

Page Disabling using DHTML masking for Data Testing

Ravikumar G K¹, Dr. B. Justus Rabi¹, Manjunath T.N², Ravindra S Hegadi³, Archana R A⁴

¹Dr MGR University, Chennai Tamilnadu, INDIA

²Bharatiar University, Coimbatore Tamilnadu, INDIA

³Karnatak University, Dharwad Karnataka, INDIA

⁴SJB Institute of Technology, Bangalore, Karnataka, INDIA

Abstract: We know that World Wide Web is made of millions of sensitive pages, we deals with an approach to disable pages in the browser after the user performs any action on the screen. This was required not only to enhance user experience but also to ensure that a transaction under process is not disturbed when the user clicks elsewhere on the screen after initiating that transaction. The solution is implemented using DHTML, CSS and JavaScript. The paper briefly discusses the approaches considered for fulfilling the requirement before describing in detail the masking process used. A basic knowledge of HTML & JavaScript is expected to understand the concepts mentioned in coming sections. We hope this will help web designers to implement and analyze these security issues of sensitive pages in the WWW which are very critical to the business.

Keywords: Worldwide web, HTML, DHTML, DOM.

1. INTRODUCTION

More and more legacy systems are migrating to web-based applications. This calls for HTML forms being used for user-interactive screens. There are a number of form elements (text boxes, select boxes, check boxes, submit buttons) on these screens over and above the hyperlinks for navigation. Since these screens are data-intensive, they take time to re-load after the user requests an action (for example selecting a new checkbox or submitting the form). However the user can do other actions before the previous request is fulfilled. This is a vulnerable area since it can result in failed transactions and the process cannot be accomplished successfully. Such a scenario was observed in our work in which the screens were to behave in an intelligent way to avoid multiple submits on the screens. The issue could be handled by disabling the page once the user invoked any action on that page; the page being enabled again only after the action was completed. This meant that the user should not be allowed to start another transaction (be it navigating through a hyperlink or selecting other actions) till the previous action was fulfilled. In other words, the screen had to be blocked from the user while still showing user the contents of the page. However there was no straight forward way by which this page disables or enable could be accomplished. And it was also required that the page be disabled as soon as the user clicks on any form element, so that the transaction under progress is completed properly. This was tested only in Internet Explorer 6.0 +. A number of approaches were considered to provide the desired

experience. The rest of the paper talks about each of these approaches briefly, their pros and cons [6].

2. TECHNIQUES USED

Since the problem required a client-side solution, it was required to use JavaScript and HTML.

2.1 Disabling Form elements

The initial path taken to fulfill the page disable requirement was to iterate over all the form elements in the page and disable them in loop in a JavaScript method [8]. The links in the document were stored in an Array and the links were made behave as dead-end, as is shown in the code below.

```
Function disableForms() {
    IELinkClicks = new Array();
    for (var h = 0; h < frames.length; h++)
    { for (var i = 0; i < frames[h].document.forms.length; i++)
      { for (var j = 0; j < frames[h].document.forms[i].elements.length;
        j++)
        { frames[h].document.forms[i].elements[j].disabled = true;
          }
        }
      }
    IELinkClicks[h] = new Array();
    for (i = 0; i < frames[h].document.links.length; i++) {
      IELinkClicks[h][i] = frames[h].document.links[i].onclick;
      frames[h].document.links[i].onclick = deadend;
    }
  }
}
```

Fig-1: Function Disable Forms

This required looping through the frames in the document & within each frame looping through the form elements & hyperlinks. When the page was to be enabled, the same loops were used to re-enable the form elements & the Array object was used to re-activate the links. But this method was not fool-proof; when the user clicked on a hyperlink with the page disabled, the link would not function but it would stop the previous request from being fulfilled; this lead to no result coming out of the request. A large number of form elements could also pose performance problems since looping through these takes some time (even though miniscule) & user could click elsewhere. Because of these flaws, the approach was modified to provide roundabout solution. Checks in JavaScript methods (which performed the hyperlink tasks) were made to prevent the deadlock. Even though this approach provided the desired

functionality, an alternative solution was thought of, which could be more direct in tackling the issue [4].

2.2 Masking using DHTML

The masking approach uses a kind of transparent layer which screens the entire page when the appropriate JavaScript method is called. Since this meant that each of the form elements didn't have to be disabled individually in a loop, it was faster than the previous approach. When the solution was being discussed there was an issue to decide on the height and width of this layer since the page content was dynamic. However, the scroll Height & scroll Width attributes of the document. Body object came to rescue which gave us the height and width of the page on-the-fly [12]. The Masking approach was finally adopted for disabling pages in the application. However, the first approach was used in part to disable some form elements because of the limitations discussed later. The next few sections talk about the steps involved in the approach in detail.

2.2 Disabling Form elements

The following DHTML technologies were leveraged to provide a solution to the problem:

a. The Document Object Model (DOM)

The HTML DOM defines a standard set of objects for HTML, and a standard way to access and manipulate HTML objects."The W3C Document Object Model (DOM) is a platform and language neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document" [13].

b. JavaScript

JavaScript allows us to access and manipulate all of the HTML DOM objects. The JavaScript methods can be invoked every time the user clicks a form element or performs an action on the page.

c. Cascading Style Sheets (CSS)

CSS is used to style HTML elements. The CSS properties let us control the style and layout model for HTML documents. The attributes of an element such as position, color, visibility, etc. can be defined using CSS styles.

2.3 Solution

The masking of the page is achieved using an empty <table> element which is placed at the end of the page (in the jsp). It is initially made invisible by marking the CSS property visibility as hidden. The table is provided with a unique id, which lets us invoke this element using DOM.

The empty table which acts as a mask:

```
<div id="mask" style="visibility: hidden; position: absolute; left: 0px; top: 0px; height: 40px; width: 780px">
<table border="0" height="0" id="tableLayer" cellspacing="0" cellpadding="0" style="visibility: hidden; position: absolute; left: 0px; top: 0px; height: 40px; width: 80px">
  <tr><td></td></tr>
</table>
</div>
```

The table element mentioned above is invoked using document.getElementById("<id name>") in the JavaScript method written to disable the page. This gives us a handle to the DOM object with the specified id name.

```
var tableLayer = document.getElementById("tableLayer");
The CSS attributes of the tableLayer DOM object are then manipulated in the same JavaScript method. The height and width of the table are set to the height and width of the screen, so that the mask covers the entire page. Then this previously hidden table is made invisible. Since the table is empty, this leads the original screen containing the form elements to be still visible, but it is shielded by a transparent layer which doesn't let the user access the text-boxes, buttons & hyperlinks on the page. However, the select boxes are window level components and so they are disabled separately by using the document.getElementsByTagName("select") and disabling them in a loop [6].
```

The complete JavaScript method is given below:

```
Function disablePage() {
  if (document.getElementsByTagName("select") != null) {
    var selectBoxes = document.getElementsByTagName("select");
    for (var i = (selectBoxes.length - 1); i >= 0; i--) {
      selectBoxes[i].disabled = true;
    }
  }

  var tableLayer = document.getElementById("tableLayer");
  if (tableLayer != null) {
    tableLayer.style.height = document.body.scrollHeight;
    tableLayer.style.width = document.body.scrollWidth;
    tableLayer.style.display = "block";
    tableLayer.style.visibility = "visible";
  }
}
```

This method is invoked just after submitting the request. The code below is part of the method invoked by the action initiated by the user:

```
document.ConfigureVehicleForm.action =
"./choosemodel.do?modName=ov";
document.ConfigureVehicleForm.submit();
disablePage();
```

A similar approach is followed to enable the page back to allow the user to perform other actions on the screen. This method will be invoked just after the user's request has been processed & rendered to the browser. The method below shows the page enabling code [15].

```
Function enablePage() {
  if (document.getElementsByTagName("select") != null) {
    var selectBoxes = document.getElementsByTagName("select");
    for (var i = (selectBoxes.length - 1); i >= 0; i--) {
      selectBoxes[i].disabled = false;
    }
  }

  var tableLayer = document.getElementById("tableLayer");
  if (tableLayer != null) {
    tableLayer.style.display = "none";
    tableLayer.style.visibility = "hidden";
  }
}
```

Fig-2: Function Enable Page

The page disable methods can be effectively used when any of the JavaScript events is triggered by the user action, be it form submit by clicking on buttons or navigating through hyperlinks. The user experience is enhanced by changing the browser status bar, to showing user-friendly messages such as "Loading..." while disabling the page. This is accomplished by using window. Status JavaScript method. It lets the user know that his request is being processed & once done the status bar is again changed to "Done" to let the user progress [7].

3. RESULTS

A simple and straightforward approach which does not involve complex logic at the client end. Page disabling operation is almost instantaneous since just a small loop for disabling select boxes is required other than the layer masking. The code is extensible & need not be altered at a later point of time if new form elements are added to the screen. It can be used along with pop-up screens by invoking the method on the opener object to disable the parent page. OnMouseOver events are also disabled which gives the user a feel that the user cannot start another transaction as already one is in progress. We have potential results showing the masking menu and sample application of masking to webpage.

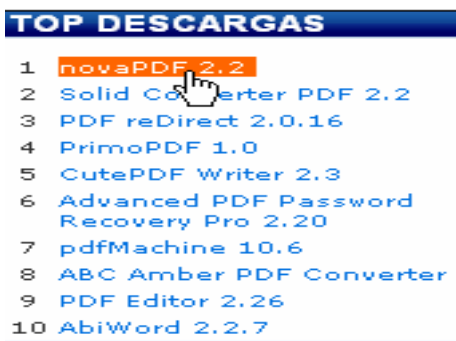


Fig-3: Mask Menu in Css DHTML

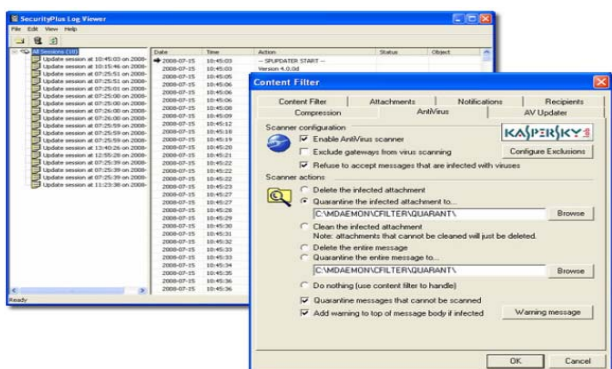


Fig-4: Application of mask to webpage

4. CONCLUSIONS

Since there was no direct way of approaching the page disable problem, we went through a lot of transitions in our solution. The final solution evolved after having undergone a lot of changes, which had to be incorporated in the dynamic requirements of the pages. The masking approach as discussed in the paper fits the problem description well, as it shields the page from any user interaction till the action initiated by user is completely rendered.

The current solution is followed across many of the project screens, and ensures that the transactions, once started, are completed & only then the user is able to proceed with the next transaction. The approach requires minimal coding (now that it has taken a good shape) and can easily be used across any of the pages which require user interaction

5. FUTURE WORK

There needs to be extra provision to disable the select boxes in the iframes. The current code does not disable the select boxes in separate iframes. The approach has not been tested for browsers other than IE 5.5+. There could be potential compatibility issues in other browsers.

6. ACKNOWLEDGEMENTS

Information presented in this paper was derived from Comprehensive conversations with Subject Matter Experts (SME), web design practitioners of various IT companies of India. The authors gratefully acknowledge the time spend in this discussions provided by Mr. Shahzad, SME, CSC USA, Mr. Parswanath Project Manager (Data Warehousing Wing), Wipro Technologies, India. Mr. Govardhan (Architect) IBM India Pvt Ltd.

REFERENCES

- [1].Muralidhar, K. and R. Sarathy "Generating Sufficiency-based Non-Synthetic Perturbed Data," Transactions on Data Privacy, 1(1).2008, p-17-33.
- [2].Li, H., K. Muralidhar, and R. Sarathy, "Assessment of Disclosure Risk when using Confidentiality via Camouflage," Operations Research, 55(6), 2007, p-1178-1182.
- [3]. Muralidhar, K. and R. Sarathy, (2006) "A Comparison of Multiple Imputation and Data Perturbation for Masking Numerical Variables," Journal of Official Statistics, 22(3), 507-524.
- [4].Muralidhar, K. and R. Sarathy, (2006) "Data Shuffling- A New Masking Approach for Numerical Data," Management Science, 52(5), 58-670.
- [5]. Muralidhar, K. and R. Sarathy, (2005) "An Enhanced Data Perturbation Approach for Small Data Sets," Decision Sciences, 36(3), 513-529.
- [6].White paper by Vivek R Gupta, Senior consultant, System Services Corporation, "An Introduction to Data Warehousing".
- [7]. Potts, William J. E., (1997), Data Mining Using SAS Enterprise Miner Software. Cary, North Carolina: SAS Institute Inc.
- [8].Manjunath T.N, Ravindra S Hegadi, Ravikumar G K."Analysis of Data Quality Aspects in DataWarehouse Systems", International Journal of Computer Science and Information Technologies, Vol. 2 (1), 2010, 477-485.
- [9]Manjunath T.N, Ravindra S Hegadi, Ravikumar G K." A Survey on Multimedia Data Mining and Its Relevance Today", International journal of Computer Science and Network Security. Vol. 10 No. 11 pp. 165-170.
- [10] Doan, A.H.; Domingos, P.; Levy, A.Y.: *Learning Source Description for Data Integration*. Proc. 3rd Intl. Workshop The Web and Databases (WebDB), 2000.
- [11] Miles, M. B. & Huberman, A. M. (1994). *Qualitative Data Analysis - A Source Book of New Methods*, Sage Publications, Thousand Oaks.

- [12] M.A. Jeusfeld, C. Quix, and M. Jarke. Design and Analysis of Quality Information for Data Warehouses. In *Proc. of the 17th International Conference on the Entity Relationship Approach (ER'98)*, Singapore, 1998.
- [13]. Wang, R. Y. (1998). A product perspective on total data quality management. *Communications of the ACM*, 41(2), pp.58-65.
- [14]. Ravikumar G K, Manjunath T. N, Ravindra S. Hegadi, Umesh I.M, Cross Industry Survey on Data mining Applications, International Journal of Computer Science and Information Technologies, Vol. 2 (2) , 2011, 624-628.
- [15].Muralidhar, K., R. Sarathy, and R. Parsa, (2001) "An Improved Security Requirement for Data Perturbation with Implications for E-Commerce," *Decision Sciences*, 32(4), 683-698.

Authors Profile



Ravikumar G.K. received his Bachelor's degree from Siddaganga Institute of Technology, Tumkur (Bangalore University) during the year 1996 and M. Tech In Systems Analysis and Computer Application from Karnataka Regional Engineering College Surthakal (NITK) during the year 2000. He is currently working towards his PhD degree in the Area of Data mining and data masking. He has published several papers in International and national level conferences. He is having around 14 years of Professional experienced which includes Software Industry and teaching experience. His area of interests are Data Warehouse & Business Intelligence, multimedia and Databases.

Dr. B. Justus Rabi : was born in Kanyakumari district of Tamilnadu. . He obtained his Bachelor of Engineering Degree from Manonmonium Sundranar University. He has M.E. and Ph.D. Degrees from Madras University and CEG, Anna University, Chennai respectively His research interests include intelligent controllers, electrical drives, signal processing, wavelet transforms embedded systems. and data mining/making He has over 35 international publications to his credit. Currently he is working as an professor in the department of Electrical and Electronics Engineering at M.G.R University, Chennai.



Manjunath T N. received his Bachelor's Degree in computer Science and Engineering from Bangalore University, Bangalore, Karnataka, India during the year 2001 and M. Tech in computer Science and Engineering from VTU, Belgaum, Karnataka, India during the year 2004. Currently pursuing Ph.D degree in Bharathiar University, Coimbatore. He is having total 10 years of Industry and teaching experience. His areas of interests are Data Warehouse & Business Intelligence, multimedia and Databases. He has published and presented papers in journals, international and national level conferences.



Dr. Ravindra S Hegadi received his Master of Computer Applications (MCA) & M.Phil and Doctorate of Philosophy (Ph.D). in year 2007 in computer science from Gurbarga University, Karnataka; He is having 15 years of Experience. He has visited overseas to various universities as SME. His area of interests are Image Mining, Image Processing and Databases and business intelligence. He has published and presented papers in journals, international and national level conferences.



Archana.R.A received her Bachelor's Degree in computer Science and Engineering Engineering from VTU ,Belgaum, Karnataka, India during the year 2007 and Master of Technology in year 2010 in computer science from VTU, Belgaum, Karnataka, India, she is working in SJB Institute of Technology, Bangalore, Karnataka, India. she is having 3 years of Experience. Her area of interests are Image Mining, Databases and business intelligence. She has published and presented papers in journals, international and national level conferences.