WEBO-KIT: AN ENHANCED WEB UI TOOLKIT

Sukeshni Kantrod, Kanchan Jondhale, Purva Kshatriya, Namita Maharanwar
UG, Department Of Computer Science, Pune Institute of Computer Technology, Pune, India.
sukeshni.kantrod@gmail.com, kanchanjondhale@gmail.com, kshatriya.purva@gmail.com,
namitamaharanwar@gmail.com

Abstract

Designing UI and backend are one of the main aspects of Web application development. Developers waste a lot time in designing tool like tree view, grid view etc. one by one whenever required as per the requirements of application. Solution to these problems are building UI toolkit library containing commonly required tools, include that library and drag-drop the tool while building application. The existing UI toolkits have less number of tools and do not contain each and every tool, have a lot of ambiguities in their behaviour pattern, look-feel, large in size and have cross browsing problem.

There is a need of powerful UI toolkit which will have all necessary tools, consistent behaviour pattern and light in weight. If we compare same tool from different libraries, they may have different properties which can be combined under one library as a feature rich library.

Our objective is to combine all the properties of a tool from different available web UI toolkits like kendo, dojo, jQuery UI, YUI etc., providing consistent behaviour pattern, look-feel, cross browser support, add new features and light in weight. On the basis of analysis done on various available toolkits, feature rich and configurable library of commonly used tools is being created.

Key words: Cross browsing

I. INTRODUCTION

A good UI is a must for correct interpretation of information by user therefore it should be well mannered. Developers waste a lot time in designing a particular tool whenever required as per the requirements of application. This was very time consuming procedure. Then developers find a library containing collection of all necessary tools. Developer has to just include that library in their application and drag-Drop the tool while building application. But there is no such toolkit which has all necessary controls in it.

Developers first decide which tool they will require in their application. As per the requirement of a particular tool, developers analyze different toolkits to get the best tool amongst them in terms of behavior, look, functionality, etc. but they didn’t get such toolkit satisfying their needs. Therefore, they have to refer at least 2-3 toolkits for building application, resulting in the increase of size of application. Also, this result in lot of ambiguities in their behaviour pattern, look-feel, as the skin of a tool from different toolkit is different.

II. ANALYSIS OF EXISTING TOOLKIT

Every toolkit has its own advantages and disadvantages. The main motto is to create a new library having combination of advantages of all toolkits into one and build that library on top of existing jQuery based libraries. For this purpose it is necessary to analyse all the possible toolkits considering the parameters like size, look-feel i.e. cross browser support. Amongst them, some are based on jQuery and some are non jQuery based. As the jQuery itself is very light in weight, therefore more stress is given on jQuery based libraries. The list of analysed toolkits is as follows:

1. Kendo UI Toolkit from Telerik
2. Infragistics
3. YUI
4. jQuery UI
5. DOJO

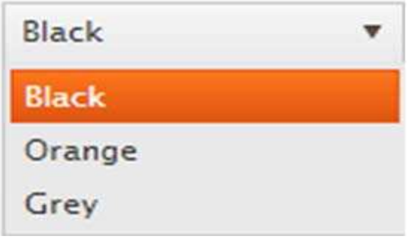
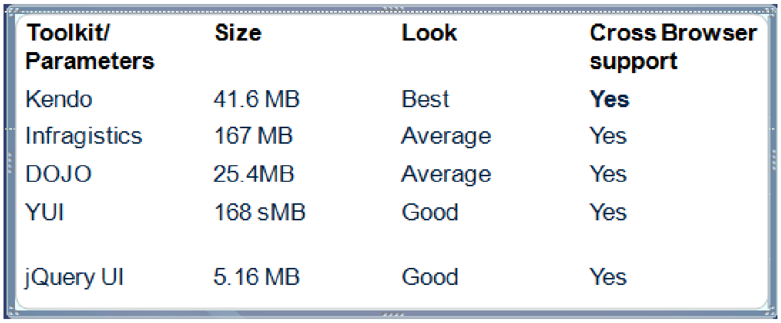


Figure 1: kendo Drop down control

Figure 1 shows kendo's dropdown control where the orange colour is not suitable at enterprise level.

III. COMPARATIVE STUDY



Toolkit/ Parameters	Size	Look	Cross Browser support
Kendo	41.6 MB	Best	**Yes**
Infragistics	167 MB	Average	Yes
DOJO	25.4MB	Average	Yes
YUI	168 sMB	Good	Yes
jQuery UI	5.16 MB	Good	Yes

Table 1: Comparative study of toolkits

According to chart in Table 1, jQuery UI is smallest in size, kendo has better looking in UI, Infragistics and YUI are very large in size and have average looking UI. Jquery UI lacks largely required controls like grid view, tree view etc. it consists of basic controls like button, checkbox etc. All the toolkits provide cross-browser support.

By observing the data mentioned it is clear that kendo toolkit has average size and better looking UI as compared to other toolkits.

IV. WEBO-KIT: THE PROPOSED TOOLKIT

The proposed toolkit has following characteristics -

- Small in size.
- Cross browser support
- Better UI
- Collection of majorly used tools
- Reduced skinning problem

The enhanced library will contain commonly used tools like Grid control, tree view, tab view, modal dialog, buttons, textbox, combo box, radio button, generic web control, check button calendar control, collapsible panel, menu bar progress bar and many more so that the developer will require refer only one library for the application to be developed.

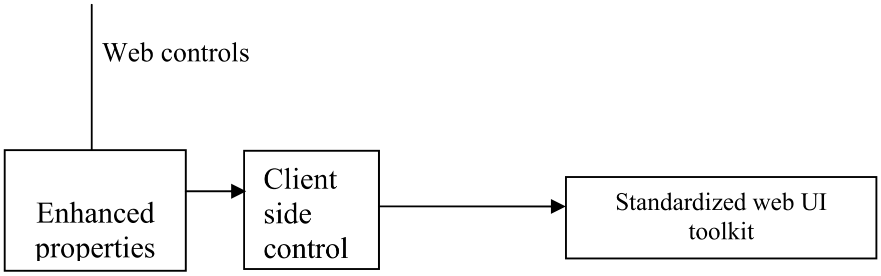


Figure 3: WEBO-KIT

V. DESIGN & IMPLEMENTATION

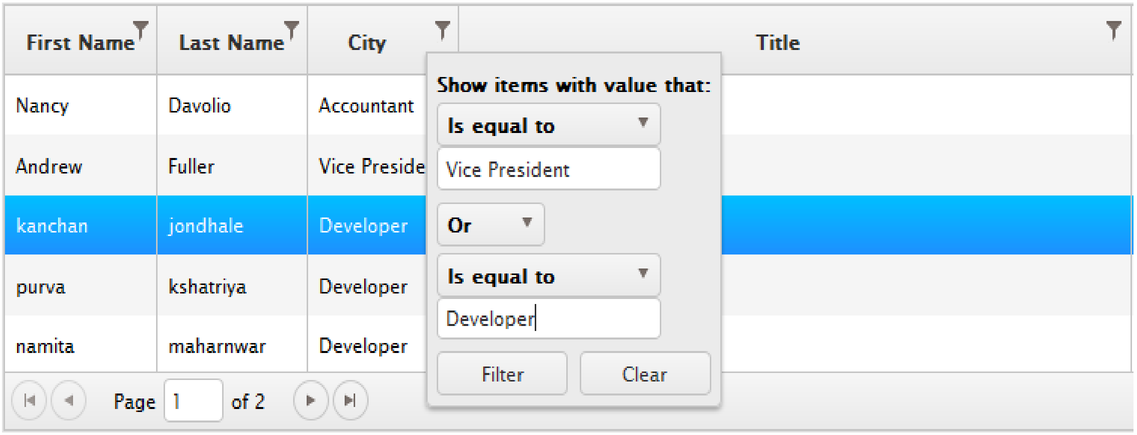
An analysis of a toolkit is done from various available toolkits and the one with the best amongst is chosen as the base for further enhancement of UI properties and features. Some new features that can be added are ascending and descending order, sorting on the basis of character at the start and end of a data word.

Inputs from web developers for proper colour to be chosen for better data interpretation at enterprise level is been taken.

On the basis of result obtained after analysis, a proper colour, feature, properties are added to the basic tool available. ASP.net framework is used for the development of a particular tool. The technologies used are jQuery, HTML5, CSS3, ajax and JSON for developing a tool. The data required by the tool is provided using web service which can provide hard-coded data or can retrieve data from some data storage like MySql database for demo purpose. The web service is located at any server so that any client can have access to it. This makes the data in a tool and UI of a tool separated from each other.

The UI controller i.e. the view of the tool, the data provided are separate from each other. Hence the tools support MVC (modal view controller) form which keeps the implementation and the data source separate from each other.

Reference of web service is provided to the asp.net project as a data provider.



First Name	Last Name	City	Title
Nancy	Davolio	Accountant	
Andrew	Fuller	Vice Preside	
kanchan	jondhale	Developer	
purva	kshatriya	Developer	
namita	maharnwar	Developer	

Show items with value that:
Is equal to: Vice President
Or
Is equal to: Developer

Figure 4: Data Grid View

For example GridView as shown in the Figure 4 the filtering feature can be obtained just by a click on the key shown in column head, blue colour is chosen for interpretation of data selected, a click on the view details button will give all the details of the selected record in pop up form.

The notable thing about Webo-kit is the data shown in tools is retrieved through web service dynamically from database.

VI. PERFORMANCE

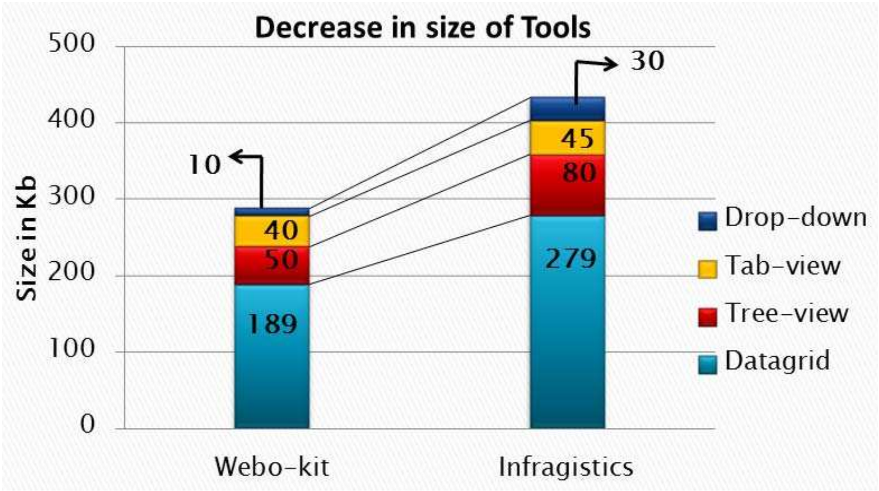


Figure 5: Toolkit

The comparison chart shows the decrease in size of toolkit in regard with decrease in size of grid-view and tree-view, tab-view, drop-down of toolkit designed as compared to infragistics's toolkit. As the size is small the loading time required for application in which this toolkit is used will be small.

VII. FORMAL MODEL

S = {I, O, F, Sc, Fc}

Input (I): {Default.aspx}

Output (O): {library}

Functions (F): {UIdesign (), getData (), addProperties (), createLibrary ()}

UIdesign (): Design UI of a particular tool.

getData (): Get data from webservice.

addProperties (): Add features/properties to the tool i.e. enhancement of tool.

For example in case of grid-view sorting, filtering etc. features are added.

In case of drop-down, checkboxes are added.

createLibrary(): To create a library of tools developed.

Success cases: Su= {Sc1^Sc2}

Sc1 → supports latest browsers (IE 7+, Chrome, and Firefox)

Sc2 → get correct data from web service

Failure cases:

Fc = {Fc1, Fc2, Fc3}

Fc1 → does not support older browsers

Fc2 → no data displayed, if data from web service is incorrect.

Fc3 → no output, if Default.aspx is not compiled correctly.

VIII. CONCLUSION

The proposed modified web UI Toolkit is easy to distribute compared to other toolkits that require installation to function. Web UI toolkit is also useful for creating the templates which will be useful for programmer to develop the application only by dragging these templates.

IX. FUTURE SCOPE

The future research is to provide user-specific customized properties window where user will have a set of skins to be applied to the control. The user will be able to change the skin of control as per his choice.

X. ACKNOWLEDGMENTS

We would like to take this opportunity to thank Professor Atish Londhe (the department of computer science) at Pune Institute of Computer Technology, Pune for constant encouragement and assistance he provided us at every stage of the project. We would also like to thank Dr. G. P. Potdar, Head of Computer Engineering, Pune Institute of Computer Technology, Pune for his encouragement and support.

REFERENCES

[1] Ganji, R.R.Mitrea, M.; Joveski, B.; Preteux, F. HTML5 as an application virtualization tool *Consumer Electronics (ISCE), IEEE 16th International Symposium* on 4-6 June **2012**

[2] Yang Jianping, Ahang Jie Towards HTML 5 and interactive 3D graphics, *Educational and Information Technology (ICEIT),* 2010 International Conference on September **2010**

[3] Mesbah, A. Mirshokraie, S. Automated analysis of CSS rules to support style maintenance. *Software Engineering (ICSE), 34th International Conference* on 2-9 June **2012**

[4] MDN website: https://developer.mozilla.org/en-US/

[5] Kendo website: http://kendoui.com