

A Study on the WAI-ARIA of Domestic Websites with High Session in Korea

Chorong Kim

Dept. of Multimedia Engineering,
Andong National University
Andong, Republic of Korea
e-mail: chfhd7379@naver.com

Eunju Park

Dept. of Multimedia Engineering,
Andong National University
Andong, Republic of Korea
e-mail: eunju@anu.ac.kr

Hankyu Lim

Dept. of Multimedia Engineering,
Andong National University
Andong, Republic of Korea
e-mail: hklim@anu.ac.kr

Abstract—Web accessibility depends on the development of universally accessible web content. The World Wide Web Consortium (W3C) has implemented many regulations for the improvement of web accessibility, and compliance with these regulations ensures that everyone will have equal access to web contents. HyperText Markup Language 5 (HTML5), which has been recently adopted as the web standard, also contains elements that support accessibility. Current web pages do not only depend on HTML and Cascading Style Sheet (CSS), but they also use various kinds of dynamic contents based on Rich Internet Applications (RIA). Therefore, HTML5 specifies the Web Accessibility Initiative-Accessible Rich Internet Application (WAI-ARIA) to improve accessibility of web applications, including RIA components. In this paper, the usability and accessibility of WAI-ARIA is evaluated targeting the top 50 websites accessed by most domestic users. According to the results, 78% of domestic websites have not applied the WAI-ARIA and only 6% have used it correctly.

Keywords-HTML5; Web; Web Accessibility; WAI-ARIA; Mobile Accessibility; Web Standard; User Interface

I. INTRODUCTION

Along with the rapid growth of the Internet, there have been a number of developments in the status of almost all fields, including politics, economics, society, culture, and administration. The quality of private lives can be enriched through the various digital cultures found on the Internet. However, informational discrimination can occur among individuals who are not familiar with or are unable to use the Internet [1][2]. Thus, it is important to create web content that guarantees accessibility for those with cognitive impairment or visual, hearing, or neuropathological disorders.

The Web Accessibility Initiative (WAI) of the World Wide Web Consortium (W3C) has established various standards and relevant guidelines [1] to improve web accessibility. Web standards are to be coded according to specifications and exclude private markup so that content can be operated on most browsers [3]. Therefore, coding regulations that comply with web standards allow everyone to equally and easily use information found on the Web without alienation due to various access issues [4].

HyperText Markup Language 5 (HTML5) was accepted as the final W3C standard on October 28, 2014 [5]. This new standard includes components supporting improvements to web content accessibility [6][7]. The Accessible Rich Internet Application (ARIA), which was newly added to

HTML5, defines accessibility enhancement methods for disabled individuals, when web contents and applications are produced using asynchronous JavaScript and XML (Ajax) and JavaScript. In HTML5, the Web Accessibility Initiative-Accessible Rich Internet Application (WAI-ARIA) specification enhances the accessibility of web applications [8][9].

Most current web pages do not only provide simple contents created using HTML and Cascading Style Sheet (CSS), but also dynamic contents using new methods, such as JavaScript. Despite the increased use of dynamic contents, no official method has been implemented to evaluate their accessibility [10]. Since 2015, W3C has provided “ARIA Validator” to evaluate the accessibility of dynamic contents. In this paper, therefore, WAI-ARIA, which was specified to substantiate the web application accessibility of HTML5, is evaluated to determine if it has been properly applied. To this end, the top 50 websites accessed by most domestic users during the first half of 2016 were selected and tested for their accessibility according to WAI-ARIA. An automatic evaluation method, the ARIA Validator provided by the W3C, was applied. According to the evaluation result, 39 (78%) out of the 50 websites tested were not using WAI-ARIA. Only 3 websites (6%), received a “Pass” rating for the accessibility test, and 8 (16%) websites received a “Fail.” Although many recent websites use dynamic contents, few of them use WAI-ARIA correctly. Thus, Web users with disabilities and/or disorders have difficulty approaching dynamic contents in many of the web pages. As Internet usage increases and more information is gathered, it should be equally available to all. Therefore, in order to make improvements, web developers and all relevant personnel should make an effort to modify their understanding and improve web accessibility.

This paper's construction goes like this. In chapter two, we introduced studies about web accessibility, accessibility supporting elements of the HTML5, entrance background and use of WAI-ARIA, ARIA validator, etc. In chapter three, we did ARIA validator evaluation of domestic websites from the top 50 high session websites in Korea and analyzed the result. Finally in chapter four, we described our conclusion and assignment from now on.

II. RELATED RESEARCH

Web emphasizes universality, and use of the web directly affects the quality of life in contemporary society. As web

content and websites have become more application-oriented, rich internet application technology has appeared, improving user experience (UX) on websites. However, it has also become a factor making the maintenance of accessibility to websites difficult and making usage of the web by the disabled more difficult. Web Accessibility Initiative-Accessible Rich Internet Applications, an accessibility guideline for RIA, provides effective accessibility to web content and web applications.

A. Web Accessibility

The power of the Web originates from its universality, and equal access for all people is its most important component [4]. In modern society, the Web is closely connected to human life and it expands to most areas; web accessibility is regarded as a necessary component for everyone, including those with various disabilities.

Web accessibility implies the development of web content that can be accessed by everyone regardless of their abilities or disabilities. Web accessibility means that everyone is guaranteed the right and opportunity to make use of services offered on websites, irrespective of physical and technical conditions and the user’s knowledge [11]. Therefore, web contents should be created so that everyone can recognize, operate, and understand them [12].

Web standards and regulations are defined by the WAI of the W3C, and are related to web technology. Websites that comply with web standards enhance accessibility.

B. Accessibility Supporting Elements of the HTML5

The web standard HTML4 was specified in December 1999. The new standard, HTML5, was specified in October 2014; it targets and follows web application development, breaking the boundaries of the pre-existing HTML concept. HTML5 contains components intended to improve accessibility [13].

The specifications for HTML5 are provided by WAI-ARIA, which has been defined by the WAI of the W3C to improve the accessibility of web content [7]. HTML5 has advantages that are compatible with all browsers and platforms and can be applied to various devices [6]. The accessibility support elements of HTML5 include Semantic Structure, canvas, audio, video, and WAI-ARIA. The semantic components <head>, <footer>, and <section> clarify the structural meaning of the document. The input form can be validated by the browser.

C. Entrance Background and Use of WAI-ARIA

Web content storage methods have evolved from coding skills based on HTML to RIA methods. Most web accessibility standards have evolved to include RIA content, such as interactive web contents containing JavaScript code, Flash, and Flex [10]. Contents that are produced by the RIA method provide dynamic and splendid User Experience (UX). However, there is a problem in that disabled individuals who depend on the use of assistive devices such as screen readers cannot use web applications created using RIA techniques [14]. In the case of web applications created using JavaScript or Ajax, such assistive techniques cannot accurately

understand the meaning of a component that was manufactured in factors that do not have a certain meaning, such as <div> or . Hence, W3C announced WAI-ARIA, aiming to improve universal accessibility to web contents and the Web.

WAI-ARIA is extremely helpful in developing applications using JavaScript and Ajax. Its role and areas of application are already supported by many screen readers, and it can be utilized to improve accessibility on subsidiary devices. With WAI-ARIA, one can add role, property, and state to the web application [7]. Table 1 shows the properties and examples of WAI-ARIA [9]. “Role” defines the function of a certain factor. It can provide a clear definition of its function—whether the area is a navigation area, a button, or a title. “Property” indicates the property or situation of each factor. For example, it lets users understand whether an input box of a form is read-only, required, or auto-complete. “State” shows the current status of a factor and it has values according to change. For example, it shows whether a menu is expanded, whether an invalid value was input, or whether contents are hidden. The use of these functions can improve the accessibility and usability of web applications.

TABLE I . WAI-ARIA ATTRIBUTES

Attribute	Example	Explanation
role		Screen reader reads a factor as a button instead of link.
property	<input type= “password” id= “user_pw” aria-required= “true”>	Thanks to the property of aria-required= “true,” users know that the corresponding item is essential in the screen reader.
state	<div role= “item” aria-expanded= “false”>	Statement of aria-expanded=“false” lets users know that it is currently folded.

D. ARIA Validator

The ARIA Validator is a program that was produced by Rick Brown in April 2015; it inspects WAI-ARIA implementation issues. It can evaluate code in the form of a web browser extension and be extended on the Chrome browser. The ARIA Validator is on the W3C web page introducing various evaluation tools related to web accessibility [15].

Figures 1 and 2 show the results of inspections of random websites using the ARIA Validator, indicating the URL of the evaluated webpage, evaluation time, and results of the “Roles Validated.” Figure 1 shows the results without any ARIA roles.

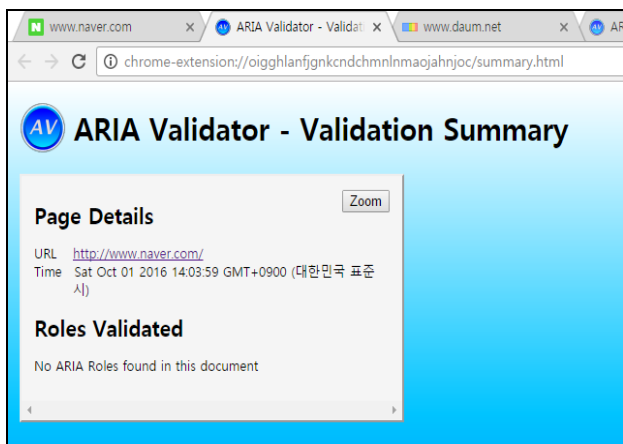


Figure 1. ARIA Validator Testing Result Screen (no ARIA role)

Figure 2 shows the test result for an ARIA that was incorrectly applied to a web page. Such pages are given a “Fail” rating and a link is given to sample pages that explain the correct usage. The page in Figure 2 indicates both correctly and incorrectly used ARIAs. While “button,” “region,” “search,” “combobox,” and “alertdialog” correctly applied ARIA, the application is incorrect in the case of several “comboboxes.” Thus, the rating achieved is “Fail.” When users select the spread menu for the “Fail” factor, they are able to see an explanation.

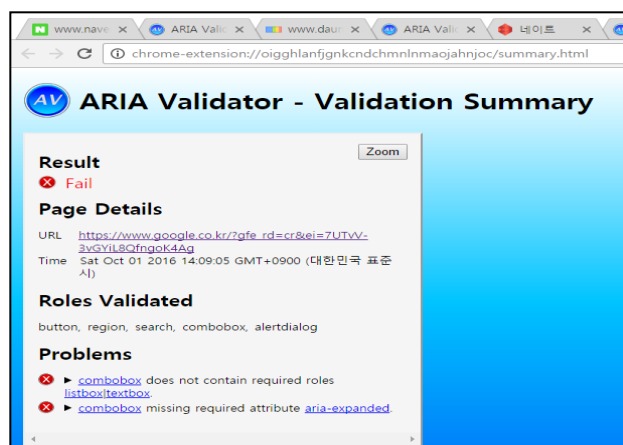


Figure 2. ARIA Validator Testing Result Screen (Fail)

III. ARIA VALIDATOR EVALUATION OF DOMESTIC WEBSITES FROM THE TOP 50 HIGH SESSION WEBSITES IN KOREA

In this paper, applying the ARIA Validator for targeting, the accessibility of WAI-ARIA was evaluated on the top 50 websites most accessed by domestic users. Target websites were selected from the rankings of March 2016 at “ranky.com,” which evaluates and explains web and mobile sites [16]. The ARIA Validator evaluation was executed on October 1, 2016.

The graph in Figure 3 illustrates the number of websites that did and did not use HTML5 in their construction in March and October of 2016.

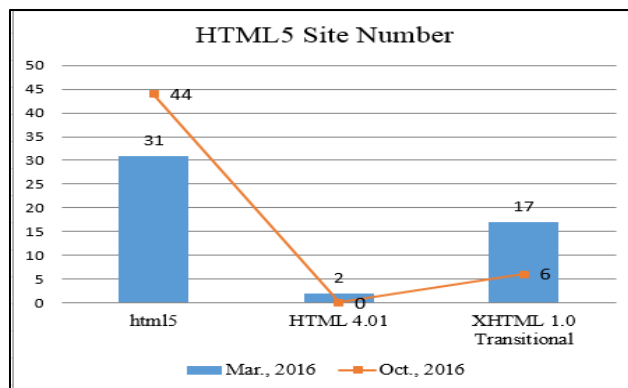


Figure 3. HTML5 Building Site Numbers

The HTML5 standardization process stresses accessibility and includes ways to enhance it. Therefore, accessibility factors that were not previously considered in earlier versions can be used in websites that are built using HTML5. As such, HTML5 helps to establish websites with improved accessibility. The results show that in October 2016, 44 out of 50 websites had been constructed using HTML5 and only 6 websites had not been based on HTML5. This is a higher ratio than that obtained for March 2016 when 31 out of 50 websites had been based on HTML5.

Figure 4 is a graph describing the research results for WAI-ARIA accessibility using the ARIA Validation targeting the top 50 websites accessed by most domestic users in October 2016. Among 50 websites, 39 (78%) were marked as “No ARIA roles found in this document,” which means the developers did not use WAI-ARIA. WAI-ARIA is supported in HTML5, so that 6 websites that did not use HTML5 also did not use WAI-ARIA. Eleven websites applied WAI-ARIA, but 8 of them rated a “Fail.” Only 3 websites—“nate,” “kakao,” and “epost” rated a “Pass.” According to the evaluation result, the majority of domestic websites had not used WAI-ARIA and most of the websites using WAI-ARIA applied it incorrectly. This implies that although new HTML5 standards are being used, the perception of accessibility is low. Currently, many dynamic contents are in use and improvements are necessary to allow users to access certain web pages; developers should make efforts to change their mindsets.

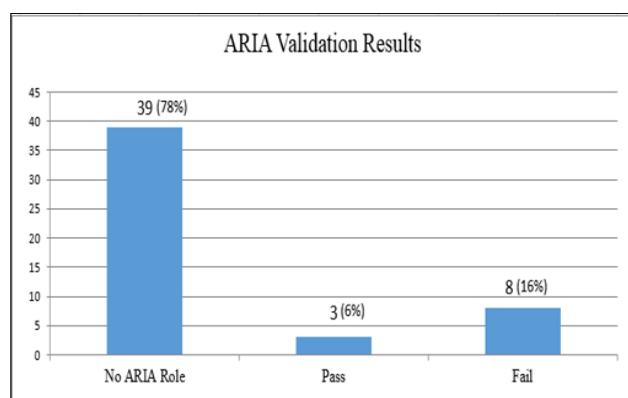


Figure 4. ARIA Validation Results

TABLE II . ANALYSIS OF PASS SITES

Site	Pass Element	Used Tag
nate	banner, search, navigation	<div id= "NateBi" class= "area_bi" role= "banner"> <div class= "area_search" role= "search"> <div id= "divGnb" class="area_gnb" role="navigation">
kakao	dialog	<div aria-hidden= "true" aria-labelledby= "urgent_notice_modal_label" class= "modal fade" id= "urgent_notice_modal" role= "dialog" tabindex= "-1">
epost	main	<div class= "slider" role="main">

Table 2 shows the websites that achieved a “Pass” with the ARIA Validation. Among the 50 websites, only 3 (6%) rated a “Pass”—“nate,” “kakao,” and “epost.” “Nate” described the “banner,” “search,” and “navigation” components using the WAI-ARIA role. Also implemented with the WAI-ARIA role were “dialog” of “kakao” and “main” of “epost.” In the cases of “nate” and “epost,” however, pages other than the main page had not applied WAI-ARIA. Moreover, although the use ARIA is required for many website factors, ARIA is clearly stated in the case of only a few of the factors. The “nate” website has a clear

ARIA statement in only 3 out of 227 <div> factors. Analysis of the source in the actual website indicated the necessity for correcting ARIA use.

Table 3 shows “Example of Fail Message” and “Roles Validated” for websites that received “Fail” in the ARIA Validation test results. “Roles Validated” shows the factors that correctly used ARIA.

“Example of Fail Message” shows the factors that incorrectly used ARIA. “Google” correctly used ARIA for a number of factors; however, it did not use ARIA in the case of several factor attributes that are related to ARIA among

TABLE III. ANALYSIS OF FAIL SITES

	Example of Fail Message	Roles Validated
Google	- combobox does not contain required roles listbox textbox - combobox missing required attribute aria-expanded.	button, region, search, combobox, alert, dialog
Facebook	- aria-required is not allowed when “an exactly equivalent native attribute is available”	presentation, main, alert, button, contentinfo
Youtube	- heading unsupported attribute aria-selected - link unsupported attribute aria-selected - menuitem not in required scope menu menubar	alert, link, navigation, menu, menuitem, complementary, main. Button, dialog
Bing	- listbox does not contain required roles option - aria-expanded is not supported on this element - aria-owns IDREF off_menu_cont must not be “aria-owned” by more than one element (repeated 2 more times)	search, combobox, listbox, navigation, group, complementary, button, contentinfo
Twitter	- textbox unsupported attribute aria-expanded - textbox unsupported attribute aria-expanded	search, listbox, presentation, option, group, button, textbox
Msn	- menu does not contain required rolesgroup menuitemradio menuitem menuitemcheckbox menuitemradio - aria-expanded is not supported on this element	banner, search, menu, main, menubar, menuitem, button, complementary, contentinfo
Yonhapnews	- button unsupported attribute aria-selected. - button unsupported attribute aria-selected.	button, slider
Microsoft	- menu does not contain required rolesgroup menuitemradio menuitem menuitemcheckbox menuitemradio - menu does not contain required rolesgroup menuitemradio menuitem menuitemcheckbox menuitemradio	banner, navigation, menubar, button, menu, search, main, region, radiogroup, radio, contentinfo

the HTML sources of “fail” “combobox,” including `<input class= “gsfi lst-d-f” id= “lst-ib” name= “q” autocomplete= “off” aria-label= “search” aria-haspopup= “false” role= “combobox” aria-autocomplete= “both”>`. For more accurate information, the ARIA Validator requires a mark on whether it is a listbox or textbox and a clear statement on the spread menu (aria-expand). It is possible to accurately define the role and the status of the combobox of `<input role= “listbox” aria-expand= “true”>`.

IV. CONCLUSION

Tim Berners-Lee, the inventor of the World Wide Web, said that “the power of the Web lies in generality and providing Web accessibility to everyone regardless of disabilities” [16]. The Web is being increasingly used by many, including public organizations, as a means of receiving and disseminating information. Therefore, individuals with disabilities should be guaranteed the same level of Web accessibility as able-bodied people.

Due to the increase in dynamic contents, the factors that support their accessibility have been in classified in HTML5. However, it was proven that not many websites build their web pages supporting accessibility according to classifications. Although HTML5 was widely used in a number of websites before its designation as a standard, there were only a few websites in our study that used WAI-ARIA. Constructing websites that comply with accessibility regulations can be burdensome to web developers. However, considering that accessibility defines the basic spirit of the Web, the development of websites that comply with accessibility standards is deserving of developers’ efforts and interest. In future work, we will investigate the ARIA compliance rate for international web pages and continue to work on ways to increase the accessibility of dynamic contents. In order to know the relationship between the pages with RIA and web accessibility, we will research the web accessibility evaluation for both the pages using the WAI-ARIA and the pages not using the pages not using the WAI-ARIA.

ACKNOWLEDGMENT

This research was supported by a grant from 2017 Seoul Accord Project (2011-0-00559) of MISIP (Ministry of Science, ICT and Future Planning) and IITP (Institute for Information and Communication Technology Promotion).

REFERENCES

[1] SeongJe Park and SeokChan Jeong, “The Comparative Analysis of Web Accessibility Evaluation Tools by Evaluating Image Contents on the Web Pages,” *The e-Business Studies*, Vol. 11, No. 3, pp. 347-366, 2010.

[2] HeonSik Joo, “A Study on Web accessibility situation of Public Institution and Major IT Companies Institutions,” *Journal of the Korea Society of Computer and Information*, Vol. 14, No. 10, pp. 175-187, 2009.

[3] Jim. Thatcher and Michael. R. Burks, “Web accessibility & web standards and regulatory compliance,” Acorn Publishing Co, 2011.

[4] SeungHwan Gu, KyongSok An, KwangMo Lee, and SungJin Choi, “Implementation of a Research Task Management System for

Support Smart Work Considering Web Accessibility,” *Journal of the Korea Contents Association*, Vol. 13, No. 9, pp. 39-48, 2013.

[5] W3C Recommendation, <http://www.w3.org/TR/html5/> [retrieved: 10, 2016]

[6] HongGi Cha, WonSuk Lee, and SeungYun Lee, “W3C HTML5 Standardization Trend,” (*ICT Standard & Certification*) *TTA Journal*, Vol. 159, pp. 94-101, 2015.

[7] Shiraishi Shunpei, “Introduction to HTML5 & API,” Freelec in Republic of Korea, 2010.

[8] Mozilla Developer Network, <https://developer.mozilla.org/ko/docs/Web/Accessibility/ARIA> [retrieved: 10, 2016]

[9] Brian P. Hogan, “HTML5 & CSS3,” Insight in Republic of Korea, 2011.

[10] Hawoong Han, “An Assessment Method of Web Accessibility for RIA Contents,” Graduate School of Information Sciences, Soongsil University, 2011.

[11] KyoungSoon Hong, “A Study on Improvements of Automatic Web Accessibility Evaluation Tool Using Empirical Knowledge,” Doctorate thesis from Graduate School of Department of Information and Telecommunication Engineering, Incheon National University, 2015.

[12] Korea Web Accessibility Evaluation Center, <http://www.kwacc.or.kr/WebAccessibility/Definition> [retrieved: 10, 2016]

[13] W3c Web Accessibility Initiative, <https://www.w3.org/WAI/PF/html-task-force> [retrieved: 10, 2016]

[14] National Information Society Agency, “WAI-ARIA,” National Information Society Agency, 2016.

[15] W3C Web Accessibility initiative, <https://www.w3.org/WAI/ER/tools>, [retrieved: 10, 2016]

[16] Rankey.com, <http://www.rankey.com> [retrieved: 03, 2016]