

# 擷取與分析動態網頁以支援網站連結架構維護

## Extracting and Analyzing Dynamic Pages for the Maintenance of Website Hyperlink Structures

陳淑玲  
南台科技大學管理與資訊學系  
slchen@mail.stut.edu.tw

劉建宏  
台北科技大學資訊工程學系  
cliu@ntut.edu.tw

### 摘要

隨著網站的大量增加以及網站內容與服務的頻繁更新，網站的維護逐漸引起重視。然而網站連結缺乏結構，且網頁又可動態產生，造成網站架構極難了解和維護。雖然目前有一些關於網站維護的工具，可以檢查出斷裂的連結或建構網頁間的連結關係。然而這些維護工具大都只考慮靜態的 HTML 網頁，而忽略動態網頁(dynamic pages)對連結架構的影響，以至於無法提供較完整的網站連結架構來支援網站的維護。本論文提出一個考慮動態網頁的網站連結架構建立與呈現的方法，列出一些有助於網站架構維護的網頁連結屬性資料，以及實做一個網站連結擷取的工具，來幫助維護人員了解連結架構，評估更改連結可能造成的衝擊，以降低網站連結維護的時間和成本。

**關鍵詞：**網站維護，連結架構，動態網頁，網站維護工具

### Abstract

As the number of Websites rapidly increases and the contents and services of Websites are frequently updated, the maintenance of Websites becomes critical. However, the unstructured hyperlinks and dynamic pages make Website structures difficult to understand and maintain. Although existing tools can help to detect broken links or to provide link relationships between Web pages, most of the tools only consider the static HTML pages and are unable to provide completed hyperlink structures for supporting Website maintenance. In this paper, we propose an approach to extract and represent the hyperlink structures of Website with the consideration of dynamic pages. Several hyperlink attributes of Web pages for supporting maintenance of Websites are suggested. A prototype tool is constructed to facilitate the understanding of hyperlink structures and the evaluation of possible change-impacts of hyperlinks in order to reduce the time and cost of Website maintenance.

**Keyword:** Website maintenance, Hyperlink structure, Dynamic page, Website maintenance tool

### 一、前言

隨著網際網路(Internet)的盛行，網際網路軟體(web Applications)的數量近幾年呈現快速的成長，並被廣泛地應用於不同的領域，像是公司的入口網站，電子商務，教育或娛樂等。因為需求的關係，許多網站會經常地更新，以提供最新的內容，例如一些 B2C 的電子商務網站可能每星期或每天更新部份的網頁內容，以便顧客能獲得最新的商品訊息或使用最新的服務。

雖然網站的頻繁更新有助於滿足顧客或商業的需求，卻也造成網站維護的沉重負擔。維護人員除了必需經常更新網頁的內容外，亦必需更改相關網頁、圖形和文件等各種的超連結(hyperlinks)，以避免使用者無法瀏覽到正確的內容或連結到適當的服務。然而網站的連結缺乏結構化(unstructured)，且網頁又可以根據使用者的互動，動態地產生，使得連結關係極為複雜，不易了解。再加上維護人員通常不是網站的設計或開發人員，以致於常因不清楚網頁之間的連結關係或網頁、圖形與文件的存放位置，無法正確地維護需要更新的網頁和連結，導致連結的斷裂(broken)或網頁內容的不一致，因而造成顧客的不便，甚至造成公司的商業損失。

雖然目前市面上有許多斷裂連結的檢查軟體，如 W3C 的 link checker 和 open source 的 hyperlink checker 等[19, 6]，可以協助維護人員檢查出斷裂的連結。然而這些檢查軟體大都只考慮靜態的 HTML 網頁，並未偵測包含在動態網頁(dynamic page)內的連結。因此，對於有許多動態網頁的大多數網站而言，這些檢查軟體並無法提供完整的連結資訊以支援其網站的維護。此外，大多數的連結檢查軟體，只偵測連結是否斷裂，並未提供網站的連結架構或網頁的連結屬性資料，例如一個網頁有多少連結指向該網頁，該網頁含有的超連結數量等，無法幫助維護人員了解網頁之間的相依關係(dependence)和評估更改該網頁連結所可能造成的衝擊(impacts)，以便做適當的判斷來減低網站維護的時間與成本。

有鑒於此，本篇論文提出一個考慮動態網頁以建立完整網站連結架構的方法。並說明一個階層

式的連結架構表現方式，以改善架構呈現的延展性 (scalability)，使得網站的連結架構容易了解。同時根據 HTML 的定義，擷取和分析網頁的連結屬性資料，提供維護人員分析網頁的重要性的對改變連結架構可能的影響。並實做一個網站連結擷取工具的雛型，以幫助網站維護工作的進行。

這篇論文的組織如下：第二部份回顧目前網站架構分析與維護的相關文獻。第三部份說明擷取與分析網站連結架構的方法。第四部份說明所提出的網站連結架構呈現方式。第五部份則描述一些網頁連結的屬性與應用。第六部份則說明連結擷取工具雛型(prototype)開發的考量與使用範例。第七部份提出本論文的結語及未來研究的方向。

## 二、相關文獻探討

雖然網站維護是個重要的課題，然而目前大多數的文獻均注重網站的開發技術，探討網站維護相關議題的研究仍不多。雖然有些研究提出分析和擷取網站連結架構的方法，但大多只考慮網頁之間的連結關係，並以測試網際網路軟體為目的，而未考慮圖形和文件檔案等其他的超連結，以及提供網頁連結屬性資料。以下就網站維護及分析網頁連結關係的相關參考文獻做一簡要的回顧：

Ricca 和 Tonella[16, 17, 18]定義了一個網際網路軟體的結構模型，描述 HTML 網頁、動態網頁、表單(form)和框架(frame)之間的關係，並且開發了一個工具 ReWeb，可以擷取和建構網頁之間的連結關係圖，以供辨別在經過一段時間後網站的差異與改變(evolution)，如網頁和連結的增刪與修改等。ReWeb 同時也可以用來分析和驗證使用者輸入的資料在網頁間的流向是否正確。然而 ReWeb 並非完全針對網站維護而建置，它無法提供關於連結斷裂、數量、分類和網頁連結屬性等資料。同時 ReWeb 主要著重於表單(form)所產生的動態網頁，對於直接附加資料在連結 URL 後面而產生的動態網頁，並未仔細分辨。此外，ReWeb 亦無法分析由 Javascript 所建立的超連結以及其所連結的網頁。

Lucca 等人[10, 11]提出一個工具 WARE，以擷取網際網路軟體的結構(architecture)。WARE 主要是分析網際網路軟體的組成物件，包含 HTML 網頁、伺服器端網頁(server page)、以及 Java 元件(components)等，並用類似 UML[1]物件圖(class diagram)的方式，呈現這些物件之間的關係，以幫助分析人員對於網際網路軟體結構的瞭解。然而 WARE 並沒有提供太多關於連結架構的資訊，例如連結斷裂、數量、以及網頁連結屬性等資料。且 WARE 也未考慮動態網頁內的連結，及其對網站連結架構的影響，因此無法有效的支援網站連結架構的維護。

此外，Lucca 等人[12]提出一個網際網路軟體維護性(maintainability)的評估模型，幫助管理者了解維護其網站的難易程度和評估可能的維護成本。基本上他們參照 Oman 和 Hagemester[15]所提之傳統軟體維護性的模型(maintainability model)，如程式大小(LOC)、控制、和資料耦合度(control and data coupling)等，並依據網際網路軟體獨有的特性和一些度量(metrics)，如顧客端網頁(client page)、伺服器端網頁、和連結等，建立網際網路軟體維護性的模型。他們並進行一些案例的分析，以驗證該維護性模型的效果。

Chung 和 Lee[2]建議應用 Unified Process[7]和 UML 的視覺模型(visual models)到網站的維護上面，並透過反向工程(reverse-engineering)的方法擷取網站瀏覽的概要(navigation scheme)和網站存放檔案的目錄結構。其中網站的瀏覽概要主要是說明網站的瀏覽結構以及元件間(elements)的相依關係，並以 UML 的元件圖(components diagram)來描繪。而網頁和元件檔案的實際存放位置，則以類似階層目錄結構(directory structure)的方式呈現，以幫助網站的管理和維護人員瞭解網站的瀏覽結構、元件的位置和元件間的關係。

Martin 等人[14]開發一個反向工程的工具 Rigi，可以解析 HTML 網頁和呈現一個網站的連結關係圖。Rigi 除了可以擷取 HTML 網頁外，它也能擷取和分析網站伺服器的操作紀錄檔案(log)，以獲得網站實際運行時的一些統計資料，提供給網站管理員。此外，Rigi 提供了圖形編輯的功能，可以讓網站管理員對所擷取得到的網站連結關係圖，進行編輯與整理，並可過濾一些感興趣的特定物件。然而 Rigi 並未支援框架、Javascript、和動態網頁，所提供的網頁連結屬性資料也有限。

Hsieh 和 Yeh[5]針對使用框架、多重視窗及動態網頁技術的網際網路軟體，提出一個超連結分析的方法和工具，擷取網站結構的一些重要屬性，以診斷網站一些常見的結構問題，如斷裂的超連結和過長路徑等，幫助網站管理者瞭解網站的結構，以利網站的擴展和減低網站的維護成本。

Linos 等人[9]開發一個網站維護工具 Perlbot，並用該工具進行案例分析。Perlbot 基本上是一種 spider，可以自動擷取和分析網頁，不過它只分析靜態的 HTML 網頁，忽略動態網頁的連結。且連結屬性資料的提供也相當有限，只考慮網頁、斷裂連結、限制連結、Javascript 和 Cookies 的數量等資料，因此對於網站維護的支援有其侷限。

Kienle 等人[8]報告其利用 Macromedia Dreamweaver[13]開發和維護一個學術網站的經驗。Kienle 等人指出 Dreamweaver 缺乏支援網站維護工作的功能，以及 Dreamweaver 檢查斷裂連結的 link checker 功能在開發者模式(developer view)和顧客使用模式(client view)不一致的情況下，可能

發生錯誤的問題。為了提供更多的維護資訊，他們利用 Rigi[14]來描繪 HTML 網頁、文件與圖形等檔案間的關係和顯示網頁內部與外部的斷裂連結，以利維護工作的進行。

Hassan 和 Holt[3, 4]提出一個靜態分析的方法來擷取網際網路軟體的結構。基本上他們使用不同的解析器(parsers)來分析用不同技術開發的網頁和元件，例如 HTML 網頁、伺服器端網頁、資料庫存取程式、分散式元件和二進位程式碼(binary code)等，並描繪它們之間的連結、控制和資料的相依關係。此外，為了減低網際網路軟體結構圖呈現(render)的複雜度，Hassan 和 Holt 依照這些網頁和元件負責的功能，將其區分(clustering)成三個階層(layers): presentation logic、business logic & database 和 infrastructure，並提供階層化的圖形表示方法。雖然此法可以描繪網頁和元件間詳細的結構，然而靜態分析的方法並不能產生動態網頁，無法分析動態網頁的內容，因此無法提供較完整的網站連結架構。

### 三、網站連結架構的擷取與分析

關於網頁連結的分析，目前文獻中主要的分析方法可以區分成兩種：一種是靜態分析 HTML 網頁和伺服器端網頁(sever page)的程式碼，並建構網頁彼此間的連結關係；另一種則是透過 web spider(即 crawler 或 robot)，自動抓取並分析、建構 HTML 網頁間的連結關係。第一種方法因為讀取伺服器端網頁的程式碼，因此可以分析嵌入於(embedded)伺服器端網頁內的動態網頁內容，所以能建構網站連結架構。然而網頁(HTML 或伺服器端網頁)可能存放在不同的機器與位置，或由不同伺服器端網頁技術所撰寫開發而成，如 ASP、JSP、PHP、或 Java Servlet。這使得分析與建立網站連結架構變得複雜，工具也不易建置。此外，此方法亦不能判斷一個連結的狀態是否為良好或斷裂。

第二種方法則為目前大多數超連結檢查軟體所採用，網站維護人員只需輸入首頁的網址，web spider 便可自動下載和分析從首頁可以連結到的 HTML 網頁，直到所有被連結的網頁均下載完畢或使用者停止搜尋為止，維護人員不需知道網頁存放的機器或位置，使用較方便。但是此種方法因為只分析 HTML 網頁，忽略需要使用者互動或輸入資料才能產生的動態網頁，因而無法提供較完整的網站連結架構。

為了能擷取完整的網站連結架構及提供網頁連結屬性資料，在這篇論文中，我們考量動態網頁所包含的超連結，並採取類似 web spider 的方法，透過 breadth-first 的搜尋方式來擷取與分析網頁，以便維護不同的網站，而不需考慮網站是使用何種伺服器端網頁開發技術。為擷取動態網頁所包含的

超連結資料，基本上我們採取動態分析的方法，亦即當所分析的超連結指向一個伺服器端網頁時，我們產生適當的 request 需求，觸發該伺服器端網頁的執行，並擷取和分析所產生的動態網頁內容，以繼續下載和分析該動態網頁所連結到的網頁，進而建立完整的網站連結架構。

必需說明的是，送至伺服器端網頁的 request 通常包含使用者輸入的資料或選項。目前我們採用最簡單的方式來獲得這些資料，即透過對網站瀏覽路徑以及相關文件的了解，由維護人員提供這些資料。這些輸入的資料可以透過網路封包擷取並紀錄下來，下次再維護該網站時可以重複使用。另外，依據不同的 request 資料，一個伺服器端網頁可以產生不同的動態網頁，這些不同的動態網頁會有不同的內容，並可能連結到不同的網頁，例如：一個具備登入驗證功能的伺服器端網頁，通常會依據其所收到的 request 資料，產生登入成功或失敗兩種動態網頁，而這兩種動態網頁又可能分別連結到不同的網頁。因此，網站維護人員可能需要依據不同的瀏覽路徑，對同一伺服器端網頁送出不同的 request，以便產生不同的動態網頁，建立較正確和完整的網站連結架構。不過，所產生的網站連結架構也會隨著維護人員所輸入的資料和順序的不同，而會有些許的差異。

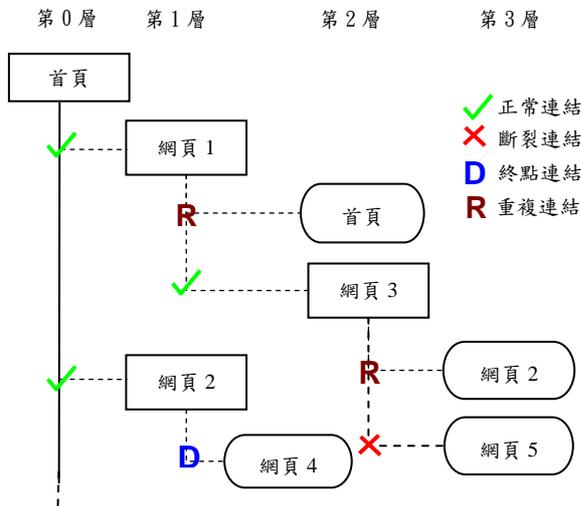
### 四、網站連結架構的呈現

在網站的連結架構建立後，一個重要的課題是如何將所建立的連結架構呈現出來，也就是網站連結架構的 visualization。基本上網站的連結架構可以區分成三種：樹狀(tree)、網路(network)和循序(sequence)結構。其中樹狀和循序結構較為單純，網頁間的連結不會造成迴圈(loop)，也就是一個網頁不會直接或間接經由其他網頁連結回到自己本身。網路結構則複雜許多，超連結可指向任何網頁，沒有任何限制，因此網頁間的連結可能會存在迴圈的情形。所以在擷取網頁時必須注意，以避免陷入無限迴圈重複擷取網頁。

目前一些網站維護工具，如 ReWeb 和 Rigi 等，是直接依照網頁連結的結構，以圖形的方式來表現。在這些連結架構圖中，節點代表一個網頁，節點間的線則代表網頁間的連結。此種方法簡單且直接，有助於維護人員瞭解網站的連結架構，但是在網頁數量過多時，缺乏階層的表現方式會讓網頁節點擠在一起，特別是網站的連結如果是採用網路結構，相互交錯的連結將使網站連結的架構不易瞭解與維護。

為了讓網站架構容易瞭解和考慮連結架構呈現的延展性，我們採取一個樹狀結構的表現方式，將網頁連結予以階層化，稱為網頁連結樹(Web Hyperlink Tree, WHT)。如圖一所示，WHT 的 root

為網站的首頁或某一特定網頁，矩形和橢圓形的節點均代表網頁，節點間的線則代表連結。不同的是，矩形的節點代表 internal node，即該網頁含連結需要分析，並可繼續擴展(extend)至其子節點(child node)。橢圓形的節點則為 leaf node，表示該網頁沒有包含需要分析的連結，無須再繼續擴展。



圖一 WHT 的範例

另外，網站的階層由上而下，網站首頁為第 0 層，首頁能連結到的網頁則位於第 1 層，第 1 層網頁能連到的網頁連結則為位於第 2 層，以此類推。例如圖一的網頁 1 位於第 1 層，其指向的網頁：首頁和網頁 3，皆位於第 2 層，其中網頁 3 可再繼續擴展，而首頁則無法繼續擴展。必須說明的是，在 WHT 中，一個網頁可以同時存在不同的階層。

由於網站的連結架構大部分為網路結構，可能包含迴圈，導致 WHT 的節點會無限制地成長。為了避免這個問題，在建構 WHT 的過程中，每新加入一個節點到 WHT 時必需檢查該網頁是否已出現在 WHT 上。若(1)該網頁尚未出現在 WHT 上；(2)該網頁在網站的搜尋範圍之內；和(3)該網頁含有需要分析的連結，則加入一個矩形節點，表示該網頁可以繼續擴展。反之，則加入一個的橢圓形節點，代表該網頁超出搜尋範圍、不包含任何需要分析的連結或已出現在 WHT 上，並可經由其對應的矩形節點得知其連結關係。例如在圖一中，首頁可以連結至網頁 1，網頁 1 也可以連結至首頁。其中網頁 1 的首頁子節點為橢圓形，表示該網頁已出現在 WHT 上，其連結關係可以經由矩形的首頁節點得知，如此將可避免 WHT 的節點無限制地成長。

此外，在 WHT 中每個連結的前面有個記號，顯示該連結的狀態是良好或斷裂。良好的連結又分成三種：正常、重複或終點連結。其中重複連結是指該連結所指向的網頁已出現在 WHT 上。例如圖一中指向第 2 層的首頁和第 3 層的網頁 2 之連結即

為重複連結。而終點連結則代表其所指向的網頁沒有包含任何指向其他網頁的連結，即該網頁為一瀏覽路徑中的最後一個網頁，使用者必須透過瀏覽器工具列的「上一頁」按鍵才可連結至其他網頁，例如圖一中網頁 2 和網頁 4 間的連結即為終點連結。

## 五、網頁連結屬性的擷取與應用

網頁連結有許多不同的屬性，這些屬性不但有助於維護人員了解網站的連結架構或網頁的重要性，亦可作為網站維護的參考。一些常見的網頁連結屬性，如 inlink(即指向該網頁的連結)和 outlink(即該網頁指向其他網頁的連結)等，可以幫助維護人員判斷更動一個網頁連結所可能產生的衝擊(change impact)，進而選擇一個較佳的更動方式以降低維護時間或成本。例如當移動一個網頁到新的 URL 時，如果該網頁有許多 inlinks，維護人員可以選擇在該網頁原來的 URL 放置一個新網頁，並讓這個新網頁將所有的 requests 重導(redirect)至原來的網頁，這樣可以減少更改所有 inlink 連結的時間，並減少可能產生的連結錯誤。

表一列出一些網頁連結屬性，可以提供維護人員參考。例如：從首頁連結至一特定網頁可能有許多瀏覽路徑，這些瀏覽路徑的數量與長度，可能會影響瀏覽該特定網頁的便利性，因而影響使用者的瀏覽行為，所以這些資料的擷取將有助於網站連結架構的維護。此外，從首頁到一特定網頁的不同瀏覽路徑可能會有交集，這些共同的路徑稱為 dominance[17]，亦即使用者必需經過這些路徑才可連結到該特定網頁，這些資料對於連結架構的維護亦相當有價值。

表一 網頁連結屬性

網頁連結屬性	說明
Server page link	產生該網頁的伺服器端網頁連結(適用於動態網頁)
Number of inlinks	其他網頁至該網頁的連結數量
Number of outlinks	該網頁至其他網頁的連結數量
Basis navigation paths and lengths	從首頁至該網頁所有的基本瀏覽路徑與路徑長度
Shortest path	從首頁至該網頁的最短瀏覽路徑
Destination of the navigation path	該網頁是否為瀏覽路徑的終點連結
Dominance paths	從首頁至該網頁所有基本瀏覽路徑的共同路徑
Number of broken links	該網頁所包含的斷裂連結數量
Number of image/file/script links	該網頁所包含的圖片、檔案或 script 等連結數量

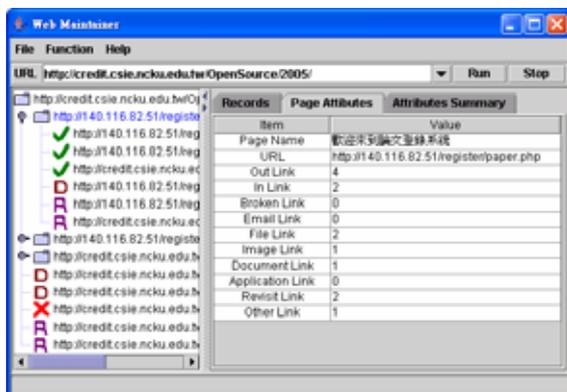
## 六、工具建置的考量與使用範例

為了驗證所提出的方法，建立較完整的網頁連結架構，我們實做一個工具雛型來擷取和分析動態網頁。由於一個網站內通常有一些網頁會連結到網站外部的網頁，為了避免該工具下載其他網站的網頁，在建立網站連結架構時，維護人員需要能夠設定網頁搜尋的範圍，例如只搜尋特定的 domain 或目錄下的網頁，任何不在該 domain 或目錄下的網頁均不予考慮，以避免不必要的搜尋與分析。

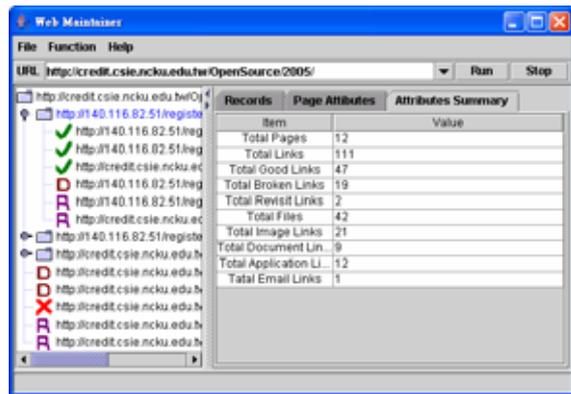
此外，若維護人員只對部份網頁的連結關係感到興趣或只需重構部份的網站連結架構，維護人員可以設定停止搜尋的條件，不需每次都搜尋整個網站，這對於一個有數千個網頁的網站，可以省下不少的搜尋時間。所實作的工具其搜尋停止條件有(1)立即停止；(2)搜尋的時間長度；(3)網頁連結階層的深度；和(4)網頁(或連結)的數目等，這些停止條件亦可混合使用。

同時，由於網頁擷取的速度會受到網頁數量、網路頻寬和網站伺服器負載的影響，為了提升工具的效能，所實做工具可限制其等待網站伺服器的回應時間(response time)，以避免因為網路擁擠或網站伺服器速度太慢，造成等待時間過長，降低工具的效能。

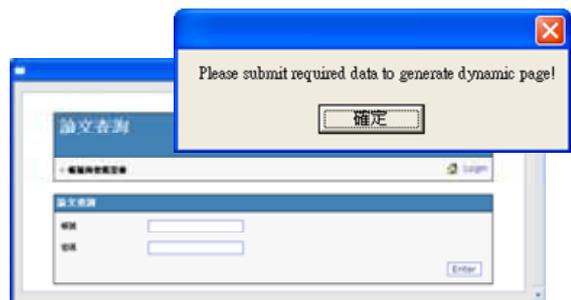
圖二至圖五為目前連結擷取工具雛型的一些使用範例。圖二為一個研討會網站其連結架構擷取和分析的結果，圖二的左邊為該網站的 WHT，顯示該網站有一斷裂連結。此外，維護人員可以點選 WHT 上的任何一個網頁，圖二的右邊便會顯示所選到網頁的一些連結屬性，不過表一中所列的屬性，工具雛型目前尚未能全部提供。圖三則為整個網站的連結統計資料。另外，圖四為工具提示維護人員輸入資料以產生動態網頁的範例。工具雛型將不斷地詢問維護人員輸入不同的瀏覽路徑資料，以便擷取由同一伺服器端網頁所產生之不同類別的動態網頁來進行分析。



圖二 網站連結架構和網頁連結資料的範例



圖三 整個網站連結資料的範例



圖四 輸入資料以擷取動態網頁的範例

## 七、結語與未來研究方向

在這篇論文中我們提出一個考慮動態網頁的網站連結架構建立與呈現的方法以支援網站連結的維護。在這個方法中，我們採取類似 web spider 的方式來擷取和分析動態網頁所包含的連結，並利用階層式的方式來呈現所產生的網站連結架構，讓連結架構容易瞭解和延展。同時，我們描述了一些連結屬性資料，有助於維護人員了解網站的連結架構和網頁的重要性。並實做一個擷取和分析網站連結的工具雛型，以協助網站維護工作的進行。

這篇論文所提出的方法仍在起始的階段，仍有許多需要進一步研究和改善的地方。例如動態網頁的擷取仍需要維護人員手動輸入資料，工具雛型亦尚未能完全提供所列出的網頁連結屬性。我們計劃未來可以自動或半自動化地擷取動態網頁以降低維護人員的負擔。此外，目前的工具雛型只能分析和擷取網頁中 HTML 定義的連結，未來我們計劃能支援由 Java script 動態所產生的連結。同時，我們亦計劃探討和提供其他有助於網站維護的連結屬性資料或相關資訊(artifacts)，並研究如何自動計算更改連結所可能造成的衝擊，以協助網站維護工作的進行。

## 八、参考文献

- [1] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*, Addison Wesley, 2000.
- [2] S. Chung and Y.-S. Lee, "Reverse software engineering with UML for Web site maintenance," In *Proceedings of the First International Conference on Web Information Systems Engineering*, 2000, pp. 157-161.
- [3] A. E. Hassan and R. C. Holt, "Towards a Better Understanding of Web Applications," In *Proceedings of the 3rd International Workshop on Web Site Evolution*, 2001, pp. 112-116.
- [4] A. E. Hassan and R. C. Holt, "Architecture Recovery of Web Applications," In *Proceedings of the 24th International Conference on Software Engineering*, 2002, pp. 349-359.
- [5] A. Hsieh and D. Yeh, "Hyperlinks Analysis of Dynamic Web Applications," *Lecture Notes in Computer Science*, Vol. 2722, 2003, pp. 349-352.
- [6] ht:// Check. <http://htcheck.sourceforge.net/>
- [7] I. Jacobson, G. Booch, and J. Rumbaugh, *The Unified Software Development Process*, Addison Wesley, 2001.
- [8] H. M. Kienle, A. Weber, J. Martin, and H. A. Muller, "Development and Maintenance of a Web Site for a Bachelor Program," In *Proceeding of the Fifth IEEE International Workshop on Web Site Evolution*, 2003, pp. 20-29.
- [9] P. K. Linos, E. T. Ososanya, and H. Natarajan, "Maintenance Support for Web Sites: A Case Study," In *Proceedings of the 3rd International Workshop on Web Site Evolution*, 2001, pp. 70-77.
- [10] G. A. D. Lucca, G. Casazza, M. D. Penta, and G. Antonioli, "An Approach for Reverse Engineering of Web-Based Applications," In *Proceedings of the Eighth Working Conference on Reverse Engineering (WCRE'01)*, 2001, pp. 231-240.
- [11] G. A. D. Lucca, A. R. Fasolino, and P. Tramontana, "Reverse Engineering Web Applications: the WARE Approach," *Journal of Software Maintenance and Evolution: Research and Practice*, Vol. 16, No. 1-2, pp. 71-101.
- [12] G. A. D. Lucca, A. R. Fasolino, P. Tramontana, and C. A. Visaggio, "Towards the Definition of Maintainability Model for Web Application," In *Proceedings of the Eighth European Conference on Software Maintenance and Reengineering*, 2004, pp. 279-287.
- [13] Macromedia Dreamweaver. <http://www.macromedia.com/>
- [14] J. Martin and L. Martin, "Web Site Maintenance with Software Engineering Tools," In *Proceedings of the 3rd International Workshop on Web Site Evolution (WSE'01)*, 2001, pp. 126-131.
- [15] P. Oman and J. Hagemester, "Metrics for Assessing a Software System's Maintainability," In *Proceedings of the IEEE International Conference on Software Maintenance*, 1992, pp. 337-344.
- [16] F. Ricca and P. Tonella, "Web site analysis: structure and evolution," In *Proceedings of the IEEE International Conference on Software Maintenance*, 2000, pp. 76-86.
- [17] F. Ricca and P. Tonella, "Understanding and restructuring Web sites with ReWeb," *IEEE Multimedia*, Vol. 8, No. 2, April-June 2001, pp. 40-51.
- [18] P. Tonella and F. Ricca, "Dynamic Model Extraction and Statistical Analysis of Web Applications," In *Proceedings of the Fourth International Workshop on Web Site Evolution (WSE'02)*, 2002, pp. 43-52.
- [19] W3C Link Checker. <http://validator.w3.org/checklink>