與NGD，客戶端接收後再顯示，便達成動態紀錄呈現。

C. 動態協同紀錄於設計變更傳遞之應用

- (1) 將正在進行的設計檔案分離成原始輸入的中介檔案以及MTPDS。
- (2) 從幾何和拓撲兩方面，針對設變前後不同版本的中介檔案進行差異比較。
- (3) 利用中介檔案的差異比較，半自動改變MTPDS，再將改變的MTPDS套用到中介檔案上，得到新的設計檔案，成功完成設變傳遞。

6.1.2 論文結論

本論文提出當不同CAD間用中介檔案做接力設計時，設計變更傳遞的方法。設計傳遞可以藉由MTPDS以及中介檔案的搭配以達成。當中介檔案因設變而改變時，差異性比較扮演一個重要的角色。本論文結合兩種面的比較法則，可以在新中介檔案中找到對應在舊機構設計參考元件的新元件，並且將其用來更新MTPDS。程式可自動辨識兩個3D模型之間的差異性，遠比使用者用肉眼來得輕鬆。

6.2 未來展望

本研究雖實作多層次協同架構、協同平台、動態紀錄並最後達成跨中介格式之設變傳遞，但尚有許多不完備可供改進之處，茲提出下列幾點，可作為未來改進：

(1) 就一個網頁為基的平台，CAD伺服器的分散計算是一個未來的研究方向。分散的計算不但可減少CAD伺服器的計算負擔，當其中的某些伺服器發生問題時，也提供一些fail-safe的機制。另外，如果非同步協同的流程回顧可以變成一個多媒體檔案或是僅一個連結就可以得到的影片，網頁的整合將會更方便。

(2) 設計變更是企業界常會碰到的問題，有許多設計變更傳遞的研究在PDM系統中零件間的相互關係做設計變更傳遞，如組裝件的某些零件產生設計變更時，組裝對應件

會產生改變。如果能將設計步驟及上游工業設計的輸入中介格式版本做資料管理時，則當新設變過後的工業設計以中介格式再次輸入時，也可以在PDM系統中做管理，並且在PDM系統中做設變前後工業設計的中介檔案相似性比對，並產生新的設計步驟。

- (3) 步驟的動態紀錄僅能夠完全紀錄，當使用者作無意義的往復操作時，僅能檢查前後步驟是否重複。系統若能各檢查無效的步驟並自動刪除，動態紀錄的功能將可更趨完善。



參考文獻

- [1] 尤春風，實體模型系統，課程講義，1995
- [2] C. Y. Kim, N. Kim, Y. Kim, S. H. Kang and P. O'Grady, "Distributed concurrent engineering internet-based interactive 3-D dynamic browsing and markup of STEP data," *Concurrent Engineering: Research and Application*, Vol.6, No.1, pp. 53–70, 1998.
- [3] Geometric Software Inc, CollabView, <http://www.geometricsoftware.com/technologies/collabview.htm>, 2002.
- [4] H. Hoppe, "Progressive meshes," *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 99-108, 1996.
- [5] G. Taubin, A. Gueziec, W. Horn and F. Lazarus., "Progressive forest split compression," *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 123–132, 1998.
- [6] CollabCAD National Informatics Centre, India. <http://www.collabcad.com> , 2001.
- [7] T. J. Nam and D. K. Wright, "CollIDE: a shared 3D workspace for CAD," *Proceedings of the 1998 Conference on Network Entities, Leeds, UK*; pp. 389–400, 1998.
- [8] L. Qiang, Y.F. Zhang and A.Y.C. Nee. "A distributed and collaborative concurrent product design system through the WWW/Internet," *International Journal of Advanced Manufacturing Technology*, Vol.17, No.5, pp. 315–322, 2001.

- [9] F. Pahng, N. Senin and D. Wallace, "Distribution modeling and evaluation of product design problems," *Computer-Aided Design*, Vol.30, No.6, pp. 411–423, 1998.
- [10] H. C. Chang, W. F. Lu and X. F. Liu, "WWW-based collaborative system for integrated design and manufacturing," *Concurrent Engineering: Research and Application*, Vol. 7, No.4, pp. 319–334, 1999.
- [11] N. Shyamsundar and R. Gadh, "Collaborative virtual prototyping of product assemblies over the Internet," *Computer-Aided Design*, Vol.34, pp. 755-768, 2002.
- [12] IX Design ImpactXoft Inc, <http://www.impactxoft.com>, 2002.
- [13] Q. Sun and K. Gramoll, "Internet-based distributed collaborative engineering analysis," *Concurrent Engineering: Research and Application*, Vol.10, No.4, pp. 341–348, 2002.
- [14] Alibre Design Alibre Inc, <http://www.alibre.com>, 2002.
- [15] GS-Design Inc, <http://www.gs-design.com>, 2002.
- [16] W. D. Li, S. K. Ong, J. Y. H. Fuh, Y. S. Wong, Y. Q. Lu and A. Y. C. Nee, "Feature-based design in a distributed and collaborative environment," *Computer-Aided Design*, Vol.36, pp. 775–797, 2004.
- [17] OneSpace CoCreate Inc, <http://www.onespace.com>, 2006.
- [18] L. Chen, Z. Song and L. Feng, "Internet-enabled real-time collaborative assembly modeling via an e-Assembly system: status and promise," *Computer-Aided Design*, Vol.36, pp. 835–847, 2004.

- [19]K. Ramani, A. Agrawal, M. Babu and C. Hoffmann, "CADDAC: Multi-client collaborative shape design system with server-based geometry kernel," *Journal of Computing and Information Science in Engineering*, pp. 170–173, 2003.
- [20]S. Chan, M. Wong and V. Ng, "Collaborative solid modeling on the WWW," *Proceeding of the 1999 ACM Symposium on Applied Computing*, San Antonio, Texas, pp. 598–602, 1999.
- [21]R. Bidarra, E. Van Den Berg and W. F. Bronsvort, "Web-based collaborative feature modeling," *Proceedings of Solid Modeling '01-Sixth Symposium on Solid Modeling and Applications*, pp. 319–320, 2001.
- [22]C. V. Foster, Y. Shapirstein, C. D. Cera and W. C. Regli, "Multi-user modeling of nurbs-based objects," *Proceedings of DETC'01, 2001 ASME Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, pp. 1–10, 2001.
- [23]Dassault Systemes Inc, <http://www.3ds.com/home>, 2002.
- [24]PTC Inc, <http://www.ptc.com>, 2002.
- [25]S. C. F. Chan and V. T. Y. Ng, "Real-time collaborative solid shape design (RCSSD) on the Internet," *Concurrent Engineering: Research and Application*, Vol. 10, No.3, pp. 229–238, 2002.
- [26]G. H. Choi, D. Mun and S. Han, "Exchange of CAD part models based on the macro-parametric approach," *International Journal of CAD/CAM*, Vol.2, No.1, pp.13–21, 2002.
- [27]M. Li, S.M. Gao and C.C.L. Wang, "Real-time collaborative

- design with heterogeneous CAD systems based on neutral modeling commands,” *Journal of Computing and Information Science in Engineering*, Vol.7, No.2, pp. 113–125, 2007.
- [28]W. D. Li, J. Y. H. Fuh and Y. S. Wong, “An Internet-enabled integrated system for co-design and concurrent engineering”, *Computers in Industry*, Vol.55, No.1, pp. 87- 103, 2004.
- [29]J. Silva and K. H. Chang, “Design parameterization for concurrent design and manufacturing of mechanical systems,” *Concurrent Engineering: Research and Application*, Vol. 10, No.1, pp. 3–14, 2001.
- [30]ISO 10303-42, Integrated generic resource: Geometric and topological representation, 1994.
- [31]ISO 10303-203, Application protocol: Configuration controlled 3D design of mechanical parts and assemblies, 1994.
- [32]ISO 10303-214, Application protocol: Core data for automotive mechanical design processes, 1997.
- [33]IGES/PDES Organization, Initial graphics exchange specification IGES 5.3, 1996.
- [34]ISO TC184/SC4, ISO 10303-42-Part 42: Industrial automation systems and integration - Product data representation and exchange - Integrated generic resources: Geometric and topological representation, ISO, 1994.
- [35]S. L. Brooks and W. J. Kirk, “Using STEP to integrate design features with manufacturing features,” *ASME International Computer in Engineering Conference*, pp. 579–586, 1995.

- [36]J.J. Shah, D. Anderson, Y. S. Kim and S. Joshi, "A discourse on geometric feature recognition from CAD models," *Journal of Computing and Information Science in Engineering*, Vol.1, No.1, pp.41–51, 2001.
- [37]M. W. Park and H. Ko, "A feature recognition system with interactive feature inspection and deletion capabilities," *The International Journal of Advanced Manufacturing Technology*, Vol. 3, pp. 190–160, 1996.
- [38]S. Myung and S. Han, "Knowledge-based parametric design of mechanical products based on configuration design method," *Expert Systems with Applications*, Vol.21, No.2, pp. 99–107, 2001.
- [39]M. Novotni and R. Klein, "A geometric approach to 3D object comparison," *International Conference on Shape Modeling and Applications*, pp 154–166, 2001.
- [40]J. Zhang and E. M. Rasmussen, "Developing a new similarity measure from two different Perspectives," *Information Processing and Management*, Vol.37, No.2, pp. 279–294, 2001.
- [41]A. Cardone, S. K. Gupta and M. Karnik, "A Survey of shape similarity assessment algorithms for product design and manufacturing applications," *Journal of Computing and Information Science in Engineering*, Vol.2, pp. 109–118, 2003.
- [42]N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman and K. Ramani, "Shape-based searching for product lifecycle applications," *Computer-Aided Design*, Vol.37, No.13, pp. 1435–1446, 2005.

- [43]V. Cicirello and W. C. Regli, "Machining feature-based comparisons of mechanical parts," *International Conference on Shape Modeling and Applications*, pp. 176–185, 2001.
- [44]M. El-Mehalawi and R. A. Miller, "A database system of mechanical components based on geometric and topological similarity. Part I: representation," *Computer-Aided Design*, Vol. 35, No.1, pp. 83–94, 2003.
- [45]M. El-Mehalawi and R. A. Miller, "A database system of mechanical components based on geometric and topological similarity. Part II: indexing, retrieval, matching and similarity assessment," *Computer-Aided Design*, Vol.35, No.1, pp. 95–105, 2003.
- [46]Y. Kawai, Y. Higashiyama, K. Koyama and M. Okada, "A fundamental study on a natural-language-based 3D CG modeling," *1999 IEEE International Conference*, pp. 5:714–719, 1999.
- [47]OBJ format, <http://en.wikipedia.org/wiki/Obj>, 2002.
- [48]Apache Tomcat, <http://tomcat.apache.org/>, 2006.

作者簡歷

姓名：趙甦農

生日：69 年 2 月 15 日

學歷：博士 台灣大學機械工程研究所 2003~2009

碩士 台灣大學機械工程研究所 2002~2003

學士 台灣大學機械工程學系 1998~2002

住址：台灣省台中市北屯區太原路 3 段 111 巷 22 號 1F

電話：0928730535

e-mail：f91522608@ntu.edu.tw



歷年發表文獻

期刊論文：

□ **Multilayer Architecture in Collaborative Environment**

DOI: 10.1177/1063293X06073128

Concurrent Engineering, Vol. 14, No. 4, 273-281, 2006.

□ **Propagation of design change between different CAD by using duplicate design procedures**

DOI : 10.1007/s00170-008-1833-6

The International Journal of Advanced Manufacturing Technology
on line first

Received: 29 March 2008 / Accepted: 31 October 2008

