

# Chameleon: An Adaptive User Interface Design Framework for Web Pages

**Abstract**— from the beginning of computer-human interfaces, designers always designed the user interface assuming that they are going to be used by an able-bodied person, who has typical perceptual and cognitive abilities. Any web interfaces such as website are not beyond this as well. Therefore, a person perhaps with vision impairment or with colour-blindness may suffer in using the user interface present in the web. We believe a solution of this mismatch of the person's effective abilities and assumption underlying the interface design can be achieved by using our adaptive user interface design framework for web pages- Chameleon. This particular design framework allows end-user customize the web pages according to their need and also adaptive nature of the front-end of the web page which changes its appearance based on the traffic and their request on personalized user interface. Chameleon is very easy to use, and allows designers to design web pages without getting deep into adaptive algorithm coding.

**Keywords**—adaptive user interface, web automation, design, framework, web page design

## I. INTRODUCTION

Computer use is incessantly part of our daily lives whether in education, at work or accessing information and even in case of entertainment. So ability of using computers effectively is becoming a very important issue.

Alongside desktop application people nowadays prefer web application. So web has become a great platform for developing applications formerly would have been used for desktops. But all these application interfaces are developed a common assumption have in mind that all the users of these application are able-bodied people. That's why effective use of computer often mostly relies on user interface of computer applications.

Typically, application developers can expect to control each pixel of their application in case of desktop application but with the growing complexity of web pages and standards prevents developers to provide customization for user in an extended level. For this reason, it's possible for desktop application developer to provide personalized user interface but in web platform it is still difficult.

When designing, web designers tries to stick to the standards but in these standards there is no space for people who are not able-bodied. Perhaps a person with defective perceptual or cognitive ability definitely will not be able to use the application effectively. Now it can be expected that each web designers will meet the design requirement of every person.

Chameleon is a design framework with built-in adaptive user interface that we are building. This particular frame work

is for the web designers to give freedom to design web pages in conventional way but with a functionality of customization for the end-user and also give the web pages an adaptive nature for the design element itself which may change according to users changing demand automatically.

Web page designed using Chameleon would have an icon in the right top corner of the homepage, an example of icon placed in a web page is shown in figure 1. Clicking which will lead user to a control panel. This control panel has options for customizing the design of the web pages to some extent, suppose if elderly user are suffering for using a particular web sign-up form since all the instruction are written in a very small font size. Then using this particular control panel user can change the font size to meet their requirement, in this case may be a bigger font size. Thus Chameleon allows user to alter attributes of design elements like font-face, font size, background colour and etc. Sample of the control panel is shown in figure 2.



Fig. 1 Link of the Chameleon control panel in a webpage

Besides, the website will also be able to track of traffic coming to the webpage and count the changes made by user for a particular design element. This information will lead the web page to adapt with the users demand after a certain time period automatically.

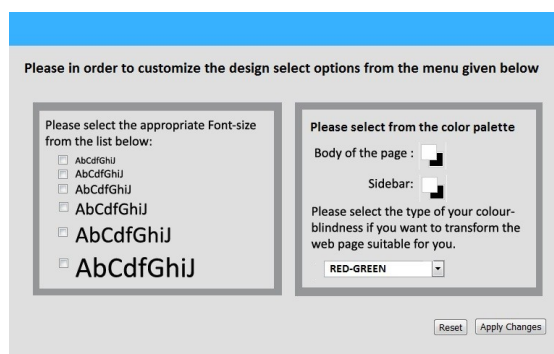


Fig. 2 Sample control panel of Chameleon

For example, if a webpage observes that say in a month duration thousand people visited the page and demanded a particular font size other than the default font size set by the

designer. Then this requested font size will be automatically set as a default by the Chameleon until more users will change it in some other time duration.

Currently Chameleon is a work in progress and ultimate goal of this particular framework is to help people with different disability- people with vision impairment, people with colour blindness or elderly people. The more designers will welcome this design frame work more people will be benefitted from it.

## I. RELATED WORK

There have been a few works done on adaptive user interface to help people. Of them, SUPPLE [3] allows adaptation of interfaces to better serve people with limited motor ability. Here SUPPLE is basically for desktop application and it assumes each design element are known and lays them into different screen and input constraint. But SUPPLE doesn't have much exposure to web interfaces yet.

Then Jiang He and I-Ling Yen [4] presented a work where they developed a frame work to provide adaptive user interface for web services. This framework is to provide a GUI to provide automation of user interface of web services accessed from different devices. But, this framework still doesn't concern with the problem associated with the appearance rather it focuses on the layout of the design and adaptation of the design element in the accessed device.

To enable end-user to automate, customize and to integrate web applications without examining their source code Chickenfoot [5], a programming system for the Firefox browser, was developed in MIT user interface design laboratory. Chickenfoot provide end-user to change appearance of a webpage but to able to do it user has to have prior knowledge of HTML and JavaScript in some level.

## II. THE FRAMEWORK

Initially Chameleon is giving designers to put features in the web pages such as option for changing attribute for instance font-face, font-size, background colour, switching of images. From the beginning we kept in mind that all these features will help people with certain disabilities, for now we are focusing on people with vision impairment and people with colour-blindness.

Visual impairment (or vision impairment) is vision loss (of a person) generally refers to the case where even with eyeglasses, contact lenses, medicine or surgery person cannot see well resulting from disease, trauma, or congenital or degenerative conditions. Of different type of vision impairment, we are concerned with Hyperopic; people suffering from it are unable to see close objects clearly, commonly called far-sighted or long-sighted [1]. Thus these people suffer from the regular UI present in web pages.

Adaptations in lighting or the size of print in the web pages may significantly help people with vision impairment especially with hyperopic. Besides, removing or changing the background of text body can also help them. That is why Chameleon presents features to change background-colour, font-size, and font-face to give better readability.

In a web interface designers almost never consider people with colour-blindness where it can be a very important issue. Typically web designers, attract users by putting different colours in their design, not only that but also for important section of a form usually highlighted in a RED colour but also they put "Warning" or "Required" sign in RED. Now a person who cannot see RED colour, for them it's futile.

About 9% of all people suffer from faulty colour perception, of them less than 1% are female and a little more than 8% are male. There are two type of colour-blindness that are common - red defective and green defective. [2] For this reason we consider only these two types of faulty colour perception, so user will have an option in control panel to select their particular type of colour-blindness. In return Chameleon will swap the colour they cannot see with another colour. For example, a person who cannot see RED colour, then Chameleon will alter the RED colour present in the web pages with BLUE which will be visible to that person easily.

Websites which are designed using Cascading Style Sheet (CSS) have the potential to give end-user to give significant control over the appearance of the website. Our frame work uses this advantage and gives designers to design website using CSS and later on give user control over the appearance and also an adaptive nature to the design elements.

Usually a typical website has header section where it contains a logo, navigation link or menu. Then it has a body section, where it may have two columns or three column formats, where one is the main body and others considered to be sidebars. Here, body contains the most important texts, pictures or other contents, and in the sidebar usually the archives, calendars, or sometimes menus. Besides, website may have a footer section where usually site has links to privacy policy page, developers page etc.

Now, for an elderly user who wants to have a bigger font size which portion of the website interest him/her most? Obviously may be the body section which has the most important section content wise, then may be the menus, sidebars and footers etc. So when it comes to change the font-size it's important to understand which portion of the website is most important, that is why while developing the framework, we obtain partial order of the DOM elements (such as *div*, *p*, *blockquote* etc) thus developed the algorithm accordingly. So giving the privilege of changing text size to users is dependent on these partial orders so in cases some texts presents in the website cannot be altered at all. However,

this work does not provide automation of identifying DOM elements and partial ordering.

In case of changing colours of the pages different partial order is obtained since for a red-green colour-blind person it's more important to see text-filed which are marked red rather than a footer text that says the name of the owner in red font colour .

Now, in short a webpage if there are n design elements such as *body*, *p*, *div* etc and m attributes such as *text size*, *text colour*, *background colour* etc to be changed partial ordering of each element varies with the value of m and n.

This frame work is written using PHP and JavaScript. In the index page where the link to control panel is placed is for the clients. Consequently, the control form is nothing but a simple HTML form leading to a PHP script processing the date input by the user. To avoid any kind of false data input or validation purpose JavaScript is used.

In the application server where PHP scripts are stored work as the main engine of this framework, where all the necessary action such as altering the Cascading Style Sheet is done. However, for each change, application server scripts save the data for distinct design element in the database server and later again evaluated the changes made by distinct users within a time period. For each design element a single attributes default value thus is changed automatically by users if the system crosses minimum expected traffic.

For the test case, when you operate a small experiment we saw that for a single attribute changes made by users are totally random. When plot against attribute value versus time, we tend to get a scatter plot of random variables. So In the algorithm for each attribute expected value is chosen according to the equation

$$E(x) = \begin{cases} \sum_{i=0}^{\infty} P(X \geq i); & T > n \\ K & ; T < n \end{cases}$$

Here, T is number of traffic made changes to that certain design element and n is the number of minimum traffic requirement set by the designer and K is the default value initially set by the designers. Now if  $T > n$  then chameleon will automatically change the default value and it will be

remain until users have it changed again in the next time period. This algorithm can handle infinite traffic and still work efficiently.

### III. FUTURE WORK

In future, we want to help more people by adding new features in the framework. So that in near future, website designed by Chameleon will also help people with motor disability to have a better web experience in both from desktop or portable devices. Besides, we intend to develop a browser independent browser extension to help this end-user to change appearance of any website from a Graphical User Interface (GUI). Nevertheless, this browser extension along with Chameleon will give all type users a better web experience overall.

### IV. CONCLUSIONS

Chameleon, a work in progress, is a novel design frame work for the web designers. This framework allows end-user to customize the web pages according to their need. Besides it also give adaptive nature to the front-end of the web page which changes its appearance based on the traffic and their request on personalized user interface. Chameleon is very easy to use, and allows designers to design web pages without getting deep into adaptive algorithm coding.

Primary goal of the framework is to make this framework feasible for designers to give never experienced before user interface for both able-bodied people and disabled people. Moreover, we also want to make a browser independent add-on to help end-user so that they can change any appearance of any website from a GUI based control panel.

### REFERENCES

- [1] A. Arditi, & B. Rosenthal, "Developing an objective definition of visual impairment." In *Vision '96: Proceedings of the International Low Vision Conference* (pp. 331-334). Madrid, Spain: 1998.
- [2] T. Ohkubo, K. Kobayashi; "A color compensation vision system for color-blind people," *SICE Annual Conference, 2008*, pp. 1286 - 1289, Aug. 2008.
- [3] K. Gajos and D. Weld. "SUPPLE: automatically generating user interfaces". In *Proceedings of the 9th inter-national conference on intelligent user interfaces*, pages 93–100. ACM, 2004
- [4] J. He; I-Ling Yen, "Adaptive User Interface Generation for Web Services," *ICEBE 2007. IEEE International Conference on e-Business Engineering, 2007*, pp.536-539, Oct.2007.
- [5] M. Bolin, M. Webber, P. Rha, T. Wilson, and R. C. Miller. "Automation and Customization of Rendered Web Pages." *UIST 2005*, pp. 163-172.