

石油暨石化產業科技學術合作 八十八年度期末報告

中油儲運管理網際資訊系統的研發

計畫編號：NSC-88-CPC-E-009-018

執行期間：88年1月1日至88年12月31日

委託單位：中國石油股份有限公司

計畫主持人：黎漢林 教授

執行單位：國立交通大學資訊管理研究所

中華民國 88 年 12 月

88CPC-E-009-018

中油儲運管理網際資訊系統
的研發

結案報告

國立交通大學資訊管理研究所

中華民國八十八年十二月三十一日

目錄

第一章.	緒論.....	1
1.1	設計緣起.....	3
1.2	Web GIS 宜有功能.....	6
1.3	Web GIS 現有系統比較.....	11
1.4	關鍵性設計議題.....	22
1.5	發展的步驟.....	24
第二章.	Web GIS 互動性技術探討.....	27
2.1	CGI.....	29
2.2	ISAPI.....	31
2.3	JavaScript 與 VBScript.....	33
2.4	Java.....	35
2.5	ActiveX.....	38
2.6	ASP.....	40
2.7	Web GIS 互動性技術比較表.....	41
第三章.	Web 上分散式物件理論分析.....	43
3.1	分散式物件之架構-OMA.....	45
3.2	分散式物件之使用者操作介面.....	51
3.3	Windows Web 上的分散式物件-ActiveX.....	55
第四章.	Web GIS 介面分析.....	59

4.1	CGI 介面 Web GIS.....	61
4.2	Embedded Program 介面 Web GIS	63
4.3	Multi-Tier Embedded Program 介面 Web GIS.....	65
4.4	Web GIS 介面比較表	67
第五章. Web GIS 設計課題與模式		69
5.1	Web GIS 設計課題	71
5.1.1	Web GIS 伺服器負荷之問題.....	71
5.1.2	使用者環境設定問題(Thin Client).....	74
5.1.3	Web GIS 資料維護問題.....	75
5.1.4	網路頻寬問題	77
5.2	Web GIS 設計模式	79
5.2.1	模式一(DCOM).....	79
5.2.2	模式二(TCP/IP->DCOM)	83
5.2.3	模式三(MIDAS).....	87
5.2.4	模式比較	92
5.3	設計功能分析	93
第六章. Web GIS 資料庫分析與設計		95
6.1	資料庫之特性.....	97
6.2	需求分析.....	99
6.3	資料庫 E-R Model	100
6.4	資料庫之綱目(Schema).....	102
6.4.1	人員權限檔 : Employee.....	102
6.4.2	底圖檔 : Map.....	103
6.4.3	圖示檔 : Logo.....	103
6.4.4	單一點之據點資料&主點之據點資料檔 : Data	104
6.4.5	主點之子據點資料 : DataSub.....	105
第七章. Web GIS Server 地圖庫之建構		107
7.1	底圖取得.....	109
7.2	底圖掃描.....	111

7.3	底圖校調.....	114
7.4	底圖資料庫之建構.....	123
第八章. 本系統說明.....		124
8.1	查詢模式.....	127
8.1.1	查詢底圖及圖層資料.....	127
8.1.2	以圖形區塊查詢.....	130
8.1.3	以保留字查詢.....	132
8.2	遠程分散式編修模式.....	133
8.2.1	身份確認.....	133
8.2.2	人員權限設定.....	134
8.2.3	圖層編修.....	135
8.2.4	據點編修.....	135
第九章. 結論與討論.....		138
參考文獻.....		141

表目錄

表 1	GIS 與目前 Web GIS 之比較表.....	3
表 2	各種 Web GIS 功能比較表.....	21
表 3	JavaScript 之範例.....	34
表 4	Java 之範例.....	36
表 5	常用瀏覽器所支援之 Java 規格.....	37
表 6	ActiveX 之範例.....	39
表 7	各種 WWW 互動性技術之比較表.....	41
表 8	Core Object Model 範例.....	47
表 9	各種 Web GIS 介面比較表.....	67
表 10	各種模式之優缺比較.....	92

圖目錄

圖 1 點落面查詢範例.....	7
圖 2 矩形區域查詢範例.....	7
圖 3 圓形區域查詢範例.....	8
圖 4 環域查詢範例.....	8
圖 5 路徑查詢範例.....	8
圖 6 美國人口普查局 Web GIS.....	12
圖 7 Xerox PARC 的 Map Viewer.....	13
圖 8 Yahoo Maps Web GIS[.....	14
圖 9 足普度大學的 INGIS.....	15
圖 10 SEDAC 的 Web GIS.....	17
圖 11 中央大學太空及遙控中心 Web GIS 畫面 1.....	18
圖 12 中央大學太空及遙控中心 Web GIS 畫面 2.....	19
圖 13 CGI 之運作流程.....	30
圖 14 OMA 架構圖[12].....	45
圖 15 COBRA 物件呼叫架構圖.....	49
圖 16 ActiveX 的驗證(Authenticate)機制圖.....	57
圖 17 CGI 介面 Web GIS 之運作流程.....	61
圖 18 Embedded Program 介面 Web GIS 之運作流程.....	64
圖 19 Multi-Tier Embedded Program 介面 Web GIS 之運作流程.....	66
圖 20 一般 Web 之架構.....	72
圖 21 Multi-Tier 架構之 Web.....	73
圖 22 DCOM 之 Multi-Tier Embedded Program 介面 Web GIS.....	79
圖 23 TCP/IP 之 Multi-Tier Embedded Program 介面 Web GIS.....	83
圖 24 MIDAS 之 Multi-Tier Embedded Program 介面 Web GIS.....	87
圖 25 本系統資料庫之 E-R Model.....	100
圖 26 PhotoImpact 操作畫面 1.....	111
圖 27 PhotoImpact 操作畫面 2.....	112
圖 28 PhotoImpact 操作畫面 3.....	113

圖 29	PhotoImpact 操作畫面 4	113
圖 30	PhotoImpact 操作畫面 5	114
圖 31	PhotoImpact 操作畫面 6	114
圖 32	PhotoImpact 操作畫面 7	116
圖 33	PhotoImpact 操作畫面 8	117
圖 34	PhotoImpact 操作畫面 9	118
圖 35	PhotoImpact 操作畫面 10	119
圖 36	PhotoImpact 操作畫面 11	120
圖 37	PhotoImpact 操作畫面 12	121
圖 38	PhotoImpact 操作畫面 13	122
圖 39	PhotoImpact 操作畫面 14	123
圖 40	底圖資料庫建構之執行畫面	124
圖 41	中油網際地理資訊系統執行畫面 1	127
圖 42	中油網際地理資訊系統執行畫面 2	128
圖 43	中油網際地理資訊系統執行畫面 3	129
圖 44	中油網際地理資訊系統執行畫面 4	130
圖 45	中油網際地理資訊系統執行畫面 5	131
圖 46	中油網際地理資訊系統執行畫面 6	131
圖 47	中油網際地理資訊系統執行畫面 7	132
圖 48	中油網際地理資訊系統執行畫面 8	133
圖 49	中油網際地理資訊系統執行畫面 9	134
圖 50	中油網際地理資訊系統執行畫面 10	134
圖 51	中油網際地理資訊系統執行畫面 11	135
圖 52	中油網際地理資訊系統執行畫面 12	135
圖 53	中油網際地理資訊系統執行畫面 13	136
圖 54	中油網際地理資訊系統執行畫面 14	137
圖 55	中油網際地理資訊系統執行畫面 15	137
圖 59、60	中油網際地理資訊系統執行畫面 19、20	139

第一章.

緒論

1.1 設計緣起

近年來國營事業紛紛民營化，油品的供應也由原本中油的一家獨佔慢慢出現競爭。本計畫擬在網路上發展一儲運管理的地理資訊系統，使石油油品儲運成本降低、效率提高。此套系統除了可以管理石油公司營運相關的土地、道路、加油站、儲運槽及油罐車路線等相關資訊，尚可結合資料庫及數學模式庫以作儲運管理之最佳化運算。可應用的範圍除了一般的文字功能查詢外，尚可延伸到有關地理資訊的查詢，成為一套完整的營運管理資訊系統並進而引入環球供應鏈(Global Supply Chain)的概念發展石油儲運最佳化的管理模式。

運輸成本為產品總成本的一部份，根據先進國家之統計資料顯示，工商產品的運輸成本約佔產品價值的 20%~30%，其中以中下游之配銷成本所佔比例最高。為此，石油產品引進資訊科技輔助經營管理確有其必要性。本計畫的目的就在為石油公司在網路上發展一儲運管理地理資訊系統，以降低油品儲存與運送成本。而目前許多地理資訊系統多存有以下缺失：

1. 無法透過網路取得地圖上的屬性資訊，譬如在網路地圖上可找到某個加油站，但無法直接在圖上點選出加油站的進一步資料（如營業時間，油品種類，服務項目等）。
2. 無法透過網路作空間資料的關聯查詢，譬如無法在網路地圖上列示出某段路上的所有加油站。
3. 網路地圖資料無法與特定運算模式結合，譬如在網路地圖上無法顯示油罐車巡行的最低成本路徑。
4. 多數為外國軟體，對本土的使用不方便。

而 GIS 乃一包含點、線、面等空間資料的整合性資料庫，讓使用者透過軟硬體的輔助得以操作這些空間或屬性資料[Tsichrizis 9]。GIS 的功能認為可分為下列幾項：資料取得(Data acquisition)、前置處理(Preprocessing)、資料管理(Data management)、資料操作

與分析(Manipulation and Analysis)與結果產生(Product generation)[Kemp 10]。使用者透過這些功能的支援即可取代傳統的地圖與資料作業，使得地理資訊的產生與分析更為精確。Web GIS¹就是在 Internet 的 WWW²上設計一 GIS，使 Web GIS 具有 GIS 與 Internet 的功能，GIS 與目前 Web GIS 之比較如表 1：

比較項目	Web GIS	傳統 GIS
系統環境	Internet 上之 WWW	單機定點式。
使用性	立即上線，資料不須額外加工。	須系統管理師加以安裝、規劃才能使用。
資料更新	可隨時。	一旦更新後需要重新發行，安裝才可。
操作介面	透過瀏覽器。	各系統不一，使用者須重新學習。
頻寬要求	高	不需要
技術層次	起步中，不成熟。	成熟且完整。
功能	簡單(受限於技術，頻寬)	完整(不受限制)。
發展工具	少且不完整。	多且完整。
系統維護	缺乏完整工具須專業人員	已有完整工具不須專業人員

表 1 GIS 與目前 Web GIS 之比較表

Web GIS 目前雖有享受到 Internet 的便利，但受限於 Internet 上 Web 技術的不成熟，而使 GIS 應有的功能大幅減少。再加上 Internet 頻寬不足，無法在 Web GIS 放置複雜資料，所以無法發揮應有的功能。目前 Web GIS 的通病為：

功能過於簡單

¹ Web GIS：Web Geographical Information Systems 網際地理資訊系統的縮寫，指在瀏覽器上面執行之 GIS，也有人稱為 IGIS。

² WWW：World Wide Web。

只提供定點式 GIS 之查詢，無法像專業 GIS 般提供完整圖形編輯查詢、屬性查詢、主題圖層疊合、模式分析...等功能。

不具分散式資料管理之功能

分散在各地的資料提供者無法透過 Internet 自行將資料更新，必須將資料集中至主電腦且透過專業人士做資料更新。造成 Web GIS 上的資料時效性差，且維護困難。

Web GIS 之地圖查詢速度太慢

對使用者而言用數據機上線，查詢一般圖形資料耗時甚多。

本文配合 Multi-Tier、DCOM 及 ActiveX 等新技術提出一套新模式用以克服上述之缺點，進而建構一套功能較為完整之 Web GIS。本系統最大特點有遠程資料維護、底圖資料快取、後端分散式處理及 Thin Client 的 Multi-Tier 架構(原理請參考章節 5.1)。且實際運用於公路災害查報系統³及新竹市政府網頁 Web GIS 系統⁴上。對於 Internet 上的資料安全及有關網路資訊所衍生的智慧財產權問題則不在討論範圍內。

³ 公路局災害查報系統：為交通大學資管所承接之計畫。

⁴ 新竹市政府之網站：為交通大學資管所承接之計畫。

1.2 Web GIS 宜有功能

傳統 GIS 有以下幾項缺點：

- (1) 無特殊(adhoc)存取的能力
- (2) 檔案結構各系統不能協調
- (3) 多數為單機系統
- (4) 不提供資料復原(recovery)與一致性(consistency)
- (5) 不支援分散式系統(Lohman[4]、Tsichrizis[9]、Kemp[10])

這些缺陷多由於傳統 GIS 是單機作業，且其空間資料是採用檔案系統。當在 Internet 上設計 GIS 時，問題 3 可自動解決，但問題 4 與 5 會更突出，一套好的 Web GIS 應做到下列功能：

多樣的資料查詢方式

Web GIS 查詢方面至少要提供兩種查詢方式

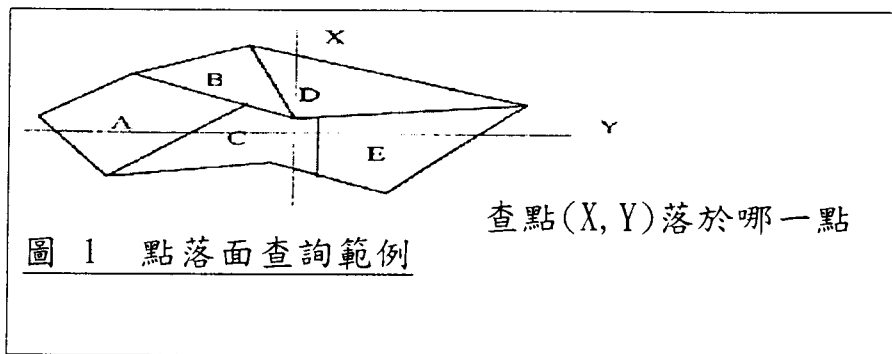
(1) 關鍵字(Keyword)查詢地圖

通常是先決定要查詢之範圍再輸入關鍵字加以查詢，而將查詢結果彙整成一張表，一一對應到相對之地圖。

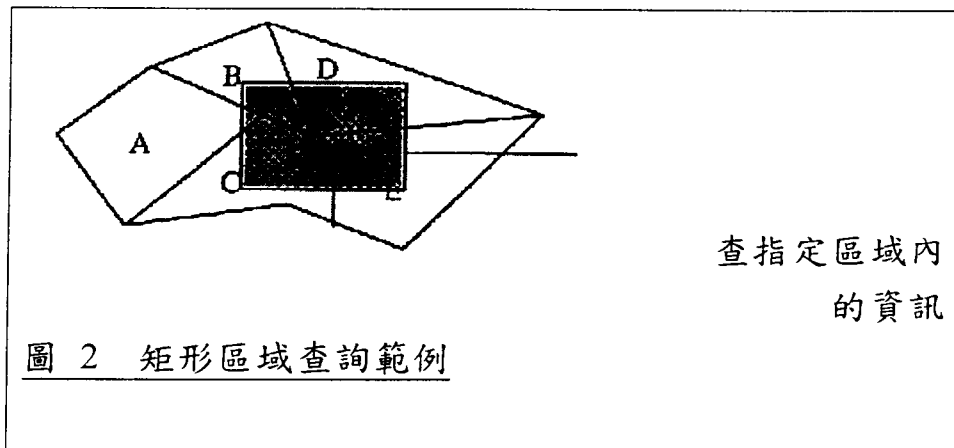
(2) 圖形查詢法查到 GIS 資料庫

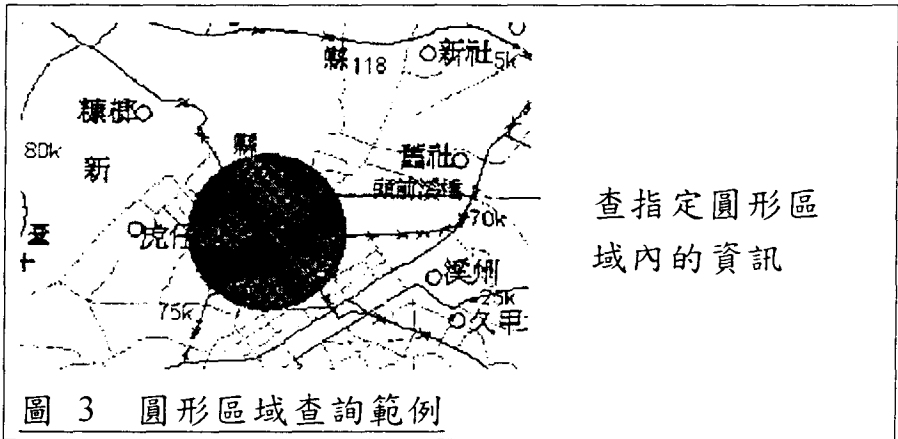
常用的圖形查詢法有下列幾種[3]:

◇點落面查詢(Point-in-polygon queries)，如圖 1。

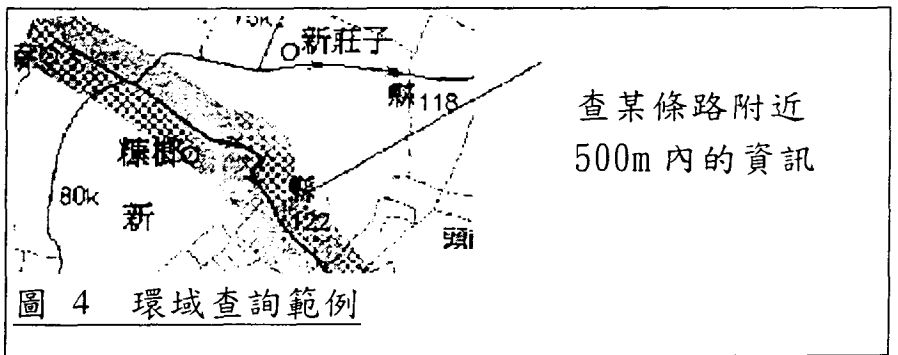


◇區域查詢(Region queries) ，如圖 2、3。

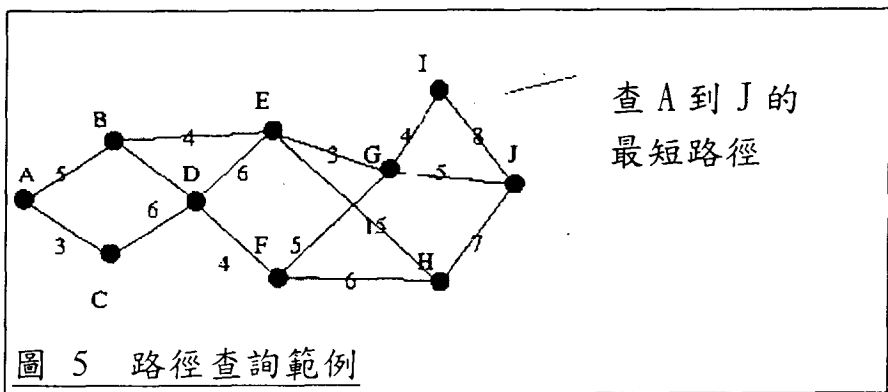




◇距離及環域查詢(Distance and buffer zone queries)，如圖 4。



◇路徑查詢(Path queries)，如圖 5。



以上的查詢方式，需要拉線、拉框及畫圓等動作，但在 WWW 中，除了點(click)動作外，無法做其他動作。也就是說，瀏覽器已限制 Web GIS 的發展空間。在這方面可採用一些前端互動技術(請參考章節 2.3)，如 ActiveX、Java...等，來彌補 WWW 瀏覽器先天的不足，以滿足上述的查詢方式。

多樣的地圖顯示功能

最重要的功能為圖層疊合，使用者可以自行決定哪些圖層要顯示，哪些圖層不顯示。而圖層多寡也決定此系統所提供資訊的多寡。除此之外尚提供地圖放大(Zoom In)、縮小(Zoom Out)、或是往四面八方移動之功能。

快速的地圖呈現

若使用者選地圖的等候時間長，且每做一個圖層疊合、放大(Zoom In)、縮小(Zoom Out)、或是往四面八方移動的時間也長，使用者往往會失去耐心而關機。所以一套好的 Web GIS 講求地圖資訊的呈現(包含產生與傳回至使用者前端)，其速度一定要比使用者自行翻書尋找資料快速。

後端可支援分散式處理

一套 Web GIS 常要同時服務許多使用者，因此要具備分散式計算及處理之能力。也就是說同時有許多台 Web GIS 前置處

理器負責與前端使用者溝通，用以分擔 Web GIS 伺服器之負荷。而此即為 Multi-Tier 架構(請參閱第 5 章)。

遠程分散式資料更新

「GIS 資料是 GIS 的運作素材，一般而言都是其建構過程中，成本最大的部份（65%~85%），亦最難完成的部份」（朱子豪 [1]）。可見資料維護對 GIS 的重要，要使 Web GIS 真正享受到 Internet 的便利性，不只一般民眾可以輕易使用。還要資料提供者也能輕易維護 Web GIS 上的資料。如此就要提供遠程分散式資料更新，資料提供者即使分散在世界各地還是可更新資料，且最好就透過瀏覽器更新。如此 Web GIS 的資料才會具有時效性，而維護成本也大為降低，資料提供者也不用具備 Web GIS 專業知識即可成為主動式更新者。

線上資料更新時之交易(Transaction)處理

如果提供第 5 點所述之功能，則在多人環境下更新 Web GIS 資料，一定要考慮到線上更新資料時之交易(Transaction)處理(包含同步處理及資料復原)。因為在 Internet 上資料隨時有資料遺失、斷線的危險。若 Web GIS 不支援交易(Transaction)處理則有可能會造成一旦斷線，資料又儲存的不完整，造成整個資料庫大亂(如參考性整合被破壞，唯一鍵被破壞...等)。

1.3 Web GIS 現有系統比較

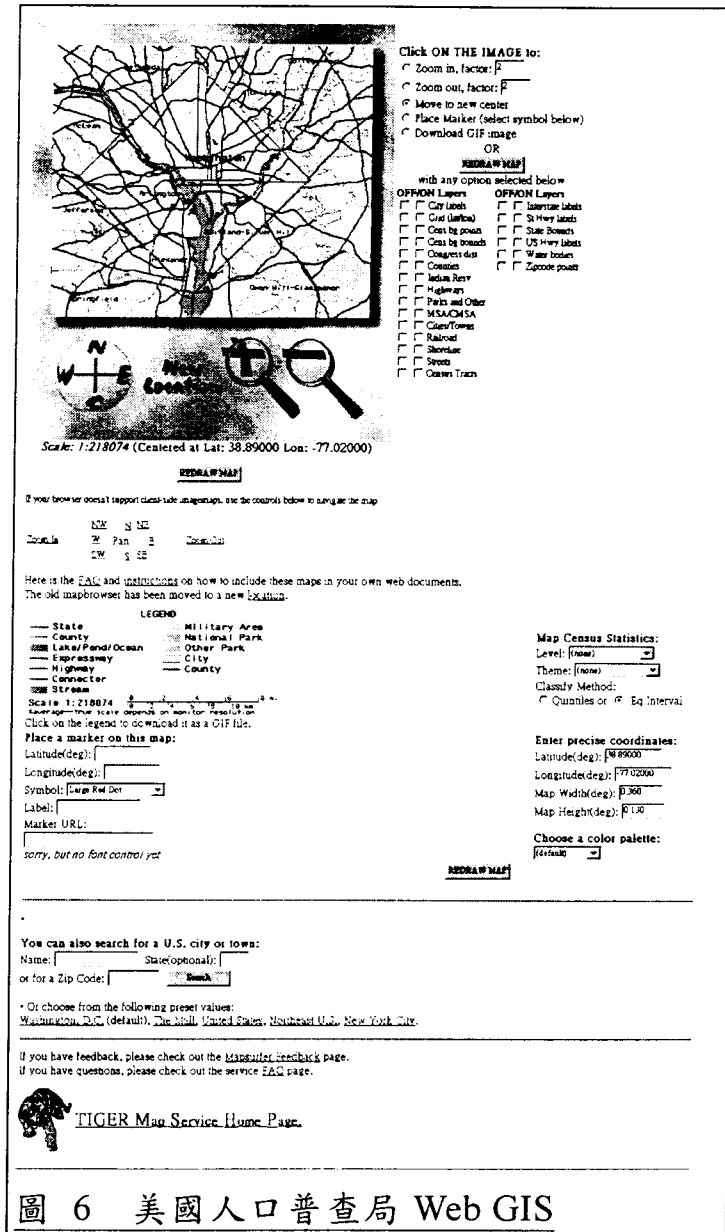
目前在國內外利用 Internet 的瀏覽器來開發地理資訊系統較著名的有：

美國人口普查局的 TMS(Tiger Mapping Service) [18]

這是一個由 1994 年 8 月開始的計畫，該計畫主要的目的是要將美國的普查統計資料與地圖資訊能夠以免費，高品質的地理資訊透過 Internet，讓一般使用者能夠很方便的取得(如圖 6)。目前該系統所能夠提供的交談式功能計有：1.能夠讓使用者以輸入經緯度或是地名的方式在螢幕上顯示出該經緯度或是地名的相對應區域的地圖。2.以固定的比例來放大地圖。3.上下左右的來平移地圖。4.能夠讓使用者利用選單來指定想在圖上顯示哪些圖層資訊。5.做一些簡易人口統計資料的顯示。6.可在地圖上加上一些標籤的記號並能連結到某個 URL 的位址。此外，該系統還提供了一套 CGI，可以讓其它的系統，在他們的 HTML 中只要給定所要顯示的地圖起始經緯度，圖幅所佔寬度與長度，所欲顯示圖層，所欲加上之標籤等參數就可經由 TMS 遠端將此圖產生。

◆ 使用技術

CGI、
JavaScript、
後端動態產生
圖形檔傳回。
(技術方面請
參考第二章)

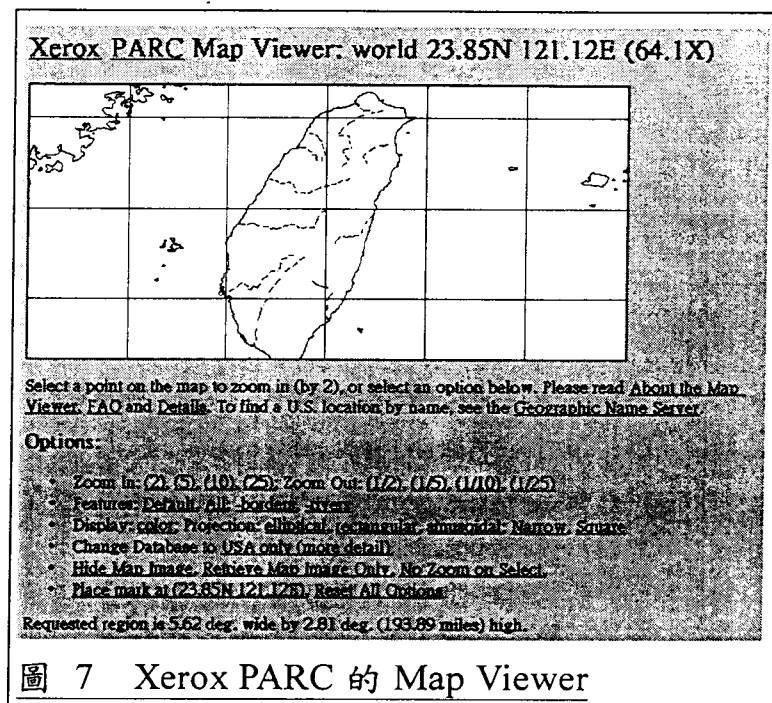


Xerox PARC (Palo Alto Research Center)的 Map Viewer [19]

此系統提供了類似 TMS(Tiger Mapping Service)功能的系統，不同的是全錄公司所提供的是全世界的地圖資料，但是其圖層只有邊界、河流二層(在美國區域多提供了公路、鐵路及聯邦政府區域三層)，範圍雖大但較不精確，功能也較少(如圖 7)。

◆ 使用技術

CGI、
JavaScript、
後端動態產生
圖形檔傳回。
(技術方面請
參考第二章)



美國加州柏克萊分校之環境設計學院人學的 REGIS

系統[21]

這些學校透過 REGIS 的地理資料庫配合上一公共所有(public domain)的 GIS 叫作 GRASS，發展出一套用瀏覽器使用 GRASS 的部份功能的地理分析工具稱為 GRASSLink。因為 GRASS 雖然是一套功能不輸一般商業軟體的 GIS，且可由網路上免費取得，但其為 UNIX 環境下的系統，因此使用者需有 UNIX 的帳號才有使用的權限。所以這些學校透過 Internet 結合該軟體可使一般使用者可利用瀏覽器的方式來使用 GRASS 的部份功能。

GRASSLinks 3.0 at REGIS

Maps of the San Francisco Bay and Delta Regions

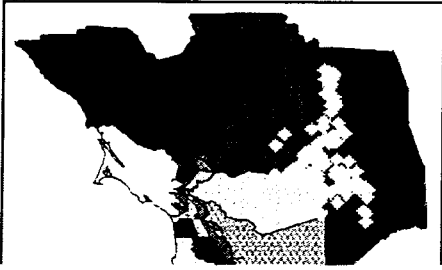
Map Display, Zoom & Query Form

Map Display Elements:

- Raster map: Counties
- Vector map: County Boundaries in black
- Sites map: Recreation Sites in yellow
- Initial display region: 12 County Bay and Delta Area
- Image Size: medium (400X400)

UTM coordinates of selected center point are: 597131E and 4229469N
Category information of base raster map is: Solano County

See bottom of display for query and display options



■ Napa County	■ Solano County
□ Contra Costa County	□ Alameda County
□ Santa Clara County	□ San Mateo County
■ San Francisco County	■ Yolo County
■ Sacramento County	■ San Joaquin County
■ Marin Bay	■ Sonoma Bay
■ Napa River	□ Solano Bay
□ Contra Costa Bay	■ Alameda Bay
■ Santa Clara Bay	■ San Mateo Bay
□ San Francisco Bay	

圖 9 足普度大學的 INGIS

GRASSLink 剛開始要選區域(Location)，與地圖集(Mapset)，其顯示地圖的方式有網格，向量，地點等三種資料集(Dataset)。使用者還可自行指定所要顯示圖層所用的顏色，接著也是由遠端立即自動作圖再送到使用者端來顯示(如圖 9)。該系統還提供了一些分析工具，例如可計算兩張地圖的重疊部份，將地圖的原始資料列出，對既有地圖做重分類(Reclass)，以歪斜的軸來顯示海灣或曲折的地

形，對地圖作緩衝區(Buffer)的計算，以及把這些計算的結果或地圖以 GRASS 的檔案格式下載到使用者端等功能。此系統的最大缺點是計算這些圖形所耗去的時間是前述系統的數十倍，主要原因是因為目前這些運算多是以網格的方式在計算，所以花的時間比向量轉成網格所花的時間更久。

◆ 使用技術

CGI、

JavaScript、

後端動態產生圖形檔傳回。

(技術方面請參考第二章)

SEDAC (Socioeconomic Data and Applications Center) 的 Data and Information Catalog Services[22]

SEDAC 為 CIESIN 多媒體搜尋資訊系統之子結點，在此供應豐富之人文社會的電子目錄資料。使用者可在所顯示之世界地圖上拉選一個矩形，形成搜尋地域(如圖 10)，再點選要找尋之資料目錄，最後輸入關鍵字用以找出相關資料。不過此系統只是傳回相關文字資料並不會傳回圖形，為一簡易以圖查文字之 GIS 系統。缺點：無法提供圖層疊合，傳回之資料為文字資料，Java 受限於 Netscape 3.0 版，無法以 Netscape 4.0、IE3.0 及 IE4.0 執行。

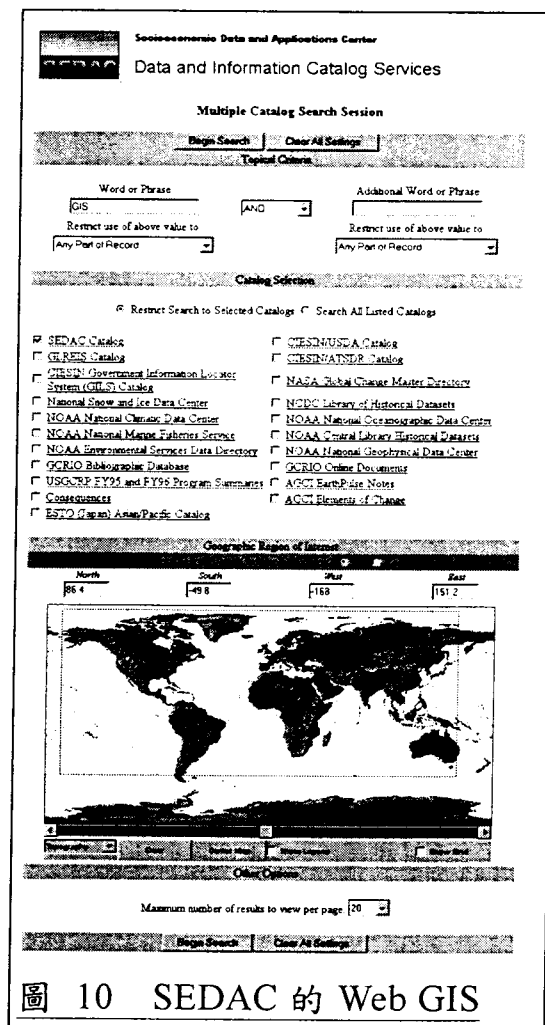


圖 10 SEDAC 的 Web GIS

◆ 使用技術

CGI、

Java。(技術方面請參考第二章)

中央大學太空及遙控中心[23]

運用 HTML 及 CGI 提供在 WWW 上對衛星影像之查詢及樣本影像之傳遞(如圖 11、圖 12)。

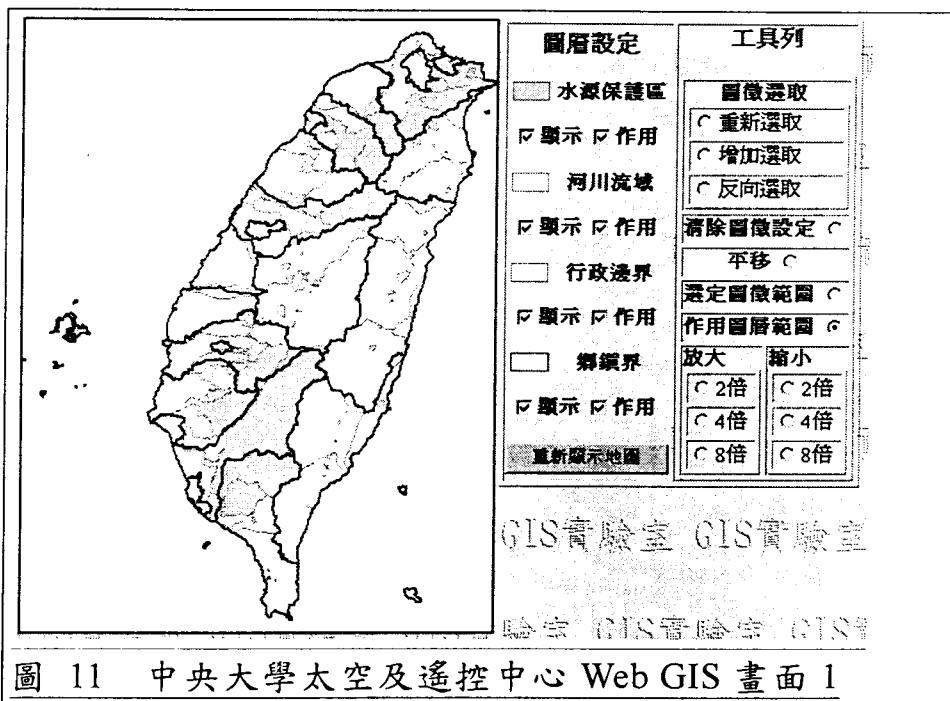


圖 11 中央大學太空及遙控中心 Web GIS 畫面 1

◆ 使用技術

CGI、JavaScript、Java、後端動態產生圖形檔傳回。

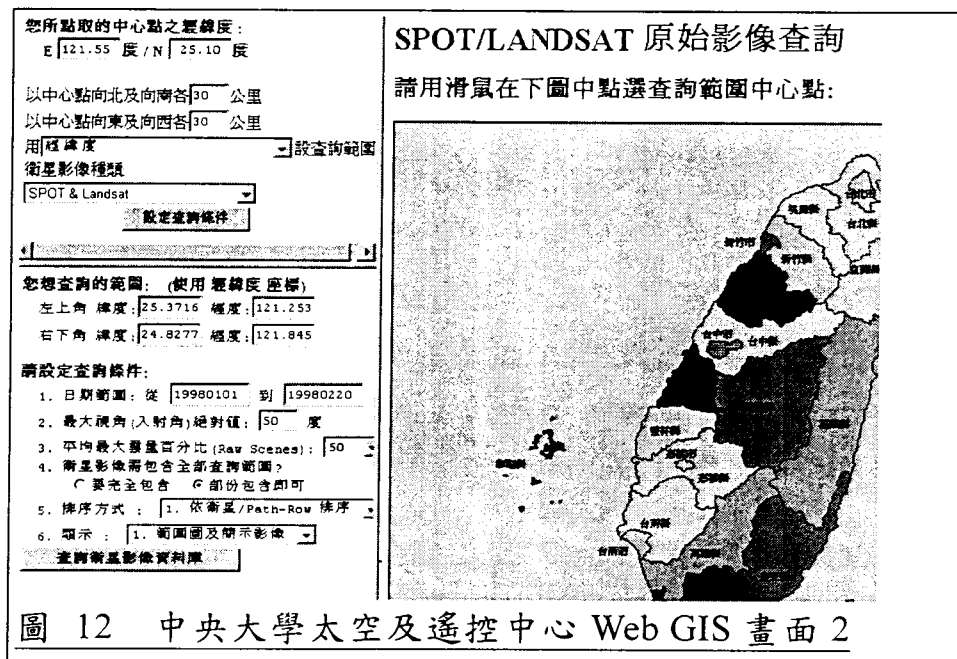


圖 12 中央大學太空及遙控中心 Web GIS 畫面 2

綜合以上分析，可瞭解以上網之 Web GIS 伺服器端仍是沿用傳統 GIS 而屬於 CGI 介面 Web GIS，其系統建構的方式是，在傳統單機的 GIS 系統上結合特殊設計的 Web Server，該 Web Server 的目的是利用 GIS 工具軟體將 GIS 資料與 HTML 結合一起。這類的 GIS 工具軟體及 Web Server 各大 GIS 系統廠商都有提供，例如 ARC/INFO、GRASSLink、MAPINFO 或 GENASYS... 等。而運作方式皆為讀取前端使用者端的參數，例如經緯度，所欲顯示圖層等資料，再用 CGI 之方式傳回到伺服器端，伺服器端利用現有的一些 GIS 工具軟體即時的在向量格式的地圖上搜尋到符合經緯度的位置，並加上圖層資訊後自動轉成網格格式的圖檔，傳回到使用者前端，所以需要較長的運算與轉換的時間。而且只要使用者前端做一點小變動此圖檔就要經過後端重新計算，且重傳至使用者前端，如此一來一往速度不僅緩慢，後端資料庫處理更是繁重，而且無法同時服務太多使用者，如上述系統 1,2,3,4,6。這種設計方式固然能較快將系統上網，但是 Web GIS 的缺點仍未解決，如快速得到地圖資訊、彈性的資料存取、遠程分散式資料更新與線上更新時同步處理及資料復原(Transaction Recover)... 等。

若改為 Multi-Tier 架構，使用者前端可以用 JAVA 語言或 ActiveX 物件 Plug-in 方式來整合大型 GIS 或融合小型之 GIS，GIS 亦可以直接驅動 WWW，在使用者前端直接做到資訊及圖像資訊上之更新、文件製作或文件連接、空間化文件之建立、空間資訊查詢與使用及資料之維護。WWW 則成為使用者使用 Web GIS 之唯一介面，以提供一般人使用 GIS 功能及資料。但此嵌入型 Web GIS 因技術較為新穎在市面上較為少見。且此類型之 Web GIS 通常不藉助大型 GIS 工具軟體所提供之功能，一切功能皆需自行撰寫，因此較為麻煩；但主控權對程式設計者而言較大，所以使用者介面較好控制，且可改善許多使用者介面及效率問題；但此類型 Web GIS 尚屬於啟蒙階段，所以功能皆不如傳統 GIS 強大，有很大之發展空間，如上述系統 5。上述各系統與本文發展系統(第 5、6 章)之功能比較表列於下面。

項目名稱	系統 1 TMS	系統 2 Map Viewer	系統 3 Yahoo Maps	系統 4 REGIS	系統 5 SEDAC	系統 6 中央大學 太空及遙 控中心	本系統
Web GIS 之分類 及介面(請參考第 四章)	CGI	CGI	CGI	CGI	Embedded Program	CGI	Multi-Tier Embedded Program
使用主技術	CGI	CGI	CGI	CGI	CGI Java	CGI Java	ActiveX DCOM
Web 設計時間	快	快	快	快	中	快	最慢
後端是否使用 GIS 工具軟體	是	是	是	是	否	是	否
地圖的產生	後端	後端	後端	後端	N/A	後端	前端
圖層疊合能力	可	可(少)	可(少)	可	無	可(少)	可
取得地圖資訊之 速度	慢	慢	慢	慢	N/A	慢	快
資料查詢之彈性	中	中	中	中	中	中	高
伺服器之負荷	重	重	重	重	較輕	重	輕
後端可支援分散 式處理	無法	無法	無法	無法	無法	無法	可以支援
同時可供民眾上 線之人數	少	少	少	少	多	少	多
遠程分散式資料 更新	無法	無法	無法	無法	無法	無法	可以
線上資料更新時 之交易 (Transaction)處理	無法	無法	無法	無法	無法	無法	可以

表 2 各種 Web GIS 功能比較表

1.4 關鍵性設計議題

以上文獻列示出理想的 Web GIS 應具備的功能，同時針對市面之 Web GIS 做比較及分析。本文在後面章節提出一個 Web GIS 的新架構並且實做之，發展此 Web GIS 的幾項關鍵性議題有：

Web GIS Server 負荷問題

Web GIS 因資料皆屬於多媒體，因資料量甚大故 Web GIS Server 的負載能力就形成重大的考量因素。本文將以 Multi-Tier 配合分散式處理來解決 Web GIS Server 負載能力不足之問題。

使用者環境設定問題(Thin Client)

對 Web GIS 而言 Server 資料的管理一定要透過資料庫。在此種 Client/Server 架構下，Client 端使用者欲存取 Web GIS Server 的資料庫必須具備驅動此 Server 資料庫之能力(也就是要事先安裝此資料庫的驅動程式在 Client 端)，此一架構理論上可行但實際運作卻會出問題。因為在 WWW 上使用者的人數為非固定的，所以不可能預知哪些人會上網，也不可能去各使用者家中安裝所需的驅動程式，而且驅動程式上的版權也會出問題。所以一些簡易型的 Web GIS 只好使用 CGI 或在不使用資料庫來管理後端 Server，如此一來就喪失 GIS 應有的強大功能。要 GIS 的功能完備，又要使用者環境不需事先經過設定，而達到 Thin Client 境界，本文提出以 Multi-Tier 架構來克服此問題。

Web GIS 資料遠程分散式維護問題

資料維護在 GIS 中是一項重要的課題。「GIS 資料佔整個 GIS 建構過程成本中約 65%~85%」(朱子豪 [1])，所以而降低 Web GIS 資料維護成本的方式就是要使 Web GIS 具有遠程分散式維護資料的能力。但目前的 Web GIS 都必須將資料集中至主電腦且透過專業人士做資料更新，造成 Web GIS 上的資料不僅時效性差，且維護困難。在本文提出以分散式物件配合 DCOM 通訊協定來解決 Web GIS 資料遠程分散式維護問題。

網路頻寬問題

一套 Web GIS 功能再怎麼強，其所需要傳遞之資料量無法減少，反而會隨著功能的提升而增加傳遞之資料量。如今電話線的最快傳輸率不過是 54Kbps，所以在 Web GIS 查詢一張圖往往要等上 30 分鐘以上。在此提出依 GIS 資料的變化率，做資料分布的最佳化，將不會變動的資料在使用者端電腦快取(Cache)起來，用以解決網路頻寬問題。

1.5 發展的步驟

發展網際地理資訊系統步驟可分為四大步驟，分述如下：

技術理論之資料搜集

主要有三方面：

(1)分散式物件(Distribution Object)的相關理論

目前分散式物件規格的制定幾乎以美國 OMG(Object Management Group)協會所制定 OMG Core Object Model 與 CORBA(Common Object Request Broker Architecture)理論為標準，其規範了分散式物件在網路的規則。而 Microsoft™ 的 ActiveX 即是在此規格中所發展的實作規格，所以要時時注意 OMG 動向及 Microsoft™ 在 ActiveX 之理論與實務上的用法。其資料收集以網站[15][16]為主。

(2)Multi-Tier 架構的相關理論

近年來由於 WWW 的崛起，使得 Client/Server 架構已不敷使用。配合分散式物件(Distribution Object)的使用勢必使用 Multi-Tier 之架構。其資料收集以網站[17]為主。

(3)空間資料庫的相關理論

因 Web GIS 會用到空間資料庫，因此就要考慮資料庫在儲存空間資料所遭遇到之問題與解決方法。有關 Web GIS 資料庫方面以 Maguire、Worboys[5][6][7]等人的研究及施保旭先生在地理資訊系統一書[2]的概念為主。

現有 Web GIS 的觀摩、使用及分析

了解國內外的 Web GIS 概況，且對其功能及類型做整理，逐一比較、分析及歸納。吸收其優點，改進其缺點。

系統設計與建構

當技術面問題解決後，就著手進行設計與建構系統。其工作有後端資料庫選擇、程式語言的挑選、空間資料庫的建構、模式庫的建立及內部與外部網路系統架構。利用網際網路之 WWW 與 AchveX 加以結合，並配合 GIS 的疊圖與屬性資料分析的功能，發展完整之 Web GIS 系統。

系統測試與未來規劃

測試本系統來找出程式臭蟲並加以修正外，並在省公路局的公路災害查報系統及新竹市政府網頁上實際測試，將系統執行時的狀況加以記錄、比較與分析，作為修正的依據。且持續加入一些 Web GIS 專業功能以供一般大眾使用。

第二章.

Web GIS 互動性技術 探討

現階段 Internet Web 上，製作 GIS 系統不是簡單的事。因 Web 上 HTML 語言的限制，沒有一般語言應具有之功能。HTML 只是一種文件表現格式，單純的把一份圖文並茂之文件透過 HTML 編碼傳給客戶端瀏覽器再加以還原成圖文原貌。所以客戶端使用者若要將某些資訊傳回伺服器，單靠 HTML 是做不到的，HTML 只能單純將文件從伺服器傳至客戶端，偏偏 GIS 又需要從客戶端讀取某些資訊以回應給伺服器，再交由伺服器做相對應之處理，以回應給客戶端。故發展 Web GIS 一定要藉助 WWW 上之互動性技術，用以輔助 HTML 製作 Web GIS。底下針對每一種互動性技術加以討論且分析優缺點：

2.1 CGI

CGI(Common Gateway Interface)是一個前端瀏覽器與 Web 伺服器溝通的標準介面，目前大部分 Web GIS 都是使用 CGI 達成互動式查詢之功能。大部分 Web 都是使用靜態的首頁，這些靜態的網頁即一個個固定的 HTML 檔。當使用者在此靜態之網頁輸入查詢資料時，瀏覽器會將此條件資料透過一個標準通道-CGI-傳給 WWW 伺服器，而伺服器一看到是從 CGI 傳來的條件資料，即將此參數透過環境變數傳給同一台機器上的特定處理程式，此程式接收參數且執行完畢後(如動態產生欲查詢之圖形檔或查詢後端的資料庫)即通知 WWW 伺服器，而 WWW 伺服器再以 HTML 的格式輸出到前端瀏覽器(如圖 13)。其優、缺點如下：

◆ 優點

- 穩定性不錯。
- 客戶端任何瀏覽器皆可執行。

◆ 缺點

- 不易處理由前端瀏覽器傳送過來的資料。
- 因為 CGI 是一支放在 Web 伺服器上的應用程式，所以當每一位使用者連線進來一次，此程式就會被執行一次，而在作業系統上占用一個 Process。當人數增加時會重複占用系統資源，不僅效率會降低，甚至會干擾到原有 Web 伺服器的運作。

- 嚴格的說來，CGI並不是一個真的互動式網際網路的方法，它只能單純的接受前端瀏覽器的命令，被動的處理事情。無法達到前端互動的境界，所以就無法在前端瀏覽器依使用者不同的反應做出不同互動的回應。

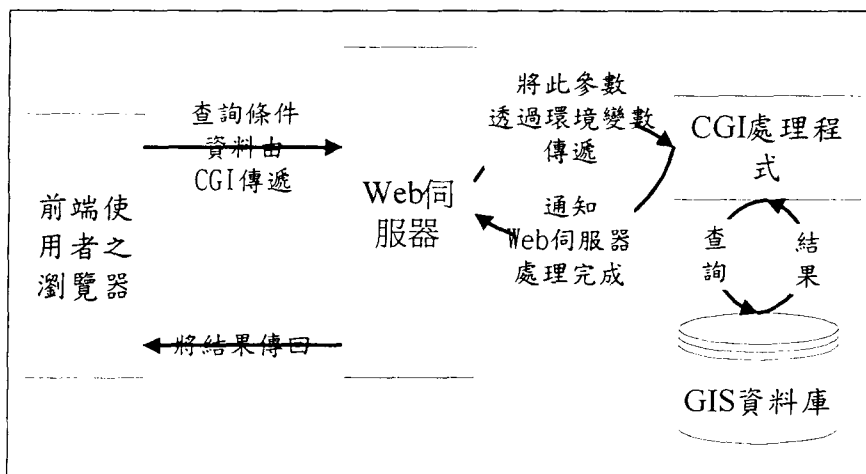


圖 13 CGI 之運作流程

2.2 ISAPI

ISAPI(Internet Server API)是 Microsoft™ WWW 伺服器(IIS⁵)所定義的一套 API。基本上為 CGI 的延伸。「啟動*.exe」這個動作非常耗時，如果同時許多前端瀏覽器要求，則 CGI 程式就會執行許多份，一般來說不能處理超過十人以上的連線；「經由標準輸出」，便無法傳遞大量資料給前端瀏覽器，功能上有所限制。所以，微軟便想出讓這些 CGI 程式的功能，變成一道道 API。且利用 DLL(Dynamic Link Library)取代可執行檔(*.EXE)，因為 DLL 具有和 Web 伺服器共用單一記憶體空間的特性，比起可執行檔會耗用較少的資源。相對的也減少 Web 伺服器的負擔。所以當前端有許多使用者同時上線而且同時執行一個伺服器上的 ISAPI 程式時 Web 伺服器將會自動產生多個執行緒(Threads)，來對應處理多個前端使用者的需求。這些執行緒實際上是共用一個行程(Process)，相較 CGI 應用執行程式每執行一次便在伺服器上起動一個 Process 的方式，執行效率上將出色許多，於是 CGI 的*.exe 程式就變成*.dll 程式。載入快、使用系統內定 API，當然與 WWW 伺服器、前端瀏覽器等傳訊都比較快、也可接受較多的連接。其優、缺點為：

◆ 優點：

- 改善了 CGI 中執行效率之問題，執行效率高。
- 穩定性不錯。
- 客戶端任何瀏覽器皆可執行。

⁵ IIS：Internet Information Server 的縮寫，為 Windows NT 4.0 Server 上之一套 Web 伺服器管理程式，其管理範圍包含 WWW、FTP 及 Gopher 等三種服務。

◆ 缺點：

- 不是一個真的互動式網際網路的方法，他只能單純的接受前端瀏覽器的命令，而被動的處理事情。無法達到前端互動的境界。

2.3 JavaScript 與 VBScript

JavaScript 與 VBScript：JavaScript、VBScript 分別是由 Netscape、Microsoft™ 所發展之巨集語言，基本上兩者的功能差不多，只不過 JavaScript 的語法類似 C 而 VBScript 的語法類似 Basic。基本上兩種都是內嵌在 HTML 的巨集語言，在此以討論 JavaScript 為主(因瀏覽器支援 JavaScript 較多)。JavaScript 的發展動機之一是為了在客戶端瀏覽器作邏輯的辨識，以免所有的工作都落在伺服器主機上，即使是簡單的資料驗證；那種模式就好像是當年被 PC 取代的終端機(主機架構)，終端機只負責輸入資料，之後的事都由主機處理。在客戶端瀏覽器提供邏輯的辨識不僅能增強客戶端的功能同時也更符合所謂的主從架構。自從 1995 年 12 月 JavaScript 推出之後，它獲得許多業界的主要大廠支援，包括 Apple™、Borland™、Sybase™、Informix™、Oracle™、Digital™、HP™ 和 IBM™ 等。且 JavaScript 本身是一個直譯式的巨集語言，它不像一般的程式語言必須先編譯成一個類似 .EXE 的執行檔才能執行，而是直接把程式加入 HTML 文件當中，由瀏覽器直接解譯執行。所以它是靠 HTML 來和使用者溝通的，大部份的 JavaScript 物件和 HTML 標籤關係密切，它提供許多事件(events)給 HTML 標籤來增強 HTML 的功能，並且允許事件驅動的程式在裡頭執行，是一個不錯的互動性工具，其中在 JavaScript Unleashed[11]一書有極詳細之介紹，表 3 是一個 JavaScript 內嵌在 HTML 檔案的例子。其優、缺點為：

◆ 優點：

- 提供了客戶端瀏覽器邏輯的辨識功能，以免所有的工作都落在伺服器機頭上，也避免因邏輯的辨識而需要將資料在 Internet 一來一往。不僅浪費客戶端等待之時間，也造成網路壅塞。
- 撰寫容易，不易出問題或造成客戶端機器當機。

◆ 缺點：

- 無法控制後端伺服器，所以資料庫的存取、動態製作地圖...等皆無法實作。
- 無法撰寫太複雜之程式，只能寫一些簡便之客戶端邏輯的辨識功能。
- 為直譯式語言，在每次執行時都要重新解譯一次，且所花的時間比較多。

```
<html>
<head>
<title>This is a Example</title>
<Script language="JavaScript">
<!--
    window.defaultStatus = "Welcome to the Web."
    Function changeStatus() {
        Window.status = "Click me to go to the GIS Home
Page."
    }
    ....
-->
</Script>
</head>
</html>
```

表 3 JavaScript 之範例

2.4 Java

Java 是由 SUN 公司以 C++ 語言及 OOP(Object Oriented Programming) 為基礎所發展的跨網路且無平台問題的共通語言。可以將程式嵌入 HTML 中(如表 4)，供瀏覽器下載後執行。如同 JavaScript 的發展動機之一是為了在客戶端瀏覽器作邏輯的辨識，以免所有的工作都落在伺服器主機上。不同的是 Java 是一個真正的網路程式語言，對於安全性做了極大的考量，使得在使用 Java 時，前端不會有安全性的疑慮。不過它需要較低階的程式設計技巧，而這些技巧是大部份想增強 HTML 表單功能的設計者所缺乏的，且 Java 是一編譯式之語言，必須先在伺服器端編譯成中間碼(P-Code)，再由瀏覽器下載後執行之。此外 JDK(Java Development Kit)在 1.02 版後也提供一個連接後端資料庫之介面 JDBC，以解決連接後端資料庫之問題，可說是一個功能強大之語言。以下列出優、缺點之比較且以 Java Applet(在瀏覽器執行之 Java 程式)為主。

```
<html>
<head>
<title>The Applet – example 1</title>
</head>
<body>

<applet    codebase="1.0.2/"    code=Animator.class
width=460 height=160>
<param    name=imagesource
value="1.0.2/images/Beans">
<param name=backgroundcolor value="0xc0c0c0">
<param name=soundsource value="1.0.2/audio">
<param name=soundtrack value=spacemusic.au>
```

```
<param name=pause value=200>
</applet>

</body>
</html>
```

表 4 Java 之範例

◆ 優點：

- 功能強大。
- 跨平台性高。
- 安全性夠。
- 是一個真正 WWW 前端互動性的技術。

◆ 缺點：

- 因是新語言，所有的程式都要重寫。
- 語言複雜，難以學習。
- Java 的虛擬機器(Virtual Machine)限制太大。Java 本身為功能強大之語言，其中還提供大量之 I/O 函式，但一旦寫成 Applet⁶這些函式皆無法使用。因 Java 的虛擬機器實作上(就是瀏覽器的實作)，對安全性的考量太嚴謹，幾乎重要的事情都無法完成，如讀取前端瀏覽器的任何資料或是開檔。因為如果沒有加以禁止的話，則網路上來的 Java 程式，甚至可能開啟前端電腦的檔案，然後把病毒寫入，這樣不是破壞 Java 的美名嗎！然而這種邏輯上的假設，與先天上的限制，使得許多程式設計師在設計 Java 程式時，總有綁手綁腳的感覺，甚至某些功能做不出來。這種考量固然考慮到安全性；但也犧牲語言本身應有之功能。

⁶ Applet：指 Java 的一種執行程式，專門用於瀏覽器上執行，可以嵌入於網頁中。但許多 Java 語言之 I/O 功能皆無法執行(凡是會壞到前端電腦之指令)。

- Java 為了要跨平台，所以 Java 的原始程式會先編譯與系統機器無關的中間碼(P-Code)，所以每當前端使用者連上 Internet 時，皆會先將中間碼傳到前端使用者的電腦，再即時編譯成機械碼(Machine Code)，但目前電腦在即時編譯成機械碼上效率相當差(甚至比 JavaScript 慢)，所以傳輸及執行時間都相對非常緩慢。
- 瀏覽器的支援混亂不統一，Java 的語言規格從 1.0，1.1，1.2(目前最新) 演進而來，每一次規格的演進都無法做到往下相容。也就是 1.1 版本的 Java 程式無法在支援 Java 1.2 版本的瀏覽器底下執行，若用支援 Java 1.2 版本的瀏覽器底下執行 1.1 版本的 Java 程式通常會出問題，甚至當機，如 SEDAC (Socioeconomic Data and Applications Center) 的 Data and Information Catalog Services[22]只能用 Netscape Navigator 3.0®觀看。
表 5 列出常用瀏覽器所支援之 Java 規格：

瀏覽器名稱及版本	所支援之 Java 規格
Microsoft Internet Explorer 3.0®	1.0
Microsoft Internet Explorer 4.0®	官司中不明確但目前為 1.1(沒有完全支援)
Netscape Navigator 3.0®	1.0
Netscape Communicator 4.0®	1.1(沒有完全支援)

表 5 常用瀏覽器所支援之 Java 規格

- 因 Java 是一個以 Client 端為重點的語言，對後端支援不容易，如要維護後端伺服器中的資料庫，就要透過如下流程

瀏覽器->Java->JDBC->ODBC->Web 伺服器->資料庫

其速度不僅慢上加慢，且 JDBC 的支援一直到 JDK 1.02 後才有，加上瀏覽器的支援又混亂無比(表 5)，所以讓使用者與程式發展者無所適從。

2.5 ActiveX

ActiveX 是 Microsoft™ 所提出之分散式物件規格(在第三章有詳細討論)，也是 OCX 加上網路功能之新技術(OCX+DCOM⁷)，使用 ActiveX 可以嵌入 HTML 中(如表 6)，供瀏覽器下載後執行。發展動機是為了在客戶端瀏覽器執行程式，以免所有的工作都落在伺服器主機頭上，不同的是 ActiveX 並不像 Java 是一個全新之東西，而是由 OLE 演變而來，故對大部分設計者而言並不需要重新學習，大大地減低設計者之負擔。而 ActiveX 有關分散式物件之理論是源於 CORBA(在下一節會有完整介紹)，故可以配合分散式物件之功能輕易發展出 Multi-Tier⁸之架構，為互動式網際網路之新一代技術，其優、缺點為：

◆ 優點：

- 功能強大，任何事皆可處理。不管是前端瀏覽器的互動反應或是要處理後端資料庫之問題。
- 因 ActiveX 是機械碼規格的統一，所以與程式語言完全無關，以前寫過之程式只要加以修改成符合 ActiveX 規格，即可重複使用。減少重新學習及重寫之困擾。

⁷ DCOM：就是 Distributed COM，是在 COM 之上多了三個要素：允許在遠端建立物件、能使用遠端方法的通訊協定即能安全存取物件之機能。DCOM 允許 Client 端在遠端建立並使用物件，對 Client 端而言，存取遠端的方式相對於本地端而言是沒有差別的。

⁸ Multi-Tier：是指在傳統 2-Tier(Client/Server)架構中再加上一層 Application Server 或是企業物件伺服器，來讓邏輯資料感應元件和 Business object 與 Client 端的應用程式分離。使整個

- 速度快。只要前端使用者連線過一次，ActiveX 元件會自動安裝在前端使用者的電腦中，下次連線時就不需要重新傳輸 ActiveX 元件。且 ActiveX 為機械碼不須重新編譯即可執行，故執行速度快。
- 可達到 Thin Client 端的境界，透過 ActiveX 結合 DCOM 規格形成 Multi-Tier 架構，前端使用者不需要加掛任何中介軟體(Middle Software)，即可執行任何互動式網際網路之功能。
- 充分支援分散式物件之規格。

◆ 缺點：

- 規格非常複雜，對於新使用者而言難以學習。
- 目前只有 Microsoft Internet Explorer 3.0®，Microsoft Internet Explorer 4.0® 瀏覽器支援，Netscape™ 的產品必須 Plug-In Third Party 的產品才能支援。

```

<html>
<body>
<h3> Web GIS </h3><center>
<object
  classid="clsid:04FF1043-8170-11D1-BB48-
0080C8583124"
  codebase="http://140.113.34.193/Scripts/Web
GISProj.ocx#version=1,0,1,7"
  width=750
  height=420
  align=center
>
</body>
</html>

```

表 6 ActiveX 之範例

系統的彈性與功能大為增加，同時減少原本 Client 端與 Server 端的負擔。

2.6 ASP

ASP：Microsoft™ 的 Active Server Page(ASP)是一種 Server 端的手稿環境(Server-Side Scripting Environment)，它能夠讓你產生動態、互動性的網站應用程式；這種手稿環境(或語言)是在 Server 端工作，並經由 Server 端的翻譯動態送出 HTML 文件給客戶端。因為 ASP 所送出的網頁是標準的 HTML 格式，不像一般 Script (JavaScript 或 VBScript)是在客戶端經由瀏覽器直譯，而產生各家瀏覽器不完全支援某種手稿語言造成執行錯誤的現象。其優、缺點為：

◆ 優點：

- 送出的網頁是標準的 HTML 格式，不會因而各家瀏覽器並不完全支援某種手稿語言而造成執行錯誤的現象。
- 非常容易就可造成動態網頁之效果，不必撰寫 CGI，困難度最低。

◆ 缺點：

- 屬於後端式之互動形態(如同 CGI，ISAPI)，對於前端互動性仍然需要其他技術加以支援，如 JavaScript、Java、ActiveX...等。

2.7 Web GIS 互動性技術比較表

比較完上述幾種技術後，可以發覺並沒有一種技術是全能的，有些偏向 Server 端之互動，有些偏向 Client 端之互動。有些很容易只要一般大眾就會使用，有些卻要專業之程式設計師才有辦法駕馭(表 7)。所以這些技術可以互相配合使用，才能發揮最大效益。

互動性技術	互動端	功能性	效率	使用複查度	普及率
CGI	Server	中	低	專業程式設計師	極高
ISAPI	Server	中	中	專業程式設計師	高
JavaScript / VBScript	Client	低	高	程式設計師	高
Java	Client	高	高	專業程式設計師	中
ActiveX	Client/ Server	極高	高	專業程式設計師	低
ASP	Server	低	中	程式設計師	中

表 7 各種 WWW 互動性技術之比較表

第三章.

Web 上分散式物件理 論分析

3.1 分散式物件之架構-OMA

OMG(Object Management Group)為目前分散式物件推行最力的單位，它制訂了物件管理架構之標準，認為理想物件的使用與設計應要達到「與作業平台無關、與程式語言無關、與網路通訊協定無關」。「與作業平台無關」，所以物件在各種作業系統上皆可通訊。「與程式語言無關」，所以物件可以運用任何一種語言寫成，但須遵循物件介面(interface)的規定。「與網路通訊無關」，所以物件可以在各種網路環境上通訊，而這事實上便是一個簡單的應用程式通訊協定(application protocol)，將電腦兩端的物件以各種格式，互相傳遞訊息。而形成如圖 14 之 OO 網路環境應。這便是著名的 OMA(Object Management Architecture)架構[12]，各部份功能解釋如下：

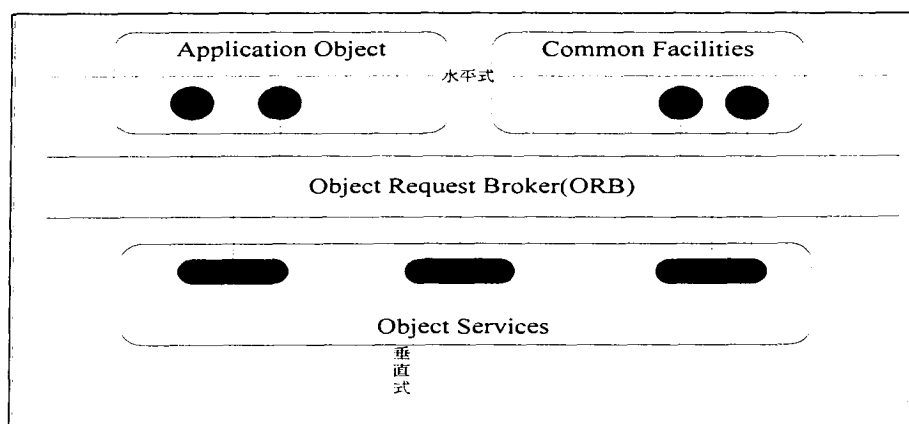


圖 14 OMA 架構圖[12]

1. Object Services(OS)

物件服務者，提供服務給各前(client)端應用程式物件。每一個服務都是一個物件，有自己的資料與函數，開放一個穩定的介面(interface) 供外界呼叫。

Common Facilities(CF)

屬於前(client)端的通用程式，裡面的物件幾乎是每個前(client)端應用程式物件所必須的。所以這些物件的功能較廣，可適用於各種應用程式。

Application Object(AO)

應用程式物件，為 client 端使用者直接操作的程式。

所以，以上三種物件形成網路物件的應用架構，每一個物件都有定義清楚的介面供其他物件呼叫。物件的資料部份透過網路通訊協定傳遞，物件的函數部份則透過遠端程序呼叫(Remote Procedure Call,RPC)執行之。而在 OMA 中制訂了兩個標準 Core Object Model 與 CORBA(Common Object Request Broker Architecture)共同規範物件在網路傳定的方式，其標準分述如下：

1. Core Object Model

指 OMA 中各個物件的表達方式，以較嚴謹的語法(虛擬碼)定義物件，詳細可參考資料 [12]，有詳細說明。簡而言之，Core Object Model 是以物件導向的角度來看 OMA 中的物件究竟該如何定義。如表 8：

Type Persion
Abstract
Supertypes::Object
Operations:
String social_security (P:Person)
String name(P:Person)
End Type
Type Employee
Supertypes:Persion
Operations:
Depaartment dept(E:Employee)
Money salary(E:Employee)
End Type

表 8 Core Object Model 範例

因為這些定義都是最基本的，幾乎每一個物件導向程式語言都可看到類似的結構。另外，在實際應用時，為了適應網路上各種作業平台與各種程式語言，許多廠商另以 IDL(Interface Definition Language)定義物件的介面，物件的實作細節則保留各程式語言的特性。

CORBA(Common Object Request Broker Architecture)

在 CORBA 中定義了 OMA 中 ORB(Object Request Broker)的運作方式，而 ORB 為一種應用程式的通訊協定，其呼叫之方式又可分為靜態與動態。如圖 15，指網路上的兩台電腦，當在前(Client)端的程式想要執行後(Server)端的函數時，首先在撰寫前(Client)端程式時，須明確指出使用哪一個後(Server)端介面中的哪一個函數，由介面儲存庫(Interface Repository⁹)調出該介面的資料(如函數原型...等)。在編譯前(Client)端程式時，則由編譯器產生網路通訊小模組(stub)附著在前(Client)端應用程式上。而後(Server)端的每個服務函數，因內含網路通訊小模組(稱做 skeleton)，因此在執行時期後(Server)端的 skeleton 可與前(Client)端的 stub 做網路溝通，彼此通訊透過軟體模擬的 ORB 橋樑(software bus)，這是 CORBA 中的靜態呼叫情形，前(Client)端應用程式已知要呼叫什麼樣的 Server 物件。

動態呼叫為當前(Client)端的程式想要執行後(Server)端的物件時，首先在撰寫前(Client)端程式時，須明確地指出使用哪一個介

⁹ Interface Repository：儲存物件中所有供使用者呼叫之介面(Interface)的資料庫，且包含了每一個介面(Interface)正確呼叫之格式及所提供之方法(Method)。

面(不需知道到底有哪幾個 server 物件實際實作此介面)，經由查詢介面儲存庫(Interface Repository)得出該介面的資料(如函數原型...等)。在編譯前(Client)端程式時，則由編譯器產生網路通訊小模組(stub)，附著在前(Client)端應用程式上。當訊息透過 ORB 傳到後(Server)端的 adaptor 模組時，adaptor 再決定到底該由哪一個 server 物件解譯此訊息，透過查詢介面儲存庫(Interface Repository)決定好物件後，再透過 skeleton 實際完成通訊。

靜態呼叫的好處是:簡單與快速，client 與 server 程式幾乎不用花多餘的時間就能完成溝通，如同傳統的主從架構應用程式，只是多了個「可以以函數呼叫來執行遠端物件而已」，動態呼叫的好處是:較具彈性，client 應用程式可在不知道到底會由哪一個 server 物件完成此通訊的情況之下發出訊息，透過 ORB 的轉換處理，自然會有對應的 server 物件負責解譯此訊息。缺點是，速度慢與較複雜。

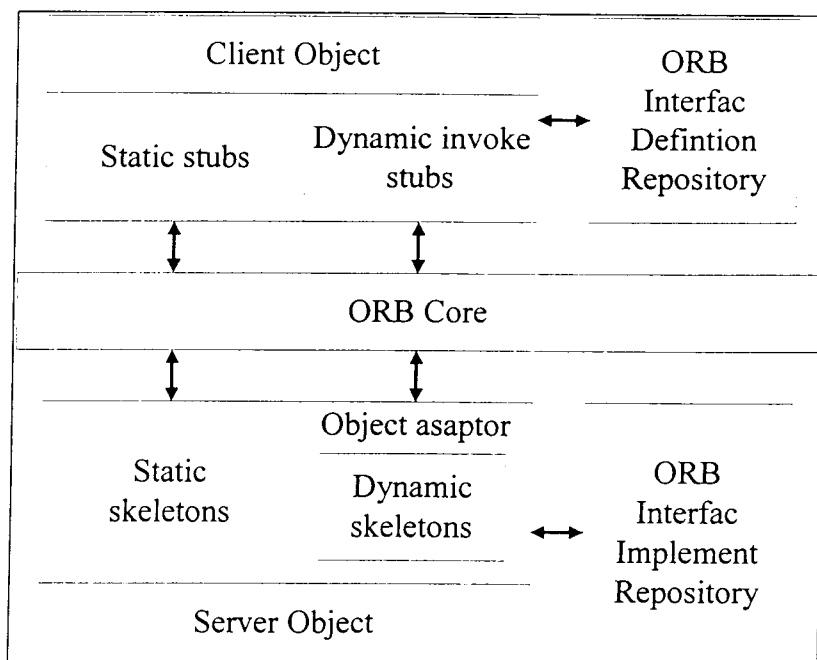


圖 15 COBRA 物件呼叫架構圖

目前 ORB 機制做得較成功、應用得較廣泛約有三個產品：Hewlett-Packard's Distributed Smalltalk、IBM's Distributed SOM Object、DEC's ObjectBroker。HP 與 IBM 的標準較接近 CORBA，DEC's ObjectBroker 則正邁向 CORBA 標準，但仍有距離。而微軟的 OLE COM(ActiveX 的前身)架構正是架在 DEC's ObjectBroker 上，造成日後應用 CORBA 架構最多的是走 DEC 與微軟路線，非正統 CORBA(儘管微軟聲稱 OLE COM 一定會符合 CORBA 之標準)。

3.2 分散式物件之使用者操作介面

基本的通訊核心定義完成後(ORB)，就要制訂一套操作與使用的介面，以便使用此分散式物件，此一介面即為上層之 GUI，一旦把 ORB 用 GUI 包封起來就形成所謂的組裝軟體的概念(在[8]中對於組裝軟體有詳細之討論)。組裝軟體的意義在於：原來沒有某功能的程式，經過一些手續後，在不影響原程式的情況下，原來的程式進化成擁有新的功能，而且是動態的進化(指不經過編譯器重新編譯此程式的原始碼)，新的功能無痕地併入原程式中，這就是組裝軟體的概念：軟體程式可以由小組件組裝起來，形成更完整的應用程式。架構則是透過分散式物件在做的，需要哪一個新功能就在電腦上(或網路上)找尋是否有提供此服務之物件，如果有就組裝進來。如同第一次裝 WORD7.0 時，常常發現沒有像 WORD6.0 一樣擁有「方程式編輯器」，但只要在進階選項選擇安裝「方程式編輯器」，然後再次執行 WORD7.0，「方程式編輯器」就會出現了。但在上層 GUI 卻有兩個標準。由於 OMG 所規定且以 IBM、Apple 主導的 OpenDoc(開放式文件架構)，及微軟在 Windows 上的 OLE COM(ActiveX 的前身)架構。其分述如下：

1. OpenDoc 架構

OpenDoc 標準是由 CI Lab(Components Integration Labs)開放協會所制訂，成立於 1993 年 9 月，主要成員有：Apple、IBM、Taligent、Adobe、Lotus、Novell、Oracle、Xerox 等。OpenDoc 在 1996 年 4 月左右成為 OMG 協會定義的 CORBA 標準之一，成為 Common Facility 的一員，可以說是將來 OO 系統或應用程式處理文件時，

最有可能的標準通用協定。整個 OpenDoc 可以分為三大部份。

(1)Document、(2)Parts、(3)Component。簡介如下：

(1)Document

就是一個元件容器(Container)，在 Document 上面可以插入任何 Parts，用以管理及使用被插入的所有 Parts，如 Word 文件檔中可以插入 WORD Picture 編輯圖形資料，HTML 檔案也可嵌入許多元件。

(2)Parts

Parts 就是前面所說 Document 內被插入的東西。所以一個 Document，可以插入許多 Parts，如「編輯文字的文件應用程式」，本身預設的 Parts (OpenDoc 稱為 Root Part)為文字，使得在沒有插入任何 Parts 的情況下，「編輯文字的文件應用程式」就可以編輯文字 Parts，插入其它圖形元件後(附屬在根元件下)，該文件便具有多種資料格式。在 OpenDoc 裡，每一個 Parts 可以插入任何 Parts 裡面，也可以被任何 Parts 插入，但是每一個 Document 只有一個 Root Parts，意即 Root Parts 只能被別的 Parts 插入，不能插入別的 Parts。所以整個 Document 成為一個類似「樹」狀的資料結構。

(3)Component

Component 為處理 Parts 的程式(這點與 OLE 名詞定義不同)，包括能讓 Parts 被修改的程式、能讓 Parts 只被看但不能改的程式、能讓 Parts 具有額外服務或是可程式化之能力的程式。您可將 Parts 想像成物件導向理論中的物件，物件內的資料成員存放在 Parts 裡面，Component 則為物件內的成員函式。假如插入一個圖形元件，則對應的 Component 則為「畫點」、「畫線」、

「儲存」、「觀察」等的函數。更進一步的說，以後如果我們想擴充 Document 應用程式的資料處理能力，只要購買新的 Component 就夠了。

OLE COM 架構(也就是 ActiveX 的前身)

OLE 原名為物件連結與嵌入(Object Linking and Embedding)，1.0 版時只提供複合文件處理的功能，在 2.0 版時加入 COM 架構，以便整合日益複雜的 OLE 軟體元件，用更物件導向的方式來操作所有 OLE 功能。演變至今，泛指微軟的物件導向技術-經過相當優秀之物件導向設計後的軟體元件，具有良好的可擴充性、統一的物件環境。其詳細之規格請參考[14]。

OpenDoc 有幾項優點是微軟 OLE 目前所達不到的，不過微軟在[13]則不客氣地指出 OpenDoc 到現在根本沒有實際大型的應用程式，如同紙上談兵，而 OLE 選擇 Office 等大量實際好用的軟體做為支持。不過，一般認為 OpenDoc 較優秀的地方有：

(1)不規則形狀的元件

OLE Document 或是 OCX 必定是矩形，OpenDoc 的元件則可以不規則形，所以我們可以在一份文件中插入一個三角形的圖形，三角形兩邊空白的地方仍然可以寫文字、排版，不像 OLE 一定是矩形，有浪費空間之嫌。如 PageMaker5.0 就具有此能力。

(2)一次可以啟動數個元件的服務程式

也就是啟動數個 Component。OLE 一次只能處理一個程式，在 Word 中不可能同時啟動 Excel 與 PowerPoint。但在 OpenDoc 則可以，不過相當浪費 CPU 資源，因為同時有數個程式在執行，而且許多人懷疑這項優點的必要性。

目前 OLE 與 OpenDoc 的 OO 理論爭論，在國外期刊上隨處可見，但不論誰輸誰贏，可以預見的是分散式物件理論將更進一步深入到你我平常使用的軟體，使得軟體功能得以多樣化、操作得以簡單化。

3.3 Windows Web 上的分散式物件 - ActiveX

ActiveX 就是 OLE COM(又稱 OCX)加上網路功能(DCOM 的關係)，所以在沒有綁住任何程式語言下，我們可用任何一種發展工具(如 VC++、Delphi 等開發工具)來開發 ActiveX 物件。目前，DEC 與 SoftwareAG 正在把 ActiveX 架構移植到『UNIX 作業平台上(微軟保證 Linux 也會)，而麥金塔與 PC 平台則由微軟自己做，所以 ActiveX 也像 OpenDoc 可以跨平台。整個 ActiveX 有下列幾個部分：ActiveX Control、Document、Script-Service、Security。

1. ActiveX Control

ActiveX Control 可以插入 HTML、OLE-Document 或 VC、Delphi 及 VB 應用程式中。ActiveX Control 可以想像為一個 Java Applet，只是它可以由 VC、Delphi 或 VB... 等所開發，編譯好後就像是一個 Applet 的 bytecode 一樣，以二進碼的方式在網路上傳遞，到了瀏覽器的電腦時，則由瀏覽器呼叫 ActiveX 解譯器執行此 ActiveX Control。

2. Document

Document 則是因為 Activex 出現所定義的一種新文件型態。觀看此 Document 的 Viewer 就好比是 Java 的 Applet Viewer 程式，可

以看一個 HTML 定義的 Applet 格式。Document Viewer 則可以利用 HTML 中 Object 格式(微軟自行定義)觀看此 ActiveX 執行情形，MSIE 3.0 以上之瀏覽器就是一個 Document Viewer。

3. Script-Service

微軟在 Office 系列產品定義了 VBA，使得這些應用程式除了使用巨集外，也可使用 VBA 使應用程式具有程式控制的能力。這其實就是 OLE Automation 的功能。現在微軟制訂 VBScript，使它們可以用 VBScript 操作 ActiveX 物件，讓 ActiveX 形成一個個小小的軟體組件，可用 VBScript 組織起來完成大工作。

4. Security

以往微軟的產品，除了 NT 比較注意安全問題外，其他的都不太注意。這項缺點，正是 Java 大力強調的「Java 擁有絕對安全的保護」特色。現在，微軟推出 ActiveX，在 ActiveX 中提出所謂的 Private Communication Technology (PCT) 技術，使得 ActiveX 安全性大增，破除以往的不良形象。

ActiveX 技術所帶來的好處在章節 2.5 已有詳細之說明，但也因為它的功能太過強大，幾乎所有的事情皆可以處理。為防止有人做出 ActiveX 型病毒或是運用 ActiveX 竊取 Client 端電腦的隱私權，所以在安全性上是使用一種安全認證(Authenticate)的方法-「軟體公司的認證」制。凡是開發 ActiveX 程式的軟體商，經過認定後(如確定此公司是正當經營)，便給一個特殊的編號(私有鍵，private

key)，以後這家公司開發 ActiveX 程式時，會先將程式產生對應的雜湊表，再用此私有鍵對雜湊表編碼，編碼後的雜湊表與原 ActiveX 程式放在一起以供使用者下載。使用者則定期(如連上微軟的網路)取得公開鍵(Public key)，然後如果下載某家公司的 ActiveX 程式時，瀏覽器先對原程式產生雜湊表；並將下載「編碼的雜湊表」用公開鍵解碼，比較這兩個雜湊表，如果一樣，就表示沒有病毒。如圖 16。

且使用雜湊表技術，使得軟體在網路上傳送時，不肖人士如果竄改程式，則竄改後程式的雜湊表一定與「由公開鍵解碼所得的雜湊表」不一樣，這種防止動作看來比較合理，而且有強大的安全性理論在背後支持。

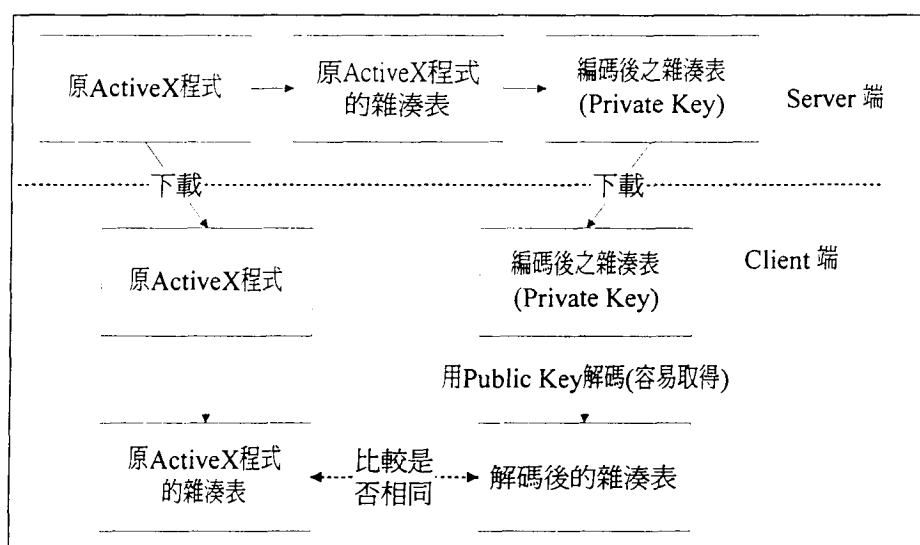


圖 16 ActiveX 的驗證(Authenticate)機制圖

第四章.

Web GIS 介面分析

現今 Web GIS 介面可分 CGI 介面 Web GIS 及 Embedded Program 介面 Web GIS 二類。CGI 介面 Web GIS 則是透過 CGI(共通閘道) 與傳統 GIS 連結，使得舊有系統可快速轉移到網際網路上。Embedded Program 介面 Web GIS 更直接將系統部分的程式移至客戶端執行，以提高系統效率。本章將詳細說明上述二種方式，且提出一種新模式 Multi-Tier Embedded Program 介面 Web GIS 用以克服 Web GIS 介面之缺點。

4.1 CGI 介面 Web GIS

CGI 介面 Web GIS 就是透過 CGI(Common Gateway Interface)來達成互動式 Web GIS(原理請參照章節 2.1)，其運作方式皆為讀取前端使用者的參數，例如經緯度，所欲顯示圖層等資料，再用 CGI 之方式傳回到伺服器端，伺服器端利用現有的一些 GIS 工具軟體，即時的在向量格式的地圖上搜尋符合經緯度的位置，並加上圖層資訊後轉成網格格式的圖檔，傳回到使用者前端，所以需要較長的運算與轉換的時間。示意圖如下(圖 17)。

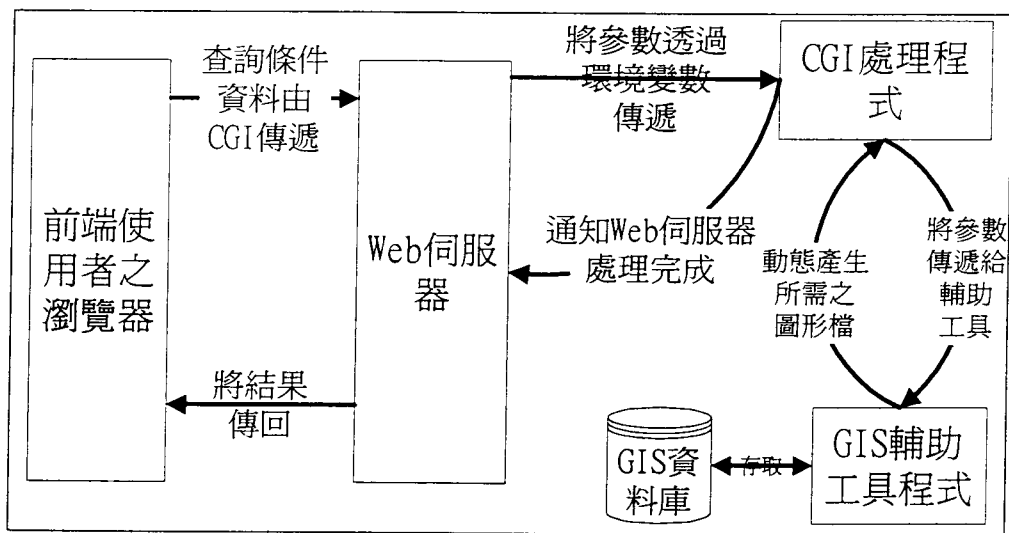


圖 17 CGI 介面 Web GIS 之運作流程

上圖的 GIS 輔助工具程式通常為 GRASS、GENASYS、MAPINFO 及 ARC/INFO... 等。以下概述各個工具軟體：

1. GRASS

GRASS(Geographic Resources Analysis Support System)為免費公益軟體，且為研發機構常用，GRASS尚有一大優點是其功能可以單獨抽出，直接與 WWW CGI 連接。而其以網格為主之處理功能，與 WWW 圖型之形式相吻合。詳細資料可以參考[24]。

ARC-INFO

相當有名的 GIS 軟體工具，為許多科研單位及自然資源管理單位常用之 GIS 軟體，工具相當完整，支援 Client/Server 架構，同時也提供 Web 服務，到目前為止是一套相當成熟的 GIS 發展工具。架構完整，也有許多支援的廠商，提供相當多的工具，可快速發展 GIS 的軟體。佔有率高，圖檔流通率高，PC 版本在 DOS 環境下操作較容易。詳細資料可以參考[25]。

GENASYS

為一套功能相當強之 GIS 軟體工具，圖形自動化編修程度較高，網格圖形處理能力較強，分析模組功能較強，座標轉換方式多而完整；但一定要在 UNIX 工作環境使用，且圖檔流通率太低，指令繁複，使人不易親近。詳細資料可以參考[26]。

MAPINFO

為一套較簡易之 GIS 軟體工具，用於 Windows 工作平台，系統開放，學習容易；基本功能完備但不強大，網格分析功能尚缺，數化較不方便。詳細資料可以參考[27]。

4.2 Embedded Program 介面 Web GIS

CGI 介面 Web GIS 雖然解決了互動式 Web GIS 無法對地圖上之地理資料做分析與處理的缺點，卻也失去了互動式 Web GIS 所擁有的高效率特性。由於瀏覽器本身的限制，CGI 介面 Web GIS 僅能提供點(Click)查詢的功能，而且只要使用者前端做一點小變動，此圖檔就要經過後端重新計算且重傳至使用者前端，如此一來一往速度不僅緩慢，後端資料庫處理更是繁重且無法同時服務太多使用者。這種設計方式固然能較快將 Web GIS 建構起來，但是 Web GIS 的缺點仍未解決，如快速得到地圖資訊、彈性的資料存取...等。而 Embedded Program 介面 Web GIS 則可解決大部分問題。Embedded Program 介面 Web GIS 可以用 ActiveX、Java 及 Plug-Ins 等方式(原理請參照章節 2.4、2.5)來實作，以突破瀏覽器的許多先天限制，而達到前端互動式之 Web GIS，展現出地理資訊系統的原貌，但相對的程式設計技巧較高，研發時間成本也較大。示意圖如下(圖 18)。

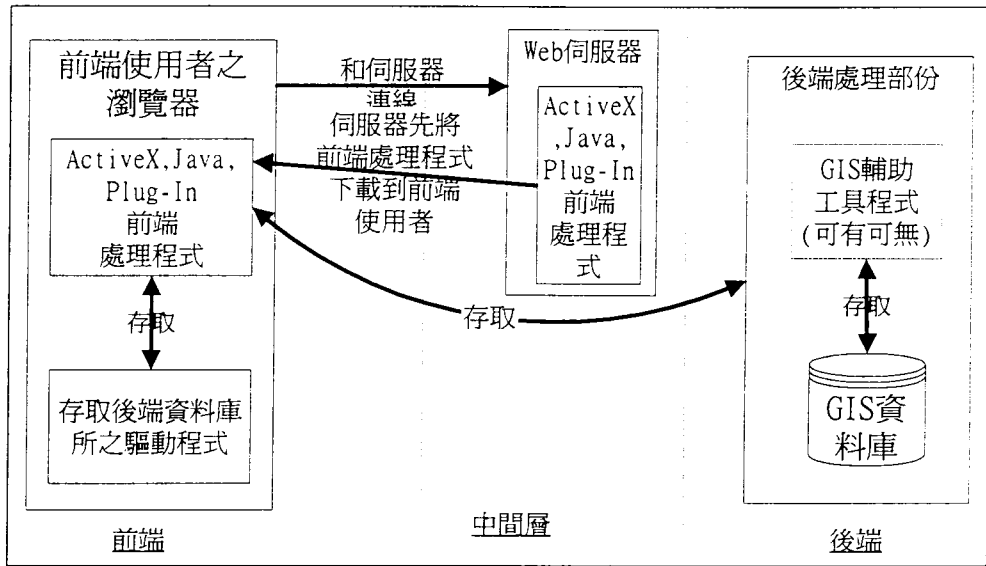


圖 18 Embedded Program 介面 Web GIS 之運作流程

4.3 Multi-Tier Embedded Program 介面 Web GIS

Embedded Program 介面 Web GIS 雖然解決大部分互動式 Web GIS 的問題，但僅只於前端互動性之境界。對於後端資料庫之控制卻是一片空白，這種顧前不顧後的做法並沒有完整解決 Web GIS 的問題，如此會造成 Fat-Client，形成前端電腦負責 Web GIS 大部分的工作，而後端卻單純且不聰明的供應資料。如此在資料量需求龐大的情況下，後端通常會忙不過來，且必須在前端電腦上具有驅動後端資料庫之驅動程式，如此在 Internet 上要做到分散式遠程資料更新與線上更新時的同步處理以及資料復原(Transaction Recover)...等便不可行，因為不是每一個前端使用者的電腦上都具有驅動後端資料庫之驅動程式。而在本文中提出以 Multi-Tier 之架構融入 Embedded Program 介面 Web GIS 擴展成 Multi-Tier Embedded Program 介面 Web GIS。其做法為將嵌入程式拆為兩部份，一部份放在端瀏覽器，一部份放在中間層，用以解決上述之問題。Multi-Tier Embedded Program 介面 Web GIS 原理上可以用 ActiveX、Java 兩種方式(原理請參照章節 2.4、2.5)來實作，但 Java 尚處於試驗階段，所以只能紙上談兵。因此目前為止只能選用 ActiveX 實作，以完全突破瀏覽器的限制，而達到前、後端互動式之 Web GIS，完全展現出 GIS 的原貌，但相對的程式設計技巧不僅更高，研發時間成本更大。示意圖如下(圖 19)。

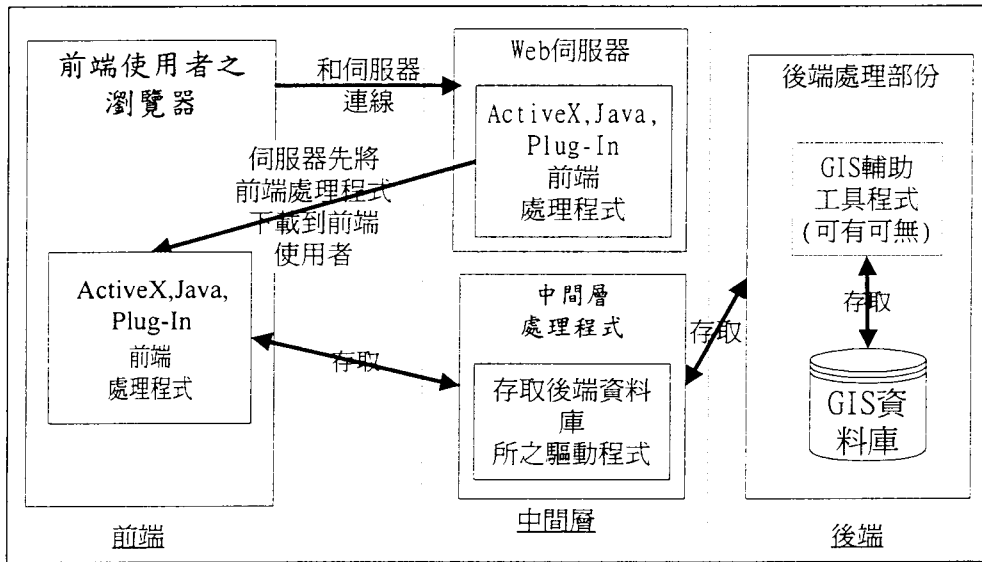


圖 19 Multi-Tier Embedded Program 介面 Web GIS 之運作流程

4.4 Web GIS 介面比較表

Web GIS 類型	優點	缺點
CGI 介面 Web GIS	<ul style="list-style-type: none"> ➢ 可與傳統 GIS 結合 ➢ 簡單、發展速度快 ➢ 相容性高、問題少 ➢ 具有後端互動性 	<ul style="list-style-type: none"> ➢ 速度慢 ➢ 不具前端互動性 ➢ 查詢無彈性，僅能點查詢 ➢ 效率低，隨使用者增加而嚴重遞減
Embedded Program 介面 Web GIS	<ul style="list-style-type: none"> ➢ 速度可 ➢ 查詢方式有彈性 ➢ 效率高 ➢ 具有前端互動性 	<ul style="list-style-type: none"> ➢ 不具後端互動性 ➢ 當使用者增加至一定程度時，後端資料庫會負擔不了 ➢ 具有前端相容性問題 ➢ 前端電腦上必須具有驅動後端資料庫之驅動程式
Multi-Tier Embedded Program 介面 Web GIS	<ul style="list-style-type: none"> ➢ 速度優 ➢ 查詢方式有彈性 ➢ 效率最高 ➢ 具有前、後端互動性 	<ul style="list-style-type: none"> ➢ 架構複雜 ➢ 具有前端相容性問題 ➢ 目前可使用技術只有 ActiveX

表 9 各種 Web GIS 介面比較表

第五章.

Web GIS 設計課題與 模式

在本章主要探討設計 Web GIS 時所遇到之問題，而提出解決方式。所引用方法之原始理論在第二、三、四章均有詳細說明。運用 Multi-Tier Embedded Program 介面 Web GIS，提出三種新一代 Web GIS 整體架構模式，一一圖解並且加以說明，同時闡明適用的環境及使用上之優缺點。經評估後本文 Web GIS 的設計模式採用模式 2，最後列出此系統所需之程式清單及其功能。

5.1 Web GIS 設計課題

5.1.1 Web GIS 伺服器負荷之問題

在 WWW 上之網站皆有一個特性，就是尖峰離峰的狀況非常明顯。平時或許少人上網；一旦有事時，可能同時湧入大量的人上此網站查詢資料。本文所做的計畫「公路災害查報系統」正是最好的寫照，平常一個人都沒有；但是一旦颱風過境，就會有許多民眾蜂擁而至，造成網站伺服器無法負荷。對一套 Web GIS 而言，其資料皆是以多媒體方式呈現，資料量之大更非一般文字資料所可比擬。此時後端伺服器的負載能力就形成重要的考量因素。

為了提高後端伺服器的負載能力，最常用之方式就是加強後端伺服器的計算能力及容量。如將一部 Pentium 伺服器升級成 Dual Pentium Pro 伺服器，或許可以將服務人數從 100 人提升為 200 人，但一旦人數超過 200 人時還是要換一次，若有一天同時湧入十萬人，那就算是美國 NASA 的超級電腦也無法一次處理那麼多使用者，所以這不是一個徹底解決之方式。

傳統上瀏覽器是直接連上 Web 伺服器(如圖 20)，所以一台 Web 伺服器所能服務的使用者有限，於是在本文提出分散式計算的架構來處理此問題，其架構是建立於 Multi-Tier 下(如圖 21)，流程如下：

- (1)一開始使用者之瀏覽器連上 Business Broker Server(可兼做 Web Server)。

- (2) 此 Business Broker Server 即馬上檢查所有的 Application Server，決定哪一台 Application Server 負載最輕及相對於處理使用者之工作而言是最佳化的(如距離最近...等)。
- (3) 將此使用者之工作交給此 Application Server。於是使用者就透過此 Application Server 與後端 Database Server 連線。

把原本一台電腦的工作，利用一台 Business Broker Server 做裁決，使用一些最佳化方式，將使用者的工作公平分發到各個 Application Server，不僅能輕易解決此一問題，又可降低成本(1000 台 Pentium 電腦的效能應不輸一台超級電腦，但其成本卻是遠低於超級電腦)。而此一架構則是利用 Multi-Tier 而達成分散式計算與處理之能力。

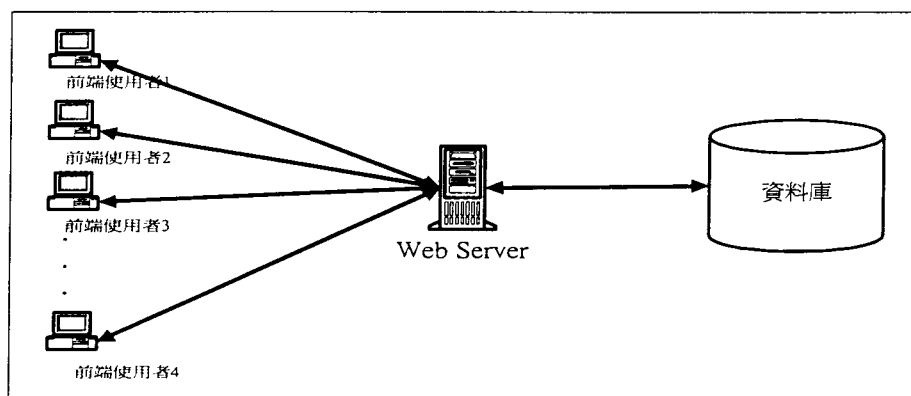


圖 20 一般 Web 之架構

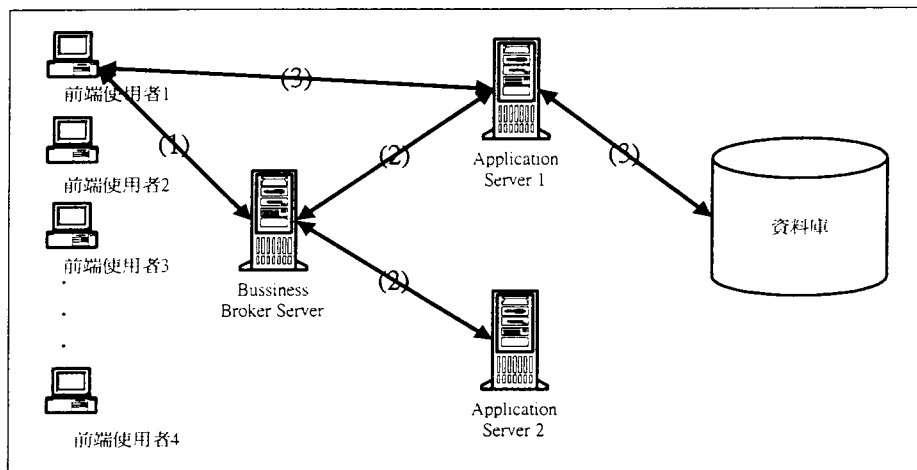


圖 21 Multi-Tier 架構之 Web

5.1.2 使用者環境設定問題(Thin Client)

Client/Server 的架構已行之有年，不論在理論或是實務上也證明它確實比傳統 Host 模式有效率。但在 WWW 上此一架構在理論上可行，實際運作卻會出現問題。因為對一套 Web GIS 而言，一定會用到 SQL Base 之資料庫，而此類資料庫也一定是 Client/Server 架構。對 Client 端使用者而言，欲存取此後端資料庫必須具備驅動此後端資料庫之能力，也就是要事先安裝此資料庫的 Client 端驅動程式，其架構如圖 18。也就是說在 Client 端必先具備資料庫之驅動程式，其 Client 端的處理程式才能存取後端資料庫。在企業網路內(Intranet)此方式可行，因為 Client 端使用者就是公司內部員工。所以 Client 端使用者人數固定，且在同一公司內，公司 MIS 部門可以事先派專業人員去各個 Client 端使用者的電腦，把所需要之驅動程式安裝且設定好；但 Internet 就不行了，因為不能事先預知哪些人會上網，就算知道也不能跑去使用者家中安裝所需要的驅動程式，所以對 Embedded Program 界面之 Web GIS 而言，使用者環境設定是一個大問題。

在本文提出以 Multi-Tier 加入 DCOM 技術用以解決此問題，如圖 19。其做法是將嵌入程式拆為兩部份，一部份放在瀏覽器端稱為前端處理程式(後稱程式 A)，一部份放在中間層稱為中間層處理程式(後稱程式 B)，如此程式 A 只要把需求告訴程式 B，而程式 B 再實際驅動後端資料庫，等到程式 B 取得資料後，再使用標準的通訊協定傳給程式 A。如此 Client 端就不用事先安裝後端資料庫驅動程式，程式 A 一樣能夠存取後端之資料庫。在此程式 A、B 均為分散式物件，而分散式物件在 Windows 平台上的通訊協定式是用 DCOM，因為程式 A 與程式 B 是被分置於兩個端點，所以就形成 Multi-Tier 架構。

圖 18 中的 Client 端稱為 Fat-Client，原因是在 Client 端必須事先安裝一些驅動程式且做一些設定，導致 Client 端程式相當龐大。而圖 19 中的 Client 端稱為 Thin-Client，用意就是不需要加掛任何驅動程式及設定，只要簡簡單單的一支程式即可。此一方式不僅可用於 Internet，在 Intranet 也同樣適用，因為一旦 Client 端資料庫驅動程式需要更新時，如圖 18 的情形，就要更新每一台 Client 端電腦；而圖 19 只要更新中間層的電腦即可。所以使用 Multi-Tier 的架構對於 MIS 人員而言更易於維護及管理。

5.1.3 Web GIS 資料維護問題

資料維護在 GIS 中是一項重要的課題。「GIS 資料佔整個 GIS 建構過程成本中約 65%~85%」(朱子豪 [1])。目前 Web GIS 在資料上幾乎是處於被動式之更新方式，不僅加重了 Web GIS 在資料成本的負擔，在時效性的功能上更是無法發揮，如災害查報系統。就算 Web GIS 的資料提供者具有 Web GIS 運作及 WWW 流程知識而達到主動式更新，他們也可能分散在不同區域，如一個據點在台北而另一個據點在屏東。

現在市面上可見之 Web GIS 的資料維護，皆是專業維護人員在後端伺服器做被動性更新。使用者只能在前端透過瀏覽器瀏覽資料，但是一套 Web GIS 應不僅只能透過瀏覽器來瀏覽資料；而應該要提供透過瀏覽器也可以做後端 Web GIS 資料的維護的功能。想要克服目前 Web GIS 之缺陷而發展出遠程分散式更新資料的技術，達成資料提供者不需專業知識的情況下，只需透過數據機及瀏覽器即可在世界各地更新 Web GIS 主機上的資料。不僅可以大幅降低資料更新之成本更可以使 Web GIS 上的資料更具有時效性，

如此才能完全享受 Internet 所帶給 GIS 的便利。欲達成此功能一定要在後端使用資料庫，且資料庫必須考量幾點：

- (1) 因為透過瀏覽器來更新後端資料庫，所以必須考慮多人同時更新的同做問題(Concurrent)。
- (2) 在 Internet 上很容易斷線，一旦一個完整的資料沒有被完整儲存到資料庫，勢必破壞資料庫的完整性，所以必須有能力處理資料庫中的回復問題(Rollback)。
- (3) 資料庫必須具備處理多媒體資料的能力。

由此可知後端資料庫已經不是一般 Dbase III 簡單形態之資料庫所能承擔，必須有支援 Concurrent、Rollback 及處理多媒體形態資料之機制，且為 Client/Server 架構之資料庫。在此我們選擇 MS SQL Server 作為後端資料庫，原因是便宜、易於使用及功能符合需求。在配合 Multi-Tier 架構使 Client 端形成 Thin Client 即可消除 3.3 節所提出之問題。

一旦開放遠程分散式更新，勢必要和以下功能一併考慮：

- (1)在 Web 上做到人員基本資料的管理及維護。
- (2)在 Web 上做到人員權限之設定。
- (3)在 Web 上做到人員身分確認。
- (4)在 Web 上做到圖層類別之管理及維護。
- (5)在 Web 上做到圖層分佈於底圖之管理及維護。

所以 Client 端必須要搭配一項互動性技術，而使得前端 Web 上具有接收使用者之指令以完成資料的更新。在此選用 ActiveX 作

為 WWW 上互動性之技術，也因為目前只有 ActiveX 能完全融入 Multi-Tier 架構且完全支援分散式物件之規格，達到前、後端互動性之境界，完全克服所提出之問題，進而達到遠程分散式更新之功能。

5.1.4 網路頻寬問題

一套 Web GIS 功能再強，其需要傳遞之資料量還是無法減少，反而會隨功能的提升而增加傳遞之資料量。對 Web GIS 而言，資料量又是非常的多。以一張地圖而言，若以 500K Byte 計算，而如今電話線的最快傳輸率也只不過是 56Kbps(使用 56K 之 Modem)，所以一張圖在全速奔馳下也需要 1.2 分鐘的時間來傳遞 ($500 \times 1024 \times 8 / 56600 / 60 \approx 1.2$)，此為理想值，實際上平均約需要 3 分鐘)，若使用者用的是 33.6K 的 Modem，且同一時間內又有 50 個使用者上線，那等一張圖往往要等上 30 分鐘以上，通常會讓使用者放棄使用此 Web GIS，所以對 Web GIS 設計而言，網路頻寬就成為最大的夢魘。但無論如何，電話線的傳輸速度是一定的，我們無法提高它的速度，也無法請每個使用者都去申請 ISDN 或光纖，所以提出以下兩種策略：

- (1)除了讀取 GIS 的資料時要從後端資料庫傳輸資料外，其他的動作一律由前端處理，如疊圖、放大縮小、移動...等。多利用前端電腦之 CPU，而儘量少傳遞資料。
- (2)依圖層資料的變化率不同，而做資料分布的最佳化。

其中第一項可以用 WWW 上之互動性技術 ActiveX 來達成，而第二項在此提出最簡單之二分法，將 GIS 圖層資料分為兩類-變動與不變動。其中不變動的就是底圖資料(包含地形、河川、山脈、等高線...等)，而會變動的就是那些據點，如加油站、餐廳、災害

點...等。而利用 ActiveX 將不會變動的資料在 Client 端電腦快取 (Cache) 起來。因對 Web GIS 而言，底圖資訊正是資料量最大的部份，一旦把底圖快取 (Cache) 在 Client 端電腦，等於把效率提升好幾倍 (除了第一次傳輸底圖時需要傳輸時間外)。在此運用 ActiveX 來作為將底圖快取 (Cache) 在 Client 端之機制，所以必須在 Client 端電腦作一些註冊的動作，針對每一個底圖檔做快取 (Cache) 記錄且加以版本控制。只要是第一次或是底圖版本不合才會下載此圖，否則一律從 Client 端電腦的快取 (Cache) 記錄中取得底圖資料，至於其他變動性資料再從後端 MS SQL Server 中讀取。如此簡單又快速解決此一問題。當然資料可分為多類，而視其使用之機率配合最佳化模式形成一個快取最佳化模式。當然快取 (Cache) 功能也可以視情況而取消，但一旦快取 (Cache) 功能取消後，即喪失本系統快速之優點。

5.2 Web GIS 設計模式

5.2.1 模式一(DCOM)

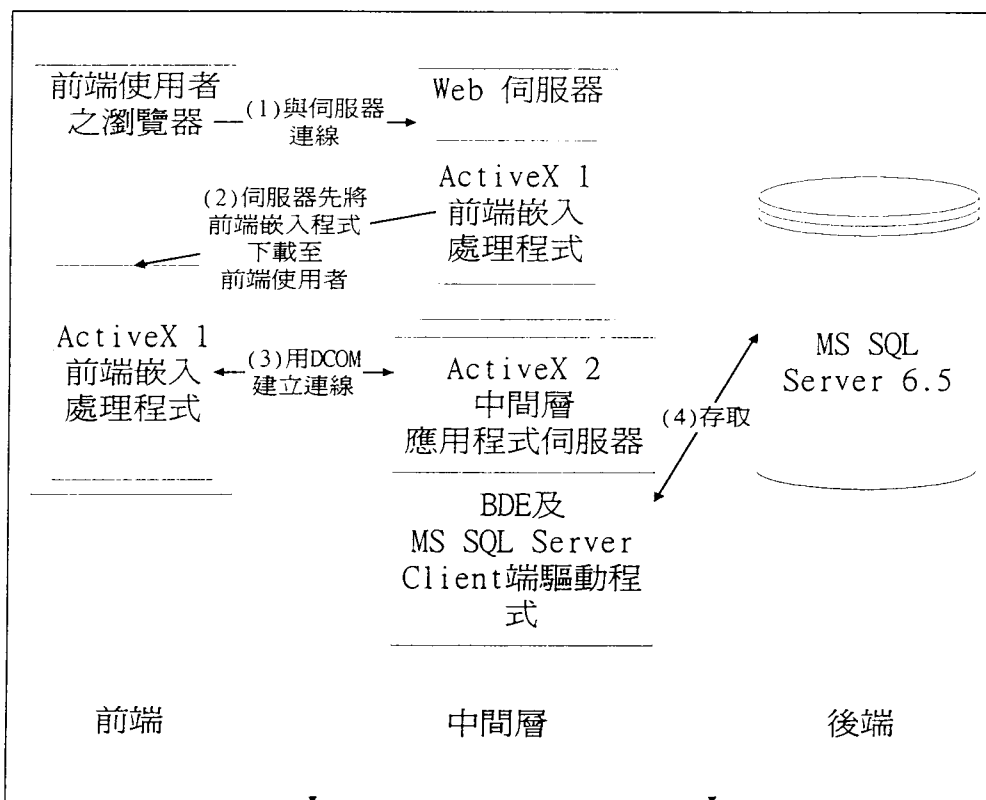


圖 22 DCOM 之 Multi-Tier Embedded Program 介面 Web GIS

架構圖及解說

(1) 使用者用瀏覽器(Microsoft Internet Explorer 3.0®以上)連上 Web Server。

- (2) 瀏覽器檢查是否已有下載 ActiveX 1 前端嵌入程式，若尚未下載或下載程式之版本不合則重新從 Web 伺服器下載 ActiveX 1 前端嵌入程式。
- (3) ActiveX 1 前端嵌入程式與 ActiveX 2 中間層應用程式伺服器利用 DCOM 建立連線。
- (4) 所有連線使用者之 ActiveX 1 前端嵌入程式皆透過 ActiveX 2 中間層應用程式伺服器統籌存取後端 MS SQL Server®。

環境需求

- (1) 後端：後端伺服器一台，其軟硬體規格如下

項目	規格
主機	Pentium® 133 以上之 IBM 相容 PC
磁碟空間	最少需要 1GB
記憶體	最少需要 32MB，建議 64MB
作業系統	Windows NT 4.0 Server®
資料庫	MS SQL Server 6.5®

- (2) 中間層：Web 伺服器一台，其軟硬體規格如下

項目	規格
主機	Pentium® 133 以上之 IBM 相容 PC
磁碟空間	最少需要 500MB
記憶體	最少需要 32MB，建議 64MB
作業系統	Windows NT 4.0 Server®
執行程式	1. IIS(Internet Information Server) 3.0® 2. 應用程式伺服器(ActiveX 2)
環境設定	1. BDE(Borland Database Engine) 驅動程式 2. MS SQL Server Client 端驅動程式

(3) 前端：使用者之電腦，其軟硬體規格如下

項目	規格
主機	Pentium® 75 以上之 IBM 相容 PC
磁碟空間	最少需要 200MB
記憶體	最少需要 16MB，建議 32MB
作業系統	Windows 95® 或 Windows 98®
瀏覽器	Microsoft Internet Explorer 3.0®以上
環境設定	1. 必須要安裝 Client for Microsoft Networks 通訊協定且設定使用者名稱及密碼。 2. 若使用 Windows 95®必須加裝 DCOM for Win95，Windows 98®則不用。

※後端伺服器 及 Web 伺服器可以放在同一台電腦。

5.2.2 模式二(TCP/IP->DCOM)

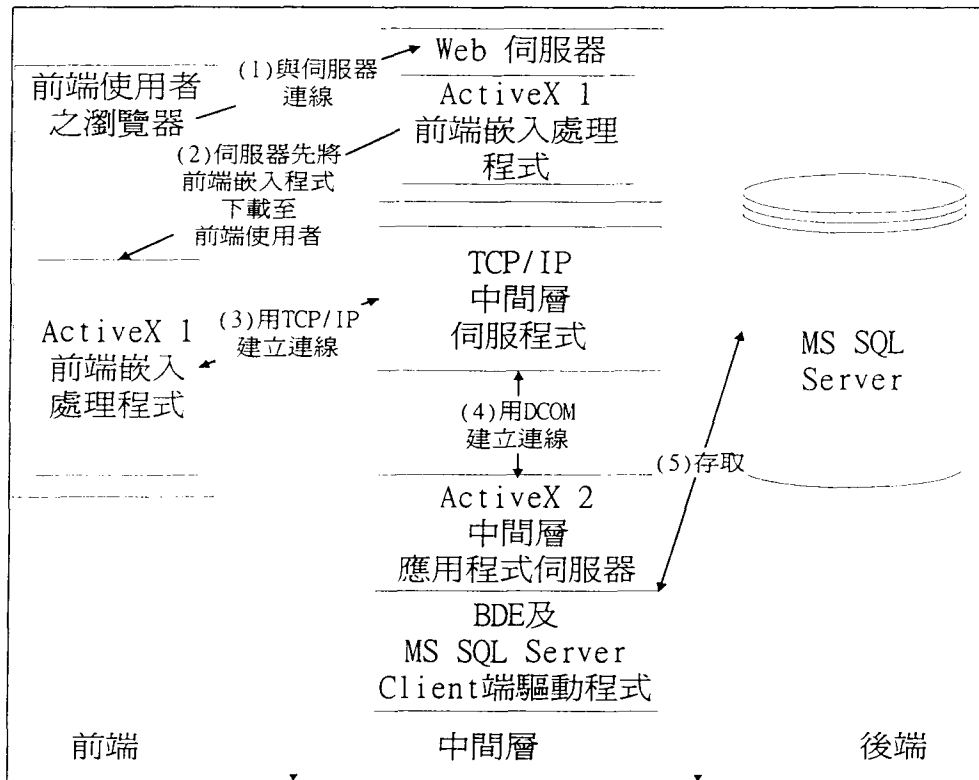


圖 23 TCP/IP 之 Multi-Tier Embedded Program 介面 Web GIS

架構圖及解說

- (1) 使用者用瀏覽器(Microsoft Internet Explorer 3.0®以上)連上 Web Server。
- (2) 瀏覽器檢查是否已有下載 ActiveX 1 前端嵌入程式，若尚未下載或下載程式之版本不合，則重新從 Web 伺服器下載 ActiveX 1 前端嵌入程式。
- (3) ActiveX 1 前端嵌入程式與 TCP/IP 中間層伺服器用 TCP/IP 建立連線。
- (4) TCP/IP 中間層伺服器與 ActiveX 2 中間層應用程式伺服器利用 DCOM 建立連線。
- (5) 所有連線使用者之 ActiveX 1 前端嵌入程式皆透過 TCP/IP 中間層伺服器再交給 ActiveX 2 中間層應用程式伺服器統籌存取後端 MS SQL Server®。

環境需求

- (1) 後端：後端伺服器一台，其軟硬體規格如下

項目	規格
主機	Pentium® 133 以上之 IBM 相容 PC
磁碟空間	最少需要 1GB
記憶體	最少需要 32MB，建議 64MB
作業系統	Windows NT 4.0 Server®

資料庫	MS SQL Server 6.5®
-----	--------------------

(2) 中間層：Web 伺服器一台，其軟硬體規格如下

項目	規格
主機	Pentium® 133 以上之 IBM 相容 PC
磁碟空間	最少需要 500MB
記憶體	最少需要 32MB，建議 64MB
作業系統	Windows NT 4.0 Server®
執行程式	1. IIS(Internet Information Server) 3.0® 2. TCP/IP 中間層伺服程式 3. 應用程式伺服器(ActiveX 2)
環境設定	1. BDE(Borland Database Engine) 驅動程式 2. MS SQL Server Client 端驅動程式

(3) 前端：使用者之電腦，其軟硬體規格如下

項目	規格
主機	Pentium® 75 以上之 IBM 相容 PC
磁碟空間	最少需要 200MB
記憶體	最少需要 16MB，建議 32MB

作業系統	Windows 95® 或 Windows 98®
瀏覽器	Microsoft Internet Explorer 3.0®以上

※後端伺服器 及 Web 伺服器可以放在同一台電腦。

5.2.3 模式三(MIDAS¹⁰)

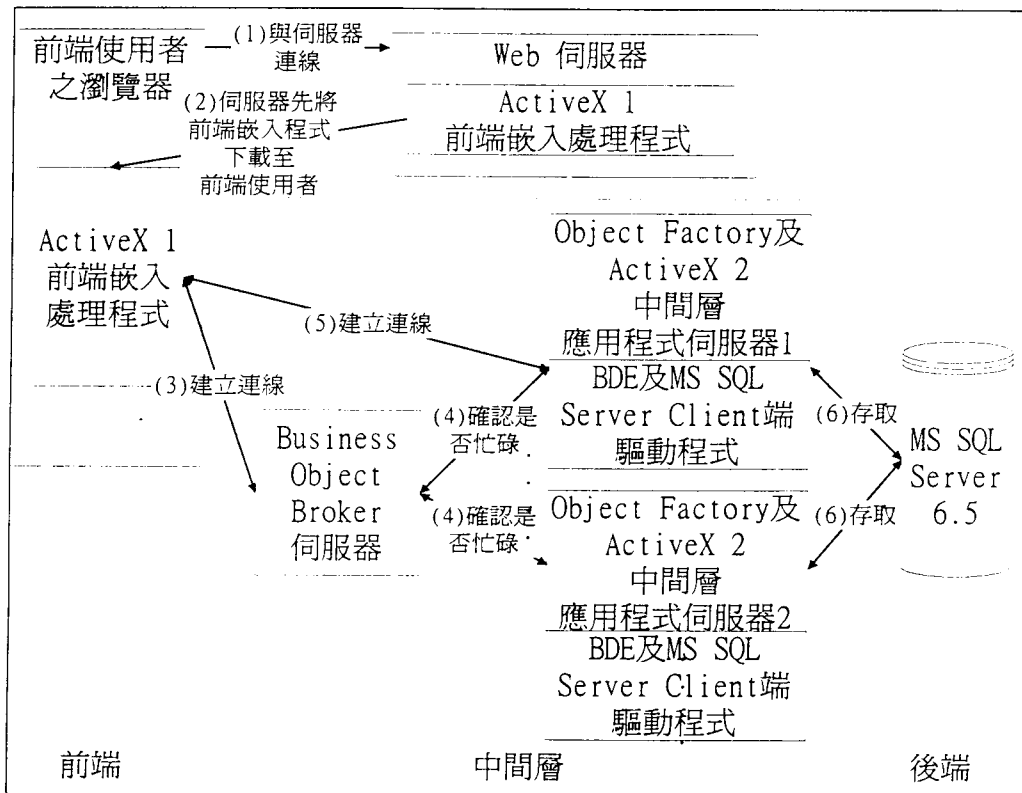


圖 24 MIDAS 之 Multi-Tier Embedded Program 介面 Web GIS

架構圖及解說

- (1) 使用者用瀏覽器(Microsoft Internet Explorer 3.0®以上)連上 Web Server。
- (2) 瀏覽器檢查是否已有下載 ActiveX 1 前端嵌入程式，若尚未下載或下載程式之版本不合則重新從 Web 伺服器下載 ActiveX 1 前端嵌入程式。

¹⁰ MIDAS 就是 Multi-tier Distributed Application Services 的縮寫，是指一種使用分散式物件的技術而達到 Multi-Tier 之架構，使程式（或物件）可以達成分散式計算與處理之功能。

- (3) ActiveX 1 前端嵌入程式與 Business Object Broker¹¹ 伺服器建立連線。
- (4) Business Object Broker 伺服器即馬上檢查所有的 ActiveX 2 中間層應用程式伺服器，決定哪一台 ActiveX 2 中間層應用程式伺服器負載最輕及相對於處理使用者工作而言是最佳化的(如距離最近...等)。
- (5) 將使用者之工作丟給由 Business Object Broker 伺服器所決定之 ActiveX 2 中間層應用程式伺服器。於是此使用者就透過此 ActiveX 2 中間層應用程式伺服器與後端 MS SQL Server 連線。
- (6) 所有連線使用者之 ActiveX 1 前端嵌入程式皆透過 Business Broker 伺服器裁決將工作平均分配給 ActiveX 2 中間層應用程式伺服器用以存取後端 MS SQL Server。

環境需求

- (1) 後端：後端伺服器一台，其軟硬體規格如下

項目	規格
主機	Pentium® 133 以上之 IBM 相容 PC
磁碟空間	最少需要 1GB
記憶體	最少需要 32MB，建議 64MB
作業系統	Windows NT 4.0 Server®

¹¹ 指 MIDAS 下之物件供應裁決者，任何使用者欲使用某種物件都必先透過此物件供應裁決者的決定，此裁決者會利用一些最佳化之方式去尋求有哪些物件可供使用，而其中得哪一個物件式最適合使用者使用（如覆載最輕...等）。

資料庫	MS SQL Server 6.5®
-----	--------------------

(2) 中間層：Web 伺服器一台、Business Broker 伺服器一台、ActiveX 中間層應用程式伺服器至少兩台，其軟硬體規格如下

➤ Web 伺服器

項目	規格
主機	Pentium® 133 以上之 IBM 相容 PC
磁碟空間	最少需要 500MB
記憶體	最少需要 32MB，建議 64MB
作業系統	Windows NT 4.0 Server®
執行程式	1. IIS(Internet Information Server) 3.0®

➤ Business Object Broker 伺服器

項目	規格
主機	Pentium® 133 以上之 IBM 相容 PC
磁碟空間	最少需要 200MB
記憶體	最少需要 32MB
作業系統	Windows NT 4.0 Server® 或 Windows NT 4.0 Workstation® 或 Windows 95® 或

	Windows 98®。
執行程式	OLEnterprise ¹² ® 之 Business Object Broker。

➤ ActiveX 中間層應用程式伺服器

項目	規格
主機	Pentium® 133 以上之 IBM 相容 PC
磁碟空間	最少需要 200MB
記憶體	最少需要 32MB
作業系統	Windows NT 4.0 Server® 或 Windows NT 4.0 Workstation® 或 Windows 95® 或 Windows 98®。
執行程式	1. OLEnterprise® 之 Object Factory ¹³ 。 2. 應用程式伺服器 (ActiveX 2)。

(3) 前端：使用者之電腦，其軟硬體規格如下

項目	規格
主機	Pentium® 75 以上之 IBM 相容 PC
磁碟空間	最少需要 200MB

¹² OLEnterprise：為 Borland™ 所出品之一套 MIDAS 倒裝軟體，其中供應 MIDAS 所會用到之功能，如 Object Explorer、Business Object Broker、Object Agent 及 Object Factory...等。

¹³ Object Factory：指 MIDAS 下之物件提供者。

記憶體	最少需要 16MB，建議 32MB
作業系統	Windows 95® 或 Windows 98®
瀏覽器	Microsoft Internet Explorer 3.0®以上
環境設定	1. 必須要安裝 OLEnterprise® 之 Object Agent ¹⁴ 。

※後端伺服器、Web 伺服器及 Business Object Broker 伺服器可以放在同一台電腦。

¹⁴ Object Agent：指 MIDAS 下之物件使用者。

5.2.4 模式比較

模式	優點	缺點
模式一 (DCOM)	1. 架構較簡單。	1. 使用者之電腦若是使用 Windows 95® 作業系統則必須安裝 DCOM for Win95 (Windows 98® 不用) 且要安裝 Client for Microsoft Networks 通訊協定且設定使用者名稱及密碼。
模式二 (TCP/IP)	1. 對使用者而言最簡單，完全不用任何設定。	1. 因為多了一層 TCP/IP 中間層伺服程式將 TCP/IP 之通訊資料轉為 DCOM 之通訊資料轉，故速度稍慢。
模式三 (MIDAS)	1. 可以真正達到分散式處理之境界。	1. 架構複雜，硬體成本高。 2. 使用者之電腦需要安裝 OLEnterprise® 之 Object Agent 才可以執行此架構。

表 10 各種模式之優缺比較

由表 10 可知，模式一是最原始之架構，簡單易懂；但使用者的電腦若是使用 Windows 95® 作業系統則必須安裝 DCOM for Win95 (Windows 98® 不用) 且要安裝 Client for Microsoft Networks 通訊協定並正確設定使用者名稱及密碼，對使用者而言過於複雜，容易導致問題發生。模式三雖是最完整之模式，但同樣地使用者之電腦需要安裝 OLEnterprise® 之 Object Agent 才可以執行此架構，對使用者而言過於複雜，更容易導致問題發生，所以模式一、三對目前的作業環境而言尚不成熟，要等到作業系統內定支援此作業方式，且完全不用作額外設定時，時機才成熟 (Windows 98® 內定支援 DCOM)。所以在目前階段選用最保守之模式二，當然日後作業系統較成熟後即可更換為模式三。

5.3 設計功能分析

本系統採用模式二(請參照章節 5.2.2)，故共需要四支程式：後端底圖資料庫建立程式、前端 ActiveX1 程式、TCP/IP 中間層伺服器程式及中間層 ActiveX2 程式，其主要功能分述如下：

1. 後端底圖資料庫建立程式

因底圖取得成本關係，此 Web GIS 底圖是採用網格式檔案(圖形檔格式為 GIF)。建構後端底圖資料庫的動作則不開放在 Internet 上(因其可行性低且效益不大)，故後端底圖資料庫只能在 Local LAN 上建立。此程式目的為：

- (1) 在 Local LAN 上輕易將圖形檔放入 Web GIS 的地圖資料庫。
- (2) 建立底圖間相鄰關係。
- (3) 建立底圖間從屬關係(底圖中包含哪些更詳細之子底圖)。

前端 ActiveX 1 程式

此程式之格式為 ActiveX，屬於網路上使用者端之分散式物件。在使用者第一次上網時會自動下載至使用者電腦中，負責一切使用者操作時之處理動作，如查詢、編修、秀圖...等。為本套系統最重要之程式，其主要功能分述如下：

- (1) 測試使用者端電腦是否有底圖快取(Cache)資訊。

- (2) 在使用者端電腦建立底圖快取(Cache)紀錄。
- (3) 提供使用者彈性之操作介面：查詢之各種功能-依區域、保留字、階層總攬、定點、疊圖...等查詢功能。
- (4) 進入編修模式時身份確認。
- (5) 使用者權限編修。
- (6) 圖層編修。
- (7) 據點編修。
- (8) 將(3)~(7)一切使用者的操作做合理性檢查，以便馬上可以在瀏覽器上反應錯誤。
- (9) 若在(8)中使用者的操作皆正確則將指令傳至程式 3- TCP/IP 中間層伺服器程式。
- (10) 接收程式 3- TCP/IP 中間層伺服器程式-傳回之資料正確顯示在瀏覽器。

TCP/IP 中間層伺服器程式

- (1) 只有一項任務做為前端 ActiveX 1 程式與中間層 ActiveX 2 程式溝通的橋樑。前端 ActiveX 1 程式將指令用 TCP/IP 通訊協定收發，中間層 ActiveX 2 程式將指令用 DCOM 通訊協定收發，其主要功能為互轉 TCP/IP 與 DCOM 兩種通訊協定。
- (2) 中間層 ActiveX 2 程式：接收由 ActiveX 1 而來的指令，經過整理後，再去驅動後端 GIS Web Server 的資料庫。

第六章.

Web GIS 資料庫分析 與設計

6.1 資料庫之特性

儲存 GIS 之資料庫又稱為空間資料庫(Spatial Database)，而空間資料庫與一般傳統純文字資料庫在考量上複雜許多，因考慮之層面由一維擴展為二維(若再加上高度則成為三維)，而主要考慮之問題有 4 點：

1. 空間查詢(Location; What is at...?)

查詢圖面上任一地點時之定位問題，而查詢之地點可分為三大類：

- 點資料—災害點、測站、水井、罪案地點、高度等。
- 線資料—道路、河流、地下管線等。
- 面資料—土壤、地質、地籍、崩塌潛力等。

空間分佈之分析(Patterns; What spatial patterns exist?)

例如犯罪地點、交通事故地點的集中情形。如癌症病患分佈與環境污染之關係、零售商店空間分佈分析及客戶資料空間分佈分佈等。

屬性查詢(條件查詢) (Condition; Where is it...?)

查詢符合特定條件的地點在那裡(區位查詢)。例如理想的住宅地、垃圾掩埋場、地下水井、火力發電廠等。

最佳路徑尋找(Routing; What is the best way...?)

任意兩點或數點之間找出最佳路徑。例如消防隊及救護車等即時服務、包裹運送路線規劃及公車及垃圾車路線規劃等。

其中 1,2 點可說與資料庫綱目(Schema of Data Base)息息相關，而第 3 點以後則是仰賴知識庫(Knowledge Base)之設計，在此研究中暫且只做到 1, 3 點，而第 4 點已完整提出理論架構且實做也正進行中。

6.2 需求分析

通常空間資料(Spatial Data)可視為三部分：

- (1)圖形資料(Graphics) 一點、線、面。
- (2)座標資料(Coordinate)一 球面座標/平面座標。
- (3)空間關係(Topology)一 相鄰/相連/組成。

在此研究中以達成點資料之查詢為主要目的。至於線、面資料之查詢暫不加以討論，故圖形資料中可分為點、面。其中面圖形資料為底圖資料(也就是未做任何疊圖時之地圖)，以網格格式的圖檔為主，而一個底圖可以包含若干個子底圖，當然子底圖更可以包含若干個孫底圖，如此一直延續下去直到最詳細之底圖為止，故底圖之層數為變動式並未固定。而點圖形資料為各個圖層之圖示(Logo)，用以疊在底圖上作為顯示之用。

當某一圖層有據點實際發生時只要將所發生之座標位置記錄下來即可，在此座標資料以平面座標為主，且記錄方式為記錄此據點發生在哪一個底圖之相對位置。雖然使用相對平面座標時，一旦此底圖的比例尺有變動，資料即宣告失效；但好處為簡單、運算快速。

而空間關係就只有地圖與底圖之間的對應(Mapping)問題，也就是說在一張父底圖上包含哪些子底圖，而父底圖上之座標點是座落於哪一張子底圖之上。在此使用下列規則來規定底圖的空間關係(Topology)：「在每一張子底圖上皆紀錄著此資料，此子底圖座落於父底圖的哪一個區間上，而用一連串之點(Point)將此區間圍繞起來」。

6.3 資料庫 E-R Model

將上述需求用 E-R Model 表示如下：

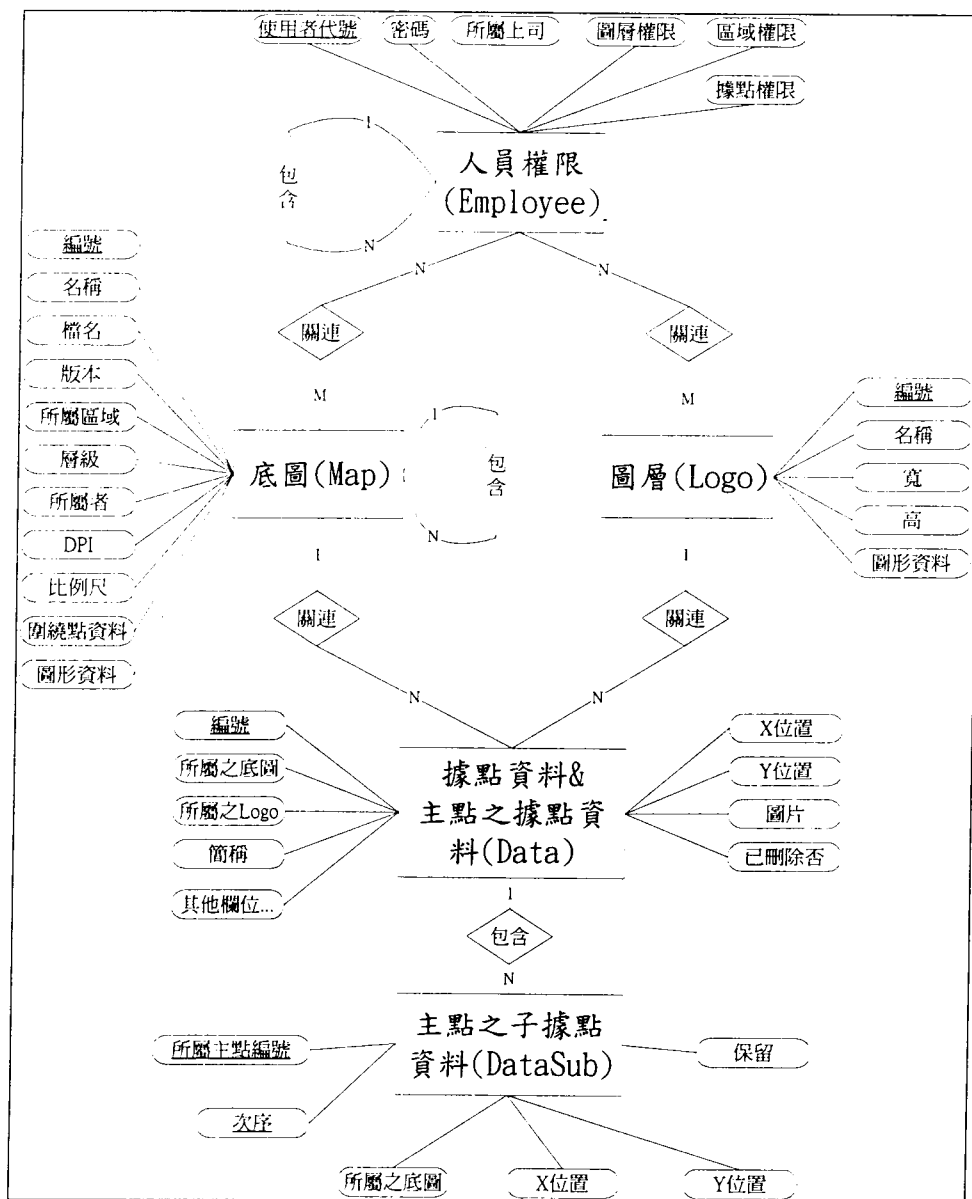


圖 25 本系統資料庫之 E-R Model

下列描述各個 Entity 之功能及其關係：

(1) 人員權限檔(Employee)

記錄所有維護人員之名稱、密碼及權限，且記錄人員中上司與下屬之關係(因此自己具有一對多之關係)，只有在此檔案中有紀錄之人員才有權限去做 Web GIS 之編修動作。在權限部分共有圖層權限、區域權限及據點權限。其中圖層權限用以記錄是否可以對圖層(Logo)檔編修。區域權限則和記錄是否允許對底圖(Map)檔上每一張底圖做據點的處理，雖然其關係是多對多，但在此系統中有限制底圖(Map)檔最多只能包含 256 張底圖，故不需將多對多關係再遷一個檔案加以記錄。據點權限則是分別記錄各個圖層據點的編修權限，其關係亦為多對多。但在此系統中限制最多只能包含 200 個圖層，亦不需將多對多關係遷出一個檔案記錄。

(2) 底圖檔(Map)

記錄所有底圖之資訊，在此系統中限制底圖檔最多包含 256 張底圖。每張底圖皆可以擁有多張子底圖(父子底圖之空間關係資訊)，所以自己具有一對多之關係。

(3) 圖層檔(Logo)

記錄所有各個圖層之圖示資料。

(4) 單一點之據點資料&主點之據點資料檔(Data)

記錄圖層資料真正發生時之所在位置，故與所屬的底圖檔(Map)與發生之圖層(Logo)具有關連。

(5) 主點之子據點資料檔(DataSub)

作為記錄線性資料，保留作為日後擴充用。

6.4 資料庫之綱目(Schema)

將上述需求加以正規化後得到五個表格(Table)，為人員權限檔(Employee)、底圖檔(Map)、圖示檔(Logo)、單一點之據點資料&主點之據點資料檔(Data)及主點之子據點資料(DataSub)。其檔案詳細規格如下：

6.4.1 人員權限檔：Employee

欄位部份

中文意義	欄位名	資料形態	大小	Null	內定值	唯一	備註(*:代表主鍵)
使用者代號	Name	Char	10	N		Y	*
密碼	Password	Char	10	N			
所屬上司	Father	Char	10	Y			若為局使用者此欄位為null
類別	Type	Binary	1	N			0:段使用者 1:處使用者 2:局使用者
圖層權限	Lright	Binary	1	N			
區域權限	Mright	Binary	32	N			每一個 bit 代表是否可編修此區域,最多 256 個區域(8*32)
據點權限	Pright	Binary	200	N			每一個 Byte 代表是否可編修此據點之地圖資料

註:LRight 及 Pright 欄位每一個 Byte 的結構如下:

Bit 位置	意義
0	0:不可新增 1:可新增
1	0:不可修改 1:可修改
2	0:不可刪除 1:可刪除

6.4.2 底圖檔 : Map

欄位部份:

中文意義	欄位名	資料形態	大小	Null	內定值	唯一	備註(*:代表主鍵)
編號	No	Smallint	2	N		Y	* 序號 From 1(為唯一)
名稱	Name	Char	50	N		Y	此底圖之意義
檔名	FileName	Char	12	N		Y	欲存在 Client 端之檔名
版本	Version	Smallint	2	N			
所屬區域	Type	Smallint	2	N			隸屬於哪一個第一層底圖
層級	Tier	Smallint	2	N			圖層 0,1,2,...(0 為第一張圖)
所屬者	Father	Smallint	2	N			所隸屬之上一張底圖編號
DPI	DPI	Smallint	2	Y	0		
比例尺	Rate	Int	4	Y			只顯示分母
圍繞點資料	Scope	Text		Y			此圖在上一層之範圍
圖形資料	Image	Image		Y			

索引部份:

名稱	唯一	組成欄位
No_Map	Y	No
Name_Map	Y	Name
Father_Map	Y	Father+No
Tier_Map	Y	Tier+No

6.4.3 圖示檔 : Logo

欄位部份:

中文意義	欄位名	資料形態	大小	Null	內定值	唯一	備註(*:代表主鍵)
編號	No	Smallint	2	Y			序號 From 1(但實際為唯一)
名稱	Name	Char	16	N		Y	*
寬	Width	Smallint	2	N			
高	Height	Smallint	2	N			
圖形資料	Data	Image		N			

索引部份:

名稱	唯一	組成欄位
No_Logo		No

6.4.4 單一點之據點資料&主點之據點資料檔 :

Data

欄位部份:

中文意義	欄位名	資料形態	大小	Null	內定值	唯一	備註(*:代表主鍵)
編號	No	Int	4	Y		Y	序號 From 1
所屬之底圖	MapID	SmallInt	2	N			* 底圖編號 From 0(主點之據點不使用此屬性)
所屬之 Logo	LogoID	Smallint	2	N			* 為 Image 之 No 欄位
簡稱	Name	Char	16	N			* 作為秀在底圖上之用(FK)
X 位置	X	Smallint	2	N	0		在 MapName 底圖之位置(主點之據點不使用此屬性)
Y 位置	Y	Smallint	2	N	0		
地點	Address	Varchar	80	Y			
電話	Tel	Varchar	14	Y			
傳真	Fax	Varchar	14	Y			
url 連線位置	URL	Varchar	255	Y			
所在道路	Road	Varchar	40	Y			
內容	Context	Varchar	80	Y			
日期 1	Date1	Date	8	Y			
日期 2	Date2	Date	8	Y			
日期 3	Date3	Date	8	Y			
已刪除否	Del	Smallint	2	N	0		0:否 1:是
描述	Description	Text		Y			
圖片	Picturee	Image		Y			

索引部份:

名稱	唯一	組成欄位
No_Data	Y	No
LogoID_Data		LogoID+MapID+Name
Name_Data		Name
XY		MapID+LogoID+X+Y

6.4.5 主點之子據點資料 : DataSub

欄位部份:

中文意義	欄位名	資料形態	大小	Null	內定值	備註(*:代表主鍵)
所屬主點編號	DataNo	Int	4	N		* 為 Data 之 No 欄位
次序	No	Smallint	2	N		*
所屬之底圖	MapID	SmallInt	2	N		* 底圖編號 From 0
X 位置	X	Smallint	2	N		在 MapName 底圖之位置(專為主點之據點用)
Y 位置	Y	Smallint	2	N		
保留	Reverse	Char	20	Y		

索引部份:

名稱	唯一	組成欄位
No_DataSub	Y	DataNo+No

第七章.

Web GIS Server 地圖 庫之建構

此系統中資料可分兩類， Web GIS Server 底圖庫與 Web GIS Server 圖層據點資料庫。Web GIS Server 底圖庫為由網站管理者建構；Web GIS Server 圖層據點資料庫則可透過過本文提出之遠程分散式方法由被授權者建構。

Web GIS Server 底圖庫必須經過下列手續：底圖取得、底圖掃描、底圖編修及底圖資料庫之操作，分述如下：

7.1 底圖取得

底圖的類別有兩種：

1. 網格式底圖(Grid Map)

直接用光學掃瞄器掃瞄市面上之地圖集。其優缺點如下：

- 優點：圖形漂亮自然、地圖取得容易且成本較低(掃瞄即可得到)、處理單純快速、檔案規格統一、在瀏覽器上顯示無問題。
- 缺點：檔案龐大不利於 Internet 傳輸、放大縮小時圖形會失真。

向量格式底圖(Vector Map)

向專業製圖公司購買向量電子圖檔。其優缺點如下：

- 優點：檔案小利於 Internet 傳輸、圖形可做任何比例之放大縮小。
- 缺點：圖形不夠自然真實、偏遠區塊的地圖取得困難且成本非常高、數化處理複雜速度慢、檔案規格不統一、在瀏覽器上顯示有困難。

基於台灣製作電子向量圖檔的公司不多，且偏遠地區也找不到相對應電子向量圖檔(一般只有都會區才有電子向量圖檔)，而數化成本高的因素下，本文選擇網格式之圖檔作為系統之底圖。

而底圖的來源以戶外生活圖書公司所出版的一系列地圖集為主。該套地圖集分為台灣全區圖(1/50 萬)、地區地圖(1/10 萬)與各都會區地圖(1/10000)。此系列地圖是目前民間編製地圖中最翔實、

精緻的一套。戶外生活圖書公司已授權交通大學資訊管理研究所可將其出版之部分圖幅上網，只要該類底圖是放在交大伺服器內且為公益使用皆為合法。

7.2 底圖掃描

圖形的掃描並無統一之方式，視其使用的掃描器驅動程式與影像處理軟體而定，在此以 UMAX 掃描器所附驅動程式及友立公司所出的 PhotoImpact 影像處理軟體作為範例。掃描器準備妥當後(掃描器的操作請參考相關手冊)依下列步驟操作：

- (1)先執行繪圖軟體（例如：PhotoImpact），而此繪圖軟體須支援掃描器取得影像功能者，才能使用。
- (2)在「檔案」選單上下拉，點選「取得」再選取「影像」(如下圖 26)。

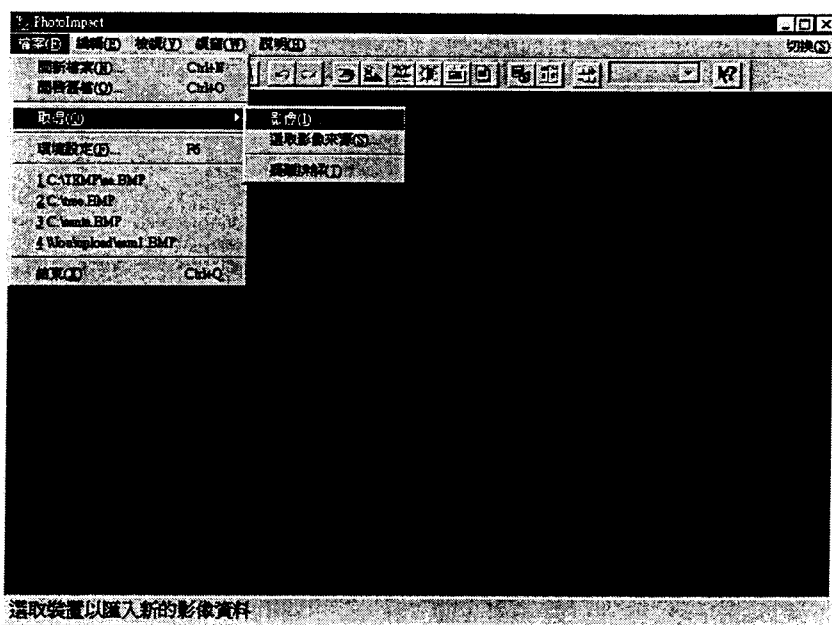


圖 26 PhotoImpact 操作畫面 1

- (3)接著 PhotoImpact 就會問要將取得的影像，存放成”新影像”、”檔案”、”印表機”或”傳真／郵件”。基本上

我們是將掃描進來的影像存放繪圖程式中，然後檢查有無問題後再存檔，所以請選擇「新影像」再按下「取得」鍵(如下圖 27)。

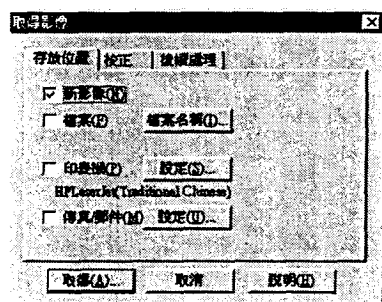


圖 27 PhotoImpact 操作畫面 2

(4)在下圖中可以看到掃描程式的畫面，請選擇影像種類為「全彩」，材質為「反射式」，掃描解析度為「150DPI」，改變影像比例為「100%」也就是不改變。再將「自動色彩校正」與「明暗度校正」點選起來，以下的 Gamma、明度、暗度、對比、亮度就可以不必設定。（當然您也可以自行設度，但是要記得最佳的值是多少，以後每次掃描地圖時都要設定，才會有一致性）。

(5)接著請按下「預掃描」鍵，就可以在右方看到大概的圖片(如下圖 28)。

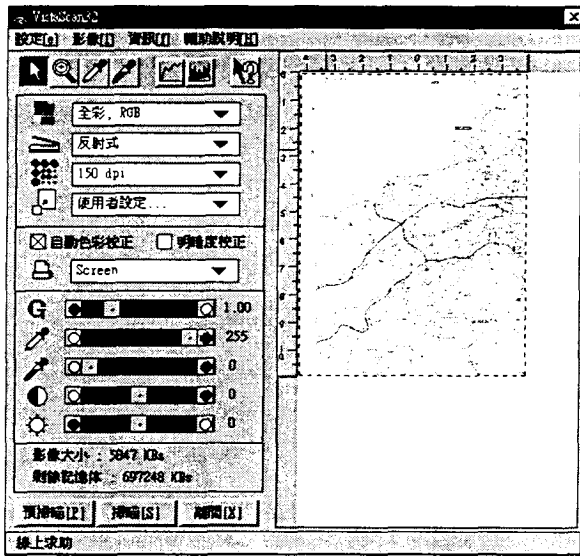


圖 28 PhotoImpact 操作畫面 3

(6) 預覽掃描沒問題後，就可以按下「掃描」鍵。圖片掃描完畢後，緊接著就會將它存入 PhotoImpact 軟體中(如下圖 29)。

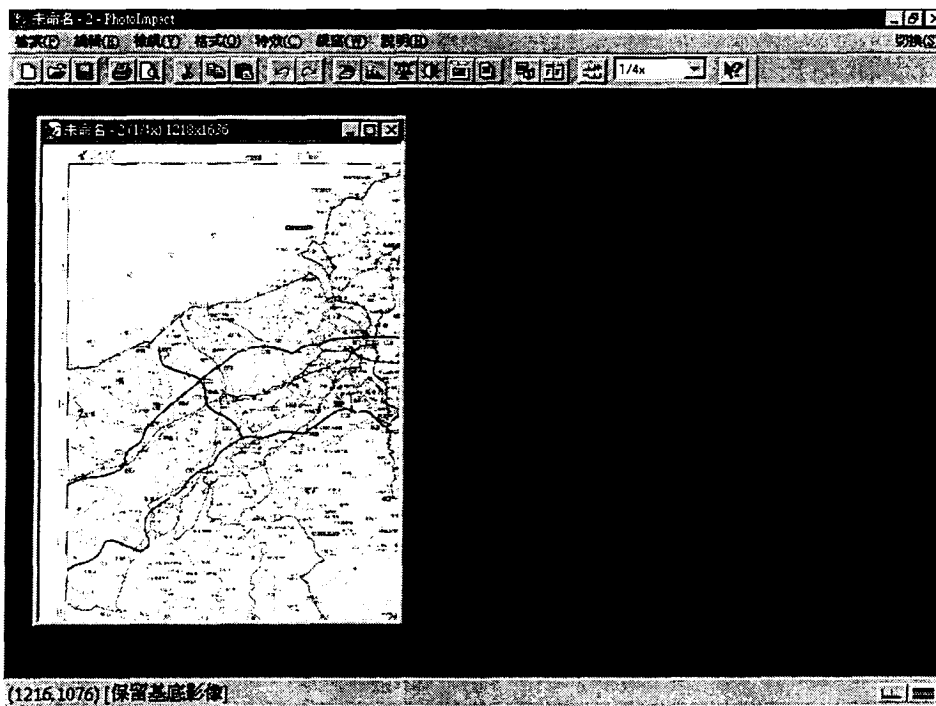


圖 29 PhotoImpact 操作畫面 4

7.3 底圖校調

圖形掃描後，必須加以校調才適合存放於資料庫以利於 Internet 上傳輸及顯示。其主要的校調事項有：

1. 圖形區塊的裁減

有時在掃描的圖形中，可能只需要其中的一小部分，此時就需將需要的部分做區塊的裁減。其操作如下：先用滑鼠將所需要的區域拖曳，在「編輯」選單上下拉，點選「複製(C)」。在「編輯」選單上下拉，點選「貼上(C)」，再點選「貼成新影像(N)」。最後在此貼上之影像中按滑鼠右鍵，點選「全部合併(A)」，即完成圖形區塊的裁減(如下圖 32)。

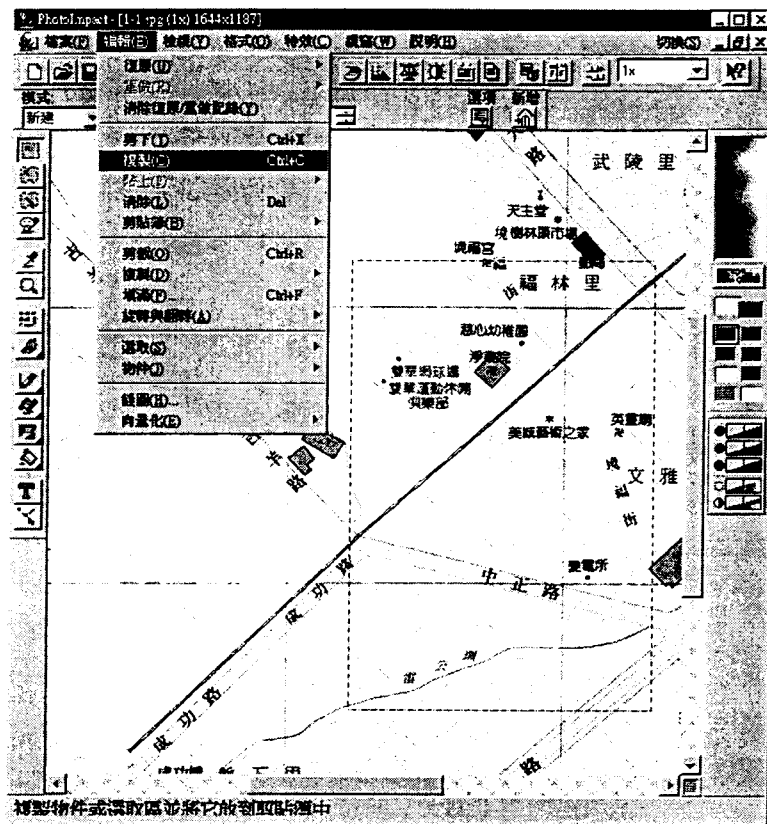


圖 32 PhotoImpact 操作畫面 7

圖形區塊的縫合

有時在掃描的圖形中，需要將兩張圖形合併形成一張，此時就需將做圖形的縫合。其操作如下：先準備兩張圖形準備要縫合，在「編輯」選單上下拉，點選「縫圖 (H)...」。則出現一畫面(如下圖 33)，在此畫面中必須移動兩張圖形直到結合時，按下確定即可做圖形區塊的縫合。

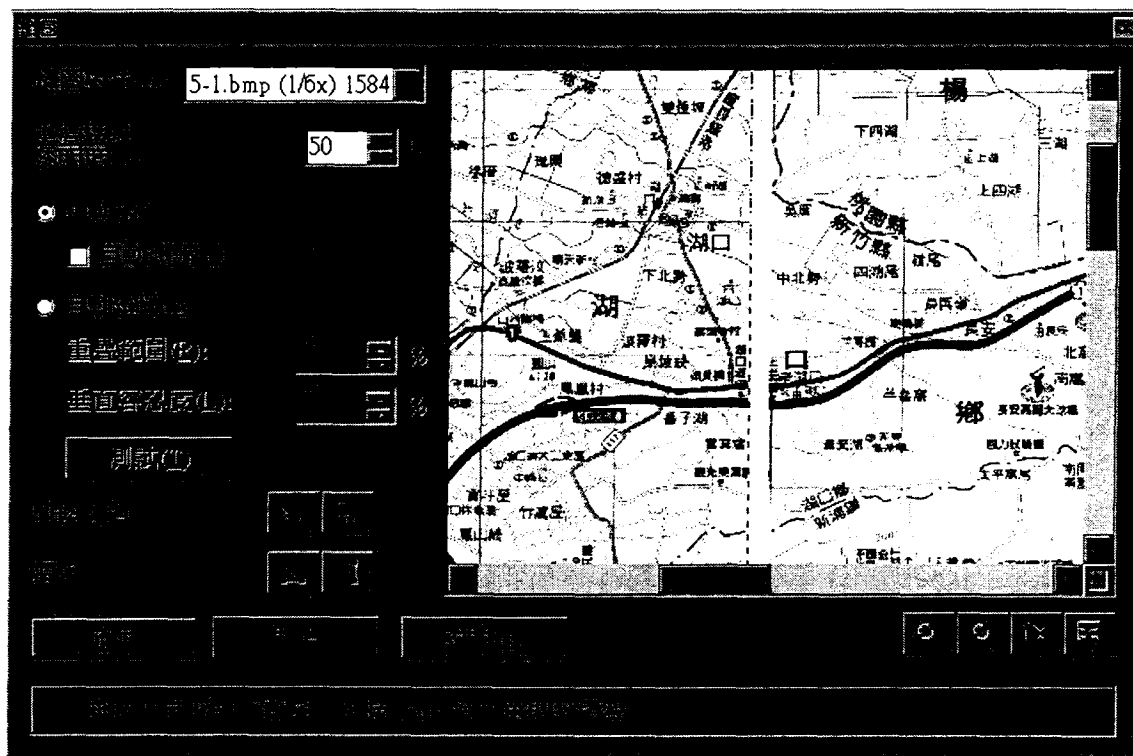


圖 33 PhotoImpact 操作畫面 8

(2) 亮度與對比

一張圖形往往會亮度與對比不對，而使品質不佳。其調整方法為：在「格式」選單上下拉，點選「亮度與對比(B)... Ctrl-B」則會出一畫面(如下圖 35)，讓您自由調整亮度與對比。

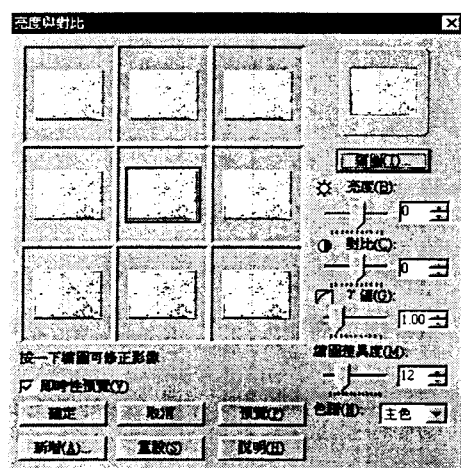


圖 35 PhotoImpact 操作畫面 10

(3) 色彩飽和度

有時色彩大鮮豔時會使人眼睛不舒服，其調整方式為：在「格式」選單上下拉，點選「色相與飽和度(H)...」則會出一畫面(如下圖 36)，讓你自由調整色彩飽和度。

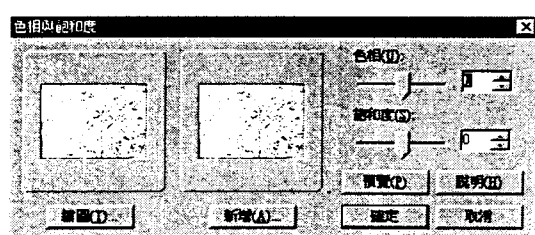


圖 36 PhotoImpact 操作畫面 11

(4) 色調調整

圖形中有最暗點、最亮點、中間值。其調整可以有效增加圖形之清晰度，而數值之調整通常要依靠經驗，其調整方式為：在「格式」選單上下拉，點選「調整色調(T)... F8」則會出現一畫面(如下圖 37)，後再點選上方的「高亮度、中間值、陰影」頁次，讓你自由調整最暗點、最亮點、中間值。

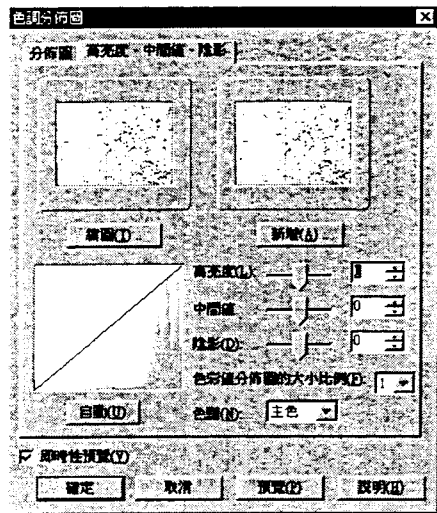


圖 37 PhotoImpact 操作畫面 12

(5) 降低色階

圖形校調完畢最就要降低色階，以最少的空間儲存所需的資訊。因底圖檔的顏色有 6 色(6 色套印)，而掃描進來的圖形皆為 True Color，在此只要將圖形色階為 256 色即可(圖形品質不被破壞且有間省記憶空間)。其操作方式為：在「格式」選單上下拉，點選「影像類型(P)」再點選「最佳化 256 色(2)(8-bit)」即可將 True Color 轉換為 256 色。(如下圖 38)

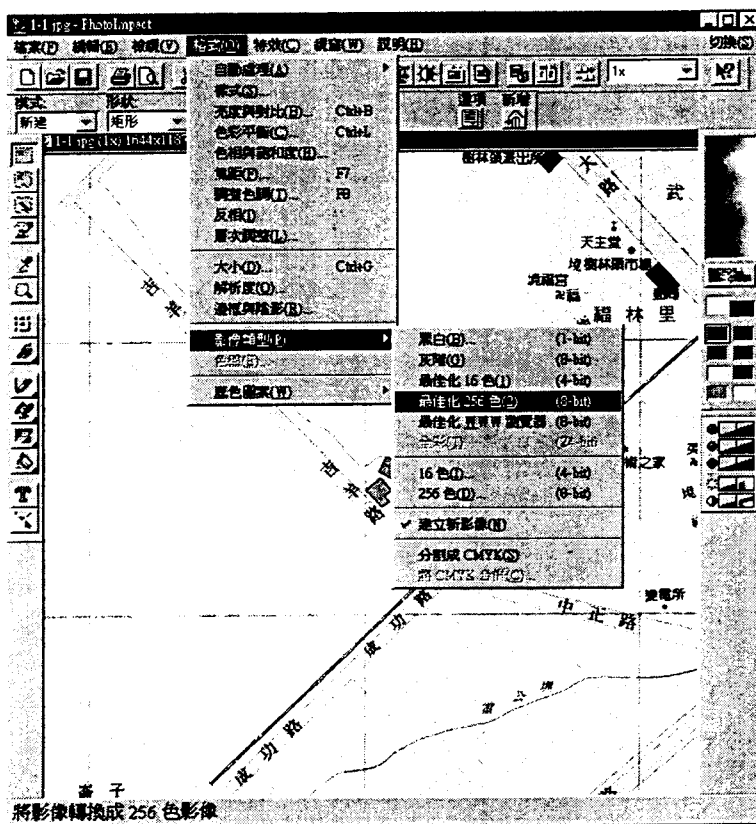


圖 38 PhotoImpact 操作畫面 13

(6) 壓縮及存檔

為節省儲存空間，必須要將圖形檔做適當的壓縮。在此選擇以不失真壓縮為主，其壓縮檔案格式在此選擇 GIF。因 GIF 檔剛好是用儲存 256 色的圖形，高壓縮率且不失真，為 Internet 上 Web 用已顯示不失真圖形的最佳選擇。其操作方式為：在「檔案」選單上下拉，點選「另存新檔(A)...」則會出一畫面(如下圖 39)，在此畫面的「存檔類型(T)」選擇 GIF(圖形交換格式)，最後在「存檔名稱(N)」中輸入名稱，按下「儲存檔案(S)」即可。

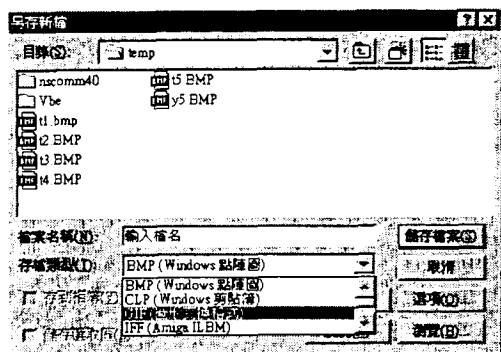


圖 39 PhotoImpact 操作畫面 14

7.4 底圖資料庫之建構

當所有圖形檔取得且處理完畢後，即可開始底圖資料庫之建構。此時需要執行後端底圖資料庫建立程式(其執行畫面如圖 40)。在底圖結構上就如同樹狀結構般，最開始只有一張圖(如台灣省縮圖)，而只要在此圖上按下新增鍵即可在此主圖上新增一個子圖(如北、中、南台灣省縮圖)，圖時在所編輸入相對應之資料。此底圖可以一層層新增下去直到最詳細之比例尺為止(在圖 40 一共有 5 層)。而修改及刪除只要先選取欲邊修之底圖，再壓下相對應按鍵即可完成工作。左下角為此圖形的基本資訊，在此可以輸入名稱、版本、DPI、比例尺及圖繞點。其中圖繞點用以構成一個封閉多邊型描述此圖位於上一張底圖位置的區域，用以表示圖形間從屬及相鄰資訊。

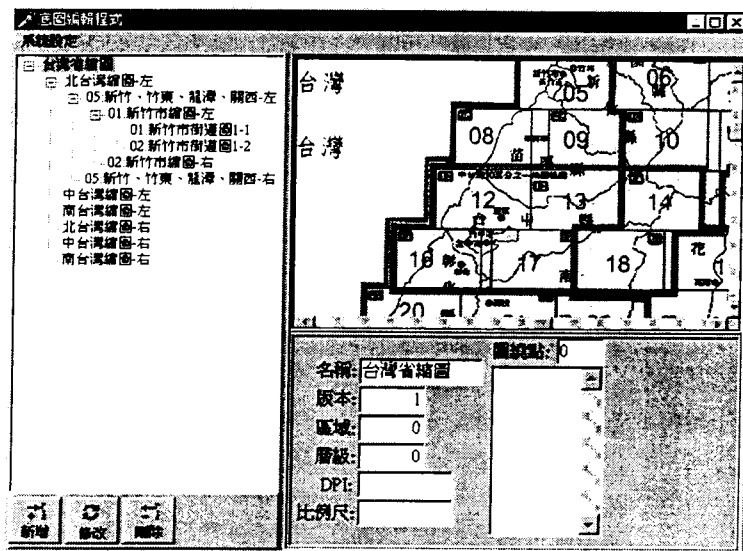


圖 40 底圖資料庫建構之執行畫面

第八章.

本系統說明

首次執行

首先啟動 Microsoft Internet Explorer 4.0，在位置上輸入 <http://140.113.73.198/system.htm> 即可進入本系統。本系統亦可以由中油網站 Hyperlink 過來。

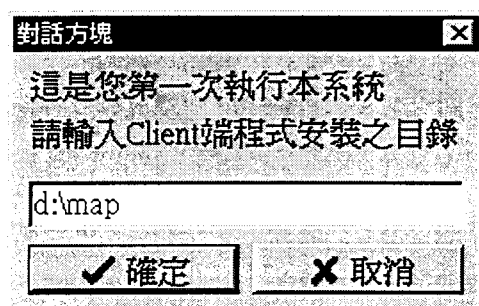


圖 41 中油網際地理資訊系統執行畫面 1

第一次連線於此網站時，會自動偵測使用者端是否有底圖快取之資料，若完全沒有則系統會顯示圖 43，詢問您要將底圖快取資料存放在哪一個目錄下。此時只要輸入一個目錄名稱，系統便會自動建立此目錄。以後就會將所有的底圖快取資料存放在此目錄下，且向系統註冊此資訊，用以管理各個底圖之版本。

8.1 查詢模式

8.1.1 查詢底圖及圖層資料



圖 42 中油網際地理資訊系統執行畫面 2

圖 42 為中油網際地理資訊系統之主畫面。上方之工具列(Tool Bar)為本系統之所有查詢功能。右上方做為顯示圖層之控制視窗，只要在方框內打勾就會動態的將據點類別(主題層)疊合於縣市底圖上。右下方為縮圖區在本例中圖 42 為台灣省之縮圖，用來選取詳細顯示的縣市，而縮圖本身可以由下方之 Track Bar 控制顯示比例。

左半部為詳細地圖顯示區，用以顯示詳細的各個縣市及主題層之疊合地圖，大小可以由右上方之 Track Bar 控制顯示比例，目前所顯示之區域為台北縣。而縮圖區及詳細地圖顯示區均可以用滑鼠在上面拖曳以便上下左右平移。



圖 43 中油網際地理資訊系統執行畫面 3

只要在右上方圖層控制視窗內的方框打勾就會動態的將主題層疊合於底圖上，圖 43 是將所有災害圖層打勾，所以左半部之台北縣地圖顯示區就將相關災害主題層疊合，且共同顯示於詳細地圖顯示區。只要在圖 43 上的任一據點上方按兩下滑鼠左鍵即可查詢據點之詳細資料之畫面，如圖 44。

據點資料		儲存容量	運輸費用
據點類別： 營業所	營業時間： AM 8:30-PM 17:30		
據點名稱： 中油公司台灣營業總處	提供產品： 各類貨品清單資訊		
電話： (02)2361-9933			
傳真： (02)2361-1501			
地址： 台北市館前路七十一號	特別活動： 5/9 母親節 親子聯歡會		
地點概述： 台北車站西北方向斜對面			
所屬單位			
名稱： 中國石油公司	地址： 台北市中華路一段八十三號		
電話： (02)2361-0221, 2371-7121			
傳真： (02)2331-4883, 2331-9645			

圖 44 中油網際地理資訊系統執行畫面 4

8.1.2 以圖形區塊查詢

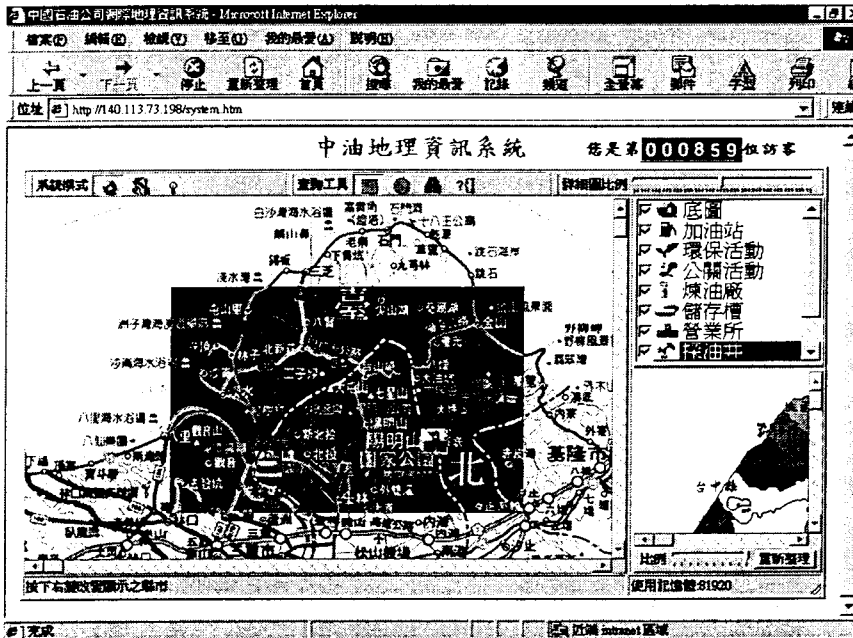


圖 45 中油網際地理資訊系統執行畫面 5

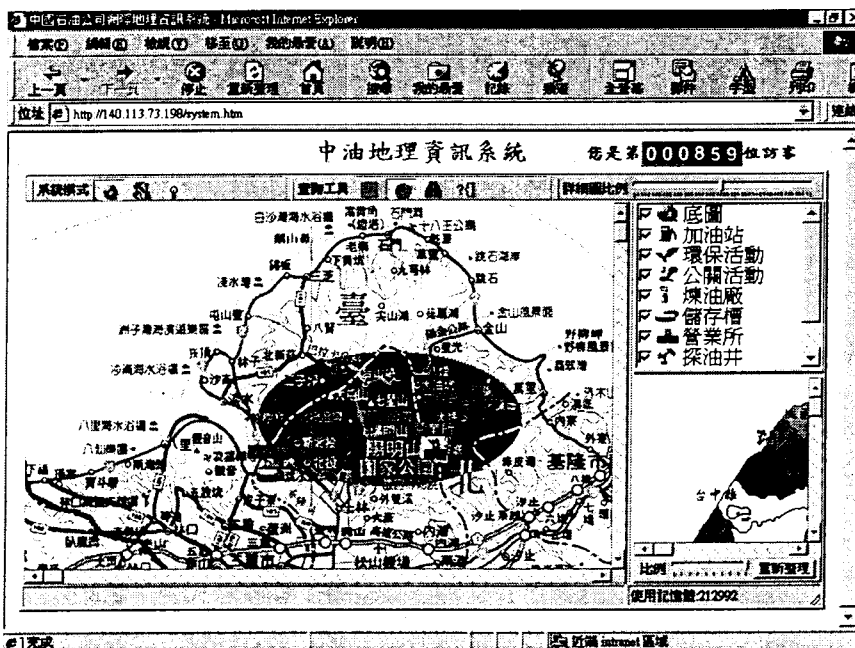


圖 46 中油網際地理資訊系統執行畫面 6

要使用圖形區塊查詢功能，只要將查詢工具中的方形或圓形圖樣 (Icon) 壓下。圖 45、圖 46 分別是壓下矩形查詢(第一個按鍵)、圓形查詢(第二個按鍵)後，用滑鼠在詳細地圖顯示區拖曳所形成之反白區塊。此時只要放開滑鼠則在此反白區塊內所有被勾選之圖層資訊(右上方圖層控制視窗內的方框中有打勾者)，皆會被搜尋到而產生圖 47 之報表。在圖 47 之報表中的任一系列用滑鼠點一下(Click)，會將詳細地圖顯示區域自動載入有包含所點選據點之底圖。

據點資料		
地點	類別	地址
中油公司台灣營業總處	營業所	台北市館前路七十一號

圖 47 中油網際地理資訊系統執行畫面 7

8.1.3 以保留字查詢

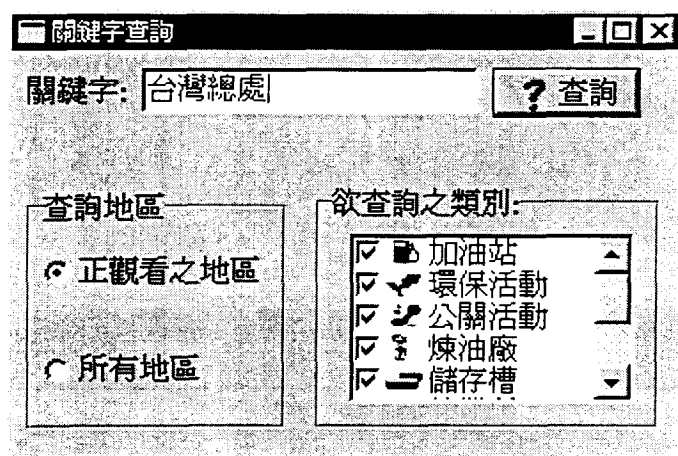


圖 48 中油網際地理資訊系統執行畫面 8

查詢方式除了上述圖形查詢法，也可以用傳統保留字查詢法。只要在主畫面上的查詢工具列壓下最右邊的圖樣即出現圖 48 之畫面。其中關鍵字欄位只要輸入欲查詢之關鍵字(查詢範圍可包含名稱、地址及描述)，其查詢之範圍可包含正在觀看之地區(主畫面左半部詳細地圖顯示區正顯示之區域)或所有地區，而查詢之圖層更可包含所選擇之圖層。一旦查詢完畢，即出現產生圖 47 之報表。在圖 47 之報表中的任一系列用滑鼠點一下(Click)，會將詳細地圖顯示區域自動載入有包含所點選據點之底圖。

8.2 遠程分散式編修模式

8.2.1 身份確認

要進入編修模式前，首先要通過身分確認，以決定編修之權限。首先在主畫面上方左邊的系統模式列第二個按鈕壓下(如圖 49)即出現圖 50 之畫面，在上面輸入正確之使用者代號及密碼可以通過身分確認，並依帳號不同取得不同編修之權限。一旦通過身分確認後，主畫面上方中間的查詢工具列即會被編修工具列取代，如圖 51。

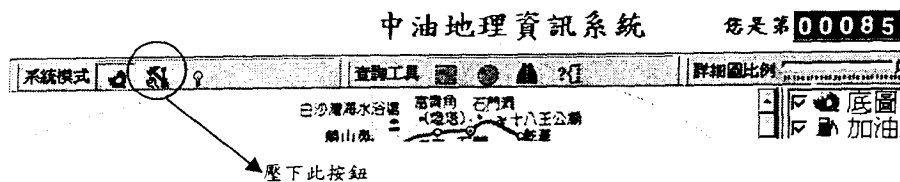


圖 49 中油網際地理資訊系統執行畫面 9

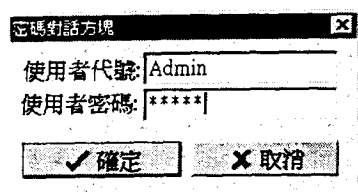


圖 50 中油網際地理資訊系統執行畫面 10

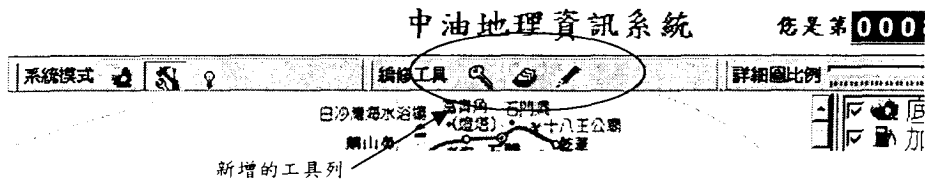


圖 51 中油網際地理資訊系統執行畫面 11

8.2.2 人員權限設定

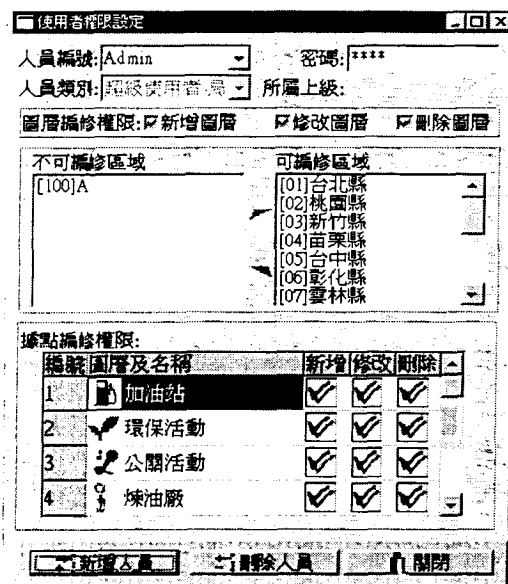


圖 52 中油網際地理資訊系統執行畫面 12

要設定每一個編修人員之權限，則是在通過身分確認的情況下，按下編修工具列的第一個按鍵壓下即出現圖 52 之畫面。此畫面用以對每一個編修人員之權限進行設定。此畫面可以設定編修人員是否可編修圖層，決定哪些底圖(縣市)可編修，及各個圖層是否可以新增、修改或刪除據點。再進入此畫面時，只能查詢到權限比自己小之使用人員。

8.2.3 圖層編修

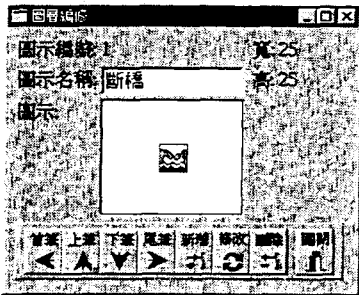


圖 53 中油網際地理資訊系統執行畫面 13

要編修圖層的種類，則是按下編修工具列的第二個按鍵，此時會出現圖 53 之畫面。此畫面用以新增、修改及刪除每一個圖層之資料。

8.2.4 據點編修

要新增、修改或刪除個別據點，則須先按下編修工具列的第三個按鍵(如圖 54)，以進入據點編修模式。此時主畫面左邊會出現一個由圖層圖示所組成之據點工具列。若主圖層控制視窗中的圖示方框內有打勾，左邊工具列之圖示即會顯示出來(如圖 55)。而據點編修可分為新增、修改及刪除其說明如下：

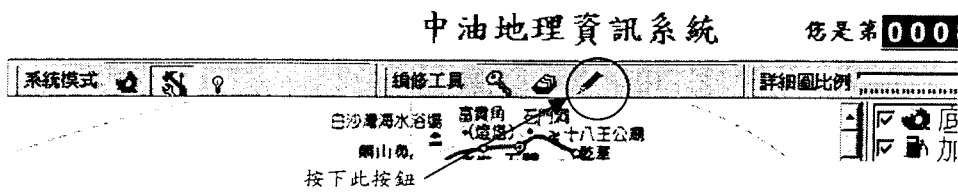


圖 54 中油網際地理資訊系統執行畫面 14



圖 55 中油網際地理資訊系統執行畫面 15

(1) 新增據點

一旦左邊工具列之圖示(淡紫色)顯示出來，即表示此圖層據點可以新增，而新增之方式為「壓下此圖示拖曳到詳細地圖顯示區的正確位置然後放開滑鼠的按鍵」，新的據點在拖到正確位置後，即會出現圖 56 之畫面，用以確認是否要新增據點，確認後便會出現圖 57 之畫面(可編修欄位為黃色)即可開始新增作

業。其中所屬單位一欄會先提供選單以供選擇，若所屬機關不在選單內，則可按下新增鍵以新增一個上級機關（如圖 58）。



圖 56 中油網際地理資訊系統執行畫面 16

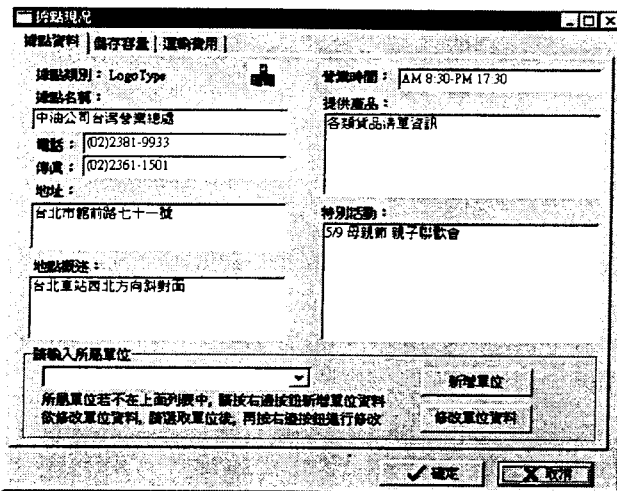


圖 57 中油網際地理資訊系統執行畫面 17

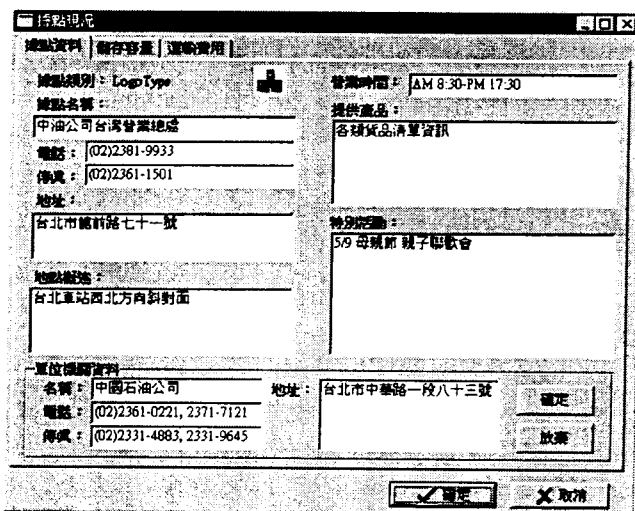


圖 58 中油網際地理資訊系統執行畫面 18

(2)修改據點

欲移動據點位置時，只要在詳細地圖顯示區上的據點圖示壓下且拖曳，直到目的地才放開滑鼠。即會顯示圖 59 之畫面用以確認是否將據點移動到此，一旦確定後表示移動到此新位置。欲修改據點內容時，只要在詳細地圖顯示區上的據點圖示壓下滑鼠右鍵，即會顯示圖 60 之畫面。此時選擇修改即出現圖 57 之畫面，用以進行修改。

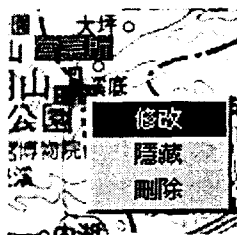
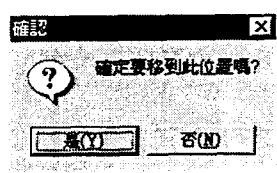


圖 56、60 中油網際地理資訊系統執行畫面 19、20

(3)刪除據點

欲刪除據點時，只要在詳細地圖顯示區上的據點圖示壓下滑鼠右鍵，即會顯示圖 61 之畫面。此時選擇刪除即完成刪除作業。

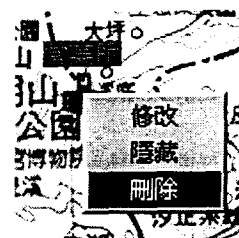


圖 61 中油網際地理資訊系統執行畫面 21

第九章. 結論與討論

本文目的乃在設計一套可滿足下列條件之網際地理資訊系統：

- 1.提供超圖形(HyperMap)的功能
- 2.提供圖查屬性的功能
- 3.提供地圖描繪的功能
- 4.解決系統在多人查詢時，效能不彰的問題。
- 5.提供中文介面之網際地理資訊系統，提高使用者使用意願

我們根據現今網際地理資訊系統的缺點以及傳統地理資訊系統的優勢，設計出了完全不遜於傳統地理資訊系統的網際地理資訊系統，亦解決了上述的問題。雖然系統仍許有許多缺失，但卻也證明網際地理資訊系統不僅可行，更可取代傳統的地理資訊系統。我們所使用的方法為：

以內嵌程式克服 Browser 的限制
以 Multi-Tier 連結傳統 Server、client、Database
加強瀏覽器查詢功能以增加效率

目前發展瀏覽器的廠商皆在開發功能更強，應用範圍更廣的瀏覽器。而超文件建構語言(HTML)標準的版本也由 1.0 提升至 3.0，甚至有廠商推出動態式超文件建構語言(Dynamic HTML)，加強客戶端瀏覽器的功能基本上已成為各方的共識。

雖然我們解決了技術上的問題，但若要發展商業用途的地理資訊系統的過程中，除了上述問題外，尚需探討以下課題：

(1)底圖使用

底圖來源:需取得準確又詳細的各種比例尺地圖。

底圖授權:可採政府印行地圖或民間印行地圖二種。但政府發行之地圖,迄今尚未准上網路;民間有部份底圖品質甚佳,可能經其授權後上網。

底圖更新:需有單位實際管理及更新底圖。但電子地圖的建立通常需要很多時間,如何有效掌控其時效,也是有待討論。

(2)經營管理

系統經營方式:系統伺服器(SERVER)的經營方式有三:1.政府管理;2.民間自辦;3.協會代辦。初期可採取資訊使用者付費方式經營,再逐步擴展為會員年金制,向會員收取固定費用。

Inter 與 Intra:在 I-GIS 未來的發展,也應考慮在一般網路與在公司內部網路的使用情形,尤其在公司內有許多資訊是具有保密性,不允許一般人任意取得,其安全分級的資料保護則日益重要。

伺服端的管理:I-GIS 可只維護基本的底圖,而大部份增值圖的資料及超連結(HYPERLINK)則由一般公司提供,而這樣的架構除了減少伺服器維護成本,更可以達到資料即時更新的效果。

地理資訊系統的應用發展除了需提昇資訊技術及開發不同領域的應用外,擴大使用層次及使用對象將促進地理資訊系統應用普及化及多樣化。為達此目的,網際地理資訊系統將成為地理資訊系統推動的主要方向。在國內目前由政府機構大力推動下,未來將有更多商業機構及大型公司逐漸將 GIS 與 MIS 系統予以整合,一方面滿足業務上圖形資料應用管理需求外,更可經由網際網路創造新商機。例如保全業、仲介業、旅遊業、零售業及各種顧問公司都逐漸關切地理資訊系統應用所能帶來之效益。於此期盼政府機構能優先建立地理資訊系統基礎環境及資料外,也展望藉由網際地理資訊系統使各行各業以至於一般人皆可充分的享有便捷低廉之地理資訊系統使用環境。

參考文獻

- [1] 國土資訊系統通訊 24，民國八十六年十二月二十日，p.2，p.6，p.7。
- [2] 施保旭，地理資訊系統，松崗，台北，民國八十六年。
- [3] 林立中，網際地理資訊系統的設計，交大資管所碩士論文，民國八十六年六月。
- [4] Lohman, G., Extensions to Starburst: Objects, types, functions and rules. Communication ACM, Vol 34, 94-109, 1991.
- [5] Maguire David, Grographical Information Systems, longman Science and Technical, 9-10, 1991.
- [6] Worboys, M.F, Hearnshaw, Object-oriented data and query modeling for geographical information system, International Journal Geographic Information, 670-688, 1990.
- [7] Worboys, M.F, K.T.Mason, Geographic Information Handling, John Wiley & Sons Ltd, 91-102, 1993.
- [8] Harmon, Paul. Object-Oriented Strategies, Cutter Information Corp.
- [9] Tsichrizis D.C.F.H. Lochovsky, DataBase Management Systems, Academic Press, New York, 1977.
- [10] Kemp, Z, An analysis of geometric modeling in database system, ACM Comp, Vol 19, 47-91, 1990.
- [11] Richard Wagner, JavaScript Unleashed, Prentice-Hall International.
- [12] Soley, M Richard., and Kent, William. The OMG Object Model. Modern Database Systems, W. Kim Ed., Addison-Wesley, 1995.
- [13] Dave, Dear., and Lisa. Differences Between Microsoft's and Sun's Visions for the Future of the Internet, <http://www.microsoft.com/>
- [14] Brockschmidt, Kraig. Inside OLE, 2nd Edition., Microsoft

Press, 1995.

- [15]<http://www.acl.lanl.gov/CORBA/>
- [16]<http://www.microsoft.com/>
- [17]<http://www.borland.com/midas/>
- [18]<http://tiger.census.gov/cgi-bin/mapbrowse-tbl>
- [19]<http://mapweb.parc.xerox.com/map>
- [20]<http://maps.yahoo.com/yahoo/>
- [21]<http://www.regis.berkeley.edu/grasslink>
- [22]<http://www.gateway.ciesin.org/>
- [23]<http://www.csrnr.ncu.edu.tw/chinese.htm>
- [24]<http://www.cecer.army.mil/grass/GRASS.main.html>
- [25]<http://boris.qub.ac.uk/shane/arc/ARChome.html>
- [26]<http://www.genasys.com/>
- [27]<http://www.mapinfo.com/>