

國立交通大學

資訊科學與工程研究所

碩士論文



研究生：謝明志

指導教授：袁賢銘 教授

中華民國 九十七 年 六月

一個 Web 2.0 啟用的服務
A Web 2.0 Empowering Service

研究生：謝明志

Student：Ming-Chih Hsieh

指導教授：袁賢銘

Advisor：Shyan-Ming Yuan



June 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年六月

一個 Web 2.0 啟用的服務

研究生：謝明志

指導教授：袁賢銘

國立交通大學資訊科學與工程研究所

摘要

自從「Web 2.0」這個詞彙出現以來，已經帶來一個全新的資訊網路世代。討論如何去設計一個 Web 2.0 網站的文章更是如雨後春筍般的出現。然而，大多數的網站仍然維持著傳統的 Web 1.0 的型式，並沒有隨著這文章如此多數而跟著轉變。這是因為大部份 Web 1.0 網站的開發者並沒有足夠的能力和時間去將網站重新建構為 Web 2.0 的型式。重建一個網站所需要花費的心力並不小。這些人最需要的是一個簡單又有效能提升網站功能的方法。

本論文提出了一個全新的服務，Web 2.0 工具列，用以將傳統的 Web1.0 網站提升為 Web 2.0 網站。此 Web 2.0 工具列能將原本 Web 1.0 的網頁和附加的 Web 2.0 資訊結合在一起。如此一來，這些網站的開發者不需要去重新設計網站，也能讓網站獲得提昇，而網站的使用者也能使用到附加的 Web 2.0 功能。本篇論文中，目前已實作完成的 Web 2.0 功能如下：Tag、Comment、Chat Room 和 RSS。

A Web 2.0 Empowering Service

Student : Ming-Chih Hsieh

Advisor : Shyan-Ming Yuan

Institute of Computer Science and Engineering

National Chiao Tung University

Abstract

Since the term “Web 2.0” appears, a new generation of Web is coming. There are many articles talking about how to design a Web 2.0 website. However, the traditional Web 1.0 websites are still multitudinous. They do not transform to Web 2.0 as massive as these articles. This is because many of these traditional Web 1.0 website developers have not enough technique or time to rebuild their websites to Web 2.0. Rebuilding a website often takes a lot of efforts. A simple and effective strategy to upgrade their websites is needed.

The thesis attempts to provide a flexible framework, Web 2.0 Toolbar, to upgrade websites from Web 1.0 to Web 2.0. The Web 2.0 Toolbar combines the original Web 1.0 web page and additional Web 2.0 information together on the fly. Therefore, the website developers need not to rewrite their websites, and the users will be able to experience additional Web 2.0 functions. In the thesis, We have implemented tag 、comment 、chat room and RSS functionalities of Web 2.0.

Acknowledgement

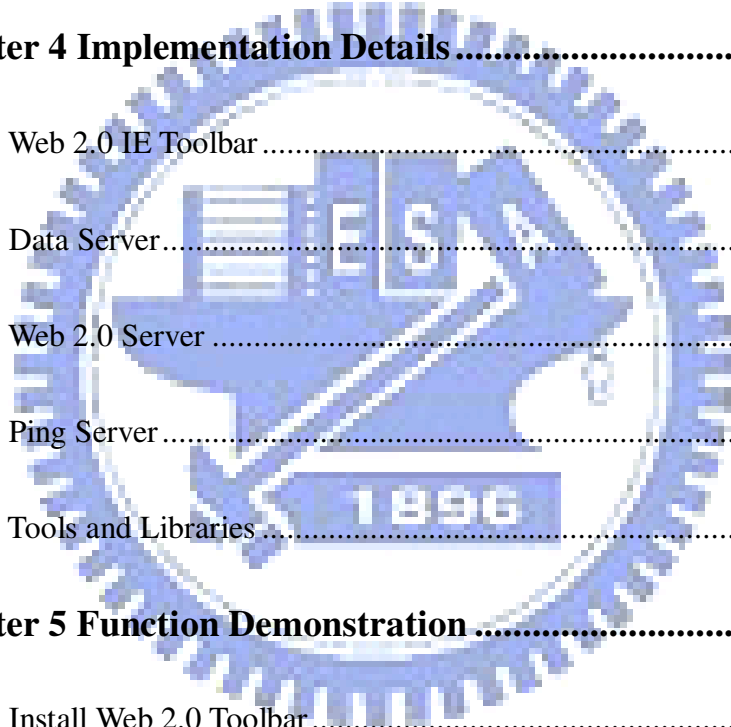
這篇論文的完成，首先要感謝我的老師袁賢銘教授的指導，這二年來一直給與我指導與研究方向的意見，並提供一個開明和歡樂的實驗室氣氛讓我們能自由發展研究。也感謝實驗室中的學長們葉秉哲、邱繼弘、吳瑞祥、鄭明俊、高永威、林家鋒等，在我研究的過程中給我不少的指導跟建議。感謝實驗室的同學簡士強、周鴻仁、林辰璞、林宜豐，在這二年來一起研究討論，讓我有許多寶貴的學習經驗。另外也感謝和我一起共同努力(玩樂?)的外宿室友們主任、ultru、子文和小晉，有你們的碩士生活才不致於過於無聊。也感謝許許多多支持我的朋友(這好像選舉台詞)。最後要感謝我的爸爸、媽媽、哥哥和姊姊讓我在求學期間能無後顧之憂，致力於學業，謹以這篇小小的學術成就來感謝您們的養育之恩。



Table of Contents

Acknowledgement	III
Table of Contents	IV
List of Figures	VII
List of Tables	IX
Chapter 1 Introduction.....	1
1.1 Preface	1
1.2 Motivation.....	2
1.3 Research Objectives.....	3
1.4 Outline of the thesis.....	5
Chapter 2 Background.....	6
2.1 Web 2.0.....	6
2.2 AJAX	7
2.3 RSS.....	8
2.4 IE Toolbar.....	8
2.5 Page Tailor.....	8
2.6 Related Works	9
2.6.1 Gabbly.....	9
2.6.2 Web 2.0 Proxy.....	10

2.7 Summary	11
Chapter 3 System Architecture	13
3.1 Overall Architecture.....	13
3.2 Web 2.0 Toolbar.....	14
3.3 Web 2.0 Management	15
3.4 Sequential Flow	17
Chapter 4 Implementation Details	19
4.1 Web 2.0 IE Toolbar	19
4.2 Data Server.....	22
4.3 Web 2.0 Server	23
4.4 Ping Server	24
4.5 Tools and Libraries	25
Chapter 5 Function Demonstration	26
5.1 Install Web 2.0 Toolbar	26
5.2 Upgrade to Web 2.0	29
5.3 Chat Room	31
5.4 RSS	32
5.5 Website Manager	33
Chapter 6 Conclusions and Future Works	35



6.1 Conclusions	35
6.2 Future Works	36
Chapter 7 Reference	38
Appendix	41
A.1 Toolbar Controller.....	41
A.2 Web 2.0 viewer	44



List of Figures

Figure 1-1: Web pages about Web 2.0 on Google Search	1
Figure 1-2: Browser Market Share in April 2008	4
Figure 2-1: A Tag cloud presenting some of the themes of Web 2.0.....	6
Figure 2-2: An Example of Gabbly	10
Figure 2-3: An Example of Web 2.0 Proxy.....	11
Figure 3-1 System Architecture	13
Figure 3-2: Web 2.0 Management.....	15
Figure 3-3: A flow of using Web 2.0 Toolbar	17
Figure 4-1: Web 2.0 IE Toolbar framework.....	20
Figure 4-2: A C++ code of running Web 2.0 viewer.....	21
Figure 4-3: Original web page appending Web 2.0 data	21
Figure 4-4: Database schema	23
Figure 4-5: XMLHttpRequest Code.....	24
Figure 5-1: Install Web 2.0 Toolbar.....	26
Figure 5-2: Original Web 1.0 Web Page.....	27
Figure 5-3: After Installing Web 2.0 Toolbar.....	28
Figure 5-4: Using Web 2.0 Toolbar	29
Figure 5-5: Web page Using Web 2.0 Toolbar with tags.....	30

Figure 5-6: Web page Using Web 2.0 Toolbar with comments 31

Figure 5-7: Web page Using Web 2.0 Toolbar with Chat Room..... 32

Figure 5-8: Web 2.0 Toolbar with RSS Manager 33

Figure 5-9: Web 2.0 Toolbar User Guide..... 34

Figure 5-10: Website Manager for Website Developer 34



List of Tables

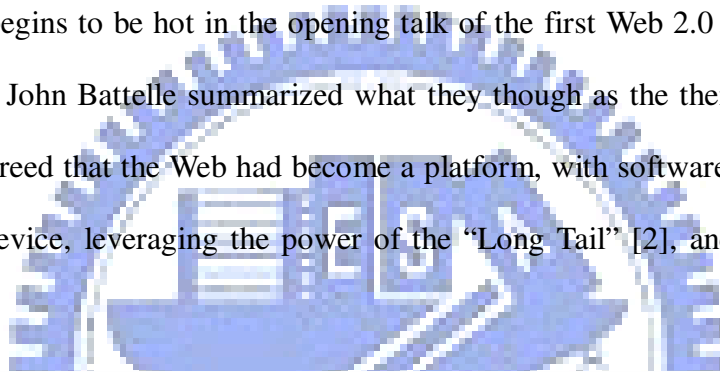
Table 4-1: Tools and libraries used..... 25



Chapter 1 Introduction

1.1 Preface

The term “Web 2.0” has been discussed a lot in recent years [1]. About the term “Web 2.0”, it is not a technical noun but more suitable to be a collective concept describing the technical features and social behaviors of some of the famous Web 2.0 websites. It begins to be hot in the opening talk of the first Web 2.0 conference. Tim O’Reilly and John Battelle summarized what they thought as the themes of Web 2.0. They both agreed that the Web had become a platform, with software above the level of a single device, leveraging the power of the “Long Tail” [2], and with data as a driving force.



Web 2.0]			進階搜尋
web 2.0	70,500,000 結果		使用偏好
web 2.0 定義	921,000 結果		語言選項
搜 web 2.0 wiki	3,250,000 結果	頁	
web 2.0 應用	869,000 結果		
web 2.0 網站	864,000 結果		
web 2.0 logo	4,190,000 結果		
web 2.0 definition	1,010,000 結果		
web 2.0 ppt	1,070,000 結果		
web 2.0 design	22,800,000 結果		
web 2.0 expo	1,010,000 結果		關閉

[廣告服務](#) - [Google 完全手冊](#) - [Google.com in English](#)

©2008 Google

Figure 1-1: Web pages about Web 2.0 on Google Search

Web 2.0 has become a very hot subject, there are about seventy-million web pages related to “Web 2.0” on Google search engine. Among them, there are almost twenty-three million results which discuss “Web 2.0 design” shown in Figure 1-1. How to build a Web 2.0 website is a large research which many website designers want to study.

1.2 Motivation

Web 2.0 has numerous definitions, but one of them obtains the identification of most people. That is information sharing and the content of website can be changed by every user’s participation. One of the most famous examples is WIKIPEDIA [3], which is a free, open and on-line encyclopedia. Every user can edit the content of the on-line encyclopedia. Another famous Web 2.0 website is del.icio.us [4]. It is a bookmarkers sharing website, every user can make bookmarkers for articles and share their bookmarkers to others. Web 2.0 websites often provide functionalities permitting users to share their knowledge on it.

Unfortunately, the mass of researches all put attentions on how to take advantage of technique for building a powerful Web 2.0 website. The thesis focuses on how to provide a simple method allowing the Web 1.0 websites to achieve the functionality of information sharing. The method must be easy enough to allow the Web 1.0 website developers to handle it lightly. In addition, the interface of the functionality should be very friendly, and it must combine to the original web page closely. There are three ways to construct a Web 2.0 website. One is to build a whole new Web 2.0 website from the scratch. Another way is to rewrite the page of Web 1.0 website. The third is to enhance a traditional Web 1.0 website to Web 2.0. Both of the first two ways often take a lot of time and need some new skills about Web 2.0. In the thesis,

we focus on the third method. Until now, there are few effective methods to achieve the goal. Hence, the thesis proposes an easy way to upgrade those Web 1.0 websites without rebuilding them. The only thing they have to do is to ask their users to install Web2.0 Toolbar on their browsers. The method not only supports a very simple way but also satisfies the needs of both web developers and users.

1.3 Research Objectives

The objectives of our framework can be summarized as the following four parts:

- **Simple and effective way to upgrade websites**

The most important thing for these Web 1.0 website developers is that the strategy to enhance their websites must be very simple and effective. These developers do not want to learn any more technique or spend more time to do some changes on their websites and can upgrade their websites outstandingly. We must provide the using of the strategy as easy as possible.

- **User friendly interface**

This thesis provides a strategy to combine the original Web 1.0 web page and additional Web 2.0 services. Integrating two difference objects may cause a problem – inharmonious sense, so how to design a friendly interface for users is a major purpose. The system must be not only simply for website developers but also easy for website users.

- **Support for mainstream browser**

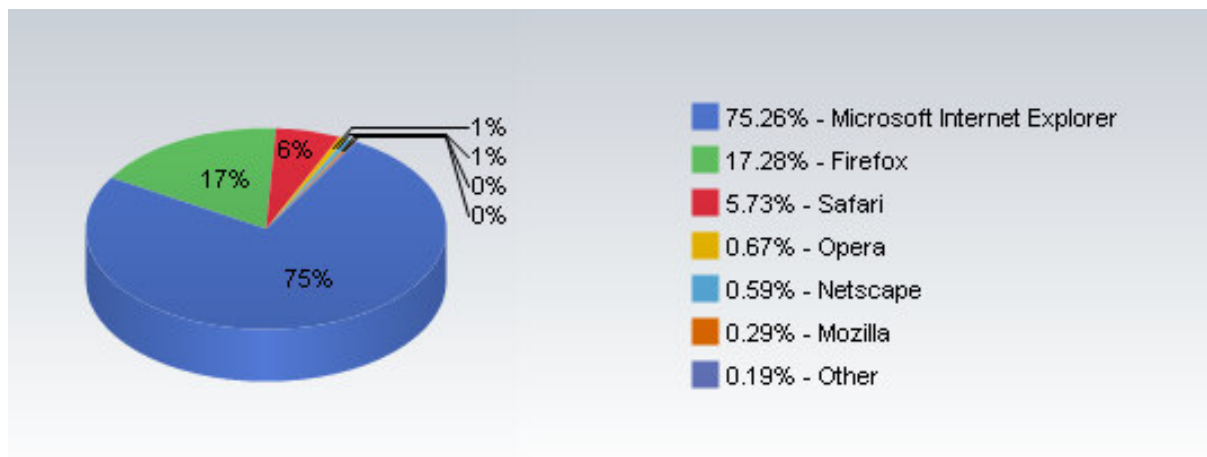


Figure 1-2: Browser Market Share in April 2008

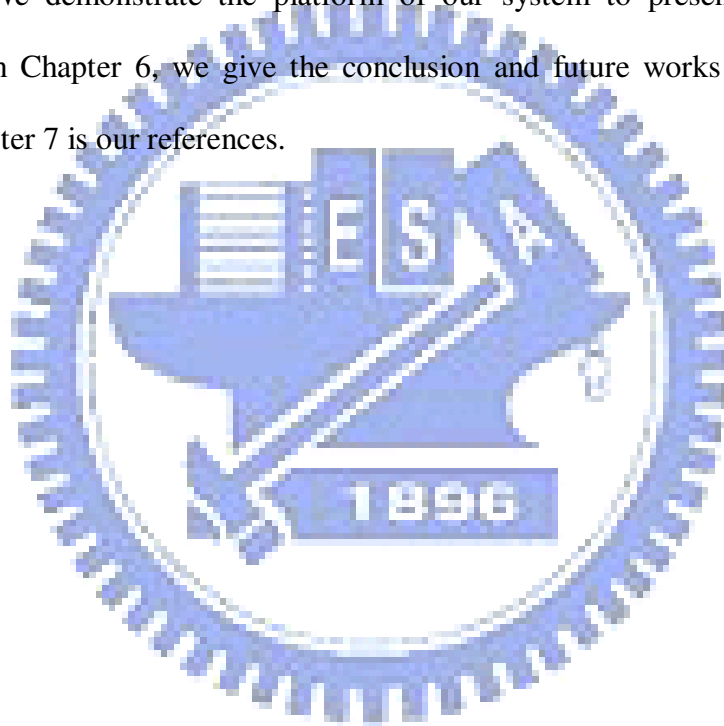
According to the browser market share survey [5] (Figure 1-2), Microsoft Internet Explorer is still far and away the most dominant browser on the Web, with 75.26 % usage market share, and Firefox has increased its share to 17.28%, and the other alternatives, such as Safari, Netscape, Opera, and Mozilla, occupy the remaining share. The toolbar should work with at the most popular browser - Microsoft Internet Explorer.

- **Easy to develop and expand**

The developing of Web 2.0 is still continued. There may appear more and more practical functions in the future. Although our system can not achieve all the services of Web 2.0, but also must provide a flexible platform to develop and expand. In order to achieve the goal, we must separate our system and the original websites as much as possible. It also responds to the first objective, to make the architecture simple and effective.

1.4 Outline of the thesis

The thesis contains seven chapters. In Chapter 2, we discuss the background of Web 2.0 websites development and the IE toolbar and then introduce the other researches to build Web 2.0 websites. In Chapter 3, we propose the architecture of our system, and discuss the purpose and responsibility of each component. For deep understanding, we describe the implementation details and the design method in Chapter 4. We demonstrate the platform of our system to present the results in Chapter 5. In Chapter 6, we give the conclusion and future works for our system. Finally, Chapter 7 is our references.



Chapter 2 Background

2.1 Web 2.0

The tag cloud pictured below constructed by Markus Angemeler on November 11, 2005, sums up some of the themes of Web 2.0, with example-sites & services attached. These white terms on the left-top part of the cloud show the concept of information sharing, and this thesis is base on it, too. Web 2.0 websites should provide some mechanisms which allow users to participate.



Figure 2-1: A Tag cloud presenting some of the themes of Web 2.0

The white terms on the right-top part of the cloud show the concept of economy. Web 2.0 website should build more interactive facilities to provide "Network as platform" computing, allowing users to run software-applications entirely through a

browser. The new technique, Ajax satisfies some of the request and we will detail it in the next section.

The white terms on the left-down part of the cloud show the concept of convergence. Videos and audios are converged and shared on some website, like Odeo [6] and YouTube [7]. A very useful technique, RSS appeared at that moment, and it supplies a subscript/publish mechanism to allow users can get the notice when a website which he subscribed has an update. We elaborated it in the following section.

There has been a dramatic proliferation of researches concerned with Web 2.0. For examples, how to combine the powerful Web 2.0 and multimedia [8] and the relationship of the developing between Web 2.0 and semantic web [9] are the most popular subjects. Although Web 2.0 has many positive effects, it also brings new problems. The gallop developing of Web design supports more powerful functions and resplendent interactions, but the change in paradigm brings new challenges to people with disabilities. A discussion of how to build an accessible web [10], [11], [12] also becomes an important subject. These subjects are tip of the iceberg, beside there is a mass of subjects about Web 2.0.

2.2 AJAX

Ajax (Asynchronous Javascript and XML) is a term defined by Jesse James Garrett [13] referring to the combination of techniques, including the Javascript XMLHttpRequest (XHR) object, DOM manipulation and XHTML, involved in the development of interactive web applications. The core of Ajax is XHR [14] which enables Javascript embedded in a webpage to issue asynchronous HTTP requests without the needs to refresh the whole page, thus resulting in a fluid user experience.

2.3 RSS

RSS (RDF Site Summary) is originally created by Netscape to describe news stories in RDF (Resource Description Framework), which in turn is defined using XML (eXtensible Markup Language). That version is known as RSS 0.9. After minor modifications to remove RDF elements to make it 0.91, it split into two branches, the RDF branch and the simple branch. The RDF branch is advocated by RSS-DEV Group, where RSS means Rich Site Summary (RSS 1.0). The simple branch is advocated by famous blogger Dave Winer, where RSS means Really Simple Syndication (RSS 0.92, 0.93, 0.94, and 2.0) [15]. No matter what format it uses, it is composed of the same things conceptually: a header section describing the whole feed and a list of entries having similar attributes, such as unique identifier, title, description, published date and time to support the mechanism of subscribe/publish.

2.4 IE Toolbar

IE (Internet Explorer) Toolbar is a plug-in program embedded in IE. It often consists of some buttons and supports some functions. Many websites have already developed and issued their own IE Toolbar to help users to use their services more conveniently. For example, Google issues its own Google Toolbar [16] and Yahoo issues Yahoo Toolbar [17], too.

2.5 Page Tailor

Page Tailor is one component of the system, Web Page Tailoring Tool for Mobile Devices [18] and can be downloaded and executed in a user's browser when he/she is

to use our Web 2.0 Toolbar. It provides some visual manipulations for users to help them specify their preferences about a Web page.

2.6 Related Works

The topic is still lack of research, so only two related works are listed, which are Gabbly and Web 2.0 Proxy. The architecture of these two works is not the same but very similar, and we will further describe in the following thesis.

Gabbly

Gabbly [19] appeared from the march 2006, and it is a Web 2.0 chat website. Gabbly supports one main function that is everyone can talk to each other which are browsing the same web page. Using Gabbly is very easily, everyone just needs to append the keyword “<http://gabbly.com/>” before the URL of a web page. Then the user can join the chat room corresponding to the web page. On another aspect, Gabbly supports a chat function to all the web pages. The figure 2-2 shows an example of Gabbly.

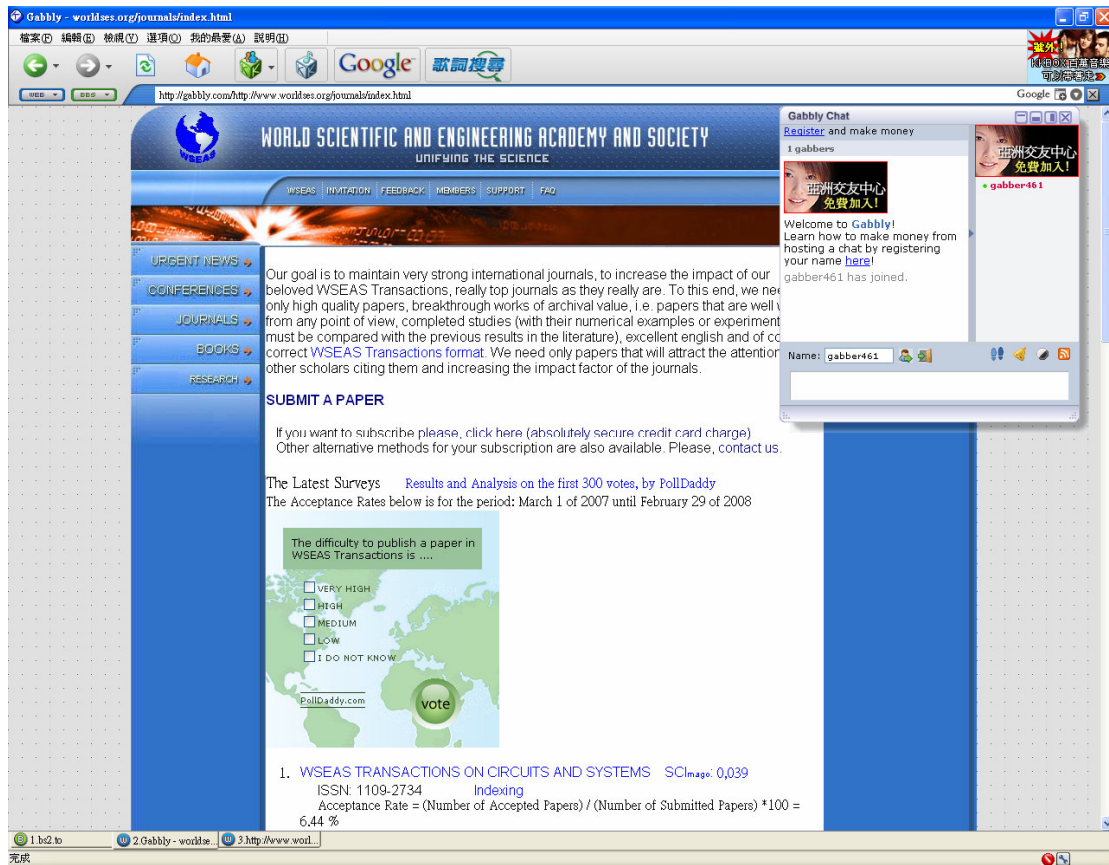


Figure 2-2: An Example of Gabbly

2.6.1 Web 2.0 Proxy

The Web 2.0 Proxy [20] is the other strategy which we tried to build Web 2.0 websites before. By using the Web 2.0 Proxy, the Web 1.0 website developers can append two services, tag and comment to their websites. But this strategy will face to some problems that can not be solved easily.

When User Browser sends a HTTP request, Web 2.0 Proxy will receive the request. Then it not only requests original web page form the Original Web Server but also requests the information which related to the original web page from the Configuration Manager. Finally Web 2.0 Proxy sends the combined information to the User Browser. Hence, User Browser receives a new version of Web 2.0 web page

which includes original contents and additional information (which indicates tags and comments in Web 2.0 Proxy implementation). The figure 2-3 shows the example of the Web 2.0 Proxy.

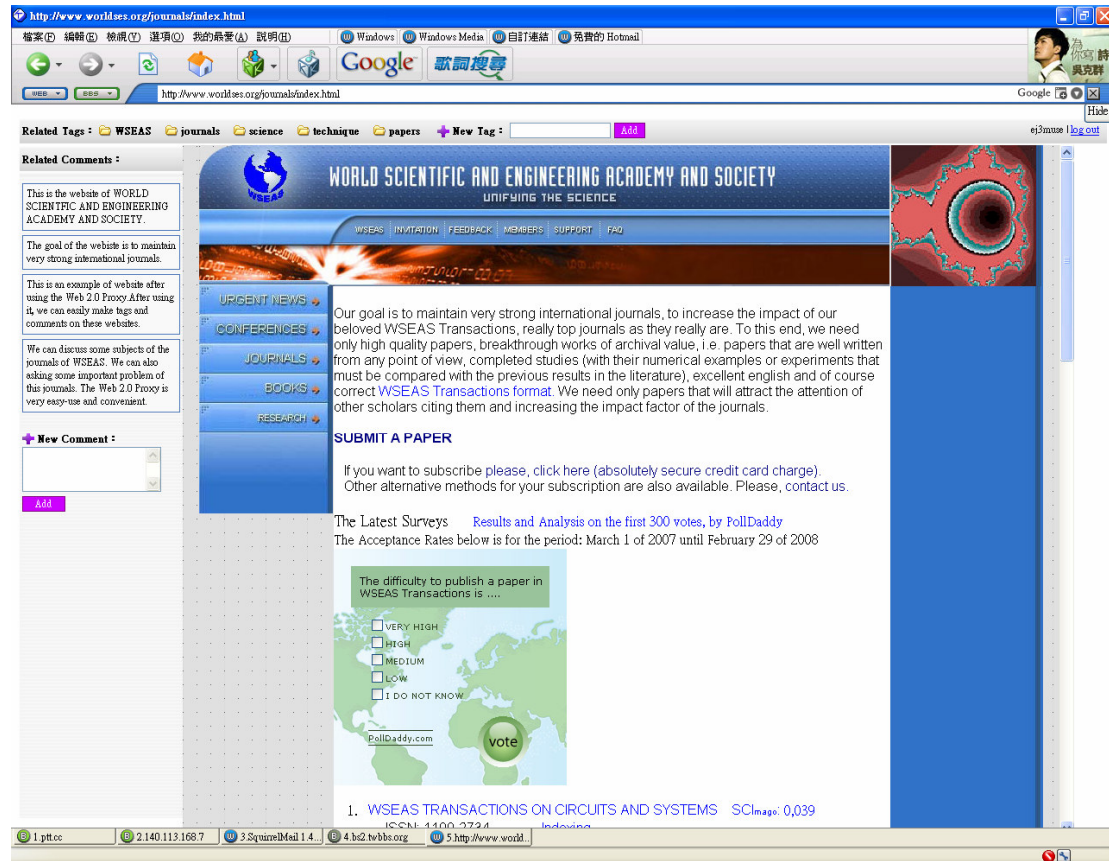


Figure 2-3: An Example of Web 2.0 Proxy

2.7 Summary

By sections 2.1, we make efforts to develop a way to help the website developers to upgrade their websites with some information sharing mechanisms. By way of combining Web 2.0 and IE Toolbar, we build a new framework, Web 2.0 IE Toolbar to achieve the goal. Besides we apply Page Tailor in our system to display the additional information and functions and use AJAX, which supports our system to get data without refreshing all webpage to develop our system. Our system also supports a

simple RSS reader. The thesis tries to help original Web 1.0 website to upgrade to Web 2.0 with a simple and effective way.



Chapter 3 System Architecture

In this chapter, we present the overall architecture of our system that includes Web 2.0 Toolbar and Web 2.0 Management and then introduce each component in our system. Additionally, in the last subsection, we describe the sequential flow in our system.

3.1 Overall Architecture



Figure 3-1 System Architecture

In regular HTTP transaction [21], the User Browser sends a HTTP request to the Original Web Server, and the Original Web Server sends a HTTP response to the User Browser. Then the User Browser will receive the original web page content. Base on

the architecture of the traditional HTTP transaction, we append another two components: Web 2.0 Toolbar and Web 2.0 Management to become the new architecture. Figure 3-1 illustrates the new architecture we proposed with the communications of every component.

In the aspect of the user, when the User Browser sends a HTTP request to the Original Web Server, Web 2.0 Toolbar can obtain the URL of this HTTP request and ask the information which related to the original web page from Web 2.0 Management. By this way the User Browser not only receives the original web page form the Original Web Server but also gets the information related to this web page. Hence, the user will see a new view of Web 2.0 web pages which include original contents and additional services (which indicate tags 、 comments 、 chat room and RSS).

In the aspect of the Website Developer, the Original Website and additional services seem be built together but can be managed separately. The Website Developer registers to Web 2.0 Management and takes over the information (tags and comments) related their websites. Web 2.0 Management supports Website Manager to allow these Website Developers to create and delete tags/comments.

The Web 2.0 Toolbar framework has two main components: Web 2.0 Toolbar and Web 2.0 Management. The detail descriptions of them are in following sessions.

3.2 Web 2.0 Toolbar

The relationship of the User IE Browser and the Original Web Server is a client/server model, and the one of Web 2.0 Toolbar and Web 2.0 Management is similar to another. Web 2.0 Toolbar not only plays the role of client in the mode but also acts as a coordinator between User IE Browser and itself. There are three major

works which Web 2.0 Toolbar deals with. First it demands the URL of the web page which User IE Browser is handling with. Second it communicates with Web 2.0 Management. Send the URL from User IE Browser as parameter to Web 2.0 Management and take the relevant information back. Third Web 2.0 Toolbar integrates the original web page and the additional relevant information together. Accurately to say, Web 2.0 Toolbar appends the additional data to the original html page. The detail implementation will be interpreted in Chapter 5.

3.3 Web 2.0 Management

Web 2.0 Management contains four subcomponents: Web 2.0 Server, Data Server, Manager Server and Ping Server. All of them are illustrated in Figure 3-2.

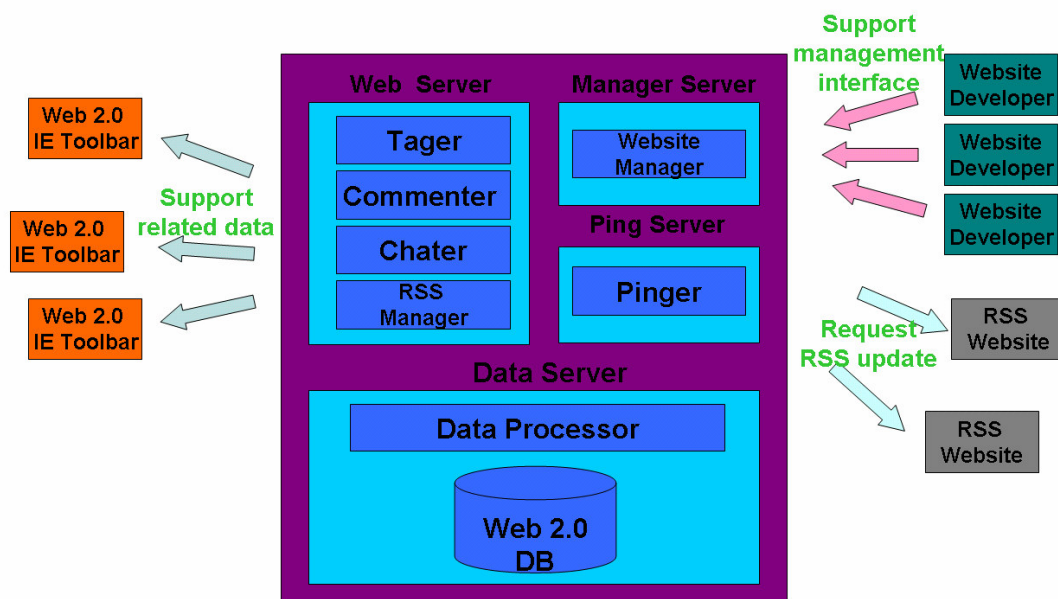


Figure 3-2: Web 2.0 Management

Data Server contains two parts, Web 2.0 DB and Data Processor. Web 2.0 DB is a database to store all entries of the web page and other relevant information. It supports all data which the system needs. Data Processor has the knowledge of both the object representation and the database schema. It supports an interface to access Web 2.0 DB for the other three subcomponents. Data Processor and Web 2.0 DB have taken sense of responsibility to manage all data.

Web 2.0 server is a HTTP web server, and it consists of four parts, Tager, Commenter, Chater and RSS Manager. We will introduce them one by one. Tager supplies three functions which are creating, rating tags for each web page and viewing these tags and their rates of each web page. Tager processes the related data by Data Processor and delivers them to Web 2.0 Toolbar. The work of Commenter is quite similar as Tager, but the only different point is that Commenter processes comments not tags. Chat supports only one function. That is the users who are viewing the same web page can talk to each other. That means every web page additionally supports a chat room function, and everyone can discuss the content of the web page very friendly. RSS Manager is a very simple RSS reader. It allows the user who has already registered to subscribe/delete some RSS feeds. Once these feeds have updated, RSS Manager will send the notices to these users.

Ping Server was built for supporting RSS Manager. It continues to check if these RSS feeds have an update or not. Once an update happened, it will modify these related data in Web 2.0 DB by Data Processor.

Manager Server supports the Website Developers who have already registered to our system to manage the related information. Website Developers can create/delete the tags and comments related their web pages. Building more suitable data to their websites, and reduce these incorrect or abusive data.

3.4 Sequential Flow

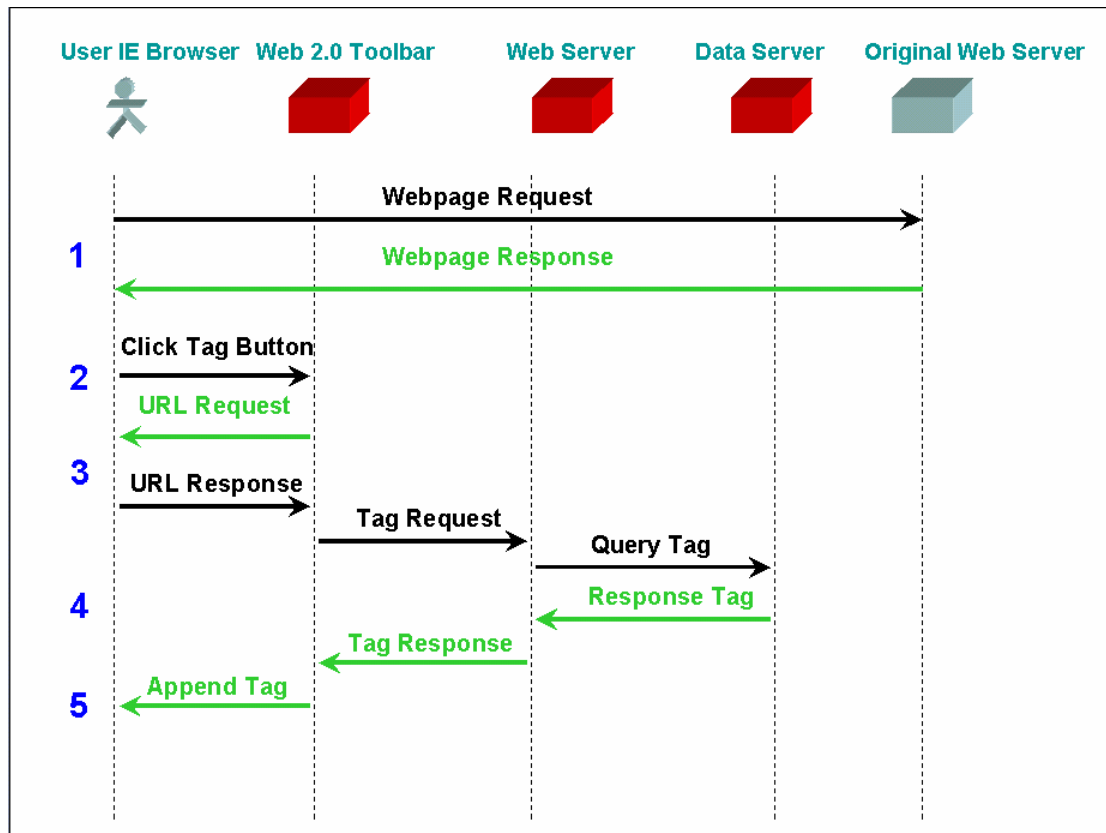


Figure 3-3: A flow of using Web 2.0 Toolbar

The Figure 3-3 illustrates the sequential flow which a user is using Web 2.0 Toolbar. The scenario contains five steps, and we will explain them as follow:

Step1: A user types a web page URL on the IE Browser, and the User Browser sends a HTTP request to Original Web Server and gets the Web page response content. Step2: the user clicks the tag button to using tag service. Step3: Web 2.0 Toolbar sends a request to query the URL of the web page and obtains the response value. Step4: Web 2.0 Toolbar sends the HTTP request of the tag web page to Web 2.0 Server. Then Web 2.0 Server queries the related tag data from Data Server, and responses the tag web page to Web 2.0 Toolbar. Finally, Web 2.0 Toolbar appends the

tag web content into User IE Browser. The other services are similar to this flow. The difference between them is that it returns the corresponding web content rather than tag web content.



Chapter 4 Implementation Details

We describe the detail implementation of our framework in this chapter, especially on how Web 2.0 Toolbar combines original web page content and additional information. Besides other related technique details will also be explained.

4.1 Web 2.0 IE Toolbar

Web 2.0 IE Toolbar can be divided into two parts, IE Toolbar Controller and Web 2.0 viewer. IE Toolbar Controller is installed in users' computer, but Web 2.0 viewer is put on Web 2.0 Server. Every time a user uses the toolbar, and then Toolbar Controller will download Web 2.0 viewer from Web 2.0 server to execute, shown as Figure 4-1. The mechanism brings a benefit that is even there are some changes on Web 2.0 Server, users do not need to reinstall Web 2.0 Toolbar program. Then we will describe IE Toolbar and Web 2.0 viewer as following and more detail source code will be placed in the Appendix.



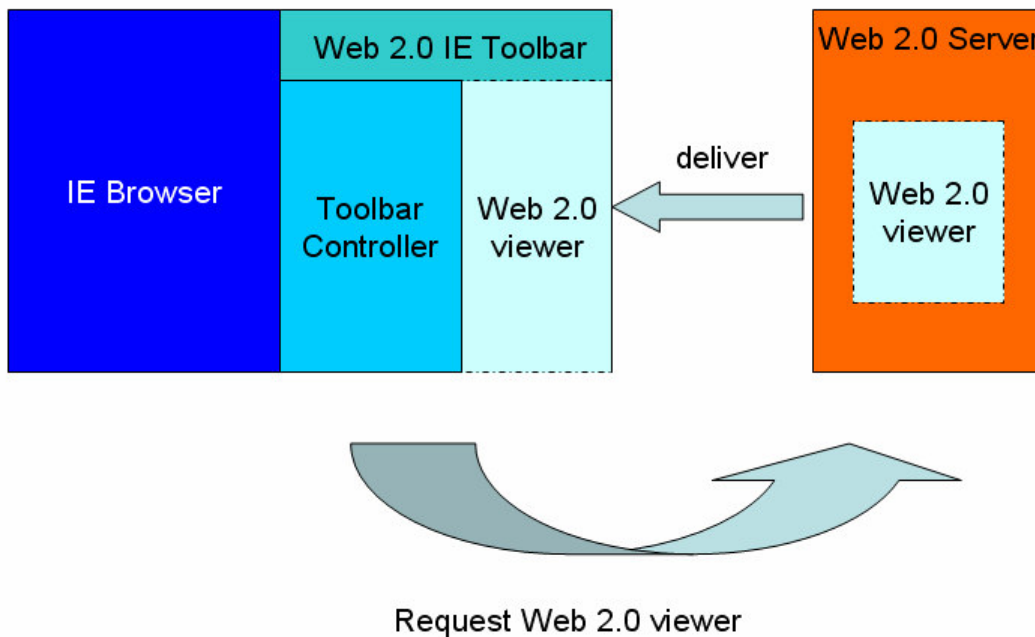


Figure 4-1: Web 2.0 IE Toolbar framework

Toolbar Controller is written by C++ language and built according to a network technical article [22] in website “The CODE PROJECT”. Toolbar Controller consists of a COM component supporting IDeskband and a few other necessary interfaces for which IE looks for when loading registered toolbars, explorer bars and deskbands. The RBDeskband ATL Object Wizard provides most of the framework. First we create a VC6.0 project, a new COM object to house our toolbar, and a few CWindowImpl classes using the CWindowImpl ATL Object Wizard. Then connecting these parts together we will produce the Toolbar Controller. Visually the toolbar consists of a word and four buttons on it. The most important part of Toolbar Controller is that when clicking the button, there will detonate an event to run one Javascript code to download and start the Web 2.0 viewer, shown on Figure 4-2.

```

if (m_pBrowser->get_Document(&spDispatch) == S_OK){
    CComQIPtr<IHTMLDocument2> spHTMLDoc = spDispatch;
    CComQIPtr<IHTMLWindow2> pHTMLWnd;

    _bstr_t strjs="javascript:void(b=document.getElementsByTagName('body')[0].
        +\"appendChild(document.createElement('script')));\"
        +\"void(b.type='text/javascript');void(b.id='tk_bookmarklet');\"
        +\"void(b.src='http://140.113.88.240:8080/web2project/javascrpts/loader.js');\";

    if(spHTMLDoc!=NULL){
        spHTMLDoc->get_parentWindow(&pHTMLWnd);
        if(pHTMLWnd!=NULL){
            CComBSTR bstrlan=SysAllocString(L\"javascript\");
            VARIANT varRet;
            pHTMLWnd->execScript(strjs,bstrlan,&varRet);
        }
    }
}

```

Figure 4-2: 4-2 A C++ code of running Web 2.0 viewer

Web 2.0 viewer is written by Javascript and CSS language and built using “Web Page Tailor”. After starting, Web 2.0 viewer will catch corresponding JSP file (tag.jsp、comment.jsp、chat.jsp or rss.jsp) from Web 2.0 Server. Besides it also requests the data relevant the web pages which users are viewing. Finally, it appends these information into the original web page by Javascript DOM and <div>, shows in Figure 4-3.

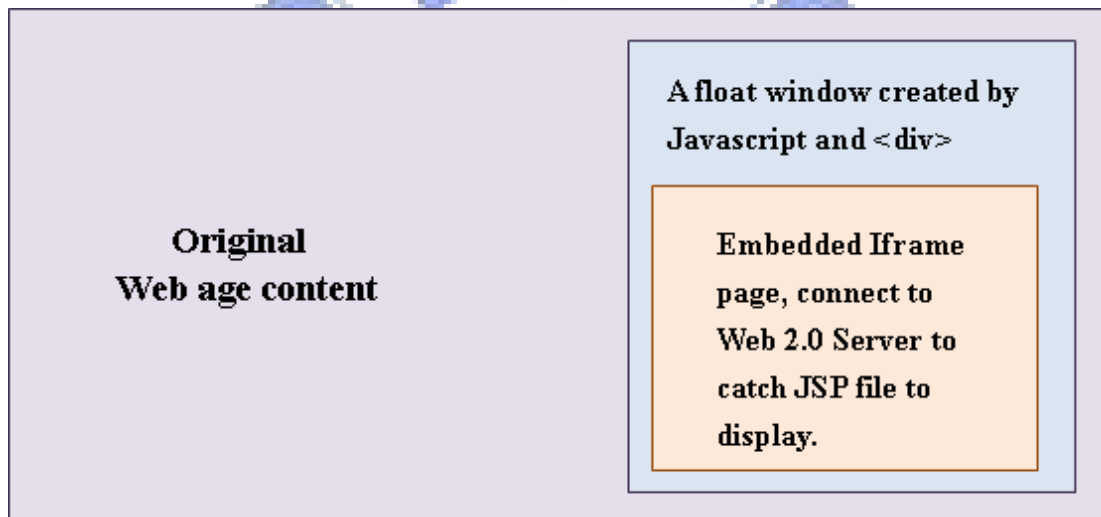


Figure 4-3: Original web page appending Web 2.0 data

4.2 Data Server

We use MySQL [23] as our Database server. The DB server contains about 13 tables which store web page data 、related tag data 、related comment data 、user data and data of RSS feed. The database schema and all tables are shown in Figure 4-4. Because each web page may own many tags and comments, the relationship between the table webpage and tag/comment must be one to many. We append two table webpage_tag and webpage_comment to record that which tag/comment should belong to the corresponding webpage data. Each RSS may own many users to subscript it, so the relationship between the table feed and user must be one to many. Finally, the table subscription contains the records that users subscript which RSS.

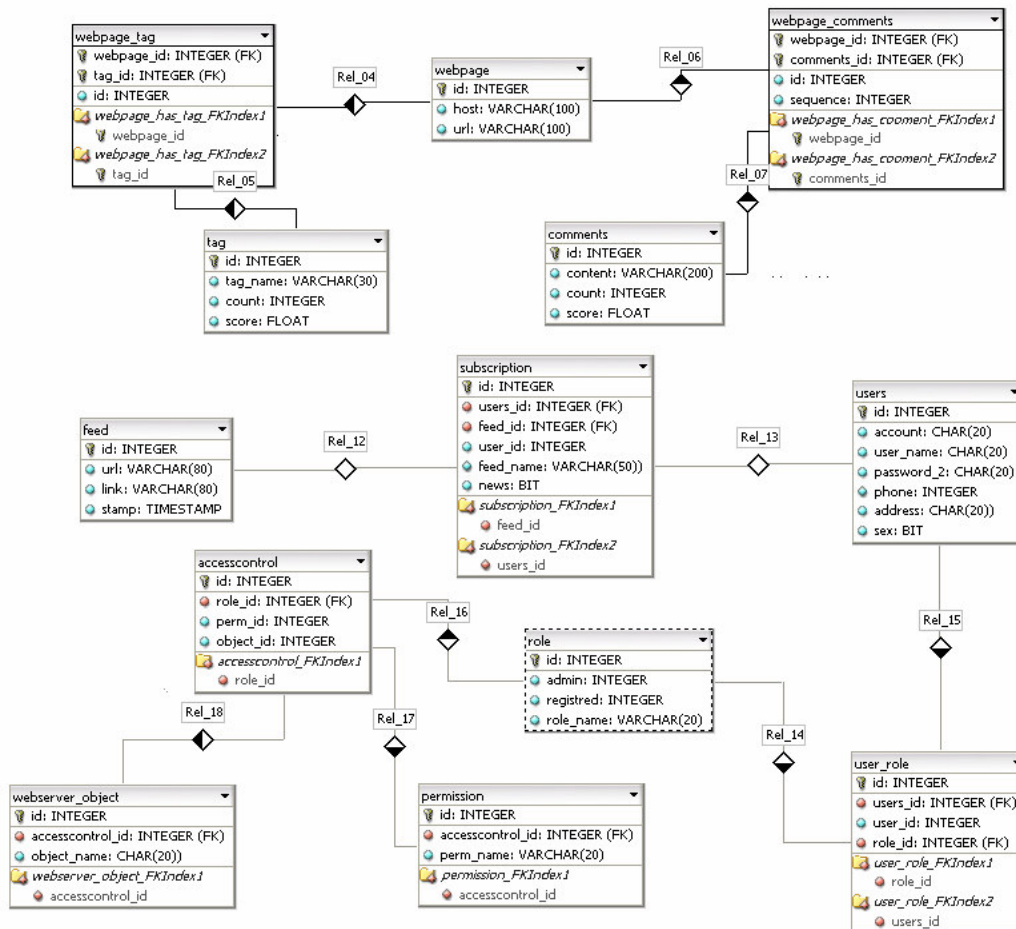


Figure 4-4: Database schema

Data Processor is written by Java with object orient model. Almost each table is corresponding to one class, and contains related attributes and functions to store , delete or modify these data. Web 2.0 Server uses Data Processor to access Web 2.0 DB very friendly by the API interface.

4.3 Web 2.0 Server

We use Apache Tomcat [24] as our web server, and write Web 2.0 services by JSP language. It contains four web pages: tag page , comment page , chat page and RSS page. The four web pages support corresponding to four functions, tag , comment , chat room and RSS, and they need one URL as parameter to ask corresponding data.. Tager , Commenter and RSS Manger request data from DB server, and add some data which users make to DB server as general behavior of using web pages.

Most importantly, All of them are designed using AJAX, they use XMLHttpRequest to replace traditional HTML form to send and get data from web server. XHR enables Javascript embedded in a webpage to issue asynchronous HTTP requests without refreshing the whole page, thus resulting in a fluid user experience. Figure 4-5 shows the example of XMLHttpRequest.

```

function add_tag(){
    if(inner_login.value!="1"){alert("Login First!");}
    else{
        var url="<%=site_url%><%=tag_session_var%>?"
            +"url=<%=request.getParameter("url")%>&data="
            +tag_data.value;
        if (window.XMLHttpRequest) {
            req = new XMLHttpRequest();
            req.onreadystatechange = addTagCallback;
            req.open("GET", url, true);
            req.send(null);
        } else if (window.ActiveXObject) {
            isIE = true;
            req = new ActiveXObject ("Microsoft.XMLHTTP");
            if (req) {
                req.onreadystatechange = addTagCallback;
                req.open ("GET", url, true);
                req.send();
            }
        }
    }
}

function addTagCallback(){
    if (req.readyState == 4) {
        if (req.status == 200) {
            var response = req.responseText;
            response=response.substring(27,response.length);
            if(response.substring(0,1)=="1"){
                var id_start = response.indexOf("@@");
                response2=response.substring(1,id_start);
                var response3 = response.substring(id_start+2,response.length);
                tmp_div= document.createElement("div");
                tmp_div.innerHTML="<div style='border-style: dotted; border-width:"
                +" lp; background-color: #FFFFFF'>&nbsp;&nbsp;&nbsp;<img name='star_atag'"
                +" src='<%=site_url%>images/star0.bmp' align='absbottom'>&nbsp;&nbsp;&nbsp;"
                +" &nbsp;&nbsp;<input type='checkbox' name='tag_checkbox' value='"+response3+"'>"
                +"&nbsp;&nbsp;<input type='hidden' name='tag_count' value='0' >"
                +"<input type='hidden' name='tag_score' value='0' >"+response2+"</div>";
                document.getElementById("show_tag").appendChild(tmp_div);
            }
        }
    }
}

```

Figure 4-5: XMLHttpRequest Code

The design of chat page is not the same as other web page. We use DWR (Direct Web Remoting) [25] to build this service. DWR allows Javascript in a browser to interact with Java on a server and helps us manipulate web pages with the results. Using DWR, we can aware that which user view the web page 、which user send a message and respond the message to other users. Finally, we use URL to separate the users to different chat rooms.

4.4 Ping Server

We built the Ping Server using Java and ROME [26] which is a set of open source Java tools for parsing, generating and publishing RSS and Atom feeds. Ping Server is a simple RSS update monitor, and all the works which it handles is to grab

the all records of feed table, and then compares the timestamp of each feed to check if there is an update happened. Once there is an update, Ping Server will update the link to latest one and notice all the users who subscript this feed. Ping Server repeats the work over and over.

4.5 Tools and Libraries

Many open source tools and libraries are used in our system. They are listed below:

Name	Usage	License
C++ ATL	IE Toolbar	Microsoft
Javascript	Web 2.0 functions	Netscape
Web Page Tailor	Web 2.0 viewer	DCSLab
JSP	Web pages	Sun
JDBC	Process Data	Sun
ROME (JAVA)	Process RSS	Sun
MySQL	Database	Sun
Apache Tomcat	Web Server	Apache
DWR	Chat Room	TIBCO

Table 4-1: Tools and libraries used

Chapter 5 Function Demonstration

To demonstrate the Web 2.0 Toolbar framework we propose, we upgrade an Original Web 1.0 web page to Web 2.0 as an example and present the all scenario.

5.1 Install Web 2.0 Toolbar

Before using Web 2.0 Toolbar, the users should install Web 2.0 Toolbar in their computers (Figure 5-1). Then Web 2.0 Toolbar can serve these users.

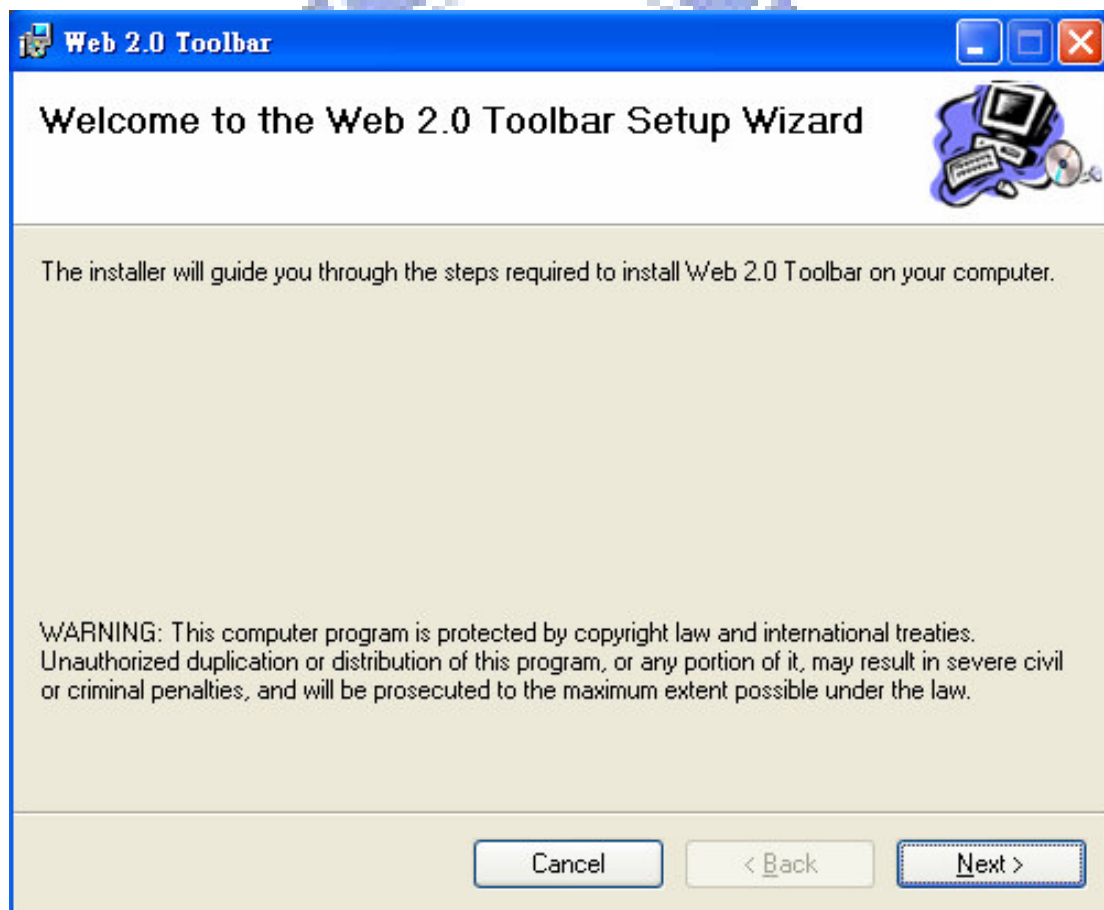


Figure 5-1: Install Web 2.0 Toolbar

Figure 5-2 shows an Original Web 1.0 web page without installing Web 2.0 Toolbar. Figure 5-3 shows the web page after installing Web 2.0 Toolbar. It appends four service buttons on the right-top region of IE.

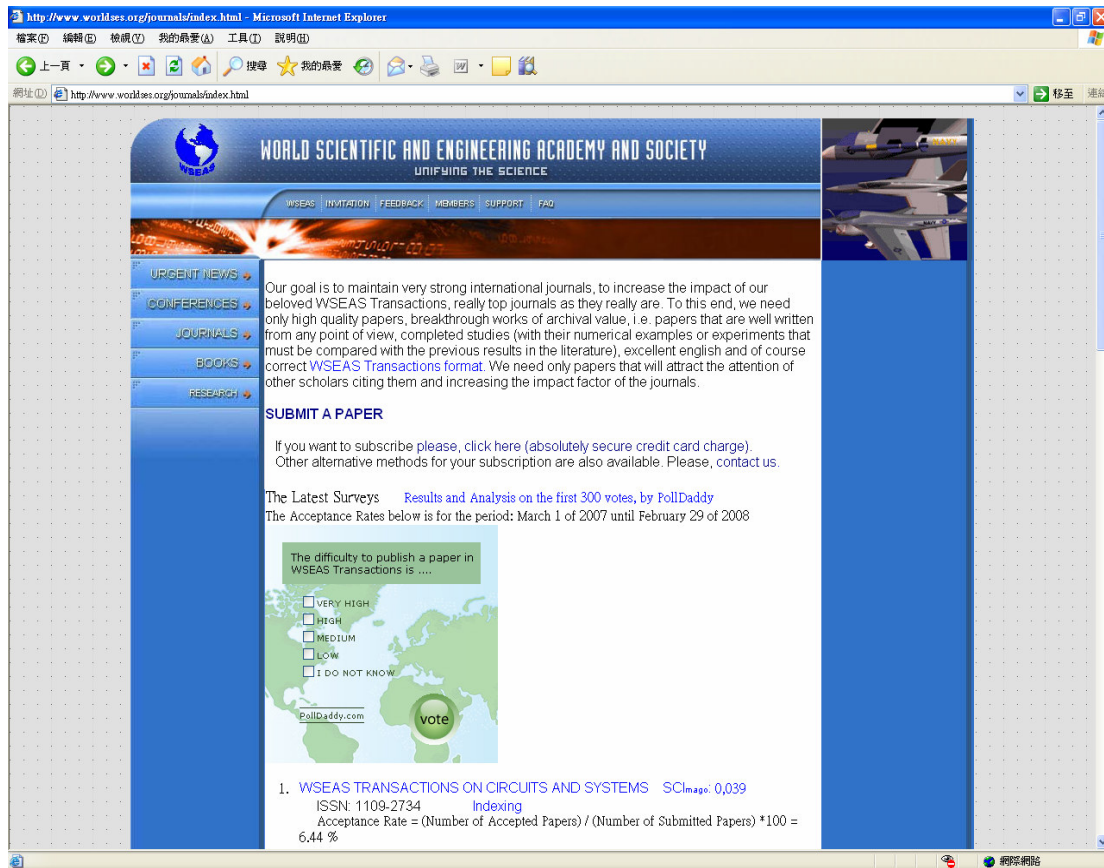


Figure 5-2: Original Web 1.0 Web Page

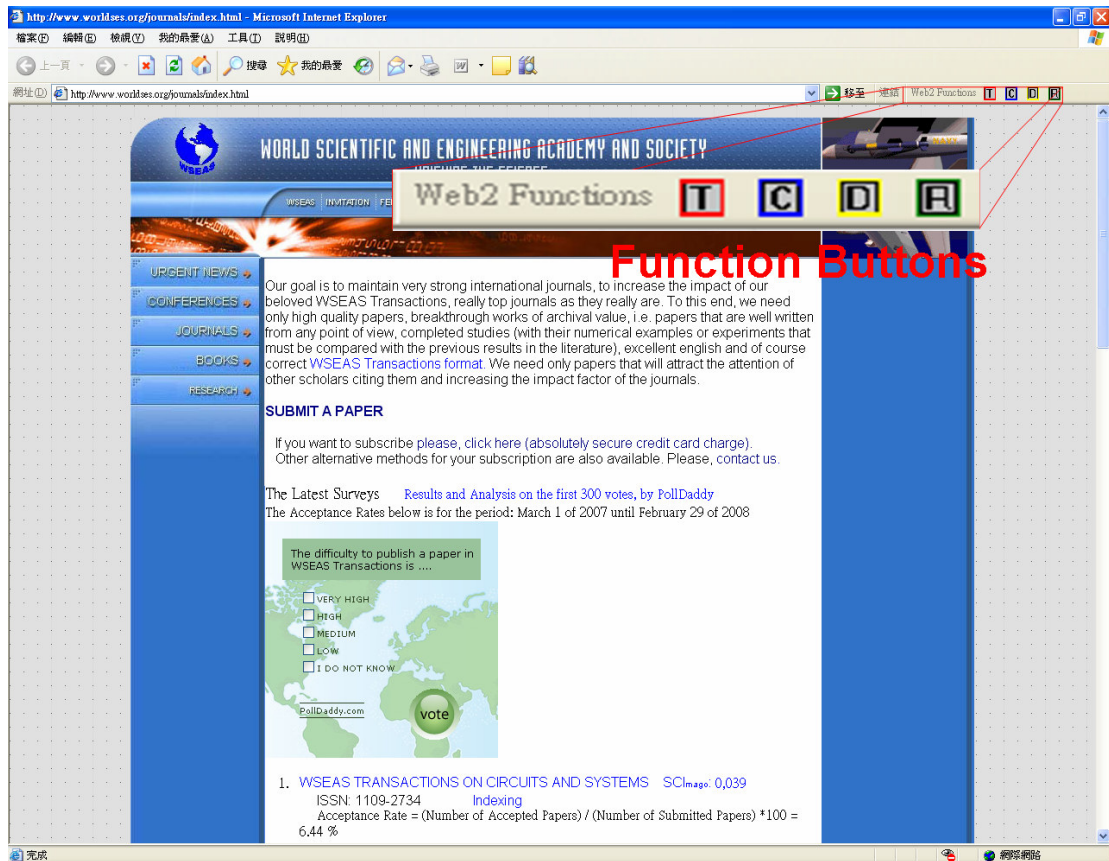


Figure 5-3: After Installing Web 2.0 Toolbar

After clicking one of these buttons to start Web 2.0 Toolbar, there is an additional float window on this web page, and the float window contains one login interface and one function interface for corresponding service, which shows on Figure 5-4.

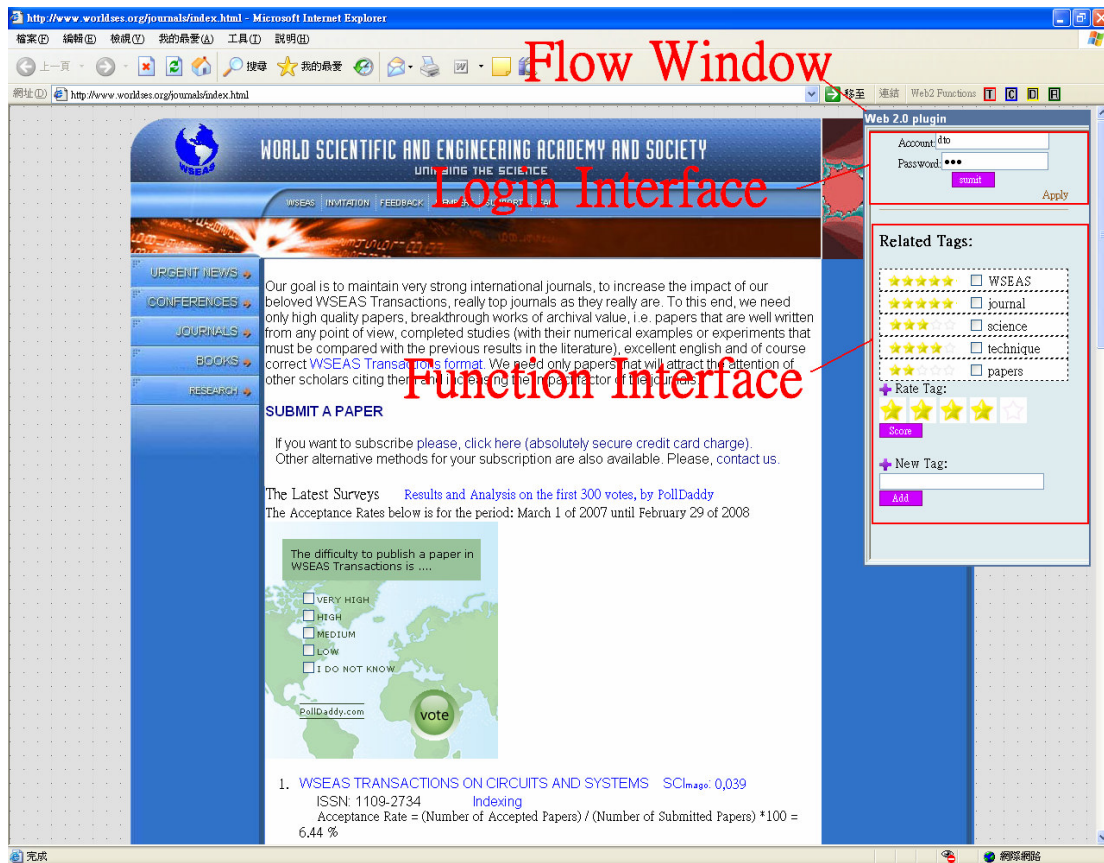


Figure 5-4: Using Web 2.0 Toolbar

5.2 Upgrade to Web 2.0

After starting Web 2.0 Toolbar, users can select the Tag button to make and rate tags. Figure 5-5 shows the web page after making and rating tags. So does Comment button. Figure 5-6 shows the web page after making and rating comments.

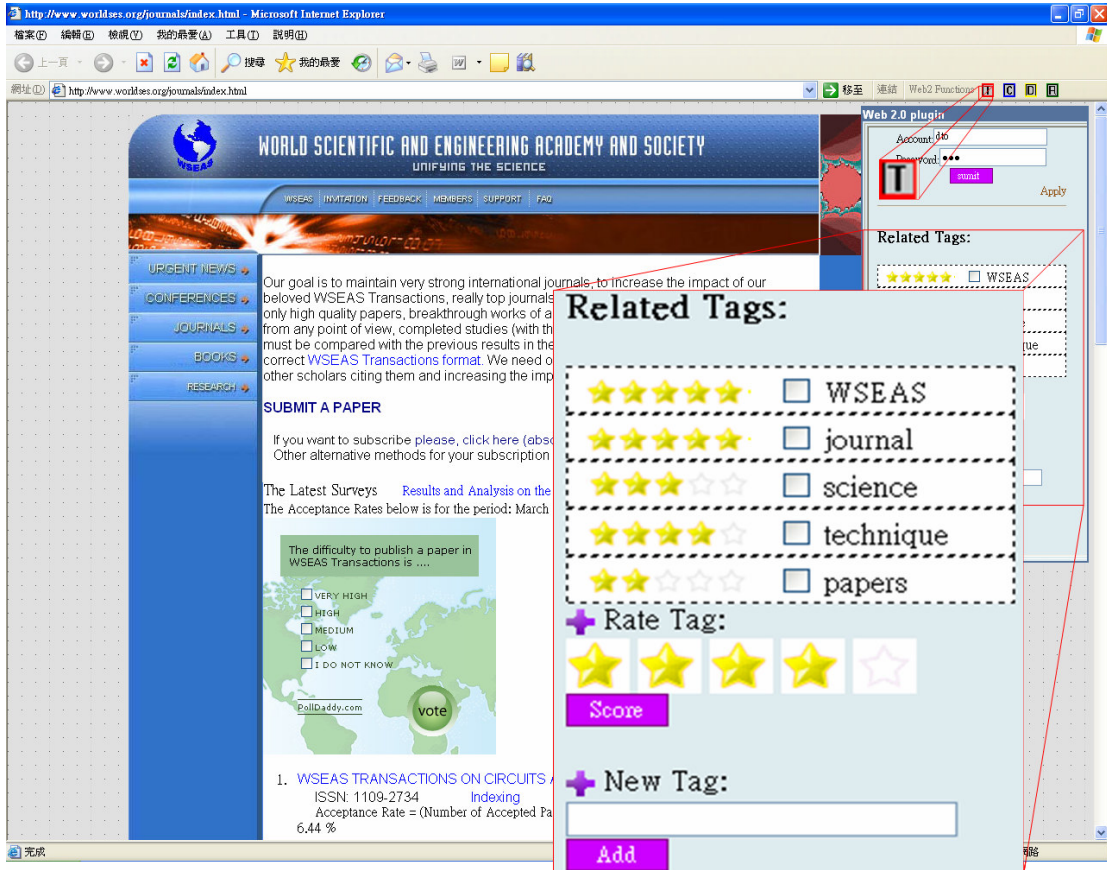


Figure 5-5: Web page Using Web 2.0 Toolbar with tags



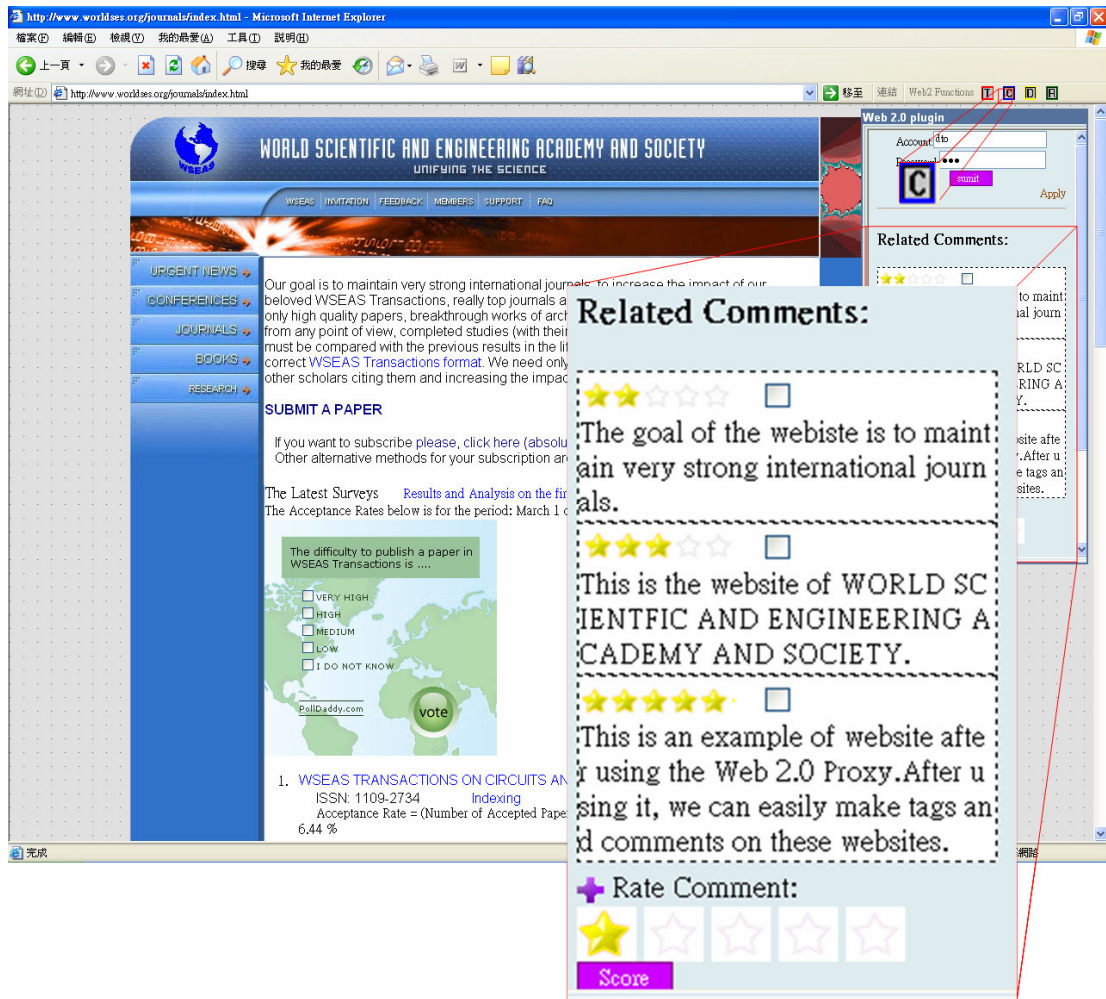


Figure 5-6: Web page Using Web 2.0 Toolbar with comments

5.3 Chat Room

When users select the Chat Room button, they can enter the chat room of this web page to talk to each other who are also viewing this web page. Figure 5-7 shows that two users, dtohuy and xc are discussing about this journal web page.

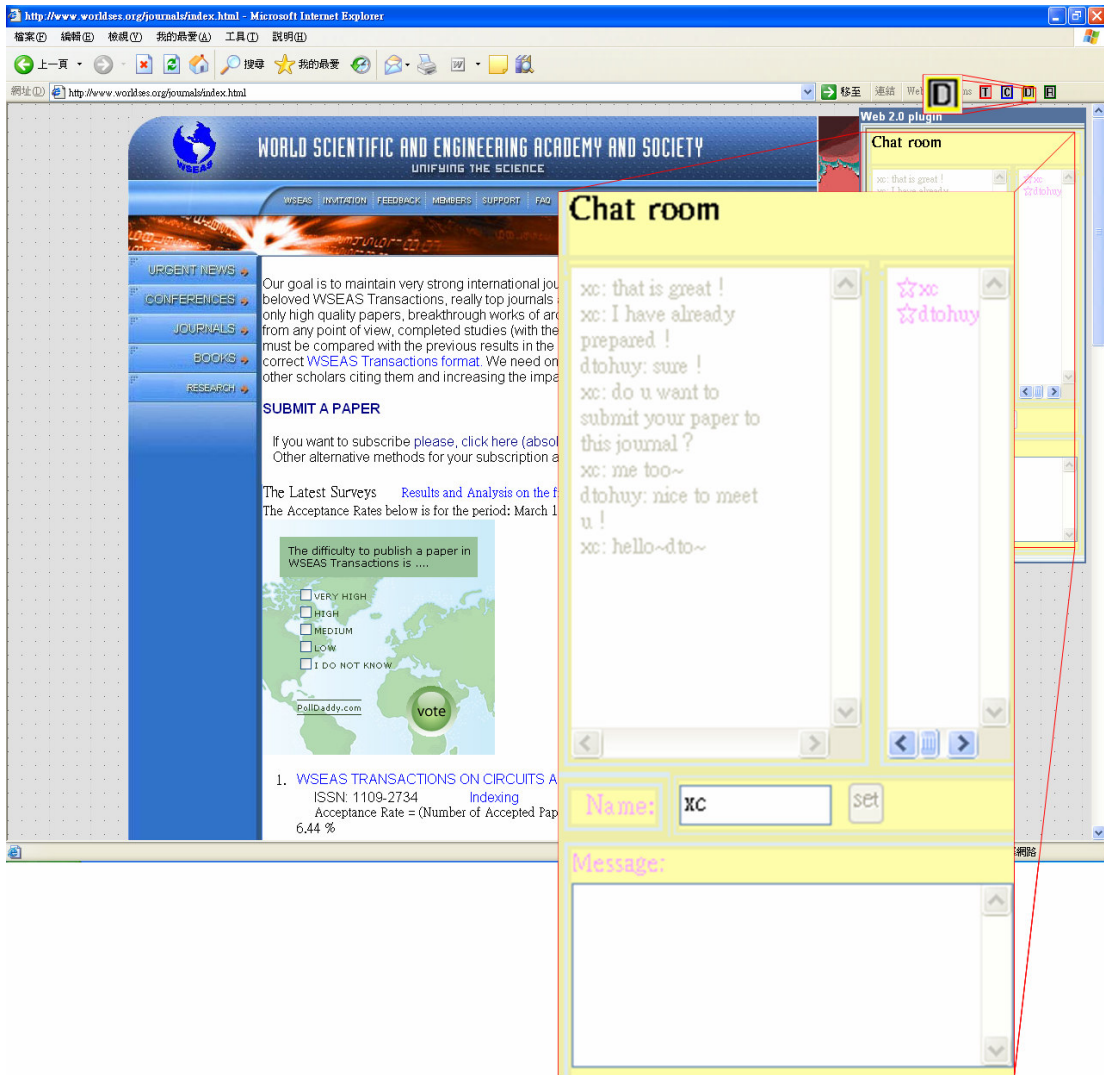


Figure 5-7: Web page Using Web 2.0 Toolbar with Chat Room

5.4 RSS

Web 2.0 Toolbar supports a simple RSS reader. Users can subscribe or un-subscribe some RSS feeds. Once some of these websites have an update, RSS Manager will show a yellow star on Web 2.0 Toolbar to notice users that there is news. Figure 5-8 shows the snapshot.



Figure 5-8: Web 2.0 Toolbar with RSS Manager

5.5 Website Manager

Our framework supports a user guide of Web 2.0 Toolbar and a manage system for Website Developers. This user guide is a web page, which supports a download point of Web 2.0 Toolbar program and a using example to tech users how to use the Web 2.0 Toolbar, which shows as Figure 5-9. The manage system allows these Website Developers to manage the information which related to their websites. Website Developers can add/modify/delete tags and comments for their web pages. Figure 5-10 shows the manage web page of Website Manger.

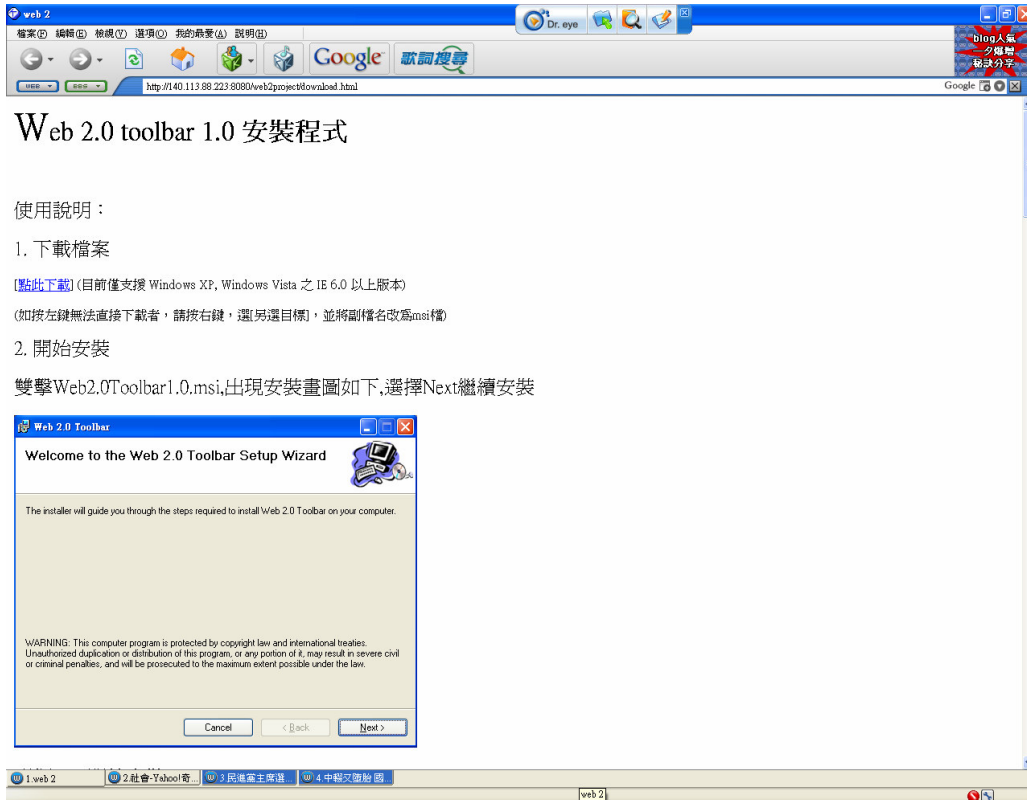


Figure 5-9: Web 2.0 Toolbar User Guide

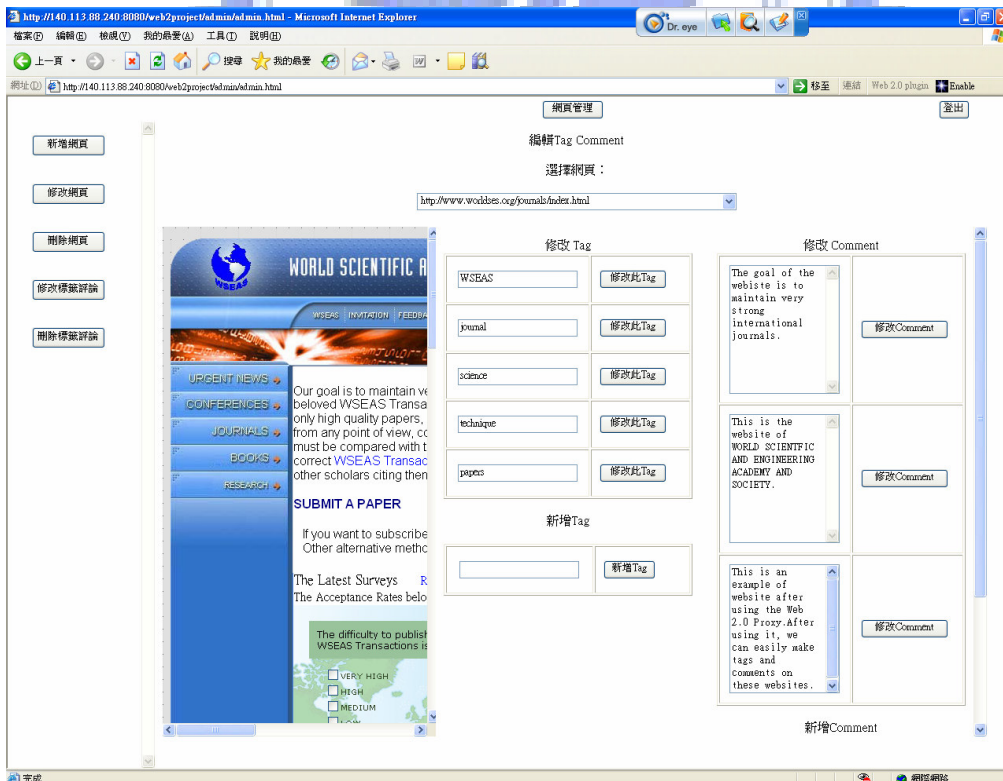


Figure 5-10: Website Manager for Website Developer

Chapter 6

Conclusions and Future Works

6.1 Conclusions

In the thesis our Toolbar framework has three contributions. First, the Toolbar framework supports a simple and effective method to allow traditional Web 1.0 websites to upgrade to Web 2.0, which append four functions, they are making , sharing and rating tags/comments , discussing in the chat rooms which are corresponding to the web page and subscribing/publishing RSS feeds to original websites. All of the four services are the most common and useful ways to get , share and discuss information in Web 2.0. Second, the Toolbar framework displays a friendly interface which integrates original web content and the related information (tags , comments) and services (chat room , RSS) closely, So that the users will not feel disharmonious. Finally, the Toolbar framework provides a flexibility interface to expand more and more services of Web 2.0.

Although our Toolbar framework presents well results, however, it still has some limitations and a big difficulty to conquer. For one thing, our system can work well on almost all the traditional Web 1.0 websites. Even though it can be used on the greater part of Web 2.0 websites, it will be liable to error (parts of Javascript code may conflict). But the existing Web 2.0 websites are not the goals which our system focuses on, and they are out of our consideration. Overall, the targets which are suitable to user our system are the traditional Web 1.0 websites which do not contain too many or complex Javascript code.

Over and above, we should face a serious problem which is the bandwidth problem on the Web 2.0 Management in the future. Although the problem is very important and difficult to solve, there are much experience of other large-scale websites which we can refer to. [30], [31], [32]

6.2 Future Works

The thesis has already presented a feasibility framework to upgrade websites from Web 1.0 to Web 2.0. The further directions of this study will be separated to four parts. First, besides IE, there is still 24.74% usage market share on the other browsers. Ex: Firefox has 17.28% 、 Safari has 5.73%. They occupy almost one quarter usage market, so we can consider porting the Toolbar framework to them. Using the framework to make Web 2.0 Firefox Toolbar or Web 2.0 Safari Toolbar is possible and even to the other browsers. Second, we can enhance the four existent services. For examples, there will support an AI or Data Mining [27], [28] mechanism to help users to automatic tagging. Supplying sounds or pictures in chat room or allowing users to make private chat room, and then users will feel more convenient. Support RSS auto-discovery or cooperate with Ponyfish [29]. Third, we can gather statistics of tags and comments. Count the numbers of tags/comments which a user made. We can make a chart to encourage users to make more tags/comments. Count the numbers of tags/comments related to a web page. We can conjecture that whether the information of a web page is worthy or not. Fourth direction is that we can use our system in special-purpose service. For examples, movie 、 food and education. Only allow the professionals to make tags and comments. This way will let the information more meaningful. Consequently, final direction is that we can find more and more practice Web 2.0 services and enrich the functions of the Toolbar framework. In the course of

time, all the needs of these Web 1.0 website developers can be satisfied with the Toolbar framework.



Chapter 7 Reference

- [1] Tim O'Reilly, "What is Web 2.0", Sept. 2005,
[http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/
what-is-web-20.html](http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html)
- [2] Anderson. Chris , The Long Tail: Why the Future of Business Is Selling Less
of More, Hyperion, 2006.
- [3] WIKIPEDIA, <http://www.wikipedia.org/>
- [4] del.icio.us, <http://del.icio.us/>
- [5] Market Share Survey BY NET APPLICATIONS,
<http://marketshare.hitslink.com/report.aspx?qprid=0>
- [6] Odeo, <http://odeo.com/>
- [7] YouTube, <http://tw.youtube.com/>
- [8] Susanne Boll, MultiTube-Where Multimediaand Web 2.0 Could Meet,
Multimedia IEEE, Vol.14, No.1, 2007, pp. 9-13.
- [9] Banff, Alberta, Canada, The two cultures: mashing up web 2.0 and the
semantic web, Proceedings of the 16th international conference on World
Wide Web, SESSION: Semantic web and web 2.0, 2007, pp. 825-834.
- [10] Becky Gibson, Enabling an Accessible Web 2.0, International
Cross-Disciplinary Conference on Web Accessibility, 2007.
- [11] Michael Cooper, Accessibility of Emerging Rich Web Technologies: Web 2.0
and the Semantic Web, International Cross-Disciplinary Conference on Web

Accessibility, 2007.

- [12] Mary Zajicek, Web 2.0: Hype or Happiness?, International Cross-Disciplinary Conference on Web Accessibility, 2007.
- [13] Jesse James Garrett, "Ajax: A New Approach to Web Applications," Feb. 2005,
<http://www.adaptivepath.com/publications/essays/archives/000385.php>
- [14] Anne van Kesteren, "The XMLHttpRequest Object", April. 2007,
<http://www.w3.org/TR/XMLHttpRequest/>
- [15] Mark Pilgrim, "The Myth of RSS compatibility", Feb 2004,
<http://diveintomark.org/archives/2004/02/04/incompatible-rss>
- [16] Google Toolbar, <http://toolbar.google.com/T4/index.html>
- [17] Yahoo Toolbar, <http://uk.toolbar.yahoo.com/>
- [18] Chi-Yang Tsai, Shyan-Ming Yuan. "Web Page Tailoring Tool for Mobile", master's thesis, Dept. Computer Science and Eng., Chiao Tung Inst., 2006.
- [19] Gabbly, <http://gabbly.com/>
- [20] Ming-Chih Hsieh et al., Web 2.0 Proxy: A New Framework for Web 2.0 Web site Development, Proceedings of the 7th WSEAS international conference on Telecommunications and Informatics, 2008.
- [21] Hypertext Transfer Protocol,
<http://www.w3.org/Protocols/rfc2616/rfc2616.html>
- [22] Erik Thompson, "Internet Explorer Toolbar (Deskband) Tutorial", Aug.2001,
<http://www.codeproject.com/KB/shell/ietoolbartutorial.aspx>

- [23] MySQL, <http://www.mysql.com/>
- [24] Apache Tomcat, <http://tomcat.apache.org/>
- [25] DWR, <http://getahead.org/dwr/>
- [26] ROME: An open source to process RSS and Atom,
<https://rome.dev.java.net/>
- [27] Steels. L, and Kaplan. F, Situated Grounded Word Semantics, Proceedings of the Sixteenth international Joint Conference on Artificial intelligence. T. Dean, Ed. Morgan Kaufmann Publishers, San Francisco, CA, 862-867.
- [28] Sheung-On Choy, Lui A.K, Web Information Retrieval in Collaborative Tagging Systems, Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference, Dec. 2006 pp 352-355.
- [29] Ponyfish, <http://www.ponyfish.com/>
- [30] Williamlong, “解決網站大流量問題的策略”, Jan. 2005,
<http://www.williamlong.info/archives/776.html>
- [31] Tong Sau Loon and Vaduvur Bharghavan, Alleviating the Latency and Bandwidth problems in WWW Browsing, Proceedings of the USENIX Symposium on Internet Technologies and Systems, December 1997. pp 20-20.
- [32] Herbert J. Carlin, A new Approach to gain-bandwidth problems, Proceedings of Circuits and System, IEEE Transactions, 2003. Vol. 24, pp 170-175.

A.1 Toolbar Controller

In this section, we show the main source code of building an IE Toolbar with four buttons and calling the Web 2.0 viewer to execute the Javascript code.

The function `onCreate` displays how to build four buttons on IE toolbar.

```
LRESULT CMFToolbar::OnCreate(UINT uMsg, WPARAM wParam, LPARAM lParam,
BOOL& bHandled)
{
// buttons with images and text
SendMessage(m_hWnd, TB_SETEXTENDEDSTYLE, 0,
(LPARAM) TBSTYLE_EX_DRAWDDARROWS);
// Sets the size of the TBBUTTON structure.
SendMessage(m_hWnd, TB_BUTTONSTRUCTSIZE, sizeof(TBBUTTON), 0);
// Set the maximum number of text rows and bitmap size.
SendMessage(m_hWnd, TB_SETMAXTEXTROWS, 1, 0L);

// add our button's caption to the toolbar window
TCHAR* pCaption = _T("");
int iIndex = ::SendMessage(m_hWnd, TB_ADDSTRING,
0, (LPARAM)pCaption);

// load our button's icon and create the image list to house it.
HICON hMotley = LoadIcon(_Module.GetResourceInstance(),
MAKEINTRESOURCE(IDI_TAG));
m_hImageList = ImageList_Create(14,16, ILC_COLOR16, 2, 0);
int iImageIndex = ImageList_AddIcon(m_hImageList, hMotley);
DestroyIcon(hMotley);
// Set the toolbar's image
::SendMessage(m_hWnd, TB_SETIMAGELIST, 0, (LPARAM)m_hImageList);

// add the button for the toolbar to the window
TBBUTTON Button;
ZeroMemory((void*)&Button, sizeof(TBBUTTON));
Button.idCommand = IDM_TAG;
Button.fsState = TBSTATE_ENABLED;
Button.fsStyle = TBSTYLE_BUTTON | TBSTYLE_AUTOSIZE;
Button.dwData = 0;
Button.iString = iIndex;
Button.iBitmap = 0;

::SendMessage(m_hWnd, TB_INSERTBUTTON, 0, (LPARAM)&Button);
```

```

TCHAR* pCaption_2 = _T("");
int iIndex_2 = ::SendMessage(m_hWnd, TB_ADDSTRING,
0, (LPARAM)pCaption_2);

HICON hMotley_2 = LoadIcon(_Module.GetResourceInstance(),
MAKEINTRESOURCE(IDI_Comment));
int iImageIndex_2 = ImageList_AddIcon(m_hImageList, hMotley_2);
DestroyIcon(hMotley_2);
// Set the toolbar's image
::SendMessage(m_hWnd, TB_SETIMAGELIST, 0, (LPARAM)m_hImageList);

TBBUTTON Button_2;
ZeroMemory((void*)&Button_2, sizeof(TBBUTTON));
Button_2.idCommand = IDM_COMMENT;
Button_2.fsState = TBSTATE_ENABLED;
Button_2.fsStyle = TBSTYLE_BUTTON | TBSTYLE_AUTOSIZE;
Button_2.dwData = 0;
Button_2.iString = iIndex_2;
Button_2.iBitmap = 1;
::SendMessage(m_hWnd, TB_INSERTBUTTON, 1, (LPARAM)&Button_2);

TCHAR* pCaption_3 = _T("");
int iIndex_3 = ::SendMessage(m_hWnd, TB_ADDSTRING,
0, (LPARAM)pCaption_3);

HICON hMotley_3 = LoadIcon(_Module.GetResourceInstance(),
MAKEINTRESOURCE(IDI_MOTLEY));
int iImageIndex_3 = ImageList_AddIcon(m_hImageList, hMotley_3);
DestroyIcon(hMotley_3);
// Set the toolbar's image
::SendMessage(m_hWnd, TB_SETIMAGELIST, 0, (LPARAM)m_hImageList);

TBBUTTON Button_3;
ZeroMemory((void*)&Button_3, sizeof(TBBUTTON));
Button_3.idCommand = IDM_CHAT;
Button_3.fsState = TBSTATE_ENABLED;
Button_3.fsStyle = TBSTYLE_BUTTON | TBSTYLE_AUTOSIZE;
Button_3.dwData = 0;
Button_3.iString = iIndex_3;
Button_3.iBitmap = 2;

::SendMessage(m_hWnd, TB_INSERTBUTTON, 2, (LPARAM)&Button_3);

TCHAR* pCaption_4 = _T("");
int iIndex_4 = ::SendMessage(m_hWnd, TB_ADDSTRING,
0, (LPARAM)pCaption_4);

HICON hMotley_4 = LoadIcon(_Module.GetResourceInstance(),
MAKEINTRESOURCE(IDI_RSS));
int iImageIndex_4 = ImageList_AddIcon(m_hImageList, hMotley_4);
DestroyIcon(hMotley_4);
// Set the toolbar's image
::SendMessage(m_hWnd, TB_SETIMAGELIST, 0, (LPARAM)m_hImageList);

```

```

TBBUTTON Button_4;
ZeroMemory((void*)&Button_4, sizeof(TBBUTTON));
Button_4.idCommand = IDM_RSS;
Button_4.fsState = TBSTATE_ENABLED;
Button_4.fsStyle = TBSTYLE_BUTTON | TBSTYLE_AUTOSIZE;
Button_4.dwData = 0;
Button_4.iString = iIndex_4;
Button_4.iBitmap = 3;
::SendMessage(m_hWnd, TB_INSERTBUTTON, 3, (LPARAM)&Button_4);
//::SendMessage(m_hWnd, TB_INSERTBUTTON, 0,
(LPARAM)&m_com_button);
// create our EditQuote window and set the font.
RECT rect = {0,0,0,0};
return 0;
}

```

The function Comment displays how to execute some Javascript code in IE when clicking the comment button.

```

void CMFToolbar::Comment () {
    if (m_pBrowser)
    {
        VARIANT vEmpty;
        VariantInit(&vEmpty);
        m_pBrowser->Stop();
        _bstr_t bsSite;
        CComPtr<IDispatch> spDispatch;
        BSTR bstrTickers = NULL;
        if (m_pBrowser->get_Document(&spDispatch) == S_OK) {
            CComQIPtr<IHTMLDocument2> spHTMLDoc = spDispatch;
            CComQIPtr<IHTMLWindow2> pHTMLWnd;
            _bstr_t
strjs="javascript:void(b=document.getElementsByTagName('body')[0].
appendChild(document.createElement('script')));void(b.type='text/j
avascript');void(b.id='tk_bookmarklet');void(b.src='http://140.113
.88.240:8080/web2project/javascrpts/loader_comment.js');"
            if (spHTMLDoc!=NULL) {
                spHTMLDoc->get_parentWindow(&pHTMLWnd);
                if (pHTMLWnd!=NULL) {
                    CComBSTR
bstrlan=SysAllocString(L"javascript");
                    VARIANT varRet;
                    pHTMLWnd->execScript(strjs,bstrlan,&varRet);
                }
            }
        }
    }
}

```

A.2 Web 2.0 viewer

In this section, we show the main Javascript source code of how to append the JSP file into the web page which IE is processing.

```
Toolkit.prototype.initialize = function()
{
    // construct toolkit window
    this.tk_window = document.createElement("div");
    this.tk_window.id = TK_WINDOW;
    this.tk_title = document.createElement("div");
    this.tk_title.id = TK_TITLE;
    this.tk_title.appendChild(document.createTextNode(tk_titleText
));
    this.tk_body = document.createElement("div");
    this.tk_body.id = TK_BODY;
    this.login = document.createElement("div");
    muse_login=this.login;
    muse_frame= document.createElement("iframe");
    if(viewpage==1)
        muse_frame.src="http://140.113.88.240:8080/web2project+"/tag.
jsp?op=input&url="+encodeURIComponent(document.location);
    if(viewpage==2)
        muse_frame.src="http://140.113.88.240:8080/web2project+"/comm
ent.jsp?op=input&url="+encodeURIComponent(document.location);
    if(viewpage==3)
        muse_frame.src="http://140.113.88.240:8080/web2project+"/chat
.jsp?op=input&url="+encodeURIComponent(document.location);
    if(viewpage==4)
        muse_frame.src="http://140.113.88.240:8080/web2project+"/rss.
jsp?op=input&url="+encodeURIComponent(document.location);

    muse_frame.id="login_doc";
    muse_frame.width=260;
    muse_frame.height=500;
    this.login.appendChild(muse_frame);
    // auto save
    this.tk_window.appendChild(this.tk_title);
    this.tk_window.appendChild(this.tk_body);
    this.tk_body.appendChild(this.login);
    // flag used to check whether webpage configurations have changed
or not
    this.tk_modified = false;
    // use Effect.Appear() to let it appear later
    this.tk_window.style.display = "none";
    // append toolkit window to <body>
    document.getElementsByTagName("body")[0].appendChild(this.tk_w
indow);
    this._stylize();
}
```