

## 以軟體元件為中心的多人合作發展環境

蔡新民

台灣科技大學  
北市基隆路四段 43 號  
smt@et.ntust.edu.tw

饒志超

恆逸資訊股份有限公司  
北市建國北路三段 54 號  
Richard\_Rau@uuu.com.tw

謝景順

台灣科技大學  
北市基隆路四段 43 號  
soon@mail.ltc.edu.tw

### 摘要

鑑於軟體元件的快速成長與潛在市場，以及 Web 上的電子交易，有必要建立儲存軟體元件的 Repository Database。設計一個儲存 COM Objects 的 Repository Web Server 網站，允許設計者或公司，將軟體元件 (Component) 建置在 Web Server 網站上，系統並將相關資訊儲存于 Repository Databases 中。軟體元件購買者可透過 Web 到專業的 Repository Server 站來查詢欲使用的軟體元件，或利用 IE4.0 Browser，來瀏覽網頁 (可內含軟體元件) 所展示的 Demo。使用者可用電子交易方式購買所需之軟體元件，也可經由網頁來回報 bug，當錯誤更正後，也可經由系統自動回報給使用者，如此便構成了一個多人合作的發展環境的雛形。

### 1 簡介

Web 的風靡全世界，促使了軟硬體之快速成長之主要因素之一。Web 的主要應用是提供產品的最新訊息，以及電子交易。微軟為了爭取網路之主導權，推出了許多新技術，如 ASP(Active Server Page)：利用 Server 端 Script 程式之執行，來控制 Server 端之 Active 軟體元件，該軟體元件用來填入 HTML 中之空格，或 Script 程式中之空格，因此很容易可以動態形成網頁。IE4.0 的 DYNAMIC HTML：HTML 網頁上的每一個 HTML TA 均是一個 Active 軟體元件 (COM 物件)，利用 Client 端的 Script 語言，來控制這些 Active 軟體元件的外觀及行為，並即時反應在 Browser 上。

有關 OO Model, Ration a 公司集合了 Booch, Rumbaugh, Jacobson 三大專家，研擬了一套 UML(Universal Model Language)，目前正向 OMG 協會申請標準。微軟與 TI 合作，設計了一個 Repository Database，用來儲存 COM 物件。Repository Databases 雖使用了 SQL Relational Database(或 Access)，但卻可儲存 COM 物件，具有 OO Database 之功能。目前 Rationa 公司與微軟合作，推出了 VB5.0 的 Visual Modeler，這個發展工具，讓設計者使用 UML model 設計 3 tie 之 Client/Server 模式之程式。Visual Modeler，即使用了微軟之 Repository Database。微軟之 Repository Databases 是儲存 UML OO Model 之重要軟體工具。

本論文分析傳統多人合作發展環境下，探討環境成員間互動的種種限制。所以提出一個利用 WWW、Repository 和 COM 的技術來擴充多人合作發展環境的構想，並利用 Microsoft 的網站相關技術來實作了一個“以元件為基礎的多人合作軟體發展環境”來驗證。

本計劃所實作的系統，包含下列幾個特點：

1. 系統完全符合 Three-tier 應用程式與 Microsoft 的 DN 架構。

2. 利用 COM 的技術來發展元件軟體：本系統實作當中，利用 VC++ 和 VB 自行撰寫幾個功能不同的軟體元件，加上再利用一些現存的軟體元件，我們實作出一個整合不同語言軟體元件的系統。
3. 提供了一個可再利用的合作 infrastructure：我們建立了一個合作 infrastructure，可以再利用於其他傳統合作環境中，以輕易的達成本論文中闡述的非同步合作關係。
4. 利用 Microsoft 的 Repository 作為儲存機制：本系統使用 Repository 作為多人合作發展環境的儲存機制。Repository 提供物件資料庫的架構且具有系統必需而關聯式資料庫所欠缺的特性。
5. 系統可移轉至各個相關產業：由於多人合作發展是各種產業所需，也是資訊高速公路產業的關鍵技術，故皆可利用本系統的架構來建構各產業的網站。

### 2 論文主體

#### 1 何謂多人合作發展環境

本計劃所闡述的“多人合作軟體發展環境”包含了兩個特點：一是“團隊發展 (Team Develop)”，縮短開發時程。一是“以元件再使用為基礎的軟體發展”。

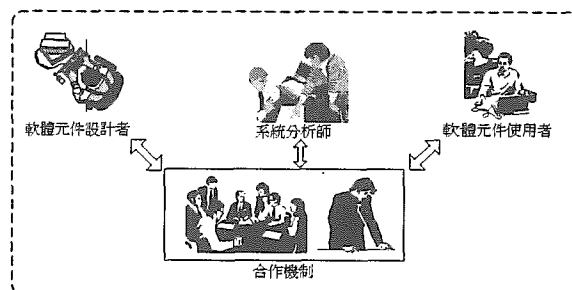


Figure 1 傳統的多人合作發展環境

在「以元件再使用為基礎的軟體發展」過程中，可以將程式設計師分為兩種型式：一為軟體元件設計師，另一為應用程式設計師。在這個多人合作發展環境下，成員之間的互動可能有下面幾種情況：

- 系統分析師將系統規格書交給應用程式設計師。
- 應用程式設計師將所需的軟體元件規格告知軟體元件設計師。
- 軟體元件設計師將新設計出的軟體元件規格公佈出來讓所有的應用程式設計師得知。
- 應用程式設計師對某一個軟體元件的使用上有所問題，而必須詢問該軟體元件的設計者。
- 應用程式設計師對發現某一個軟體元件有一個臭蟲，而必須告知該軟體元件的設計者。
- 多人合作發展環境成員們的進度報告，管理者以此掌握軟體發展的進度。

我們根據 Figure 1 的架構，提出一個成員互動的例子：某一軟體元件使用者 U 正使用一個個的軟體元件撰寫其應用程式時，他可能需要一個具有特定功能的軟體元件，那他可以在開會的時候，將他所需的軟體元件之規格書交給軟體元件設計者 D，詢問是否已有這種軟體元件存在可供其使用或是直接要求 D 為其設計一個新的軟體元件。當 D 找到或是設計出 U 所需的軟體元件時，他可能必需打電話告訴 U 來取得這個軟體元件，使 U 得以繼續其設計工作。這個例子就是一個典型的多人合作發展環境所可能發生互動的情形，例子中的合作機制就是：“利用電話連絡”，而其中我們所要加以強調說明的是：

- 多人合作發展環境的各個成員可能必需位於同一家公司的建築物當中，如 Figure 1 虛線所圍成的範圍。
- 成員工作的時間，就是大家共同的上班時間。
- 成員間彼此的互動要求必須對方在場。

以上三點的確會限制多人發展環境的效率與彈性，所以我們提出了這種環境的擴充與改進，能有效的解決上面三點限制。

## 2 成員散佈於全世界的多人合作發展環境合作發展環境

應用 WW 來建構多人合作發展環境將有下列優點：

- 通透性 (Transparency)：WW 是於 Internet 上通行的一種應用，而 Internet 容許多種傳輸協定共存，所以 WW 具有網路通透性 (Network transparency)；使用一致的文件標準— HTML 以及標準的文件存取協定— HTTP，讓各種平台，各種作業系統的使用者皆能使用 WW 上的資源及服務，因而具有平台、作業系統通透性 (Platform and O.S. transparency)。
- 介面一致 (Identical U)：透過 HTML 格式的規範，各種作業系統的使用者都能看到網頁外觀，擁有一致的顏色、字形、座標等使用者介面屬性，並能用相同的滑鼠驅動方式來操作使用者介面。
- 放置軟體元件 (Wedging components)：自從 HTML 3.0 標準後，已經允許網頁設計者利用 “Object” 標籤來指定軟體元件在網頁中的位置、大小，以及軟體元件的 URL 等，並可利用 Script 語言來驅動軟體元件之執行，如此可大幅提供了網頁的互動性。
- 檔案上傳及下載 (File upload and download)：使用者透過 URL (Uniform Resource Locator) 的指定，可以提供多人合作發展環境成員利用檔案的上傳及下載來達到資料交換的目的。

如果將上節所敘述的多人合作發展環境的概念加以擴充，利用目前非常普遍的 Internet、Intranet 與全球資訊網 (World Wide Web) 來提供多人合作發展環境中的合作機制，則傳統多人合作發展環境的種種限制將得以解除。如 Figure 2，我們計畫建立一個 “網站應用程式” (Web Application) 來提供多人合作發展環境成員一個不限時間、地點的合作機制。我們稱此網站應用程式為 “多人合作機制網站”。

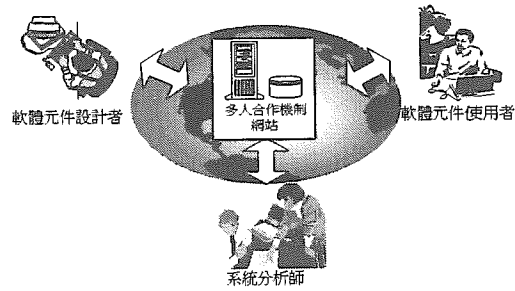


Figure 2 經 WW 擴充過後的多人合作發展環境

## 3 架構分析

如 Figure 3，這時擴充過後的多人合作發展環境是一個典型的 Three-Tiered 應用程式，整個系統可以分成下列兩個部分：

- 客戶端應用程式 (Client side application)：這個部分可以是散佈於全世界的任何一台連接上網際網路的一臺電腦，電腦上安裝一個具有執行 ActiveX Control 能力的瀏覽器。
- 網站應用程式 (Web application)：這個部分就是前述的多人合作機制網站，且可以再細分為下面兩個部分：

一個 WW 伺服器：經由 WW 伺服器所發行的一系列網頁構成了網站應用程式的使用者介面，而透過網頁中的 script 語言去控制軟體元件，再經由該軟體元件去存取後端的資料庫系統，並以這些自伺服端的資料庫取出的資訊動態地產生網頁。

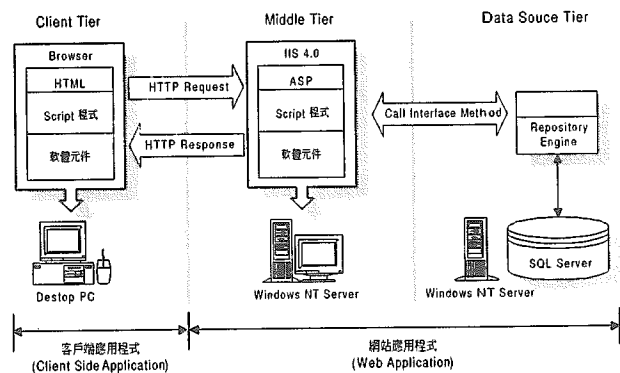


Figure 3 擴充過後的多人合作發展環境之架構

一個資料庫系統：用來儲存合作機制中成員間需要交換的各項資訊，如：現有軟體元件的各項規格、軟體元件使用者所需軟體元件的規格書、成員進度報告...等。比較特殊的地方是，系統的這個部分，我們採用的是 Microsoft 的 “Repositor” 作為後端的資料庫系統，有別於一般的關聯式資料庫管理系統 (Relational DBMS; RDBMS)，Repositor 提供了許多特性可以滿足系統的需求，Table 1 是一份 Repositor 和 RDBMS 簡單的比較。

	查詢方式	物件導向的特性	其他
Repository	Navigation	物件具有屬性與方法	介面分享

Relational Database	使用 SQL 語言	物件只具有屬性	不具介面分享
---------------------	-----------	---------	--------

Table 1 Repositor 與 RDBMS 的比較

#### 4 系統額外需求

當一個多人合作發展環境擴充至如 Figure3 的架構，它必須包括下列的額外需求。

##### 4.1 系統加密

由於這時候，整個多人發展環境是公開於網際網路上，所以適當的加密措施是必須的。Figure 4 便是以 Figure 3 的架構為例的一個簡易的系統流程：

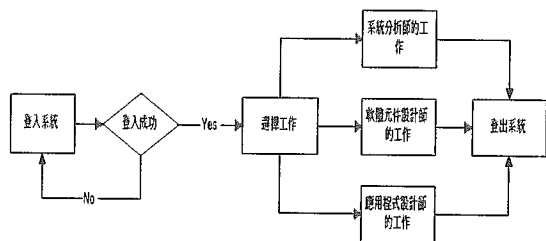


Figure 4 系統流程

相較於傳統多人合作發展環境，這時增加了身份查核的步驟，若是使用者沒有提供正確的密碼，便無法登入系統。而且多人合作發展環境成員有不同的職務與權限，對於他人的工作區域也應該加以保密，所以系統經由成員登入系統時所查核的身份，可以規範出該成員所能存取的範圍。

##### 4.2 非同步合作關係

在傳統多人合作發展環境下，各個成員間的互動都是在雙方彼此同時在場的情況下，作一即時的溝通，這是一種“同步”（synchronous）的訊息傳遞形式和另一種“非同步”（asynchronous）訊息的傳遞形式，我們可以把它們分別想像成日常生活中的“打電話”與“寄信”。假如多人合作發展環境是位於同一個辦公室或是同一棟建築物之中，前者是比較有效率的，因為我們可以馬上得到回覆；但是考慮到如果合作機制變成一個網站應用程式，它的使用者可能是散佈於全世界，而且沒有辦法要求訊息的接收者一定要在電腦（或電話）旁邊；再者可能會有一對一或者一對多的訊息發送情況，所以本系統是採取一個非同步的訊息傳遞形式，而訊息傳播的方法，就是如同 Figure 5。

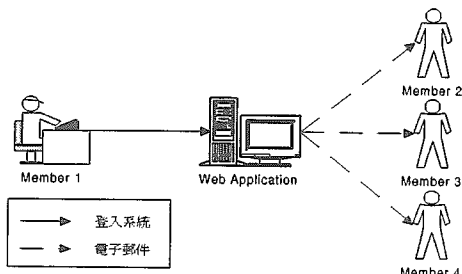


Figure 5 非同步的訊息傳播的方法

在 Figure 5 中，欲發送訊息的成員登入多人合作機制網站後，到相關的網頁填寫想要發送的訊息後，系統便會對應該接受訊息的成員以發送電子郵件（electronic mail）的方式來通知他們。使用電子郵件這樣的主動通

知的方式有下面幾個好處：

- 成員不須定期登入網站，查看是否有人發送訊息給他。
- 電子郵件的內容可以是簡單的訊息摘要和一個指向訊息詳細內容的 URL，成員可以由此自行過濾對其有用的訊息，而不是全盤的接收大量的所有訊息。
- 不限訊息發送或接受的時間與地點。

由於系統不限定各個成員登入的時間，而且使用電子郵件來傳佈訊息，故訊息的發送與接受並非同一時間，如此便形成了一個非同步的合作關係。

##### 4.3 元件版本通知與臭蟲通報之交易

一個軟體元件管理網站，對於其中儲存的某一現有軟體元件，在得知該軟體元件的設計者已提供了一個新的版本之後，系統有能力取得所有使用舊版本軟體元件的使用者資料，然後各發一封電子郵件給他們，告知他們曾經下載的某一軟體元件已經有新版發行，請他們可以連結上網站下載新版的軟體元件。這便是前述的非同步合作關係的一個實例，由於這是主動式的由系統通知，使用者不必要定期登入網站，察看是否有新版軟體元件發行，這項特點可以減少軟體元件的使用者不必要的上網時間，同時使軟體元件設計者與使用者的互動更加緊密。

當軟體元件的使用者發現他們所使用的軟體元件有任何的臭蟲或使用上的疑問時，他們可以登入網站，對那個他們有疑問的軟體元件做一個“臭蟲通報”（bug reporting）的動作。對於每一隻通報的臭蟲，系統也會將其資訊記載於 repositor 之中；然後系統會發一封電子郵件給該軟體元件的設計者，而這封電子郵件的內容就是一個 URL。當軟體元件的設計者收到這封標題為“發現臭蟲”的電子郵件之後，他可以在使用郵件閱讀軟體觀看這封電子郵件內容的時候，只要使用滑鼠點一下這個 URL，就可以連接上軟體元件管理網站的“觀看臭蟲”網頁，察看臭蟲的描述或是由此回信給通報此臭蟲的軟體元件使用者。同樣的，這也是非同步合作關係的一個實例。

#### 5. 系統構架分析及相關技術探討

##### 5.1 系統構架分析

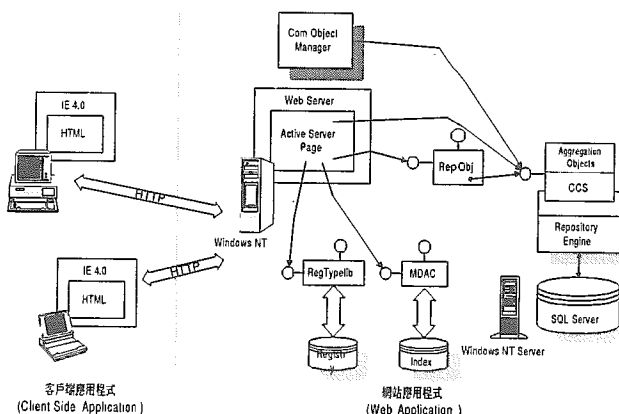


Figure 6 系統架構

在這個架構下，我們所實作的包括下面幾個部分：

1. COM Object Manager: 這是一個可單獨執行的應用程式，用來管理、檢視及存取系統用來儲存元件合作環境成員間交換的各項資訊的 repositor（以下簡稱“系

統 repository ”)。

- 軟體元件管理網站的 Tool Information Mode (TIM) : 由於整個軟體元件管理網站就是一個 repository-aware 工具 (Tool), 所以我們必須設計一個 Tool Information mode 來明白的定義這個工具所能處理的資料種類。當 TIM 設計好以後, 各種工具即可根據這個 TIM 所定義的型態來 populate 資料。從另一個角度去看, “設計 TIM” 這項工作就像是 RDBMS 中 “設計綱目 (schema)” 的工作。
- 數個軟體元件以及讓 ASP 或其他的軟體元件所使用的一些 In-Process ActiveX Code Component。如下 Table 2 就是這些軟體元件簡單的列表。

軟體元件名稱	功能說明	實作環境 / 語言
Cooperation Components (CCS)	實現事件機制的介面實作物件。	VC++ / ATL
RegTypeLib	可以登錄軟體元件及讀取軟體元件的型態資訊庫 (type library)。	VC++ / ATL
Populate Respositor Object (PRO)	可 populate 資料到 respositor 之中。	Visual Basic
Aggregation Objects	聚合 respositor 物件的 ActiveX Component。	VC++ / ATL

Table 2 系統所實作之軟體元件

- 軟體元件管理網站：包括一系列架設於 Microsoft Windows NT Server 4.0 的 Internet Information Server 的 HTML 和 Active Server Page (ASP) 網頁。如 Figure 7 所示。

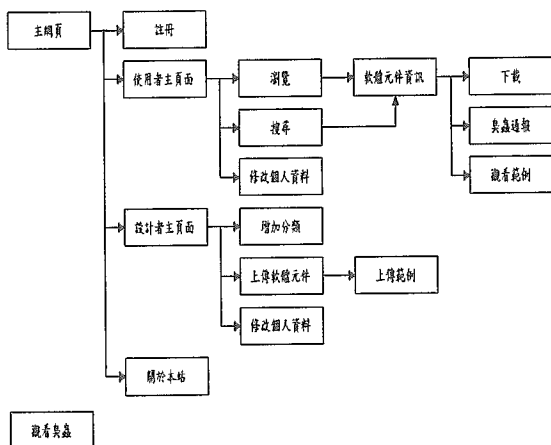


Figure 7 軟體元件管理網站之架構

## 5.2 使用之相關技術

### ● Component Object Model (COM)

Microsoft 的 Component Object Model (COM) 是 Microsoft 物件架構的基礎, 也是其用來實現軟體元件的機制。它定義了一個二元標準用來描述元件對元件之間的一種 “早期繫結” (Early-bound) 的呼叫協定, 這種協定是與語言無關的; 也就是說: 用一種語言撰寫的軟體元件可以無縫地呼叫用另一種語言撰寫的軟體元件。

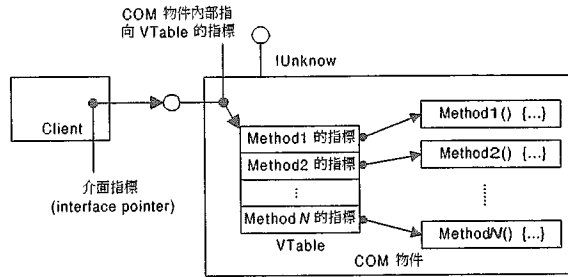


Figure 8 COM 物件的介面指標

### ● Automation

Automation 是實現 “晚期繫結” (Late-bound) 叫用 COM 物件的機制。Automation 的核心就是一個名為 “IDispatch” 的 COM 介面, 和一般的介面不同的是: 它所擁有一個 method — Invoke, 我們可以利用這個 method 去呼叫其他的 method, 並傳入所需的任何參數。要達成這個動作, 實作 IDispatch 的物件發展者必須標明可供叫用的 method, 也就是定義 “dispatch 介面” (dispatch interface; dispinterface)。

- 使用 IDispatch::Invoke 叫用 dispatch 介面中的 method。

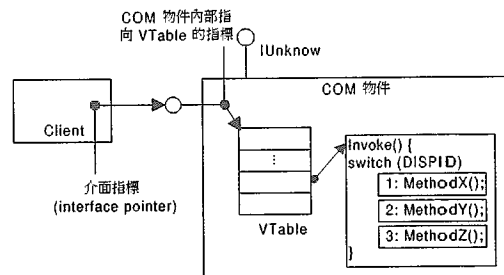


Figure 9

- 同時利用 vtable 與 IDispatch 提供其 method 的雙重介面。

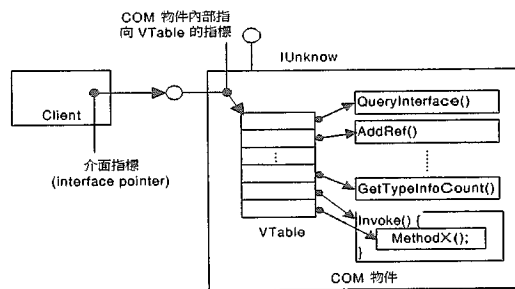


Figure 10

### ● Respositor

Microsoft Respositor 是一個物件導向的資料儲存機制, 它的主要構成部分包含了下面兩項:

- 一組 Active 介面, 程式發展者可以使用這些介面來定義 “資訊模式” (Information Model)。
- 一個 repository engine, 是用來儲存上述的 “資訊模式”。這個 repository engine 是架構於 Microsoft SQL Server 或 Microsoft Jet 之上, 也就是說: 真正資料儲存的地方還是傳統的關聯式資料庫。

Respositor 提供了一個以 COM 和 Automation 為基底的物件導向資料庫 (OODB), 其架構如 Figure 11 的架構。

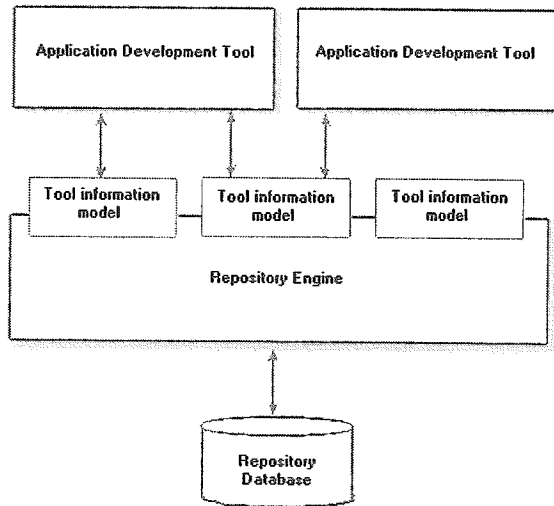


Figure 11 Repository 的架構

### ● Active Server Page

Active Server Pages (ASP) 是一個伺服器端 scriptin 環境，而我們可以利用這個環境去建立及執行動態的、互動的“網站應用程式” (Web Application) 而不是像傳統網站一樣，只是發行其內容。

### ● Microsoft 的資料存取元件 (Data Access Components; MDAC)

包含下列元素

- ☐ ActiveX Data Objects (ADO)。
- ☐ Remote Data Service (RDS)。
- ☐ Microsoft OLE DB Provider for ODBC。
- ☐ Open Database Connectivity (ODBC)。
- ☐ ODBC drivers for (Microsoft SQL Server, Microsoft Access and other desktop databases, and Oracle databases)。

### ● Dynamic HTML

DHTML，是使用一個物件模型 (object model) 擴充現有的 HTML，而這個物件模型允許 Script 語言或一般的程式語言去改變網頁元素 (也就是一個個的物件) 的樣式 (style) 與屬性或甚至替換已存在的元素。換句話說，針對網頁上的每一個 HTML TA 均是一個 COM 物件 (ActiveX Control)，利用客戶端之 Script 程式，直接改變 TAG 物件之屬性或呼叫其方法，IE 4.0 瀏覽器即時反應其 TAG 之外觀與行為。

### 3 結論

本論文我們提出一個利用現有的 WW 網站架構來擴充多人合作發展環境的構想，使得環境中的成員可以散佈於全世界，且不論任何時間，都可以利用我們所提出的“多人合作機制網站”，來完成以往的合作動作。然後更進一步的，我們實作了一個“以軟體元件為中心的多人合作發展環境”來驗證這個構想，而整個環境的核心就是一個軟體元件管理網站，它扮演著環境中合作機制的角色，透過軟體元件管理網站，這個以元件為基礎的軟體發展環境能夠使成員更方便也更有效率地達成工作。

在系統的實作過程中，我們建立了一個可以在其他環境再利用的基礎建設——一個名為 Cooperation Components (CCS) 的 infrastructure，透過 CC 的再利用，其他的多人合作發展環境可以輕易的達成本論文中闡述的非

同步合作關係。

既然元件軟體將會成為主流，故本系統可以朝下面幾點發展，使得其更具實用價值與競爭性：

- 提供更多的功能：既然軟體元件管理網站是扮演一個以軟體元件為中心的多人合作發展環境中合作機制的角色，它可以提供更多的功能，例如：
  - 元件使用者可以透過軟體管理網站提出它們所欠缺的軟體元件的規格，同樣經由主動通知，讓元件設計者依據使用者的需求設計出對使用者真正有用的軟體元件。
  - 提供線上的交易：由於上網購物已是目前的趨勢，故軟體元件管理網站可以提供線上的交易，讓使用者可以使用信用卡付款，如此一來廠商會更有意願透過網站來販賣他們所發行的軟體元件。
  - 會員制：系統可採用會員制，任何使用者經過一定的程序認證後，即可成為系統的會員，而對於會員，系統可以主動提供更多的資訊與服務，而且可以有助於系統的管理。
- 更豐富的軟體元件資訊：如果針對 COM 物件這種軟體元件來說，我們所提供的資訊還是有限，不過我們也實作出可以到 COM 物件的 type library 去讀取其資訊的能力。這一個部分的功能將來可以發展的更為強大，也就是不單只是 CoClass 與介面的名稱，應該連介面中的方法、方法的參數都可以詳列於元件的資訊頁面中，以提供使用者更詳盡的軟體元件資訊。
- 提供更完善的系統加密：由於系統目前處於原型的階段，除了密碼的比對之外，系統並沒有一般商業網站所具備的加密特性。所幸的是，目前網站的加密措施已有相當多的現成技術可以使用，所以這部分也是可以再加以研究與探討，進而去補強。

國科會 87 年度研究計劃

編號：NSC 88-2213-E-011-046

參考文獻：

- [1] <http://www.microsoft.com/com/>
- [2] <http://www.microsoft.com/Repository/>
- [3] Kraig Brockschmidt, "Inside OLE Second Edition", Microsoft Press, Redmond, WA, 1995
- [4] David Chappell, "Understanding ActiveX and OLE", Microsoft Press, Redmond, WA, 1996
- [5] Philip A. Bernstein, Brain Harry, Paul Sanders, David Shutt and Jason Zander, "The Microsoft Repository", Proceedings of the 23rd VLDB Conference Athens, Greece, 1997
- [6] Berenson, H., P. A. Bernstein, J. Gray, J. Melton, E. O'Neil and P. O'Neil, "A Critique of ANSI SQL Isolation Level", Proc. ACM SIGMOD 1995, ACM, N.Y
- [7] Bernstein, P.A., "Repositories and Object-Oriented Databases", Proceedings of BTW '97, Springer, March 1997, pp. 34-46.
- [8] Bernstein, P.A. and U. Dayal, "An Overview of Repository Technology", International Conference on Very Large Data Bases, Morgan Kaufmann Publishers, San Francisco, 1994, pp. 705-713.
- [9] Cattell, R.G.G., T. Atwood, D. Barry, J. Duhl, J. Eastman, G. Ferran, D. Jordan, M. Loomis and D.

- Wade, "The Object Database Standard: ODMG-93", Morgan Kaufmann Publishers, San Francisco, CA 1995.
- [10] Constantopoulos, P., M. Jarke, J. Mylopoulos and Y. Vassiliou, "The Software Information Base: A Server for Reuse", VLDB Journal, 4 (1995), Boxwood Press, Pacific Grove, CA, pp. 1- 43.
- [11] Geiger, K., "Inside ODBC", Microsoft Press, Redmond, WA, 1995.
- [12] Gray, J., R. Lorie, G. Putzolu and, I. Traiger, "Granularity of Locks and Degrees of Consistency in a Shared Data Base", in Readings in Database Sys, 2nd Edition, Chapter 3, Michael Stonebraker, Ed., Morgan Kaufmann 1994 (originally published 1977).
- [13] "Unified Modeling Language Resource Center", Rational Corp., <http://www.rational.com/uml>.
- [14] Rogerson, D., "Inside COM", Microsoft Press, Redmond, WA, 1997
- [15] Wakeman, L. and J. Jowett, "PCTE – The Standard for Open Repositories", Prentice-Hall, '93.
- [16] Zdonik, S.B., and D. Maier, "Readings in Object-Oriented Database Systems", Morgan Kaufmann Publishers, San Francisco, 1990.
- [17] "What is Microsoft Repository?", Microsoft Corporation, 1996-1997
- [18] "Microsoft Repository Home Page", Microsoft Corporation, 1997  
<http://www.microsoft.com/repositor/>
- [19] R. Elmasri and S. B. Navathe, "Fundamentals of Database System", The Benjamin/Cummings Publishing Company, Redwood, CA, 1994
- [20] "Microsoft Developer Network Library – July 1997", Microsoft Corporation, 1997
- [21] "Site Builder Network", Microsoft Corporation, 1997 <http://www.microsoft.com/sitebuilder/>
- [22] "Internet Explorer 4.0 Home", Microsoft Corporation, 1997 <http://www.microsoft.com/ie/ie40/>
- [23] "Site Builder Network Specs & Standards", Microsoft Corporation, 1997  
<http://www.microsoft.com/standards/>
- [24] S. Henninger, "An Evolutionary Approach to Constructing Effective Software Repositories", ACM Transaction on Software Engineering and Methodology, Vol. 6, No.2 April 1997, Page 111-140
- [25] Pountain, Dick, "The component enterprise", Byte, May 1997 v22 n5, p93(4)
- [26] Krill, Paul , "Repository is key to Oracle data strategy", InfoWorld, Oct 13, 1997 v19 n41 p1(2)
- [27] Linthicum, David S., "Microsoft Repository 1.0", DBMS, June 1997 v10 n7 p28(3)
- [28] Linthicum, David S., "Another tool, another repositior ", DBMS, May 1997 v10 n5 p26(3)
- [29] Hanna, Mary, "OO Development's silent partner ", Software Magazine, Feb 1997 v17 n2 p45(2)
- [30] King, Julia, "Parts, design repository to cut development time", Computerworld, Jan 13, 1997 v31 n2 p59(2)
- [31] Bernstein, Philip A. , "The repository: a modern vision", Database Programming & Design, Dec 1996 v9 n12 p28(8)
- [32] "Sybase: adaptive component architecture", Computer Reseller News, Oct 6, 1997 n757 p154(4)
- [33] Teti, Frank , "Open the door to components NOT objects", HP Professional, Oct 1997 v11 n10 p16(4)
- [34] Spitzer, Tom, "Component architectures", DBMS, Sep 1997 v10 n10 p56(6)
- [35] Pountain, Dick, Montgomery, John, "Web components", Byte, August 1997 v22 n8 p56(9)
- [36] Frey, Anthony, "Is DCOM truly the object of middleware's desire?", Network Computing, July 15, 1997 v8 n13 p98(5)
- [37] Pountain, Dick "The component enterprise", Byte, May 1997 v22 n5 p93(4)
- [38] Roy, Mark, Ewald, Alan, "Inside DCOM", DBMS, April 1997 v10 n4 p26(6)
- [39] Shepherd, George, Wingo, Scot, "Windows and distributed components", Software Development, April 1997 v5 n4 p39(5)
- [40] Moriarty, Terr , Mandracchia, Christine , "Heart of the warehouse", Database Programming & Design, Nov 1996 v9 n11 p70(3)
- [41] Rennhackkamp, Martin, "Building a DBA repositior system", DBMS, Jan 1996 v9 n1 p59(5)
- [42] 謝景順, "一個以物件基底為基礎的自動化物件資訊庫", 臺灣工業技術學院電子工程技術研究所碩士學位論文, 民國八十六年六月