行政院國家科學委員會專題研究計畫　成果報告

以 XML 為基礎的網站系統(I)

計畫類別：個別型計畫
計畫編號：NSC91-2213-E-009-116-
執行期間：91 年 08 月 01 日至 92 年 07 月 31 日
執行單位：國立交通大學資訊科學學系

計畫主持人：李嘉晃

計畫參與人員：陳柏愷、陳日昶、莊盟錫、藍彥琨、王明德

報告類型：精簡報告

處理方式：本計畫可公開查詢

中　華　民　國 92 年 12 月 23 日

行政院國家科學委員會補助專題研究計畫成果報告

以 XML 為基礎的網站系統(I)

計畫類別:個別型計畫
計畫編號:NSC91-2213-E009-116
執行期間:91 年 08 月 01 日至 92 年 07 月 31 日

計畫主持人:李嘉晃

本成果報告包括以下應繳交之附件：

□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

執行單位:國立交通大學資訊科學所

中 華 民 國　　年　　月　　日

行政院國家科學委員會專題研究計畫成果報告

以 XML 為基礎的網站系統(I)

XML-based Web Site System

計畫編號: NSC91-2213-E009-116

執行期限:91 年 8 月 1 日至 92 年 7 月 31 日

主持人:李嘉晃

執行單位:國立交通大學資科所

一.中文摘要

目前的網站，皆以 html 檔案為主要內容，搭配以 CGI、PHP、JSP、ASP 等方式來動態產生網頁。然而，以 html 為主要輸出內容的網站已不能符合越來越多變化的網路環境；如目前上網的裝置越來越多樣化，包括 PC、PDA、手機……等等各種不同的螢幕大小、不同的呈現方式(只能顯示文字或可以顯示圖形)，不同的協定(html、WML 或其他)，一再給網頁建構者(web designer)帶來考驗；另則由於 html 不具有結構性，帶來的困擾包括資料的可重用性低、不易交給程式做再處理、資料搜尋僅限於關鍵字(key word)比對、遠端系統間的資料溝通不易等等……，都使得網站的能力大為受限，並且要耗費相當多的人力來處理資料。

因此，吾人提出以 XML 為基礎的新一代網站架構，由於 XML 的樹狀結構，以及資料內容(content)與呈現方式(presentation)分離的特性，使得網站的功能性能夠再提昇。包括更方便的網頁建構、資料再利用、更精準的資料搜尋、更有彈性的存取管制、遠端系統間的資料溝通等等。

此系統計劃整合現有的相關技術，包括使用 Linux 為 OS、搭配 Apache HTTP server、Apache Tomcat servlet engine、Cocoon XML publishing framework 等自由軟體，並撰寫整體系統所欠缺的軟體元件包括 XML 存取權限管理、遠端溝通等部分，以建構出一個完整的新一代網站架構。

關鍵詞:XML、網站系統、權限控管、資訊交流

Abstract:

Almost every web site is constructed with HTML documents cooperating with CGI, PHP, JSP, ASP and so on for dynamic page generation for this moment. However, it is not adequate nowadays for such various network environments as the various network devices, including PC, PDA, mobile phone with different monitor sizes, different presentation styles (some can show pictures and the other can't), different protocols (such as HTML、WML and etc.). These complicate environments bring tremendous challenges for web designers. Furthermore HTML documents are not well structured, hence they are not easy to reuse, not easy for further processing. At best, they can only perform search based on key word comparisons, and they are not easy for such tasks as remote data exchange. It limits the ability of web site and cost a lot of manpower.

In this project, we propose a new web site structure which is based on XML. Combining with tree structure, and separation of content and presentation by XML, the web site will be equipped with more functions. These functions include more convenient page construction, data reuse, more accurate data search, more flexible access controlling management, remote data exchange and so on.

This project will integrate all the related technologies, including Linux for OS, cooperating with Apache HTTP server, Apache Tomcat engine, Cocoon XML publishing framework and other free software. We will study and develop other components including XML search engine, XML access controlling management, remote data exchanging, and translation tools in order to build a complete new generation web site system.

Keywords: XML, Web site system, access control, information exchange

二.緣由與目的

儘管 XML 在網站上的應用極為優異，然而目前網路上實際架設的 XML 網站仍然寥寥無幾，追究其原因，除了推廣不足，還是相關建構工具仍不充足。目前具備 XSL[1]轉換能力的網站系統已經相當成熟，如 Apache 的 Cocoon[2]計畫。但是如資訊自動交流系統、XSL 撰寫工具等相關工具卻不盡齊全，使得建構 XML 網站需要專業的程式人員投注相當大的心血去開發必要的應用程式，才能達成以上所提 XML 網站的好處。一般人更不易跨過這個門檻，XML 網站無法普及，XML 搜尋引擎的理想也無法實現。

目前來說，XML 網站的相關工具以 XSL 撰寫工具及資訊自動交流系統最為缺乏，而本計畫將設計出一套結合現有 XML 技術的資料自動交流系統，並具權限控管功能，讓 XML 網站架設者，能夠輕易達到資料收集與分享的功能，並能將收集來的資料直接轉換成許多不同的網頁，或者以 XML 的形式再分享出去。本系統的重點在於讓網站架設者不需自行撰寫程式，而能輕易的收集及分享資料，並與現有 XML 網站技術相結合，轉換成適合觀賞的網頁或適合給程式處理的 XML 內容。

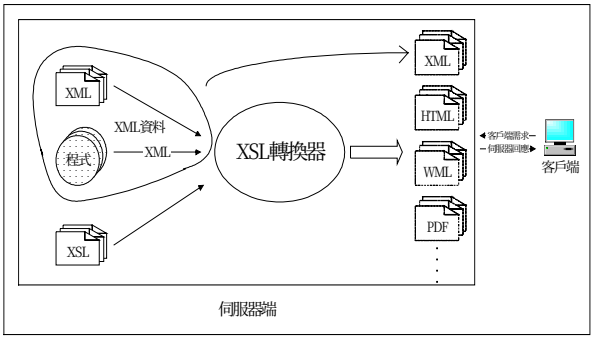
我們可以想見，在各行各業不同的領域當中，以網站的方式來做資料的分享會非常普遍。舉例來說，當一個客戶向供應商要求報價時，供應商可立刻由各下游廠收集各種零件的報價，即時提供給客戶最新的價格。再譬如像便利商店這樣的系統，各分店可架設自己的網站並將銷售情形上網，母公司可隨時向各分店收集資料，掌握即時的產品銷售情形。而在學術上的用途，譬如有許多機構積極研究 DNA，即適合以本系統做資料的交流及分享，交流來的資料既適合做分析應用，也便於轉換成網頁對外發表。再像是行政處理方面，譬如一個學生要從學校畢業，得向教務處確認學分數足夠、向圖書館確認沒有未還圖書、向總務處確認沒有缺繳款項……等等，經由人力一一到各單位確認蓋章，然而以上的資料各單位皆有將之上網，就因為 HTML 格式的網頁不適合給程式做資訊自動收集與處理，因此得花費無謂的人力到處去蓋章。這類的問題都將因為本計畫的研究，獲得解決。

三.系統設計

當 XSL 定案，相關 XML 技術一一成型之後，開始有網站系統在伺服器端結合 XSL 轉換技術，其中以 Apache Cocoon 計畫最具代表性，此系統明確的把網頁內容及呈現方式切開，而網頁內容又可由程式邏輯動態產生，因此程式邏輯(logic)、網頁內容(content)、呈現方式(style)可由不同專才的人來分別撰寫，各司其職，更能符合實際的應用環境。舉例來說，一個新聞網站，內容的部分是由專業記者負責撰寫，網頁外觀由美工人員負責設計編排，而程式設計師可負責撰寫新聞搜尋等程式邏輯，不同的部分由不同的專才負責，並且三者之中任一有更動，不需要其他兩者的配合修改就能立刻反映到最後輸出的網頁，如圖一。

從圖中可以看出，伺服器端不但可以提供給人觀看的 HTML、PDF 等格式的資料，也可直接輸出 XML 格式的資料。由於 XML 不含外觀呈現的資訊，並且具有良好的語意及結構性，

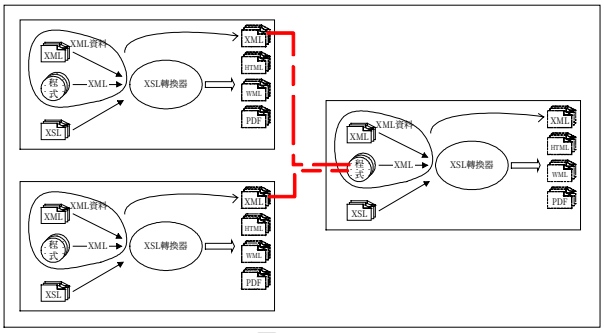
因此很適合作資料的再利用，也能應用到比關鍵字比對更深入的搜尋機制。



圖一

我們可以看到，這樣的網站架構下，已經比傳統利用程式存取資料庫簡單很多。若是網站建構者，欲由相同內容建構多個版本的網頁，則只需要建構多份 XSL，對應到同一份 XML 內容即可；若是要由不同的內容，建構相同版面編排的網頁，則是建立多份符合相同 DTD 的 XML 文件，對應到同一份的 XSL 版面編排。若是要由傳統網站架構來達到這樣的目的，雖然只是存取資料與格式轉換，仍要撰寫對應的程式。而在 XSL 網站架構下，除非要把資料再作邏輯運算，否則存取資料跟格式轉換是不需要網站建構者自行處理的。

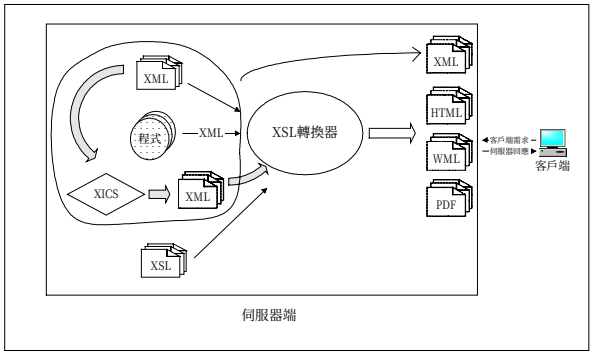
當兩個網站間要交換資料，可撰寫一個程式，透過 HTTP 協定向遠端的網站取得 XML 格式的資料，如圖二所示。



圖二

我們可以看到，這樣的架構仍有兩個比較不方便的地方，一是收集遠端網站的資料必須由網站建構者撰寫程式，第二個問題是，遠端網站的提供的 XML 資料是不具個別化的；因為單一的 URL 要求，會回應單一的 XML 資料，不因不同的客戶端而回應不同內容，也就侷限了實際應用的場合。雖然這兩個問題都可以由網站建構者自行撰寫程式來解決，然而若是能透過簡單的設定，就能達成資料自動交流的目標，將能夠把 XML 網站的應用再提升一步。

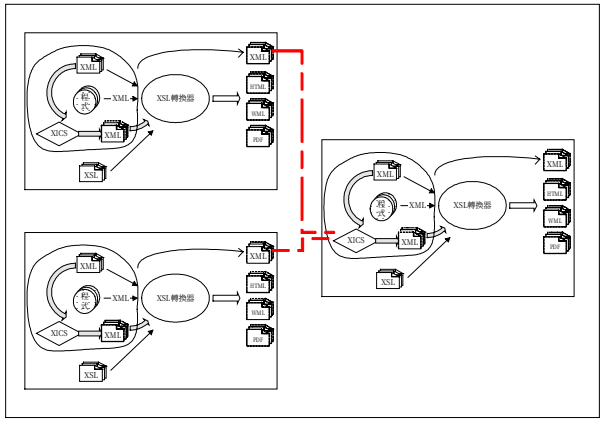
由於 XSL 網站模型仍有其不足之處；因此，我們以 XSL 網站為基礎，改良成為 XICS(XML-based web site Information Communicating System)網站架構。我們的目標即是讓網站建構者，只要透過簡單的設定，就能夠與遠端的 XML 網站系統進行資料交流，並且也能根據不同的客戶端權限，給予個別化的資料。我們的架構如圖三。



圖三

從圖中可以看到，除了儲存在硬碟的 XML 檔案，以及由程式產生的 XML 文件之外，我們多了一個 XICS 元件，這個元件的作用是，讓網站建構者可以透過簡單的設定，即可擷取站內或遠端網站的資料，並產生多份不同權限的 XML 內容，因此當不同權限的客戶端向本站發出需求時，XICS 元件會負責擷取所需要的資料，並產生符合客戶端的 XML 內容，再交給 XSL 轉換器處理或者直接輸出。

而 XICS 元件除了可擷取站內的 XML 資料，也可透過 HTTP/HTTPS 協定抓取遠端的資料，因此網站間的資料交流變的格外簡單。如圖四。



圖四

同樣的，擷取遠端伺服器的資料也是由網站建構者於 XICS 元件作設定，而擷取來的資料，是遠端伺服器根據本機的權限所給予的。如此一來，網站建構者不需要自行撰寫程式，即可作到極佳的資訊自動交流，並具權限控管。

我們比較傳統資料庫式網站系統、XSL 網站系統 以及 XICS 網站系統，從資料取得(content) 、外觀呈現(style) 、程式邏輯(logic)、資料分享(sharing)四個面向分析探討。

傳統網站系統模型：

1. 以資料庫的方式獲得資料，由於資料庫系統經過多年的演進，已經具有相當好的速度及穩定性，因此資料的取得相當有效率。
2. 即使只是單純的擷取資料作成網頁，不含邏輯運算，也須由程式設計員撰寫，因此程式設計師的工作量格外沉重。
3. 若要從相同的內容，產生不同外觀呈現的網頁，則要撰寫許多相似的程式，這些程式碼在資料擷取的部分相同，而外觀呈現的部分不同。並且外觀呈現的工作難以切割出來，由美工專職人員負責，往往得由程式人員跟美工人員協同工作。
4. 若是本站的資料，除了作成網頁之外還想讓他站擷取利用，則必須對外開放資料庫。若是開放給特定的對象則是可行的，但資料庫並不適合像網頁一樣開放給所有人存取；如此一來，資料分享的對象僅能限於少數站台。

XSL 網站系統模型：

1. 資料的來源為站內一個一個的 XML 檔案，在存取的效率上會不如資料庫來的有效率。然而目前已經有許多 XML 資料庫在開發中，可以預見不久的將來這個問題將會獲得改善。
2. XSL 的撰寫仍然稍具有難度。目前尚未出現可見即可得的 XSL 編輯器，因此都由人工撰寫。然而如同在 HTML 網站開始流行之後，Front page、Dreamwave 等網頁編寫工具相繼出現，我們可以預期在短期內即可有方便的 XSL 編輯器。
3. 明確的把資料本身(content)及外觀呈現(style)分成兩個檔案，因此可以交由兩類不同專長的人分開負責；在應用上也更有彈性，可把多份文件(符合同一份 DTD)對應同一份 XSL，產生外觀編排相同，資料不同的網頁，舉例來說，每個實驗室的人事資料皆不相同，然而希望作成外觀編排相同的網頁，可使用這種方式。也可把同一份資料對應多份 XSL，可把相同的內容作成 HTML 版、WML 版、PDF 版等等，並且資料的部分一修改，所有的版本都會立刻反映出來，不會有資料不同步的情形。

4. 程式邏輯的部分，同樣可撰寫程式，輸出為 XML 文件的格式，配合 XSL 運作，搭配情況良好。
5. 除了可輸出為適合人觀看的網頁之外，可把 XML 文件直接輸出，因此客戶端即可獲得不含呈現語法的純資料，很容易作資訊處理及再利用；未來的 XML 搜尋引擎、自動代理機器人也都將以此為基礎。
6. 若兩個網站之間要作資料交流，得由網站建構者自行撰寫程式作資料蒐集的工作，還有網站提供的資料沒有依據需求來源的網站作個別化，這兩個方面都得由網站建構者自行處理，比較不方便。

XICS 網站系統模型：

XICS網站系統由XSL網站系統改進而來，因此繼承了XSL網站系統的優點，並特別針對資訊分享的部分作了以下改良。

1. 擷取遠端網站資料：可以讓網站建構者，透過簡單的設定即可擷取遠端網站資料，並可立即配合本地端的 XSL 檔案，轉換成網頁輸出。因此對於向本網站發出需求的客戶端來說，他並不需要關心本站的資料來源來自何方，只要本站能夠收集所有必要的資訊，一併呈現給他就好了，並且遠端的資訊一改變，本站也會立刻反映出來，沒有同步性的問題；這在電子商務、行政流程等應用會非常方便。
2. 網站提供的資料具個別化：可以讓網站建構者，透過簡單的設定來作個別化，當擁有不同權限的客戶端來存取時，可以獲得不同的資料。如此一來，網站可以服務不同權限的客戶，不再是一視同仁的提供相同資料，更能符合實際應用的情況。
3. 支援 HTTPS[3]傳輸協定來保護資料的安全性，確保資訊傳輸的過程不被他人窺視，也認證交流的雙方並非他人所冒充。確保了資料的私密性及可信任性，資訊的交流才更有保障。

四.語法定義：

我們定義了一套 XICS 標記語言，我們的目的是讓網站架設者只要很簡單的利用這套語言，即可輕鬆的收集他站資料、分享本站資料、並具存取權限控管、安全性傳輸，而不需由網站架設者自行撰寫任何程式。

這套語言必須作到下列功能：

1. <import src= "URL">:導入一份 XML 文件，作為來源的文件內容。而我們使用 URL 來指向 XML 文件的來源，如此一來，既可導入本地端檔案，也可指定 HTTP/HTTPS 協定抓取遠端的資訊。
2. <get xpath="xpath">:從導入的 XML 文件，以 XPath 的方式指定一個節點，此節點的內容將經過接下來的步驟後輸出。
3. <drop xpath="xpath">:由 get 得來的節點，以 XPath 的表達方式，再將不需要的部分丟棄。
4. 加入使用者定義標籤：在前三個步驟之後，我們已經從一份外來的 XML 文件，產生另一份 XML 文件，這時候使用者還可以加入新的標籤，使得新的 XML 文件能更符合需求；並且要注意的是，由於 XML 規格限定根元素只能有一個，如果前三步驟產生不只一個根元素，那使用者一定要再定一個新的標籤將其包裹起來。
5. <acl id="id" key="key" default="true" | "false">:經過之前的步驟，我們已經有一份完整的新 XML 文件，在這裡我們將針對不同的客戶端權限(不同的 id/key)，指定他們可以讀取的部分，提供個別化有差異性的內容。若 default="true"，則此段落下的定義將適合所有沒有特別權限的人。
6. <get xpath="xpath">:我們仍然用 get 這個標籤，來定義使用者可以讀取的節點。
7. <drop xpath="xpath">:由 get 得到的節點，再用 drop 丟棄掉不需要的部分。
8. 將最後的結果輸出，如此一來客戶端即可得到個人化的資料。

整份 XICS 標記語言的 DTD 如下

```
<!ELEMENT XICS (acl+,import*) >
<!ATTLIST  acl id ID #IMPLIED
             key CDTA #IMPLIED
             default (true | false) #IMPLIED >
<!ELEMENT acl (get+) >
<!ELEMENT get (drop*) >
<!ATTLIST  get xpath CDATA #REQUIRED >
<!ELEMENT drop xpath CDATA #REQUIRED >
<!ELEMENT import (get+) >
<!ATTLIST  import src CDATA #RQUIRED >
```

接下來我們舉一個人事資料的例子，來看看 XICS 的運作流程。我們將從一份記載實驗室相關資訊的 XML 文件，抽取出人事資料的部分，並依據客戶端權限的不同，提供不同程度的資訊。

我們有一份記載實驗室資訊的 XML 文件，放在目錄 /content/，檔名為 lab.xml，內容如下

```
<lab>
    <lab-info>
        <title> islab </title>
        <phone>59275<phone>
        <addr>708</addr>
    </lab-info>
    <member>
        <person>
            <name>Pokai</name>
            <E-mail>paikai@cis.nctu.edu.tw</E-mail>
            <tel>0926811115</tel>
            <addr>Tao-yuan</addr>
        </person>
        <person>
            <name>Daylong</name>
            <E-mail>daylong@cis.nctu.edu.tw</E-mail>
            <tel>59275</tel>
            <addr>Lean-tou</addr>
        </person>
        <person>
            <name>Moses</name>
            <E-mail>moses@cis.nctu.edu.tw</E-mail>
            <tel>59275</tel>
            <addr>Taipei</addr>
        </person>
    <member>
</lab>
```

我們欲抽取出人事資料的部分以作成網頁，並且不希望任何人都能看到實驗室成員的電話及住址。因此我們撰寫一份 XICS 文件如下。

```
<XICS xmlns:XICS="http://xml.cis.nctu.edu.tw/xics">

  <XICS:acl default="true">                    <!--step 4-->
     <XICS:get xpath="islab-member">
        <XICS:drop xpath="/islab-member/person/tel"/>
        <XICS:drop xpath="/islab-member/person/addr"/>
        </XICS:get>
  </XICS:acl>

  <XICS:acl id="islab" key="708">              <!--step 5 -->
        <XICS:get xpath="/islab-member"/>
  </XICS:acl>

<islab-member>                                 <!--step 3 -->
                                               <!--step 1 -->
  <XICS:import src="http://islab.cis.nctu.edu.tw/lab.xml">
    <XICS:get xpath="/lab/member/person"> <!--step 2-->
  </XICS:import>
</islab-member>

</XICS>
```

這份文件將交由 XICS 元件來處理。處理流程如下

1. 在 import 的 src 屬性裡面指定來源 URL，從引入 http://islab.cis.nctu.edu.tw/lab.xml 檔案。
2. 取出 lab.xml 文件中的/lab/member/person 節點，得到的內容如下

```
        <person>
            <name>Pokai</name>
            <E-mail>paikai@cis.nctu.edu.tw</E-mail>
            <tel>0926811115</tel>
            <addr>Tao-yuan</addr>
        </person>
        <person>
            <name>Daylong</name>
            <E-mail>daylong@cis.nctu.edu.tw</E-mail>
            <tel>59275</tel>
            <addr>Lean-tou</addr>
        </person>
        <person>
            <name>Moses</name>
            <E-mail>moses@cis.nctu.edu.tw</E-mail>
            <tel>59275</tel>
            <addr>Taipei</addr>
        </person>
```

3. 加上使用者自定義的<islab-member>標籤，如此一來即完成一份新的 XML 文件，比原本的 XML 文件更適當的表達了實驗室的成員資料。接下來的客戶權限控管，也將以此內容為依據。

```
<islab-member>
        <person>
            <name>Pokai</name>
            <E-mail>paikai@cis.nctu.edu.tw</E-mail>
            <tel>0926811115</tel>
            <addr>Tao-yuan</addr>
        </person>
        <person>
            <name>Daylong</name>
            <E-mail>daylong@cis.nctu.edu.tw</E-mail>
            <tel>59275</tel>
            <addr>Lean-tou</addr>
        </person>
        <person>
            <name>Moses</name>
            <E-mail>moses@cis.nctu.edu.tw</E-mail>
            <tel>59275</tel>
            <addr>Taipei</addr>
        </person>
</islab-member>
```

4. 如果一個不具有 ID 及 key 的客戶端來存取，將參考<acl default="true">下的定義，獲得<islab-member>元素下，不含<tel>及<addr>資訊的內容如下。

```
<islab-member>
        <person>
            <name>Pokai</name>
            <E-mail>paikai@cis.nctu.edu.tw</E-mail>
        </person>
        <person>
            <name>Daylong</name>
            <E-mail>daylong@cis.nctu.edu.tw</E-mail>
        </person>
        <person>
            <name>Moses</name>
            <E-mail>moses@cis.nctu.edu.tw</E-mail>
        </person>
</islab-member>
```

5. 若客戶端從 URL 需求裡面傳入參數 ID="islab", key="708"，則可得到步驟三產生的全部內容。

以上我們即可獲得，一份符合我們需求的新 XML 文件，並可後續配合 XSL 處理，產生適合觀賞的網頁，或者直接輸出到網路上，把資料分享出去。

我們並以此系統實際建構一校園網站環境，用於校園內行政資料交流，運作情形良好。

五.結論

我們可以預想到，在未來的幾年內，全球資訊網的應用將不只像現在這麼單純，只能提供給人觀看，搜尋引擎只能作關鍵字比對。同樣是把資訊公佈在網路上，XML 網站能夠提供具有語意、不含外觀呈現的純資料。目前討論熱烈的語意網(Semantic Web)[4]，即是希望網路資訊包含語意，能夠讓機器看懂，這樣一來就可以採用程式代理人(agent)來幫我們處理這些資訊，這裡所謂的處理，可能是資料收集、資料搜尋、資料分析等等。舉例來說，如果各航空公司的班次資料若具有語意，我們即可以讓程式代理人幫我們搜尋網路上所有航空公司的資訊，自動幫我們篩選出符合需求的班次。諸如此類的應用還有很多，然而這一切都根基在網路上的系統必須從目前的 HTML 網站轉移成 XML 網站。雖然 XML 網站的遠景如此美好，然而目前架設 XML 網站的情況並不普遍，主因在於相關軟體工具仍不充分，使得網站架設人員必須具備相當的能力，自行撰寫程式來做資料收集等等，一般人不易跨過這個門檻，自然無法造成流行。

在本計畫中，我們改良了現有的 XML 網站系統。我們的系統以現有的 XML 網站系統 Cocoon 為基礎，加強了資訊交流、權限控管的功能。雖然目前網站資訊交流的技術有很多種，但都是由網站架設者自行撰寫程式來處理，並且也不具一般性，針對特定環境打造的資訊交流元件並不容易移植到別的應用場合。因此我們定義了一套 XICS 語言，讓網站架構者輕鬆的指定資料的收集以及資料開放權限。我們的系統具備一般性，適用在各種應用的資料自動交流，如電子商務、行程流程、學術資料分享等等，因此在各領域不需程式專長的人皆可使用。並且支援 HTTPS 來保護傳輸的安全性，可以確保我們的資料來源無法被仿冒，而且資料本身不被偷窺，可適用在對安全性有高要求的應用情況。期望透過本計畫所提出的系統，可以讓 XML 網站系統的架設更普及。

六.參考文獻

[1] World Wide Web Consortium (W3C), Extensible Stylesheet Language (XSL) Version 1.0, Oct 2001, http://www.w3.org/TR/xsl/

[2] Apache Cocoon Project, http://cocoon.apache.org/2.0/index.html

[3] Mohammed J. Kabir, Apache Server 2 Bible, John Wiley & Sons Publish

[4] Tim Berners-Lee, James Hendler and Ora Lassila , The Semantic Web, http://www.scientificamerican.com

[5] 陳柏愷,XML 網站資訊交流系統,碩士論文