

支援軟體維護之 Web 資料庫軟體模型 A Web Database Application Model for Software Maintenance

廖琬洲* 陳俊良* 許嘉麟* 王豐堅* 宋麗麗*
Hsian-Chou Liao Jiun-Liang Chen Chia-Lin Hsu Feng-Jian Wang Li-Li Soung

*國立交通大學資訊工程系
Department of Computer Science and Information Engineering
National Chiao-Tung University, Hsinchu, Taiwan
{hcliao, jlchen, fjwang, glshu}@csie.nctu.edu.tw

*財團法人資訊工業策進會軟體工程實驗室
Software Engineering Laboratory
Institute for Information Industry, Taipei, Taiwan
sy666@iii.org.tw

摘要

Web 資料庫軟體的程式模組幾乎是透過檔案來區分，可能由數百個、甚至數千個檔案所組成，導致整個軟體的作業流程及運作方法是分散在這些檔案上。此外大部分的程式並沒有完整的設計文件，增加了 Web 資料庫軟體維護的困難。本篇文章中，我們提出 Web 資料庫軟體模型協助維護的需求，此模型利用三種圖形-超鏈結圖、實體關係圖及資料流程圖-來表示程式架構。這些表示法可以為 Web 資料庫軟體之結構性影響分析與資料性影響分析的基礎，協助解決軟體維護上的困難。此外，我們在本文提出一個 Web 資料庫軟體維護工具的系统架構。

關鍵字：全球資訊網、軟體維護、超鏈結、資料庫。

Abstract

The program modules of the Web database applications consist of files, of which the number can be large. The interoperations among these files could be sophisticated, and most of them are lack of design documents. These may complicate the maintenance of such applications. In this paper, we present a model for the maintenance of the Web database applications. The model is composed of three diagrams, a hyper-link diagram, an entity-relationship diagram, and a data-flow diagram to represent the structure of application programs. During the maintenance, the analyses of hyper-link structure and database changes can be performed through these diagrams. In addition, we design the architecture of an environment based on this model to facilitate maintenance of the Web database applications.

Keywords: WWW, software maintenance, hyper-link, database.

1. 簡介

網際網路(Internet)技術的演進使得連接廣域網路的電腦逐漸普及，而全球資訊網(WWW: World-Wide Web)提供了一個跨平台的資訊交流管道，加速了網際網路上的資訊與知識的交流。WWW 的運用軟體主要是透過網頁(Web page)讓使用者經由網際網路來操

作。製作網頁通常是用 HTML (Hyper-Text Markup Language) [3]語言撰寫，HTML 主要是定義在瀏覽器上可以用超文字結構(hyper-text structure)方式呈現的資訊，這樣的結構可以讓使用者很容易的瀏覽這些資訊。資料庫[9]是一般應用軟體相當普遍重要的一部份，但是目前的 HTML 並未包含資料庫存取方式的描述。為了網頁上提供資料庫的功能，許多的公司提供了特定的作法。例如，Microsoft[®]所提出的 Internet Database Connector(idc)及 HTML Extension (htx) [5]。這些擴充雖然讓網頁具有更強大的資料庫存取功能；相對地，也使得網頁的製作更為複雜。

Web 上的應用軟體與一般應用軟體最大的不同是，程式模組幾乎是透過檔案來區分，因此 Web 上的軟體大都是由數百個、甚至數千個檔案所組成，導致整個軟體的作業流程及運作方法是分散在這些檔案上。就軟體的製作與維護而言，Web 上的應用軟體基本上比一般的應用軟體要複雜的多。Internet 盛行之後，也有許多的資料庫系統被引進來製作 Web 的應用軟體。這些程式大部分並沒有完整的設計文件，增加了 Web 資料庫軟體維護的困難。另一方面，就維護的動作而言，並未有工具針對 Web 資料庫軟體提供支援，在現有的網頁製作工具(例如：FrontPage、NetObjects Fusion 等)以顯示網頁的超鏈結情況為主，當引進資料庫之後，單單使用超鏈結結構已經不足以應付種種維護上的動作；例如：某個資料庫表格(table)中某欄位(field)之型態，其長度原先為 20 字元，但是現在根據需要改為 30 個字元，從超文字結構中無法知道有哪些檔案需要修正。

本篇文章針對 Web 上資料庫軟體，探討維護動作上的需求，根據這樣的需求，提出可用於表達 Web 資料庫軟體的模型。此模型包含三種表示法：一是超鏈結圖(HLD: Hyper-Link Diagram)，用以表示超媒體系統中超鏈結之關係；二是實體關係圖(ERD: Entity-Relationship Diagram)，用以表示資料庫的規格；三是資料流程圖(DFD: Data-Flow Diagram)，用以表示資料庫的操作行為。這些表示法是根據 Web 資料庫軟體的原始程式建構出來的，無須原來系統設計的規格文件。在軟體維護過程中，此模型可以做為結構性與資

料性方面影響分析之基礎。在結構性影響分析方面的所提供的功能有未附著鏈結之分析、無法瀏覽網頁之分析與限制性瀏覽路徑之分析；此類分析是針對 Web 資料庫軟體的程式架構需異動時，讓程式設計者可以知道整個超鏈結架構所受到的影響的部分。在資料性影響分析方面可分為架構異動分析與資料異動分析，其目的是讓程式設計者能瞭解資料庫架構異動與資料來源改變會對其他部分造成那些影響。這些分析可以協助解決軟體維護上程式異動的困難。此外，利用此 Web 資料庫軟體的模型，我們也將發展一網際網路上資料庫軟體之分析工具，以便於 Web 資料庫軟體的維護。

2. Web 上資料庫應用軟體的維護需求

依據微軟所提的方法[5]，Web 資料庫軟體的撰寫方式如下，當網頁製作人員要存取資料庫的內容時，首先必須將這個存取動作使用結構化查詢語言(SQL: Structured Query Language[6])表達出來並儲存在一 .idc 檔案中¹，當瀏覽器中透過超鏈結而連結到此 .idc 檔時，其中的 SQL 敘述會直接送給 SQL Server 來執行，而 SQL 執行後所得到的資料將會被合併到指定的 .htx 檔案，合併結果會以 HTML 的格式送回給瀏覽器。Web 資料庫軟體所包含程式的檔案數目會因此大量增加。

軟體維護(maintenance)過程中，通常需要對某些程式部份進行異動(change)的動作，這些異動基本上會影響到其他程式中許多相關的部份，這些被影響到的部份往往也必須一併異動才可以使得軟體可以順利的執行下去。維護動作中的困難之一是在於判斷(identify)出異動的種類以及找出對於受到影響的相關程式[7]。Web 應用軟體大多是由一堆相關檔案組成(有些由數百至數千個檔案組成)，面對這些檔案尋找因為異動而需要跟著變更的部分是件相當費時與耗人力的工作，隨著網頁的複雜與資料庫功能的提供，Web 軟體檔案的數目會增多，維護上的問題會更為嚴重。

一個 Web 資料庫應用軟體的異動可分為二類：

1. 網頁異動：對於網頁的異動，包含資料庫的應用軟體與一般沒有資料庫的應用軟體基本上的維護需求是類似的。其異動可分為下面二項：
 - 增加或刪除網頁：當增加一網頁時，程式設計師只需知道此網頁對應的檔案是否已經被包含在原有的「超文字結構」當中，且這個新檔案中的超鏈結是否都連結到現有存在的檔案。而刪除一個網頁時，程式設計師需要知道原有超鏈結是連結到這個檔案以及有哪些檔案被這個檔案所連結，看看這些超鏈結是不是要刪掉或是否連結到別的檔案，而原先被連結到的檔案是不是要刪掉或是由別的檔案來連結之。

- 增加或刪除超鏈結：當某個檔案加入了一條超鏈結，程式設計師需要知道鏈結的目的是不是存在的。而刪除了一條超鏈結時，程式設計師需要特別處理原先這條鏈結的目的地，確定有別的鏈結可以連到這個目的地。
2. 資料庫異動：資料庫軟體經過一段時間後，往往需要更動資料庫才能滿足資料量與資料型態的增加。維護動作中，資料庫的異動對原來的應用軟體所造成的衝擊也最大。下面就四種不同的資料庫異動分別討論其維護需求：
 - 刪除表格(table)：當刪除一表格，程式設計師需要知道有哪些檔案原有對這個表格有存取的動作，這些部份都需要跟著修正。
 - 增加或刪除欄位(field)：當在某個表格中增加或是刪除一個欄位時，程式設計師需要知道哪些檔案對於這個表格有存取動作，便於將新增的欄位加入檔案中或是由檔案中將此欄位刪除。
 - 更改欄位的資料型態：修改了某個欄位的型態之後，程式設計師需要知道這個欄位在哪些檔案中被使用到，然後修正之。
 - 增加或刪除表格關係(relationship)：表格關係的增加或是刪除所牽涉到的是表格之間的合併(join)動作。當增加一個表格關係時，對於這個關係所牽扯到的表格，程式設計師需要知道哪些檔案用到這二個表格，便於檢查是否要將 join 的動作加入。而刪除一個關係時，對於這個關係雙方的表格，程式設計師需要知道哪些檔案原先曾對這二個表格進行合併的動作，以便於檢查與修正。

由於 HTML 所撰寫的程式不需經過編譯，直接被瀏覽器讀取。因此，許多語法及語意上的錯誤與不當的超鏈結等，若沒有適當的工具分析，不易被檢查出來。在現有的網頁製作工具(例如：FrontPage, NetObject Fusion 等)，都以顯示網頁的超鏈結情況為主，並未提供維護時部份程式異動分析的支援，與整個相關程式檔一致性分析。

3. Web 資料庫軟體模型

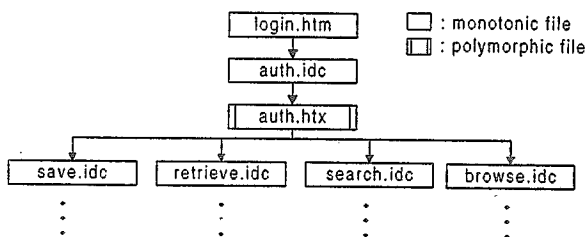
針對 Web 資料庫軟體，我們提出一模型來描述系統的組織架構以協助軟體的維護。此模型主要包含三種圖示：超鏈結圖(HLD: Hyper-Link Diagram)，實體關係圖(ERD: Entity-Relationship Diagram)，與資料流程圖(DFD: Data-Flow Diagram)。

3.1 超鏈結圖

超鏈結圖(HLD)是用來表示 Web 資料庫軟體的透過網頁所呈現之超鏈結結構(hyper-link structure)，每個構成網頁的檔案由一個節點代表，節點與節點之間的有向線段表示超鏈結關係(可視為瀏覽的路徑)。節點又可區分為兩類：單調節點(monotonic node)與多形節點(polymorphic node)；單調節點是指其所代表的網頁在任何情況下都顯示一樣的內容，而多形節點是指這網頁的顯示內容視執行時所傳入的參數值決定，這

¹ 本文所提 .idc、.htx、.htm、.map 檔案分別是指附檔名為 .idc、.htx、.htm、.map 之檔案。

種節點是在資料庫引進 Web 之後才會有的。

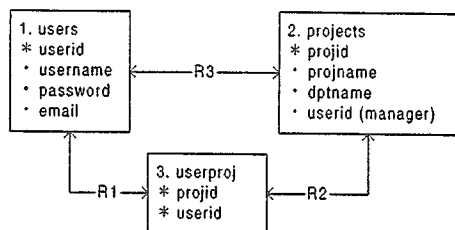


圖一：超鏈結圖的範例

圖一是超鏈結圖的一個例子，每個長方形代表一個節點。在圖中表示 auth.htx 是一個多形節點，其餘皆是單調節點。此例子是一個應用軟體，它有四種功能（存入、取出、搜尋及瀏覽元件），而使用者有二種等級：會員可使用四種功能；非會員只能使用搜尋及瀏覽功能。所以圖中使用者藉由 login.htm 填入代號及密碼等資料後，透過 auth.idc 做身份的確證，確證結果再送給 auth.htx 後，auth.htx 會決定是要同時顯示四個功能的超鏈結或是只有顯示到 search.idc 及 browse.idc 這二個功能的超鏈結。在圖一中將節點分為單調與多形之後，可以讓程式設計師更清楚的明白使用者可能的導覽模式(navigation model)。另外還有一點要說明，由於一個 Web 上應用軟體所包含的檔案並不單單只有 HTML 檔案，若是有的檔案有特定的語法來定義二個檔案之間的關係的話，這樣的檔案關係也需要顯示在超鏈結圖中。例如：在 .idc 檔的程式中會有 template 敘述指定一檔案，此檔中事先定義好與資料庫中之資料的合併方式，便於執行時合併資料之用。當一個 .idc 檔被另一個檔案透過超鏈結來讀取時，此檔中所定義的 SQL 敘述會先被執行，然後再叫用此 .idc 檔中 template 敘述所指定的檔案，與 SQL 敘述結果合併成為 HTML 檔，最後再傳送給瀏覽器顯示。因此，這樣的關係也會顯示在超鏈結圖中。在圖一中“auth.idc”的程式中 template 敘述所指定的是“auth.htx”檔案，所以有一條超鏈結線段由“auth.idc”連到“auth.htx”檔。

3.2 實體關係圖

實體關係圖[1]是用來表示關連式資料庫的規格，主要是由實體(entity)及關係(relationship)組成。關連式資料庫是由表格(table)組成，每個表格包含一組欄位(field or column)。在這裡的實體關係圖中，表格將會被視為實體，而欄位將被視為屬性(attribute)。



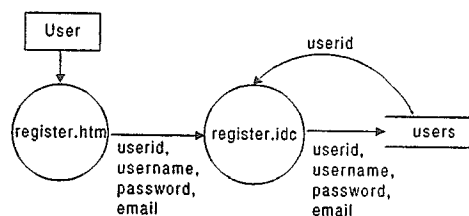
圖二：實體關係圖的範例

圖二是實體關係圖的範例，圖中共有三個實體 users, projects, userproj 及三個關係 R1, R2, R3，每個實

體中所包含的屬性都直接列在實體名稱下面，屬性前面有星號“*”標示代表此屬性是這個實體的主鍵(primary key 或 identifier)。例如實體 users 包括四個屬性 userid, username, password 及 email，其中 userid 是主鍵。

3.3 資料流程圖

資料流程圖[4]是用來表示應用軟體中資料的操作行為。一般的資料流程圖包含四種元素：外部實體(external entity)、資料儲存器(data store)、資料流(data flow)及處理元(process)。目前瀏覽器操作應用軟體的方式是無狀態性(stateless)的，所以 Web 上的應用程式中並沒有變數(variables)的使用。在資料流程圖中的外部實體通常是使用者，資料儲存器可以定義為實體關係圖中的所用到的實體（即資料庫中的表格），資料流的内容是某實體內的屬性值（即表格中欄位的資料）在不同的網頁間，而處理元代表超鏈結圖中的節點(即程式檔案)。



圖三：描述使用者註冊動作之資料流程圖

資料流程圖的範例如圖三所示，圖三所顯示的是一個註冊動作的資料流程圖，首先處理元 register.htm 取得由外部實體 User 所輸入的四個資料流 userid, username, password 及 email。輸入的資料接著送到另一個處理元 register.idc，register.idc 會判斷輸入的 userid 是否已經存在於資料庫，所以有一條資料流 userid 由資料儲存器 users 流向 register.idc，若是使用者輸入的是尚未存在的 userid 時，所輸入的使用者資料將存入資料儲存器 users 中。由圖三的例子中，可顯示透過資料流程圖將可以容易地追蹤資料庫中各個表格及欄位在檔案中被處理的過程。

4. Web 資料庫軟體模型之建構

根據前面所定義 Web 應用軟體模型的三種圖形後，本節將說明如何建構這三種圖形。這些建構方式是依據所涵蓋的檔案格式來做說明，由於其他的資料庫處理方式可能會透過不同的檔案格式來完成，需要額外的考量。本文所討論的建構方式是基於 Java scripts [2]與四種檔案類型：.htm (.html), .idc, .htx 及 .plx (.pl or .perl)檔案。

4.1 建構超鏈結圖

超鏈結圖的建構以組成 Web 軟體的程式檔案為節點，超鏈結是根據程式中有關超鏈結的敘述取得。在剖析原始程式時，可由下列規則得出超鏈結關係：

1. 在 .idc 檔中，根據“Template:”敘述所指定的 .htx 檔案為何，可得知一超鏈結由此 .idc 檔連到該 .htx 檔。例如：檔案 compbrow1.idc 程式中指定 template

如下：

Template: compbrow1.htx

將記錄有一超鏈結由 compbrow1.idc 鏈結到 compbrow1.htx。

2. 在.htm 檔或.htx 檔案程式中如有“<A href=”的敘述即為一超鏈結：

- 如果“<A href=”後面所接字串是.map 檔時，超鏈結的指向就是此檔中所指定的檔案。由於.map 檔定義在一個圖形的哪些部份按下去會有超鏈結，所以此檔中所指定的檔案即為超鏈結的目的地。例如：在檔案 projadd1.idc 中發現下面這行敘述：

```
... <A href="/DCMS/IMAGES/sysproj.map"> ...
而 sysproj.map 的內容為：
```

```
rect (6,4) (52,31) /dcms/sm/projadd1.idc
rect (66,4) (154,31) /dcms/sm/projadd1.idc
```

在例子中，將記錄有二條超鏈結由 projadd1.htx 分別連結到 projadd1.idc; projdel1.idc。

- 如果“<A href=”後面所接的不是.map 檔，則後面所接的目的地檔名即為超鏈結的目的地。例如：在檔案 compbrow1.idc 中發現下面這行敘述：

```
...<LI><A href="/scripts/dcms/common/compbrow2_1.idc"> 依照專案：</a>...
```

將記錄有一條超鏈結由 compbrow1.htx 鏈結到 compbrow2_1.idc。

3. 在.htm 檔或.htx 檔中若是發現 Java scripts 中（指包含在“<Script> ... </Script>”中的內容）有超鏈結“href”敘述的話，後面所接的參數為鏈結的目的地。例如：在檔案 compbrow1.htx 的 Java scripts 中發現下面的敘述：

```
... top.f4.subf2.location.href="/scripts/dcms/comcontent.idc?compid=...
```

將記錄有一條超鏈結由 compbrow1.htx 鏈結到 comcontent.idc。

4. 在.htm 檔或.htx 檔中若是發現“<form ... action=”的敘述，後面所接的參數為超鏈結的目的地。例如：在檔案 compsrch0.htx 中發現下面的敘述：

```
...<form name="menuform" action="/scripts/dcms/common/compscont.idc">...
```

將記錄有一條超鏈結由 compsrch0.htx 鏈結到 compscont.idc。

5. 在.htm 檔或.htx 檔中若是發現“<frame src=”的敘述，後面所接的參數為超鏈結的目的地。例如：在檔案 auth.htx 中發現下面的敘述：

```
...<frame src="/scripts/dcms/frame3_1.idc?userid=<userid%>">...
```

將記錄有一條超鏈結由 'auth.htx 鏈結到 frame3_1.idc。

單調節點與多形節點，也可由程式中分析得出。以下是區分此兩節點的方式：對於.htx 檔案，若是發現其超鏈結被包含在一個 IF-敘述（<%if...then%>...<%else%>...<%endif%>）區塊中，則這個.htx 檔案用一多形節點表示。例如，在 auth.htx 中出現：

```
...<%if TM EQ "Y" then%>
<FRAME SRC="/scripts/dcms/frame3_1.idc?userid=<userid%>"
NAME="f3" MARGINWIDTH=0
MARGINHEIGHT=0 SCROLLING="no">
<%else%>
<%if PM EQ "Y" then%>
<FRAME SRC="/scripts/dcms/pm/frame3_2.idc?userid=<userid%>"
NAME="f3"
MARGINWIDTH=0 MARGINHEIGHT=0 SCROLLING="no">
<%else%>
<%if SM EQ "Y" then%>
<FRAME SRC="/scripts/dcms/sm/frame3_3.idc?userid=<userid%>"
NAME="f3"
MARGINWIDTH=0 MARGINHEIGHT=0 SCROLLING="no">
<%endif%>
<%endif%>
<%endif%>...
```

則 auth.htx 將被設定為一個多形節點。

4.2 建構實體關係圖

實體關係圖的建構可從程式中對資料庫存取的 SQL 指令得出。實體的建構是以表格為單位，表格的欄位為其屬性，在剖析程式時，遵循下列方式得出。

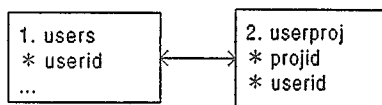
1. 程式中的 SQL 敘述已經有指明欄位名稱。例如，由“select userid, username from users”的敘述可知道名稱為 users 的表格至少包含二個欄位 userid 及 username。
2. 程式中的 SQL 敘述使用“update ...”的語法，敘述中會指出欄位名稱。例如，由“update users set username = '%username%', password = '%password%', email = '%email%' where userid = '%userid%'”的敘述中可知表格名稱為 users 中包含至少有四個欄位 username, password, email 及 userid。
3. 程式中的 SQL 敘述使用“insert into...”的語法，敘述中會指出欄位名稱。例如：由“insert into users (userid, username, password, email) values ('%userid%', '%username%', '%password%', '%email%')”的敘述中可知表格名稱為 users 中包含至少有四個欄位 userid, username, password 及 email。
4. 當某個.idc 檔中的 SQL 敘述是“select * ...”，並沒有指明要選出那個欄位，需要由對應的.htx 檔中來取得欄位。例如，SQL 的敘述為“select * from users”，其對應的.htx 內容如下：

```
...<%begindetail%>
<OPTION value="<userid%>"><userid%> (<username%>)
<%enddetail%>...
```

.htx 檔案內容中“<%begindetail%> ... <%enddetail%>”定義了 SQL 敘述的執行結果所得到的資料要如何合併成所需的 HTML 檔案。所以，“<%begindetail%> ... <%enddetail%>”的中間會指明哪些欄位要如何合併上去。例子中，從上面的.idc 檔可以知道有一個 users 的表格，再從.htx 可以知道 users 至少有二個欄位：userid 及 username。

實體關之間的關係可從程式中 SQL 敘述有二個表格的 join，即可知這二個表格有關係(relationship)。例如：SQL 的敘述為“select projid from users as U, userproj as UP where U.userid = UP.userid and U.userid = '%userid%'”，可知 users 及 userproj 這二個表格有關係，關係是建立在 users 的 userid 欄位與 userproj

的 userid 欄位之上，如下圖所示：



圖四：資料流程圖之範例

4.3 建構資料流程圖

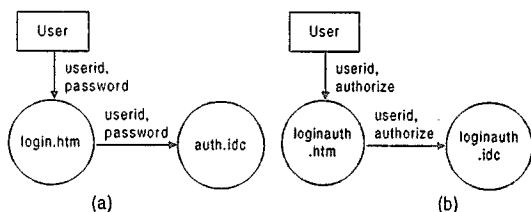
資料流程圖的建構可在剖析原始程式碼時，遵循下列規則得出。

1. 在.htm 檔案中若定義了一個表單(form) 區塊，也就在“<form...>...</form>”之間區域，且此區塊包含一起鏈結連到某個.idc 檔。如此可以得知有一個外部實體 User，送出資料流給這個.htm 檔，資料流的名稱來自.htm 中 form 所提供的各個輸入項目名稱(包括：<input type= “text”(or “hidden”, “password”, “checkbox”, “radio”)... ， <select ... 或是 <textarea ...)，同時這些資料流也會流入.idc 檔案中。例如，login.htm 的部份內容如下：

```

...<form name="menuform" action="/scripts/dcms/auth.idc"...
  <div align="center"><center><table border="3">
    <tr><td><font color="#FFFF00" size="4">使用者代號:</font></td>
    <td><font color="#FFFF00" size="4"><input type="text"
      size="20" name="userid"</font></td>
    </tr><tr>
    <td><font color="#FFFF00" size="4">使用者密碼:</font></td>
    <td><font color="#FFFF00" size="4"><input type="password"
      size="20" name="password"</font></td>
    </tr><tr>...
  </div></form>...
    
```

將產生如圖五中(a)的資料流程圖。



圖五：.htm 檔與.idc 檔間資料流程圖的範例

2. 在.htm 檔案中若定義了一個表單(form)區塊，且發現此區塊中有一起鏈結連到某個.idc 檔。如此可以得知有一個外部實體，送出資料流給這個.htm 檔，資料流的名稱來自一指定.idc 檔敘述之“?”後面所附的字串，欄位之間以“&”隔開，而且這些資料流也同樣流入.idc 檔案中。例如，loginauth.htm 的部份內容如下：

```

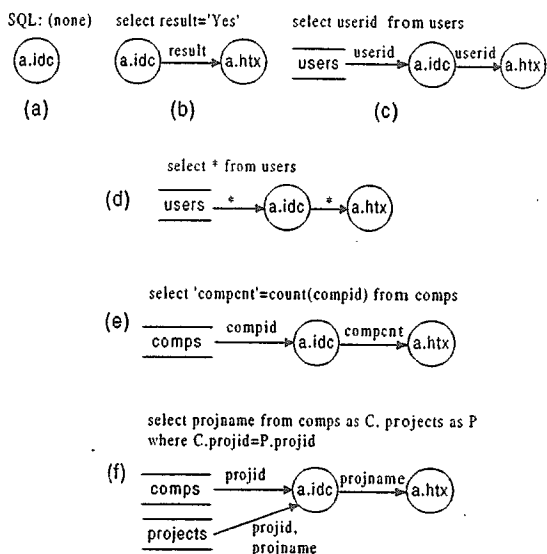
... top.f5.location.href="/scripts/dcms/sm/loginauth.idc?
userid="+userid+"& authorize="+authorize...
    
```

將產生如圖五中(b)的資料流程圖。

3. 在.idc 檔中 SQL 敘述中不是空的，亦即包含 select, insert, update 及 delete 為關鍵字的敘述。每個 SQL 依據下面不同的方式處理(假設為 a.idc)：

- SELECT：使用 select 可能的敘述共有六種，在圖六中共列出了六種情形的範例並說明如何將之轉換成資料流程圖。(a)表示.idc 檔中沒有 SQL 敘述，所以只有一個 a.idc 的處理元。(b)中的用法是使用 select 來傳出一個數值，所以，由一個資料流 result 從 a.idc 到 a.htx。(c)表示使用 select 來

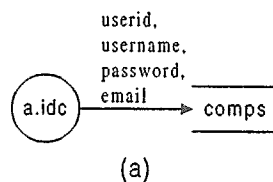
取出一個表格中的欄位，所以，會從資料儲存器 users 中送出一個資料流 userid 流入 a.idc 中，接著流入 a.htx 中。(d)表示從 users 中選出所有的欄位，和(c)類似，直接在資料流上標示星號“*”。(e)中 select 的用法是配合一些 SQL 的統計函數(count, max, average, etc.)，所以由資料儲存器中送出的欄位(compid)經過計算之後會變成另一個資料流(compent)送出。(f)表示二個表格進行 join 時的情形，例子中將 projname 歸入從 projects 流出的資料流的原因是透過查閱先前產生的實體關係圖而決定。



圖六：六種不同使用 select 之敘述轉換為資料流程圖的範例

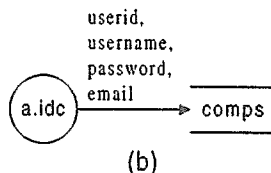
```

insert into users (userid, username, password, email)
values ('%userid%', '%username%', '%password%',
'%email%')
    
```



```

update users set username='%username%',
password='%password%', email='%email%'
where userid='%userid%'
    
```



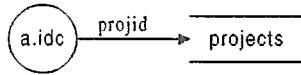
圖七：使用 insert 及 update 之敘述轉換為資料流程圖的範例

- INSERT 及 UPDATE：圖七中列出了這樣的 SQL 範例及轉換方式。在圖七中，(a)表示一個 insert 的轉換方式，a.idc 送出了 SQL 敘述中指定的欄位。(b)表示一個 update 的動作，雖然和 insert

的 SQL 敘述不相同，但是轉換出來的資料流程圖是相同的。

- DELETE:圖八列出使用 delete 的 SQL 敘述轉換為資料流程圖的範例。

```
delete from projects where projid=%projid%
```



圖八：使用 delete 之敘述轉換為資料流程圖的範例

4. 在.htx 檔中，除了使用上述的第 1 及第 2 條.htm 檔的規則來處理外，.htx 檔提供一個語法，可以拿到傳給叫用此.htx 檔之.idc 檔的參數值，這時，這個參數也需要加入.htx 檔的輸入資料流中。例如：在 frame2.htx 中發現這樣的内容：

```
... <td><input type = "button" name = "compsrch" value = "元件  
搜尋" onclick="loadframes  
( '/scripts/compsrch0.idc?userid=<%idc.userid%>',  
' /scripts/compsrch2.idc' )"></td>...
```

這段敘述中使用到 Java script 的語法，其對於一個按鈕提供一個 onclick 的事件 (event)，也就是當按了這個鍵之後，會去執行 loadframes 的副程式。其中使用了“<%idc.userid%>”，表示這個值是取自於之前流入 frame2.idc 的資料流，所以，userid 也需要加在流入 frame2.htx 的資料流中。

根據上面三種圖的建構規則，我們可以透過分析檔案的内容而自動化的將一個 Web 資料庫軟體用這三種表示法顯示出來。這些規則的適用範圍當然只限於涵蓋到的檔案格式，若是要能使這些規則還可以分析另一新的檔案格式，只需將現有的規則加以擴充即可。

5. 維護需求的支援

由前面所述之 Web 資料庫軟體維護過程中，程式設計者在執行異動時所需對程式的分析可分為結構性影響分析與資料性影響分析。Web 資料庫軟體模型可以做為分析基礎，達到支援維護需求的目的。

5.1 結構性影響分析

Web 資料庫軟體的結構性異動主要是新增或刪除超鏈結與檔案，這些異動經由超鏈結圖可以反映出相關受影響的部分。結構性影響分析可進一步區分為未附著鏈結之分析，無法瀏覽網頁之分析與限制性瀏覽路徑之分析。

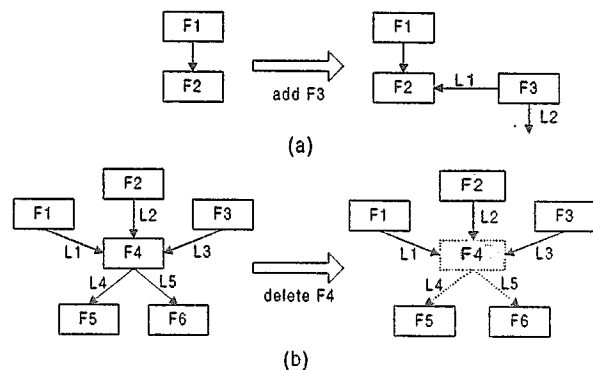
1. 未附著鏈結之分析 (Dangling Hyper-link Analysis)：對於 Web 資料庫相關程式檔案，新增超鏈結與刪除檔案造成某些超鏈結會指向是一個不存在的網頁，此類的超鏈結稱未附著鏈結。透過超鏈結圖，對每個節點的指向外部的超鏈結做檢查，即可找出未附著鏈結。
2. 無法瀏覽網頁之分析 (Unreachable Home Page Analysis)：刪除超鏈結與新增檔案造成使用者無法由進入點瀏覽到某些網頁。在超鏈結圖中，由進入點到此類的網頁是沒有任何有向線段所形成的路徑。只要從表是進入點的節點沿著對每一有向線段

搜尋相關節點，未被搜尋到的節點即是無法瀏覽網頁。

3. 限制性瀏覽路徑之分析 (Restricted Navigation Path Analysis)：新增檔案造成超鏈結時，因為增加了瀏覽路徑，可能會破壞依使用者權而有所限制的瀏覽方式，此種情況可以算是資訊外漏 (Information Leakage) [10]，要避免此問題可以利用超鏈結圖來分析使用者各種可能的瀏覽路徑。分析時，由進入點開始到達一節點的所有路徑，原先需經過某些多形節點才能到達；但是，增加新瀏覽路徑後，使用者可透過此新路徑而進入此節點。此狀況即可以辨識出有限制性瀏覽路徑遭破壞。

圖九顯示新增及刪除一個檔案的範例：在(a)中左邊表示尚未新增前的超鏈結圖，已知有二個檔案 F1 及 F2，當加入檔案 F3 之後，透過超鏈結圖的表示可以分析 F3 而得知其有二條超鏈結 L1 及 L2，但 L2 是未附著鏈結；F3 是無法瀏覽網頁，因為沒有任何超鏈結連到 F3。

在(b)中，原先的超鏈結圖如左邊所示，當其中的檔案 F4 被刪除後，L1, L2 及 L3 變成未附著鏈結，因為已經指向一個不存在的檔案了。而且原先由 F4 透過 L4 及 L5 鏈結的 F5 及 F6 也成為無法瀏覽網頁。



圖九：新增或刪除一個檔案之結構性影響分析範例

5.2 資料性影響分析

Web 資料庫軟體的資料性異動主要是新增或刪除表格與欄位與表格關係的變更，這些異動的影響分析可以分為架構異動分析與資料異動分析。Web 資料庫軟體模型中的實體關係圖與資料流程圖可以做為這些分析的依據。

1. 架構異動分析 (Schema Change Analysis)

- 資料庫異動之刪除表格：當使用者要刪除表格時，可以讓程式設計師在實體關係圖中來選擇，而刪除後所影響到的檔案可以由資料流程圖中找出所有被影響到的處理元 (檔案)。此動作可以分為二類：一是對於這個表格 (即資料儲存器) 有資料存取的動作，亦即有某個處理元其資料流入此資料儲存器或是從資料儲存器有資料流入某個處理元。二是在某個處理元的輸入或輸出資料流中包含此表格中的欄位。必須將所找到屬於這二類的這些處理元 (檔案) 列出供程

式設計師一一修正。

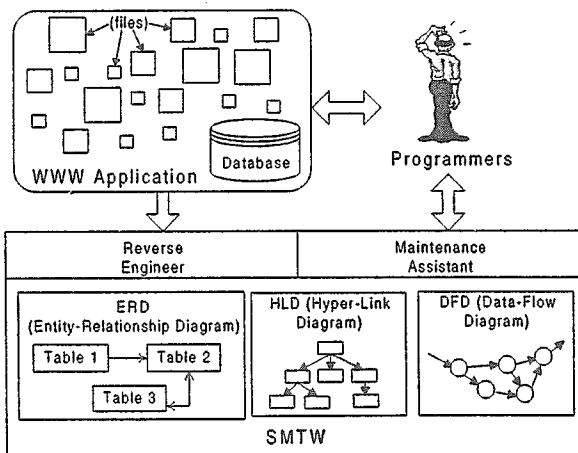
- 資料庫異動之增加或刪除一個欄位：當增加一個欄位時，需從資料流程圖中找出對於此欄位所屬表格有資料存取的動作，亦即有某個處理元有資料流流入此資料儲存器或是從資料儲存器有資料流流入某個處理元，列出這些處理元讓程式設計師決定需要將新增欄位的處理動作放入哪些檔案中。而刪除一個欄位時，只需在資料流程圖中找出有哪些處理元的輸入或輸出資料流中包含此欄位，列出這些處理元供程式設計師來一一修正之。
- 資料庫異動之更改欄位資料型態：更改某個欄位的資料型態時，與刪除某個欄位的維護支援相同，都需在資料流程圖中找出有哪些處理元的輸入或輸出資料流中包含此欄位，列出這些找到的處理元供程式設計師一一修正。

2. 資料異動分析(Data Change Analysis)

- 資料庫異動之增加或刪除關係：當增加一個關係時，牽涉到這個關係二邊的表格，所以，需從資料流程圖中對於這二個表格（即資料儲存器）有資料存取的動作，亦即有某個處理元有資料流流入這二個資料儲存器或是從這二個資料儲存器有資料流流入某個處理元，列出這些處理元供程式設計師來決定是否將這樣關係的處理動作加入。而刪除一個關係時，只需在資料流程圖中找出某個處理元同時對這個關係二邊的表格有資料存取的動作，亦即此處理元同時有資料流流入或流出這二個資料儲存器。列出這些原先有處理這個關係的處理元供程式設計師一一進行修正。

6. Web 資料庫軟體維護系統架構

一個 Web 資料庫軟體維護系統架構如圖十所示（簡稱 SMTW: Software Maintenance Tool for Web database applications）。



圖十：SMTW 的系統示意圖

Web 上的應用軟體 (Web Application) 是由一堆檔案及一個資料庫所組成，這裡用一堆方塊來表示這

些檔案，方塊的大小代表檔案的大小。在圖下面是 SMTW 的系統，主要可分為三個部份：一是圖形表示部份，包括三種圖形（實體關係圖、超鏈結圖及資料流程圖）；第二部份是逆向工程處理 (reverse engineer)，此部份負責分析組成 Web 應用軟體的這些檔案，並產生三種所需的圖形；第三部份是維護助理 (maintenance assistant)，其負責利用這三種圖形來協助程式設計師對於 Web 資料庫軟體進行各種維護的動作。

7. 結論及未來工作

在本篇文章中，我們對於 Web 上的資料庫軟體，提出了一套可以支援維護動作的方法及工具架構。根據維護時會進行的異動，我們提出 Web 資料庫軟體的模型包含超鏈結圖、實體關係圖及資料流程圖來支援這些異動的需求。針對目前 .idc 檔及 .htx 檔的語法，我們提出一套規則來建構這三種圖形，對於其他處理資料庫的方式，只需在這些規則加以擴充即可。這三種表示法不僅僅在維護上提供支援，還可以使程式設計師容易的瞭解應用程式與資料庫的架構。

目前已經開始進行這樣工具的製作，而我們也將透過一個包含三百多個程式檔案的 Web 資料庫軟體來實際地驗證此工具的實用性。對於未來的目標是期望其他的相關工具，例如程式撰寫或是除錯的輔助工具等，使得在 Web 上製作應用軟體可以透過一個整合性的程式設計環境，以便有效地加開發與維護 Web 資料庫軟體。

參考文獻：

- [1] P. Chen, *The Entity-relationship Approach To Logical Database Design*, Wellesley, Mass. QED Information Sciences, 1991.
- [2] D. Flanagan, *JavaScript : The Definitive Guide*, Bonn O'Reilly, 1997.
- [3] I.S. Graham, *The HTML sourcebook: A complete guide to HTML 3.0*, John Wiley & Sons, 1996.
- [4] J. Martin and C. McClure, *Diagramming Techniques for Analysts and Programmers*, Prentice-Hall Inc., 1985.
- [5] "Microsoft FrontPage 97 and SQL Server," Microsoft® Corporation white paper, 1997.
- [6] B. Musteata and R. Lesser, *Standard SQL relational database language guide and reference menu*, Computer Technology Research Corp., 1988.
- [7] S.S. Yau and S.S. Liu, "Some Approaches to Logical Ripple-effect Analysis," Software Engineering Research Center, SERC-TR-24F, University of Florida, Oct. 1988.
- [8] S. Shlaer and S.J. Mellor, "Chapter 2: Information Modeling Concepts," *Object Lifecycles: Modeling the World in States*, Yourdon Press, 1992, pp.11-31.
- [9] G.W. Hansen and J.V. Hansen, *Database management and design*, Prentice-Hall Inc. 1996.
- [10] D.G. Virgil and S. Gupta, "Towards a Theory of Penetration-Resistant Systems and its Applications," *Journal of Computer Security* 1, pp133-158, 1992.