Fall 2017

ARIA A11Y ANALYZER: Helping Integrate Accessibility into Websites

Jayashree Prabunathan
San Jose State University

Aria Accessibility Analyzer

A Project Report

Presented To

Department of Computer Science

San Jose State University

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

By

Jayashree Prabunathan

December 2017

The Designated Project Committee Approves the Project Titled

ARIA A11Y ANALYZER: Helping Integrate Accessibility into Websites

By

Jayashree Prabunathan

APPROVED BY THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSE STATE UNIVERSITY

December 2017

Dr. Robert Chun	Department of Computer Science, SJSU
Dr. Philip Heller	Department of Computer Science, SJSU
Mr. Vignesh Sunderrajan	Broadcom Limited

ABSTRACT

Today, nearly 1 in 5 people have a disability that affects their daily life. These varied disabilities can include blindness, low vision or mobility impairments. When interacting with web content, users with such disabilities rely heavily on various assistive technologies, such as screen readers, keyboard, voice recognition software, etc. Here, assistive technologies are software applications or hardware devices that allows users with disabilities to interact with web and software applications. For instance, a screen reader is a software application that navigates through the page and speaks the content to users.

Web accessibility is defined as the ability for assistive technology users to interact and perceive information on a webpage. For example, screen readers are used by users who are blind to read the content on a webpage and to interact with its elements, for example by activating a button. However, this is not always straightforward and easy. Accessibility is generally not a priority for many publishers and developers when building a product. This can lead to difficulties in understanding and perceiving content on the page for assistive technology users. For instance, a retail website without alternative descriptions for images is difficult for users who are blind to “look” for a product and get its information. This can result in the user leaving the website without making a purchase. Until recently, users with disabilities were not part of the usability testing phase. Due to the Americans with Disabilities Act (ADA) and the numerous lawsuits that are being filed on major companies and educational institutions, digital accessibility awareness is growing and more web content designers and developers are building websites with accessibility in mind.

In order to help test the accessibility of a webpage, a number of online applications are

available. This project reviews three such major applications that test for accessibility, and proposes to build an application prototype called ARIA Accessibility Analyzer (AAA). The main aim of AAA is to allow users to perform accessibility tests and remove accessibility barriers in an effective way. AAA is a Chrome browser extension that users can download to manage accessibility tests. These users can include developers, designers, Quality Assurance testers, students and professors. We finally conducted surveys and interviews to understand limitations of these existing technologies and to determine if these limitations have been satisfied in AAA application.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Dr. Robert Chun, for his encouragement, patience, expertise and continuous guidance throughout my project and my graduate studies.

A special thanks to my committee member, Dr. Philip Heller for providing me constructive feedback and support throughout the project. I would also like to thank my committee member, Mr. Vignesh Sunderrajan, for reviewing my work and supervising progress of the project. Finally, a big thank you to the Computer Science Department at San Jose State University for giving me this opportunity. I couldn't have done it without you.

Additionally, I would sincerely like to thank Bryan Garaventa, the founder of WhatSock, and Level Access (formerly SSB BART Group) for providing a vast repository of knowledge that saved me a lot of time.

I also appreciate the interviewees and survey respondents for sharing their time and valuable insights. I am grateful to my husband, who showed his support and exhibited patience during my late night and weekend hacks.

Table of Contents

Table of Figures

1. INTRODUCTION

In simple terms, "**Accessibility**" is defined as the extent to which a consumer or user can access records or retrieve information from an archive, computer system, or website. Consumers and users might be the broadest range of people, including people with temporary and permanent disabilities and older people. Accessibility also means usability for the maximum possible set of specified users is accommodated – **Universal Design / Design for All** [1]. According to the Americans with Disabilities Act (ADA), buildings are required to provide ramps in buildings and Braille on elevator buttons for people with accessibility needs. ADA also requires that web or software applications be accessible to people with disabilities. Applications not accessible to a particular user are not usable by that person [1]. This paper discusses accessibility concerns in webpages and proposes a solution to help web content developers build their web pages with accessibility in mind or allow Quality Assurance testers to look for accessibility barriers. Accessibility barriers are obstacles that make it hard – sometimes impossible – for users with disabilities to do things, such as reading an article or performing a task.

As the web is moving more toward dynamic web content, also called as "Rich Internet Applications" (RIA), it is important that changes in web content, without a page refresh, are indicated to users of Assistive Technologies such as screen readers. Assistive technology (AT) is any item, piece of equipment, software program, or product system that is used to improve the functional capabilities of persons with disabilities. For instance, screen readers are software programs that allow users who are blind or visually impaired to read the text that is displayed on the computer screen with a speech synthesizer or braille display.

If a webpage has a chat dialog open in a corner of the page and a screen reader user is scrolling through the main content of the page, information must be conveyed to the user, without delay, when a new message pops up in the chat window. In this case, the screen readers must interrupt what is being conveyed at that moment and convey the new chat message notification to the user as it takes higher priority than navigating through the web content. If less important changes occur on the page, screen readers must not be interrupted and must wait for the content to be read completely before conveying any new notification. Thus, every important element on the page must be accessible to assistive technologies users.

Each element in a web application has a “name”, “role”, “state” and “property” attribute. ATs such as screen readers will convey this information to screen reader users. This is considered straightforward in a typical static website. However, it is not always very straightforward for dynamic web content. Developers of RIA web applications must introduce accessibility features such as those described in the Web Accessibility Initiative's Accessible Rich Internet Applications specification (WAI-ARIA, or just ARIA). ARIA is a set of attributes that are incorporated into HTML elements and provide the role, state and property of an element on the web page. ARIA is an alternate way to bridge areas with accessibility issues that cannot be resolved by native HTML. The presence of ARIA in the HTML may or may not require JavaScript scripting and it is used to convey the state information or any change to screen reader users.

As illustrated in Figure 1, ‘Page Tabs’ are structures on a webpage that display selected content on the screen without a page refresh. This structure must be indicated to screen reader users to allow them to interact with it easily. Currently, there are no native HTML elements to construct a

Page Tab. Hence it is necessary for screen reader users to make use of ARIA in order to understand the structure of a Page Tab

For example, here are the list of expected behaviors of a Page Tab for screen reader users:

- Screen reader users must know that the structure on a webpage they are on is a Page Tab. For a sighted user, the presence of borders and outline indicate that it is a Page Tab. Whereas, for screen reader users, the Page Tab can be indicated using ARIA roles and attributes. For example, `role="tablist", role="tab", role="tabpanel"` are some of the required ARIA attributes for a Page Tab [3].
- The selected (or currently open) Page Tab must be indicated to screen reader users. For example, in Figure 1, "Section 1" is the currently selected element. Color change is used to indicate the currently selected page tab to visual users. However, an alternate indication for screen reader would be provided using off-screen (visually hidden screen reader text) or WAI-ARIA attributes such as `"aria-selected"` and `"aria-expanded"`.
- Arrow keys need to be used to toggle between the tabs. The "Tab" key must be used to move keyboard focus to the "tabpanel" region and "Shift+Tab" key to move back to the "tablist" region.

With the help of attributes, such as `role="tablist", aria-selected, aria-expanded`, etc., screen reader users will understand the structure on a webpage. They will know that the links on the page are 'tabs' within a tablist and that they can navigate using the keyboard to the "tabpanel" region. Knowing the 'tabpanel' region will help screen reader users understand the start and the end of the content.

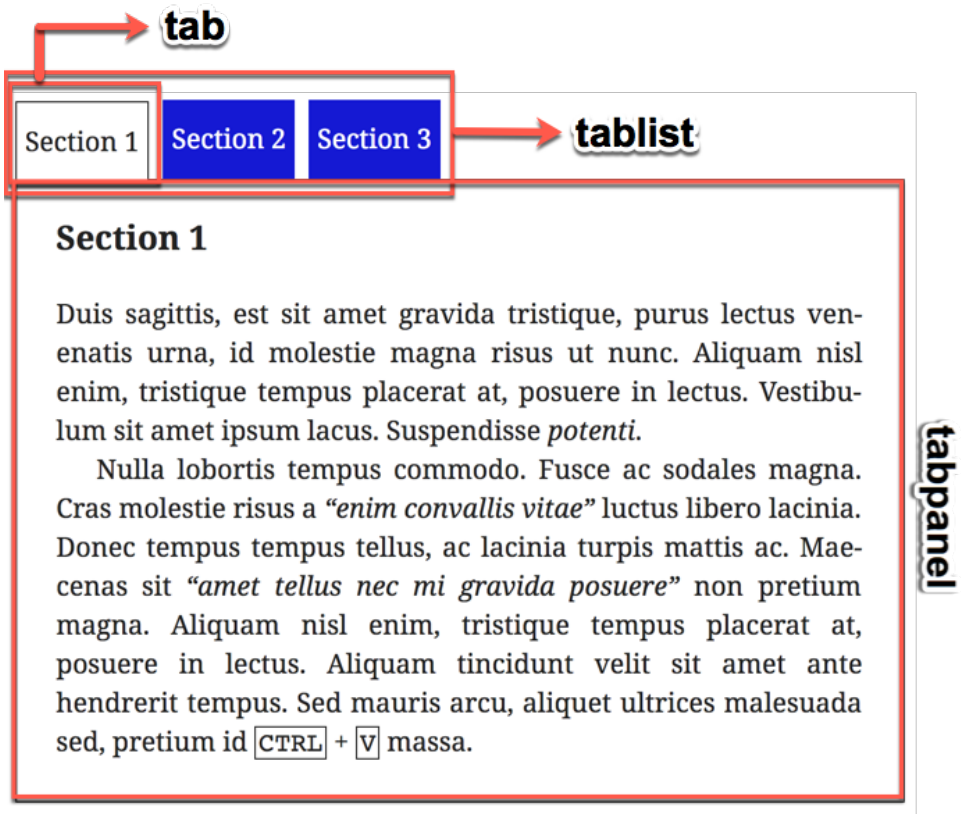


Figure 1 - Example of a Page Tab and its various components.

Currently, the drawback in the development lifecycle is that there are none or very few software engineering methodologies for RIA [2]. We cannot expect developers to design with accessibility in mind if there are no existing RIA design methodologies. Connaghan (2008) explains that RIA will continue to grow and it is of immense importance that accessibility features be introduced early in the development lifecycle. That is, it is recommended that accessibility features be introduced to the application at design time [2] and that Information architect and UX interaction designers also take accessibility into consideration from the very start of the application development.

WCAG 2.0 and Section 508 guidelines are gaining popularity in recent years. Web Content

Accessibility Guidelines (WCAG) are developed by W3C process in cooperation with individuals and organizations around the world, with the goal of providing a single shared standard for web content accessibility that meets the needs of individuals, organizations, and governments internationally. It is primarily intended for web content developers, web accessibility tool developers and others who want or need a standard for web accessibility [12]. On the other hand, Section 508, an amendment to the United States Workforce Rehabilitation Act of 1973, is a federal law mandating that all electronic and information technology developed, procured, maintained, or used by the federal government be accessible to people with disabilities [19].

Currently, since web content developers do not have a specified amount of time allocated for blending accessibility at the beginning stages, they often cannot spend a lot of time researching and testing at a later stage, to ensure that the correct state and role information is indicated to AT users., this paper proposes a tool, “ARIA A11Y ANALYZER” (AAA) that helps solve this problem. Though there are multiple existing solutions that relate to similar needs in accessibility, the scope of testing in these applications does not include deep and corner cases in accessibility. As we’ll describe later, most of these existing solutions are outdated and unreliable. One pro of AAA is that it addresses both WAI-ARIA 1.0 and the latest WAI-ARIA 1.1. Developers can use AAA to test if all the required attributes are incorporated in an element. QA Testers can use AAA to test if accessibility is incorporated correctly. Students can use AAA to test for accessibility in their web projects and instructors can use it for educational purposes. In terms of level of effort and overall cost of fixing accessibility within the application, AAA is very cost effective as the time required to fix issues and the amount of work required is less. This is because AAA is more informative than existing applications and aims to provide all the information the user requires in one place rather than having to look around. All this information and analysis will be studied in more detail in the later topics of this paper.

1.1. PROBLEM STATEMENT

As web content developers are starting to incorporate and test for accessibility, the amount of time taken to find and fix an accessibility barrier on a webpage must be cost effective and timesaving.

The working of Assistive Technologies (AT) is based on its ability to parse the DOM (Document Object Modal) into an Accessibility Tree and to extract an element's state, role and property information. An Accessibility Tree is a hierarchical structure that screen readers or other assistive technologies construct to group and categorize the different elements and its properties on the page. Each element is associated with a role, state, property and name information. When ARIA is present in the DOM, this information is transferred to the Accessibility Tree.

Consider, for example, the elements in Figure 2. If the elements in the DOM are laid out as input field (1) followed by Search button (2) followed by Filter radio buttons (3), screen reader users tabbing through the element will assume that the Search button is the end of the form. This will result in screen readers not knowing that the filter radio buttons exist on the page. This is because the elements in the Accessibility Tree are ordered as (1) -> (2) – (3). The correct focus order would be (1) -> (3) -> (2) for screen reader users to get to all the elements before submitting the form. To change the elements to this order in the Accessibility Tree, the elements need to be changed in the DOM. Another alternative is to use the `tabindex` attribute that overrides the focus order of the DOM on the page based on the value provided.

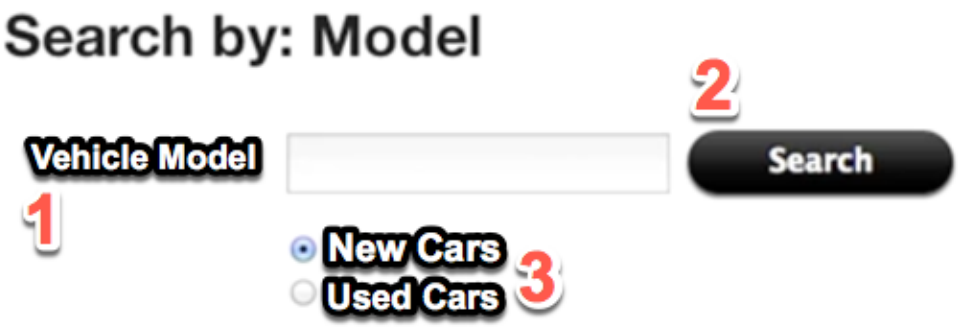


Figure 2 - Example indicating importance of keyboard focus order on the page for screen reader users.

Another example would be headings placed on the page. Headings are one of the most important elements on the page and also the first thing screen reader users look for in a page to understand its structure. At least one heading must always be present on any page. It is also important that the headings levels in the Accessibility Tree be structured hierarchically. That is, headings must be structured consecutively on the page from <h1> to <h6>. If any heading levels are skipped, the structure of the page gets confusing for screen reader users. It is important that headings levels are ordered properly in order for screen readers to understand the structure of the page.

The cause, effect, severity and the location where the violation occurs are some things that users will need to know in order to make improvements to the web content. Unfortunately, existing solutions lack proper indication of information to users. Attempts at improving accessibility on pages are not rewarding. Surveys performed on the existing solutions and AAA reveal the need to provide descriptive information to users. That is, most existing applications provide the location of an error. However, the cause of the error and the effect of that error are

not indicated. The most promising advancement in AAA attempts to overcome these limitations and help improve the user experience.

Hence the two major aims of this project are:

- To measure the usability of ARIA Accessibility Analyzer (AAA) application in comparison to existing applications and to improve web page accessibility testing process. The success of this project is measured in terms of ease of use, learnability, efficiency, memorability, errors and overall satisfaction. Is information provided to allow users to make appropriate decisions?
- AAA must reduce the cost (in terms of Level of Effort) and the accessibility barriers on the page is comparison to existing applications.

1.2. SOLUTION

A number of software applications that test website accessibility are available for consumers but each have their noted drawbacks. To aid the job of web developers or other users of these applications, it is important that information such as the compliance rating, the standards, best practice violations and where any violations that occur on the DOM are provided. It is also useful for users to know the effect of each best practice violation and how it affects Assistive Technology users. Possible recommendations on how to fix the issue must also be provided. Finally, users need to be able to export these results to share it with management or other developers to track improvements.

The solution that this project proposes meets the above specifications. ARIA A11Y ANALYZER

(AAA) attempts to ease the use of the application and allow users to concentrate more on fixing the violations rather than researching the issue.

AAA is a web based Chrome extension that opens as another webpage on the browser and displays the results and accessibility barriers. Displaying the results in a separate webpage allows the application to provide detailed information due to ample space available. Each accessibility barrier is associated to a Best Practice (BP), which are WCAG 2.0 standards. When a particular standard is not satisfied, it is considered an accessibility barrier.

1.3. TARGET USERS

the main target users of this application are web developers or QA testers who are attempting to make their webpage accessible to ATs such as screen readers and screen magnification software. Some other target user groups are students, educators and business management and testers. These target users have varying degrees of knowledge about accessibility. Some users who have good knowledge about accessibility will need minimum information to fix or analyze issues on their page whereas other users with minimal accessibility knowledge will need extra information regarding each violation and possible recommendations to fix them.

Accordingly, AAA has all information structured in such a way that it is accessible to users as needed. Also, users with disabilities would also be the target users of this application. Hence, in addition to providing structured violation information to users, AAA itself must also be accessible. Appropriate accessibility measures are described in the ‘System Overview’ section that has been taken to make AAA accessible.

1.4. BACKGROUND

According to the US Census Bureau [4], approximately 56.7 million people (18.7%) of the US population have a disability of some kind, and about 38.3% (12.6%) have a severe disability [5]. Hence, users with disabilities occupy a large sector of the consumer market.

Nearly 85% of disabled consumers prefer to limit their shopping to sites which they know are accessible, and 81% of them have chosen to pay more for a product from an accessible website rather than buy the same from a website that is not accessible [8]. Many companies and organizations are occupied in expensive and time-consuming lawsuits initiated by consumers, businesses, employees and vendors, because of inaccessible websites [7]. In another survey, only two of the nineteen award winning colleges that provide instructional software responded that they were aware of accessibility issues. About 65% of the remaining seventeen companies were not aware of accessibility as an issue and 100% of them were not currently addressing accessibility as an issue [9].

When the Congress formed the ADA in 1990, the internet had really not developed. So it did not explicitly cover internet accessibility. However, the ADA was built flexibly and expansions were made to accommodate new technologies as they developed [7]. Currently, the regulations are being amended by the ADA to more specifically cover website accessibility.

1.4.1. WAI-ARIA

As accessibility related lawsuits become increasingly common, companies and organizations are taking measures to address accessibility concerns on their web applications. However, as websites also continue to get complex and dynamic, number of accessibility features and

problems started to appear [10]. Additionally, complexity of these problems increases as, currently, many native HTML elements do not support all complicated structures. Page tabs, accordions, menus, and modal dialogs are some examples of structures on a webpage that cannot be made completely accessible using native HTML elements.

The solution to these structural complexities was the introduction of Web Accessibility Initiative's Accessible Rich Internet Applications specification (WAI-ARIA, or just ARIA). WAI-ARIA is an added specification developed by the World Wide Web Consortium (W3C), defining a set of additional HTML attributes that can be used on elements in order to provide additional semantics and improve accessibility. ARIA helps define three important features of an element: role, state and property [10].

Roles simply define what the element is or does. For example, custom checkboxes, which are built without native HTML elements, must ensure that their role is indicated to screen reader users. That is, in addition to ensuring that the JavaScript functionality work perfectly, `role="checkbox"` must also be applied to the custom control. Properties can be used to give them extra meaning or semantics. For example, `"aria-required"` specifies that an input field needs to be filled in order to be valid. Finally, states are properties defining the current conditions of an element, such as `aria-disabled="true"` [10].

ARIA allows us to modify the structure of the Accessibility Tree without necessarily modifying the HTML element's inherent behavior (Figure 3). This means, if a element is present and role="button" is applied, ARIA only changes the role of the . ARIA does not automatically make the element focusable, like a native <button> element, or give it keyboard event listeners.

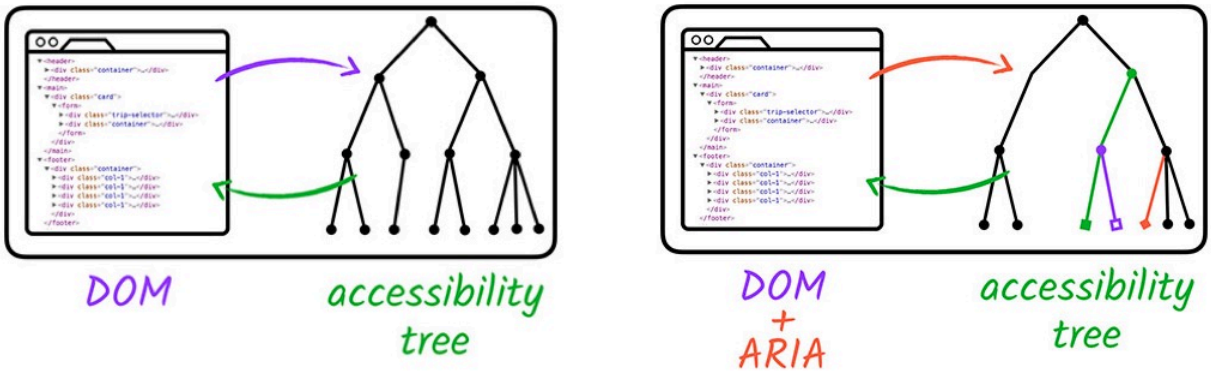


Figure 3 - (Left) Accessibility Tree with native HTML elements; (Right) Accessibility Tree with HTML and ARIA attributes.

WAI-ARIA builds upon WAI-ARIA 1.0 and the most recent version of WAI-ARIA guidelines were published on October 27, 2016 [13]. It deals with more advanced technologies and different types of web technologies. Many roles and attributes were deprecated. A full list of this can be found at the official w3.org report [12].

The next section describes some applications that are currently available to test if WAI-ARIA is implemented correctly. Each of these applications test for accessibility and we will analyze their features and drawbacks.

2. EXISTING SOLUTIONS

In this section, we'll take a look at some of the "existing solutions" mentioned below. We refer to these applications as "Existing Solutions" throughout this project. Not many applications are available to perform automatic tests for accessibility. Amongst the few that are available, the following three applications are some of the widely used ones. These applications only test for certain accessibility barriers and do not provide a platform for users to manage accessibility of their website. Managing accessibility would include things like the ability to export the violations to share and tracking changes while fixing accessibility barriers. Another feature that would be useful is a number (or rating) indicating the current level of accessibility of a webpage. Also, these existing solutions only provide information on where the accessibility features are violated but do not provide any information on what caused it, its effect or what the recommendations are to fix it.

2.1. Visual ARIA BookMarklet

Figure 4 shows Visual ARIA BookMarklet [14] highlighting areas of a page where WAI-ARIA is being implemented. This gives a visual representation of how the ARIA is structured on this page. There are various applications of this - Educators can use this to show sighted students how ARIA is used or sighted developers can incorporate Visual ARIA within the development process to see how ARIA is used within their projects [14].

Also as indicated in Figure 4, Visual ARIA only indicates where ARIA is incorporated in a webpage. Here, the "landmark" region and the aria-describedby attribute is indicated and where they are located on the page.

The drawbacks of Visual ARIA are that there is NO indication of -

- cause of an accessibility violation.
- effect of that violation to AT users.
- recommendation to fix the violation.

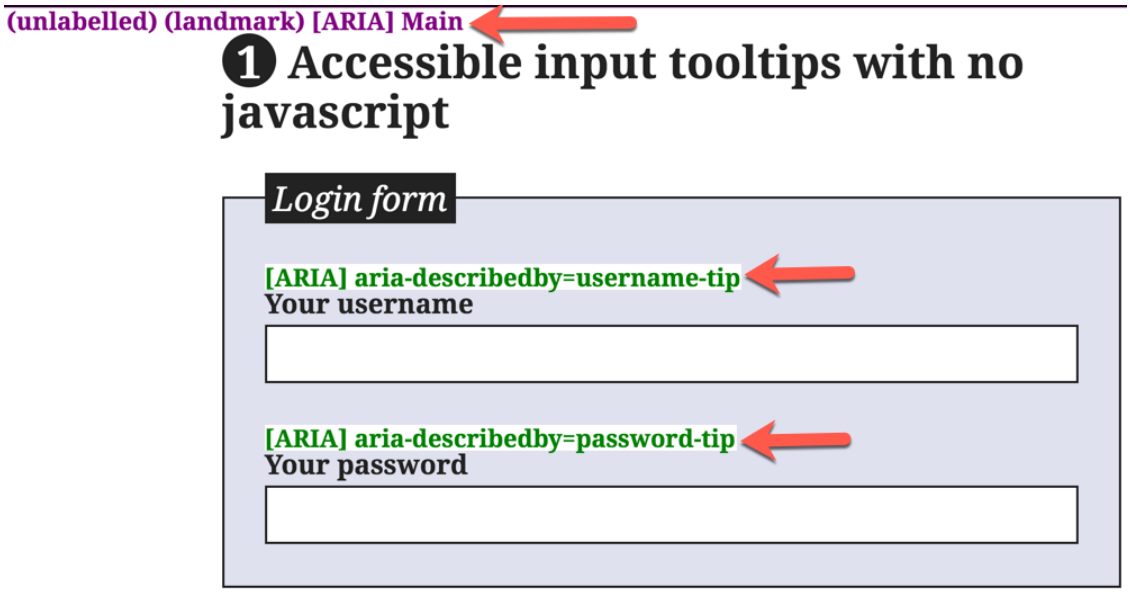


Figure 4 - Visual ARIA outlines all the ARIA roles and attributes implemented on the page.

2.2. ARIA Validator

ARIA Validator [15] is a Chrome extension that also checks for ARIA and provides a pass or fail result for the site depending on how ARIA is being implemented. As indicated in Figure 5, the results of ARIA Validator are shown on a separate tab unlike Visual ARIA Bookmarklet [14]. The drawback of ARIA Validator is that it does not capture all the ARIA roles and properties implemented. Some of the ARIA roles and attributes can be missed and this can cause inconsistencies in results. Similar to the previous application, ARIA Validator can be used to learn how ARIA is being implemented on the page or can be used by engineers to manage

ARIA implementation correctly on their page. Also, there are a number of false positives which is not effective for users. Finally, the cause and effect of an accessibility barrier and the recommendations to fix it is not indicated.

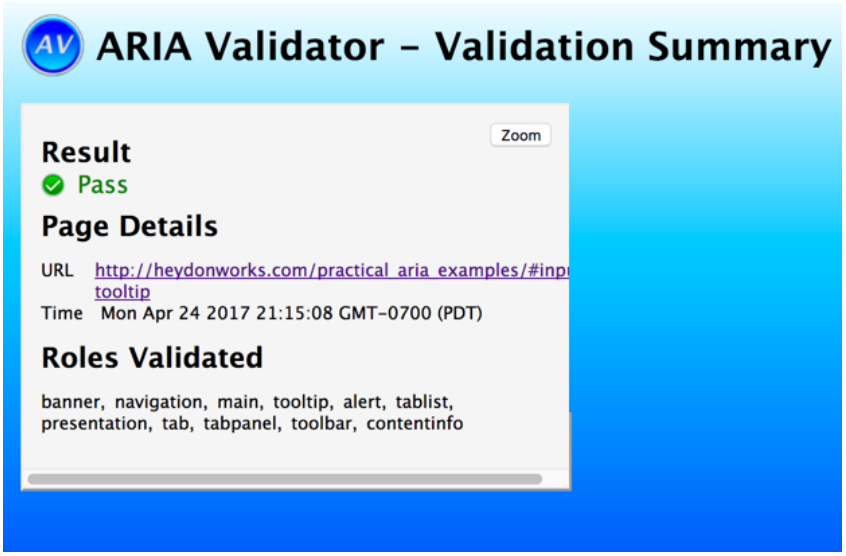


Figure 5 - ARIA Validator provides a list of roles used on the page along with a computed result.

2.3. Khan Academy's Tota11y

Khan Academy has been actively involved in making their website accessible for people with disabilities. In their journey to make their websites accessible, they have developed a tool that will help spread the knowledge that they gained while checking for accessibility to their various teams in the organization, and eventually spreading it across the world. This application is called "Tota11y" and currently, this application is designed to detect images with or without 'alt' attributes, label elements with insufficient color contrast ratio, outlines general structure of a webpage such as headings, improper input fields and labels, labelling ARIA landmarks on a page and detecting unclear link text such as "Click Here" or "More".

That is, in terms of ARIA, this application is similar to what the above two applications do. As indicated in Figure 6, this application provides a visual representation of how ARIA Landmarks is implemented on the page. Some of the common ARIA Landmark roles are ‘Banner’, ‘Main’, ‘Navigation’, ‘Region’, ‘Search’, etc. The purpose of these landmarks are to provide a structural information of the page to screen reader users as well as the boundary of elements. For example, most webpages have navigational links on top of the page allowing the user to easily navigate to the most common page on the website. This list of links are given a `role="navigation"` to help screen reader users understand that these are navigational links and not just any links. Doing this also helps the screen reader know the start and the end of the navigation region. Screen readers will read “Navigation region” when the region starts and “End of navigation region” when it gets to the bottom of the region. Some of the cons of this application includes indicating multiple false positives. Some functionalities are not indicated such as the accessibility barriers or violations.

For example, if form input fields do not have associated labels, it makes it difficult for screen reader users to perceive the meaning of those input fields. The purpose of an input field is visually indicated because of the association and proximity of the text and input field. However, when users who are blind tab to an input field that does not have an associated label, the screen reader will not know the purpose of each form field. This will cause them to enter wrong information in the form resulting in extra time completing the form. In order to associate the input field and the label, the “`for`” and “`id`” attributes are used. Also, Dragon NaturallySpeaking (which is a voice command software used by users with mobility disorder) will not be able to speak the input field’s name to enter information. If an input field is not associated correctly, Tota11ly does not indicate the input field that does not have the associated label. In addition to this, it does not indicate the effect of this for users and the recommendations to fix it. Without

effect and recommendation information, users who have very little knowledge on accessibility will not understand the effect of this to assistive technology users.

One pro of this application is it provides some explanation on where errors occur and recommendations to fix it, which none of the other applications seem to provide.

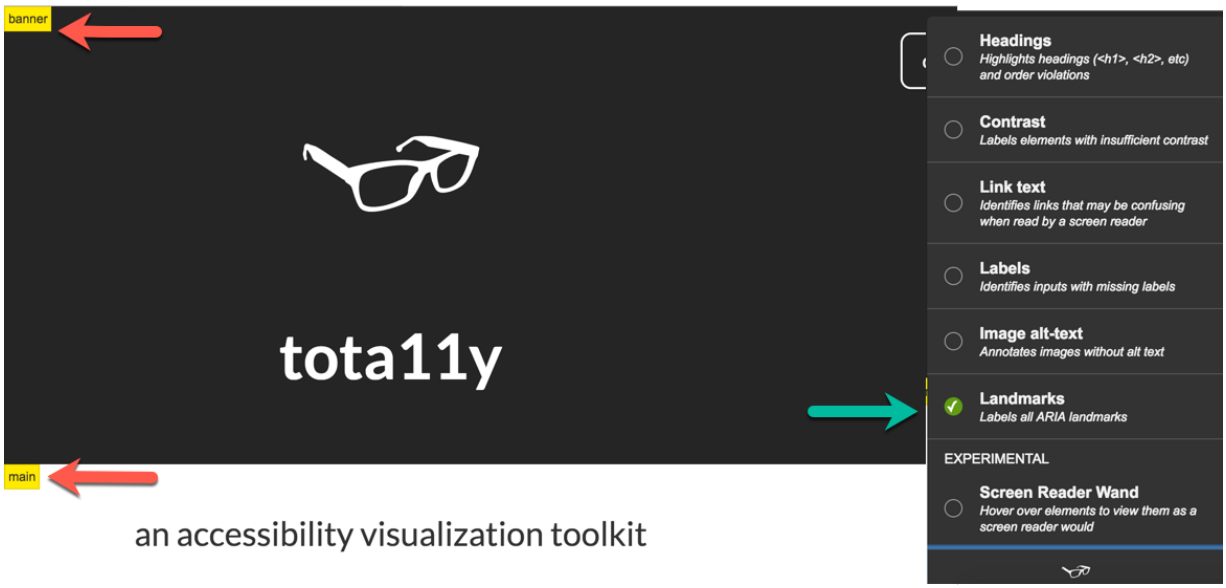


Figure 6 - Tota11y provides a visual representation on where ARIA is implemented on the page.

Based on our analysis of these existing solutions and their limitations, AAA application was designed and developed.

Based on this review and analyzes, AAA application was designed and developed. In the next chapter, we will review the AAA application and analyze if all the drawbacks and limitations of the existing applications were eliminated in AAA.

3. ARIA Accessibility Analyzer

3.1. OVERVIEW

AAA application is a Chrome extension based application that allows users to understand how compliant their website is according to WCAG 2.0 A and AA standards. This application reads the current state DOM (Document Object Model) of the webpage and analyzes the markup to check for any violations or improper use of WAI-ARIA. These violations can hinder the use of the web content for Assistive Technology users. The results of testing the page are displayed in a detailed and organized fashion. The information is structured in such a way that users can choose a way to see information, if required, or to skip past unwanted content.

3.2. PHYSICAL SETUP

The AAA application requires a desktop or laptop computer with the latest Chrome browser (Version 62.0.3202.89). The physical devices needed are the keyboard and mouse. Since this application is accessible to screen reader users and keyboard users, this application does not necessarily need a pointing device such as the mouse. It can be navigated using the keyboard alone.

The AAA extension package can be downloaded and enabled to show up next to the URL Bar. Once the extension is downloaded, the application can be used on the page to be tested.

3.3. USER INTERFACE

The User Interface of AAA is intuitive. Once the application is downloaded, it needs to be opened on the page to be tested. This is done by clicking the extension button on the page to be tested, as indicated in Figure 7.

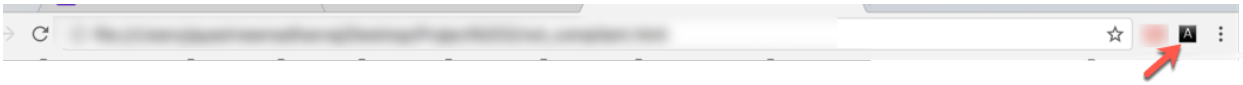


Figure 7 - The AAA application can be accessibility by activating the Chrome extension button.

After this extension is clicked, a new page is opened. This page is the main interface where all the information is provided to users. As indicated in Figure 8, when the user gets to the page for the first time, the page does not provide much information and everything is blank. Only after the developer presses the “Validate” button is the page that was opened in the other tab tested for accessibility. This is designed this way to allow users to choose the URL to be tested. That is, this program is developed to test the first URL in the page tabs. So, only the first URL of the page tabs are tested for accessibility.

The results are displayed on the AAA web page in different sections. Here are the different components of the page as indicated in Figure 8.

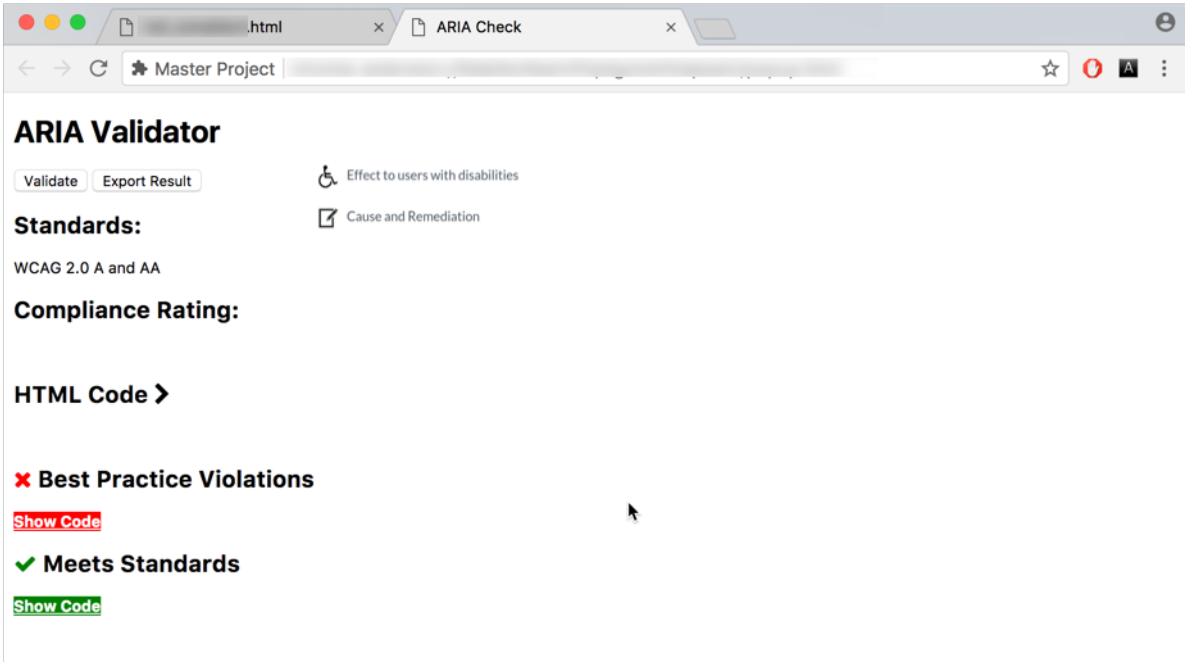


Figure 8 - The AAA main page opens when the extension is activated.

The “Validate” and “Export Result” buttons are located on top of the page as they are the most important functions on the page. They must be visible to users as the first thing on the page. Activating the “Validate” button will cause the program to get the HTML code of the page that needs to be tested (The page that opened the AAA application). As a convenience, this program is also developed to test the page that is present on the first tab on the screen. This allows the user to go to another page and “Validate” the page without having to re-open the application.

The legend on the right side indicates the meaning of the icons that will be displayed next to the violations that will be listed under the “Best Practice Violations”. As mentioned earlier, “Best practices” are WCAG 2.0 standards and a violation is marked when the best practice criteria is not met. The first icon (“Effort”) is for the popup button that opens to indicate the effect of a particular accessibility barrier for assistive technology users whereas the second icon (“Cause”)

indicates the cause and remediation for the violation.

The compliance standards against which the webpage was tested is indicated under the “Standards” section. Currently, AAA checks for WCAG 2.0 standards and in the future will include Section 508 standards.

The “HTML Code” section is an accordion structure that contains the entire HTML code of the page that was tested. This allows the users to perform a DOM inspection and AAA also allows the user to highlight where a particular issue occurs as indicated in the next section.

The “Best Practice Violation” section indicates all the violations or accessibility barriers in the webpage. As indicated in Figure 9, the “Show Code” button allows the user to inspect where in the HTML code the issue occurs. Similarly, the “Meets Standards” sections gives the users an idea of the good practices that have been satisfied. This section also has the option to view the code where the standard is being violated.

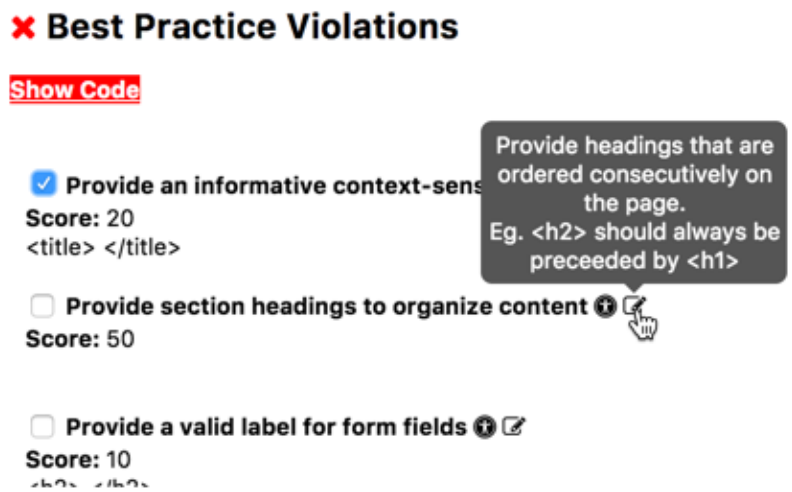


Figure 9 - The checkbox next to each application allows users to view the code in the HTML.

3.4. INFORMATION ARCHITECTURE

As indicated in Figure 10, when the “Validate” button is activated, the “Compliance Rating” score is displayed based on the number of Best Practice violations.

The Compliance Rating is calculated according to the score of each violation and the total number accessibility issues reported. That is,

Compliance Rating = $\left(100 - \frac{Sum\ of\ Severity\ scores}{Number\ of\ violtion * 100}\right) * 100$

If the Compliance score calculated is between 0 – 79%, the compliance rating is indicated in red whereas the compliance rating between 80 – 100% is indicated in green. This provides a way for the users to determine how serious the accessibility is compromised on the web page.

The HTML Code on the page is the current HTML code at that instance when the page was validated. This HTML Code is hidden within an accordion and it can be expanded or collapsed to view and hide content respectively.

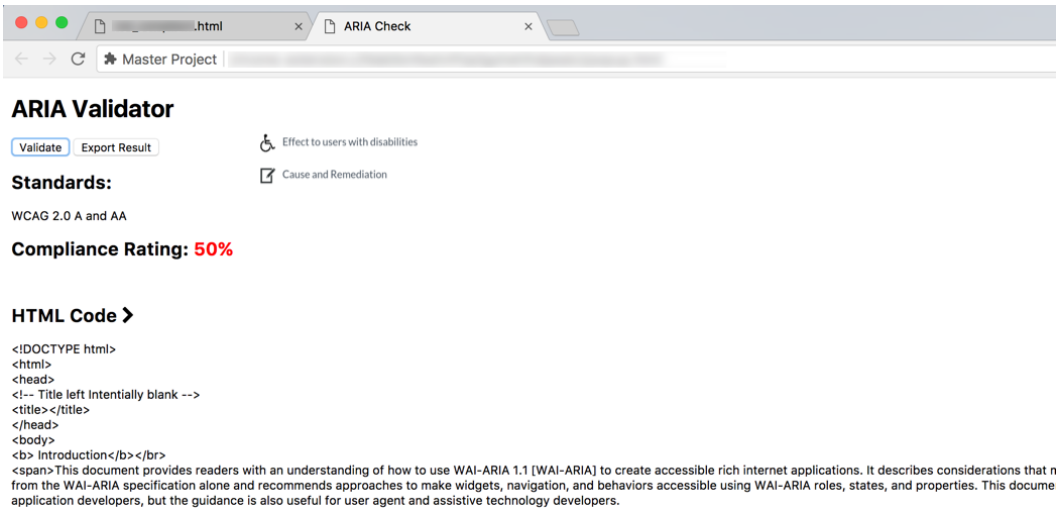


Figure 10 - The results are displayed on the AAA page.

The Best Practice violations are listed on the next section. Each violation within this section is preceded by a checkbox. Users can check the violation that they need to see the HTML Code. On the right of each violation, the icon button to open the tooltip to show the cause, effect and recommendations is present. The information is presented in a tooltip rather than displayed on the page to allow the user the choice to see the information only if required and to ignore if not necessary. This takes into regard, the varying types of users for this application.

Figure 11 indicates the effect of not having headings on the page. Screen reader users fully depend on the headings to navigate through the page and understand the structure of the page. Figure 11 indicates the possible recommendations for the violation. In this case, it recommends having <h*> elements and to organize them on the page in such a way that the heading levels are not skipped. If heading levels are skipped, screen reader users will find it difficult to understand the relationship between the content.

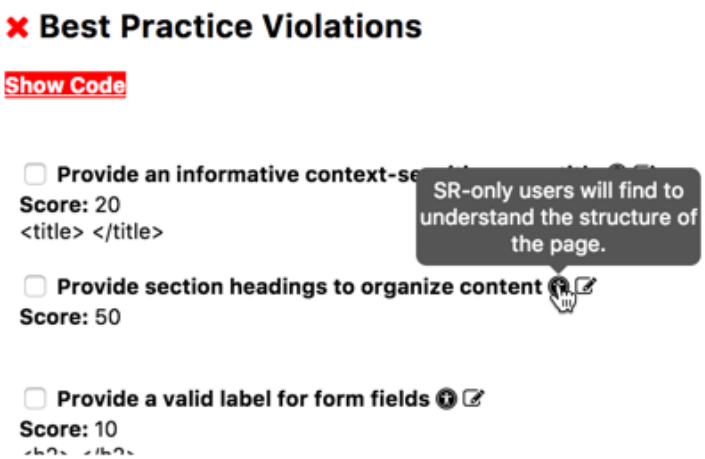


Figure 11 - Additional information is presented to the users within tooltips.

In addition to being an application that tests for accessibility, it is important that the application be accessible. Screen reader users and keyboard users must be able to use the application without any show-stoppers or barriers. Hence, AAA application was built with accessibility in mind. The next section describes some of the accessibility measures taken to make AAA accessibility compliant.

3.5. ACCESSIBILITY MEASURES

In order to meet all necessary WCAG 2.0 and Section 508 requirements, some of the accessibility considerations taken into account while building the application are:

- All form elements are built using native controls and are labelled appropriately. Since the page has buttons and form elements such as checkboxes it is important that these elements and their purposes are correctly indicated to screen reader users. This is done by building all the elements using native HTML controls for these buttons and checkboxes. Also, these elements are programmatically labelled so that their purpose and description is indicated to AT users. For example, Dragon Naturally Speaking (Voice

command software) users can easily speak out the name of the element they want to activate if the element is labelled correctly.

- Headings are used on the page to indicate the structure of the page to screen reader users. Headings are one of the first things that screen reader users look for when they navigate to a page. This gives them a sense of structure of the page. Headings levels must also be carefully followed. For example, a heading level <h1> must always be followed by <h2> which must be followed by <h3>, etc. It is not a good practice to skip the headings as the structure of the page will not be indicated to screen reader users correctly.
- Color must not be the sole means of indicating information. If color is used to indicate information, screen reader users will not be able to perceive this information. It is important that any color used on the page has alternate text equivalent for screen reader users. For example, the AAA application results page indicates both the good and the bad practice violations. Though a colored icon is used to indicate the distinction, there is also text next to the icons to indicate the content.

3.6. MESAURING SUCCESS

The success of the AAA application is measured based on the number of limitations that are present in the existing applications that are addressed in the AAA application. Another important parameter is the efficiency with which information is presented to users and how effectively users can understand the problem and fix it. In other words, the less research required on the users' part, the more the goal of this project is achieved.

In simple terms, the existing applications only indicate "what" accessibility barriers that occur,

and, sometimes, “where” it occurs. However, they do not indicate what is the cause, effect and the recommendations.

In the next chapter, usability studies were performed on AAA and the other existing applications to analyze if AAA is a better accessibility testing application.

4. USABILITY STUDIES

The user needs and expectations are studied using a combination of survey and quantitative analysis. Quantitative analysis is done using surveys on the prototype. The expectations of the target users were identified and compared with existing applications. Their feedback was one of the most important factors used to improve the quality of the AAA application.

AAA mainly focuses on improving factors such as:

- **Speed** – Since most potential target users are web developers, the process of checking the compliance of the page being built must be quick and convenient for them.
- **Cause and Effect of issue** – This is important for users to understand the effect of the issue to users using Assistive Technologies. For example, if an element on a page is designed to be a button without native HTML element but does not have the appropriate keyboard event handlers, this will effect keyboard-only users as they will not be able to access those elements, causing loss of business to these end users. Similarly, an idea of the cause of the violation will also be helpful for developers to know where in the page the issue occurs.
- **Severity of Issue** – A number based on the severity and effect of this accessibility barrier on AT users. This number serves as a parameter while calculating the overall compliance on the page. For the purposes of this project, we use the scales 1 to 100 (1 being lowest severity whereas 100 being the most severe and will have a magnificent impact on the AT users) and the number depends on the impact. For example, a highly visible.

- **Compliance of page** – The compliance is the overall rating on the page. This will be used to compare if any new changes or modification on the page causes any 'new' accessibility issues. If so, how severe is the score effected. Severity (mentioned above) is one of the parameters in calculating the compliance of the page. The compliance and severity numbers provide a measurement on the rating and the level of effort needed to fix accessibility issues on the page.
- **Recommendations to fix violations** – This is important for all categories of users. Users with less accessibility knowledge will need information for quick fixes. Recommendations must be straightforward and easy to implement. Amongst the many possible solutions that is situated for a particular problem, the recommendations that have the lowest tractability (Easiness to fix an issue) will be provided.

The limitations of existing applications were surveyed and the results were grouped. A prototype was then created based on these limitations which was then named the "ARIA A11Y ANALYZER" (AAA). Finally, a follow-up survey was presented to the same target user group to compare the improvements or limitations, if any.

4.1. SURVEYS AND INTERVIEWS

To understand experiences of using various accessibility applications, potential target users were identified and were asked to participate in surveys and interviews. The scope of target users varies from different ages, experience, and different purposes for the application. Since the categories of target users are less, the survey was performed mainly on developers, students, business professionals and Quality Assurance testers. Two surveys were conducted

during this project to identify the overall experience of using –

1. The existing applications such as Visual ARIA, Khan Academy Tota11y or ARIA Validator.
2. The ARIA Accessibility Analyzer.

This gives us a clear distinction between AAA and the existing applications and if AAA has improved on the drawbacks and limitations of existing applications. Google Forms was used to perform the surveys and gather data.

Before the participants had a chance to use the applications, an interview was conducted where few questions were asked. These questions were very informal and mostly cover the expectations that potential users have when using Accessibility testing software and if these applications met those expectations.

4.1.1. INTERVIEW QUESTIONS

We interviewed 13 participants for this project and later, these participants were also emailed to perform the survey. The questions and conversation outline is indicated in the Appendix section (9.1.2). The results for these questions are summarized in the Discussion section.

4.1.2. SURVEY QUESTIONS

First, the participants were asked to install all the existing software applications that were part of this survey. Since all these applications are available to download and use for free, it was straightforward. The only requirement that users had was to have the latest version web browser. For the purpose of this paper, we used Chrome browser. The survey questions that

were given to analyze the existing application is similar to the questions given for AAA. In addition to this, the survey questions for AAA also included questions related to the user interface of the application and to analyze if AAA's design is intuitive for users.

Questions regarding the existing applications:

I. Select the category of target user that you apply to. (Web developer, student, tester, Management, Others.)
II. Select the list of applications that were used to review accessibility features in a webpage. (Visual ARIA, Khan Academy Tota11y, ARIA Validator, Others.)
III. How often do you use these type of applications on a day-to-day basis?
IV. Rate your knowledge on accessibility from 1 (Basic) to 10 (Experienced).
V. Does the existing application provide sufficient information for fixing the violation? (Yes/No)
VI. If "Yes" for the above question, rate the description of the recommendation provided. Is it sufficient to fix the violation?
VII. Does the existing application provide an indication of the effect of the violation?
VIII. If "Yes" for the above question, does that provide sufficient information to indicate the severity of the violation depending on the effect?
IX. How would you rate your overall experience using the existing applications from a scale of 1 (Not Satisfied) to 10 (Highly Satisfied)?

Questions provided regarding AAA:

Related Definitions:

- *Learnability:* Indicates how easy user can learn the main system functionality and achieve the skill to do the job [17].

- *Efficiency:* After learning to use the system, how fast can users perform their task using the systems [17].
- *Memorability:* This reflects how well the user reestablished proficiency with the system functionality after a time period gap [17].

Similarly, following are questions asked relative to AAA application. Most of these questions are similar to the questions above but include additional questions which are more focused towards user satisfaction and usability of the product.

I. Select the category of target user that you apply to. (Web developer, student, tester, Management, Others.)
II. Rate your knowledge on accessibility from 1 (Basic) to 10 (Experienced).
III. How would you rate the learnability of this application from a scale of 1 (low) to 10 (high)?
IV. How would you rate the efficiency of this application from a scale of 1 to 10?
V. How would you rate the memorability of this application from a scale of 1 to 10?
VI. How often do you use these type of applications on a day-to-day basis?
VII. Does the existing application provide sufficient information for fixing the violation? (Yes/No)
VIII. If “Yes” for the above question, rate the description of the recommendation provided. Is it sufficient to fix the violation?
IX. Does the existing application provide an indication of the effect of the violation?
X. If “Yes” for the above question, does that provide sufficient information to indicate the severity of the violation depending on the effect?

XI. How would you rate your overall experience using AAA from a scale of 1 (Not Satisfied) to 10 (Highly Satisfied)?

4.2. RESULTS

4.2.1. INTERVIEW RESULTS

We received very valuable feedback from the informal conversations with the 13 participants during the interview phrase. Since the interview took place before the survey, the aim of the interview was to understand user's expectations for an accessibility testing tool. This also gave the opportunity to discuss topics that were not covered in the survey questions.

Based on the responses received from the participants, the 6 highly expected factors from most users were –

1 The application must be accessible to Assistive Technology Users, such as screen readers and keyboard-only users.
2 The application must take into consideration the level of accessibility knowledge of the user.
3 The application must test both, ARIA 1.0 and the recent ARIA 1.1 attributes.
4 Processing Time of the application must be fast.
5 Less number of False Positives
6 Detailed information about the violation. For example, what causes an issue, effect of that issue and how to fix it.

4.2.2. SURVEY RESULTS

The surveys were conducted successfully and we received valuable feedback from the

participants. 13 people participated in this survey and interview. As illustrated in Figure 12, 61.5% of the participants were Quality Assurance (QA) testers, 15.4% were web developers, 15.4% were students and 7.7% were from the business management. Also, as indicated in Figure 13, 84.6% of these survey takers use such Accessibility testing applications ‘rarely’ and not on a regular basis. Finally, from Figure 14, it is fair to say that the participants have varying degrees of accessibility knowledge. 5 out of the 13 participants have less accessibility knowledge whereas the remaining 8 participants have a fair knowledge on accessibility. Based on these results, we can say that the participants of the survey and interview are a good representation of the potential target users and their experiences using the applications and expectations would account to most users.

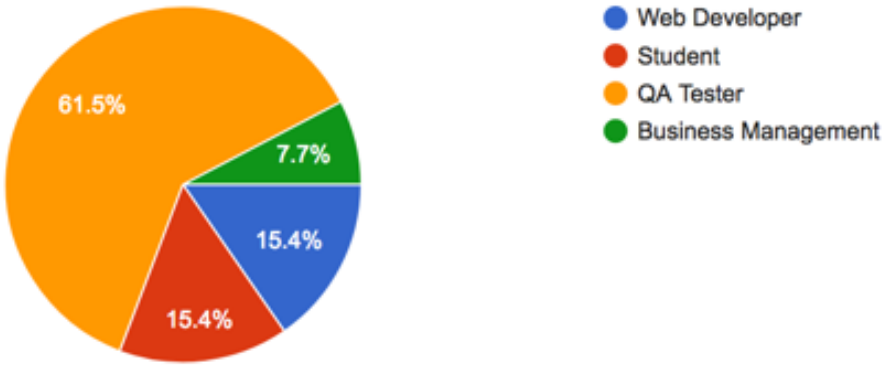


Figure 12 - Survey Result - Target user category.

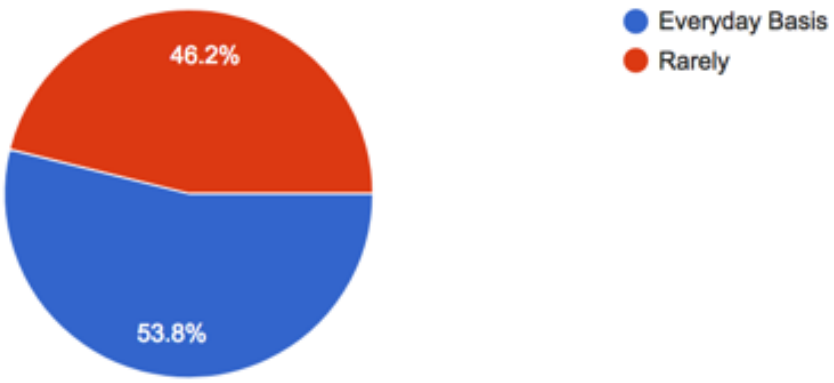


Figure 13 - Survey Result - How often target users utilize application.

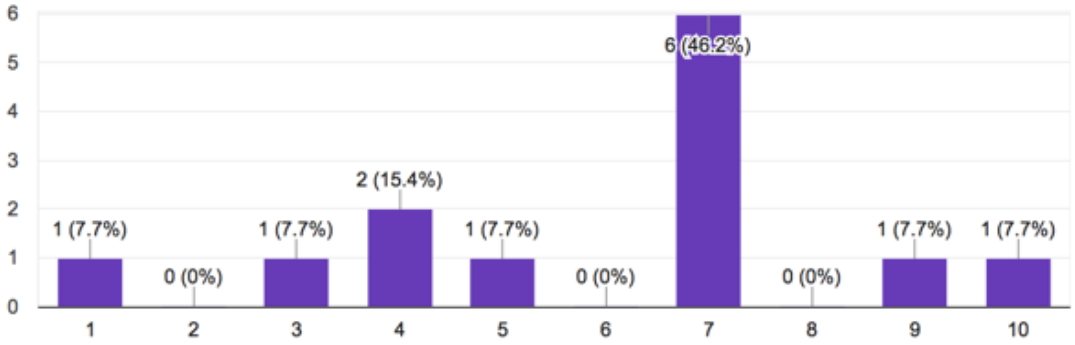


Figure 14 - Survey Result - Level of target users' accessibility knowledge.

Additionally, Figure 15 indicates that most participants chose to test "ARIA Validator" amongst the other available existing applications. Thus, the results and conclusions drawn from the survey questions are a comparison between ARIA Validator (AV) and AAA. These surveys are taken by participants after having a chance to use these applications. The data presented below will indicate the user experiences on "ARIA Validator" and compare if AAA made improvements to any of AV's limitations.

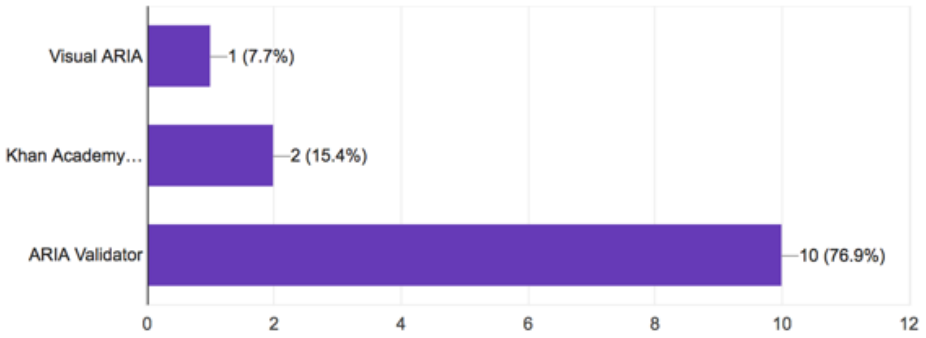


Figure 15 - Survey Result - Application most users downloaded.

The first set of results will compare the similar questions related to the applications –

I. *Does the existing application provide sufficient information for fixing a violation? (Yes/No)*

II. *If "Yes" for the above question, rate the description of the recommendation provided. Is it sufficient to fix the violation?*

From the survey results indicated in Figure 16 (left) and (right), it is clear that ARIA Validator (AV) did not provide sufficient information for fixing a violation whereas AAA does provide this information to users. 76.9% of the participants chose "No" for AV but 92.3% of the participants chose "Yes". As illustrated in Figure 16 (left), the only information AV provides is "what issues could be present on the webpage tested". In AAA, this information is provided alongside the violation (Figure 17 - right) within a tooltip.

Also, the quality of the description is important. From Figure 18 (top), the one user that indicated ARIA Validator provides recommendations to fix an issue rated the description of this recommendation as "low" or "not sufficient". On the other hand, in Figure 18 (bottom), all the 13 participants who chose "Yes" or "Maybe" indicated that AAA provides highly sufficient recommendations.

This concludes that AV makes it difficult and time consuming for users to fix a particular issue. Since recommendations are not readily provided, users will have to spend extra time researching for solutions rather than fixing the violations. This in turn can make their development process and their path to an increased compliance rating slow. In contrast, AAA saves time for users.

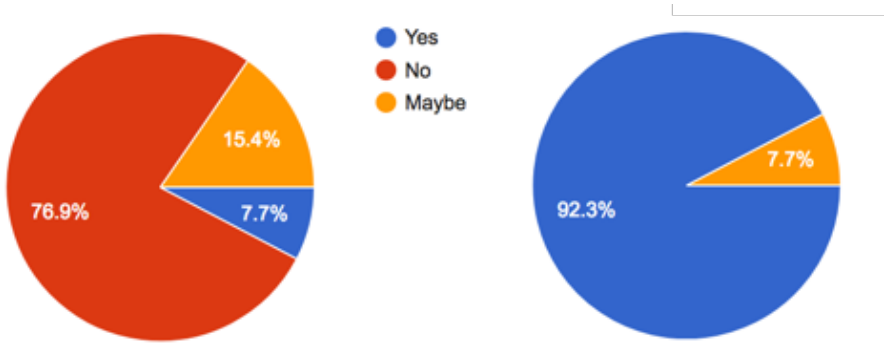


Figure 16 - Survey Result - Sufficient information to fix a violation is indicated on (Left) Visual ARIA and (Right) AAA.

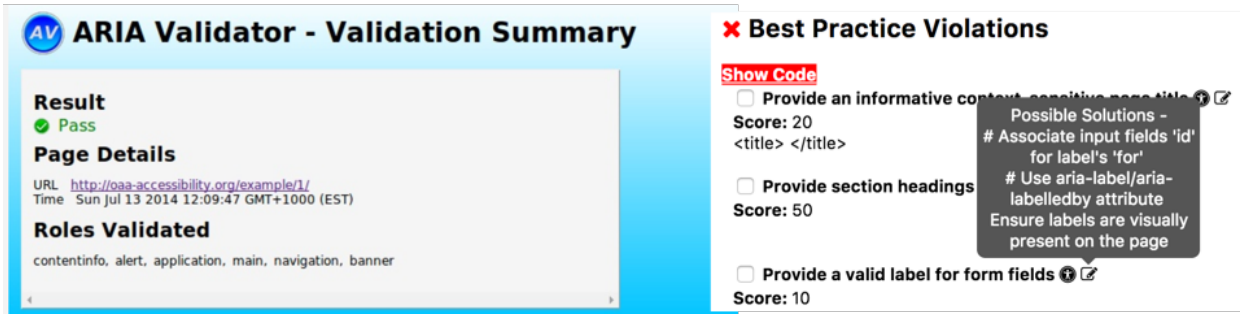


Figure 17 - Indicating recommendation for violation in (Left) Visual ARIA vs. (Right) AAA.

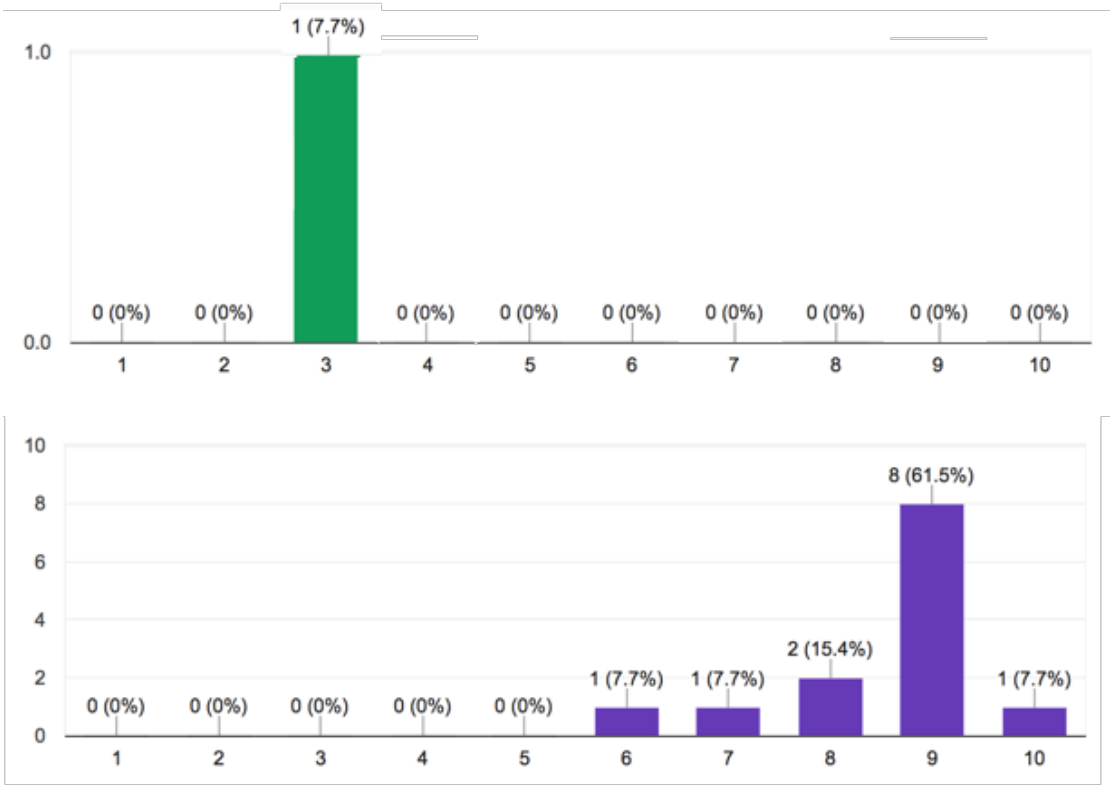


Figure 18 - Survey Result - Rate description of recommendation information for (Top) Visual ARIA and (Bottom) AAA.

III. *Does the existing application provide an indication of the effect of the violation?*

IV. *If "Yes" for the above question, does that provide sufficient information to indicate the severity of the violation depending on the effect?*

From the survey results indicated in Figure 19, 92.3% of the (or 12 out of 13) participants indicate that AV (left) does not provide information regarding the effect of the violation whereas AAA (right) does. As illustrated in Figure 20, AV (left) only indicates the error but does not indicate the effect or severity of the error. On the other hand, this information is indicated in AAA within a tooltip (Figure 20 – right).

It is also important that the description and severity of the error is indicated to users to really understand the impact of an accessibility barrier on a webpage. As indicated in Figure 21, AV (left) no such information is provided whereas in AAA (right), 76.9% of the users indicated that sufficient information regarding the severity and impact is provided.

Thus, we can conclude AAA does a better job indicating the effect of an accessibility barrier on a webpage than AV. Knowing the effect and severity of a violation allows users to understand which ones have priority and which ones have a high impact on assistive technology users.

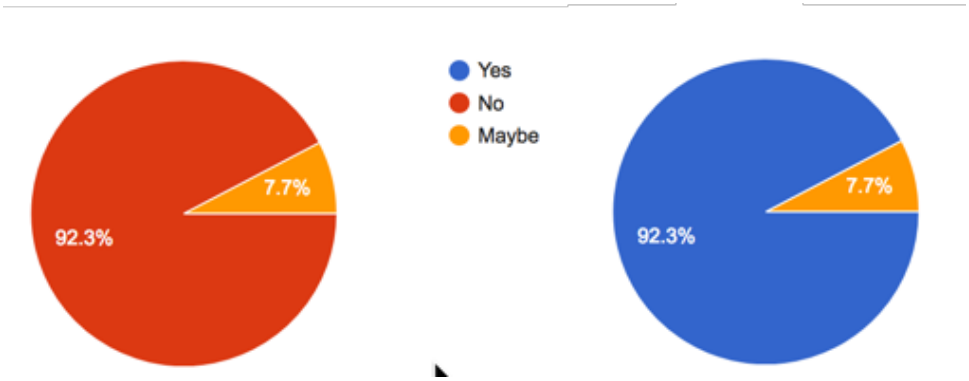


Figure 19 - Survey Result - Indicate effect of violation on (Left) Visual ARIA and (Right) AAA.

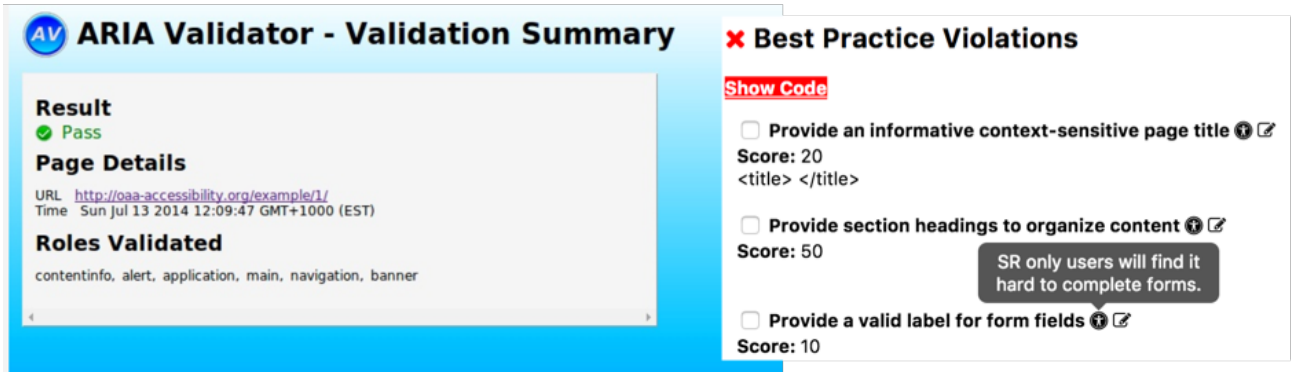


Figure 20 - Indicating Effect of Violation on (Left) Visual ARIA vs. (Right) AAA.

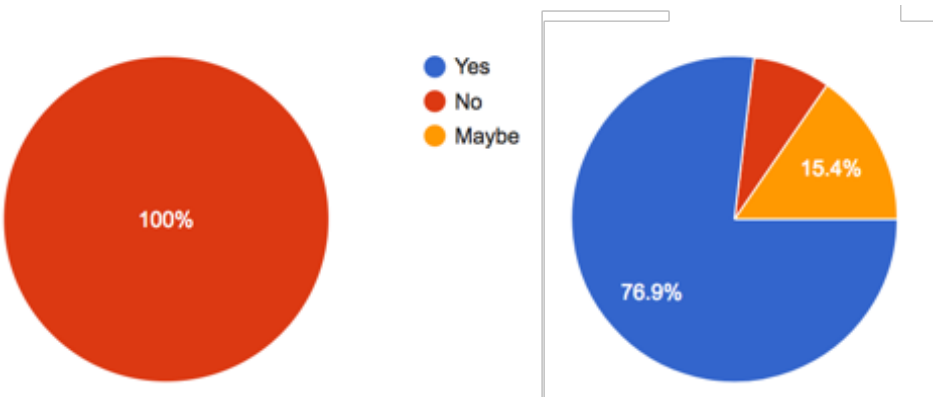


Figure 21 - Survey Result – Is the severity information indicated in (Left) Visual ARIA and (Right) AAA sufficient.

V. *How would you rate your overall experience using AAA and the existing solution from a scale of 1 (Not Satisfied) to 10 (Highly Satisfied)?*

From Figure 22, it is clear that participants were more satisfied with the experience using AAA (bottom) than AV (top). Around 11 participants gave a score of 8 and greater (Highly Satisfied) for AAA whereas around 12 participants gave a score of 5 or lesser (Not Satisfied). The AAA application appears to be more intuitive and efficient for accessibility testing. Additionally, AAA provides information that AV does not such as the cause and effect of an accessibility barrier. This saves time and users have a better experience and satisfaction when using AAA.

Thus we can conclude that the overall experience of using AAA is highly satisfactory compared to AV.

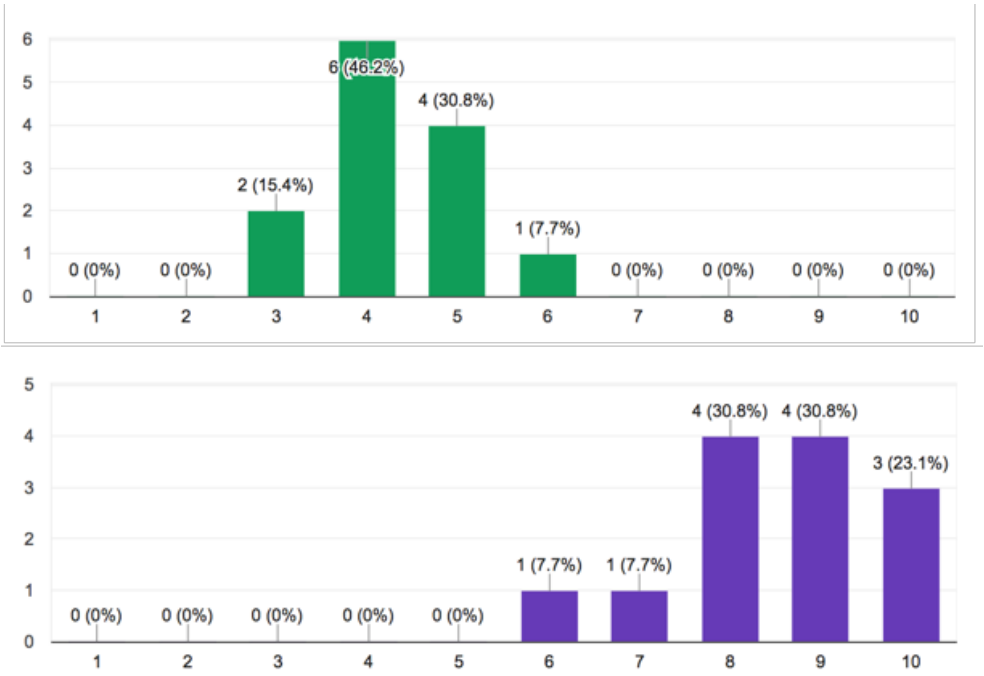


Figure 22 - Survey Result - Overall Satisfaction of (Top) Visual ARIA vs. (Bottom) AAA.

Next, we will analyze the result for survey questions that are applicable only to AAA. Since the user interface of AAA is new and not previously tested, it is important to understand if the user expectations have been met and if the user can easily navigate through the page. The three important factors that are analyzed are “Learnability”, “Efficiency” and “Memorability” of the application.

VI. *How would you rate the learnability of this application from a scale of 1 (low) to 10 (high)?*

VII. *How would you rate the efficiency of this application from a scale of 1 to 10?*

VIII. *How would you rate the memorability of this application from a scale of 1 to 10?*

As indicated in Figure 23, most of the users felt that the ease of use and the **learnability** of the

application was highly satisfactory. More than 92% of the users indicate the application is easy to learn and accomplish tasks. This shows that AAA is intuitive and user interface of AAA is fairly simple. Users do not require a lot of time to get used to. The information that users need is readily available and structured well and easy to find.

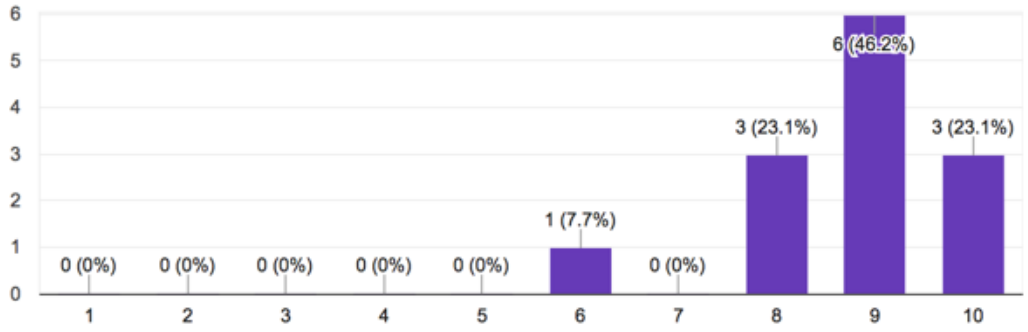


Figure 23 - Survey Result - Learnability of AAA.

As indicated in Figure 24, almost all users rate the **efficiency** of AAA as high. All 13 participants gave a rating of 6 and above (Highly Satisfied) for the efficiency of the product. Since AAA has very few steps and require very less user interactions to get the results, the amount of time it takes to perform a task is considerably less. This saves time for users and more amount of work getting completed. That is, more number of accessibility barriers are identified and fixed in a considerably short amount of time.

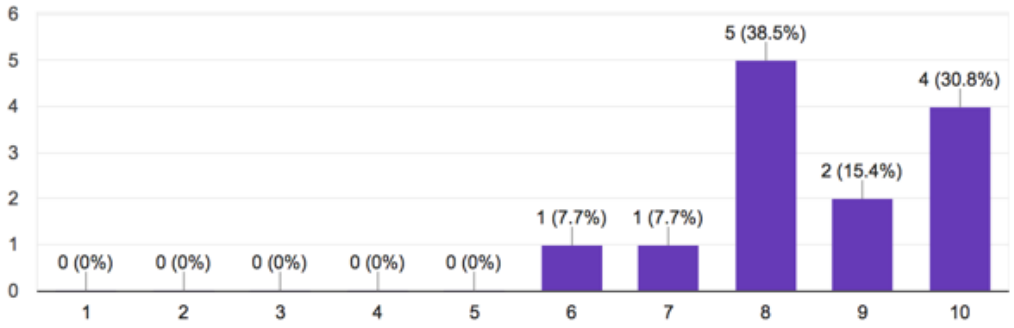


Figure 24 – Survey Result - Efficiency of AAA.

As indicated in Figure 25, all 13 participants also gave AAA a high rating in terms of **memorability**. As mentioned in the previous sections, 46.7% of the participants use this application “rarely”. Hence, it is important that when these users return to use the application, they find it familiar and do not have to look remember the steps required to complete a task. An important reason for AAA’s highly memorability rating is the intuitive user interface. All the elements on the user interface are laid out in a linear fashion to allow users to perform the steps to complete the tasks. They would not have to search for the controls or the process.

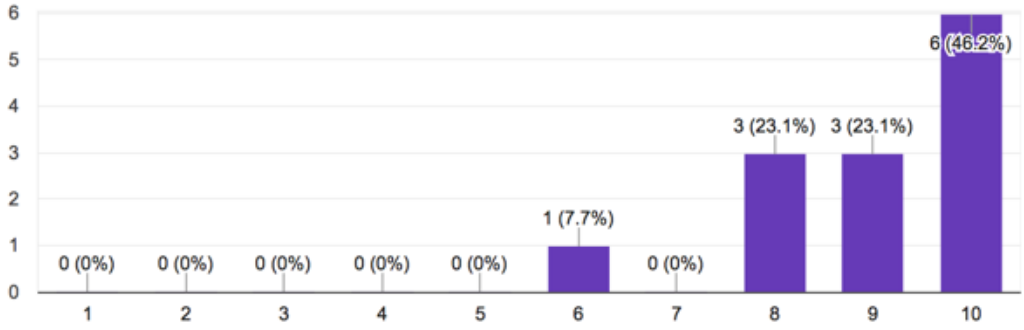


Figure 25 – Survey Result - Memorability of AAA.

5. DISCUSSION

From the results obtained from the surveys and interviews, we can conclude that AAA overcomes all the limitations that are present in existing solutions, such as ARIA Validator. A summary of all the results is provided below. The table compares AAA to Visual ARIA as most of our survey results were based upon these two applications.

Table 1 - Summary of Survey and Interview results

	ARIA Accessibility Analyzer	ARIA Validator
Recommendations to fix violation	**Yes**	No
Cause & Effect of Violation	**Yes**	No
Overall Satisfaction	**Highly Satisfied**	Not Satisfied
Is the application accessible to AT users?	**Yes**	No
Takes User's Accessibility Knowledge Into Consideration	**Yes**	No
Tests for latest ARIA 1.1	**Yes**	No
Processing Time	**Immediate**	Immediate
Number of False Positives	**Few to None**	High

6. FUTURE WORK

As part of the future work of this project, we would like to add more features to this application. Some examples are testing accessibility on complex nodes and different instances in a webpage. The current application only tests accessibility on the page at that point of time and does not take into account dynamic state and property changes. These additional features would allow the application to test different instances. The results will be appended to the existing results instead of displaying new results.

Additionally, the current application prototype allows tests only on webpages. Another improvement that could be done is to make the application test mobile content. This includes responsive web content and native applications on mobile devices. This would enable the application to test for accessibility on mobile platforms and various Operating Systems such as iOS, Android and Windows.

7. CONCLUSION

The goal of our project was to analyze limitations in available applications that test for accessibility, build an application that improves on these limitations and also introduce additional features. The application developed, ARIA Accessibility Analyzer, aims to make accessibility testing easy and require less Level of Effort on the users' end. From the obtained results, we can say that the AAA prototype works conveniently for users to test accessibility on a webpage.

It is important that accessibility is taken into consideration from the design phase but there are many instances where accessibility barriers are uncovered only during the development phase. This application, AAA, will allow web content developers to keep accessibility in mind during the development phase and make their website Accessibility Compliant.

8. REFERENCES

[1] Burgstahler et al., “Software Accessibility, Usability Testing And individuals With Disabilities”, Univ. Of WA, Seattle, WA, Rep. Volume X Num. 2, Dec. 2004

[2] D. P. Connaghan, “Accessibility in Rich Internet Applications”, Dublin Inst. Of Tech., Dublin, Ireland, Sep. 2008

[3] B. Garaventa. (2016, December 15). The ARIA Role Matrices [Online]. Available: http://whatsock.com/training/matrices/

[4] M. W. Brault. (2012, July). Americans With Disabilities: 2010 [Online]. Available: https://www.census.gov/content/dam/Census/library/publications/2012/demo/p70-131.pdf

[5] The Paciello Group. Why Accessibility [Online]. Available: https://www.paciellogroup.com/accessibility/

[6] H. Schneiderman (2017, April 25). Intro to Advanced ARIA (2nd ed.) [Online]. Available: https://www.deque.com/blog/advanced-aria/

[7] Apexx Group (2014). Web Accessibility Whitepaper [Online]. Available: http://apexxgroup.com/results/thought-leadership/

[8] R. Williams (2016). Click-Away Pound Survey [Online]. Available: http://www.clickawaypound.com/cap16finalreport.html

[9] S. Burgstahler, “Designing Software that is Accessible to Individuals with Disabilities”, DO-IT, Univ. Of WA, Seattle, WA, July 28, 2014.

[10] C. Mills et al. (2016, Dec. 05). WAI-ARIA Basics [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/Accessibility/WAI-ARIA_basics

[11] B. Garaventa (2016, December 15). AccDc Technical Style Guide (Version 3.3) [Online]. Available: http://whatsock.com/tsg/

[12] M. Cooper et al. (2016, Oct. 27). Accessible Rich Internet Applications (WAI-ARIA) 1.1

[Online]. Available: https://www.w3.org/TR/wai-aria-1.1/

[13] B. Garaventa. (2017, Jan. 06). Difference between ARIA 1.0 and 1.1: Deprecations & Additions [Online]. Available: http://www.ssbbartgroup.com/blog/differences-aria-1-0-1-1-deprecations-additions/

[14] B. Garaventa. (2017). The Visual ARIA Bookmarklet [Online]. Available: http://whatsock.com/training/matrices/visual-aria.htm

[15] R. Brown. (2014, August 06). ARIA Validator [Online]. Available: https://chrome.google.com/webstore/detail/aria-validator/oigghlanfjgnkcndchmnlnmaojahnjoc

[16] J. Scales. (2016, June 08). Tota11y – An Accessibility Visualization Toolkit [Online]. Available: http://engineering.khanacademy.org/posts/tota11y.htm

[17] J. Nielson. (2012, January). Usability 101: Introduction to Usability [Online]. Available: https://www.nngroup.com/articles/usability-101-introduction-to-usability/

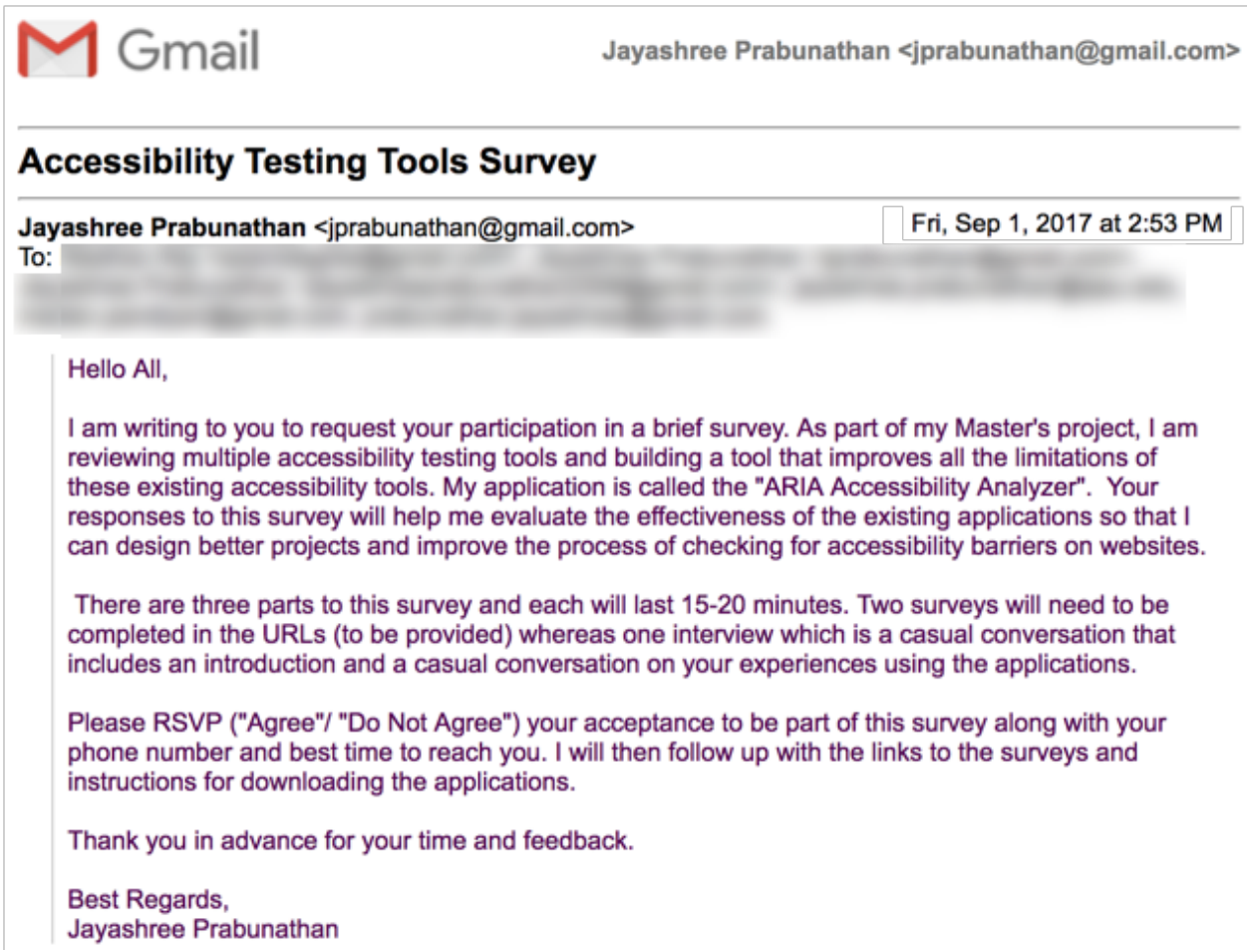
[18] T. Siobhan, "*DiGRA '09 - Proceedings of the 2009 DiGRA International Conference: Breaking New Ground: Innovation in Games, Play, Practice and Theory*". ISSN 2342-9666 Brunel University, Sep 2009.

[19] J. Avila (2017, Aug 08). *Accessibility Conformance Levels: Standards*, from https://www.levelaccess.com/accessibility-conformance-levels-standards/

9. APPENDIX

9.1. User Research

9.1.1. Email to invite participants



Gmail

Jayashree Prabunathan <jprabunathan@gmail.com>

Accessibility Testing Tools Survey

Jayashree Prabunathan <jprabunathan@gmail.com> Fri, Sep 1, 2017 at 2:53 PM
To: [redacted]

Hello All,

I am writing to you to request your participation in a brief survey. As part of my Master's project, I am reviewing multiple accessibility testing tools and building a tool that improves all the limitations of these existing accessibility tools. My application is called the "ARIA Accessibility Analyzer". Your responses to this survey will help me evaluate the effectiveness of the existing applications so that I can design better projects and improve the process of checking for accessibility barriers on websites.

There are three parts to this survey and each will last 15-20 minutes. Two surveys will need to be completed in the URLs (to be provided) whereas one interview which is a casual conversation that includes an introduction and a casual conversation on your experiences using the applications.

Please RSVP ("Agree"/ "Do Not Agree") your acceptance to be part of this survey along with your phone number and best time to reach you. I will then follow up with the links to the surveys and instructions for downloading the applications.

Thank you in advance for your time and feedback.

Best Regards,
Jayashree Prabunathan

Figure 26 - Email to invite participants.

9.1.2. Interviews

Note to interviewer: Keep in mind the level of accessibility knowledge the user has. Adapt your questions according to this to better extract information from users. Also ask AT users the accessibility of the applications they are surveying. This is the survey performed after the users

have had the opportunity to use one or more of the existing applications.

Introduction: Hello. Thank you for agreeing to participate. Your responses will be very valuable for my project. I thought I'd start by giving an introduction to my project to provide some context. After that, we can spend some time talking about your perspective of an accessibility testing application. I'm going to take notes here.

I am doing my Master's project that aims to analyze applications used to test for accessibility barriers on web applications. Based on your feedback and my own analysis, I will build an application called "ARIA Accessibility Analyzer (AAA)" that improves on all the limitations present in applications currently available. Additional features will also be included. The aim of this application is to make it easier for users to test for accessibility and fix them using less research and time on the user's part.

Question Guide *(remember to go with the flow and ask follow up questions):*

- Do you develop HTML Websites?
- Has accessibility been a part of building your websites?
- What is your expectation of an accessibility testing software?
- Have you previously used accessibility testing software?
- If yes, what is your experience using them?
- How familiar are you with ARIA 1.0 and ARIA 1.1?

Conclusion: These were all the questions I had. Is there any other information you think I should consider as I develop this application? Are there any questions you have for me? Thank

so much for taking the time to do this interview. You were super helpful and we appreciate it! Feel free to follow up with any questions or comments by email.