

ActiveSchematics AutoRendering Mobile Device Optimization

Ferdinand Kusuma^{1,a}, Mieke van Vucht^{2,b}, Raymond de Vijlder^{3,c}

¹Department of Informatics, Petra Christian University, Surabaya, Indonesia

^{1,2}Information and Communication Technology, Fontys University of Applied Sciences, Eindhoven, The Netherlands

³Bosch AA-AS/EDS-WIR2 Manager, Robert Bosch B.V., Breda, The Netherlands

^aferdinand.kusuma94@gmail.com, ^bm.vanvucht@fontys.nl, ^craymond.devijlder@nl.bosch.com

Abstract. In this modern era, technology is one of our main necessities. Technology is always evolving and changes from day by day. One of the major developments in technology is the increasing usage of mobile devices for personal use or even for company use. AA-AS/EDS-WIR2 is one of the sub departments in the Robert Bosch Company that creates applications for the automotive market. The department has created an application named ActiveSchematics AutoRendering. This application is used in car workshops to help mechanics to repair car defects by rendering the car schematics. This application is constructed with an ASP.NET framework combined with a standalone Delphi program as a plugin. The front-end development is done using HTML5, CSS, JavaScript code, and the back-end development is done using C#. The schematics are SVG-based with HTML as the container file. At present, the rendered schematics are not compatible when opened from a mobile browser, because the legacy support for interactivity (for example zooming and panning) is implemented without any mobile support code. The company wants the application to become a mobile device friendly application. The research proposal to be addressed is to solve which front-end libraries can be used to help the SVG schematics to implement mobile-based gesture compatibility. After some trial and error experiments, mobile device improvement will be implemented by adding jQuery, Hammer.js, jQuery PanZoom, and Bootstrap library. The experimental results have been successful and the company can have the mobile device friendly feature for their clients. This research shows that jQuery and hammer.js can be implemented on an SVG object to become a mobile device friendly schematic.

Keywords: AutoRendering, ActiveSchematics, jQuery, Hammer.js.

1. Introduction

Robert Bosch GmbH or Bosch (the nickname that people mostly know) is a German multinational engineering and electronics company that was founded in 1886 by Robert Bosch. Their products consist of automotive parts, power tools, security systems, home appliances, engineering tools and electronics. The headquarters of the company is located in Gerlingen, Germany [1].



Figure 1. Company logo

The company has round 375,000 employees (December 2015) and has spread to around 50 countries in the world. Their ISO standards are both ISO 9001 certified (quality) and ISO 14001 certified (environmental protection). In addition, their management is compliant with OHSAS 18001 [1].

1.1 Company Organization

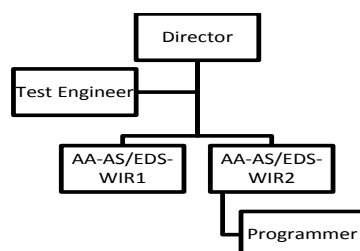


Figure 2. AA-AS Robert Bosch Organization in Breda

Figure 2 shows the organization chart in the Bosch Breda office.

2. Project Overview

2.1 Initial Situation and Problem

ActiveSchematics Auto Rendering has an ability to render the data schematic for a specific vehicle. Currently, the Active Schematics Auto Rendering works only on desktop browsers. As the world is increasingly using mobile devices, the client wants the application to support the mobile device environment as well. However, the code that is constructed in the application does not support the mobile device environment and so needs to be changed.

The benefit of the mobile device development is that it will help workshop engineers to access their schematics rendering application more easily and quickly, rather than using a desktop or laptop.

2.2 Schematic Overview

The application works by rendering the data schema and generating it into HTML files. After generating the HTML files, the application will show the schematic by embedding it into the iframe tag from the HTML code. The schematic will be drawn with <svg> tag from HTML.

Figure 3 is an example of the rendered schematic. The HTML file is contained in <iframe>.

The workshop mechanic can interact with the schematic in this environment.

Figure 3. Rendered Schematic Example

The main development focus is the front-end part of the application. The company wants to renew their old code feature that does not work on mobile devices, and also to add more features in their front-end schematic code.

- All of the interaction events have to be the same for desktop and mobile. (For example, the user needs to double click a component to show the menu. This should be the same as double tapping a mobile device to show the menu.)
- Zooming (swipe on a mobile device and scroll on a desktop) and Panning (drag and hold)
- Highlight specific connected wires in the schema
- Show tooltip information
- Show a menu of the components

The company application needs to be released every 2 weeks, therefore the company wants agile scrum as the project methodology for the assignment. The agile scrum methodology makes the development phase flexible, especially when some features will need to be updated and improved within 2 weeks, before the new release.

The assignment will be chopped into a number of sprints. The company will hold departmental scrum meeting every week, to check the department's performance. There

2.4 Tools and Programming Language

- HTML5, JavaScript, C#, and ASP.NET for the application development
- Perforce (Helix) for the repository [4]
- JIRA as the communication board
- Visual studio 2013 for the C# development
- Notepad++ for HTML5 and JavaScript development

There are 3 areas to be researched for this project:

3.1 Mobile Device Target

3.2 Development Libraries

- Manipulation object library (jQuery [5])
- Event handler library (jQuery Mobile [6])
- Tooltip library (tooltipster 3.0 [7])
- PanZoom library (jQuery PanZoom)
- CSS frameworks (Bootstrap [8])

manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write Java Script [5].

jQuery is chosen to answer the expendability and maintainability problem in the company. The other libraries are jQuery plugin beside Bootstrap as the CSS framework.

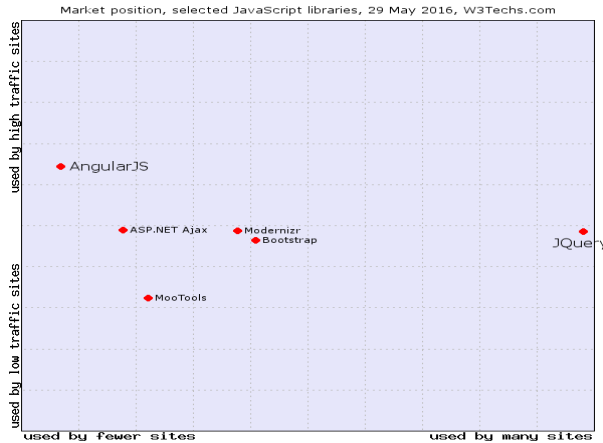


Figure 4. Market Position JavaScript Library [9]

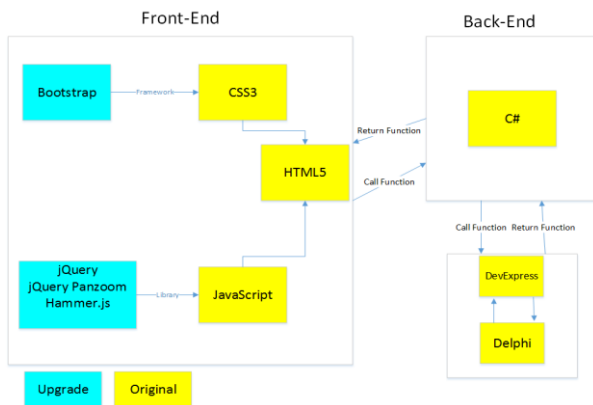


Figure 5. Technology Diagram

Figure 4 shows the popularity of jQuery use. Popularity will ensure a large community also. It can minimize the time spent to find bugs and also minimize the time to get the tutorial spread across the internet.

Unfortunately, there are incompatible libraries on the list. In the development process, jQuery Mobile was replaced by hammer.js [10] and tooltipster was replaced by a custom tooltip without library.

Figure 5 shows the detailed architecture and improvement on the application.

3.3 Testing Methodology

The testing method for the application is the manual testing document method.

Figure 6 shows how to test the project. The team will check the file and give confirmation for every working feature and release. The confirmation will be an OK or NOK mark meaning that the feature is not working.

	A	B	C	
1	Description:	Mobile Feature	Date:	1-5-201
2	Todo number:		Administrator:	Ferdinan
3	Test Script			
4			1	
5				
6				
7				
8	ID	Action	Status	
9	OS related			
10		Android	OK	
11		iOS	OK	
12		Firefox (Windows)	OK	
13	Product related			
14		Tooltip & Highlighting	OK	
15		Zoom In	OK	
16		Zoom Out	OK	
17		Reset Zoom	OK	
18		Menu	NOK	
19				
20				
21				
22				

Figure 6. Test Document Release

4. Development Process

4.1 Main Feature Table

The main part of this project is to add mobile device event behavior for the schematics. To update and renew the rendered HTML file, the code from C# function should be changed and the new generated HTML schematics tested.

Table 1. Old and New Feature

Category	Old Schematics	New Schematics (Mobile Device Support)
Menu Show	Single Click	Double Click/Tap
Hover Tooltip	Mouse Over on the component and wait for a second	Mouse Over on the component without any waiting time
Highlight and Hold Tooltip	None	- Single click on a component - Remove highlight with single click also
Panning Schematics	Drag and Hold with mouse	- Hold and drag with mouse/tap
Zooming Schematics	Mouse wheel	- Mouse scroll/pinch

Table 1 shows the differences between the old and new schematics event behavior. All the C# functions to generate the HTML file have been changed into this new schematics by adding library and renewing all the JavaScript code. Therefore, all the schematics will have these new features and mobile device support.

4.2 Schematic Menu

Menu identification is provided from an xml file. Each Wire will know which menu and sub-menu is defined for it. Implementation is done from the back-end of the application. Read the xml file from C# and generate each menu in the rendered html schematic.

To create the event, double tap hammer.js event is used, so that the menu can be accessed by double clicking a component.

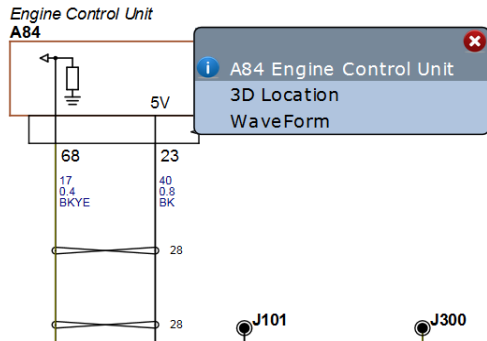


Figure 7. Schematic Menu

Figure 7 shows the menu after a component is double clicked. The menu gives the option to move to another feature or page.

4.3 Schematic Highlight and Tooltip

Highlight and tooltip will give information for the user to know which wire is being accessed currently.

In the original product, highlight is implemented, but it is only a hover event from css. To improve this, highlight and tooltip must always be shown on the screen when needed. To implement the feature, single tap event will call the class identifier for each component.

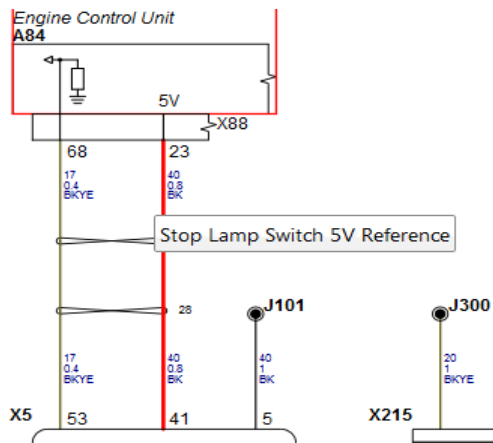


Figure 8. Highlight and Tooltip

Figure 8 shows the red highlight feature with a single click and the tooltip information for the schematic. Menu show, Hover tooltip, Highlight are constructed by jQuery selector combined with hammer.js event for a mobile device.

Panning and zooming is finished by adding jQuery PanZoom library into the JavaScript.

4.4 Dimension Fix Position

The menu and tooltip position is wrong after users zoom or pan the scheme. The position should be near the clicked/tapped object. This issue led to more research about the

detection of click and tapping from the event handler main library. In the first development, this menu position handler was detected using jQuery function syntax to get the x and y axis from the cursor, which is different when another event handler such as hammer.js library is used as the main event handler. The code below shows the first function to get X and Y coordinates from an object.

```
rect = $('#container').position();
cords = getCanvasCoords(e.clientX - rect.left,
    e.clientY - rect.top, pan);

$('#toolTip').html($(this).children('title')
    .text());
ShowToolTip('toolTip', e, cords.x, cords.y);
```

The solution is to use the right function handler from the main event handler libraries. In this case, the library was hammer.js event handler library. The code below shows the right way to get X and Y coordinates from hammer.js event.

```
var coords = getCanvasCoords(
    e.gesture.center.x - rect.left,
    e.gesture.center.y - rect.top, pan);
```

4.5 Thin Wire Touch Support

A scheme consists of components. These components consist of wire objects and non-wire objects. The problem is when a user wants to tap (to highlight/show the menu) the thin wire object. Finger tapping is not as accurate as mouse clicking, and it is hard to give specific input points for the handler code to detect the point from finger tapping activities.

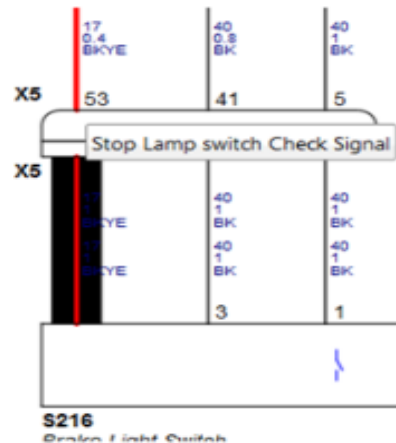


Figure 9. Touch Improvement Illustration

Figure 9 shows the solution to this problem. Add an invisible rectangle on each of the wires like the rectangle in Figure 9. The purpose of the rectangle is to be the selector class/id of the event handler. The rectangle will increase the tapping and click range for each component. The black rectangle in Figure 9 is for illustration purposes only.

4.6 Grade-X View

GRADE-X application uses the rendered schematics from the ActiveSchematics AutoRendering application. The

container for the schematic is the same as the original application (using iframe).

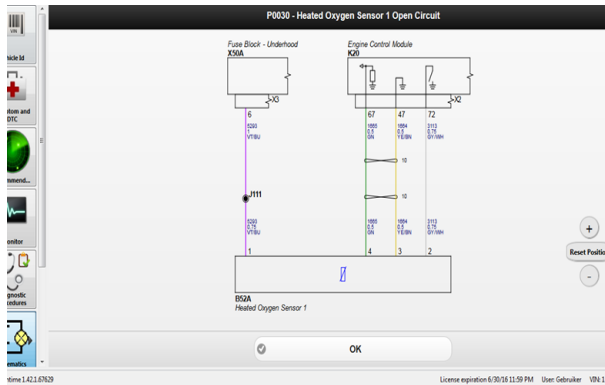


Figure 10. GRADE-X Preview

Figure 10 shows a GRADE-X preview in the schematic section. This is the phase where tooltip library is not working and has been changed to a custom tooltip.

4.7 Position Button

To help a user without a mouse wheel, the position button is developed. Implementation is done using jQuery PanZoom event. This group button consists of 3 functions - zoom in (+ button), reset position, and zoom out (- button).

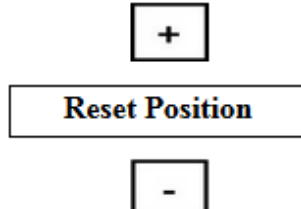


Figure 11. Position Button

Figure 11 shows the position button design. It is located on the bottom right of the schematic.

4.8 Wire Color Keeping Method

The wire color changes after the highlight state has been removed. The wire color should not change because each color has a specific meaning.

The solution is to add a dummy line with the correct highlight color and zero opacity in exactly the same position for each wire. Implementation is done by adding a recursive identifier for specific wires to add a rectangle.

With this method, the developer does not need to change the original wire attributes for highlighting especially the stroke color. The logic is just to change the fake line opacity to overcome the real line for an event.

5. Web Testing Process

In the testing process, the main objectives are Unicode test and W3C validator test.

5.1 Unicode Character Test

There is a string to test the Unicode compatibility. The string will be shown in the schematic tooltip. This is an example of the string:

Testing «\u03b2\u03b1»: 1<2 & 4+1>3, now 20% off!

The string contains HTML-sensitive characters, ASCII, upper-half ISO characters, and multi-byte Unicode characters

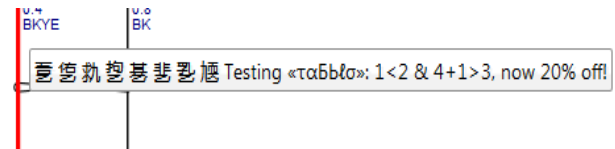


Figure 12. Unicode Test Result

Figure 12 shows the Unicode test result in the schematic tooltip.

5.1.1 iOS Hover Problem

There is wire highlighting overlap in iOS tablets. The problem is that the native iOS browser/Safari does not ignore the “:hover” action from CSS. Therefore, there is a highlighting conflict when a user taps a component on an iOS tablet.

To solve this, there is a function to remove all “:hover” CSS when the scheme is opened from tapping a mobile device.

5.2 W3C Test

The generated HTML is tested in the W3C validator. Some errors show in the results.

Figure 13 shows some errors from the generated HTML. Unfortunately, while changing the Delphi plugin can solve some errors; these are out with the scope of this project. The company’s official development team will carry out the final solution.

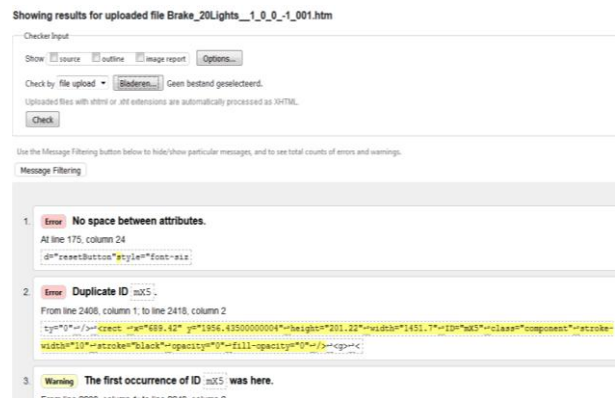


Figure 13. W3C Validator Test

6. Improvement Process

6.1 Position Button Design Improvement

In the company preview, the first design of the position button needs to be improved.



Figure 14. New Position Button Design

Figure 14 shows the new position button design. There is a new feature for this and the user can do panning by holding and dragging this button.

6.2 Company Expert Feedback

At the end of sprint 4, there was some feedback from the schematic expert of the company. The feedback related to a comparison of the behavior of iOS and Android.

The feedback was about the differing performances of iOS and Android devices. This occurred when users were using zooming and panning. Zooming works smoothly on Android devices, but had some glitches on iOS devices. In the overall feedback, experts gave good comments about the research, because the schematics behavior was acceptable to the schematic engineer.

7. Result

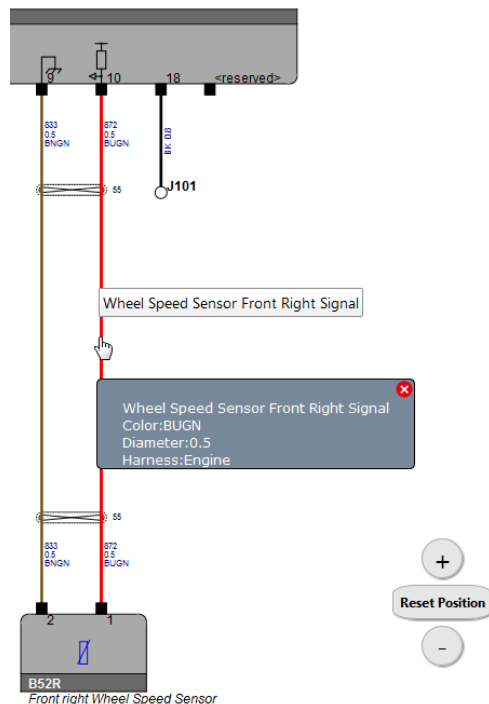


Figure 15. Final Schematic Overview

The final result of the project was to get the schematics working as shown in Figure 15. Some features that should work include tooltip, menu, highlighting, and control zoom button, that are implemented using jQuery, hammer.js, jQuery PanZoom library as the final library to be chosen. This research shows that jQuery and hammer.js can be implemented on SVG objects to become mobile device friendly schematics.

8. Conclusion and Recommendation

During the development process, flexibility was the most important thing. As mentioned in the sprint activities and research above, the application needed to be updated every 2 weeks. Therefore, the agile scrum methodology was the right method to choose. The scrum methodology allowed the development process to become flexible, the developer could improve and update the features in the next sprint after a new release update (after getting the feedback from the upper party or client) and also the scrum meeting helped to solve the problems that occurred in the process.

The recommendation would be to research a better alternative to replace the iframe element in the next development. The iframe environment raises compatibility issues during the development process and therefore a better alternative for iframe should be free from the issues that iframe has.

References

1. Wikipedia, *Robert Bosch GmbH*, retrieved from https://en.wikipedia.org/wiki/Robert_Bosch_GmbH.
2. James, M., *An Empirical Framework for Learning (Not a Methodology)*, retrieved from <http://scrummethodology.com>.
3. Wikipedia, *Scrum (software development)*, retrieved from [https://en.wikipedia.org/wiki/Scrum_\(software_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development)).
4. Perforce, *What is perforce?*, retrieved from <https://www.perforce.com/support/tutorial-video-library/video/what-perforce>.
5. The jQuery Foundation, *jQuery – Write Less, Do More*, retrieved from <http://jquery.com>.
6. The jQuery Foundation, *jQuery Mobile*, retrieved from <http://jquerymobile.com>.
7. Jacob, C. and Ameline, L., *Tooltipster – The jQuery Tooltip Plugin*, retrieved from <http://iamceege.github.io/tooltipster>.
8. Otto, M. and Thornton, J., *Bootstrap – The World's Most Popular Mobile-First and Responsive Front-End Framework*, retrieved from <http://getbootstrap.com>.
9. W3Techs, *jQuery vs AngularJS Usage Statistics*, retrieved from <https://w3techs.com/technologies/comparison/js-angularjs.js-jquery>.
10. Tangelder, J., *Getting Started – Hammer.js*, retrieved from <http://hammerjs.github.io/getting-started>.