行政院國家科學委員會專題研究計畫　期中進度報告

Web 萃取資料之資料管理及資料模式之研究(1/2)

計畫類別： 個別型計畫
計畫編號： NSC93-2213-E-009-088-
執行期間： 93 年 08 月 01 日至 94 年 07 月 31 日
執行單位： 國立交通大學資訊工程研究所

計畫主持人： 吳毅成

報告類型： 精簡報告

處理方式： 本計畫可公開查詢

中　華　民　國 94 年 5 月 30 日

行政院國家科學委員會補助專題研究計畫 □ 成 果 報 告
■期中進度報告

Web 萃取資料之資料管理及資料模式之研究

The Study of Data Management and Data Model for Extracted Web Data

計畫類別：■ 個別型計畫　□ 整合型計畫
計畫編號：NSC　93－2213－E－009－088
執行期間：2004 年　08 月　01 日至　2006 年　7 月　31 日

計畫主持人：吳毅成
共同主持人：
計畫參與人員：

成果報告類型(依經費核定清單規定繳交)：■精簡報告　□完整報告

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

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢
□涉及專利或其他智慧財產權，□一年□二年後可公開查詢

執行單位：國立交通大學資訊工程學系
中　華　民　國　九十四　年　五　月　三十 日

中文摘要

隨著全球資訊網(World Wide Web)的快速發展，如何在這些大量資料中萃取出有用的訊息是非常重要的事情，例如：比價系統須萃取出相關電子商務網站中的有用訊息，如產品名稱、價格、購買方式等；其他，如萃取網路上一些新聞、及出版單位之書籍文章目錄等。在我們過去的國科會計畫中，我們設計了一個資料萃取語言叫做 BODED(Browser Oriented Data Extraction Description)語言及其雛形系統來解決這些問題，並已技術轉移至業界。然而如何對所萃取的資料進一步做一般化的管理是接下來的一件非常重要研究課題。萃取出 Web 資料的管理之基本應用有：資料貯存(data storing)、資料查詢(data query)、網站再生工程(web site reengineering) 、網站資料整合(web site integration)等。

這些 Web 資料管理應用的共通需求，是如何將 Web 的資料與資料庫的資料容易地互轉及整合，以便於管理。然而由於 Web 資料較為不規律，解決這些需求並非易事。首要工作是研究明確及有彈性的 Web 資料模式，如此方能簡化這些需求。因此，本計畫將以過去設計出的 BODE 資料萃取系統為基礎，提出一套適用於 BODE 的資料模式，以便於管理所萃取出的資料。主要工作項目如下：(前四項為第一年計畫，而後四項為第二年計畫)

1. 收集並分析過去 Web 資料模式及資料管理的研究。
2. 提出一套適用於 BODE 的資料模式，並對此模式制定資料定義語言。
3. 研究此模式與關聯式資料庫之對應關係。這研究含如何從這資料定義語言自動產生相對應的關聯式資料庫 schema。
4. 研究 BODE 萃取系統與此模式之對應關係。這研究含如何從 BODE 系統所產生的 script，自動產生相對應的資料定義語言。結合前項研究工作，可使 BODE 萃取系統自動萃取網站資料至關聯式資料庫。
5. 研究並設計如何整合不同網站的資料。這須研究及分析如何將不同的資料定義整合於同一個資料定義。
6. 研究並設計適用於此資料定義語言的查詢語言及系統。
7. 研究並設計適用於此資料定義語言的網站再生工程系統。
8. 以現有電子商務網站為實例，展示所研究的系統對萃取資料之管理能力。

Abstract

With the rapid growth of World Wide Web, it becomes very important to extract information from such a huge amount of database. For example, a price comparison system needs to extract information, such as product names, prices, purchase methods, etc, from some related e-commerce sites. Other examples are to extract news from news providers and to collect categories of publishers. In our past projects from NSC, we defined a data extraction language, named BODED (Browser Oriented Data Extraction Description) Language, and designed to solve the problem of data extraction a system for it whose technique has been transferred to industry. However, the way to manage the extracted data for data management is the next very important research topic. The basic applications include: data storing, data query, web site reengineering, and web site integration.

The common requirements for these basic applications are: how to easily translate and integrate data between web pages and databases. However, since web-based data is less regular, the requirements are not easy. The key research is to study a flexible and definitive web data model in order to simplify the requirements. Therefore, this project proposes a data model suitable for BODE for facilitating the management of extracted data, based on the experiences of designing the BODE system. The work items of this project are: (The first four items are for the first-year project, whereas the next four items are for the second-year project.)

1. Collect and analyze the research of web-based data management and data model in the past.
2. Propose a new Web-based data model suitable for BODE and define a data definition language for the model.
3. Study the mapping relation from the data model to relational databases. This research includes how to map the data definition language to database schemas.
4. Study the mapping relation from BODED scripts to the data definition language. Combined with the previous item, we can map BODED scripts directly to database schemas.
5. Study and design the method to integrate information from different web sites. This requires to study and analyze how to integrate different data definitions into one definition.
6. Study and design the query language and the query system suitable for the data definition language.
7. Study and design the web site reengineering system suitable for the data definition language.
8. Demonstrate our system by using it to manage web data of current e-commerce web sites.

Keywords: (data management, data model, data definition model, data extraction, data storing, data query, web site reengineering, web site integration)

報告內容

一、前言

由於網際網路的普及以及 World Wide Web 的快速發展，有越來越多的資訊，以網頁的方式發布在網際網路上，因此有許多的使用者在網頁上蒐集有用的資訊。全球資訊網（World Wide Web）可說是一個超級大的一個資料庫，內容相當豐富。除了一般生活資訊外，另有大量之專業科技、新聞、商業、金融與人文資料。然而要在 Web 網頁中萃取出有用的資料，並加以彙整與整理，其實並不容易。

過去資料管理(Data Management)的研究主要是在傳統的資料庫上。隨著全球資訊網(World Wide Web)的快速發展，在 Web 上的資料管理也愈形重要。一般而言，資料庫採用較規律且較有結構的關聯模式或物件為基礎的模式，來存取及管理資料。然而，在 Web 上的資料，大多是較為不規律的 semi-structure 格式的文件，如 HTML (HyperText Markup Language) [14]或 XML(Extended Markup Language) [5]。因此，Web 的資料管理是個很難的研究課題。

二、研究目的

在我們過去的國科會計畫中，我們設計了一個資料萃取語言叫做 BODED (Browser Oriented Data Extraction Description)及其雛形系統來解決這些問題。然而如何對所萃取的資料進行儲存，並進一步做一般化的管理也是一件非常重要的事情。萃取出 Web 資料的管理之基本應用有：

- 資料貯存(data storing)：將萃取的資料貯存下來。
- 資訊查詢(information query)：查詢萃取下來的資訊。
- 網站再生工程(Web site reengineering)：可將資料庫內的資料反向顯示成網站。對有些網站年久失修，或使用舊的 Web 伺服器技術，亦可藉由此技術重新修整。
- 資料資料整合(Web site integration)：將不同網站萃取下來的資料整合在同一個資料庫，或甚至整合成一個新的網站。

這些 Web 資料管理應用的共通需求，是如何將 Web 的資料與資料庫的資料容易地互轉及整合，以便於管理。然而由於 Web 資料較為不規律，解決這些需求並非易事。首要工作是研究明確及有彈性的 Web 資料模式，如此方能簡化這些需求。因此，本計畫將以過去設計出的 BODE 資料萃取系統為基礎，提出一套適用於 BODE 的資料模式，以便於管理所萃取出的資料。

三、文獻探討：

過去我們進行許多相關的研究，如 WebOQL[1]、XML Information Set、XQuery and XPath Data Model[15]、Relational Data Model、Object Data Model、OQL、ORDB 等資料庫系統或資料定義語言，並參考 BODE 系統以及語言的特性，改良發展為 DESDL Data Model。在說明如何在網頁上萃取資料以及以有系統的方式儲存這些資料之前，我們先來看一個例子。假設有一個簡化的論文資料庫網站，在這個網站中包含兩層的分類，圖 1 中顯示包含主分類的論文資料庫網站首頁，在主分類網頁中，有超鏈結會連結到含有多個次分類的次分類網頁。其中一個次分類網頁的內容顯示在圖 2 中。次分類網頁中則有超鏈結連結到屬於該次分類的論文列表的網頁。圖 3 則顯示列有好幾篇論文以及其作者名稱、標題

與出版商的論文列表網頁。在論文列表網頁的結尾處，有一個 URL 鏈結到下一個論文列表網頁以顯示更多的論文列表。在論文列表網頁的每一篇文章的標題，皆有一個超鏈結連結到論文網頁。圖 4 顯示一個論文網頁，在論文網頁中，有論文的標題、摘要、作者與出版商等資訊。在每一個作者名稱上，皆有一個超鏈結連結到作者資訊的網頁。圖 5 顯示一個位於 wu.html 的作者資訊網頁，在作者網頁中，包含作者的稱謂、電子郵件地址、電話，以及其相關著作。每一篇著作，則又有超鏈結連結到上一層的論文網頁。

```
<TABLE>
  <TR><TD><A href="db.html">Databases</A></TD></TR>
  <TR><TD><A href="al.html">Algorithms</A></TD></TR>
    . . .
</TABLE>
```

圖 1、包含主分類的論文資料庫網站首頁

```
<TABLE>
  <TR><TD><A href="de.html">Data Extraction</A></TD></TR>
  <TR><TD><A href="dm.html">Data Mining</A></TD></TR>
    . . .
  </TR>
</TABLE>
```

圖 2、位於 db.html，包含次分類的網頁

```
<TABLE border=1 width="100%">
  <TR>
     <TD><A href="p1.html">On the Web Data Extraction Model</A></TD>
     <TD>I-C. Wu, J.-Y. Su, L.-B. Chen </TD>
     <TD>SEKE 2005.</TD>
  </TR>
  <TR>
     <TD><A href="p2.html">A Web Data Extraction Description
                          Language and Its Implementation</A></TD>
     <TD>I-C. Wu, J.-Y. Su, L.-B. Chen </TD>
     <TD>COMPSAC 2005</TD>
  </TR>
</TABLE>

<A href="nextpage.html">NEXT</A>
```

圖 3、位於 de.html 中屬於 Data Extraction 次分類的論文列表網頁

```
<TABLE>
  <TR>
     <TD>Title: </TD>
     <TD><B>On the Web Data Extraction Model</B></TD>
  </TR>
  <TR>
     <TD>Abstract: </TD>
<TD><I>…</I></TD>
  </TR>
  <TR>
     <TD>Authors: </TD>
     <TD> <A href="wu.html">I-C. Wu</A>
          <A href="su.html">J.-Y. Su</A>
          <A href="chen.html">L.-B. Chen</A> </TD>
  </TR>
  <TR>
```

```
      <TD>Publisher: </TD>
      <TD>SEKE 2005</TD>
   </TR>
</TABLE>
```

圖 4、位於 p1.html 的論文網頁

```
<P>I-Chen Wu</P>
<TABLE>
   <TR>
      <TD>Title</TD>
      <TD>Associate Professor</TD>
   </TR>
   <TR>
      <TD>Email</TD>
      <TD>icwu@csie.nctu.edu.tw </TD>
   </TR>
   <TR>
      <TD>Tel</TD>
      <TD>035731855</TD>
   </TR>
   <TR>
      <TD>Publisher</TD>
      <TD>
         <TABLE>
            <TR><TD>
               <A href="p1.html">On the Web Data Extraction Model</A>
            </TD></TR>
            <TR><TD>
               <A href="p2.html">A Web Data Extraction Description
                              Language and Its Implementation</A>
            </TD></TR>
            <TR><TD>
               <A href="p3.html">BODE: A Data Extraction Service
                              Description Language</A>
            </TD></TR>
         </TABLE>
      <TD>
   </TR>
</TABLE>
```

圖 5、位於 wu.html 的作者資訊網頁

在上述的例子當中，我們要萃取所有的論文資料，並且將這些資料自動的儲存到資料庫中。為了要萃取整個網站的論文資料，我們必需瀏覽網站中的所有論文列表網頁，然後連結到論文網頁中萃取相關資料。

在之前的研究[1][2][3][8][10][11][12][15][16]中，要萃取網站上的資料並儲存，通常會使用搜尋引擎、撰寫自訂的 wrapper、或是撰寫網頁查詢程式，然後利用特定的系統將網頁中的資料萃取出來。當資料萃取出來後，再利用撰寫好的資料對應程式，將資料對應儲存到資料庫當中。然而，在萃取資料的過程當中，有一些重要的資訊在網頁中的資料被萃取完成，並進入到下一頁進行其他的萃取之後即遺失。比如頁面之間的超鏈結關係。舉例來說，在圖 2 當中，我們萃取了 Data Extraction 次分類，並萃取了該次分類中的論文列表。當整個網站萃取完成後，該如何知道哪些論文列表中的論文是屬於哪些次分類呢？如果這項資訊未在萃取 Data Extraction 次分類資料與該次分類所鍊結的論文列表網頁資料的期間記錄下來，當這兩頁萃取完成後，該資料即已遺失。因此這一類的資訊，必須要在萃取資料的期間就記錄下來。這樣的記錄通常必須要與網頁資料的萃取系統搭

配，在萃取期間紀錄此項資訊。在過去的研究中，Araneus[5]系統與 WebOQL[1]即是使用這種模式進行相關訊息的紀錄。

目前我們是使用在之前的計畫中所開發的 BODE[17][18]網頁資料萃取系統來萃取網頁上的資料。BODE 系統提供了一個外掛程式(Plug-in)的系統，稱為 BODEDLet[19]。在 BODEDLet 中，我們可以控制 BODE 系統的資料萃取行為，並且讀取 BODE 系統中從網頁上萃取的資料。因此目前我們是使用 BODEDLet 來作為在萃取網頁資料期間鏈結資訊的紀錄。

BODE 系統

BODE 系統是一套 Web 文件之資料自動萃取系統。這套系統主要解決有關網頁文件上的資料自動萃取問題。網頁文件的資料萃取涵蓋兩個部份，一是瀏覽順序，二是資料萃取。

許多網頁並沒有辦法直接以網址來取得。比如許多網站如 104 人力網需要登入才能瀏覽重要資料，有些網站如奇摩站的超連結是經由執行某些程式才會產生。因此在萃取網頁資料前，常常必須瀏覽至所需的網頁後才能做資料萃取。因此網頁間瀏覽的順序相當重要。這個計畫除了解決網頁資料萃取的問題之外，也必須同時解決網頁瀏覽的問題。

為了讓網頁資料萃取更有彈性且更易設計，我們設計了一個以 XML 為基礎的新的描述語言，叫做瀏覽器導向資料萃取描述(Browser Oriented Data Extraction Description)語言，簡稱 BODED，用來描述資料萃取過程中的流覽與資料萃取。

BODED 語言

在 BODED 中，一個 script 程式包函一組網頁的服務，每一個網頁的服務從指定的網頁上萃取資料並處理這些資料。舉例來說，儲存這些資料到資料庫或利用這些資料來瀏覽下一頁。

舉例來說，若要以 BODE 系統萃取圖 1 中的主分類資料，則要以 BODED 語言撰寫一個資料萃取程式。圖 6 顯示一個用來萃取圖 1 中的主分類的 BODED 萃取程式。在這個萃取程式中，會載入論文資料庫網站首頁，然後針對每一個主分類萃取主分類的名稱，然後在該主分類上模擬使用者按滑鼠左鍵一下，以瀏覽到下一頁。BODED 元素(element)是 BODED 程式的最外層元素，包含整個 BODED 萃取程式。這個元素包含兩種不同型態的元素，即 INIT 與 PAGE。從 INIT 元素當中，BODE 系統得知要在第一個瀏覽器中輸入 url 屬性指定的超鏈結以取得網頁，並交由名稱屬性為 MainCat 的 PAGE 元素在該瀏覽器上執行。

FOREACH 元素是一個迴圈，可以用來指定要針對每一個萃取出來的 HTML 元素、屬性或文字進行指定的動作。比如針對每一個主分類，萃取其名稱，並利用主分類上的超鏈結連結到下一頁。在名為 MainCatPage 的 PAGE 元素中，名為 MainCat 的 FOREACH 元素是一個迴圈結構。BODE 系統在執行該 FOREACH 元素時會產生一個由該 FOREACH 名稱為名的變數。在 PAGE 元素所對應的流覽器當中，FOREACH 元素先萃取所有的主分類的超鏈結元素(由 xpath 屬性指定所在的位置)，假設有 n 個。這時 FOREACH 元素所包含的所有元素皆會被執行 n 次。在第 i 次的迴圈中，MainCat 變數會指向第 i 個主分類的超鏈結。在 FOREACH 當中的每一個元素皆是以 MainCat 變數所指向的內容為根節點，在此根節點上使用 xpath 的相對路徑來萃取資料。在圖 6 的 FOREACH 中，使用 VAR 元素來萃取主分類的名稱到名為 MainCatName 的變數中，並儲存於 BODE 系統中的萃取資料集區(extracted data pool)。由於主分類的名稱是在 HTML 文件的文字部分，因此 xpath 屬性中

的值為"./text()"。在這個 XPath 當中，"."代表目前的節點。在圖 6 的例子中，若是目前在第 i 次的迴圈中，則"."指的是第 i 個主分類的超鏈結。而"./text()"則是取出目前節點的的文字部分，也就是第 i 個主分類的名稱。接著利用 EVENT 元素模擬使用者在瀏覽器中的主分類項目上按滑鼠左鍵的 Click 動作。而萃取的規則如果是使用 XPath 語言，則由 xpath 屬性指定。而上述的 MainCat 以及 MainCatName 變數中，包含指向被萃取出來的資料或元素的指標。

```
<BODED Name="CSArchive">
<INIT page="MainCatPage"
url="http://bode.csie.nctu.edu.tw/index.html" />
<PAGE Name="MainCatPage">
<FOREACH Name="MainCat" xpath="//TD/A">
        <VAR name="MainCatName" xpath="./text()" />
        <EVENT name="LinkToSubCat" xpath="." service="SubCat"
                type="OnClick" />
    </FOREACH>
  </PAGE>
 ...
</BODED>
```

圖 6、用來萃取圖 1 主分類的 BODED 萃取程式

此外，BODE 系統包含一個 WYSIWYG 的視覺化的工具，如圖 7，讓使用者能快速的產生能萃取網站資料的 BODED 萃取程式。我們過去的經驗是，透過這視覺化工具的輔助，我們可以在一小時內完成 BODED 萃取程式的設計。

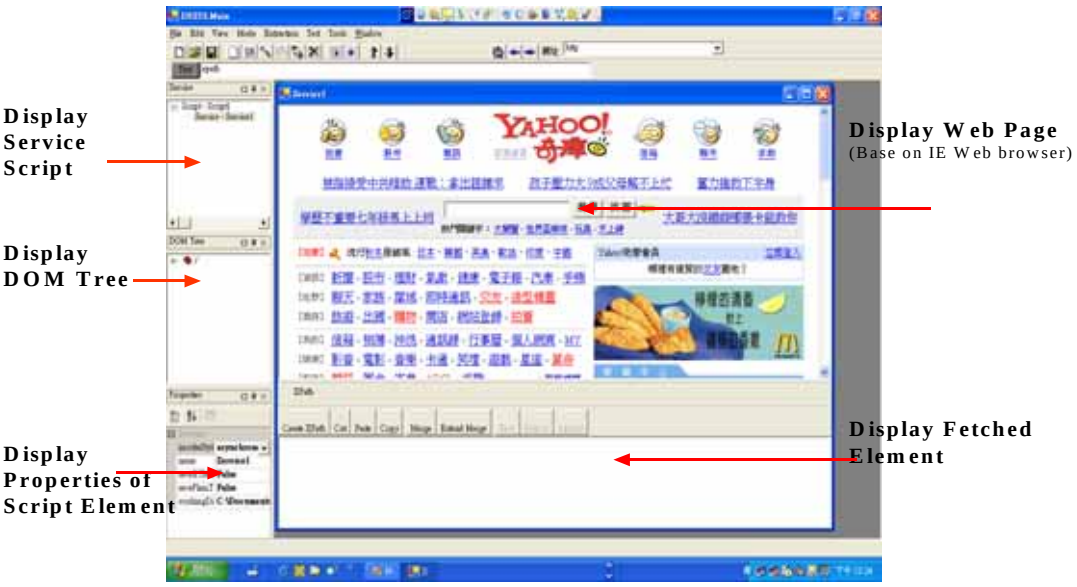

圖 7、BODE 萃取系統的視覺化介面

圖 8 是一個利用 BODE 系統將網頁上的資料萃取出來並儲存到資料庫中的例子。在 BODE 系統中，若是在萃取資料結束後才將資料儲存於資料庫中，則，網頁之間連結的關聯資訊已經消失。在圖 1 中，href 屬性為 db.html 的超鏈結與圖 2 的網頁中的內容具有包含的關係，也就是在圖 1 的網頁中，Database 主分類連結到 db.html 的次分類網頁中。其意義為 Database 主分類包含 Data Extraction 以及 Data Mining 次分類。若超鏈結與目的網頁間的關係消失，則無法復原此項關係。因此這項資訊必須在當使用可連結到下一個網頁的元素來建立下一頁的瀏覽器時，儲存這些資訊。BODE 系統提供一套外掛程式介面，可以在瀏覽與萃取網頁的同時，儲存所萃取的資料。同時，這套萃取系統也可以控制 BODE 系統的

運作。

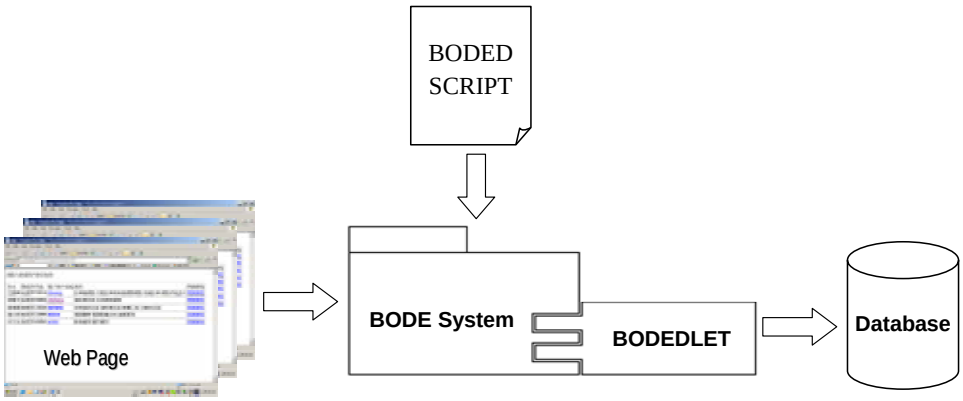

圖 8、使用 BODE 系統萃取資料並儲存到資料庫中

圖 9 是一個利用 BODEDLET 來儲存資料，並瀏覽到下一頁繼續萃取的範例。在

```
<BODED Name="CSArchive">
. . .
<PAGE Name="MainCatPage">
<FOREACH Name="MainCat" from="" xpath="//TD/A">
        <VAR name="MainCatName" xpath="./text()" />
        <EVENT name="LinkToSubCat" xpath="." page="SubCatPage"
type="Info" />
        <BODEDLET name="BODEDB" code="saveDB" archive="db.dll"
                  value="LinkToSubCat"/>
     </FOREACH>
   </PAGE>
  ...
</BODED>
```

圖 9、儲存萃取資料並且控制 BODE 系統瀏覽下一頁的 BODEDLET 範例

四、研究方法

在 Web 資料模式的研究中，我們設計了一個新的資料模式，叫做 BODE 物件模式，並定義描述該模式的語法，用以描述網頁上所萃取的資料。同時，我們也設計一個對應語言。這個對應語言能夠將 BODED script 中指定要萃取的資料對應到資料模式中的特定欄位。

BODE 物件模式

在這個資料模式中，包含了 BODE 物件模式(BODE Object Model)、關聯資料表(Relation Table)與對應語言。BODE 物件模式是以 XML 為基礎的語法來描述。圖 10 顯示一個描述圖 1 的 BODED 語言所萃取的主分類名稱資料的 BODE 物件模式定義。在圖 10 中定義了名為 MainCat 的類別。在該類別中，有一個欄位叫做 MainCatName，其型態是字串(string)，長度為 20。

BODE 物件模式中定義了兩個標籤，一個是 CLASS，一個是 FIELD。CLASS 標籤定義資料物件的類別。其內含 FIELD 標籤。FIELD 標籤則是描述類別中的欄位。CLASS 標籤包含兩個屬性，一個是 name，及該 CLASS 的名稱，另一個是 key，指定該類別的物件是以哪些欄位當作主鍵。在物件資料庫中，具有同樣主鍵的資料會被視為是同一筆資料。

FIELD 標籤包含三個屬性，第一個是 name，為該欄位的名稱，第二個是 type，為該欄位的資料型別，第三個是 length，是該欄位所儲存的資料的長度。

```
<CLASS name="MainCat" key="Name" >
   <FIELD  name="MainCatName" type="string" length="20" />
</CLASS>
```

圖 10、以 Data Model 定義語言定義主分類項目資料

BODE 資料模式對應語言

為了要讓資料的萃取與儲存能夠自動化的進行，因此我們設計了一個能夠將 BODED 萃取程式對應到 BODE 物件模式中的資料定義的語言。該語言利用 BODED 語言的標籤結構，將資料對應到物件模式中。圖 11 的對應規則即是將圖 9 中的 BODED 語言對應到圖 10 的類別定義中。在圖 11 中，FOREACH 元素會對應到圖 10 中的 MainCat 類別。而 FOREACH 元素中的 VAR 元素則對應到 MainCat 類別中的 Name 欄位。因此在 BODE 系統執行當中，便可依據該對映程式將資料自動的儲存於資料庫中。

```
<MAP script="CSArchive">
  <PAGE name="MainCatPage" . . . >
    <FOREACH name="MainCat" class="MainCat" . . . >
      <VAR name="MainCatName" field="Name" />
      . . .
    </FOREACH>
  </PAGE>
  . . .
</MAP>
```

圖 11、將 BODED 程式中的萃取資料對應到資料模式中的主分類項目資料的對照程式

BODE 的資料模式處理系統

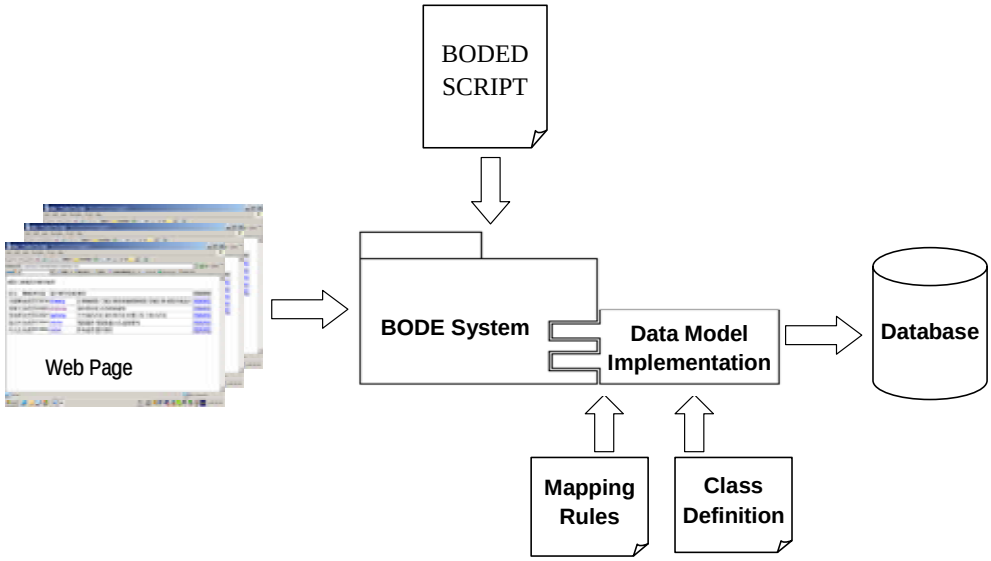

圖 12、在 BODE 系統中的資料模式實作

由於資料的對應與儲存必須在 BODE 系統執行當中進行。因此在 BODE 系統上，是以一個 BODEDLET 外掛程式來實作資料模式處理系統，如圖 12。這個資料模式實作(Data Model Implementation)在執行時讀取類別定義與對應規則，將萃取出來的資料對應到物件資料庫中。並建立資料之間的關聯。

資料關聯

在網頁當中的資料與資料之間，有些會具有特定的關係，這樣的關係我們稱為關聯。以資料庫的角度來看，關聯大致有四種，即一對一、一對多、多對一以及多對多等關聯，如圖 13。在網頁當中的資料，也存在這四種關係。例如圖 4 的論文網頁當中，論文的標題

(Title)、摘要(Abstract)與出版商(Publisher)具有一對一的關係。也就是一篇論文當中，會有一個標題、一份摘要以及由一個出版商出版。這些資料都是屬於一篇論文中的屬性，且對於一篇論文當中，具有唯一性。因此我們可以建立一個論文的類別，如圖 14 中的 Paper 類別，並將這些彼此之間是一對一的資料，作為論文類別中的欄位。在 BODED 萃取程式中，要萃取的資料是由 VAR 標籤來描述該資料位於何處。由於 BODED 萃取程式的 XML 結構中，僅有變數類型的標籤才能指定要萃取的資料。因此，這些標籤會被對應到類別中的欄位。為了使資料定義更加簡單，在 BODE 資料模式中規定 CLASS 當中不能包含其他 CLASS。因此 CLASS 沒有辦法表達多層的關係。所以可以包含變數的其他標籤由於包含了欄位，因此會被對應到類別。

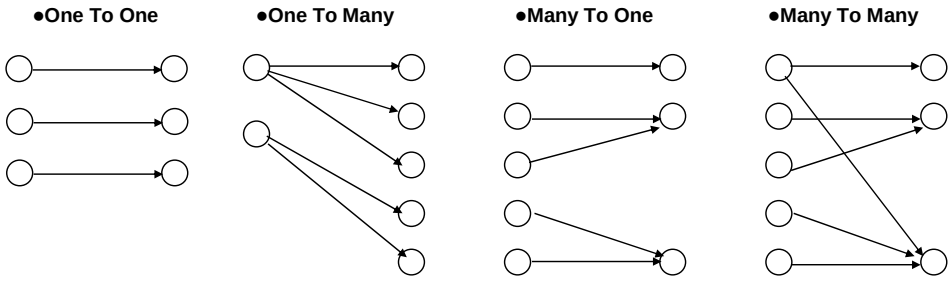

圖 13、資料間的一對一、一對多、多對一與多對多關聯

```
<CLASS name="Author">
   <FIELD name="AuthorName" type="string" length="20" />
</CLASS>

<CLASS name="Paper" key="Title" >
   <FIELD name="Title" type="string" length="50" />
   <FIELD name="Abstract" type="string" length="300" />
   <FIELD name="Publisher" type=" string" length="100" />
</CLASS>
```

圖 14、圖 4 論文網頁的類別定義

```
<PAGE name="Paper">
   <VAR name="Title" xpath="//TR[0]/TD[1]/B[0]" />
   <VAR name="Abstract" xpath="//TR[1]/TD[1]/I[0]" />
   <VAR name="Publisher" xpath="//TR[3]/TD[1]" />
   <FOREACH name="Authors" xpath="//TR[2]/TD[1]/A">
      <VAR name="AuthorName" xpath="." />
      <EVENT name="LinkToAuthor" xpath="." page="Author" type="OnClick" />
   </FOREACH>
</PAGE>
```

圖 15、用來萃取圖 4 論文網頁的 BODE 程式片段

```
<PAGE name="Paper" class="Paper" . . .>
   <VAR name="Title" field="Title" />
   <VAR name="Abstract" field="Abstract" />
   <VAR name="Publisher" field="Publisher" />
   <FOREACH name="Authors" class="Author"
            parentrelation="Paper">
      <VAR name="AuthorName" field="AuthorName" />
   </FOREACH>
</PAGE>
```

圖 16、將圖 15 之 BODE 程式對應到圖 14 資料模式中的類別定義的對應程式

舉例來說，圖 16 是一個能將圖 15 的 BODED 萃取程式中所萃取的欄位對應到圖 14 所定義的類別中的對應規則。由於圖 15 中的 VAR 元素會被對應到 CLASS 定義中的 FIELD，而圖 15 的 PAGE 包含對應到 Paper 類別中的欄位的 VAR 元素，因此 PAGE 被對應到 Paper 類別。在 BODE 資料模式的對應規則中，規定 BODED 萃取程式中的 VAR 元素的上一層元素必須被對應到一個類別。

然而在圖 4 當中，除了論文的標題、摘要與出版商之外，還有作者的名字。在作者的名字上，有超鏈結連結到作者網頁中。由於一篇論文可能不只有一個作者，因此一篇論文與其作者之間可能具有一對一或是一對多的關聯。圖 15 是一個用來萃取圖 4 的網頁的 BODED 萃取程式片段。在這個萃取程式中，除了萃取論文的標題、摘要與出版商之外，並使用 FOREACH 元素依序萃取作者的名稱與超鏈結。並利用連結到作者網頁的超鏈結，建立一個新的瀏覽器，並連結到作者網頁當中，以名稱為 Author 的 PAGE 服務繼續萃取作者的資料。

當資料之間具有一對多、多對一或多對多的關係時，就沒有辦法以目前單純的 CLASS 定義來定義這樣的關係。為了保持簡單的資料定義格式，並且同時保留可以彈性的描述各種物件之間的關係，因此對於物件之間的對應關係，我們採用一個稱為關聯資料表的關聯表格來記錄。

Container Class ID	Element Class ID	Container Class Name	Element Class Name	Script Name
670296112	670296922	PaperList	Paper	CSArchive
670296922	670297201	Paper	Author	CSArchive

圖 17、描述論文列表/論文以及論文/作者的關聯資料表片段

圖 17 是一個關聯資料表的片段。在關聯資料表中，為了要建立物件與物件之間的關聯性，BODE 資料模式使用以下方法：

(1) 每一個物件具有一個獨一無二的識別碼，簡稱 ID。

(2) 將物件與物件之間的關聯，利用物件的 ID 紀錄在關聯資料表中。

使用此關聯資料表具有以下的優點：

(1) 可以紀錄各種關連性，如一對一、一對多、多對一或多對多。由於在 Web 資料萃取的應用上，在某些網站的某些資料之間可能具有一對一的關係，但是在其他網站上，同樣的資料之間可能具有一對多或多對一等關係。

(2) 同樣的資料之間，可能存在不只一種關係。關聯資料表可以紀錄某一項資料物件所具有的所有可能的關係。經由此一關聯資料表，可以得知某兩個物件之間具有那些關聯性。

由於在萃取圖 4 中資料時，由於一篇論文可能擁有多位作者，因此這個關係是一對多的關係，圖 14 顯示論文相關資訊的類別定義。在圖 14 中定義了 Paper 以及 Author 類別，在 BODE 資料模式中如何描述 Paper 與 Author 類別之間的關係呢？

五、結果與討論

在 BODE 資料模式的研究中，我們設計了一個 BODE 物件模式，並且定義了 BODE 物件定義語言以及 BODE 資料模式對應語言。此外，在 BODE 資料模式中，利用關聯資料表(Relation Table)，建立物件之間的各種關聯。因此 BODE 資料模式除了具有簡單的資料定義格式，且同時可以更彈性的方式描述各種物件之間的關係。

在 BODE 物件定義語言上，利用 CLASS 與 FIELD 標籤建立資料物件的類別，而利用與 BODE 萃取語言一樣的語言結構，來定義如何將以 BODE 萃取語言撰寫的萃取程式中所萃取的資料對應到 BODE 物件模式中的類別的規則。

由於利用與 BODE 萃取語言一樣的語言結構來描述萃取資料與資料類別的對應關係，因此資料類別與對應規則可以由 BODED 萃取程式來自動產生。省去設計資料類別的時間。

六、參考文獻

[1] Gustavo Arocena and Alberto Mendelzon. “WebOQL: Restructuring Documents, Databases, and the Web”, In *Proceedings of ICDE*, 1998, Orlando, Florida.

[2] R. Baumgartner, S. Flesca, G. Gottlob. “Visual Web Information Extraction with Lixto”, In Proceedings of the 27th International Conference on Very Large Data Bases (VLDB'01), 2001.

[3] Alin Deutsch, Mary Fernandez, Daniela Florescu, Alon Levy, and Dan Suciu. “XML-QL: A Query Language for XML”, In *Proceedings 8th International World Wide Web Conference (WWW8)*, 1999. Computer Networks 31(1116) : 1155-1169.

[4] M.Fernandez et al.,“Declarative Specification of Web Sites with STRUDEL,” VLDB Journal,vol.9,no.1,2000,pp.38-55.

[5] G. Mecca, P. Merialdo, P. Atzeni. “ARANEUS in the Era of XML”, *IEEE Data Engineering Bullettin, Special Issue on XML*, September, 1999

[6] Steve Holzner. “XML Complete”, McGraw-Hill, 1998.

[7] David Konopnicki , Oded Shmueli. “Information gathering in the World-Wide Web: the W3QL query language and the W3QS system”, *ACM Transactions on Database Systems (TODS)*, Volume 23 Issue 4, Dec. 1998.

[8] Phillip Merrick, Charles Allen. “Web Interface Definition Language”, W3C NOTE, Sep. 1997. http://www.w3.org/TR/NOTE-widl.

[9] Microsoft Corporation. “WebBrowser Control”, Programming and Reusing the Browser, MSDN Library, 2002. http://msdn.microsoft.com/library/default.asp?url=/workshop/browser/webbrowser/browser_control_node_entry.asp.

[10] Jonathan Robie, Joe Lapp, David Schach. “XQL：XML Query Language”, Workshop on XML Query Languages, Dec. 1998. http://www.w3.org/TandS/QL/QL98/pp/xql.html.

[11] W3C Consortium, “XML Query”, Apr. 2000. http://www.w3c.org/XML/Query.

[12] W3C Consortium. “XQuery 1.0: An XML Query Language”, W3C Working Draft, 16 Aug. 2002. http://www.w3.org/TR/xquery/.

[13] W3C Consortium. “Extensible Markup Language (XML) 1.0 (Second Edition)”, W3C Recommendation, Oct. 2000. http://www.w3.org/TR/2000/REC-xml-20001006.

[14] W3C Consortium, “HTML 4.01 Specification” W3C Recommendation, Dec. 1999. http://www.w3.org/TR/html4/.

[15] W3C Consortium. “XQuery 1.0 and XPath 2.0 Data Model”, W3C Working Draft, Aug. 2002. http://www.w3.org/TR/query-datamodel/.

[16] W3C Consortium. “XML Path Language (XPath) 2.0”, W3C Working Draft, Aug. 2002. http://www.w3.org/TR/xpath20/.

[17] I-Chen Wu, J.-Y. Su, and L.-B. Chen. “On the Web Data Extraction Model”, submitted to the Proceedings 17th International Conference on Software Engineering and Knowledge Engineering (SEKE2005), 2005.

[18] I-Chen Wu, J.-Y. Su, and L.-B. Chen. “A Web Data Extraction Description Language and Its Implementation”, The 29th Annual International Computer Software and Application Conference (COMPSAC 2005), Edinburgh, Scotland, July, 2005.

[19] I-Chen Wu, J.-Y. Su, and L.-B. Chen. “The User Guide of the BODEDlet Plug-in System”, internal document, 2004.

七、計劃成果自評

本計劃兩年之工作項目如下：

1. 收集並分析過去 Web 資料模式及資料管理的研究。
2. 提出一套適用於 BODE 的資料模式，並對此模式制定資料定義語言。
3. 研究此模式與關聯式資料庫之對應關係。這研究含如何從這資料定義語言自動產生相對應的關聯式資料庫 schema。
4. 研究 BODE 萃取系統與此模式之對應關係。這研究含如何從 BODE 系統所產生的 script，自動產生相對應的資料定義語言。結合前項研究工作，可使 BODE 萃取系統自動萃取網站資料至關聯式資料庫。
5. 研究並設計如何整合不同網站的資料。這須研究及分析如何將不同的資料定義整合於同一個資料定義。
6. 研究並設計適用於此資料定義語言的查詢語言及系統。
7. 研究並設計適用於此資料定義語言的網站再生工程系統。
8. 以現有電子商務網站為實例，展示所研究的系統對萃取資料之管理能力。

第一年的部份已即將如期完成前四項工作。相信這將會作為第二年計畫好的基礎。

此外，我們做的過程中，亦有發表論文到 SEKE 2005 [17]及 COMPSAC 2005 [18]（接受機率：72/278~25.9%），兩個不錯的國際會議。相信這表示我們 BODE 系統的研究已經被國際相關領域肯定。