# Allowing Responsive Web Modules

Ben Trovato
Institute for Clarity in Documentation
1932 Wallamaloo Lane
Wallamaloo, New Zealand
trovato@corporation.com

G.K.M. Tobin
Institute for Clarity in Documentation
P.O. Box 1212
Dublin, Ohio 43017-6221
webmaster@marysville-ohio.com

Lars Thørväld
The Thørväld Group
1 Thørväld Circle
Hekla, Iceland
larst@affiliation.org

## ABSTRACT

## Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

## General Terms

Theory

## Keywords

ACM proceedings, LaTeX, text tagging

## 1. INTRODUCTION

A module is an interchangeable and independent part of a program that typically has a single and well-defined responsibility [1]. Modular programming is a technique to reduce complexity and enable reusability. In order for a module to be reusable it must not assume in which context it is being used.

Responsive Web Design (RWD) is an approach to make the application design respond to the viewport size, in order to support varying devices. This is achieved by using CSS media queries to define conditional style rules.

The problem is that there is no way to make a module responsive without it being context-aware, due to media queries only being able to target the viewport. Thus, a responsive module using media queries is layout dependent and has therefore limited reusability.

The desired behavior of a responsive module is having its inner design responding to the size of *its frame* instead of the viewport. Only then is a responsive module independent of its layout context.

This can be achieved with the theoretical feature *element queries* that enables conditional CSS rules by an arbitrary element size. This paper presents a novel implementation of element queries in JavaScript, and discusses the new possibilities of GUI design that our implementation enables.

## 2. EXAMPLES OF BROKEN RWD TODAY

- MQ is not the solution to RWD. (MQ was not designed for RWD as the feature was released long before RWD)
- All elements adapt their inner design by the viewport width.
- Menu Example shows how MQ are broken.
- Limitations of MQ regarding font-size (em).

Media queries were designed to enable developers to conditionally design content by the media, such as using serif fonts when printed and sans-serif when viewed on a screen. Therefore, it is only applicable for RWD of non-modular static applications. In a world where no better solution than media queries exists for RWD, changing the layout of a responsive application becomes a cumbersome task.

Imagine an application that displays the current weather of various cities as widgets, by using a weather widget module. The module should be responsive, so that more information such as a temperature graph over time is displayed when the widget is big. When the widget is small it should only display the current temperature. Users should also be able to add, remove and resize widgets.

Such application cannot be built with media queries. Since the widgets can have varying size, the module cannot change design by breakpoints relative to the viewport. To overcome this problem we must change the application so that widgets always have the same size. If we also assume that all widgets always have a fixed percentage, for example 25% of the viewport width, it is possible to make the module responsive by using media queries.

The problem now is that we have removed the reusability of the weather module, since it may only be used in applications that grant it 25% of the viewport width. Also, if we decide to add a vertical menu to the application we need to change the media queries of the module. In more complex applications such change might result in changing hundreds of media queries. Even worse, if the menu is supposed to hide on user input the responsiveness of the module breaks since the layout changes dynamically.

Additionally, it is popular to define breakpoints relative to the font size. Media queries can only target the font size of the document root, limiting the functionality drastically. With element queries, breakpoints may be defined relative to the font size of the targeted element.

As we can see, even with limited requirements there still are significant flaws with using media queries for responsive modules.

# 3. REQUIREMENTS OF A SOLUTION

- Parents should decide the layout of their children, and the children should adapt their inner design accordingly.
- Valid language syntaxes (HTML, CSS, JS).

First, a solution must enable developers to change the design of an element by its parent size. Elements should automatically respond to changes of the parent size so that the correct design can be activated for each size.

Second, a solution must conform to the syntax of HTML, CSS, and JS so that the compatability of tools, libraries and existing projects is retained.

# 4. WHY IS A NATIVE IMPLEMENTATION TROUBLESOME?

- Performance issues.
- Cite Tab Atkins of RICG (he states that it is infeasible to standardize this).

# 5. A JAVASCRIPT IMPLEMENTATION

- Why is this pragmatic? Compatability, no impact (performance, language) on apps that do not need responsive modules.
- Satisfies the requirements for a solution given above.
- Present Elq's API.
- Present the performance.
- Note drawbacks (but only drawbacks for added functionality!).

# 6. DISCUSSION AND SUMMARY OF RELATED WORK

- Performance, APIs, Features.
- The mirror functionality of Elq makes it uniquely suitable for nested modules.

# 7. CONCLUSIONS

- Production ready.
- Probably no standard (or not in a long time).

# 8. ACKNOWLEDGMENTS

This section is optional; it is a location for you to acknowledge grants, funding, editing assistance and what have you. In the present case, for example, the authors would like to thank Gerald Murray of ACM for his help in codifying this *Author's Guide* and the **.cls** and **.tex** files that it describes.

# 9. REFERENCES

[1] D. L. Parnas. On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, 15(12):1053–1058, 1972.