UNIVERSITY OF WATERLOO
Faculty of Engineering

An Analysis of Using Global Scoped Style Sheets and Atomic Scoped Style Sheets in the Context of Styling Component Based Applications

Yahoo!
Sunnyvale, CA

Aditya Sridhar
3B Software Engineering
Student ID 20539123
User ID a3sridha
September 8, 2016

Aditya Sridhar
201 Lester Street
Waterloo, ON N2L 3G1

September 8, 2016

Patrick Lam, Director
Software Engineering Program
University of Waterloo
200 University Ave. West
Waterloo, Ontario. N2L 3G1

Dear Patrick Lam:

This report, entitled “An Analysis of Using Global Scoped Style Sheets and Atomic Scoped Style Sheets in the Context of Styling Component Based Applications”, was prepared during my 3B work term at Yahoo!, where I worked in the Mail Front-end department.

Description of your co-op position. This should be about three sentences

Description of report and how it related to co-op position.

Acknowledgements

I hereby confirm that I have received no help, other than what is mentioned above, in writing this report. I also confirm this report has not been previously submitted for academic credit at this or any other academic institution.

Yours,

_____

Aditya Sridhar, 20539123

Executive Summary

EXECUTIVE SUMMARY

Table of Contents

1 Introduction

1.1 Background Information

Yahoo! is an internet technology company and is recognized as one of the pioneers of the early-internet era. Yahoo! Mail is a web-based email service owned by Yahoo! with hundreds of millions of users worldwide. The next generation of Yahoo! Mail web client is currently being developed and is built using the ReactJS framework. ReactJS is a component-based JavaScript library used for building user interfaces. The idea behind a component-based framework is that a complex user interface should be broken down into multiple components. Each component should follow the Single Responsibility Principle (SRP), which implies that it should be responsible for single part of the user interface.

Traditionally the development of webpages was based on the principle of Separation of Concerns (SoC) which advocates breaking a problem into different concerns and using a resource to address a particular concern. In the context of web pages, the structure, presentation and behaviour layers were identified as separate concerns with Hyper Text Markup Language (HTML) defining the structure of a web page, Cascading Style Sheets (CSS) defining the content presentation styles and JavaScript (JS) defining how the web page behaves with user interaction. Hence CSS aids in separating the web page content from the web page presentation

1.2 Introduction to the Issue

CSS is a collection of rules, each rule containing one or more selectors and a block of styles. A selector is used to identify a particular HTML element in the web page content and the styles specified in the rule are applied to that particular HTML element. The following figure better elaborates ...

One of the main concerns of using CSS is maintainability. As the content of the web page grows, CSS selectors are required to be more descriptive to target a particular element in the web page. The number of rules in a stylesheet also increases. This significantly affects the developers in the team and hampers scalability from a design standpoint.

This report compares global-scoped style sheets with component-scoped style sheets in the context of styling component-based projects. It first provides some additional information about the issue and lists a set of design constraints and the evaluation criteria. It identifies the benefits and

pitfalls of each accepted solution. The report uses AHP to perform quantitative analysis of the alternatives against a set of evaluation criteria. The report then concludes by identifying the best solution and provides some recommendations.

2 Report Body

2.1 Design Constraints

A solution should adhere to the following design constraints, in order to be deemed as an accepted solution:

2.2 Design Criteria

The following evaluation criteria are used to compare the two accepted solutions:

1. **Redundancy**:
2. **Style changes should be intuitive and predictable**
3. **Low specificity**
4. **Decouple markup and styles**
5. **Cacheability**

1) Redundancy

2) Style changes should be intuitive and predictable

3) Low specificity

4) Decouple markup and styles

5) Cacheability

2.3 Accepted Solutions

2.3.1 Global scoped

2.3.2 Component scoped

2.4 Alternative Not Considered

3 Conclusions

CONCLUSIONS

4 Recommendations

RECOMMENDATIONS

Acknowledgements

ACKNOWLEDGEMENTS