

國立交通大學

資訊學院 資訊學程

碩士論文

社群網路 API 之開發與研究

The Development and Research of Social Web API



研究生：張貴智

指導教授：吳毅成 教授

中華民國九十八年四月

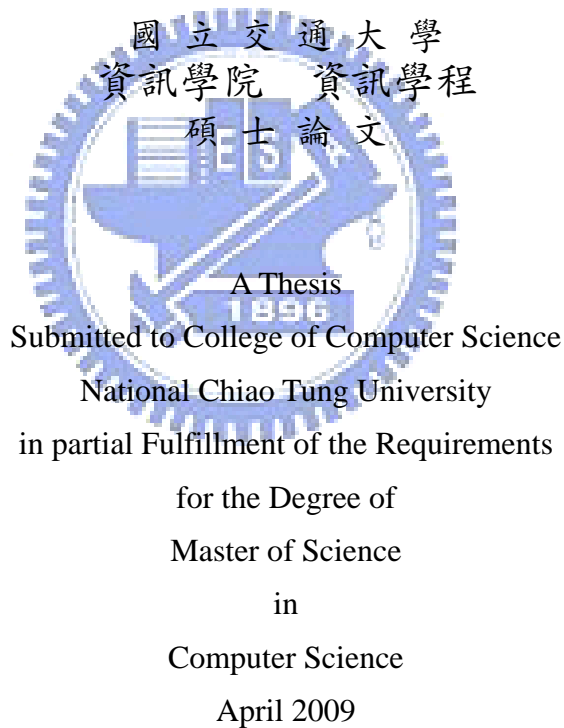
社群網路 API 之開發與研究
The Development and Research of Social Web API

研究生：張貴智

Student : Kuei-Chih Chang

指導教授：吳毅成

Advisor : I-Chen Wu



Hsinchu, Taiwan, Republic of China

中華民國九十八年四月

社 群 網 路 A P I 之 開 發 與 研 究

學生：張貴智

指導教授：吳毅成 教授

國 立 交 通 大 學 資 訊 學 院 資 訊 學 程 碩 士 班

摘 要

本研究的目的是希望發展一個系統，其能夠以最簡單的方式，將任何網站轉變成為一個社群網站，並賦予這個網站具有社群網站大部份的基本功能。本系統的 Client 端元件是一 Firefox 的 Extension，而 Server 端則為一 Application Server，其間的通訊方式為 Ajax。本系統提供 RSS 與 Comment 及檔案分享等功能，並加入群組的機制，使得其上的資源可以具有權限管理的功能，同時提供 API，使得其他人能從基本的架構上，加以延伸出更多的社群網路應用。詳細特點包括針對 URL/XPath 的 Post 應用與管理，可針對任何網頁上的段落進行社群討論與檔案分享；客製化的 RSS 產生器，目標包含 Web Pages 及 Comments，使用者可針對任何網頁自訂 RSS，並將此 RSS 分享給其他人，同時社群討論的結果也能讓社群訂閱；社群網站的群組管理與針對使用者與群組的權限控管，可使得社群的討論具有私密性從而排除不相關的人士與留言；使用者與群組的活動分享機制，使得同一群組的人能得知其他成員的活動狀況，加強成員之間的社群交流；Obsolete Post 管理以及原始網頁段落取回機制，能在網頁發生變動時，仍能瀏覽變動前的網頁內容與討論；發展 Social Web API 以供其他人擴充出更多的社群網路應用，從而使得本系統的研究能藉由更多人的投入而擴大應用的層面。

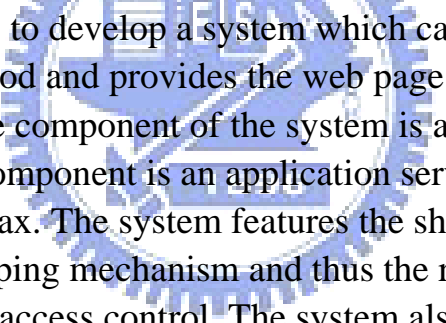
The Development and Research of Social Web API

Student : Kuei-chih Chang

Advisor : Dr. I-chen Wu

Degree Program of Computer Science
National Chiao Tung University

ABSTRACT



The goal of this research is to develop a system which can turn any web page into a social web by a simplest method and provides the web page most social web functionalities. The client-side component of the system is an Extension of Firefox browser and the server-side component is an application server. The communication method between the two is Ajax. The system features the sharing of RSS, comments and files. It also features grouping mechanism and thus the resource on it can be shared by the management of access control. The system also has its own API which can be used by third party to extend more social web applications on the base of fundamental structure.

Detailed features include the management of post which can be stucked to specific URL and XPath and thus people who use it can do social discussion and file sharing on any paragraph of any web page on earth. Customized RSS maker makes users generate their own customized RSS of any web page and share them with others. Also the contents of social discussion can be subscribed by users with their RSS reader. The user and group management provides the access control system and it can be applied on the share of social discussion. The activity sharing mechanism of user and group makes the members in a group aware of others' activities and thus enhances the relationship between members. Obsolete post handling as well as commented paragraph history retrieving can keep the web page contents and posts even after the paragraphs on web page are lost. The provided social web API makes other people capable of extending more social web applications by expanding the function the this social web system.

誌 謝

本人能夠完成本篇論文，首先必須感謝的是我的指導教授吳毅成老師。吳老師無論是課堂上的教學、對論文內容的討論，均給我細心的忠告，使後學受益良多、永誌難忘。除此之外，後學特別感謝各位口試委員對本論文的寶貴意見，使本論文能更臻完美，謹此致謝。

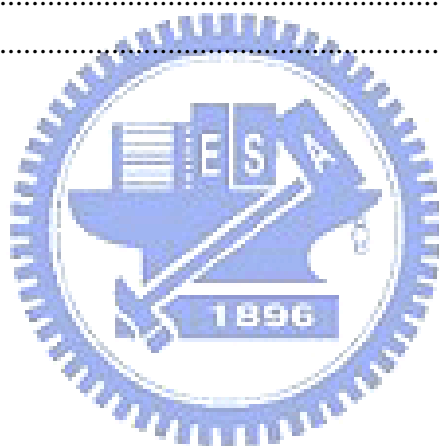
于碩士修業期間，陳隆彬學長給予許多的指導與協助，在系統開發期間，給我各種的建議，引領我跨過一個又一個的難關，才得以完成此系統，在此特別感謝他。同時，也要感謝其他學長們與學弟妹們，對於我的各種幫助。這些年的相逢與相識，將長存我心。

最後特別感謝我的太太朱秋玲女士以及我的父母，能在我課業繁忙的時候，做我最堅強的後盾，使我無後顧之憂，能專心在課業上而得以順利畢業。沒有他們在背後的支持，就沒有此篇論文的完成，謹以此篇論文獻給所有愛我以及我愛的人。

目 錄

中文提要	i
英文提要	ii
誌謝	iii
目錄	iv
表目錄	vi
圖目錄	vii
一、 簡介	1
1-1 背景	1
1-2 動機	3
1-3 論文架構	5
二、 背景知識介紹	6
2-1 Web Platform API	6
2-1-1 Information Services API	6
2-1-2 Search Services API	7
2-1-3 Storage Services API	8
2-1-4 Messaging Services API	8
2-1-5 Compute Services API	8
2-1-6 Web 2.0 Services API	9
2-2 Firefox Add-on	11
2-2-1 Add-ons的種類	11
2-2-2 Extensions的開發	12
2-2-3 XUL	14
2-3 網頁技術	20
2-3-1 DOM	20
2-3-2 XPath	21
2-3-3 AJAX	23
2-3-4 Web Service	24
2-3-5 RSS/Atom	26
2-3-6 Social Web	27
三、 系統設計與實作	30
3-1 RSS類元件系統架構	31
3-1-1 系統架構	31
3-1-2 Client端元件	33
3-1-3 Application Server端元件	38
3-1-4 RSS File Generator元件	43

3-2	Social Web類元件系統架構.....	47
3-2-1	系統架構.....	47
3-2-2	Client端元件.....	50
3-2-3	Application Server端元件.....	57
3-3	系統API.....	63
3-3-1	RSS類API.....	67
3-3-2	Social Web類API.....	70
3-3-3	使用者及群組管理類API.....	72
3-3-4	其他類API.....	73
3-4	應用範例.....	75
四、	相關文獻探討.....	77
4-1	Google Friend Connect.....	77
4-2	Fleck.....	79
4-3	本系統與其他Social Web系統之比較.....	80
五、	結論.....	82
	參考文獻.....	83



表目錄

表 1 RSS類元件使用軟體版本	31
表 2 RSS類元件之Overlay Unit之context menu之指令說明	36
表 3 RSS類元件之RssMaker_P2 Unit之UI之欄位說明	38
表 4 Social Web類元件使用軟體版本	47
表 5 Social Web類元件之Overlay Unit之UI圖示意義	53
表 6 Social Web類元件之Overlay Unit之context menu指令說明	53
表 7 Social Web類元件之Overlay Unit之Sidebar欄位說明	56
表 8 Social Web類元件之PostAdder Unit之欄位說明	57
表 9 Social Web Service與其他Social Web系統之比較	80



圖目錄

圖 1 Google Maps API之範例程式	7
圖 2 Google Maps API之範例畫面 資料來源: Google	7
圖 3 XUL範例 1 之圖形	16
圖 4 XUL範例 1 之代碼	17
圖 5 XUL範例 2 之圖形	17
圖 6 XUL範例 2 之代碼 1	18
圖 7 XUL範例 2 之代碼 2	18
圖 8 XUL範例 2 之目錄結構	18
圖 9 XUL範例 2 之第 1 個overlay.dtd檔案	19
圖 10 XUL範例 2 之第 2 個overlay.dtd檔案	19
圖 11 HTML DOM中的物件階層	21
圖 12 System Model Diagram of the Social Web Service.....	30
圖 13 RSS類元件系統架構圖 1	32
圖 14 RSS類元件系統架構圖 2	33
圖 15 RSS類元件之Client端元件系統架構圖	34
圖 16 SocialWeb Extension Options UI	35
圖 17 RSS類元件之Overlay Unit之menu	35
圖 18 RSS類元件之Overlay Unit之context menu	36
圖 19 RSS類元件之RssMaker Unit之UI	37
圖 20 RSS類元件之RssMaker_P2 Unit之UI	37
圖 21 RSS類元件之Application Server端元件系統架構圖	39
圖 22 RSS類元件之RSS Adding Unit之UI	40
圖 23 RSS類元件之RSS Sidebar Unit之UI	41
圖 24 RSS類元件之RSS Query Form Unit之UI	42
圖 25 RSS類元件之RSS Query Action Unit之返回XML	43
圖 26 RSS類元件之RSS Query Action Unit之返回HTML	43
圖 27 RSS類元件之RSS File Generator元件系統架構圖	44
圖 28 RSS類元件之Data Handling Unit之執行畫面	45
圖 29 RSS類元件之JREx Unit之執行畫面	46
圖 30 RSS類元件之XML File Generator所產生的XML內容	46
圖 31 Social Web類元件系統架構圖 1	48
圖 32 Social Web類元件系統架構圖 2	49
圖 33 Social Web類元件之Client端元件系統架構圖	50
圖 34 Social Web類元件之Overlay Unit之UI之一	51
圖 35 Social Web類元件之Overlay Unit之UI之二	52

圖 36 Social Web類元件之Overlay Unit之UI之三.....	52
圖 37 Social Web類元件之Overlay Unit之context menu	53
圖 38 Social Web類元件之Overlay Unit之Sidebar圖一.....	54
圖 39 Social Web類元件之Overlay Unit之Sidebar圖二.....	55
圖 40 Social Web類元件之PostAdder Unit之UI.....	56
圖 41 Social Web類元件之Application Server端元件系統架構圖.....	58
圖 42 Social Web類元件之Post Adding Unit之UI.....	59
圖 43 Social Web類元件之Post RSS Unit之UI.....	60
圖 44 Social Web類元件之Post Query Action Unit之返回HTML	60
圖 45 Social Web類元件之Post Query Action Unit之返回XML	61
圖 46 Social Web類元件之Post Graphviz Unit產生之dot檔.....	62
圖 47 Social Web類元件之Post Graphviz Unit之返回	62
圖 48 API Model Diagram of the Social Web Service	63
圖 49 Google Friend Connect例子.....	77



一、簡介

1-1 背景

在web 1.0的時代，網頁的內容提供者是單純的網站本身，其溝通是單向的，網友只能被動的接收訊息，網友無法表達其想法及意見，因此其內容缺乏與網友互動的能力。網友與網友之間也缺乏橫向的溝通，這對於網路內容的發展產生了一定的限制。漸漸地，web 2.0^{[1][2]}網站開始萌芽，終至蓬勃發展而成了網際網路的主流。所謂的web 2.0網站，就是網站內容由眾多網友所共同參與撰寫。著名的web 2.0網站有facebook^[3]，各式blog，digg.com，相簿、flickr，無名小站，youtube等，其共通的特點即是其網站內容是由網友所共同提供的，由於有了這些網站，網友因此有了互相交流的管道，網站內容不再是單向的而是成了雙向的交流，網際網路的發展由此邁向一個全新的紀元。

在web 2.0的時代，社群網路的模式是獨立自主的，網友不但扮演了內容供應者(content suppliers)的角色，在此同時，網友也扮演另一個角色，也就是消費者的角色(content consumers)。以facebook交友網站為例，不但網頁內容是由網友所編撰，甚至其網頁的internationalization方式也是web 2.0式的，也就是由網友一起來完成將英文翻譯成各國語言的工作。這與web 1.0的傳統模式形成了一個對比，傳統模式的內容供應者與消費者是截然不同的角色。在web 2.0的自主模式中，收入來源基本上是來自於廣告，但當會員與內容的水準相當高的時候，以訂閱為基礎的商業模式也是可能的一個方式。其他的商業模式像是虛擬商品(網頁個人化、頭像或角色、背景音樂、網頁外型、禮物等)也是另外可以創造利潤的地方。

目前Social Web之類的網站正值方興未艾之際。所謂的Social Web(社群網路)，是指人們透過World Wide Web來與其他人社交或互動的一種方式。由此，人們經由各種共同的興趣，而聚集在一起，今日在網際網路上已有許多

不同的方式，讓人們可以從事社交活動。大體上 Social Web 包含兩大類，即以社交為目的的社群網站，如 Facebook，MySpace 等；以及以興趣分享為目的的社群網站，如 Flickr，Photobucket 等。

建立線上社群，讓人們可分享他們自己的興趣與活動，眾多社群網站的興起，代表了網際網路是發展人際關係的一個重要管道。在 Internet 早期，支援社群網路的電腦通訊方法有許多，例如 Usenet，ARPANET，LISTSERV，BBS，EIES 等，但是 Internet 發展到現在，目前大部份的社群網路都已是 web based，並且提供了各種其他的方法，讓使用者之間可以互動。像近年來佔據了社群網站龍頭位置的 facebook 交友網站，其具備了眾多的服務如即時通訊，The Wall（即留言板），RSS Feed，Gift（虛擬禮物）等功能。然而，對於傳統的網站而言，如何在網站上增加網友之間互動的功能，雖不是一個十分龐大的工程，卻也絕非一簡單的工作。

在現今網路的重要走向是網站會提供各式各樣的 API，以使得各式各樣的網路應用能植基於這些 API 而被開發出來。目前，這些 API 可大致分為以下幾類，第一類是 HTML 代碼的方式，提供 API 的網站會依使用者需求產生一段 HTML 代碼，使用者將此代碼插入自己網頁內適當的地方即可應用此功能，例如 Google Maps 即為此例，這種方式的 Web Server 與 Social Server 非同一個；第二類是 Server Add-on 的方式，由網站提供 API，third party 寫成 Add-on 放於網站上，使用者即可在網站上點選使用這些程式，例如 Facebook 提供的 API 即為此例，這種方式的 Web Server 與 Social Server 是同一個；第三類是 Client Add-on 的方式，third party 依網站提供的 API 寫成 browser Add-on 後，使用者在自己的瀏覽器上安裝這些 Add-on，即能啟用 API 所提供的功能。

舉例來說，在 2007 年時 facebook 交友網站就開放了他們的網站 API，允許他人發展的 add-on 應用程式在其網站執行。其中有些 add-on 可畫出一個使用者其自己的社群網路，而另一些 add-on 則可讓使用者送禮物 (gift) - 包括虛擬的和實體的禮物 - 給他們的社群朋友。諸如此類各式的 add-on 成為了

其網站的一個重要賣點，也成為了社群網路的一個發展趨勢。

1-2 動機

有鑑於目前許多網站缺乏 RSS Feed 的功能，但 RSS 功能因為能帶給網友不小的便利，所以其功能實相當重要。從另一方面來看，目前即使有部份網站已提供了 RSS 功能，然而，很有可能其所提供的 RSS Feed 並不是使用者們所需要的。而對網站本身來說，欲開發 RSS 供網友使用，基本上就要耗費許多人力物力，而由於每個人的需求不盡相同，欲開發能滿足所有使用者需求的 RSS，更是個不可能的任務。因此，我的研究目的是在本社群網路 API 之中，提供一個平台，這個平台能夠讓使用者對世界上任何一個網站，讓使用者針對網頁上的條目與內容，製成自己所需的 RSS Feed，除了可讓自己訂閱之外，也能夠給他人訂閱，如此一來，任何一個網站，除了自己原本的 RSS Feed 之外，也能提供各式各樣五花八門的網友自訂 RSS，因此有兩大優點：其一，網站自身不需耗費人力物力來產生各種多樣化的 RSS，其二，使用者不會因網站未提供自己所需的 RSS 而感到不便，完全可依自己的需求來自訂其所需的 RSS。

在前面已提到，網友之間互相連絡的功能，對社群網站來說，是至關重要的，而在各種的通訊方式之中，留言版無疑是最常用的一種，然而，對於沒有留言版的網站，網友之間就相當難以互留訊息了，因此，在我的研究中的社群網路平台，亦能夠針對沒有建置留言版的網站，利用瀏覽器上的附加程式，在這些網站上加入外掛留言版，使得任何的網站，都可以帶有留言版的功能。

更進一步，絕大部份的現有留言版，都不帶有群組能力，因此，我希望藉由本研究的社群網路平台，提供對留言分群的能力，也就是說，網友的留言，能夠自由的設定，是只給某群組的人看，或是設定為所有造訪此網站的人都能看到。藉由此能力，網友可以保障個人的留言隱私，由此更能延伸出各式的群組應用。

而在另一方面，留言亦能結合 RSS Feed，來做更多樣化的應用，在這一方

面，本研究所開發的社群網路服務，亦能將網友的留言，不管是群組留言或是公開留言，都能以 RSS Feed 的方式，讓網友訂閱，所以網友能夠藉由這個功能，在不需實際連到網站的情況下，就能輕鬆看到眾多網站的最新留言標題與摘要，方便網友隨時掌握他所關心的議題。

因此總結來說，我的研究計畫，就是發展一個社群網路服務，其與 Google 的 Social Graph API 及 facebook 的 API 屬於同類，皆是屬於 Web 2.0 Services API 中的一種，其可以讓使用者在世界上任何一個網站上，都能建立自己的社群，提供建立社群網路所需的服務如 RSS Feed、留言、傳檔、群組等功能，並發展出 API，成為一個社群網路平台，讓任何人皆能在此平台上發展自己的社群網路服務。



1-3 論文架構

本論文各章節內容及說明茲分述如下：

- 第二章：說明論文相關的背景技術及知識，包括了 Web Platform API 的介紹、Firefox Add-on 的開發技術、XUL、DOM、AJAX、RSS/Atom 及 Social Web 等。
- 第三章：說明系統主要架構的設計與實作，包括 RSS 的 Client 端元件、App Server 端元件、RSS File Generator、Social Web 的 Client 端元件、App Server 端元件、及系統 API 等。
- 第四章：說明相關文獻探討，介紹目前 Social Web 的應用，包括 Google Friend Connect 與 Fleck，以及本研究與它們的比較。
- 第五章：本論文的結論。



二、背景知識介紹

2-1 Web Platform API

在目前 web 2.0 的發展，網站提供應用程式 API，是許多網站的趨勢，例如 Google Maps，其提供 API，可讓使用者在他們自己的網頁上放置 Google 地圖的時候，亦能做些設定，例如設定要顯示的控制項、決定地圖的中心點位置、在地圖上放置標點說明等等。facebook 交友網站亦提出 facebook API，讓第三方可以開發在 facebook 網站執行的程式，例如顯示使用者的人際關係連結圖、或是有趣的小遊戲等，內容五花八門，不但能吸引網友使用，同時也豐富了網站的應用層面，更進一步，有些具有交易功能的程式，借由第三方與網站拆賬，更能因此為網站帶來營收。由此看來，API 的發展與建立，對網際網路應用來說，確實為一不可忽視的發展潮流。

目前網際網路上的 Web Platform API 可大略被分為六大類^[4]，即 Information Services, Search Services, Storage Services, Messaging Services, Compute Services and Web 2.0 Services。這六大類的性質與簡介，會在下面分別介紹之。

2-1-1 Information Services API

Information Services 是提供大量的專門資訊的網路服務，此類服務包含有提供地理資料的服務，如 Google 的 Google Maps API 或如提供產品資訊的服務如 Amazon E-Commerce 或 Amazon Historical Pricing Services。

以 Google Maps API 為例，我們可利用此 API，在自己的網頁上放置一個 Google 地圖，並在地圖上顯示我們想要顯示的東西，像是顯示自家附近的地圖，或指定顯示成衛星圖或是道路圖等，並可在地圖上放置標記。以下是一個利用 Google Maps API 的範例程式。


```

<!--Google Map 顯示的位置，可自由決定大小-->
<div id="mymap" style="width: 500px; height: 500px"></div>

<!--以下為控制Google Maps的JavaScript-->
<script type="text/javascript">

//
var map = new GMap(document.getElementById("mymap"));

//設定要顯示的控制項
map.addControl(new GSmallMapControl()); //適合給小型地圖的控制項。
map.addControl(new GMapTypeControl()); //顯示地圖型態切换的控制項。

//決定你 Google 地圖的中心點位置和縮放大小
map.setCenter(new GLatLng(24.789618, 120.999599), 16); //設定 Google

//標記在 Google 地圖上的經緯度
var point = new GLatLng(24.789618, 120.999599); //設定座標值
var marker = new GMarker(point);
map.addOverlay(marker); //在地圖上放置標點

//在地圖上放置標點說明
var html = "交通大學";
map.openInfoWindowHtml (map.getCenter(), html);

//]]&gt;
&lt;/script&gt;
</pre>
</div>
<div data-bbox="258 352 508 369" data-label="Caption">
<p>圖 1 Google Maps API 之範例程式</p>
</div>
<div data-bbox="219 399 910 481" data-label="Text">
<p>利用其傳遞給 Google Maps API 的參數，如地理座標、顯示的控制項等，我們即可在網頁上顯示一個如下的地圖，並且此地圖並非靜態的而是動態的，也就是說造訪網頁的人可以利用滑鼠來拖拉或縮放此地圖。</p>
</div>
<div data-bbox="260 528 608 767" data-label="Image">
<img alt="A screenshot of a Google Maps browser window showing a map of National Central University (交通大學) in Taiwan. A red location pin is placed on the university's campus, and a white information window is open above it, displaying the name '交通大學'. The map interface includes standard navigation controls like zoom in/out and pan, and a top menu with 'Map', 'Satellite', and 'Hybrid' options. The bottom of the window shows the Google logo and copyright information for 2008."/>
</div>
<div data-bbox="258 778 508 795" data-label="Caption">
<p>圖 2 Google Maps API 之範例畫面</p>
</div>
<div data-bbox="258 800 391 816" data-label="Text">
<p>資料來源: Google</p>
</div>
<div data-bbox="138 849 406 865" data-label="Section-Header">
<h2>2-1-2 Search Services API</h2>
</div>
<div data-bbox="260 879 911 897" data-label="Text">
<p>另外一類的 API 是 Search Services API，顧名思義其目的就是提供搜尋</p>
</div>
<div data-bbox="514 923 530 939" data-label="Page-Footer">
<p>7</p>
</div>
```

服務，較著名的有 Google AJAX Search API 及 Alexa Search Platform 等。Google AJAX Search API 讓你可以你的網頁上，加入一個搜尋引擎，可以用來搜尋網頁、新聞及 blog 等。而 Alexa Search Platform 的 API 除了提供類似的功能外，程式設計師並可利用它海量的資料庫及 API 做一些更特殊化的搜尋，例如可利用其庫存網頁資料庫中相片中的 EXIF 資訊，找出某特定品牌或特定型號的相機所拍出的照片；又或是不只對網頁做檢索，也可以針對其庫存網頁資料庫中所儲存的網際網路上的壓縮檔，對壓縮檔中的檔案做檢索。這些都是利用 Alexa Search Platform 所能做到的功能。

2-1-3 Storage Services API

Storage Services API 是提供儲存服務的 API，像 Amazon S3 及 openemey 皆是此種 API 的代表。S3 是 Simple Storage Service 之意，其所提供的 API 類似一個基本的 hash table，讓你可以輕易的儲存及取回檔案及資訊，其 API 是以 SOAP 及 REST 協定來做通訊協定，而且，除了可以用 HTTP 來傳檔之外，還提供了 BitTorrent 的傳輸模式，這使得傳送大型檔案的時候可以更加的快速。除了以上所提到的 Amazon 及 openemey 這兩家公司之外，Google 也在開發他們的 GDrive，而微軟則是預備推出類似概念的 LiveDrive，而這兩者很有可能也會推出對應他們的儲存服務的 API。

2-1-4 Messaging Services API

第四類 API 是 Messaging Services API，其概念類似傳統的 middleware (中介軟體，用來在異質平台之間交換資訊的軟體)。此類的 API 因為不論在技術層面或是商業層面，都會比一般的 API 更加複雜，所以目前的數量並不是很多。目前較著名的是 Amazon 的 Simple Queue Service API，簡稱為 SQS。

2-1-5 Compute Services API

第五類是 Compute Services API，簡單來說即是網格運算的 API，讓你可以利用其所提供的強大計算能力做大量的運算，之後再將結果傳回給你。像 SUN 的 Grid，DataSynapse 的 GridServer 及 Platform 的 Symphony 皆屬於此類的 API。

2-1-6 Web 2.0 Services API

Web 2.0 Services API 即是指操作網友上傳的內容作為應用的 API，其可方便的查詢或是變更每位網友自行上載到網站上的內容。而在架構上來說，Web 2.0 Services API 可利用前述的其他五大類 API 做為基礎元件而開發出來。目前 del.icio.us, Flickr 及 Basecamp 等 Web 2.0 網站皆提供此類的 API。

以 del.icio.us 網站為例，其是一個書籤網站，讓網友們可以上傳他們自己的書籤到網站上，並與他人分享交流，是一個典型的 Web 2.0 網站。其所提供的 API 是使用 http-based 的介面，讓程式開發者可以開發出程式，讓網友們存取他們的美味書籤及標記。並且為了安全性的考量，所有的 Delicious APIs 皆透過 https 及 HTTP-Auth 協定來做為通訊的標準。其目前提供的 APIs 分為幾大類，一類是 Posts，即書籤類，讓你可以對書籤做新增、刪除、查詢等動作。另外一類是 Tags，可以讓你讀取所有的標記、對所有的書籤刪除某個標記，或是對所有的書籤重新命名某個標記等。

再來看 flickr 網站，其是一個相簿網站，可以讓使用者在這個網站上分享交流討論他們自己上傳的照片，所以自然也是屬於 web 2.0 網站之一。他提供的 API 所使用的協定包含 REST, XML-RPC 及 SOAP，其 API 最主要的功能就是讓第三方開發者能夠開發程式讓網友不必透過網頁也能上傳照片到 flickr 網站上。除了這個功能之外，另外還有一些功能類別，包含基本的 authentication、編寫存取 blog、編寫存取你的好友名單、設定你的最愛攝影師名單及設定群組等，功能詳盡而繁多。

另一個提供 API 的著名 Web 2.0 網站是 Basecamp.com，它是一個 project

management 網站，可以讓一個群組有組織有計畫的共同完成一件工作，所以它又稱自己為 Business 2.0 網站。其所提供的 Basecamp API 所使用的協定是 HTTP，而資訊交換則是以 XML 的格式來傳輸。其 API 功能種類繁多，包含了訊息傳遞、事項加註、待辦事項管理、進度管理、成員通訊錄管理等等。

由以上所述看來，目前若想要開發功能強大且易用的 browser-based 應用程式或是桌面應用程式，只要能善加利用這些 API，開發都是非常的容易且方便。而且因為底層的元件都已俱備且相當完善，開發者將更加能把心力專注在程式的可用性、易用性及內容的強化上。

在本研究中即是以發展一個應用於社群網路的 web 2.0 API 為主題之研究。



2-2 Firefox Add-on

Firefox 是近年來一個快速崛起的 web browser，憑藉著快速的效能及功能眾多的 add-on，目前在全世界已有兩成以上的佔有率，是僅次於 Microsoft Internet Explorer 之外，最受歡迎的 web browser。

而Firefox的Add-ons^[5]，指的是針對Mozilla基金會的計畫（及其相容的變形，例如可攜版的Firefox），所開發的可安裝的加強元件。Add-ons讓使用者可以增加應用程式的功能，或根據喜好使用theme，或是能對應新形態的內容。

2-2-1 Add-ons 的種類

1. Extensions

Extension 可以被用來改變應用程式原始的功能或是新增一個全新的功能。Extension 在 Firefox 中特別受到歡迎，因為 Mozilla 的開發者傾向於將 Firefox browser 定位在只包含最基本的功能，以儘可能的縮小程序體積並減少 bugs，同時保持一高度的延展性，所以個別的使用者可以根據個自的喜好新增他們需要的功能。

Extension 使用到的 technologies:

- CSS (Cascading Style Sheets)
- DOM (Document Object Model)^[6] — 用來即時改變XUL^[7]^[8]或是編輯目前載入的HTML
- JavaScript — Mozilla browsers 的主要語言
- XPCOM (Cross-Platform Component Object Model) - 跨平台元物件模型
- XPConnect
- XPI (Cross-Platform Installer) - 跨平台的安裝包
- XUL (XML User Interface Language) — 用來定義使用者介面

(UI, User Interface) 並和使用者互動

用途:

■ 新增功能

Extensions 通常用來對應用程式新增功能,一個 Extension 可新增的功能可能包含: RSS 閱讀器、bookmarks 整理工具 (organizers)、工具列、針對特定網站的客戶端程式、FTP、e-mail、滑鼠手勢、代理伺服器切換、或是開發者工具。許多 Firefox Extensions 是實做之前 Mozilla 套件 (Mozilla Suite) 的部份功能,例如 ChatZilla IRC 客戶端及 calendar 等。

■ 改變使用者觀看網頁的方式

許多 Extension 在網頁被 render 時可改變網頁的內容,例如, Adblock Extension 可使 browser 不要載入廣告圖片, Greasemonkey 讓使用者安裝一些 scripts,而這些 scripts 可以即時的變更某些網站的網頁。



2. Plugins

一般的 plugins 包含 Acrobat Reader, Flash Player, Java, Quicktime, RealPlayer, Shockwave, 及微軟的 Windows Media Player 等。

2-2-2 Extensions 的開發

Firefox 的 Extension 是以 zip 檔案的方式來包裝與散布,其附檔名為 xpi (發音為 zippy),其 xpi 檔的內容的 layout 如下所示:

```
Extension.xpi
  /install.rdf
  /components/*
  /components/cmdline.js
```

```
/defaults/  
/defaults/preferences/*.js  
/plugins/*  
/chrome.manifest  
/chrome/icons/default/*  
/chrome/  
/chrome/content/
```

開發 Extension 第一步是建目錄~/Extension/my_Extension，然後在 my_Extension 之內，建立 chrome/content 並建立兩個檔案：chrome.manifest & install.rdf

Install.rdf 是使用者在安裝 Extension 時的設定檔，開發人員需在其內指定以下內容：

- `<em:id>`: 此 Extension 的 id。
- `<em:type>`: 若值為 2，代表此為一 Extension；若值為 4，代表此為一佈景主題 (theme)。
- `<em:minVersion>`: 此 Extension 所能對應的 Firefox 的最小版本。
- `<em:maxVersion>`: 此 extension 所能對應的 Firefox 的最大版本。

構成 Firefox 的使用者介面的是 XUL & JavaScript。XUL，即 XML User Interface Language，它是一個 XML 文法，其提供了使用者介面小元件 (user interface widgets)，例如按鈕、目錄、工具列及樹等等。而使用者所產生的動作皆以 JavaScript 的功能所控制。

當我們想要擴展 browser 的介面時，我們可以靠著增加或修改 widget 的方式，來變更 browser UI。舉例來說，
\$FIREFOX_INSTALL_DIR/chrome/browser.jar 之中，包含了
content/browser/browser.xul 這個檔案，這個檔案即是 Firefox 瀏覽器的介面的描述檔，其中的`<statusbar id="status-bar">`節點，即描述了 browser

的狀態列的組成。

XUL 覆蓋 (XUL Overlay) 是一種在執行時期 (run time) 將其他的 UI widgets 附加到 XUL 文件上的技術。一個 XUL Overlay 是一個 .xul 檔案，其指定了 XUL 片段，用來插入到主文件中的合併點 (merge points)。這些片段設定讓 widgets 可以被插入、刪除及修改。

Chrome URI 代表的是使用者介面的 URI，其與 Internet 上的 URI 有異曲同工之妙。一個 chrome URI 的例子如下：

```
chrome://browser/content/browser.xul
```

1 2 3 4

1. URI scheme (chrome): 此部份告訴 Firefox 的 networking library 這是一個 Chrome URI。
2. package name (browser): 使用者介面元件的包裹。
3. 資料型態: 有三種型態: content, locale 及 skin。在此的 content 代表了應用程式 UI 的結構與行為。
4. 檔案的路徑

在 chrome.manifest 檔案中，如下的程式碼：“content sample chrome/content/” 定義了一名為 sample 的 package，其 content 檔案位於 chrome/content。而此程式碼：“overlay chrome://browser/content/browser.xul chrome://sample/content/sample.xul” 則告訴 Firefox，當 browser.xul 載入時，將 sample.xul 合併到 browser.xul 之中。

在本研究中社群網路 API 之開發，將會以 Firefox Extension 為基礎，應用 Firefox 與 XUL 檔案之間的階層架構，以及 Firefox Extension JavaScript 語言，做為開發的 framework 之一。

XUL，即 XML User Interface Language，是一個由 Mozilla 計畫所開發出的 XML 使用者介面標記語言，它被使用在 Mozilla 的跨平台應用程式之中，例如 Firefox 及 Flock。目前只有 Gecko 排版引擎 (layout engine) 提供了 XUL 的完整實作。

XUL 必須依靠在許多現存的網頁標準及技術，包括 CSS, JavaScript, 及 DOM。如此的關係使得具有網頁程式設計及網頁設計的背景的人學習起 XUL 時變得相對的容易。XUL 並無正式的規格，並且無法在非 Gecko 的實作上動作，雖然如此，它使用的 Gecko 實作是開放原始碼的，而且有三種授權方式：GPL, LGL 及 MPL。Mozilla 提供了一個實驗性質的 XUL Runner 版本，以讓開發者在 Mozilla 應用程式框架 (Mozilla application framework) 及 XUL 上建構他們的應用程式。XUL 的主要優點，即是提供了一個簡單並且可攜的，對於一般視窗部件 (widgets) 的定義，這大幅的減輕了軟體開發的工作，就有如 4GL 工具所提供的一樣。

程式設計師基本上用三組分散式的元件來定義 XUL 介面：

1. content: 即 XUL 文件，其中的元素定義了使用者介面的 layout。
2. skin: 即 CSS 和影像檔，skin 定義了一個應用程式的外觀。
3. locale: locale 檔案包含了使用者介面的字串，由此可以方便的做到軟體本土化 (software localization)。

XUL 定義了許多的元素，大致上可分為以下幾類：

- 頂層元素 (Top-level elements)
如 window, page, dialog, wizard 等
- 視窗部件 (Widgets)
如 label, button, text box, list box, combo box, radio button, check box, tree, menu, toolbar, group box, tab box, color picker, spacer, splitter 等
- 盒子模型 (Box model)

- 事件與手稿程式 (Events and scripts)
如 script, command, key, broadcaster, observer 等
- 資料來源 (Data source)
如 template, rule 等
- 其他
如 overlay, iframe, browser, editor 等

XUL 文件之中，可以使用其他 XML 應用程式的元素，像是 XHTML, SVG 及 MathML

Mozilla 在 Gecko 1.9 的開發過程中新增了一些常見的視窗部件 (widgets)，如 <scale/>, <spinbox/>, 日期與時間選取器 (time and date picker) 等。

下圖是 XUL 的一個簡單範例：

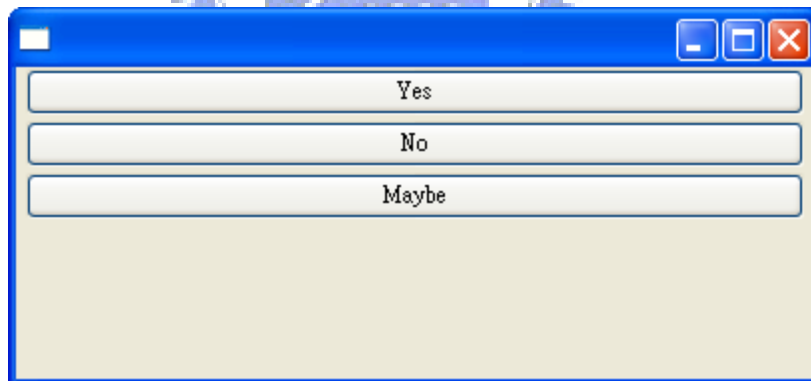


圖 3 XUL 範例 1 之圖形

這是垂直並排的三個按鈕，其 XUL 檔案內容如下所示：

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>

<window id="vbox example" title="Example"
xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">

<vbox>
  <button id="yes" label="Yes"/>
  <button id="no" label="No"/>
  <button id="maybe" label="Maybe"/>
</vbox>

</window>
```

圖 4 XUL 範例 1 之代碼

Xul 檔案的 window tag 標示了此視窗的 title 及 name space，而檔案的主體則是一 vbox 元素，代表一 vertical box，其內包含三個 button，並指定了三個 button 上的顯示字串。

底下是 XUL 的第二個例子：

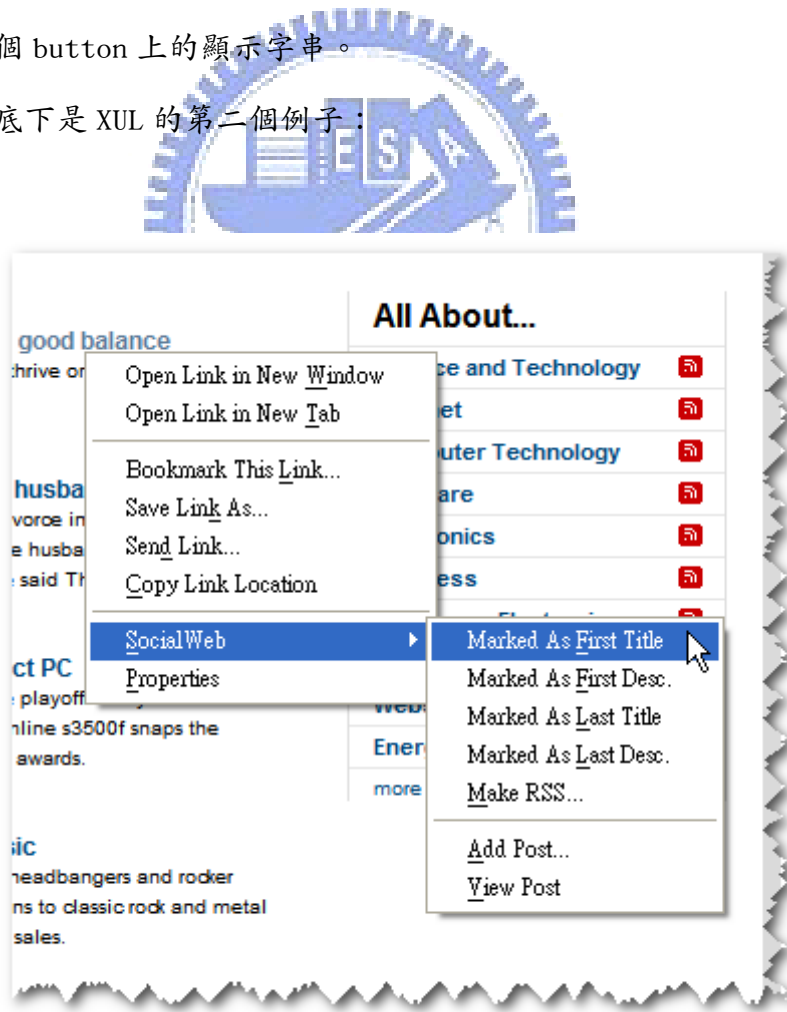


圖 5 XUL 範例 2 之圖形

在上圖中，若我們欲使此 menu 能依照使用者所使用的語言顯示其上的字串，我們可在 xul 檔案中做如下的處理：

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://socialweb/skin/overlay.css" type="text/css"?>
<!DOCTYPE overlay SYSTEM "chrome://socialweb/locale/overlay.dtd">
```

圖 6 XUL 範例 2 之代碼 1

檔頭部份，我們指定一個 dtd 檔做為儲存各國語言字串的檔案。

```
<menuitem id="miRssmaker-firstTitle"
  label="&browser.cxtmenu.miRssmaker-firstTitle;"
  accesskey="F"
  oncommand="setExpatState('FT');" />
```

圖 7 XUL 範例 2 之代碼 2

而在 menuitem 元素的 label 屬性，則以 "&" 開頭，代表要依各國語言做不同的顯示。

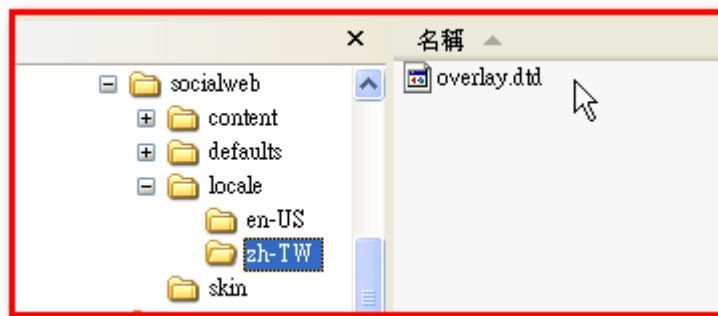


圖 8 XUL 範例 2 之目錄結構

我們在 locale 目錄下放置兩個子目錄：en-US 及 zh-TW，分別代表美國英文及台灣中文，其下各有一個 overlay.dtd 檔，內容如下：

```
<!ENTITY browser.cxtmenu.miRssmaker-firstTitle "Marked As First Title">  
<!ENTITY browser.cxtmenu.miRssmaker-firstDesc "Marked As First Desc.">  
<!ENTITY browser.cxtmenu.miRssmaker-lastTitle "Marked As Last Title">  
<!ENTITY browser.cxtmenu.miRssmaker-lastDesc "Marked As Last Desc.">
```

圖 9 XUL 範例 2 之第 1 個 overlay.dtd 檔案

```
<!ENTITY browser.cxtmenu.miRssmaker-firstTitle "設為第一個標題">  
<!ENTITY browser.cxtmenu.miRssmaker-firstDesc "設為第一個描述">  
<!ENTITY browser.cxtmenu.miRssmaker-lastTitle "設為最後一個標題">  
<!ENTITY browser.cxtmenu.miRssmaker-lastDesc "設為最後一個描述">
```

圖 10 XUL 範例 2 之第 2 個 overlay.dtd 檔案

如此一來，此 xul 檔案便能依照使用者使用語言的不同而達到軟體的 localization。



2-3 網頁技術

2-3-1 DOM

DOM 是一個平台無關並且是語言無關(platform-and language-independent)的標準物件模型(standard object model)，其是用來表示 HTML, XML 等等類似的格式。Web browser 在展現一個 HTML 文件的時候，並不需要使用 DOM，但若要讓 JavaScript scripts 能夠動態的去查看或是修改網頁時，這時就需要用到 DOM 了。換句話說，DOM 是讓 JavaScript 看見 HTML 網頁與瀏覽器狀態的一個途徑。

因為 DOM 支援任何方向的導覽(navigation)，例如往 parent 方向或是往 sibling 方向，並且允許任意的修改，因此瀏覽器必須暫存(buffer)到目前為止已被讀取的文件，因此，DOM 最適合用在文件會被重覆存取或是隨機存取的應用程式上。若應用程式傾向於使用循序存取及一次存取(one-pass)，則 SAX model 將會更加的快速且使用較少的記憶體。除此之外，非萃取的 XML 剖析模型(non-extractive XML parsing models)，例如 VTD-XML，則是另一個選項，可以在使用較少記憶體的情況下操作。

W3C 在 1990 年代中期開始了 DOM 的發展。雖然 W3C 從未發表 DOM 0 的規格，但是它的模型有一部份卻被記錄下來並且被包含在 HTML 4 的規格之中。在 1998 年十月，W3C 發表了 DOM 的第一份規格，亦即 DOM 1。而 DOM 2 則是在 2000 年十一月發表，同時發表的還有 style sheet 物件模型及 style information 的操作方式。DOM 3 發表於 2004 年四月，它也是 DOM 規格的現行版本。

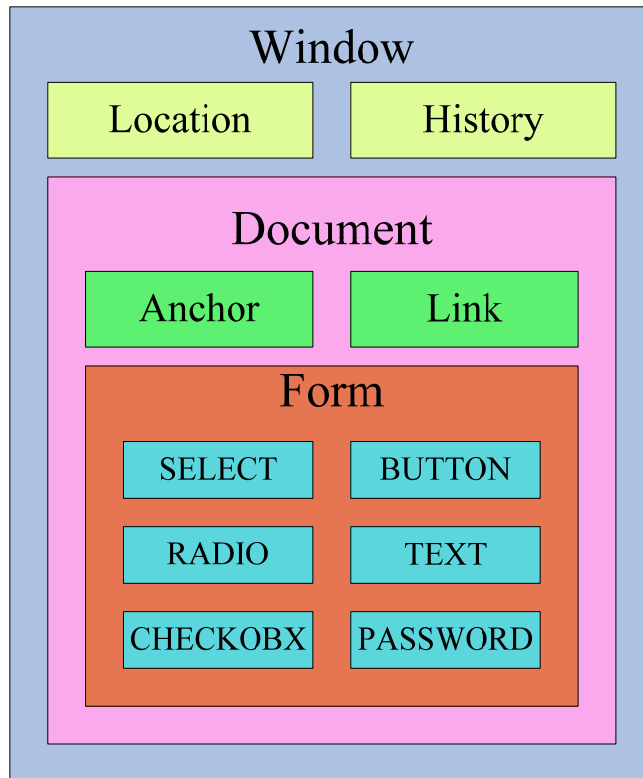


圖 11 HTML DOM 中的物件階層

在早期，每個 Web browser 只支援它自身的 DOM，因此造成了很多相容性問題。為了使程式碼能夠相容於多個瀏覽器 (cross-browser compatible)，很大一部份的動態 HTML 程式碼必須針對某一特定瀏覽器而重寫，這對開發網頁形成了重大的阻礙。因此，一個共同的 DOM 能夠大幅的簡化複雜網頁應用程式的開發。

這種情況後來有了改變。在 2005 年時，W3C DOM 的大部份都已被使用 ECMAScript-enabled 的網頁瀏覽器所支援，這些 web browser 包含了 Microsoft Internet Explorer version 5 and version 6，使用 Gecko 網頁引擎的 browsers 如 Mozilla, Firefox 及 Camino，及其他 browsers 如 Konqueror, Opera 和 Safari。至此，W3C DOM 成為網頁應用程式開發人員所依賴的主要標準，因為它擁有最大的 web browser 相容性。

XPath (XML Path Language)[⁹][¹⁰] 的主要用途是在一個XML文件中選擇節點，為了這個目的，XML文件以樹的模型展現，XPath讓節點能夠以分層的導覽路徑(hierarchical navigation path)經由文件樹而被選擇到。

XPath 2.0[¹¹]是目前World Wide Web Consortium (W3C) 所定義的XPath語言的最新版本，W3C在 2007 年一月時建議使用此版本。

XPath 2.0 是 XSLT 2.0 的子語言(sublanguage)，也是 XQuery 1.0 的子集，這三種語言使用相同的資料模型、型態系統及功能函式庫，它們同時也是一起開發的並且在同一天發表。

1. 資料模型 (Data Model)

XPath 2.0 中的每一個值是一序列的項目，項目可以是節點或值，序列不能是巢狀的 (nested)。

節點共有七種，對應到 XML 文法的不同構造：元素 (elements)、屬性 (attributes)、文字節點 (text nodes)、註解 (comments)、處理指令 (processing instructions)、命名空間節點 (namespace nodes) 及文件節點 (document nodes)。

節點可以是有型態的或無型態的 (typed or untyped)。節點被以 XML schema 驗證之後的結果最後將成為節點的型態。

2. 路徑表示式 (Path Expressions)

非正式的說，一個路徑表示式就是一序列的步驟 (steps)，步驟之間以 “/” 運算子來做分隔，例如：a/b/c (其是 child::a/child::b/child::c 的簡寫)。然而，正式些來說，“/” 就是一個二元運算子，其將 “/” 右邊的表示式應用到 “/” 左邊的表示式所選到的每個項目，所以在這個例子中，“a” 表示式選擇了命名為 “a” 的所有內文節點；而表示式 child::b 接著被應用到每一個這些節點之上，即選擇 a 元素的所有被命名為 b 的子元素；再來 child::c

表示式又被應用到這個序列中的每人節點，即選擇 b 元素的所有被命名為 c 的子元素。

“/” 運算子在 XPath 2.0 之中被通用化了，其允許任何種類的表示式做為其運算元，舉例來說，一個 function call 也能放在 “/” 運算子的右手邊。此運算子的型態規則要求第一個運算元必須是一序列的節點，而右手邊的運算元則可傳回節點或是值（但不可是兩者的混合）。若結果包含節點，則重覆的節點會被消除，並且以原始 XML tree 中節點的相對位置來做為傳回的順序。

2-3-3 AJAX

AJAX^{[12][13]}即 Async JavaScript And XML，是一種web開發技術，用來實現互動式的web應用程式。目的是讓web pages在使用上的回應更加快速，而方法是藉由在背景與server交換小量的資料，所以每當使用者在web page上做了某些動作時，整個web page不需要reload即能反映使用者的輸入。

AJAX 使用到的技術包含 XHTML (或 HTML), CSS, DOM, XMLHttpRequest 物件和 XML (或 HTML, 純文字...)。傳統上 web application model 是如此工作的：大部份的介面上的使用者的動作會觸發一個 HTTP request 回到 web server 上，server 做了一些處理之後 - 像是取得資料、處理數值、與其他系統交換資料等 - 然後傳回一個 HTML 頁面到 client 端，這很適合 Web 的原始使用方式 - 做為一個超文字媒體 (hypertext medium)，但如此的方式對一個應用軟體來說卻是不適合的。

這個方法有許多技術上的理由，但是它卻不能提供很好的使用者經驗，當 server 做它的事時，使用者只能乾等，而且在一個 task 的每個步驟，使用者也都要等待。

很明顯的，如果我們以應用程式為目的來重新設計 Web，我們不會讓使用者在那邊等。一旦一個使用者介面被載入後，為什麼每當應用程式要從 server 抓取東西時，整個動作都必須停止下來？事實上，使用者根本不需要每次都看

到應用程式去 server 抓取資料。

AJAX 應用程式消除了 Web 上的互動方式的天性：開始-停止-開始-停止，其消除的方式是利用一個中間物 - Ajax engine — 置於使用者與 server 之間。乍看之下，在應用程式上加了一層會讓它的反應變慢，事實上卻剛好相反。

在一個 session 的初始，browser 不再是載入一個 webpage，取而代之的，是載入一個 Ajax engine。此 Ajax engine 是以 JavaScript 所寫成並被塞在一個隱藏的框架 (frame) 之中，這個 engine 負責兩件工作，第一是呈現 user 所看見的介面，第二是依照使用者的動作來與 server 溝通。Ajax engine 允許 user 與應用程式的互動可以是非同步的 — 與對 server 的通訊互相獨立。所以 user 永遠不需看著一個空白的 browser window 與一個沙漏游標來等待 server 傳回資料。

AJAX 的優點之一是互動性，例如可快速的讓使用者刪除或修改 db 的 records，或是傳回簡單的查詢等。其另一個優點是可攜性，因為目前主流的瀏覽器皆支援 AJAX 技術，而且，因為可攜性的原因，AJAX 已在某些方面取代了 Java applets 的應用。

AJAX 也有缺點，缺點之一是可用性，例如可能使得 browser 的 back button 無法如使用者預期的作用，而且，使用者可能也無法 bookmark 網頁的某個特定的 state。另一個缺點是回應時間 (response-time)，網路的延遲可能使得使用者做了某動作後網頁上卻沒有回應，所以最好在 AJAX 動作時能夠在網頁上予以提示，例如顯示一” Loading…” 字樣。

2-3-4 Web Service

Web Service^{[14][15]}在W3C的定義之中，指的是”設計用來支援機器與機器之間透過網路來交換資料的軟體系統”，Web service通常只是可透過網路存取、並且由遠端的機器提供服務的Web APIs。W3C的Web service的定義包含了許多不同的系統，不過一般來說，Web service這個詞是的是client與server

之間，透過Web上使用的HTTP協定，來做為溝通的方式。這樣的服務大抵分為兩大陣營：Big Web Services及RESTful Web Services。

Big Web Services 使用遵守 SOAP 標準的 XML 訊息，並且此種 Web Services 已在傳統企業之中獲得了廣泛的使用。在此種系統之中，通常服務會提供一個機器可判讀的操作描述，而此描述以 Web Services Description Language (WSDL) 撰寫。WSDL 並非是 SOAP 標準的必要條件，然而，在許多 Java 和 .NET 的 SOAP 架構之中，要能自動產生 client-side 的 code，WSDL 卻是必要的。所以，有些產業組織，像是 WS-I，將 SOAP 和 WSDL 都一同列在 Web service 的定義之中。

近年來，RESTful Web services 重新受到歡迎，特別是在 Internet 公司方面。此種 Web services 同樣符合 W3C 的定義，而且較之 SOAP-based 的服務，他們能夠與 HTTP 協定整合的更好，這些服務也不需要使用 XML 做訊息交換，而且也沒有 WSDL 的 service-API 定義。

為了改善 Web Services 的交互操作性，WS-I 推出了 profile 來定義它們。一個 profile 指的是一組核心規格 (SOAP, WSDL, ...) 的特定版本 (例如 SOAP 1.1, UDDI 2, ...) 加上一些額外的要求以限制核心規格的使用。WS-I 也發表了用例及測試工具，以協助部署相容於 profile 的 Web services。

為了擴增 Web Services 的能力，有些規格已開發出來或正在開發之中，這些規格通常被稱為 WS-*，下列是一部份的 WS-* 規格。

- WS-安全性 (WS-Security):

為了保證訊息交換的安全性，WS-Security 定義了如何在 SOAP 之中使用 XML Encryption 及 XML Signature 的方法，做為 HTTPS 之外的另一個方法或其延伸。

- WS-可靠性 (WS-Reliability):

一個稱為 OASIS 的標準協定，可在 Web services 之間獲得可靠的訊息傳遞。

- WS-交易 (WS-Transaction):

定義處理交易的方法

這些額外規格的一部份是來自於 W3C。另一方面，OASIS 將許多的 Web service 延伸加以標準化，包含 Web Services Resource Framework 及 WSDM。

Web services 是一組工具，我們有一些方法來使用它，三種最常見的使用方式是遠端程序呼叫 (RPC)、服務導向架構 (SOA) 及代表性狀態傳遞 (REST)。

- 遠端程序呼叫 (Remote procedure calls, RPC):

代表了分散式的 function call 介面，這種方法是許多開發人員所熟悉的，基本上，RPC Web services 的基本單位即是指 WSDL 的操作。

- 服務導向架構 (Service-oriented architecture, SOA):

在這種方式之下，通訊的基本單位是一個訊息，而非一個操作，所以也被稱為“訊息導向服務”(message-oriented services)。

- 代表性狀態傳遞 (Representational state transfer, REST):

最後一種是 REST，此種架構使用 HTTP 或類似的協定，而將其介面限制在眾所週知且標準的操作之上 (如 HTTP 的 GET, POST, PUT, DELETE)，此方式注重在狀態資源的互動之上，而非訊息或是操作。

2-3-5 RSS/Atom

RSS 與 Atom 一樣，是一種特殊的 web feed 的 format，而所謂的 web feed，即是讓 Internet 的使用者，對周期性增加或變更內容的 web sites，執行訂閱的功能。RSS 也是一個開放性的標準，其以 XML 為基礎，一般最常使用在 weblogs、新聞網站及 web mail 等方面。要使用 RSS，網站擁有者必須自製或取得一個特殊的軟體(例如內容管理系統,content management system)，以機器可判讀的 XML 格式，將新文章的標題條列出來，並需能產生一到兩行的文章摘要及完整文章的連結。

目前 RSS 主要的應用，即是在 weblog 社群，可以用來分享新進文章的標題；另外在行銷方面，可用來告知新進產品的項目；在協同開發方面，可用在 bug 報告上；而其他任何需要週期性的更新或通報的網路應用，都可利用到 RSS。舉例來說，例如一個新聞網站，可以把社會新聞的標題與內容摘要，製成 RSS，也可以把影劇新聞的標題，同樣的製成另一個 RSS。

在 client side 的 RSS 閱讀器，主要有三種方式，一種是獨立的閱讀器程式，另一種則是附加在 browser 上，目前主流的瀏覽器，如 Safari, Opera, Firefox 和 Internet Explorer, 皆已內建 RSS 閱讀器。而第三種則是 web-based 的方式，以 browser 來閱讀。

所以這對使用者來說是非常方便的，因為使用者不須親自造訪這些網站，只須要利用 RSS Reader，就能輕鬆的掌握到各種訊息的摘要精華，例如當使用者經由 RSS 看到一個社會新聞的標題，而想要深入了解其內容時，只要點下該標題，就能連到該則新聞的網頁，來詳細閱讀其內容。在這資訊充斥的時代，RSS 的發明讓使用者能輕易且有效率的獲取他所想知道的知識。

在本研究中，會以 RSS 標準，將一部份社群網路產生的資訊，或是某個網頁上的定點資訊，傳回給使用本研究的服務的使用者。

2-3-6 Social Web

Social Web (社群網路)^[16] 目前被用來描述人們如何透過 World Wide Web 來與其他人社交或互動的一種方式。由此，人們經由各式各樣的共同興趣，而能聚集在一起。今日在網際網路上已有許多不同的方式，讓人們可以從事社交活動。第一種的社交方式以”人們焦點”(people focus) 網站為代表，例如 Bebo, Facebook 及 MySpace 皆是這類的網站。這類的網站將人提昇為社交互動的焦點，因此，網友的 profile 是由每位使用者自行建立的。在很多方面，profile 的作用就好像護照一樣。

第二種的社交方式則以一種”興趣焦點”(hobby focus) 網站為代表。

例如，若一個人對攝影有興趣，並且想與志同道合的網友分享心得與照片，那麼，他便可以參與這類的網站，像是 Flickr, Kodak Gallery 及 Photobucket。

在 Social Web 之中，也有兩種方法人們可以與他人進行社交活動，最普遍的方式，就是人們永遠保持距離，只在 World Wide Web 上進行活動，在這種社交方式之下，自然永遠沒有面對面的個人接觸。在 Flickr 上，大部份的社交方式，就是分享照片及評論他人的照片，然而，Flickr 的會員若來自相近的地理區域，那麼他們很可能會相約一同出去外拍，而這即是透過 World Wide Web 進行社交活動的第二種形式：會有真正實體接觸的社交方式。後者的基本例子，在歷史上是由大學院校的校園內外所興起的。Facebook 的起源即是哈佛大學內的學生與教職員名冊 (facebook)。

Social Web 也可以被指稱為兩種相異但相關的概念，第一種是 web 2.0 技術中，若特別注重在社交的互動與社群，則亦可稱為 Social Web，第二種是類似於 World Wide Web 的未來網路的提案。

社群網路的一個重要基礎是社群成員之間互相溝通的能力，以目前 web 2.0 的一個重要應用-blog-來說，blog 上必定設有留言板，讓 blog 的讀者能夠與 blog 作者或其他網友相互交流，另外一個例子是，一部份的新聞網站，如聯合新聞網，會在每則新聞的下方，設有讀者留言版，讓讀者發表意見，甚或幫網站勘誤等，這其實正是 web 2.0 網站的精神-網友不但是網頁內容的消費者(Content Consumers)，同是更是網頁內容的供應者(Content Suppliers)，有時甚至網友的留言比網頁原始內容更加精彩豐富，這些例子充份說明了留言板在社群網路中實在扮演了一個重要的角色。然而，對很多網站來說，或是缺乏經費，或是缺乏技術，更可能的是，對於留言板的重要性認識不足，因此在世界上所有的網站中，有留言版的網站其實不多，這對想要在某個網站上進行討論的網友是一個缺憾，而從網站的角度來看，若網友能在其上進行討論，不但能豐富網站的內容，而且，當網友在其上留言，其可能會回來看看別人對其留言的看法，從而能直接提升網站的回訪率。

而另一方面，即使是目前有留言版的網站，功能亦很可能不符使用者所需，例如其一是群組功能，即是將網友分成一個一個的群組，群組成員之間的留言可設定不讓群組以外的人所看見，甚或不被網站方面所看見，這樣的應用例子有很多，例如：某棋友社可在某圍棋網站對某棋譜進行討論，但希望他們之間的討論只能被此棋友社的成員所看見，以免戰術為他人所悉而遭到破解。又或是一個攝影社，其成員可在某攝影網站，針對某張照片進行教學與討論，但不希望其內容為他人所得見，以維持其智慧財產權並從而使社團能夠以此向社員收費。這些都是群組討論相當實際的應用。

而另一個很有用處的功能，就是不止能針對網頁留言，更進一步的，能針對網頁裡的某段落留言，這對於一些內容較多的網頁是很有用處的，舉例來說，網友可針對網頁的某段落進行討論，從而使討論聚焦而不會因網站內容太長而失焦。

此外，現在的留言版普遍欠缺 RSS Feed 的能力，若有此功能，網友不需實際造訪網站，即能輕易的看到新舊留言的標題，若想得知留言的詳細內容，只要輕輕一點，即可連到網頁看到訊息全文。這對關心網頁留言最新內容的網友來說，絕對是一個很方便的渠道。

三、系統設計與實作

在此章節將介紹社群網路 API 系統的設計與實作。本系統分為兩大類元件：第一類為 RSS 類元件，含三個子元件，分別是 Client 端元件、App Server 端元件及 RSS File Generator 元件；第二類為 Social Web 類元件，含兩個子元件，分別是 Client 端元件及 App Server 端元件。系統概觀如下圖。

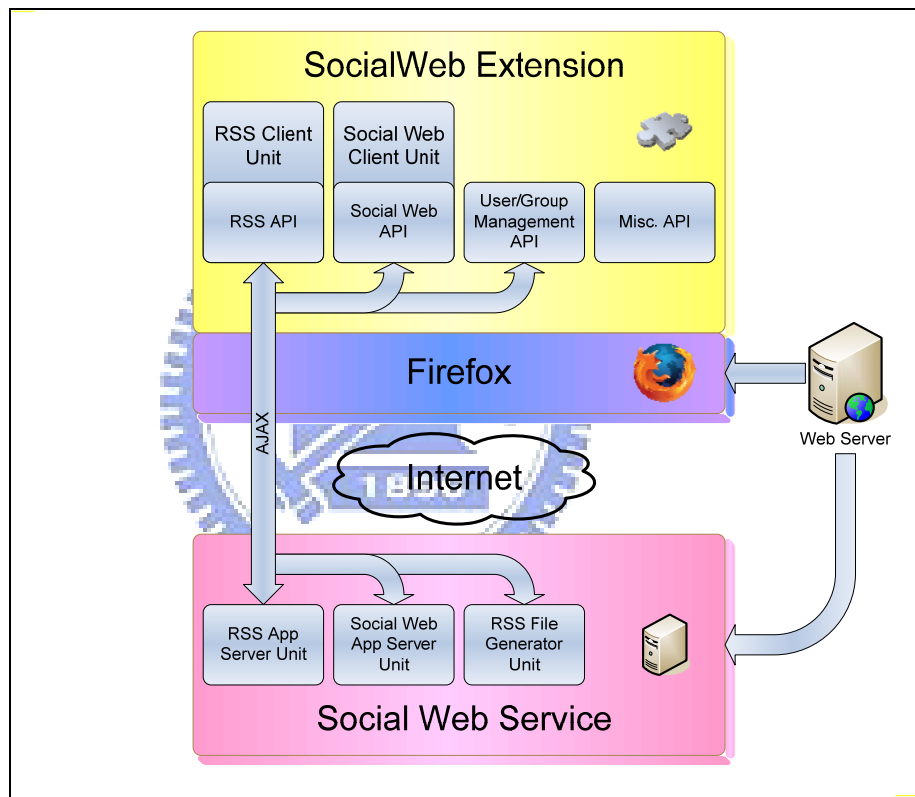


圖 12 System Model Diagram of the Social Web Service

本系統的兩大類元件，皆各分為 Client 與 Application Server 兩部份。在 Client 的部份，是架構在一個 Firefox 的 Extension 上，稱為 SocialWeb Extension，其內有兩個元件，四種 API，底層以 XUL 及 JavaScript 所開發。在 App Server 的部份，包含有三個元件，其中的 RSS App Server Unit 及 Social Web App Server Unit 是以 JSP Server 及 DB Server 所開發，而 RSS File Generator

Unit則是一利用JREx Browser Component^[17]這一Java Class Library所開發出的Java Service，目的在背景抓取網頁內容並萃取出程式所需的資訊。在Client與Server之間則以AJAX方法來通訊，輸出文件格式包含Web Service模式的XML，以供Client程式讀取並可成為API讓third party的Extension使用；輸出亦包含HTML，以提供user一個friendly的UI。

3-1 RSS 類元件系統架構

3-1-1 系統架構

本系統使用的軟體，在 server 方面，Application Server 為 Tomcat，Database Server 為 MySQL DB Server，Service Program 為一 Java 程式，使用到了 JREx Class Library；而在 client 方面，則是利用 Firefox 瀏覽器為 platform，在其上使用 JavaScript 與 XUL 來開發 Extension，來做為本研究的客戶端。以下是詳細的版本資訊：

表 1 RSS 類元件使用軟體版本

角色	軟體
應用伺服器	Apache Tomcat 6.0
JAVA Virtual Machine	JDK 1.5.0
資料庫軟體	MySQL Server 5.0
JAVA 類別庫	JREx 1.0b1
瀏覽器	Mozilla Firefox 2.0.0.20

系統架構請參考如下的系統架構圖：

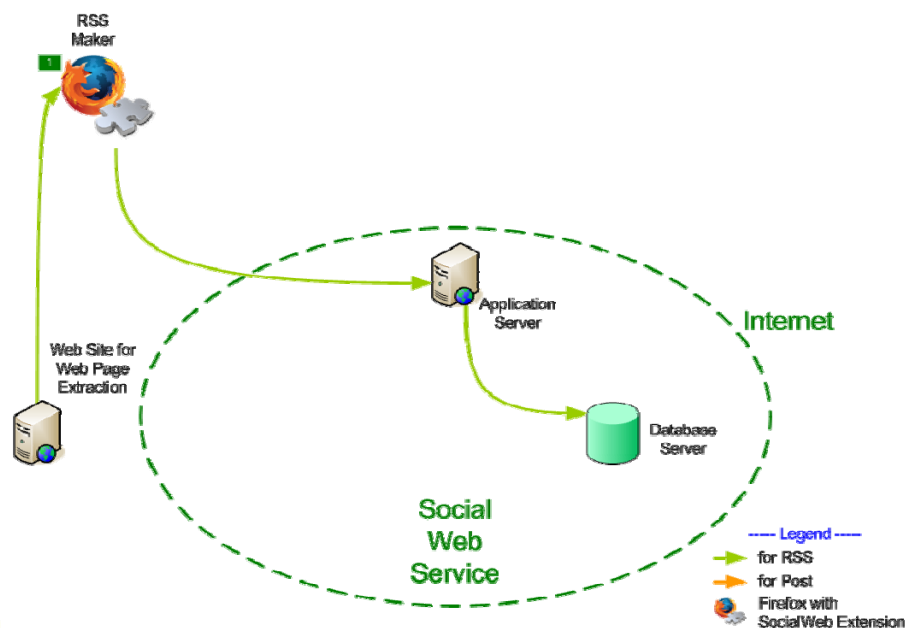


圖 13 RSS 類元件系統架構圖 1

在系統架構圖中的深綠色(深色)虛線圓圈就是本研究所開發的社群網路服務，本社群網路服務中 RSS 系統的部份，其資料流在上圖中是以淺綠色(淺色)實線來表示。當使用者欲在任意一個網頁中，自訂自己所需的 RSS，需要先在使用者的 Firefox 瀏覽器上安裝本研究開發的 Extension，即 SocialWeb Extension (簡稱 SWE)，在此以 1 號瀏覽器表示之，火狐圖示代表 Firefox，而 puzzle piece 圖示代表 Firefox 上安裝的 SWE。使用者在此瀏覽器上載入某個網頁，在網頁上選擇他所需的資訊以做為 RSS 的內容，SWE 會將使用者選取的網頁條目，分析其 XPath 之後，連同使用者提供的一些資訊，例如使用者名稱、密碼、RSS 類別、更新 RSS 的時間間隔等，將此資訊使用 AJAX 技術傳遞給 Application Server (簡稱 App Server)，記錄在 DB 之內，如上圖所示。

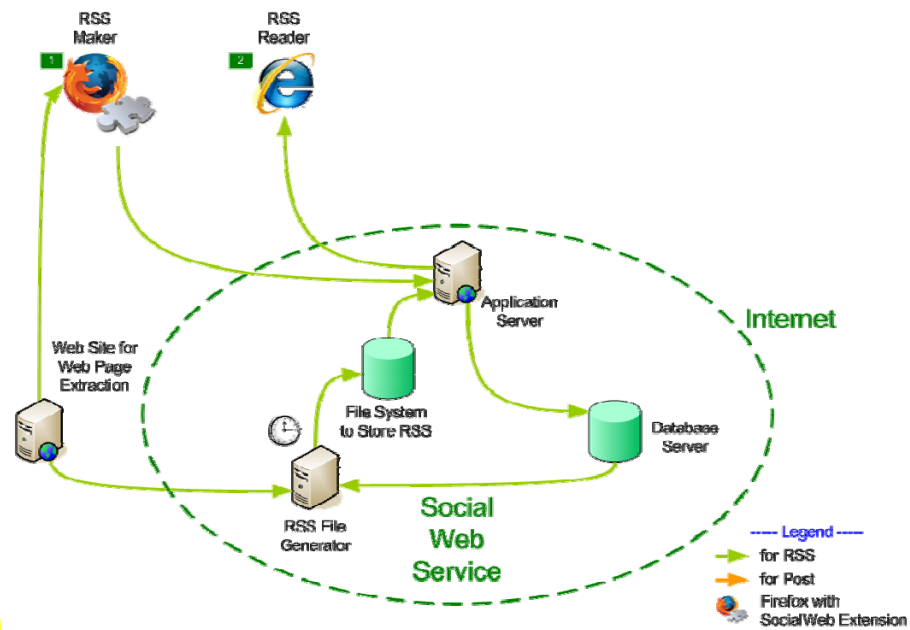


圖 14 RSS 類元件系統架構圖 2

之後，App Server 上的一個 Service，即上圖的 RSS File Generator，會依照使用者要求的更新 RSS 時間間隔，每隔一段時間，就連線到當初使用者抓取 RSS 條目的網站，即圖左的 Web Site for Web Page Extraction，此服務將使用者設定的網頁抓取下來之後，會去剖析 (parse) 這個網頁，依據 DB 中所存的 XPath，抓到目標網頁中對應內容後，產生 RSS file，放在一個 File System 中，App Server 便能依此提供 RSS 服務。

RSS Feed 完成之後，任何一個使用者，不論是否安裝 SocialWeb Extension，也不論是否使用 Firefox 瀏覽器，只要他所使用的瀏覽器支援 RSS 或使用 RSS Reader，即圖中的 2 號瀏覽器，皆能連到 App Server，來訂閱此項 RSS。此即本研究中社群網路服務中 RSS 服務的系統架構。

3-1-2 Client 端元件

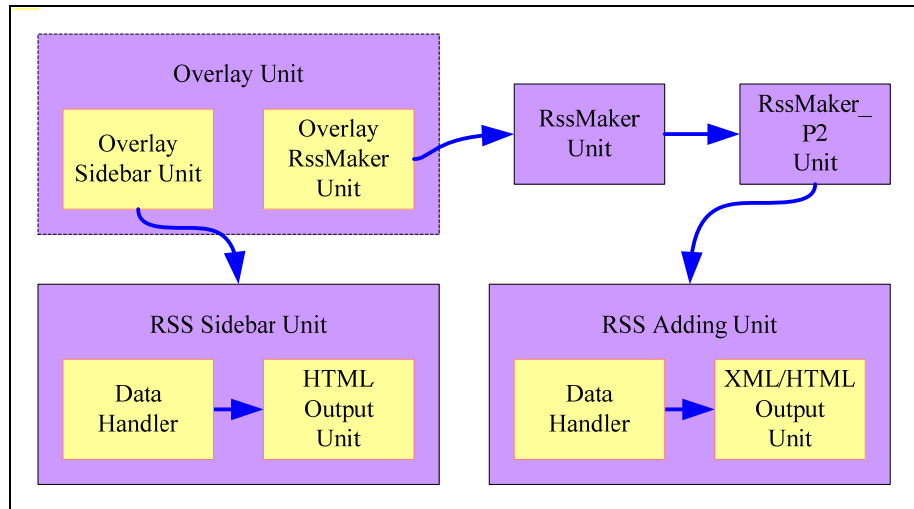


圖 15 RSS 類元件之 Client 端元件系統架構圖

Client 端元件主要包含 Overlay Unit, RssMaker Unit, RssMaker_P2 Unit 等三個元件，其中 Overlay Unit 又可細分為 Overlay Sidebar Unit 及 Overlay RssMaker Unit 等兩個元件。其流程是當 user 在瀏覽器上選擇 RSS 條目之後，即交由 RssMaker Unit 及 RssMaker_P2 Unit 讓 user 確認並輸入更多的資訊，之後送到 App Server 端的元件 RSS Adding Unit 加以分析儲存；另一方面 user 亦可在瀏覽器上觀看此網頁的 RSS，此是由 RSS Sidebar Unit 加以處理。各元件分別描述如下：

1. Overlay Unit

此元件的主要功能是用自訂的使用者介面 (UI) 覆蓋 (overlay) Firefox browser 的 UI，其 UI 之一畫面如下：

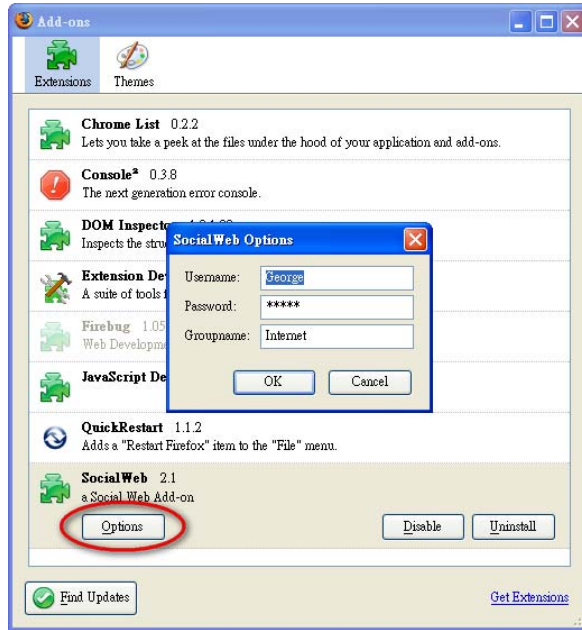


圖 16 SocialWeb Extension Options UI

此 UI 是讓使用者設定 SocialWeb Extension 的選項，包含 Username, Password 及 Groupname。此元件的其餘子元件在以下介紹。

2. Overlay Sidebar Unit

此 Unit 的 UI 是在 Firefox 的 top menu 中顯示 SocialWeb Extension 的部份功能選項，其 UI 如下：

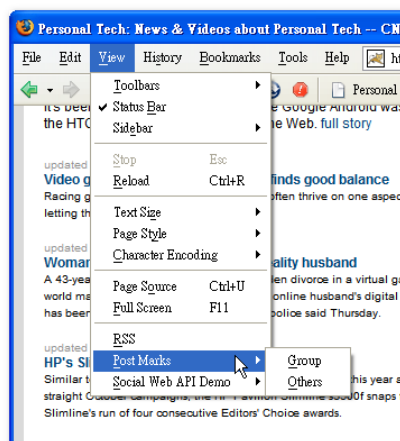


圖 17 RSS 類元件之 Overlay Unit 之 menu

其中屬於 RSS 類元件的指令為” RSS”，其功用是在 browser 左側顯示一

sidebar，此 sidebar 會呼叫屬於 Application Server 端元件的 RSS Sidebar Unit，以顯示目前網頁中已存在的 RSS。當 user 造訪另一網頁（網址發生改變）時，此元件會被觸發，而重新呼叫 RSS Sidebar Unit 以同步更新而能對應到新的網址。

3. Overlay RssMaker Unit

此 Unit 的 UI 是在 Firefox 的 context menu 中，展現 SocialWeb Extension 另一部份的 menuitem，其 UI 如下：



圖 18 RSS 類元件之 Overlay Unit 之 context menu

其中屬於 RSS 類元件的指令及其說明如下表所示。

表 2 RSS 類元件之 Overlay Unit 之 context menu 之指令說明

指令	說明
Marked As First Title	將滑鼠所在段落記錄為 RSS 的第一項目的 title
Marked As First Desc.	將滑鼠所在段落記錄為 RSS 的第一項目的描述
Marked As Last Title	將滑鼠所在段落記錄為 RSS 的末項目的 title
Marked As Last Desc.	將滑鼠所在段落記錄為 RSS 的非第一項目的描述
Make RSS...	呼叫 RssMaker Unit 以開始製作 RSS
View RSS	顯示目前網頁中已存在的 RSS，功能同” View” menu 中的” RSS”

此單元並包含每個 menuitem 按下後的對應處理，如記錄對應段落的 XPath 等，當使用者按下” Make RSS...”後，將記錄的資料傳往下一單元：

RssMaker Unit。

4. RssMaker Unit

此單元將取得的 Title XPath & Desc. XPath 加以分析後，將網頁資料萃取並顯示出來，以供使用者判斷資料的正確性，其畫面如下所示：

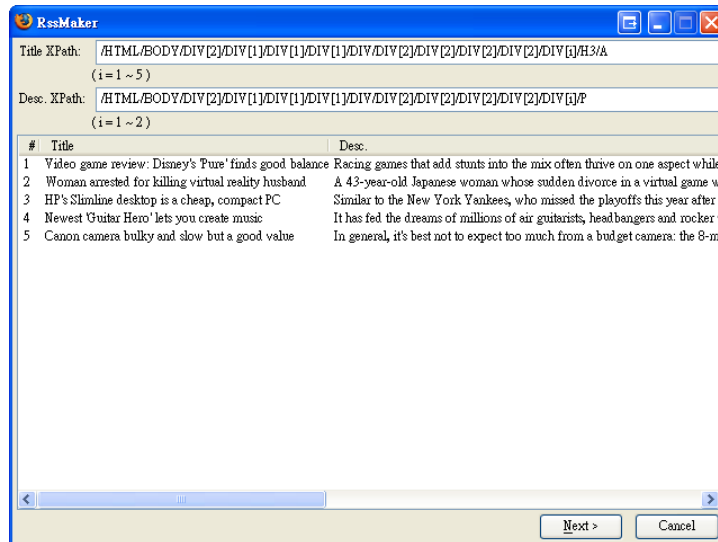


圖 19 RSS 類元件之 RssMaker Unit 之 UI

當 user 按下 Next 後，資料便送往下一單元：RssMaker_P2 Unit

5. RssMaker_P2 Unit

此 unit 匯集相關資料如網頁 title、網址、XPath 等供 user 檢視，並讓 user 鍵入其他資料，其 UI 如下所示：

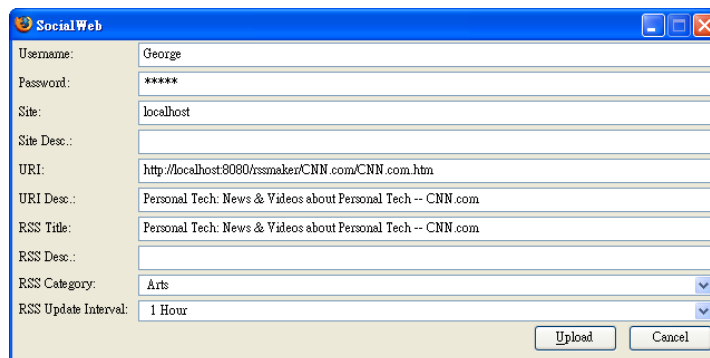


圖 20 RSS 類元件之 RssMaker_P2 Unit 之 UI

在此部份的視窗，是要求使用者輸入關於此 RSS 的相關資訊，包含有：

表 3 RSS 類元件之 RssMaker_P2 Unit 之 UI 之欄位說明

欄位	說明
Username	user 在 app server 上所註冊的帳號，系統自動由 Options 帶入
Password	user 在 app server 上所註冊的密碼，系統自動由 Options 帶入
Site	網站的網址，系統自動帶入
Site Desc.	對此網站的描述
URI	網頁的網址，系統自動帶入
URI Desc.	對此網頁的描述，系統自動帶入網頁的 title
RSS Title:	系統自動帶入網頁的 title
RSS Desc.	對此 RSS 的描述
RSS Category	讓使用者選擇此 RSS 是屬於何種 category，有如下的選擇： Arts, Business, Computers, Games, Health, Home, Kids and Teens, News, Recreation, Reference, Regional, Science, Shopping, Society, Sports, World
RSS Update Interval	user 決定系統多久產生 RSS 一次，預設為 1hr

當使用者填完此 RSS 的相關資訊後，按 Upload，資訊即會傳送到 Application Server 端的元件。

3-1-3 Application Server 端元件

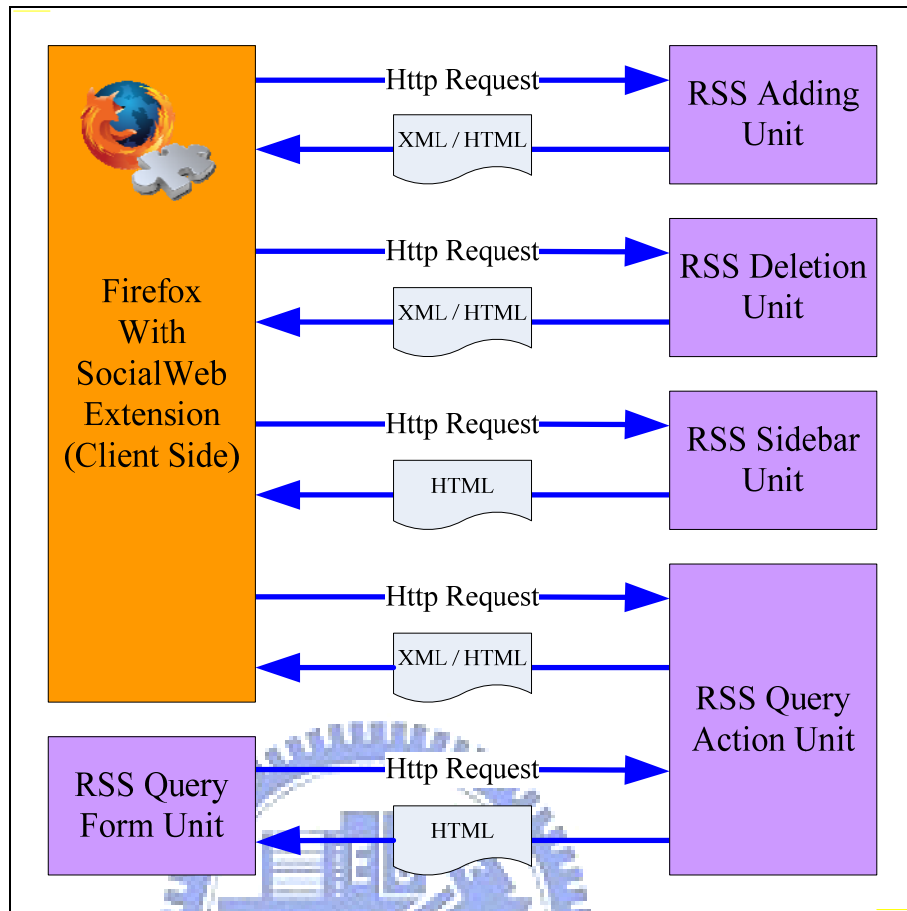


圖 21 RSS 類元件之 Application Server 端元件系統架構圖

此子系統主要包含 RSS Adding Unit, RSS Deletion Unit, RSS Sidebar Unit, RSS Query Form Unit, RSS Query Action Unit 等五個元件，可針對 SocialWeb Extension 或網頁送來的 RSS 要求，執行新增、刪除、查詢等動作，並以 XML 或 HTML 回傳，或是送出某網頁的 RSS 查詢結果。各元件分別描述如下：

1. RSS Adding Unit

此為記錄 RSS 相關資訊的單元，當 user 在 RssMaker_P2 Unit 按下 Upload 後，此單元即被呼叫，處理完後顯示如下的 UI。



圖 22 RSS 類元件之 RSS Adding Unit 之 UI

此畫面代表 RSS 已儲存成功，並顯示連結可呼叫 RSS Query Action Unit，查詢並訂閱系統中現存的 RSS。

2. RSS Deletion Unit

此為刪除 RSS 相關資訊的單元。

3. RSS Sidebar Unit

此單元可由 Overlay Unit 呼叫之，以在 browser 左側顯示 RSS sidebar，其 UI 如下圖。

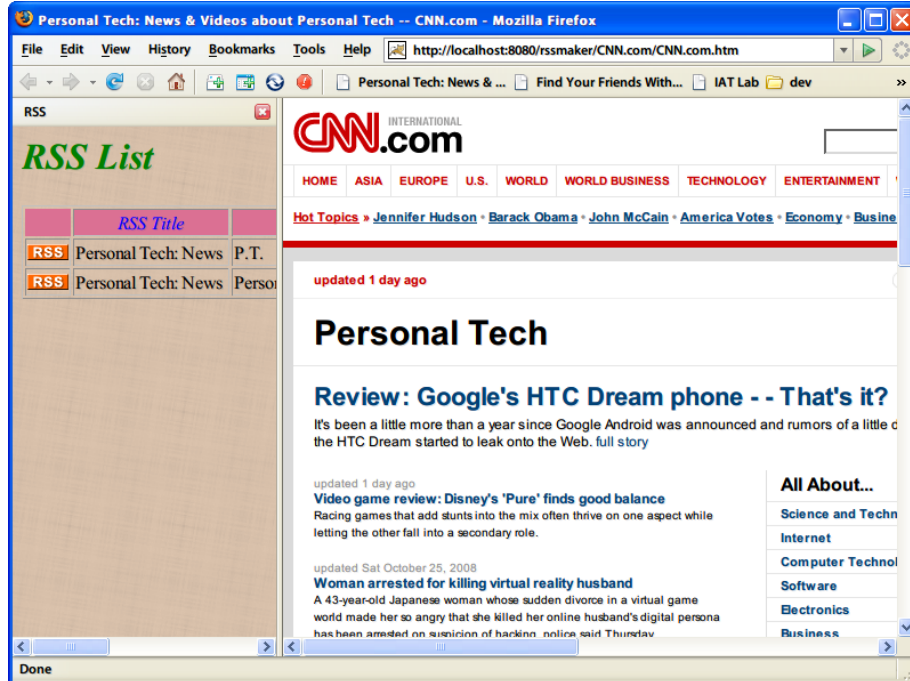


圖 23 RSS 類元件之 RSS Sidebar Unit 之 UI

User 可依此來訂閱此網頁的 RSS。當 user 造訪另一網頁時，此元件會感知新的網址，而同步更新以對應到新的網址。

4. RSS Query Form Unit

讓 user 由網頁對 RSS 做條件查詢，其 UI 如下圖。

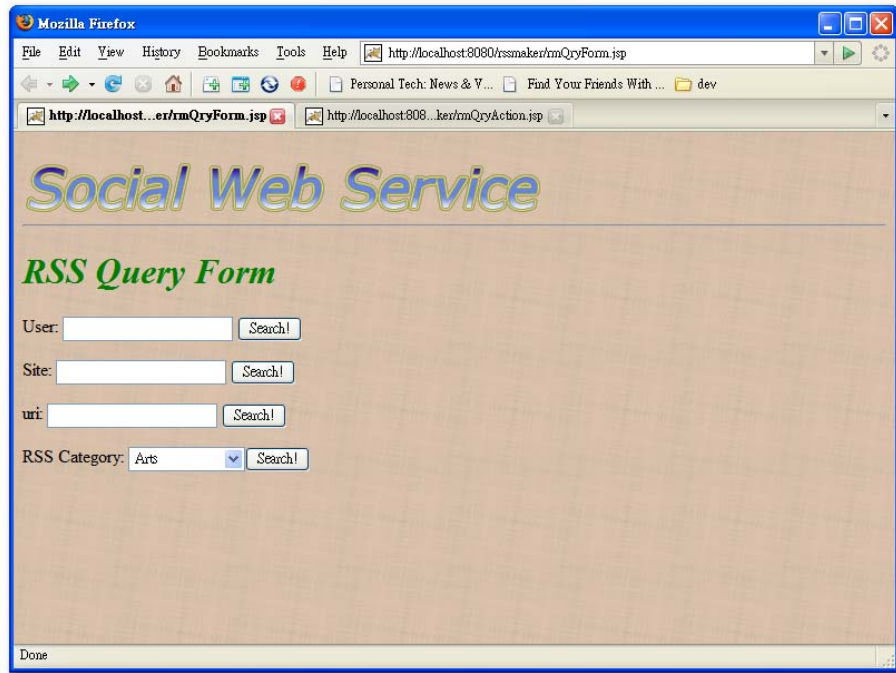


圖 24 RSS 類元件之 RSS Query Form Unit 之 UI

5. RSS Query Action Unit

RSS 之查詢引擎，接受來自 SocialWeb Extension 及 RSS Query Form Unit 之查詢，並返回 XML 或 HTML，其 XML 格式與 HTML 如下兩圖。

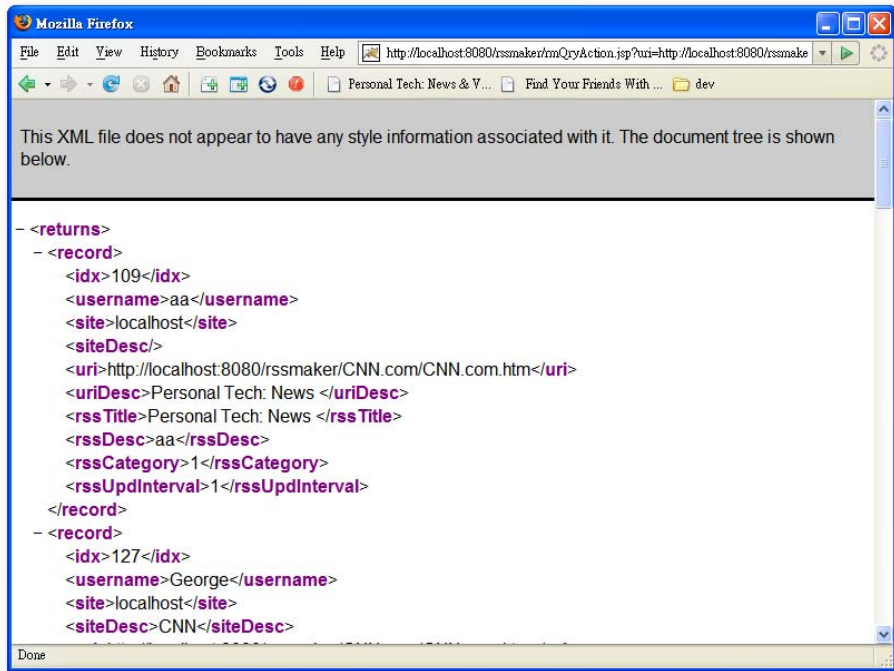


圖 25 RSS 類元件之 RSS Query Action Unit 之返回 XML

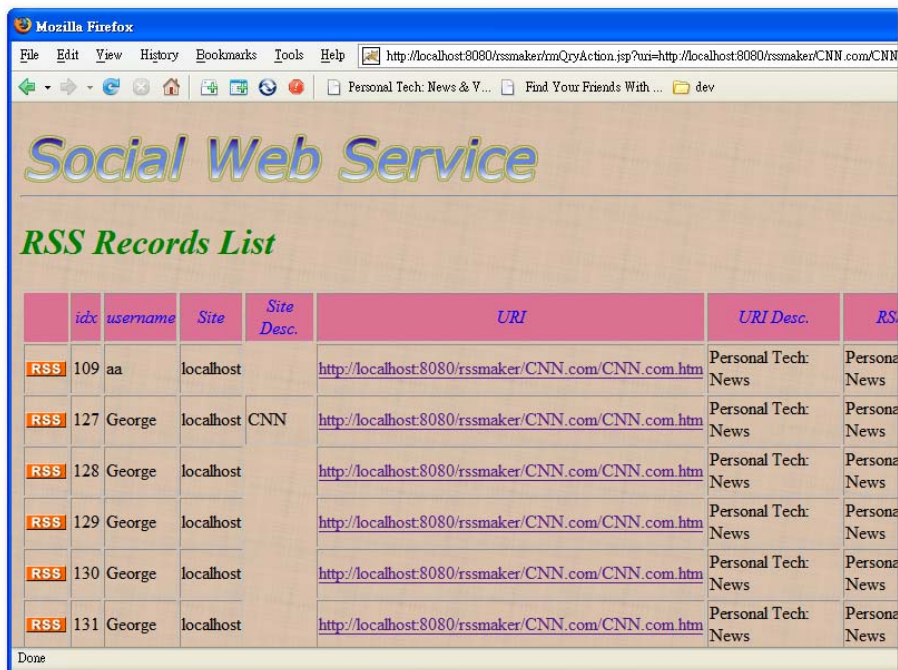


圖 26 RSS 類元件之 RSS Query Action Unit 之返回 HTML

本節所述之各單元的詳細呼叫參數說明請見3-3-1 RSS類API一節所述。

3-1-4 RSS File Generator 元件

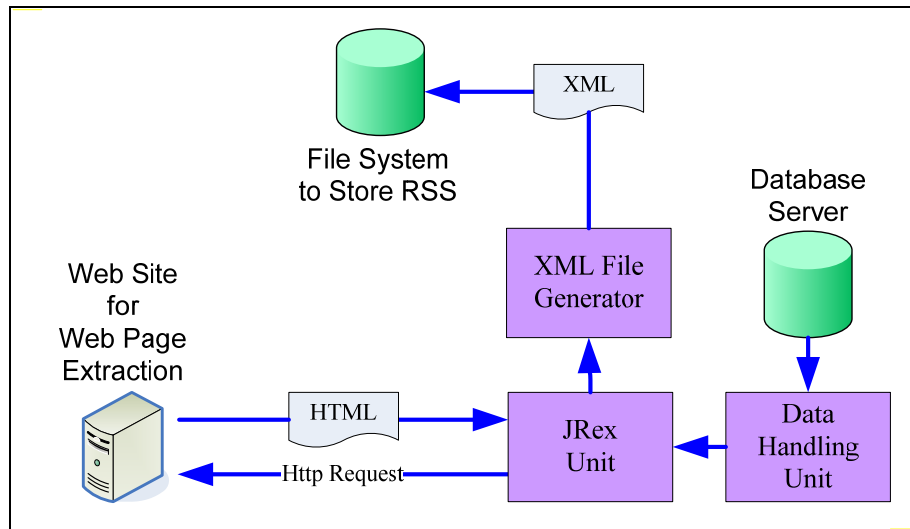


圖 27 RSS 類元件之 RSS File Generator 元件系統架構圖

此子系統主要功能為定時連線到目標網站抓取網頁，剖析網頁內容後，產生 RSS 檔案。此子系統主要包含 Data Handling Unit, JRex Unit, XML File Generator 等三個元件，分別描述如下：

1. Data Handling Unit

此元件主要功能為前往 DB 抓取資料，包含網址、XPath 等，得到資料後送往 JRex Unit 處理。

```

C:\WINDOWS\system32\cmd.exe
C:\Carlos\My Documents\Graduate\Orientation to Research\rssmaker-daemon>C:\Program Files\Java\jdk1.5.0_16\bin\java -cp .;C:\Program Files\MySQL\mysql-connector-java-5.1.5-bin.jar;C:\Program Files\Green\JReX\jrex-bin-nolog-1.0b1_dom3\JReX.jar;..\rssmaker-uehapp\WEB-INF\classes -Djrex.dom.enable=true -Djrex.gre.path=C:\Program Files\Green\mozilla-win32-1.7.13\mozilla -DJREX_DEBUG=false RssMaker
sql=SELECT * FROM rn_uri_xpath where enabled=1 order by idx
idx=109 begin
***** JReXL inited *****
***** JREX-Logging disabled *****
xpath=/HTML/BODY/DIV[2]/DIV[1]/DIV[1]/DIV[1]/DIV[1]/DIV[2]/DIV[2]/DIV[2]/DIV[2]/DIV[1]/H3/A
xpath=/HTML/BODY/DIV[2]/DIV[1]/DIV[1]/DIV[1]/DIV[1]/DIV[2]/DIV[2]/DIV[2]/DIV[2]/DIV[2]/H3/A
.. \rssmaker-uehapp\rnRss\109.xml
idx=109 end

idx=127 begin
xpath=/HTML/BODY/DIV[2]/DIV[1]/DIV[1]/DIV[1]/DIV[1]/DIV[2]/DIV[2]/DIV[2]/DIV[2]/DIV[1]/H3/A
xpath=/HTML/BODY/DIV[2]/DIV[1]/DIV[1]/DIV[1]/DIV[1]/DIV[2]/DIV[2]/DIV[2]/DIV[2]/DIV[2]/H3/A
xpath=/HTML/BODY/DIV[2]/DIV[1]/DIV[1]/DIV[1]/DIV[1]/DIV[2]/DIV[2]/DIV[2]/DIV[2]/DIV[3]/H3/A
xpath=/HTML/BODY/DIV[2]/DIV[1]/DIV[1]/DIV[1]/DIV[1]/DIV[2]/DIV[2]/DIV[2]/DIV[2]/DIV[4]/H3/A
xpath=/HTML/BODY/DIV[2]/DIV[1]/DIV[1]/DIV[1]/DIV[1]/DIV[2]/DIV[2]/DIV[2]/DIV[2]/DIV[5]/H3/A
xpath=/HTML/BODY/DIV[2]/DIV[1]/DIV[1]/DIV[1]/DIV[1]/DIV[2]/DIV[2]/DIV[2]/DIV[2]/DIV[6]/H3/A
xpath=/HTML/BODY/DIV[2]/DIV[1]/DIV[1]/DIV[1]/DIV[1]/DIV[2]/DIV[2]/DIV[2]/DIV[2]/DIV[7]/H3/A
xpath=/HTML/BODY/DIV[2]/DIV[1]/DIV[1]/DIV[1]/DIV[1]/DIV[2]/DIV[2]/DIV[2]/DIV[2]/DIV[8]/H3/A
xpath=/HTML/BODY/DIV[2]/DIV[1]/DIV[1]/DIV[1]/DIV[1]/DIV[2]/DIV[2]/DIV[2]/DIV[2]/DIV[9]/H3/A
xpath=/HTML/BODY/DIV[2]/DIV[1]/DIV[1]/DIV[1]/DIV[1]/DIV[2]/DIV[2]/DIV[2]/DIV[2]/DIV[10]/H3/A
.. \rssmaker-uehapp\rnRss\127.xml
idx=127 end

idx=128 begin
xpath=/HTML/BODY/DIV[2]/DIV[1]/DIV[1]/DIV[1]/DIV[1]/DIV[2]/DIV[2]/DIV[2]/DIV[2]/DIV[1]/H3/A
xpath=/HTML/BODY/DIV[2]/DIV[1]/DIV[1]/DIV[1]/DIV[1]/DIV[2]/DIV[2]/DIV[2]/DIV[2]/DIV[2]/H3/A

```

圖 28 RSS 類元件之 Data Handling Unit 之執行畫面

2. JReX Unit

此元件利用 JReX 這一 free 且 open source 的 Java Browser Component，連線到目標網站抓取網頁，取回後用 XPath 去取得網頁特定位置的内容，之後送往 XML File Generator 處理。

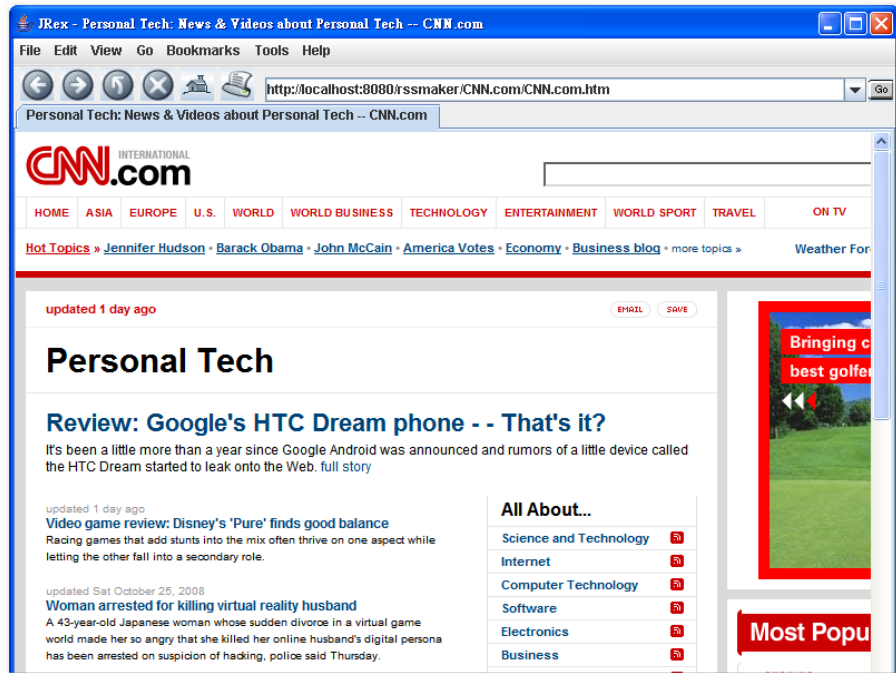


圖 29 RSS 類元件之 JReX Unit 之執行畫面

3. XML File Generator

此元件依照 JReX Unit 所取得的資料，依序產生 RSS 檔案，之後 user 即可利用 RSS Reader 連線到 Application Server 來取得 RSS。

```

82.xml
1 <?xml version="1.0" encoding="big5"?><rss version=
2 <channel>
3 <title>Personal Tech: News </title>
4 <link><![CDATA[http://edition.cnn.com/TECH/ptech/a
5 <description><![CDATA[Personal Tech]]></descriptio
6 <webMaster><![CDATA[Carlos Zhang]]></webMaster>
7 <copyright><![CDATA[Carlos Zhang]]></copyright>
8 <language>zh-TW</language>
9 <ttl>60</ttl>
10 <pubDate>2008-10-28 20:07:46</pubDate>
11
12 <item>
13 <title><![CDATA[Video game review: Disney's 'Pure'
14 <pubDate>Tue, 22 Jan 2008 21:40:00 GMT</pubDate>
15 <author>RssMaker</author>
16 <link><![CDATA[http://edition.cnn.com/TECH/ptech/a
17 <description><![CDATA[Racing games that add stunts
18 </item>
19
20 <item>
21 <title><![CDATA[Woman arrested for killing virtual
22 <pubDate>Tue, 22 Jan 2008 21:40:00 GMT</pubDate>
23 <author>RssMaker</author>

```

圖 30 RSS 類元件之 XML File Generator 所產生的 XML 內容

3-2 Social Web 類元件系統架構

3-2-1 系統架構

本系統使用的軟體，在 server 方面，Application Server 為 Tomcat，Database Server 為 MySQL DB Server，Service Program 為一 Java 程式，使用到了 JREx Class Library；而在 client 方面，則同樣是利用 Firefox 瀏覽器為 platform，在其上使用 JavaScript 與 XUL 來開發 Extension，來做為本研究的客戶端。以下是詳細的版本資訊：

表 4 Social Web 類元件使用軟體版本

角色	軟體
應用伺服器	Apache Tomcat 6.0
JAVA Virtual Machine	JDK 1.5.0
資料庫軟體	MySQL Server 5.0
瀏覽器	Mozilla Firefox 2.0.0.20

系統架構請參考如下的系統架構圖：

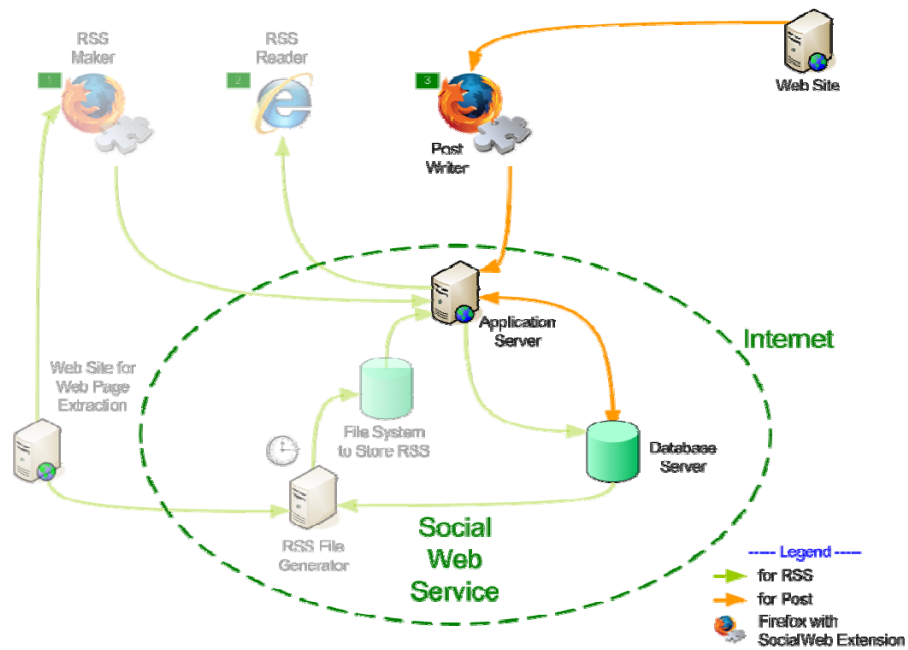


圖 31 Social Web 類元件系統架構圖 1

Social Web 系統的資料流是由上圖中的橘色(深色)實線所表示。首先，要使用此功能的 Firefox 瀏覽器，必須先安裝本研究的 SocialWeb Extension，之後即可在任何網站上編寫自己的 post 或閱讀他人所留的 post。在上圖中，3 號瀏覽器即是代表執行 post 功能的瀏覽器，此瀏覽器必須是 Firefox 並且安裝了 SocialWeb Extension，所以 icon 是一帶有 puzzle piece 的火狐。之後，此 browser 的使用者便可以針對任何網站上的網頁，並且針對此網頁中的任何段落，留下 post。留言的同時，可設定此 post 的瀏覽權限，是本 user 所屬的 group 能觀看，亦或是任意人皆能觀看，與此同時，亦可一併上傳附檔，以供其他人參考。當 3 號瀏覽器的使用者按下上傳 button 之後，其留言以及其他相關的設定和附檔等，便由 SocialWeb Extension 中的 post API，上傳至 App Server，然後由 App Server 儲存至 Database Server 中。

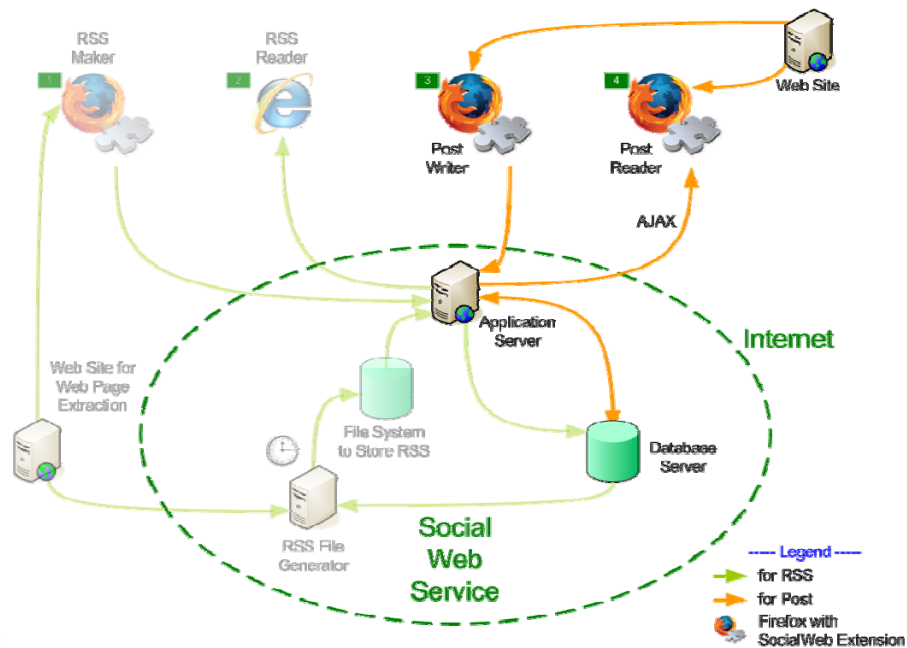


圖 32 Social Web 類元件系統架構圖 2

往後，若有瀏覽器欲觀看某網頁上，由 SocialWeb Extension 所留下的 post，即可利用 Extension 上的 View - Post Marks 功能，選擇欲觀看 Group 或 Others 的 Post Marks。之後，此網頁上有 post 的段落，便會出現 Post 的編號標記，使用者點選某一編號標記後，在網頁的左邊，便會出現 sidebar，經由 SocialWeb Extension 中的 AJAX Post API，往 App Server 抓取留言之後，顯示在左方的 sidebar 之內。除此之外，亦可在編號標記的旁邊，顯示一對話泡泡，其中的內容亦是其他網友針對此網頁段落所留下的 post。

RSS 功能為 Web 2.0 中社群網路的重要功能之一，所以本社群網路研究的 post 系統亦提供了 RSS 的能力。在顯示 post 的 sidebar 上方，有一 RSS icon，使用者可點選此 icon，便可訂閱此網頁段落的 post，如此一來，即可隨時看到其他人對此網頁段落的最新留言，而能即時掌握到重要的訊息。

針對網頁中的某段落留言，本研究中的 Social Web 系統是藉由記錄段落的 XPath 來達成。這樣做的好處是，當網頁的段落有部份的變動時，若 XPath 沒有變動到，則針對此段落的留言仍能黏 (stick) 在此段落上，不會使留言變成文不對題。然而，當網頁有較大幅度的變動時，某段落可能完全被移除，自

然此段落的 XPath 也不復存在。針對這種情形，本研究也特別做了處理，做法是針對網頁來設置一個 post 收集筒，置於網頁的下方。當 SocialWeb Extension 發現到有留言的 XPath 已不存在時，便將這些 post 放置在 post 收集筒之中，當使用者點選此 post 收集筒，便能看到所有收集的留言，除此之外，亦能看到此留言原先對應的網頁段落，而能得知留言與網頁段落的上下文關係。

3-2-2 Client 端元件

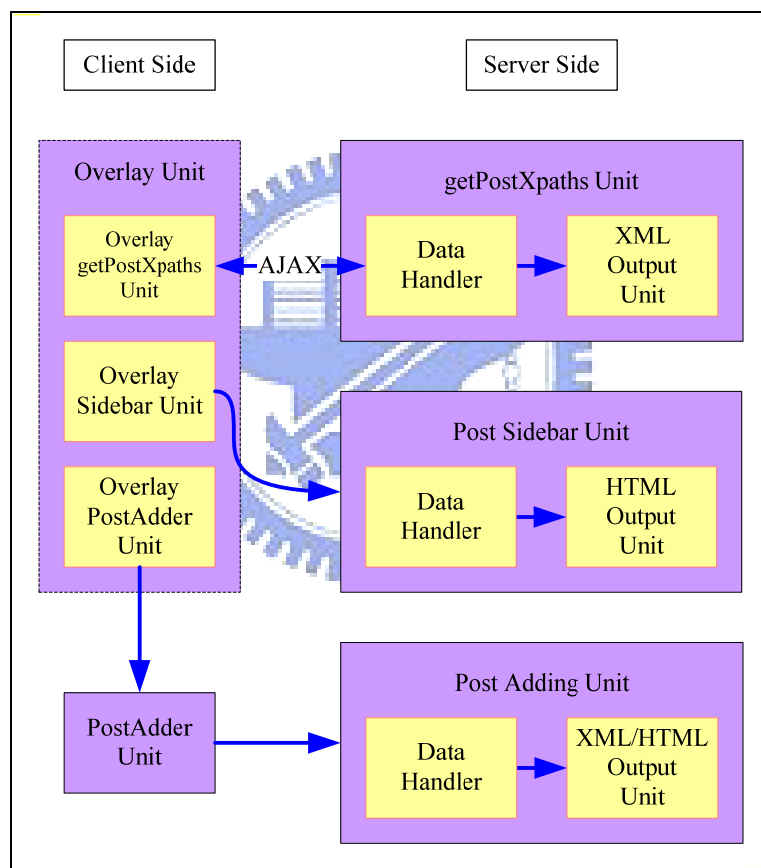


圖 33 Social Web 類元件之 Client 端元件系統架構圖

此子系統主要包含 Overlay Unit, PostAdder Unit 等兩大元件，其中 Overlay Unit 又包含三個元件，分別是 Overlay getPostXpaths Unit, Overlay Sidebar Unit 及 Overlay PostAdder Unit。Overlay getPostXpaths Unit 用以取得此網頁的 post 資訊，Overlay Sidebar Unit 用以在 browser 左側顯示

sidebar, Overlay PostAdder Unit & PostAdder Unit 合作以讓 user 新增 post。

各元件分別描述如下：

1. Overlay Unit

此元件的主要功能是用自訂的使用者介面 (UI) 覆蓋 (overlay) Firefox 的 UI。

2. Overlay getPostXpaths Unit

其 UI 是在 Firefox 的 menu 中顯示 Social Web 類元件的指令，即” Post Marks” 子 menu。

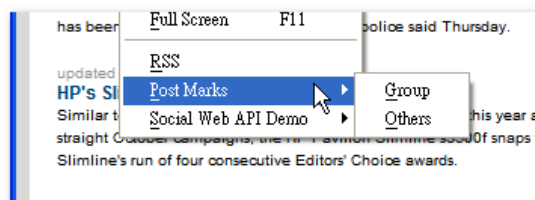


圖 34 Social Web 類元件之 Overlay Unit 之 UI 之一

此子 menu 的功能是在網頁中存在 post 的段落，在段落的末端顯示一代表 post 的圖示。選擇 Group 時，為顯示 group post 的圖示；選擇 Others 時，為顯示 public post 的圖示，如下所示。

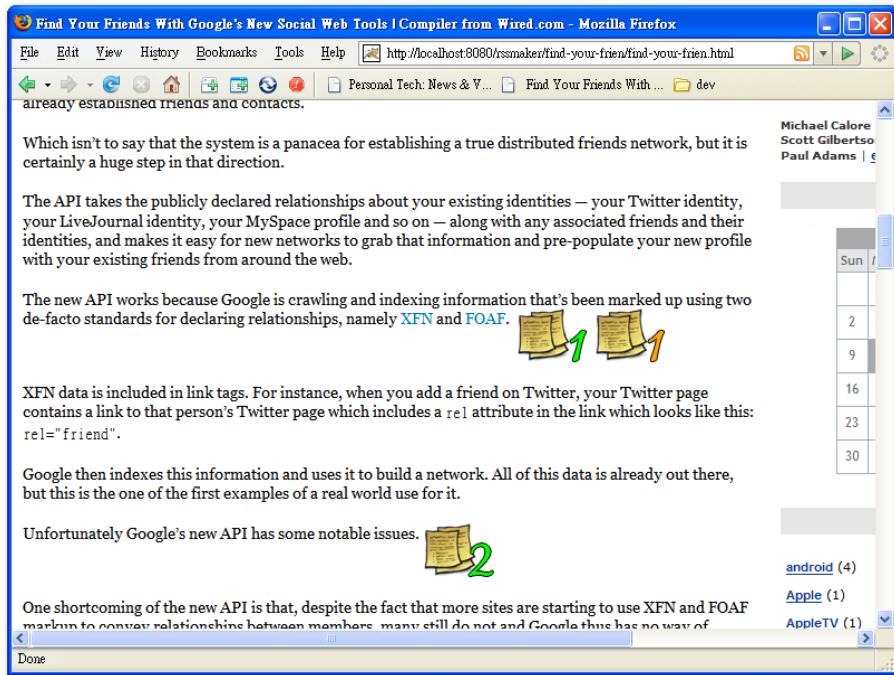


圖 35 Social Web 類元件之 Overlay Unit 之 UI 之二

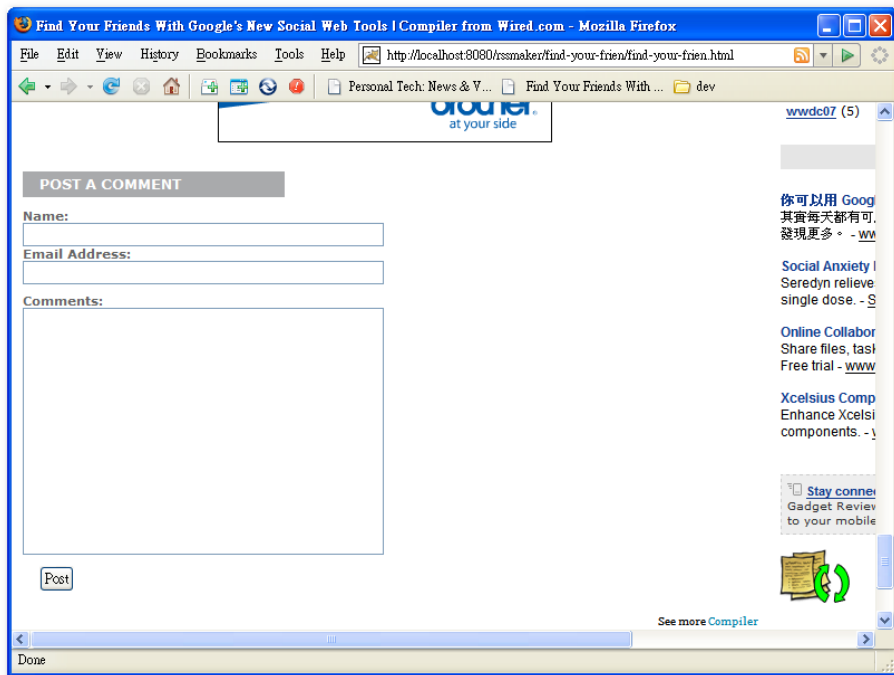






圖 36 Social Web 類元件之 Overlay Unit 之 UI 之三

網頁中出現的  圖示意義，如下表：

表 5 Social Web 類元件之 Overlay Unit 之 UI 圖示意義

圖示	意義
	群組留言 (Group Post)，數字為綠色。數字表示留言的時間序，post 只限定同群組的人閱讀。
	公共留言 (Public Post)，數字為橘色。數字表示留言的時間序，post 是任何人皆可閱讀。
	留言收集筒。post 所對應的段落不再存在時，便出現此圖示。綠色為 group post 的收集筒
	同上，橘色為 public post 的收集筒。

3. Overlay Sidebar Unit

其 UI 是在 Firefox 的 context menu 中，SocialWeb Extension 另一部份的 menuitem：

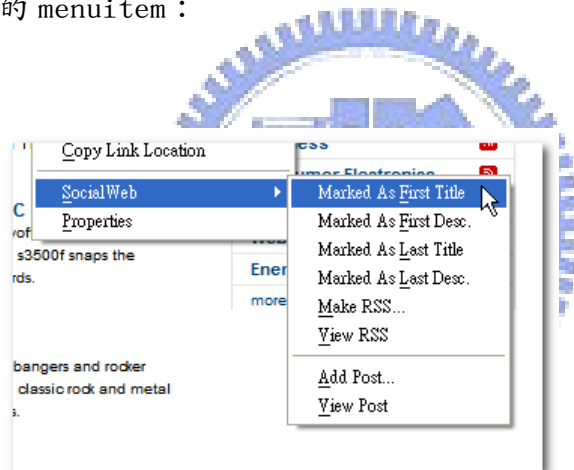


圖 37 Social Web 類元件之 Overlay Unit 之 context menu

其中屬於 Social Web 類元件的指令及其說明如下表所示。

表 6 Social Web 類元件之 Overlay Unit 之 context menu 指令說明

指令	說明
Add Post	在滑鼠所在段落新增一 post
View Post	顯示滑鼠所在段落的 post

本元件使用的是其中的” View Post” 指令，當 user 在某個 post 的 icon 上按下” View Post” 後，此元件便會將此 icon 的資訊，送往 Application

Server 上的 Post Sidebar Unit，同時在 Firefox browser 的左側，會出現 Sidebar，網頁中則出現 bubble box，如下圖所示。

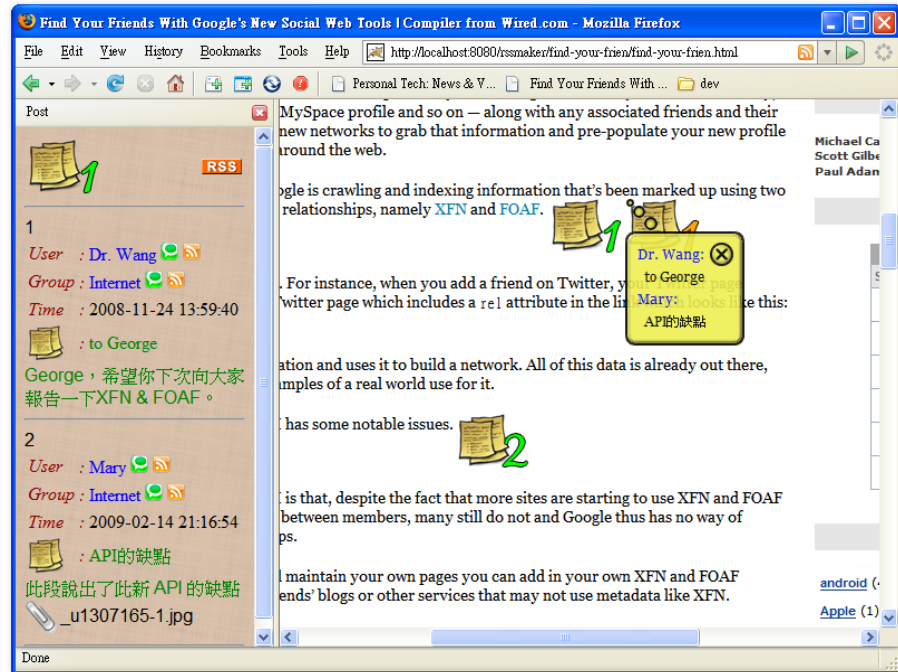


圖 38 Social Web 類元件之 Overlay Unit 之 Sidebar 圖一

當 user 點選收集筒的圖示亦會出現類似的 Sidebar 如下圖。

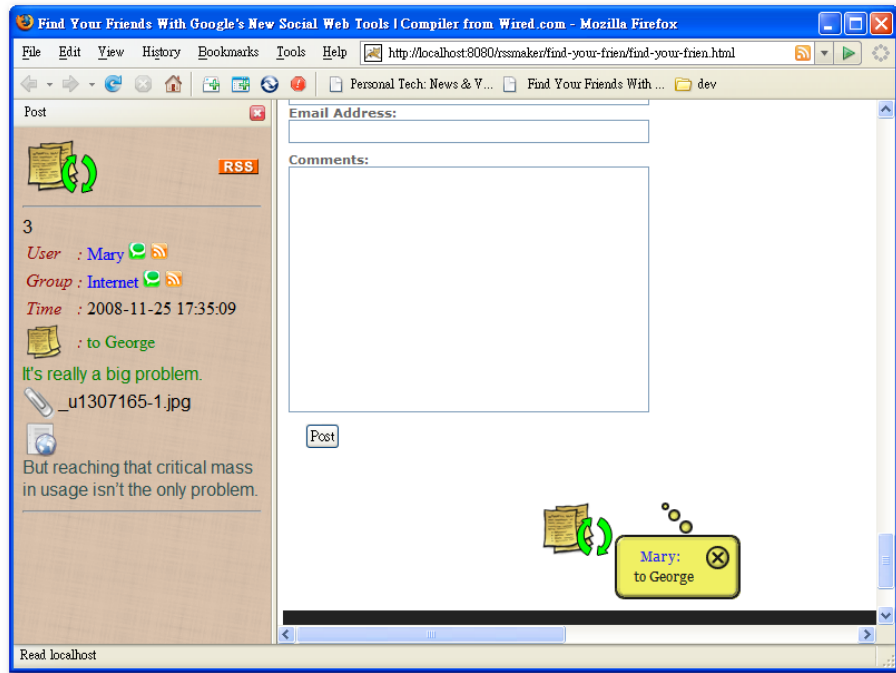




圖 39 Social Web 類元件之 Overlay Unit 之 Sidebar 圖二

網頁中出現的黃色 bubble box，內容為此 post icon 所在段落之 post 的 username 與 title。左側的 Sidebar 則顯示詳細的內容，其中左上角的  圖示代表底下的 post 是對應到 Group Post 第 1 號，而  圖示代表底下的 post 是對應到 Group Post 收集筒；右上角的 **RSS** 圖示則讓 user 可以訂閱此段落之 post 的 RSS。水平線分隔的是每則 post，其欄位說明如下表。

表 7 Social Web 類元件之 Overlay Unit 之 Sidebar 欄位說明

欄位	說明
User	留下 post 的 username。其後的  圖示為此 user 曾在哪些網頁留過 post；  圖示為此 user 曾在哪些網頁自製 RSS。
Group	留下 post 的 user 所屬的 group。其後的  圖示為此 group 曾在哪些網頁留過 post；  圖示為此 group 曾在哪些網頁自製 RSS。
Time	post 的時間
	post 的 comment title，其下為 comment 內容。當點選 comment title 時，會開啟新網頁，列出相同 group 且相同 title 的網頁縮圖；若 comment 中包含 url，亦可顯示這些網頁縮圖之間的連結關係圖。
	post 的 attachment。User 點選後即可下載。
	post 對應的原始段落內文。此為顯示收集簡 post 時才會顯示，以因應網頁的原始段落內文可能已經消失，以方便 user 觀看 post 時可得知原文與 post 的關係。

此元件並包含每個 menuitem 按下後的對應處理，如記錄對應段落的 XPath 等，當使用者按下” Add Post...” 後，將記錄的資料傳往下一單元：PostAdder Unit。

4. PostAdder Unit

此元件的 UI 如下所示。

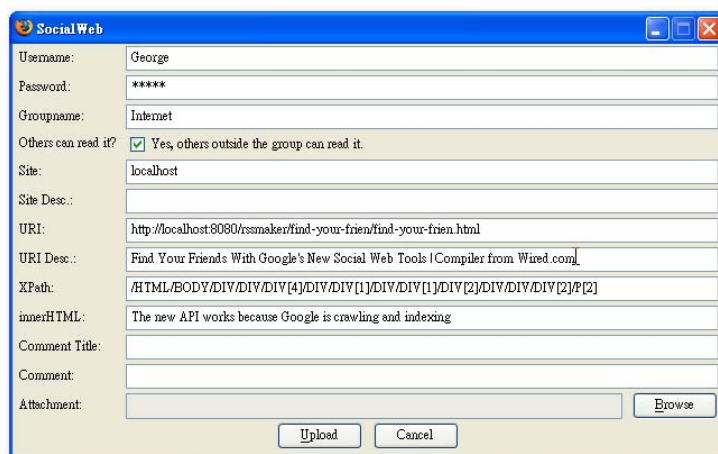


圖 40 Social Web 類元件之 PostAdder Unit 之 UI

表 8 Social Web 類元件之 PostAdder Unit 之欄位說明

Username	使用者 id。系統自動由 Options 帶入
Password	密碼。系統自動由 Options 帶入
Groupname	使用者所屬的 group，系統自動由 Options 帶入。在此若未填入 groupname，則表示 post 不屬於任何 group，任何人皆可讀取。
Others can read it?	若未選取則代表只有此 group 的成員有閱讀的權利。
Site	post 所在的網站。系統自動帶入。
Side Desc.	對此網站的描述。
URI	post 所在的網頁。系統自動帶入。
URI Desc.	對此網頁的描述。系統自動帶入網頁的 title。
XPath	post 所在段落的 XPath。系統自動帶入。
innerHTML	post 所在段落的內文，當 post 的 xpath 不存在時，此資料可讓使用者得知上下文關係。
Comment Title	post 的 title。
Comment	post 的內容。
Attachment	user 可用 Browser 鈕上傳檔案以供他人下載參考。

寫完之後，便可按下 Upload 將你的 post 資訊傳至 Post Adding Unit，此將在 Application Server 端元件介紹。

3-2-3 Application Server 端元件

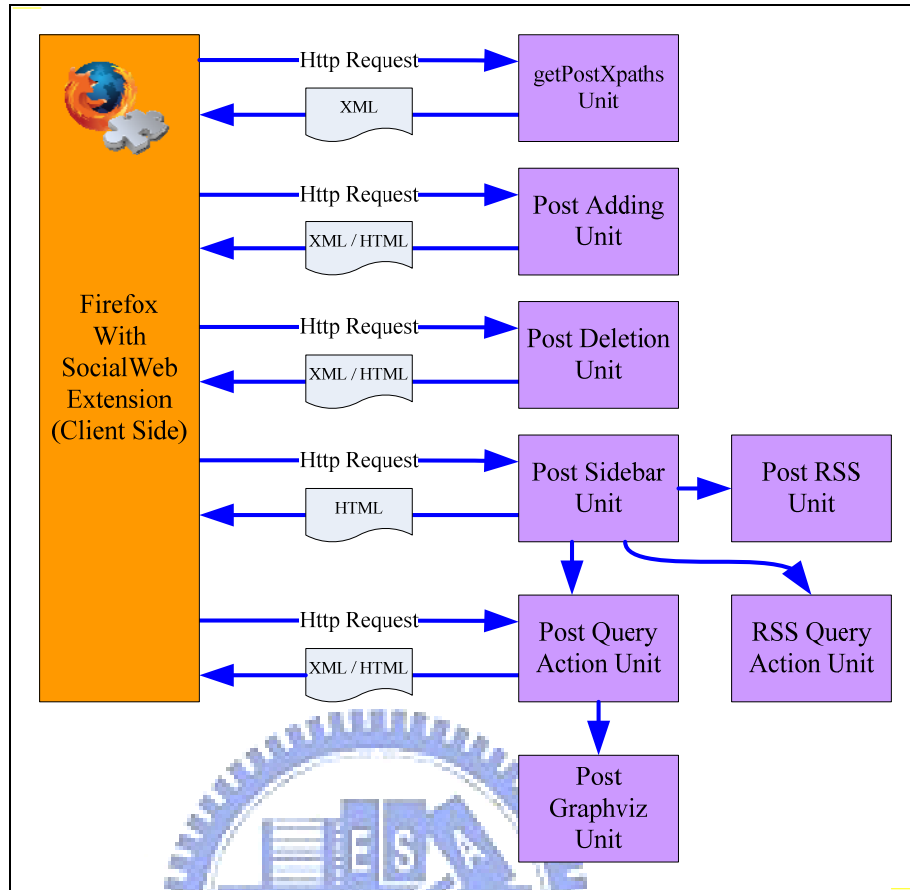


圖 41 Social Web 類元件之 Application Server 端元件系統架構圖

此子系統主要包含 getPostXpaths Unit, Post Adding Unit, Post Deletion Unit, Post Sidebar Unit, Post RSS Unit, Post Query Action Unit 等六個元件。getPostXpaths Unit 用以回傳 post 資訊，Post Adding Unit, Post Deletion Unit & Post Query Action Unit 分別用以執行 comment 及附檔等的新增、刪除、查詢等工作，Post Sidebar Unit 用以在 browser 左側顯示 post 相關資訊的 sidebar，並可呼叫其他元件產生更多的應用。各元件分別描述如下：

1. getPostXpaths Unit

此元件的功能是傳回某 uri 中全部 post 的 XPath，SocialWeb Extension 取得 XPath 資訊後，即可在網頁中顯示 Post Marks。

2. Post Adding Unit

此為記錄 Post 相關資訊的元件，當 user 在 PostAdder Unit 按下 Upload 後，此元件即被呼叫，以供 user 上傳 comment 以及 attachment，並傳回 HTML 或是 web service 方式的 XML，下圖是 HTML 的畫面。

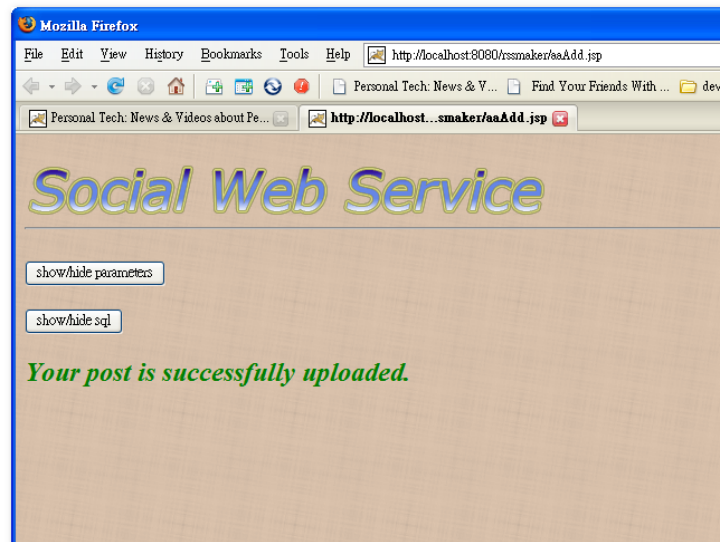


圖 42 Social Web 類元件之 Post Adding Unit 之 UI

3. Post Deletion Unit

此元件之功能是供 user 或程式刪除 post。

4. Post Sidebar Unit

此元件是當 SocialWeb Extension 之 sidebar unit 被 trigger 時，被其呼叫以將 post 資訊送往 client 端，以在 browser 左側顯示 Post sidebar，其 UI 即介紹 Client 端元件時之錯誤！找不到參照來源。及錯誤！找不到參照來源。。

5. Post RSS Unit

此元件是當 user 按下 sidebar 右上角的 **RSS** 圖示時，讓 user 可以訂閱此段落之 post 的 RSS，將會出現 RSS subscription 的 UI 如下圖：

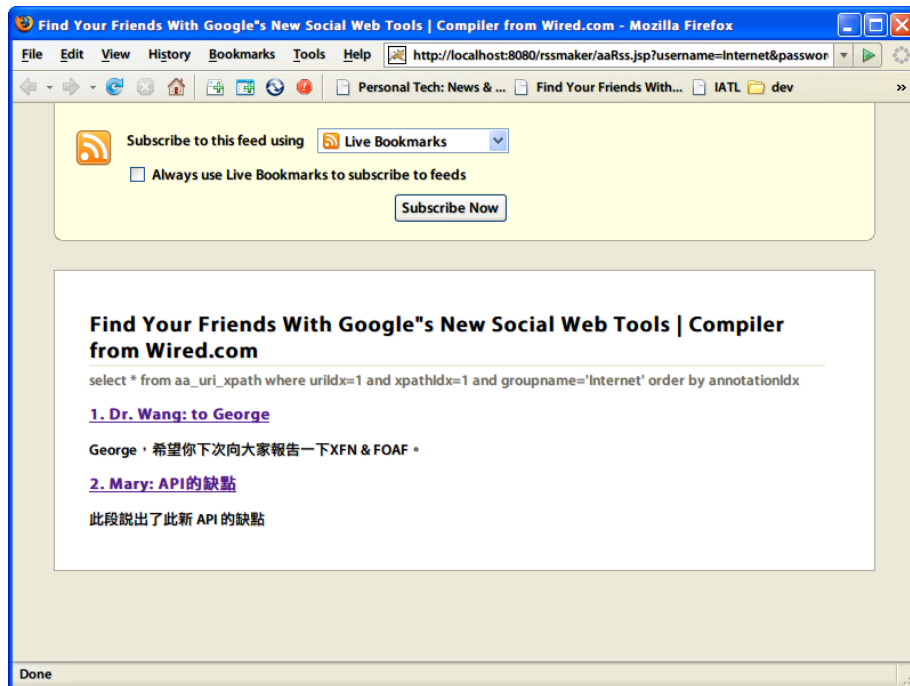


圖 43 Social Web 類元件之 Post RSS Unit 之 UI

6. Post Query Action Unit

此元件為 post 之查詢引擎，接受來自 SocialWeb Extension 之查詢，並能以 HTML 或 XML 格式輸出，其 HTML 輸出畫面如下：

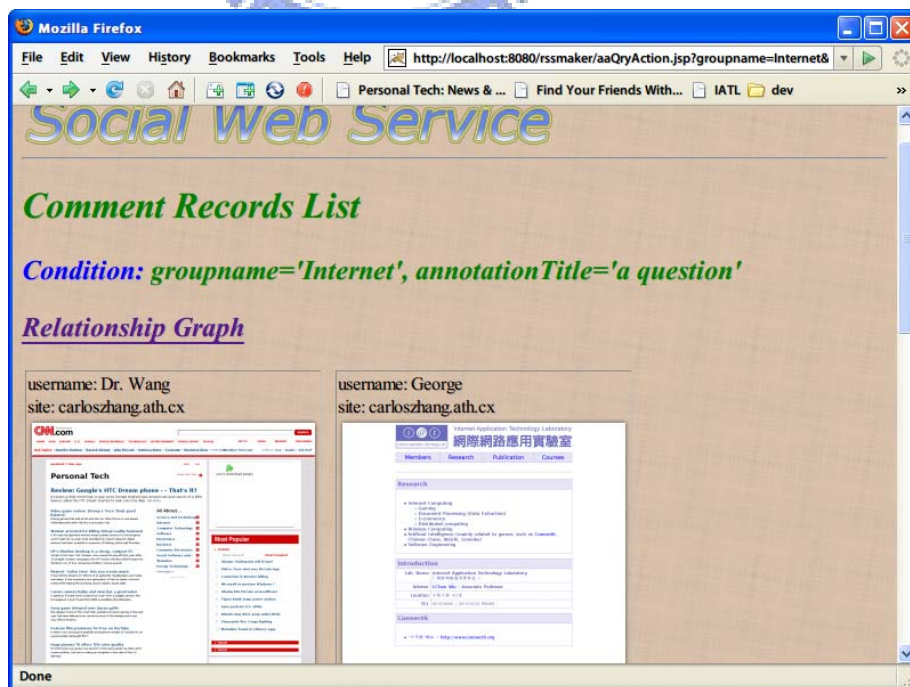


圖 44 Social Web 類元件之 Post Query Action Unit 之返回 HTML

其中的Condition為其查詢條件，然後此元件會利用MozShot^[18]服務在網頁下方顯示post所在網頁的縮圖。當post之中有url link時，網頁上的Relationship Graph連結會呼叫Post Graphviz元件顯示在同一群組中，具有相同討論標題的各網頁之間的連結關係圖，此圖將於下節詳述。另外此元件可選擇輸出XML，畫面如下：

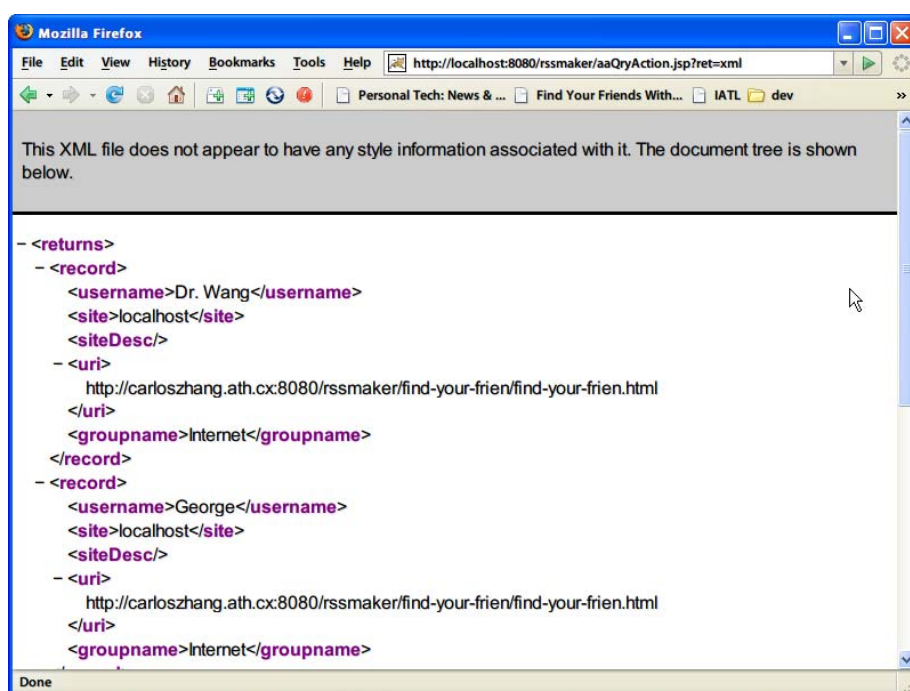


圖 45 Social Web 類元件之 Post Query Action Unit 之返回 XML

7. Post Graph Unit

此元件來自於上節中 Post Query Action 元件的呼叫，當 Post 之中有 URL Link 時，其用以顯示在同一群組中，具有相同討論標題的各網頁之間的連結關係圖，如此使用者可概念性的了解哪些網頁是被連結最多次，因而此網頁可能相對的會較具有重要性。此元件之輸出係先將各網頁之縮圖利用 MozShot 服務存入伺服器後，並將查詢結果產生一 dot 檔如下例：

```

1 | digraph G {
2 |   graph [bgcolor=transparent];
3 |   node [color=blue style="setlinewidth(1)"];
4 |   rankdir=LR;
5 |   0 [shapefile="0.png", URL="http://carloszhang.ath.cx:8080/rssmaker/find-your-frie
6 |   1 [shapefile="1.png", URL="http://www.google.com"];
7 |   2 [shapefile="2.png", URL="http://carloszhang.ath.cx:8080/rssmaker/CNN.com/CNN.co
8 |   3 [shapefile="3.png", URL="http://carloszhang.ath.cx:8080/rssmaker/IATL/Internet%
9 |   4 [shapefile="4.png", URL="http://www.yahoo.com"];
10 | 0->2;
11 | 0->4;
12 | 1->4;
13 | 2->4;
14 | 3->4;
15 | 4->1;
16 | }

```

圖 46 Social Web 類元件之 Post Graphviz Unit 產生之 dot 檔

接著便能利用 Graphviz 程式，產生其連結關係圖檔及影像地圖檔後，嵌入 HTML 中如下所示：

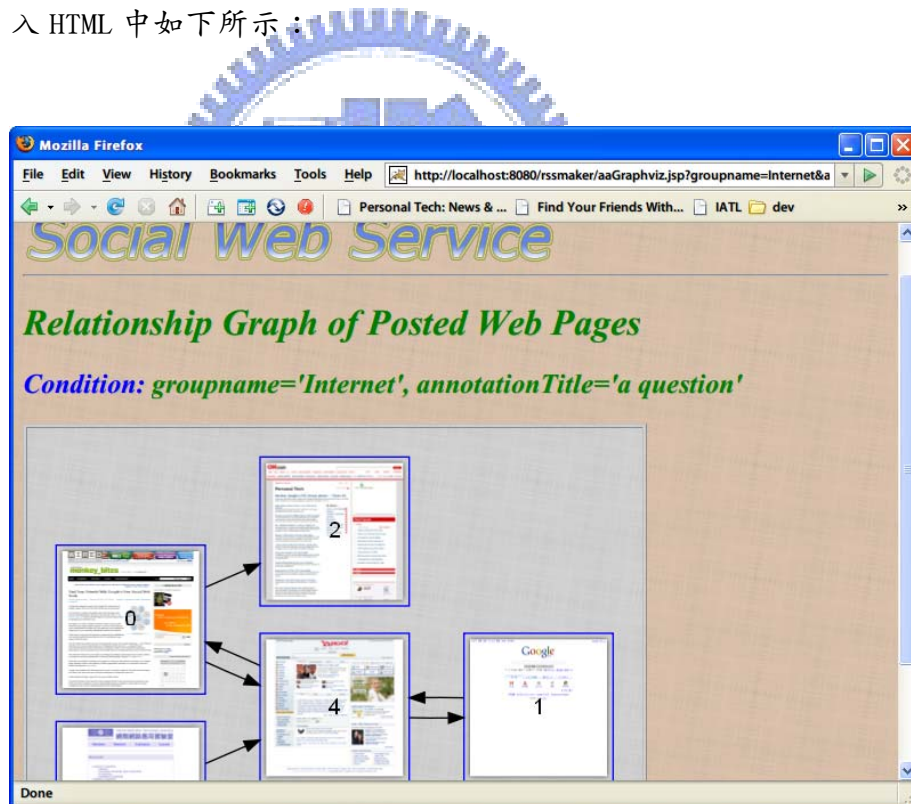


圖 47 Social Web 類元件之 Post Graphviz Unit 之返回

本節所述之各單元的詳細呼叫參數說明請見 3-3-2 Social Web 類 API 一節所述。

API 在目前網站應用的潮流，是至為重要的，可以讓網站的應用更為多樣化，本社群網路服務的研究，亦發展出一套 API，讓 third party 能夠依據自身的需求，訂製自己的 Firefox Extension。有了這些 API，third party 廠商能夠以此來做成各式各樣的 Firefox Extensions，進而使得各種的應用能被發展出來。而如此的 Social Web Service API 對於商業模式的應用來說，在未來也可能可以藉著例如留言者頭像、或是留言 sidebar 的自訂網頁等來創造出更多的使用方式。

Social Web 系統的 API 系統架構如下圖所示：

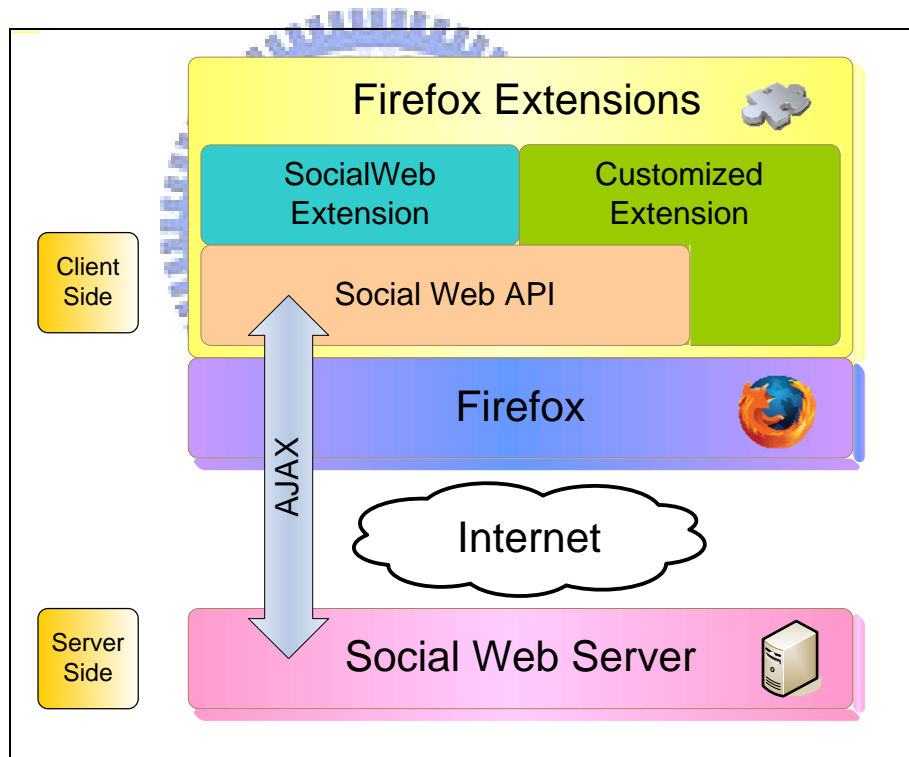


圖 48 API Model Diagram of the Social Web Service

在上圖中，Internet 上方的方塊是代表 client side，即 Firefox browser，Internet 下方的方塊則是代表 server side，即 Social Web Server。在 Firefox browser 之上，則是本研究所開發的 Social Web API，其利用 AJAX

技術進行與 Social Web Server 的溝通，在 Social Web API 之上，則是本研究的另兩個子系統：RSS 子系統與 Annotation 子系統，兩者合稱為 SocialWeb Extension，其利用了 Social Web API 所提供的 function calls，與 Social Web Server 通訊，以達成他們各自的系統功能。

而在右上角綠色的部份 (Customized Extension)，則是第三方 (third party) 所自行開發的 Extension。在這個部份，它可以與 Firefox 直接結合，亦可以利用本研究所開發的數十種 Social Web API，輕易的開發出各式各樣的可讓 Firefox 使用者利用的社群網路 Extension。

Social Web Service 之 API 基本上是一組 AJAX function call。所謂的 AJAX，即 Async JavaScript And XML，是一種網頁的發展技術，用來制作互動式的網頁應用程式。其依靠與後端的伺服器傳遞小量的資料，而讓使用者感覺網頁的回應性更為即時，每次使用者做了一些動作之後，整個網頁不再需要重新整理就能做幾乎立即的回應。AJAX 所使用的技術包含了 XHTML, DOM, XMLHttpRequest object 及 XML，所以，實際上，AJAX 是一群技術的合稱。

目前，本研究所開發的 Social Web API 的 AJAX function Call 主要分為四大類，包含了使用者及群組管理類、RSS 管理類、Annotation 管理類及其他類，以下分別介紹之。

A. 使用者及群組管理類：

本類為使用者及使用者所屬群組的管理，包含

- 新增 user
- 刪除 user
- 修改 user 的 group

修改 user 的 group

新增 user to group

刪除 user from group

- 新增 group
- 刪除 group
- user 所屬的 groups
- group 所含的 users

B. RSS 管理類：

本類為對網頁所產生的 RSS 的管理，包含

- 新增 RSS
- 刪除 RSS

可依據 index 或 site 或 URI 或 RSS category 而刪除之。

- 查詢某 user 的 RSS
- 查詢某 user 的 RSS category

針對此 user/group 播放廣告做行銷，廣告平台

廣告平台：在網頁上方放廣告，可由 Extension 開發者提供或由我提供

- 查詢某 site 的 RSS
- 查詢某 site 的 RSS category

- 查詢某 URI 的 RSS
- 查詢某的 RSS category

- 查詢某 RSS category 的 RSS
- 查詢某 RSS category 的 user

與人互動，找到志同道合的朋友，可以互相留言，或邀請加入 group 等，結合成社群網路。

- 查詢某 RSS category 的 site
- 上列的組合，如

查詢某 user 於某 site 的 RSS

查詢某 user 於某 uri 的 RSS

- 輸出可選擇 html 或是 xml

C. Annotation 管理類

本類為對使用者所產生的 annotation 的管理，包含

- 新增 annotation
- 刪除 annotation
- 查詢某 URI 的 group annotation
- 查詢某 URI 的 others annotation
- 查詢某 URI 的 user (在某網頁裡留言的使用者)
- 查詢某 URI 的 group (在某網頁裡留言的群組)
- 輸出可選擇 html 或是 xml

D. 其他類：

本類為對伺服器的設定以及一些顯示的細項的管理，包含

- 設定伺服器主機
- 設定伺服器連接埠
- 設定伺服器路徑
- 設定群組留言之影像檔名前綴
- 設定群組留言之影像檔延伸名
- 設定其他留言之影像檔名前綴
- 設定其他留言之影像檔延伸名

上述的 RssMaker 子系統以及 Annotation 子系統即為使用本研究的 Social Web API 所實作而成。此 Social Web API 主要分為三大部份，第一類是使用者管理類，例如新增或刪除使用者、新增或刪除群組、變更使用者所屬的群組等。第二類是 RSS 管理類，例如新增某網頁的 RSS、查詢某使用者的 RSS 屬於哪些類別、查詢某 RSS 類別有哪些使用者等。第三類是 Annotation 管理類，基本的如新增或刪除 Annotation、進階的如查詢在某網頁裡某群組的 annotation、

或查詢在某網頁裡留言的使用者、或查詢某使用者或群組曾在哪些網頁留言等。詳細的 API 列表如下所述。

3-3-1 RSS 類 API

RSS add	
Description	新增 RSS
URL	http://localhost:8080/rssmaker/rmAdd.jsp
Method	post/get
Parameters	<p>site=...</p> <p>siteDesc=...</p> <p>uri=...</p> <p>uriDesc=...</p> <p>rssTitle=...</p> <p>rssDesc=...</p> <p>rssCategory=...</p> <p>rssUpdInterval=...</p> <p>titleXPathPrefix: title xpath 的前綴。例如當 xpath = “/HTML/BODY/DIV[2]/DIV[1]/DIV[1]/DIV[1]/DIV/DIV[2]/DIV[2]/DIV[2]/DIV[2]/DIV[i]/H3/A” , i=1~3 時, titleXPathPrefix = “/HTML/BODY/DIV[2]/DIV[1]/DIV[1]/DIV[1]/DIV/DIV[2]/DIV[2]/DIV[2]/DIV[2]/DIV[2]/DIV[2]/DIV[”。</p> <p>titleXPathSuffix: title xpath 的後綴。如前例, titleXPathSuffix = “]/H3/A”。</p> <p>titleXPathIdx1: title xpath 的索引 1。如前例, titleXPathIdx1 = “1”。</p> <p>titleXPathIdx2: title xpath 的索引 2。如前例, titleXPathIdx2 = “3”。</p> <p>descXPathPrefix: description xpath 的前綴</p> <p>descXPathSuffix: description xpath 的後綴</p> <p>descXPathIdx1: description xpath 的索引 1</p> <p>ret=xml/html</p>
Returns	<p><return code></p> <p>0: successful</p> <p>1: failed</p>

RSS delete		
	Description	删除 RSS
	URL	http://localhost:8080/rssmaker/rmDel.jsp
	Method	post/get
	Parameters	idx=... site=... uri=... rssCategory=... ret=xml/html
	Returns	<return code> 0: successful 1: failed

user's RSS		
	Description	某 user 的 RSS
	URL	http://localhost:8080/rssmaker/rmQryAction.jsp
	Method	post/get
	Parameters	username=... ret=xml/html

user's RSS category		
	Description	某 user 的 RSS category
	URL	http://localhost:8080/rssmaker/rmQryAction.jsp
	Method	post/get
	Parameters	username=... ret=xml/html

site's RSS		
	Description	某 site 的 RSS
	URL	http://localhost:8080/rssmaker/rmQryAction.jsp
	Method	post/get
	Parameters	site=... ret=xml/html

site's RSS category		
	Description	某 site 的 RSS category
	URL	http://localhost:8080/rssmaker/rmQryAction.jsp
	Method	post/get

	Parameters	site=... ret=xml/html
--	------------	--------------------------

uri's RSS		
	Description	某 uri 的 RSS
	URL	http://localhost:8080/rssmaker/rmQryAction.jsp
	Method	post/get
	Parameters	uri=... ret=xml/html

uri's RSS category		
	Description	某 uri 的 RSS category
	URL	http://localhost:8080/rssmaker/rmQryAction.jsp
	Method	post/get
	Parameters	uri=... ret=xml/html

RSS category's RSS		
	Description	某 RSS category 的 RSS
	URL	http://localhost:8080/rssmaker/rmQryAction.jsp
	Method	post/get
	Parameters	rssCategory=... ret=xml/html

RSS category's user		
	Description	某 RSS category 的 user
	URL	http://localhost:8080/rssmaker/rmQryAction.jsp
	Method	post/get
	Parameters	rssCategory=... ret=xml/html

RSS category's site		
	Description	某 RSS category 的 site
	URL	http://localhost:8080/rssmaker/rmQryAction.jsp
	Method	post/get
	Parameters	rssCategory=... ret=xml/html

3-3-2 Social Web 類 API

annotation add	
Description	新增 annotation
URL	http://localhost:8080/rssmaker/aaAdd.jsp
Method	post/get
Parameters	username=... password=... groupname=... othersCanReadIt=... site=... siteDesc=... uri=... uriDesc=... xpath=... annotationTitle=... annotation=...
Returns	<return code> 0: successful 1: failed

annotation delete	
Description	刪除 annotation。可依據 username 或 site 或 uri 或 groupname 而刪除之。
URL	http://localhost:8080/rssmaker/aaDel.jsp
Method	post/get
Parameters	username=... site=... uri=... groupname=...
Returns	<return code> 0: successful 1: failed

uri's group annotation

	Description	某 uri 的 group annotation。可依據 uri 或 xpath 或 group 而查詢之。
	URL	http://localhost:8080/rssmaker/aaSidebar.jsp
	Method	post/get
	Parameters	uri=... xpath=... group=...
	Returns	html

uri's others annotation		
	Description	某 uri 的 others annotation。可依據 uri 或 xpath 而查詢之。
	URL	http://localhost:8080/rssmaker/aaSidebar.jsp
	Method	post/get
	Parameters	uri=... xpath=...
	Returns	html

uri's user		
	Description	某 uri 有哪些 user 留言
	URL	http://localhost:8080/rssmaker/aaQryAction.jsp
	Method	post/get
	Parameters	uri=... groupBy=username [ret= html xml] (default: html)
	Returns	html/xml

uri's group		
	Description	某 uri 有哪些 group 留言
	URL	http://localhost:8080/rssmaker/aaQryAction.jsp
	Method	post/get
	Parameters	uri=... groupBy=groupname [ret= html xml] (default: html)
	Returns	html/xml

user's uri		
	Description	某 user 曾在哪些網址留言

	URL	http://localhost:8080/rssmaker/aaQryAction.jsp
	Method	post/get
	Parameters	username=... groupBy=uri [ret= html xml] (default: html)
	Returns	html/xml

3-3-3 使用者及群組管理類 API

useradd		
	Description	新增 user
	URL	http://localhost:8080/rssmaker/userGroup.jsp
	Method	post/get
	Parameters	action=useradd username=...

userdel		
	Description	刪除 user
	URL	http://localhost:8080/rssmaker/userGroup.jsp
	Method	post/get
	Parameters	action=userdel username=...

usermod		
	Description	修改 user 的 group
	URL	http://localhost:8080/rssmaker/userGroup.jsp
	Method	post/get
	Parameters	action=usermod username=... groupname=...

groupadd		
	Description	新增 group
	URL	http://localhost:8080/rssmaker/userGroup.jsp
	Method	post/get
	Parameters	action=groupadd groupname=...

groupdel		
Description	刪除 group	
URL	http://localhost:8080/rssmaker/userGroup.jsp	
Method	post/get	
Parameters	action=groupdel groupname=...	

groups		
Description	user 所屬的 groups	
URL	http://localhost:8080/rssmaker/userGroup.jsp	
Method	post/get	
Parameters	action=groups username=...	

getent_group		
Description	group 所含的 users	
URL	http://localhost:8080/rssmaker/userGroup.jsp	
Method	post/get	
Parameters	action=getent_group groupname=...	

3-3-4 其他類 API

setServerHost		
Description	設定伺服器主機	
URL	n/a	
Method	setServerHost(String host)	
Parameters	host (ex: "localhost")	

setServerPort		
Description	設定伺服器連接埠	
URL	n/a	
Method	setServerPort(String port)	
Parameters	port (ex: "8080")	

setServerPath		
	Description	設定伺服器路徑
	URL	n/a
	Method	setServerPath(String path)
	Parameters	path (ex: "/rssmaker/")

setGroupAnnoImgFileNamePrefix		
	Description	設定群組留言之影像檔名前綴
	URL	n/a
	Method	setGroupAnnoImgFileNamePrefix(String namePrefix)
	Parameters	namePrefix (ex: "gray")

setGroupAnnoImgFileExt		
	Description	設定群組留言之影像檔延伸名
	URL	n/a
	Method	setGroupAnnoImgFileExt(String ext)
	Parameters	ext (ex: ".png")

setOthersAnnoImgFileNamePrefix		
	Description	設定其他留言之影像檔名前綴
	URL	n/a
	Method	setOthersAnnoImgFileNamePrefix(String namePrefix)
	Parameters	namePrefix (ex: "yellow")


setOthersAnnoImgFileExt		
	Description	設定其他留言之影像檔延伸名
	URL	n/a
	Method	setOthersAnnoImgFileExt(String ext)
	Parameters	ext (ex: ".png")

3-4 應用範例

1. 應用例一：論文網站

有一個論文網站，教授可將此論文题目的最新發展成果經由 Social Web Service 的 RSS Making 功能製成 RSS，以便實驗室的所有成員或是對此题目有興趣的其他人都能即時且很方便的經由 RSS Feed 得知論文的最新發展，另一方面，實驗室的成員之間也能藉由 Social Web Service 的 Annotation 功能，在網頁的各個段落留下各自的心得；而藉由群組的設定，這些心得與討論可以只限於實驗室的成員才能看見；而且，更進一步的是，實驗室成員也能經由 Annotation RSS 的功能，即時且方便的得知實驗室其他成員的留言，或是教授的最新指示。

2. 應用例二：新聞網站



如棒球的大聯盟網站，雖然網站未提供留言的功能，但經由 Social Web Service，世界各地的球迷因此而能在此網站交流與討論；而且，網站或是網友，都能藉由 Social Web Service 的自訂 RSS 服務，產生網頁上各個條目的 RSS，並且能讓其他人訂閱此 RSS。RSS 可以是本日各比賽的比數，或是球員的最新動態等。

3. 應用例三：股票

股票網站上，使用者可以在網頁上顯示關注股票的股價，然而經由 Social Web Service 提供的自訂 RSS 功能，使用者可以將這些股票的股價或成交量等，方便且容易的製成 RSS，如此一來，不需要進入網站，也能很容易的隨時了解股價的變化。

4. 應用例四：API

網站的經營者，若想使用此 Social Web Service，然而有些想客製化的地

方，像是想建立自己的伺服器而不使用官方的伺服器，或是想開發自己的程式來對使用者的使用習慣作分析，或是要對使用者的權限作更詳盡的管理，可使用 Social Web Service 所提供的 API 來開發所需的功能，建立屬於自己網站的 Firefox Extension。



四、 相關文獻探討

4-1 Google Friend Connect

Google Friend Connect (以下簡稱GFC^[19]^[20]) 是一個Google所提供的online service, 其允許internet上的users在不同的網站上都可以與他們的朋友連絡。

GFC 是 Google 所開發的一個 Open Social 應用, 發表於 2008 年五月, GFC 的主要目的是簡化社群網站與非社群網站之間的連接, 並且將社群應用程式與內容的處理與表現方式加以標準化。它使用了許多的公開標準, 像是 signin 使用 OpenID、資料控制使用 oAuth、以及應用程式使用 Open Social 等。



圖 49 Google Friend Connect 例子

上圖是一 Google Friend Connect 的範例, 此人在其網站上的右邊欄加入了一個 GFC 的元件, 讓參觀此網站的網友可以加入並留言, 如此便成了一個小型的社群網站。

Google Friend Connect 雖是免費的，然而你要安裝在網站上卻必需要取得網站的同意。它不需要 web programming 的知識就能夠使得幾乎任何網站可以擁有社群應用程式與社群內容。

在目前其介面是英文的並且你無法選擇使用其他的語言，Google 迄未宣佈任何翻譯計畫，而且使用 GFC 的網站也無法自行翻譯這些語言。

茲列舉使用 GFC 的一些範例：

1. E-commerce Sites: 讓 users 在產品網頁上可以檢查他的朋友們是否已經購買或評論他所感興趣的產品。
2. News Sites: 讓 users 可以在網頁上寫下評論與意見，並留下他們真實的 profile。
3. Organization Sites: 對於組織的會員們之間，可以增強行銷與溝通。
4. Blogs and Personal Websites: 可以和作者的朋友們加強交流。



Fleck^[21]是一個Firefox的add-on，它讓你可以輕易的bookmark網站並且和你的朋友及小組分享，你也可以在每個網站上寫notes，讓其他看這些網站的人知道你對這些網站的觀點。這個add-on在你的browser上增加了一個Fleck按鈕，讓你可以輕易的存取Fleck介面。

這個 add-on 不需要 reload 網頁，它直接放一層 fleck layer 在網頁上面，所以像 Facebook profiles 這樣的網頁也能使用 fleck。

Fleck 的特點茲列舉如下：

- 組織並搜尋網站：此功能類似 del.icio.us，即線上書籤的功能，你可以把所有有趣的網站儲存在線上，因此你在任何地方、任何電腦、任何時間都能存取到它們。經由標籤(tags)的功能，你可以輕易的組織你的網站，有了這個方法可以方便的管理一大群的網站。
- 將網站存在 Fleck：經由 Fleck bookmarklet 的功能，你可以輕易的將你看到的任何網站儲存到 Fleck 上，並且和小組或朋友分享。
- 在 Fleck 分享並尋找網站：你可以經由 adding friends 的功能，在 Fleck 上看看你的朋友們都是在看哪些網站。
- 群組的工作方式：和一群朋友、工作團隊、或志同道合的團體分享 flecks，可以節省大量的時間，並且使得瀏覽網站變得更有效率。
- 與你的社群網路互通：你可以將你的公開 post 傳送到網路上其他的 services，如此你將可以使你的 work 能被更多人看見，目前你可以將你的 flecks 在 twitter 上做分享。
- 搜尋功能：你可以搜尋你自己的 flecks，而且你也可以搜尋你的朋友们的 flecks。

4-3 本系統與其他 Social Web 系統之比較

表 9 Social Web Service 與其他 Social Web 系統之比較

	Google Friend Connect	Fleck	本系統之 Social Web Service
Supported Web Site	Only Configurable Web Sites	Any Web Site	Any Web Site
Supported Browser	Any Browser	Firefox (for Now)	Firefox (for Now)
Stick Message to	Page Only	Coordinate Position	Paragraph / Page
Commented Paragraph History	N	N	Y
Gadgets / API	Versatile	None	Fixed
What can be shared	Message / Video	Message	Message / File
Customized RSS & Message RSS	N	N	Y
Group	N	Y	Y
Relationship Graph	N	N	Y

上表是本系統與其他 Social Web 系統做一比較，包含 Fleck 與目前最受歡迎的 Social Web 服務之一 — Google Friend Connect。在 Supported Web Site 方面，由於 GFC 在安裝時必須要在網站的目錄放置兩個檔案，若網站未開放此權限，則 GFC 就無法使用。舉例來說，Facebook 基於商業競爭的因素，即不開放其網站使用 GFC；而就 Social Web Service (下稱 SWS) 與 Fleck 來說，因其是安裝在 Client 的 browser 上，故網站無法對其做限制，因而可以用在任何網站。在 Supported Browser 方面，目前 SWS 只開發了 Firefox 的版本，所以造成使用上的一些限制，在未來，可能開發 Internet Explorer 的版本，如此即可涵蓋九成以上的使用者。在 Stick Message to 一項，SWS 可針對段落做 post，功能上比 GFC 更強大，而 Fleck 的 post 是以座標方式依附在網頁上，

這在網頁變動時，有可能會因此使得 post 無法對應到原來的段落。對於網頁的可能變動性，本研究的 SWS 也特別做了處理，當網頁產生變動時，會有一收集筒，使得舊有的 post 可以被收集在裡面，而且在讀取這些舊有的 post 的時候，SWS 更可以顯示當時留下 post 時的網頁內容，以供 user 對照。在擴充性方面，GFC 支援 OpenSocial 這個公開標準，故擴充性不錯；而 SWS 則提供了 Social Web API，可供 third party 開發相關的應用程式；Fleck 在這方面則是付之闕如。另外，在 Share 的能力上，GFC 可貼上 YouTube 的連結，因而可說具有 Video 的分享能力；而 SWS 則是可上傳檔案，以分享給他人參考運用。在 Customized RSS & Message RSS 一項，SWS 可對任何網頁自訂 RSS，並可分享他人，同時也能對 posts 產生 RSS Feed，讓 user 可隨時獲取最新的訊息，這兩項能力都是 GFC 與 Fleck 所缺乏的。在 Group 方面，SWS 與 Fleck 皆有 user 分群的能力，因而可對 post 實施讀取權限的管理，這在 GFC 則是沒有的。最後在 Relationship Graph 方面，SWS 可在當 Post 之中有 URL Link 時，用以顯示在同一群組中，具有相同討論標題的各網頁之間的連結關係圖，如此使用者可概念性的了解哪些網頁是被連結最多次，因而此網頁可能相對的會較具有重要性。

五、結論

本研究提出一個以 Firefox 為平台所開發出的系統，其能夠使得任何網站都能輕易的轉變成為一個社群網站，提供 RSS 與 Comment 及檔案分享等功能，並加入群組的機制，使得其上的資源可以具有權限管理的功能，同時提供 API，使得其他人能從基本的架構上，加以延伸出更多的社群網路應用。本研究的特點列舉如下：

- 針對 URL/XPath 的 Post 應用與管理
- 客製化的 RSS 產生器，目標包含 Web Pages 及 Comments。
- 社群網站的群組管理
- 針對使用者與群組的權限控管
- 使用者與群組的活動分享機制
- Obsolete Post 管理以及原始網頁段落取回機制
- 發展 Social Web API 以供其他人擴充出更多的社群網路應用

未來方向希望能提升以下的功能：

- 目前本研究的服務在 Microsoft Internet Explorer 上只能使用一部份，未來希望能開發出完整的 IE 版本，以使更多人能利用到本服務。
- OpenSocial API 是 Social Web 的一個趨勢，未來希望能納入此標準，以使得本研究的服務能更開放以吸引更多人參與開發，進而提供更多的應用。

參考文獻

- [1] Andreas Weigend, “Veränderungen in einer vernetzten Welt —die METRO Group im neuen Mitmach-Web” , 2. August 2007
- [2] 曾鈺絜, 「Web 2.0 與數位島嶼」, <http://content.ndap.org.tw/index/?p=688>, 2008-05-27
- [3] "Facebook", <http://zh.wikipedia.org/wiki/Facebook>
- [4] Alex Iskold, Richard MacManus, "Web Platform Primer - what's available via API?", http://www.readwriteweb.com/archives/web_platform_primer.php, August 22, 2006
- [5] “Add-on (Mozilla)” , [http://en.wikipedia.org/wiki/Add-on_\(Mozilla\)](http://en.wikipedia.org/wiki/Add-on_(Mozilla))
- [6] "Document Object Model", http://en.wikipedia.org/wiki/Document_Object_Model
- [7] “XUL” , <http://en.wikipedia.org/wiki/XUL>
- [8] “XUL” , <http://zh.wikipedia.org/wiki/XUL>
- [9] “XPath” , <http://en.wikipedia.org/wiki/XPath>
- [10] “XPath” , <http://zh.wikipedia.org/wiki/XPath>
- [11] “XPath 2.0” , http://en.wikipedia.org/wiki/XPath_2.0
- [12] “AJAX” , <http://www.cnix.org/wiki/AJAX&variant=zh-tw>
- [13] Jesse James Garrett, "Ajax: A New Approach to Web Applications", <http://adaptivepath.com/ideas/essays/archives/000385.php>, February 18, 2005
- [14] “Web service” , http://en.wikipedia.org/wiki/Web_service
- [15] “Web 服務” , <http://zh.wikipedia.org/wiki/Web%E6%9C%8D%E5%8A%A1>
- [16] “Social web” , http://en.wikipedia.org/wiki/Social_web
- [17] “JREx - The Java Browser Component” , <http://jrex.mozdev.org/>
- [18] “MozShot” , <http://mozshot.nemui.org/>
- [19] “Google Friend Connect” , <http://www.google.com/friendconnect/>
- [20] “Google Friend Connect” , http://en.wikipedia.org/wiki/Google_Friend_Connect
- [21] “Fleck” , <http://fleck.com/>