A web browser for Ajax approach with asynchronous communication model

Noriko Hanakawa, Nao Ikemiya
Graduate school of Corporation information
Hannan University, Japan
hanakawa@hannan-u.ac.jp

Abstract

We can receive large information from various web sites using the Internet. The web sites are important means to collect the required information in researching, learning, and even in commerce activities. Importance of web browsers grows increasingly as web sites increase. Web browsers are powerful tools to refer to web sites. However, due to necessary of synchronous communication with web servers, operationality of web pages on web browser is not better than operationality of desktop applications. Therefore, we propose a new web browser for Ajax approach with asynchronous communication model. Our web browser can improve the operationality that is equivalent to the operationality of desktop applications without revising program codes of the web applications. A feature of the browser is partial updating of a web page even if the web application does not adopt Ajax approach. As a result of experiments, we have confirmed improvement of operationality of "Yahoo auction sites" on our web browser when the load of the web server increases.

1. Introduction

Recently, we have received valuable benefits from large web sites on the Internet. Much information is provided through the web sites. Net-surfing is more important means to search the information. Moreover, web sites are useful in cases not only searching for the information but also commercial activities such as Net-shopping and Net-auctions. Even education in universities and schools can not ignore information from web sites. Web sites are indispensable information resources on various scenes in current societies. On the other hands, the most usual tools in order to refer to web sites are web browsers such as Internet Explorer (IE) and Firefox. The web browsers generate a visible web page while the browsers are analyzing HTML source codes. This is a typical client-server system called "web application". Everyone can access easy web applications through web browsers.

However, we have problems of web applications on web browsers. Operationality of web applications on web browsers is less than operationality of desktop applications, that is, slow performance and limited interactivity of operationality on web application. Desktop applications mean that all software is installed to a user computer. Desktop applications are able to achieve quick responses to user requests because desktop applications do not need communication with servers, and waiting for calculations in server computers. Especially, if it takes long time for communicating with servers in web applications, a visible web page on a web browser may become a white-out page that has no texts and no images. The white-out page occurs frequently, when a user requests "Reload" under heavy loaded servers. The white-out web page leads to not only no-better operationality but also users' confusions because users lose sight of operation on a web page.

To resolve the weakness of the operationality of web applications, Ajax (Asynchronous JavaScript + XML) technology is adopted to web applications [1]. If a web application is constructed based on Ajax technology, the white-out web pages will be avoided because client programs (e.g. JavaScript) in HTML source communicates asynchronously with web servers. Client programs in HTML source do not need to wait for finish of communication with server. Because Ajax technology can improve operationality of web applications, many developers adopt Ajax technology to their projects. A most famous web application is Google Map [2]. The web application of Google map avoids the status of white-out web pages because of asynchronous communication with web servers.

Although Ajax approach is a better way of improving operationality of web applications, there is a significant problem. Usually, developers have to change program codes of client software and server software of web applications. Users can not receive benefits from Ajax approach in web applications

Proceedings of the 2006 IEEE/WIC/ACM International Conference
on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)
0-7695-2747-7/06 $20.00 © 2006 IEEE

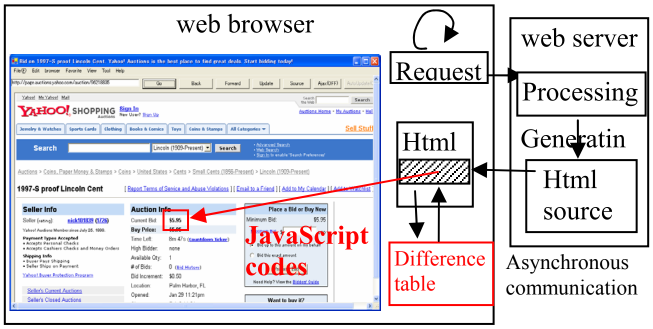


Figure 1. A basic concept of the browser

unless developers modify current program codes of the web applications. General public users only do nothing but waiting for modifying program codes of the web applications. However, there are a large number of web sites and web applications in the Internet society. Because opportunities to receive the benefits from Ajax depend on progresses of modification of program codes, users need to wait long time for receiving the benefits from Ajax approach in current web applications.

Therefore, we propose a new web browser based on an asynchronous communication model. The browser supports asynchronous communication with web server computers like Ajax approach. If users access web applications using our browser, users will be able to receive similar benefits from Ajax even if the web applications do not adopt Ajax approach. The browser has a function of communicating asynchronously with web servers. Section 2 shows related works, section 3 presents a basic idea of our browser, and the detail of the asynchronous communication model of the browser. Section 4 shows experiment results of comparing current normal web browsers such as Internet Explore. In section 5, we discuss usefulness and limitation of the browser. Section 6 summaries our research.

2. Related works

Many useful Internet technologies have been proposed for improving performance of web applications. Sieminski focused on the local system cache of Internet Temporary files [3]. Analysis of the local cache is useful for not only estimating upper limit of caching efficiency, but also measuring changeability of Web Objects. The changeability is derived from values of the cacheability factor (CF) of the caches. Using the changeability, the CF is a simple measure of susceptibility to caching and could be used to predict the latency. Moreover, Li et al. also studied the local system cache of Internet object for improving the browser performance [4]. They proposed a peer-to-peer Web document sharing technique, called a "browser-aware proxy server". To improve the performance of loading the web pages, the browser searches not only own local caches but also the other computers' caches through network. Therefore, even if a user has never seen a web page, the web page will be able to be displayed on the browser using caches of the other computers. Performance of displaying the previous web pages is dealt with on both researches, that is, the performance of viewing the web pages that have once displayed on web browsers. However, our browser can improve performance in updating partially a current web page that has never displayed anywhere.

On the other hand, the intelligent web browsers have been proposed. Many useful functions are embedded to web browsers for improving user operationally. Bergasa-Suso et al. have proposed a new browser that embedded a function of filtering, a function of inferring learning style, and a function of recommending relevant web pages [5]. Shigesada et al. constructed a new type of browser called BKB (BTRON Kiosk Browser), which has the function of presenting WWW content through a user interface suited to the kiosk terminal [6]. Matsuda et al. developed a graphical history browser. The purpose of the browser is to help a user to utilize Undo selectively and easily [7]. These browsers have valuable functions in a case of a specific situation. Our browser is also a kind of these browsers including a special function. Each browser is suitable to each situation. When a user views a web page which is updated partially and frequently, our browser will be useful as same as these browsers.

3. A new web browser

3.1. A basic concept of the browser

Ajax technology has various possibilities of operationality on web applications. For example, "Google suggestion" [8] can generate a new candidate list as users input characters to a text box on the Google suggestion web page. In Google Map site [9], when users click on the map image of Google Map, only a part of the map image is updated without reloading the whole web page. Users can operate continuously the map site without waiting for processing in the web servers. Therefore, white-out web pages of the site never occur because client programs on a web browser does not need waiting for communication with the web servers. Although we can put various useful functions into practice using Ajax technology to web applications, we focus on a function of updating partially different parts on a web page without reloading the whole web page. Even if users click “Reload” button, our browser can avoid updating the whole web page. Our browser has the following basic concepts;

(1) No change of client programs and sever programs of web applications,
(2) Asynchronous communication with web servers,
(3) Updating only different parts of a web page.

Figure 1 shows a concept of our browser. At first, our browser requests a web page of a specific URL to a web server. The browser displays just the web page according to HTML source that is sent from the web server. After that, the user requests again the same web page like clicking “Reload” button, the browser requests again the web page to the web server. However, at the second time, the browser does not display the whole web page. Before the browser displays again the web page, the browser analyzes differences between a current web page and a next web page. The differences between the two web pages are listed to a difference table in the browser. After the analysis, the browser generates automatically “JavaScript code”. The JavaScript code can modify parts of the web page using “innerHtml” in JavaScript.

In addition, the browser’s request to the web server at the second time communicates asynchronously. That is, users can operate next functions on the current web page without waiting time for completion of processing on the web server.

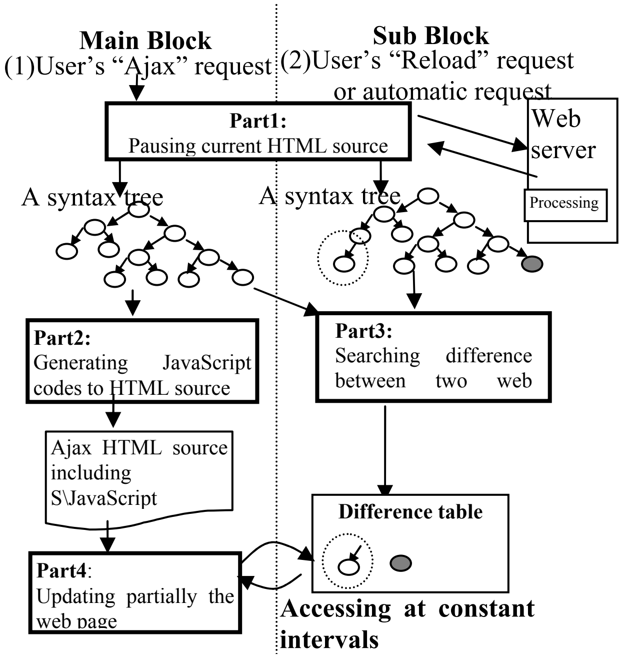

Main Block
Sub Block
(1)User’s “Ajax” request
(2)User’s “Reload” request or automatic request
Part1:
Pausing current HTML source
Web server
Processing
A syntax tree
A syntax tree
Part2:
Generating JavaScript codes to HTML source
Part3:
Searching difference between two web
Ajax HTML source including S\JavaScript
Difference table
Part4:
Updating partially the web page
Accessing at constant intervals

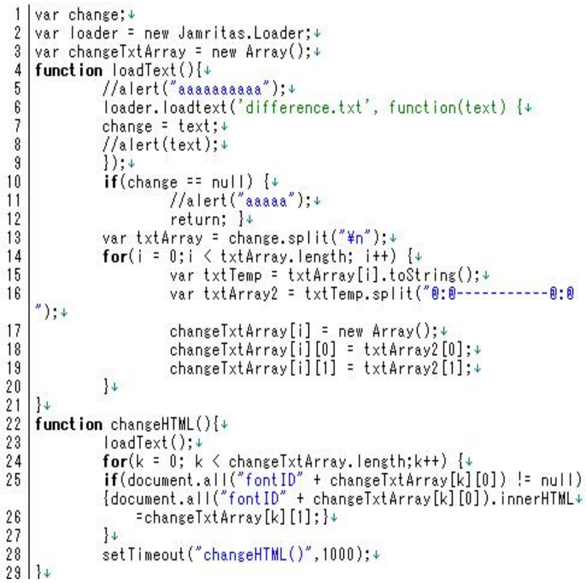
Figure2. The asynchronous communication model

3.2. The asynchronous communication model

The asynchronous communication model has been established to bring the basic idea to fruition. Figure 2 shows the asynchronous communication model. The model consists of four components. The feature of the model is separation between “Main block” and “Sub block”. In the Main block, an original web page changes to a partial updatable web page including JavaScript codes. The partial update of the web page executes on the browser with accessing a difference table. Role of the Sub block is communication with web server in order to get a next web page. Moreover, in the Sub block, a difference table between the current web page and the next web page is generated using the two syntax trees. Because the Main block and the Sub block are executed independently in the model, the browser can establish asynchronous communication with the web servers while users are operating the current web page on the browser. The following sub sections describe the four parts of the model.

3.2.1. Part1: Parsing. HTML source is analyzed by a parser based on HTML tags. Syntax trees are generated by the parser. Nodes of the syntax tree mean HTML tags, leaves of the syntax tree mean text contents such as String, numerical, and JavaScript codes. Because the analysis is based on HTML tags, the other elements such as Script codes are dealt with as general texts (leaves of the syntax tree). In addition, to update partially a web page, new tags’ properties named “ID” are inserted to each node of the syntax trees. When a web page is updated partially, the browser distinguishes no-update parts from update parts of the web page using the “ID” property.

3.2.2. Part2: Generating JavaScript. The original HTML source does not have a function of partial updating. Therefore, new JavaScript codes are added to the original HTML source. The JavaScript codes refer to the difference table that is generated in Part3. JavaScript codes including “innerHTML” method are inserted to the original HTML source. Figure 3 shows the JavaScript codes. The JavaScript codes consist of 2 functions; LoadText() and ChangeHTML(). The function of ChangeHTML() is called when a user wants to update partially the current web page. The function of ChangeHTML() calls the function of LoadText() (See the 23rd line of Fugure3). The function of LoadText() searches the difference table. The file name of the difference table is “difference.txt”(See the 6th line). The information of the difference table saves to a two-dimensional variable named “changeTextArray [][]” (See from the 17th line to the 19th line). The value of “changeTextArray[k][0]” means “ID” property that is given to each node of the syntax tree, the value of “changeTextArray[k][1]” means the new text with



```
 1 var change;↓
 2 var loader = new Jamritas.Loader;↓
 3 var changeTxtArray = new Array();↓
 4 function loadText(){↓
 5         //alert("aaaaaaaaaa");↓
 6         loader.loadtext('difference.txt', function(text) {↓
 7         change = text;↓
 8         //alert(text);↓
 9         });↓
10         if(change == null) {↓
11                 //alert("aaaaa");↓
12                 return; }↓
13         var txtArray = change.split("¥n");↓
14         for(i = 0;i < txtArray.length; i++) {↓
15                 var txtTemp = txtArray[i].toString();↓
16                 var txtArray2 = txtTemp.split("@:@-----------@:@
   ");↓
17                 changeTxtArray[i] = new Array();↓
18                 changeTxtArray[i][0] = txtArray2[0];↓
19                 changeTxtArray[i][1] = txtArray2[1];↓
20         }↓
21 }↓
22 function changeHTML(){↓
23         loadText();↓
24         for(k = 0; k < changeTxtArray.length;k++) {↓
25         if(document.all("fontID" + changeTxtArray[k][0]) != null)
           {document.all("fontID" + changeTxtArray[k][0]).innerHTML↓
26             =changeTxtArray[k][1];}↓
27         }↓
28         setTimeout("changeHTML()",1000);↓
29 }↓
```

Figure3. JavaScript codes in HTML source

the "ID" property. The new text is a leaf of the syntax tree. In the 25th line, using the "document.all().innerHTML" method, the partial updating actually execute. The tag with "ID" (changeTextArray[k][0]) of HTML source is modified by the new property (changeTextArray[k][1]). At a moment of execution of the "document.all(). innerHTML" method, the web page has been updated. In addition, the function of ChangeHTML() includes "setTimeout()" method at the last line of the JavaScript codes. The function of ChangeHTML() is repeated at 10 seconds intervals. In short, the function of ChangeHTML() refer to the different table at 10 seconds intervals, after that, if there are some difference between a current web page and a next web page, the two-dimensional variable is created. By the two-dimensional variables, "document.all(changeText Array[k][0]).innerHTML" method runs.

The JavaScript codes are common codes to various HTML sources. The difference table has been built in the Sub block of Figure 2. The "ID" properties have been assigned to the nodes of the syntax trees. Therefore, the common JavaScript codes are available to various HTML sources. Because of common JavaScript codes, a "JS" file including the JavaScript codes shown in Figure 3 is prepared beforehand. The new code that should be inserted to the original HTML source is only "<SCRIPT onload=xxx.js></SCRIPT>". After that, the browser updates once again the whole new web page including the JavaScript codes. The "changeHTML()" runs at 10seconds intervals.

3.2.3. Part3: Searching difference. With comparing the two syntax trees, difference between the two web pages is detected. There are many researches about difference calculation between two tree-structures. In the model, xmdiff algorithm [10] is adopted to detect the difference. In the xmdiff algorithm, delta scripts (insert, delete, and update) and costs are used. The delta scripts mean construction process of a tree. The costs mean weighting of the construction process using the delta scripts. Edit Graph (dynamic programming) is achieved using the delta scripts and costs. The nodes of one tree in depth first searching are plotted to on x-axis of the Edit Graph. The nodes of another tree in depth first searching are plotted to on y-axis of the Edit Graph. The horizontal members of the Edit Graph mean "delete" of a node. The vertical members of the Edit Graph mean "insert" of a node. The hypotenuse members of the Edit Graph indicate "update" of a node or leaves. The each edit script (delete, insert, and update) is set to each weight named "cost". The path with minimum cost among the plots on the Edit Graph means the most adaptable delta scripts. According to the adaptable delta script, the different nodes and leaves are determined. If you want to see the details of xmdiff algorithm, please see [10].

In our model, the cost of each edit script is defined as; Insert Script is *costI*, Delete Script is *costD*, Update Script is *costU*. In addition, we have to determine whether the next web page updates wholly. If the difference between two web pages is large, partial updating of the current web page is meaningless. Therefore, in the searching algorithm, we set a threshold value for judging the whole updating. If the value of sum of costs on minimum path in the Edit Graph is greater than a threshold value, the searching algorithm selects the whole updating. The judgment is the following;

$$\frac{\sum(costI + costD + costU)}{node} > A \qquad then\ whole\ updating \quad ..\ (2)$$

cost : cost of edit script such as "insert", "delete", "update"
node : the total number of the node of two syntax trees.
A : threshold value

After that, if partial update of the web page is required, a difference table is generated. The difference table includes the difference node information such as tag name, ID, tag's properties, text.

3.2.4. Part4: Updating partially. According to the difference table, a current web page added JavaScript codes updates partially. In order to update, the current web page accesses to the difference table at constant intervals. The reason of the constant accessing to the difference table is that the current web page in the Main block is executed independently from the communication with a web server in the Sub block. In the Sub block in the model, communication with the web server occurs corresponding to events such as "clicking reload button" or "automatic reload". In short, the Main block and the Sub block in the model are executed independently each other. Only accessing to the difference table is a relationship between the two blocks. Therefore, the asynchronous communication with web servers can be achieved.

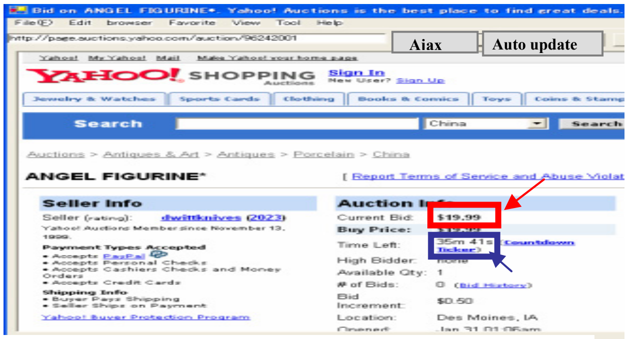
3.3. Implementation of the browser

Based on the asynchronous communication model, a new web browser has been implemented in C# language. Figure 4 shows a screenshot of the web browser. The browser has especial buttons; “Ajax”, “Auto-update”. At once, the browser displays normally a web page. If a user improves an operationality of the web page, the user clicks the “Ajax” button. The HTML source of the web page is modified in the part1 of the asynchronous communication model mentioned above. JavaScript code shown in Figure 3 is added to the original HTML source. Based on the modified HTML source, the whole web page is updated once again on the browser. The new JavaScript code is started in the web page using “SetTimeOut” method (See the last line of Figure3). Therefore, in the JavaScript code, the web page refers to the difference table at constant intervals.

After that, when the user clicks “Update” button, the browser requests a next web page to the web server. The new web page is analyzed in the part 1 of the Sub block of the asynchronous communication model. A syntax tree of the next web page is generated in the browser. Next, in the part 4 of the asynchronous communication model, difference between two syntax trees is detected. A difference table is generated in the browser. Because the modified web page (the original web page added the JavaScript code) is accessed to the difference table at constant intervals, the web page is updated partially. In Figure4, only a red rectangular area is updated without updating wholly.

In addition, if the web page is frequently modified such as Yahoo auction web page near the closing time, users can select “Auto update”. The browser requests a next web page to a web server at constant intervals. The value of the constant interval such as ten seconds is set by an optional menu of the browser. Therefore, the browser updates partially the web page without users’ clicking “Update” button, automatically.

If the browser judges a whole updating for a next web page at the part 3 of the asynchronous communication model, the next web page has few relationships with the current web page. For example, when a next page moves to other URL link, difference of two syntax tree becomes large. Analysis at the part3 of the model judges whether the whole updating is better than partial updating. This case is caused almost by changing URLs from the current web page to the next web page. Although the change of URLs can indicate the whole updating of the web page, sometimes URL has parameters to web servers such as “http//portal.hannan-u.ac.jp/login.do;jsessionid=893E”. In this case, the URL is different from the previous URL, although the next web page has changed partially from the current web page. Therefore, the browser can not judge the whole updating using only the change of URLs. Of course, the judgment of the whole updating is available at “On” status of “Ajax” button on the browser. When “Off” status of “Ajax” button on the browser, the web pages are transferred in same ways as normal browsers such as Internet Explorer.


Bid on ANGEL FIGURINE*. Yahoo! Auctions is the best place to find great deals.
File(E) Edit browser Favorite View Tool Help
http://page.auctions.yahoo.com/auction/96242001
Aiax
Auto update
Yahoo! My Yahoo! Mail Make Yahoo! your home page
YAHOO! SHOPPING Auctions
Sign In
New User? Sign Up
Jewelry & Watches
Sports Cards
Clothing
Books & Comics
Toys
Coins & Stamp
Search
China
Search
Auctions > Antiques & Art > Antiques > Porcelain > China
ANGEL FIGURINE*
[Report Terms of Service and Abuse Violat

Seller Info

Seller (rating): dwittknives (2023)

Yahoo! Auctions Member since November 13, 1999.

Payment Types Accepted
- Accepts PayPal
- Accepts Personal Checks
- Accepts Cashiers Checks and Money Orders
- Accepts Credit Cards

Shipping Info
- Buyer Pays Shipping
- Seller Ships on Payment

Yahoo! Buyer Protection Program

Auction Info

Current Bid:	$19.99
Buy Price:	$19.99
Time Left:	35m 41s (Countdown Ticker)
High Bidder:	none
Available Qty:	1
# of Bids:	0 (Bid History)
Bid Increment:	$0.50
Location:	Des Moines, IA
Opened:	Jan 31 01:06am

Figure4. The browser displaying USA Yahoo auction web page

4. Experiments

4.1. Updating texts in Yahoo auction sites

In Yahoo auction site [11], we have confirmed the partial update of a web page on our browser. Figure 4 shows a web page of Yahoo auction. “Angel figurine” item is exhibited to Yahoo auction. Seller is “dwittknives”, current bid is “$19.99”. In the web page, the value of time left such as “35m 41s” (See blue rectangle area of Figure 4) changes without “Reload” event because the value of the time left is written in JavaScript. However, the value of the current bid such as “$19.99” does not update because “$19.99” is a simple text content that is sent from the web server. The value of “$19.99” is not updated as long as the user doesn't click “Reload” button on the browser. When the user clicks “Reload” button, the whole web page is updated even if the value of the current bid did not change. A flickering of the web page occurs when the web page is updated wholly. Therefore, the web page of Yahoo auction is displayed on our browser. When a user clicks the “Ajax” button on our browser, only the value of the current bid is updated without updating wholly. Of course, if the value of the current bid does not change, nothing occurs on the web page.

In addition, if the web page is modified frequently such as Yahoo auction web page near the closing time, users can select "Auto update". Just before the closing time, many bidders bid for the bidding item. Therefore, although almost parts of the auction web page do not change, only the value of the current bid is updated frequently. If a user clicks "Auto update" button on the browser, only the value of the current bid is updated automatically and continuously. In short, although the Yahoo auction site's programs have not adopted Ajax approach, the Yahoo auction site was able to behave like a web page adopting Ajax approach.

4.2. Updating images in MSN map sites

Map sites on the Internet are useful. However, even if a user wants to move a little the central of the map on the Map sites, the whole web page is updated. Because the web page including map images flickers by the whole updating, users often lose a sight of a specific point on the map image. Because Google map site adopts Ajax technology, the operationality of the map site is better. However, the other map sites such as Japanese MSN map do not adopt Ajax technology. Therefore, we confirm possibility of our browser in Japanese MSN map site that does not adopt Ajax technology. In Japanese MSN map site (See Figure5), when a user clicks at a point of the map image, the whole web page including not only the map image but also all contents of the web page is refreshed. Therefore, we display a web page of Japanese MSN map site on our browser. We expect that only the map image (See the red rectangle area of Figure5) is updated when a user clicks at a point of the map image. As a result, the area of the map image is updated with no-updating the other contents .

5. Discussion

A most important feature of the browser is to update partially a web page without revising programs of web applications. The browser avoids flickering the web page by "Reload" and "Update". The partial update of a web page provides better operationality of web applications. However, the browser has some problems. In this section, we discuss the problems, and explore to resolve the problems.

5.1. Performance on a heavy load server

Although the browser can update partially without revising programs of web applications, we focus on two performance problems; speed of the partial updating, and correctness of judging the whole web page. At first, we evaluate the updating time of a web page under comparing with the existing web browser Internet Explorer. Of course, because Internet Explore displays web pages in normal, user clicks "Reload" button in order to update the whole web pages. The conditions of the experiments are as follows;

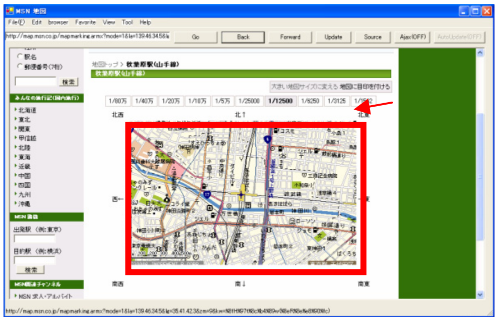


Figure5. Japanese MSN map site on our browser

(1) The web server is Apache ver.1.3.4
(2) The web pages are a local file that is downloaded from Yahoo auction site to the home directory of the web server.
(3) The updating time means a period from an event occurring time to finishing time to update partially or wholly.
(4) Comparing three types; IE's updating, updating on our browser, and auto-updating on our browser.
(5) The server is given various loads by other tasks. The tasks are the downloading files using Wget software [12].
(6) The variation of the loads is generated by the number of executions of the tasks from 2 times per second to 10 times per seconds.
(7) To compare, caches (Internet temporary files) for Internet Explorer are cleared.
(8) When "Auto update" is ON in our browser, the browser requests the web server at one second intervals.

Figure6 shows the results of the experiments of speed performance of updating time. The x-axis of Figure6 means the server load. The y-axis of Figure6 means the updating time. When the load is between 2 times and 5 times, the effect of our browser is clear. Updating on our browser is quicker than updating on IE. In addition, updating on IE in 8 times or more loads leads to a white-out page. However, our web browser leads to no white-out page even if the load is large. In auto-updating on our browser, when the server load is bigger than 5, the performance suddenly decreases. We confirm that auto-updating of our browser is also not suitable when server load is large.

Next, we discuss the correctness of judgment of updating wholly. In the experiments, according formula (1) and formula(2) in section 3.2.3, "Insert" script's cost is set to 2.0, "Delete" script's cost is set to 2.0, "Update" script's cost is set to 0.5, and the threshold value is set to 0.3. That is, when about 30 % of HTML source is different between a current web page and a next web page, the current web page should be updated wholly in replacing with the whole next web page. In Yahoo auction site, and MSN map site, the correctness of the judgment using the threshold

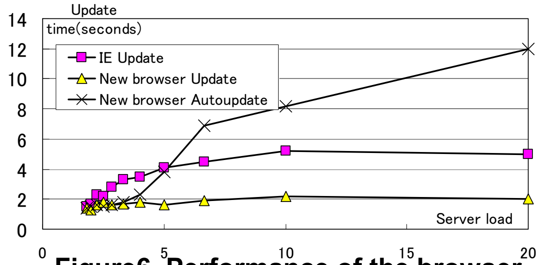


Figure6. Performance of the browser

value is 70%. The correctness of the other web sites such as MSN home page is less than 50%.

The miss-judgment of updating wholly is caused by meaningless contents of the web page. For example, the meaningless contents are codes for displaying commercial advertisements, blank code such as “ ”, carriage return code such as “\n”, and comment lines in HTML source. Especially, many advertisements of a web page lead strongly miss-judgment. Therefore, a function of omitting the meaningless contents such as advertisements will be implemented to the browser. Moreover, our judgment way may be more applicable to the web pages including specific targets such as auction sites than general information web pages such as top page of MSN. However, because of various types of web pages, we have to improve the values of the costs and the values of the threshold, and the way of judgment of whole updating.

5.2. Performance on no-load server

We discuss the performance of our browser when the server’s load is a little. When the Ajax button becomes “On” status in our browser, the additional calculations for making syntax trees, detecting difference based on xmdiff algorithm is required. Of course, the calculations need certain time. The other web browsers such as Internet Explorer do not need such calculation time. Therefore, the speed performance of our browser is lower than normal browsers when users click “Ajax” button on our browser. In addition, when the server load is a little, low performance of our browser is clarified. The time to wait for communication with servers in the normal browsers is less than the time to wait for calculation of our browser. This is demerit of our browser.

However, a most important feature of our browser covers the demerit of the speed performance. The most important feature is partial updating. Especially, when a web page has no change between a previous web page and a next web page, nothing occurs on the web page on our browser. In addition, even if the web page changes a little, almost all parts of web page do not change on our browser. The flickering of meaningless of a web page is avoided on our browser. Although the calculation in our browser is required, users can usually continue to operate the web pages on our browser. Because communication with servers and the calculations (making tree, xmdiff algorithm) in our browser are executed in the Sub block (See Figure 2), users will not feel significant inconvenience caused by execution of the calculations in the Sub block. The users operate on the Main block of our browser. Because the two blocks of the model on our browser execute independently, users can operate a web page as same operation as web applications which are adopted by Ajax technique.

6. Conclusion

We have proposed a new web browser. The browser can update partially a web page that is not adopted Ajax technique. Because the browser communicates asynchronously with web servers, the partial update of a web page is possible while user continue to operate the web page. As a result of experiments, a web page of Yahoo auction, and a web page of MSN map can be updated partially on the browser. Moreover, when a server load is heavy, benefits of our browser have been recognized. In future, the speed performance of the browser will be improved, at same time the correctness of the judging whole updating will be improved.

Acknowledgement

This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Exploratory Research, 18500032, 2006.

References

[1]Paulson, L.D., “Building rich web applications with Ajax”, *Computer*, IEEE, vol.38, no.10, Oct. 2005, pp.14-17.

[2]http://maps.google.com

[3]Sieminski, A., “Changeability of Web objects - browser perspective”, *Proceeding of the 5th International Conference on Intelligent Systems Design and Applications*, Sept. 2005, pp.476- 481.

[4]Li X., Xiaodong Z., Zhichen X., “On reliable and scalable peer-to-peer Web document sharing”, *Proceeding of Parallel and Distributed Processing International Symposium*, 2002, pp.23-30.

[5] Bergasa-Suso J., Sanders D.A.. Tewkesbury G.E., “Intelligent browser-based systems to assist Internet users”, *Education, IEEE Transactions on* , vol.48, no.4, Nov. 2005, pp. 580- 585.

[6]Shigesada Y., Koshizuka N., Sakamura K., “A Kiosk WWW browser for digital museums”, *Systems and Computers in Japan*, vol.34, no.13, 2003, pp.59-70.

[7]Masuda H., Imamiya A., “Design of a graphical history browser with Undo facility, and visual search analysis”, *Systems and Computers in Japan*, vol.35, no.12, 2004, pp.32-45.

[8]http://www.google.com/webhp?complete=1&hl=en

[9]http://maps.google.com

[10] Chawathe S., “Comparing hierarchical data in external memory”, *Proceedings of the 25th International Conference on Very Large Data Bases. Edinburgh,* Scotland, U.K., 1999, pp.90-101.

[11]http://auctions.yahoo.com

[12]ftp://gnjilux.cc.fer.hr