行政院國家科學委員會專題研究計畫　成果報告

Web 文件自動萃取系統之視覺化工具之研究

計畫類別：個別型計畫
計畫編號：NSC91-2213-E-009-114-
執行期間：91 年 08 月 01 日至 92 年 07 月 31 日
執行單位：國立交通大學資訊工程研究所

計畫主持人：吳毅成

計畫參與人員：洪憲忠、許傳杰、簡廉哲、蔡銘韓

報告類型：精簡報告

處理方式：本計畫可公開查詢

中　華　民　國 93 年 2 月 5 日

中文摘要

為了萃取出有用的訊息，我們過去研究過相關的 XQL、XML-QL 與 WIDL (Web Interface Definition Language)語言，並已將之改良制定出了一套資料萃取系統之描述語言——DESDL (Data Extraction Service Description Language)，作為資料萃取系統之萃取依據。我們過去研究經驗顯示，對 DESDL 而言，每個網頁的萃取描述語句通常都不會超過百行，這已經大幅簡化萃取的複雜度。同時，DESDL 的遞迴式萃取法，可比 Robot 更有效率地自動萃取整個網站資料。然而問題是：

1. 使用者必須熟悉的萃取描述語法。
2. 使用者必須耗時間來完成並測試萃取描述語法。
3. 當網頁內容架構稍做調整時, 查詢語法便須重新設計。

因此本計畫，繼續研究如何用視覺化工具來自動產生或協助使用者產生描述語言。本計畫的主要工作項目主要包含如下部份：

1. 研究與設計一個類似例子查詢(Query by Example)方式為基礎之視覺化資料萃取介面之萃取工具，來自動產生 DESDL 萃取描述語言。
2. 研究並設計一最佳化演算法, 使得所自動產生之萃取描述語言之查詢敘述為最精簡, 精簡查詢敘述的好處除了簡化維護的需求外，還能對網站文件的變更，具有相當程度的錯誤容忍度。
3. 實作此視覺化工具的雛形系統。

Abstract：

In order to extract useful information from Web, we defined an extraction script language, named DESDL (Data Extraction Service Description Language), in the past. From our research, we find that most web page extraction is no more than 100 lines, more accurate and efficient than the current robot tools. In this project, we want to further solve the following problem:

1. Users must be familiar with the script language.
2. Users must spend a lot of time writing and testing scripts.
3. When web pages are changed, scripts must be changed accordingly.

Therefore, the project will continue to research on the visualization tools for DESDL and include the following working items.

1. Study and design an extraction visualization tool, used to automatically generate DESDL scripts.
2. Study and design the optimization for the algorithms of shortening Xpath scripts. This will make maintenance easier and help scripts more fault-tolerant.
3. Implement a prototype for the visualization tool.

Keyword: data extraction, navigation, DESDL, plug-in.

一、前言

Web 文件之資料自動萃取引擎之分析研究本計劃將研究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，叫做資料萃取服務描述語言(DESDL), 用來描述資料萃取服務中的流覽與資料萃取。在 DESDL 中，一個 script 程式包函一組服務，每一個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。DESDL 使得控制瀏覽順序與資料萃取變的更加容易。簡單的說，DESDL 具有下列的特色。

(1) 使用 XPath 當作萃取網頁內部資料的查詢敘述格式。

(2) 能以特定的順序瀏覽網頁。

(3) 可以填寫表單並啟動下一個服務來萃取提交表單後的下一頁。

(4) 支援外掛程式，這裡稱為 DESDLet 來處理萃取的資料。舉例來說，儲存到資料庫或瀏覽下一頁。

(5) 與目前的瀏覽器的規格一致。

(6) 模擬按一下的動作並啟動下一個服務來萃取下一頁。

二、研究目的

我們過去研究經驗顯示，對 DESDL 而言，每個網頁的萃取描述語句通常都不會超過百行，這已經大幅簡化萃取的複雜度。同時，DESDL 的遞迴式萃取法，可比 Robot 更有效率地自動萃取整個網站資料。然而問題是：

1. 使用者必須熟悉的萃取描述語法。
2. 使用者必須耗時間來完成並測試萃取描述語法。
3. 當網頁內容架構稍做調整時，查詢語法便須重新設計。

因此本計畫，繼續研究如何用視覺化工具來自動產生或協助使用者產生描述語言。

三、研究背景

過去我們是從許多相關的研究，如 XQL、XML-QL、XQuery、WIDL、WebOQL、W3QL 等萃取系統或語言，改良發展為 DESDL。由於本計畫是研究發展 DESDL 的開發系統。我們將專注在 DESDL 語言的分析及回顧。

首先，我們先來看圖一（如下）的一個 HTML 文件，我們可以用圖二的 DESDL script 把 HTML 文件中的 Title 及 Author 資料，萃取出來。其中，變數萃取的表達格式是採用 W3C 的標準萃取格式 XPATH。例如： "//b[0]//text()"。

```
. . .
  <td><b><a href="http://...">
        DESDL: A Data Extraction Service
        Description Language</a></b>
      <b>I-Chen Wu,  . . . </b>
      <i>Proceedings of . . .</i>
  </td>
  <td><b><a href="...">
        WebOQL: Restructuring Documents,
        Databases, and the Web </a></b>
      <b> G. Arocena and ... </b>
      . . .
  </td>
. . .
```

圖一：一個 HTML 的例子

```
<DESDL>
  <INIT SERVICE="GetPaper"
       URL="..."/>
  <SERVICE NAME = "GetPaper"/>
    <VAR NAME="Title"
         PATH="//b[0]//text()"/>
    <VAR NAME="Author"
         PATH="//b[1]/text()"/>
  </SERVICE>
</DESDL>
```

圖二：萃取圖一的 DESDL Script

```
<DESDL>
  <INIT SERVICE="GetPaper"
        URL="..."/>
  <SERVICE NAME="GetPaper">
    <VAR NAME="SearchBase"
         PATH="//table/tr/td"/>
    <FOREACH FROM="$SearchBase">
      <VAR NAME="Title"
           PATH="b[0]//text()"/>
      <VAR NAME="Link"
           PATH="b[0]/a/@href"/>
      <VAR NAME="Author"
           PATH="b[1]/text()"/>
      . . .
      <INVOKE SERVICE="GetAbs"
              URL="$Link"/>
    </FOREACH>
  </SERVICE>
  <SERVICE NAME="GetAbs">
    ...
  </DESDL>
```

圖三：Multi-way Navigation 模式的例子

在瀏覽模式上，本計畫是採用 Multi-way Navigation 的模式。Multi-way Navigation 模式就是，能同時瀏覽到下一個要搜尋的網頁去萃取資料。主要是利用 FOREACH 的標籤(Tag)，來一次取出數個資

料。例如：圖三中，用 FOREACH 一次取出數本書的資料在 SEARCHBASE 中。

DESDL 支援 plug-in 的寫法，這樣可以讓程式設計者容易地加入自己所希望的功能。DESDL 的 plug-in 程式，我們稱之為 DESDLet，圖四顯示一個 DESDLet 的例子，在此例子中，在下一個萃取瀏覽動作前, 要先執行 RemoveRep.dll 這個 DESDLet 來去除重複的網頁。

```
<DESDL>
  <INIT SERVICE="GetPaperAttr"
        URL="..."/>
  <SERVICE NAME="GetPaperAttr">
    <VAR NAME="SearchBase"
         PATH="//table/tr/td"/>
    <FOREACH FROM="$base">
      <VAR NAME="Title"
           PATH="b[0]//text()"/>
      <VAR NAME="Link"
           PATH="b[0]/a/@href"/>
      <INVOKE SERVICE="GetAbs"
           DESDLET="RemoveRep.dll"
           URL="$Link"/>
    </FOREACH>
  </SERVICE>
</DESDL>
```

圖四：DESDLet 的例子

```
<Script language=Javascript>
   function Directto(){. . .
     window.open("http://www.csie.nctu
     ...")
      . . . }
</Script>
<FORM NAME="Form1"
     ACTION="results.cfm?...
     METHOD=POST onSubmit="Directto()">
   <b>Search DL</b>
   <INPUT TYPE="Text" NAME="query"
     VALUE="">
   <INPUT TYPE="Submit" NAME="Go">
</FORM>
. . .
```

圖五：模擬填表的例子

DESDL 萃取的方式，容許使用者用填表的方式，或用模擬按鍵的模式來產生下一個網頁瀏覽，如圖五。

四、研究方法

本計畫的研究重點是設計一個 Web 文件自動萃取系統之視覺化引擎，並作分析研究。我們進行的步驟如下：

1. 分析研究現有之資料萃取方法及 Script 語言，如 DESDL、XQuery、XQL、XQL-ML、XPath、DOM、WIDL。
   在此精簡報告, 我們將 DESDL 為主要的研究對象。
2. 研究設計一個能解讀並產生 DESDL scripts 之視覺化萃取引擎，並設計此引擎。
3. 研究並分析 XPATH 之最佳化 (optimization)。
4. 實作 DESDL 視覺化萃取引擎之雛形系統。

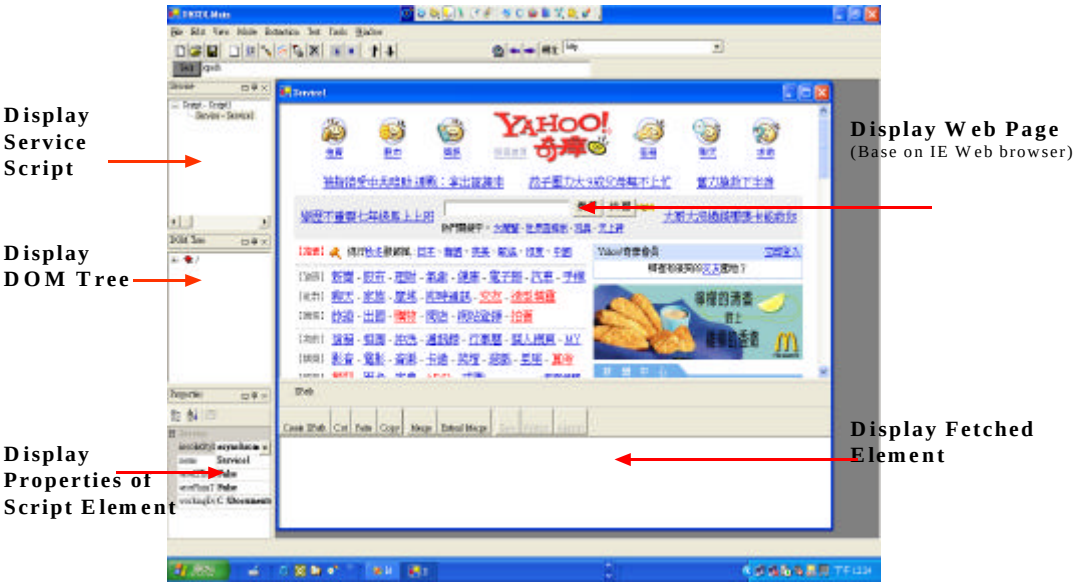
五、成果與討論

由於不同的網站，其網頁中的結構與格式有很大的差異，因此在設計這類萃取 scripts 時，指定資料所在的位置並不是可以很快的寫出。我們過去的經驗是，若沒有工具輔助，則設計一個電子商務網站的萃取 script 需要花約半天到一天的工夫。

為了更進一步縮短設計的時間，我們研究設計了一個 WYSIWYG 的視覺化的工具，讓使用者能快速的產生能萃取網站資料的 DESDL script。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 DESDL scripts 的設計。

此視覺化工具，利用所謂的 Model-View-Control 的模式設計，建構在微軟最新的.NET 技術。此視覺化工具的外觀如下圖，各個區域分別敘述如下：

- "Web Page"的區域：顯示對應網頁。
- "Fetched Element"的區域：顯示所選取資料的 XPATH。
- "Service Script"的區域：以 DOM tree 的形式顯示 DESDL 的 script。
- "DOM Tree"的區域：以 DOM tree 的形



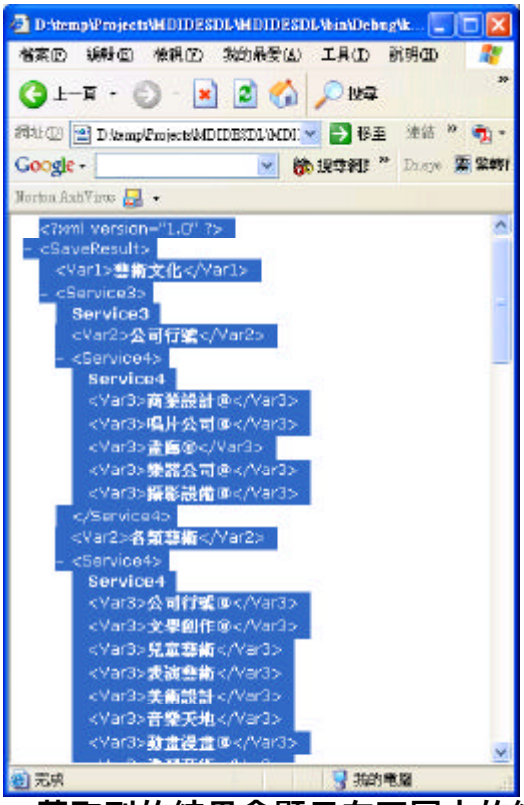
式顯示此網頁。

- “Properties”的區域：顯示 DESDL 標記(element)的屬性值。

此系統分編輯模式與執行模式。在編輯模式下，使用者可先瀏覽所要的網頁於“Web Page”的區域，直接藉由點選來選擇所要的資料，其 XPATH 路徑可顯示於“Fetched Element”的區域，這可大為簡化路徑的編輯。由於在許多的網頁上，資料常分散放置在網頁中的特定區域，當網頁中的資料量較多時，點選並不容易。因此，在 DESDL Visual Tool 中，可以先點選兩個或兩個以上的資料，如下圖，再讓此工具自動的辨識相關資料所在的位置，並選取全部的相關資料。

要達成這項功能，Visual Tool 上有一個 XPATH 路徑的合併與擴充按鈕，點選這項按鈕，系統會自動修改 XPATH 路徑使該路徑能萃取同類型的所有資料，如下圖。使用者也可以自行編輯此路徑。當編輯完成後，即可利用儲存的功能將 XPATH 路徑存入”Service Script”的各項可儲存 XPATH 的項目中。實際上，此系統還用許多技術來使編輯的路徑更具有彈性。然後，使用者可利用取得的路徑，在“Service Script”及“Properties”區域，編輯 DESDL scripts。

編輯完成後即可將 DESDL script 存檔或執行 script。此系統會依據 script 中的描述，依序遞迴的進行資料萃取，填入表單資料，以及進行瀏覽下一個網頁。在執行時瀏覽順序採用深先(depth first)的順序進行以避免佔用過多記憶體。



萃取到的結果會顯示在下圖中的結果視窗，我們可以利用結果視窗將結果儲存成 XML 檔案，如下圖，或利用 DESDLet 將結果儲存至資料庫中。

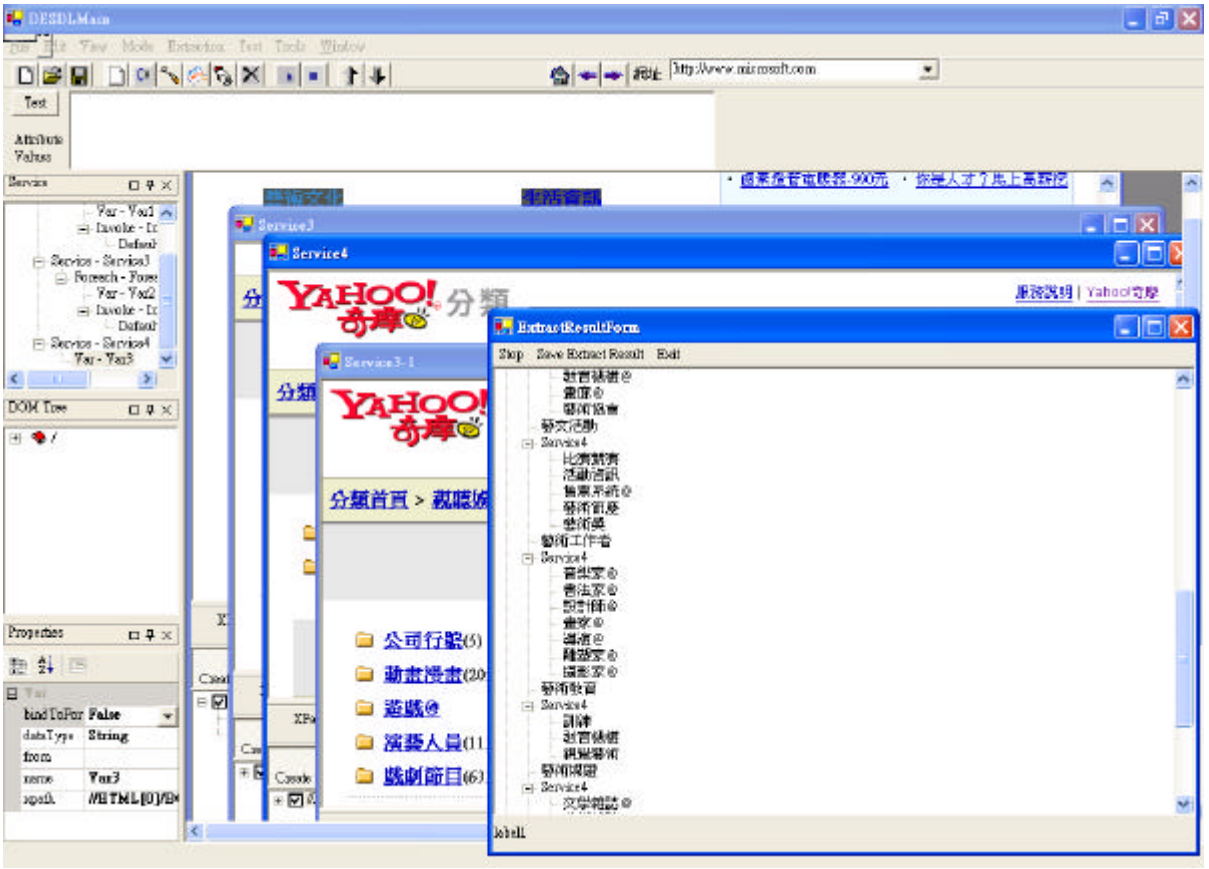
討論

在 Visual Tool 中，我們使用瀏覽器元件來瀏覽網頁，因此在視覺與操作上與使用瀏覽器相當一致。此外，本系統也提供填寫表單, 外掛程式以及模擬 click 動作以驅動網頁內嵌的 scripts 等功能。因此可以提供更接近人實際操作瀏覽器進行資料瀏覽的動作，因此在獲得正確網頁內容方面，提供較好的能力。

本計劃對網頁資料的自動萃取所需的瀏覽與資料萃取問題，提出一套較有彈性的解決方法，利用一個以 XML 為基礎的資料萃取服務描述語言來描述瀏覽過程與萃取的資料位置。並提供一個 DESDL 的視覺化操作介面，使得在操作，控制瀏覽順序與資料萃取上變的更加容易。

計劃成果自評

本計劃對網頁資料的自動萃取所需的瀏覽與資料萃取問題，提出一套較有彈性的解決方法，利用一個以 XML 為基礎



的資料萃取服務描述語言來描述瀏覽過程與萃取的資料位置。並提供一個 DESDL 的視覺化操作介面，使得在操作，控制瀏覽順序與資料萃取上變的更加容易。本系統具有下列的特色。

(1) 使用與瀏覽器一致的視覺介面與相似的操作環境，減少操作者須改變習慣的麻煩。
(2) 與目前的瀏覽器的網頁資料解析規格一致。
(3) 可以填寫表單並啟動下一個服務來萃取提交表單後的下一頁。
(4) 模擬按一下的動作並啟動下一個服務來萃取下一頁。
(5) 能以特定的順序瀏覽網頁。
(6) 支援外掛程式，這裡稱為 DESDLet 來處理萃取的資料。舉例來說，儲存到資料庫或瀏覽下一頁。
(7) 使用 XPath 當作萃取網頁內部資料的查詢敘述格式。

我們除了實作了 DESDL 視覺化操作雛形系統，並從這個實作中，研究到相關的理論，如模擬按鈕的理論，及自動擴展 XPATH（如第五段所提）等。

另外，由於本雛形系統理論與實際兼顧，此系統亦已經經由國科會技術移轉至廠商。因此，自評此計畫對學術研究及台灣相關業界有相當的貢獻。

References:

[1] Association for Computing Machinery. “ACM Portal to Computing Literature”, ACM, New York, 2002. http://portal.acm.org/portal.cfm.

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

[3] G. Arocena. “WebOQL: Exploiting Document Structure in Web Queries”, Master's Thesis, University of Toronto, 1997.

[4] 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.

[5] Ebay Inc., "ebay: The World’s Online Marketplace", 2002. http://www.ubid.com.tw.

[6] Alan Freier, Philip Karlton, Paul Kocher. “The SSL Protocol Version 3.0”, Internet Draft, Mar. 1996. http://wp.netscape.com/eng/ssl3/ssl-toc.html

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

[8] D. Konopnicki, O. Shmueli. “W3QL: A Query System for the World Wide Web”, in *Proceedings of the 21th International Conference on Very Large Databases*, Zurich, 1995.

[9] 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.

[10] Sean Mcgrath. “XML by example: building E-commerce applications”, Prentice Hall PTR, 1998.

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

[12] 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.

[13] Netscape Communications Corporation. “Secure Sockets Layer”, 2000. http://wp.netscape.com/security/techbriefs/ssl.html.

[14] Openfind Information Technology, Inc. "Openfind Enterprise Portal Technology Provider", 2002. http://www.openfind.com.tw.

[15] 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.

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

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

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

[19] W3C Consortium. “Hyper Text Markup Language”, Jan. 1998. http://www.w3c.org/Markup/.

[20] W3C Consortium, “HTML 4.01 Specification”

W3C Recommendation, Dec. 1999.
http://www.w3.org/TR/html4/.

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

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

[23] W3C Consortium. "XML Pointer Language (XPointer) Version 1.0", W3C Working Draft, Aug. 2002.
http://www.w3.org/TR/xptr/.

[24] Yahoo! Inc, "Yahoo Search Engine", 2002.
http://www.yahoo.com.