Capability Survey of Japanese User Agents and Its Impact on Web Accessibility

Takayuki Watanabe
Dept. Communication,
Tokyo Woman's Christian University
2-6-1 Zenpukuzi, Suginami-ku,
Tokyo, 167-8585, JAPAN
+81-3-5382-6524
nabe@lab.twcu.ac.jp

Masahiro Umegaki
Dept. Systems Engineering,
Graduate School of Electro-Communications
1-5-1 Chofugaoka, Chofu-shi
Tokyo, 182-8585, JAPAN
umegaki@se.uec.ac.jp

ABSTRACT
Capabilities of major Japanese user agents, three screen readers and one voice browser, were investigated with the following test files: W3C UAAG 1.0 Test Suite for HTML 4.01, an accessible PDF file, an accessible Flash file, and test files which test Japanese specific issues. Using the UAAG 1.0 Test Suite, 20 out of 48 Priority 1 checkpoints were met by all user agents, while all of the user agents failed to meet 11 of the checkpoints. Test results of all test files were assigned into three categories: capabilities satisfied by almost all user agents, capabilities not satisfied by any of the user agents, and capabilities that were satisfied by some of the user agents only. The test results indicated that 1) two major Japanese user agents do not have enough functions to navigate through a Web page using the structure information of the content, and 2) none of the user agents have enough functions to control multimedia and time-dependent interactions. These results provide an objective evidence to define the Japanese baseline, a set of technologies that a user agent is assumed to support, which is required in the WCAG 2.0 working draft. Accessibility responsibility between Web content and user agents is also determined by the current survey.

Categories and Subject Descriptors
K.4.2 [**Social Issues**]: Assistive technologies for persons with disabilities, H.5.2 [**User Interfaces**]: Evaluation/methodology, Standardization, H.5.4 [**Hypertext/Hypermedia**]: User issues

General Terms
Measurement, Human Factors, Standardization, Verification

Keywords
Web, accessibility, user agent, W3C, UAAG, MSAA, DOM, Japan, visual disability.

W4A at WWW2006, 23rd-26th May 2006, Edinburgh, UK

1. INTRODUCTION
The importance of Web accessibility has been recognized worldwide. The W3C (World Wide Web Consortium) developed the first version of the Web Content accessibility Guidelines (WCAG 1.0) [3] in 1999. WCAG 1.0 was subsequently used to guide or regulate Web content accessibility in many countries. In the United States, Section 508 of the Rehabilitation Act [11], which includes Web accessibility requirements, came into effect in 2001.

In Japan, private companies developed their own guidelines based on WCAG 1.0. In the public sector, in 1999, "Guidelines for the Creation of Internet Web Content Accessible by People with Disabilities" were jointly announced by the Ministry of Posts and Telecommunications and the Ministry of Health and Welfare. In 2004, JIS X 8341-3 "Guideline for older persons and persons with disabilities -information and communications equipment, software and services- Part3: Web Content" [7], was published as a Japanese Industrial Standard. JIS X 8341-3 was developed paying attention to the harmonization with WCAG 1.0, and with the subsequent WCAG 2.0 working draft [17,18]. Section 67 of the Industrial Standardization Law of Japan states that "When the nation and local public bodies determine standards, they must pay attention to JIS." Thus the JIS X 8341 guidelines affect public sector activity in Japan.

In spite of the above guidelines, Web accessibility remains deficient, even in the public sector. In the United States, according to "The State of Federal Websites: The Pursuit of Excellence" [12] published by PricewaterhouseCoopers Endowment for The Business of Government, only 13.5% of federal web sites were fully accessible (i.e., the Bobby accessibility testing software reported no error for those sites.). Web accessibility has also been evaluated in Europe, as reported by the UK Cabinet Office in a document titled "eAccessibility of public sector services in the European Union" [13]. According to that document, detailed assessment of 436 government service websites across Europe showed that only 3% of them achieved Level A conformance with WCAG 1.0. In Japan, "Nikkei Pasocon (Personal Computer)" carried out a survey of government web sites and found that even the most accessible of them lacked fundamental considerations of accessibility [10].

Improvements of Web accessibility require web site design, authoring, and programming to conform to Web content accessibility guidelines such as WCAG 1.0, Section 508, or JIS X

8341-3. Web authors, however, sometimes cannot make accessible content if they do not know how user agents render their content. For example, even when "accesskey" attributes are used to make easy access to some important parts of the content, if a user agent does not support an access key function, a user cannot directly access that part with use of an accessskey. Another example is alternative information about an image: an image can have an "alt" attribute which is used to specify alternate text of the image, a "title" attribute used to offer advisory information about the image, or a "longdesc" attribute used to specify a link to a long description of the image. Even if an author uses a "longdesc" attribute to describe the image in detail, a user agent which does not support "longdesc" attributes cannot make use of that information. Another example is heading elements: even if an author uses heading elements such as "h1" and "h2", a voice browser which does not use different voice when reading heading elements cannot convey the structure information to the user. Therefore, improvement of Web accessibility requires knowledge of user agent capabilities.

Japanese users who are blind cannot use user agents (screen-readers or voice browsers) sold in the United States because these user agents do not speak Japanese. Japanese users need either user agents developed in Japan or non-Japanese user agents that are localized to Japan. However, the Japanese market is small relative to the global market for English language user agents. Thus developers of Japanese screen-readers and voice browsers face difficulty generating sufficient revenue to implement enough capabilities to their user agents. As a result Japanese language user agents suffer in comparison to their English language counterparts.

Harmonization of Web accessibility guidelines is of great interest today. For example, Judy Brewer, domain leader of W3C/WAI, argues for the importance of standards harmonization in her presentation [1]. In order to use one international guideline, we must know how Japanese user agents differ from English ones, since Web accessibility is accomplished both by accessible Web content and by accessible user agents [5].

2. OBJECTIVES OF THE CURRENT SURVEY

Based on the evidence and arguments presented in the preceding section, a capability survey of Japanese user agents is a necessary step towards improved Web accessibility in Japan. Carrying out this survey can also provide objective data concerning the issues discussed in the following subsections.

2.1 Conformance to UAAG 1.0

Capabilities of user agents used in the United States have been investigated with the UAAG 1.0 Test Suites. Testing Japanese user agents with the same test suites enables us to show 1) how well Japanese user agents conform to UAAG 1.0 and 2) the differences in capabilities between Japanese and English language user agents. This testing also shows us to examine the applicability of UAAG Test Suites to Japanese user agents.

2.2 Japanese Baseline

Some checkpoints of WCAG 1.0 include the phrase "Until user agents ..." "Checkpoints that contain the phrase 'until user agents ...' require content developers to provide additional support for accessibility until most user agents readily available to their audience include the necessary accessibility features."[3].

In order to make WCAG independent of the capabilities of user agents and Web technology, a new concept of "baseline" was incorporated into the WCAG 2.0 working draft [2]. "About Baselines and WCAG 2.0" [8] explains the baselines as follows:

> A "baseline", as used in WCAG 2.0, is the set of technologies that an author assumes are supported and turned on in accessible user agents. Authors must ensure that all information and functionality of the Web content conform to WCAG 2.0 even when a user agent supports and uses only the technologies in the baseline.

This document [8] says

> A baseline may be set by a government body, client, organization, author, or combination of these.

In order to apply WCAG 2.0 in Japan, the Japanese baseline must be specified by some authoritative body with objective evidence. In order to specify a baseline in Japan, we need to examine the capability of Japanese user agents, i.e. what technologies are supported by user agents, in detail. If every Japanese user agents can handle JavaScript technology so as to make it accessible to users, then authors of Web content do not have to write alternative content and function given by JavaScript. On the other hand, if some user agents cannot make JavaScript accessible, authors are required to prepare alternative content.

PDF and Flash are widely used in the current Web. To set a Japanese baseline, we must know how well current Japanese user agents adapt these new technologies to the requirements of users with disabilities.

2.3 Accessibility Responsibility Between Content and User Agents

Content and user agents are dependent on each other. Poor user agents require content authors to provide a "repair" content to fix a problem in the content side, while poor content hinders Web browsing even with excellent user agents. This joint dependency between user agents and content hinders the progress of Web accessibility because, when problems arise, authors can blame poor user agents and developers of user agents can blame poor content. Thus methods are needed to distinguish between problems caused by content and by user agents respectively.

3. RELATED WORK

W3C/WAI UAWG (User Agent Accessibility Guidelines Working Group) has developed "a second generation test suite of UAAG 1.0 and implementation reporting system" [4]. "UAAG Implementation report for HTML 4.01" [15] shows test results of various English language user agents such as Internet Explorer, Firefox, Opera, Safari, and Home Page Reader. "Evaluations" [16] shows test results of various English language user agents and English language assistive technologies such as Safari, Window-Eyes with Internet Explorer, Mozilla, Opera, Internet Explorer, JAWS, Home Page Reader, and Firefox.

4. RESARCH METHODS

Capabilities of user agents were examined using test files. Every test was carried out on Windows XP SP2 with all Windows-

update patches implemented. The following subsection shows three kinds of test files used in the current survey.

4.1 UAAG 1.0 HTML 4.01 Test Suite

As shown in the previous section, UAWG had developed test suites which check if the user agent conforms to UAAG 1.0 [6]. Draft "UAAG 1.0 Test Suite for HTML 4.01" [14] consists of about 400 test files, which are classified into checkpoints of UAAG 1.0.

As illustrated in the sample screenshot shown in Figure 1, each test file consists of a Requirement Reference, Procedure, Test, Source Code, and Results. Testing is done with the Test section of the test file and results are examined in the Result section.

A sample of user agents was tested to see if they meet the UAAG 1.0 checkpoints.

The results in each test file were evaluated according to the following categories given in the Test Suite:

- C (Complete): The user agent passes all tests.
- AC (Almost Complete): The user agent passes all tests but one.
- PI (Partial Implemented): The user agent passes some tests.
- NI (Not Implemented): The user agent passes no tests.
- NR (Not Rated): The reviewer did not or could not run this test.
- NA (Not Applicable): The test does not apply to this user agent.

Secondly, results of every test files were summarized as a result of each checkpoint. When there was no test file for a checkpoint, that checkpoint was evaluated by a human judge based on the textual content of the checkpoint.

4.2 PDF and Flash

The PDF file[1] made by Adobe Systems Inc. as a reference accessibility test was used to our test. In using this test file, the following points were examined.

1) Every piece of content is read.
2) Heading elements are read.
3) Caption and alternative information for graphs is read.
4) Tables are read and navigation is enabled inside a table.
5) Alternative information for images is read.

The Flash file[2] made by Fujitsu Inc. to show the accessibility features of Flash was also used. With use of this test file, the following points were examined.

1) Every piece of content that was made accessible is read.
2) Objects which were made hidden from a display but were set to be read are read.
3) Tab key navigation is provided between buttons.
4) Reverse Tab key navigation is provided between buttons.
5) All button labels are read.
6) Objects which should be skipped are skipped.
7) Links are activated.

[1] http://www.comm.twcu.ac.jp/~nabe/data/UAResearch2005/TestData/wordsample.pdf

[2] http://jp.fujitsu.com/museum/pdp/plasma1.html

4.3 Japanese Specific Issues

WCAG 1.0 and UAAG 1.0 do not pay attention to language specific issues. JIS X 8341-3 includes some requirements which are important in Japanese and languages which use Han characters (Kanji ideographs). Test files were created[3] to check the following issues:

1) How Japanese symbols are read.
2) How language attribute is used to read content.
3) How ruby elements are read.
4) How words that contain whitespace characters are read. (e.g. W o r d)
5) How ambiguous date, time, and money representations are read. (e.g. 2006/03/22 09:50)
6) If list number of "ol" element is read.

Some of these tests are not Japanese specific issue but we created these tests because the UAAG Test Suites do not test these issues.

All tests were carried out in the default reading-mode, reading line by line, of the user agents.

5. SUBJECT USER AGENTS

The current survey investigates Japanese user agents for person with visual disabilities. The current survey treats a user agent as a combination of Web browser and assistive technologies. Three user agents most popular in Japan, and one user agent (JAWS) known to have high capability, were selected for the survey.

1) IBM Home Page Reader 3.04 SP3 (Japanese edition)
2) PC-Talker XP Version 3.04
3) 95 Reader Version 6.0
4) JAWS for Windows Professional Version 6.2 (Japanese edition)

Home Page Reader is a voice browser, while other three user agents are screen readers which read Internet Explorer. PC-Talker and 95 Reader are domestic screen readers developed in Japan.

6. SUMMARY OF TEST RESULTS

This paper describes summary test results for each user agent.

Detailed test results for the UAAG 1.0 test files and for the PDF, Flash, and Japanese specific issues are shown on our Web site [19].

Test results for each UAAG 1.0 checkpoint are summarized in Table 1. We could not evaluate some checkpoints because evaluating these checkpoints required the knowledge of internal specifications of the software. Those checkpoints were rated as NR and were not included in the following summary and discussions.

The results described in the following subsections are assigned into three categories: capabilities satisfied by almost all user agents, capabilities not satisfied by any of the user agents, and capabilities that were satisfied by some of the user agents only.

[3] These test files are at: http://www.comm.twcu.ac.jp/~nabe/data/UAResearch2005/TestData/Japanese/

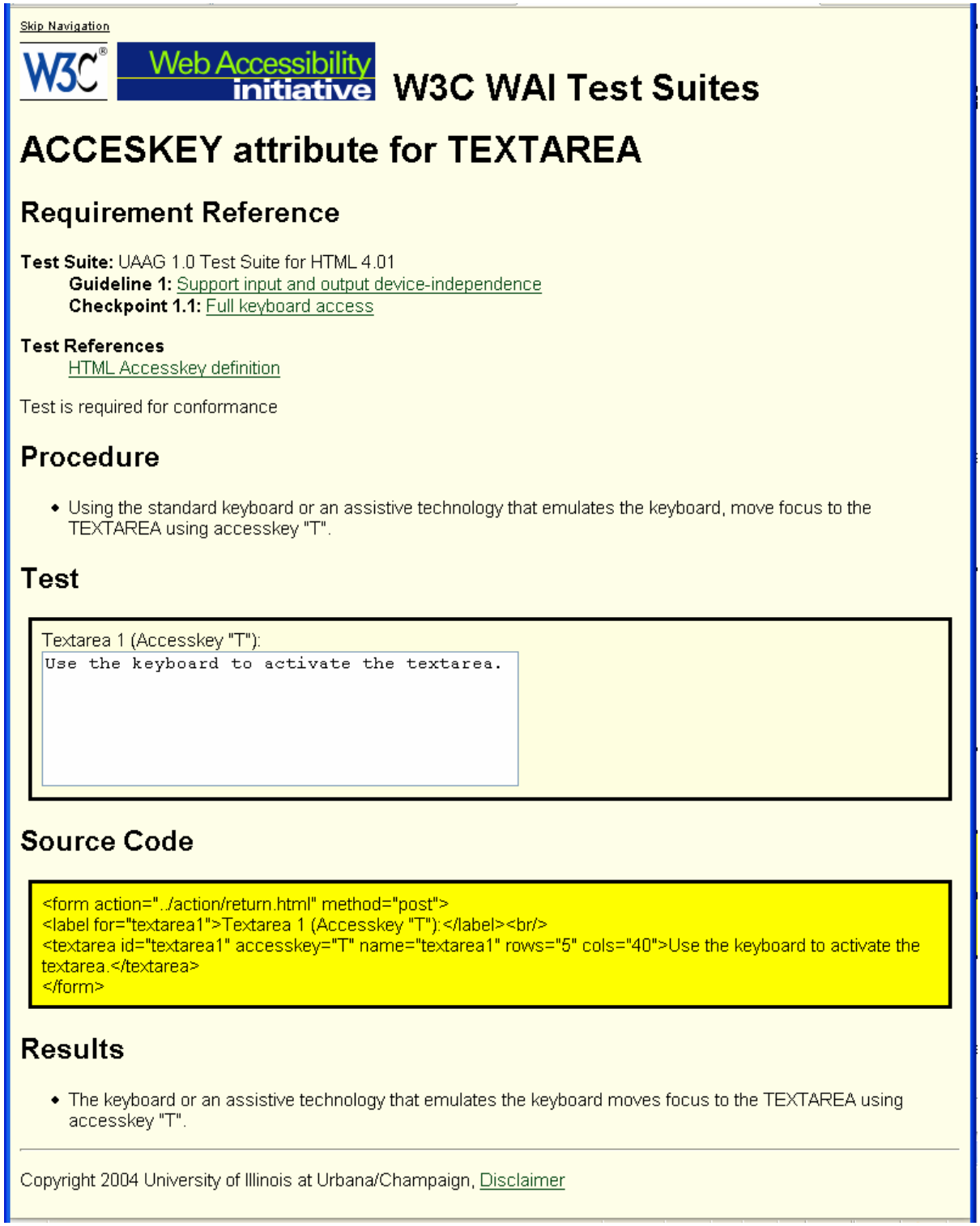

Figure 1. Example of test file (Test 97) in UAAG 1.0 Test Suite for HTML 4.01 [14] (http://cita.rehab.uiuc.edu/wai-eval/show-test/index.php?test_id=97)

Table 1. Results of UAAG 1.0 Test Suite for HTML 4.01

UAAG 1.0 Checkpoint	Priority	PC-Talker 3.04	95 Reader 6.0	JAWS 6.2	HPR 3.04
1.1: Full keyboard access	**1**	AC	AC	C	C
1.2: Activate event handlers	**1**	AC	PI	AC	AC
1.3: Provide text messages	**1**	C	C	C	C
2.1: Render content according to specification	**1**	C	C	C	C
2.2: Provide text view	**1**	C	C	C	C
2.3: Render conditional content	**1**	PI	PI	C	C
2.4: Allow time-independent interaction	**1**	NI	NI	PI	PI
2.5: Make captions, transcripts, audio descriptions available	**1**	NI	NI	NI	NI
2.6: Respect synchronization cues	**1**	PI	PI	PI	PI
2.7: Repair missing content	2	PI	PI	PI	AC
2.8: No repair text	3	C	C	C	C
2.9: Render conditional content automatically	3	PI	PI	C	C
2.10: Don't render text in unsupported writing systems	3	NI	NI	NI	NI
3.1: Toggle background images	**1**	C	C	C	C
3.2: Toggle audio, video, animated images	**1**	PI	PI	PI	PI
3.3: Toggle animated or blinking text	**1**	NI	NI	NI	NI
3.4: Toggle scripts	**1**	C	C	C	C
3.5: Toggle automatic content retrieval	**1**	NI	NI	NI	PI
3.6: Toggle images	2	AC	AC	AC	AC
4.1: Configure text scale	**1**	C	C	C	C
4.2: Configure font family	**1**	PI	PI	PI	PI
4.3: Configure text colors	**1**	AC	AC	AC	AC
4.4: Slow multimedia	**1**	NI	NI	NI	NI
4.5: Start, stop, pause, and navigate multimedia	**1**	NI	NI	NI	NI
4.6: Do not obscure captions	**1**	NI	NI	NI	NI
4.7: Global volume control	**1**	C	C	C	C
4.8: Independent volume control	**1**	NI	NI	NI	NI
4.9: Configure synthesized speech rate	**1**	C	C	C	C
4.10: Configure synthesized speech volume	**1**	C	C	C	C
4.11: Configure synthesized speech characteristics	**1**	C	C	C	C
4.12: Specific synthesized speech characteristics	2	PI	PI	PI	PI
4.13: Configure synthesized speech features	2	AC	AC	C	AC
4.14: Choose style sheets	**1**	AC	AC	AC	AC
5.1: No automatic content focus change	2	NI	NI	NI	PI
5.2: Keep viewport on top	2	NI	NI	NI	NI
5.3: Manual viewport open only	2	NI	NI	NI	PI
5.4: Section and focus in viewport	2	NI	NI	C	C
5.5: Confirm form submission	2	PI	PI	PI	PI
6.1: Programmatic access to HTML/XML infoset	**1**	NR	NR	NR	NR
6.2: DOM access to HTML/XML content	**1**	NR (NI/ PI)	NR (PI)	C	C
6.3: Programmatic access to non-HTML/XML content	**1**	NR	NR	NR	NR
6.4: Programmatic access to information about rendered content	**1**	NR	NR	NR	NR

6.5: Programmatic operation of user agent user interface	**1**	NR	NR	NR	NR
6.6: Programmatic notification of changes	**1**	NR	NR	NR	NR
6.7: Conventional keyboard APIs	**1**	NR	NR	NR	NR
6.8: API character encodings	**1**	NR	NR	NR	NR
6.9: DOM access to CSS style sheets	2	NR	NR	NR	NR
6.10: Timely exchanges through APIs	2	NR	NR	NR	NR
7.1: Respect Focus and selection conventions	**1**	C	C	C	C
7.2: Respect input configuration conventions	**1**	C	C	C	C
7.3: Respect operating environment conventions	2	C	C	C	C
7.4: Provide input configuration indications	2	C	C	C	C
8.1: Implement accessibility features	**1**	NR	NR	NR	NR
8.2: Conform to specifications	2	PI	PI	AC	AC
9.1: Provide content focus	**1**	C	C	C	C
9.2: Provide user interface focus	**1**	C	C	C	C
9.3: Move content focus	**1**	PI	PI	AC	C
9.4: Restore viewport state history	**1**	NI	NI	NI	NI
9.5: No event on focus change	2	NI	NI	NI	C
9.6: Show event handlers	2	NI	NI	NI	PI
9.7: Move content focus in reverse1	2	PI	PI	AC	C
9.8: Provide text search	2	NI	NI	C	C
9.9: Allow structured navigation	2	PI	PI	C	AC
9.10: Configure important elements	3	NI	NI	NI	NI
10.1: Associate table cells and headers	**1**	PI	PI	AC	AC
10.2: Highlight selection, content focus, enabled elements, visited links	**1**	PI	PI	PI	AC
10.3: Single highlight configuration	2	NI	NI	NI	NI
10.4: Provide outline view	2	NI	NI	C	PI
10.5: Provide link information	3	NI	NI	C	AC
10.6: Highlight current viewport	**1**	NI	NI	AC	AC
10.7: Indicate viewport position	3	NI	NI	C	C
11.1: Current user input configuration	**1**	NI	NI	C	C
11.2: Current author input configuration	2	NI	NI	NI	C
11.3: Allow override of bindings	2	NR	NR	C	NI
11.4: Single-key access	2	PI	PI	C	PI
11.5: Default input configuration	2	AC	AC	C	C
11.6: User profiles	2	C	C	C	C
11.7: Tool bar configuration	3	NI	NI	NR	C
12.1: Provide accessible documentation	**1**	AC	AC	C	AC
12.2: Provide documentation of accessibility features	**1**	C	C	C	C
12.3: Provide documentation of default bindings	**1**	C	C	C	C
12.4: Provide documentation of changes between versions	2	C	C	C	C
12.5: Provide dedicated accessibility section	2	C	C	C	C

6.1 Capabilities Satisfied by Almost All User Agents

6.1.1 UAAG 1.0 Test Suites

Almost all user agents cleared the UAAG 1.0 Test Suite for HTML 4.01 at the following functions of the test file:

- Accesskey attribute. However, access keys where the key combinations were identical to the short cut key of the operating system or Internet Explorer could not be used.
- Button function in the form element. However, 95 Reader did not handle input button of the submit type.
- Tabindex. Except that 95 Reader did not use "tabindex".
- Activation of event handlers such as "onchange", "onfocus", and "onblur". Except that 95 Reader sometimes did not activate "onchange" event handlers.
- Alt attributes of "img" and input elements.
- "Title" attribute of "img" element. Except that PC-Talker and 95 Reader did not read "alt" attributes when both "alt" and "title" attributes were specified.
- "Caption" element of table. But "Caption" elements were read as normal text with no extra voice characters or voice attributes being specified.

The following functions were also satisfied by every user agents as a function of Internet Explorer:

- Display "noscript" element.
- Display alternate information for the "object" element.
- Activation of "link "element.
- Navigation of "link" elements, "textfield" elements and button controls within a form by the Tab key.
- Change of text size.
- Not display images and background images.
- Disabled scripts.
- Application of user style sheets.

6.1.2 PDF and Flash

Results for PDF were as follows:

- Sequential reading of headings and body from the top of the page to the end of the page. But Home Page Reader did not read headings.
- Every user agent sequentially read a table from the left-top cell to the right-bottom cell. But only JAWS could navigate within a table.

Results of Flash were as follows:

- Forward and backward navigation of button focus using the Tab key.
- Reading of button labels.
- Focus skip

Home Page Reader did not read Flash well in comparison to the other user agents. User agents other than Home Page Reader also had the following functions:

- Reading of displayed information.
- Reading of objects which were hidden from a display but were set to be read.

6.1.3 Japanese specific issues

No specific results were obtained for Japanese specific issues.

6.2 Capabilities Satisfied by None of the User Agents

6.2.1 UAAG 1.0 Test Suites

None of the user agents satisfied the following tests.

- Activation of "ondblclick" event when both "onclick" and "ondblclick" were specified.
- Toggle (stop) animated image.
- Control of multimedia objects embedded in "object" (or "embd") elements.
- Toggle (stop) animated or blinking test.
- Toggle (stop) redirect and refresh. (JAWS could stop refresh in virtual view when configured.)
- No automatic focus change to a new window.
- Navigation among "th" elements.
- Navigation among "thead", "tbody", and "tfooter". Every user agents read "tfooter" before "tbody" because "tfooter" was written before "tbody" in the HTML.
- Only JAWS navigated among "list" elements.
- Only JAWS read tables using the "axis" attribute.
- Only JAWS read tables using "col" or "colgroup" attributers.
- Only JAWS had functions to list every heading elements ("h1", "h2", …) and "caption" elements of the table in a separate window.
- Only Home Page Reader had functions to list all access keys used in the page.

The above lists do not include tests where results were ambiguous because of incomplete test files or ambiguous test procedures.

6.2.2 PDF and Flash

As for PDF, no user agents read headings differently from other text so as to make users distinguish which were headings and which was not. No user agents had navigation function among heading elements.

No Problems were found in Flash.

6.2.3 Japanese specific issues

Results of Japanese specific issues were as follows:

- Reading (pronunciation) of Japanese symbols were different among user agents. For example, Japanese (Zankaku) symbol "dash" was read as "bar", "dash", or "hyphen".
- "Lang" attribute did not change language of text-to-speech voice.
- No user agents could read ambiguous date, time, and money representations appropriately.

6.3 Capabilities Differing Among User Agents

6.3.1 UAAG 1.0 Test Suites

Test Results for the UAAG 1.0 Test Suites showed that JAWS and Home Page Reader had the following functions which were not supported by the other two Japanese user agents (PC-Talker and 95 Reader).

- Navigation of heading elements inside the page.
- Reading table with use of structure markups such as "summary" elements, "th" elements, and "scope" attribute.
- Link to anchors in the same page.

- Text search in a page.
- Reading "title" attributes of "abbr" and "acronyms" elements.
- Selection of arbitrary parts of text in a page using a keyboard.
- Customization of reading functions.

6.3.2 PDF and Flash

For PDF,

- All user agents except 95 Reader inserted a line break between displayed lines, which resulted in unnatural reading of Japanese text. (e.g. one line was read as "All u//ser a//gents// exce//pt 95 ..")
- Only JAWS had dedicated functions for reading a table.
- PC-Talker did not read alternative information of a graph. It only read the caption of a graph.
- Home Page Reader did not read headings.
- 95 Reader and JAWS read alternative information of images. PC-Talker did not read this information. Home Page Reader read that information in some images.

With Flash, only Home Page Reader did not read well.

6.3.3 Japanese specific issues

Results of Japanese specific issues were as follows:

- JAWS and Home Page Reader read list numbers of ordered list elements.
- 95 Reader did not read Japanese (Zenkaku) symbols even if it was configured to read them.
- Home Page Reader read character entity references of © (copyright) and ® (registered trademark) as "C Tyosakuken" ("Tyosakuken" means copyright) and "R Toroku Syohyo" (Toroku Syohyo means registered trademark).
- PC-Talker and JAWS read characters inside the "rp" elements of ruby.
- JAWS ignored whitespace characters inserted inside a word.
- JAWS read the word as one word even if a part of that word was modified by a "span" element.

7. DISCUSSION

The current survey examined capabilities of Japanese user agents in detail and found the following facts.

7.1 Conformance to UAAG 1.0

Each checkpoint of UAAG 1.0 is assigned to one of three priorities. Priority 1 is the basic requirement and is described as "If the user agent does not satisfy this checkpoint, one or more groups of users with disabilities will find it impossible to access the Web. Satisfying this checkpoint is a basic requirement for enabling some people to access the Web."[6]

As shown in Table 1, among 48 Priority 1 checkpoints, the following 20 checkpoints (CP) were met[4] by all user agents:

- CP 1.1: Full keyboard access
- CP 1.3: Provide text messages
- CP 2.1: Render content according to specification
- CP 2.2: Provide text view
- CP 3.1: Toggle background images
- CP 3.4: Toggle scripts
- CP 4.1: Configure text scale
- CP 4.3: Configure text colors
- CP 4.7: Global volume control
- CP 4.9: Configure synthesized speech rate
- CP 4.10: Configure synthesized speech volume
- CP 4.11: Configure synthesized speech characteristics
- CP 4.14: Choose style sheets
- CP 7.1: Respect focus and selection conventions
- CP 7.2: Respect input configuration conventions
- CP 9.1: Provide content focus
- CP 9.2: Provide user interface focus
- CP 12.1: Provide accessible documentation
- CP 12.2: Provide documentation of accessibility features
- CP 12.3: Provide documentation of default bindings

Thus, all of the user agents had basic capabilities to ensure Web accessibility.

None of the user agents met[5] the following 11 Priority 1 checkpoints:

- CP 2.4: Allow time-independent interaction
- CP 2.5: Make captions, transcripts, audio descriptions available
- CP 2.6: Respect synchronization cues
- CP 3.2: Toggle audio, video, animated images
- CP 3.3: Toggle animated or blinking text
- CP 3.5: Toggle automatic content retrieval
- CP 4.2: Configure font family
- CP 4.4: Slow multimedia
- CP 4.5: Start, stop, pause, and navigate multimedia
- CP 4.6: Do not obscure captions
- CP 4.8: Independent volume control
- CP 9.4: Restore viewport state history

The above list shows that current user agents lack functions relating to the control of multimedia and time-dependent interactions.

Of the preceding checkpoints, we recommend that the following functions should be implemented in Internet Explorer rather than user agents:

- Toggle (stop) animated image.
- Control of multimedia objects embed in "object" (or "embd") elements.
- Toggle (stop) animated or blinking text.
- Toggle (stop) redirect and refresh.
- No automatic focus change to the new window.

These functions are basic functions of a web browser and thus related to not only accessibility but also usability.

7.2 Capability Difference

As shown in Section 6.3.1, there was a clear difference between the capability of JAWS and Home Page Reader and that of PC-Talker and 95 Reader.

The following Priority 1 checkpoints of UAAG 1.0 was met by JAWS and Home Page Reader but not met by PC-Talker and 95 Reader:

- CP 2.3: Render conditional content
- CP 6.2: DOM access to HTML/XML content

[4] Test result is C or AC.

[5] Test result is PI or NI.

- CP 9.3: Move content focus
- CP 10.1: Associate table cells and headers
- CP 10.6: Highlight current viewport
- CP 11.1: Current user input configuration

Clause 5.2 a) of JIS X 8341-3 states "Web content shall define document structure using heading, paragraph, list, and other elements." and clause 5.2 c) states "A table must have an intelligible and explicit caption, and a structure as simple as possible which is specified by appropriate markup." In other words JIS X 8341-3 requires content authors to use markups to specify the page structure. User agents of PC-Talker and 95 Reader could not use these markups because they did not have the following functions:

- Navigation of heading elements inside the page.
- Reading table with use of structure markups such as "summary" elements, "th" elements, and "scope" attribute.

These user agents cannot make use of structure even if the content conforms to JIS X 8341-3.

The current WCAG 2.0 working draft has Guideline 2.4 "Provide mechanisms to help users find content, orient themselves within it, and navigate through it."

User agents of PC-Talker and 95 Reader could not use navigation mechanism because they did not have the following functions:

- Link to anchors in the same page.
- Text search in a page.

These user agents could not use a table of content which consists of links to anchors. They also could not use navigation-skip links and they could not find text directly using a search function of a web browser.

PC-Talker and 95 Reader also did not have the functions listed below, which enhance the usability and accessibility of Web content.

- Reading list numbers of ordered list elements.
- Reading "title" attributes of "abbr" and "acronym" elements.
- Selection of arbitrary parts of text in a page using a keyboard.
- Customization of reading functions.

The above results show that PC-Talker and 95 Reader lack indispensable accessibility functions.

User agents can use MSAA (Microsoft Active Accessibility) and DOM (Document Object Model) to retrieve information from a Web browser. MSAA is a standard technology for assistive software in Windows OS. Assistive software such as screen readers can obtain information from a Windows application and the OS through MSAA. Assistive software can also obtain or operate Web content through DOM. DOM enables an application to retrieve structured Web content and operate Web content and style. Using DOM enables an application to navigate through heading elements or table cells or list elements. It also enables an application to read the searched text. MSAA does not have dedicated functions to retrieve and operate a complete tree structure of a Web page. The current MSAA does not provide information in "longdesc" attribute. UAAG 1.0 requires "DOM access to HTML/XML content" as Priority 1 in checkpoint 6.2. Major user agents in the United States such as Home Page Reader, JAWS, and Window-Eyes use DOM to read Web pages.

The above capability differences of Japanese user agents probably reflect how actively a user agent uses DOM as well as MSAA. Programming with MSAA is simpler than that with both MSAA and DOM. Application performance is better when only MSAA is used. The number of Japanese disabilities is small, with only 0.3 million who are legally blind, compared to those of English speaking people. This small market may make it difficult to develop high capability user agents which use DOM.

As written in "Designing with web standards"[20], the current Web standard consists of three separate components: structure, presentation, and behavior. (X)HTML markup language markups the *structure* of the content. *Presentation* style is specified with CSS. As for *behavior*, "A standard object model (the W3C DOM) works with CSS, XHTML, and ECMAScript…" [p.56 of Ref.20] In addition to that, as described in the previous subsection, Web content accessibility guidelines require appropriate markup of content. Thus, user agents must have the capability to use content's structure information.

7.3 Japanese Baseline

Results of the current survey can be used to determine the Japanese baseline, i.e., technologies authors of Japanese Web pages can use in their content without providing alternative format.

The current results show that PDF, which was made accessible, is not as accessible as (X)HTML. What can be done by all user agents was sequentially reading content, paragraphs and table cells, from the top of the page to the bottom of the page. Even JAWS, which reads PDF well, did not distinguish headings and did not navigate through headings. Therefore, information which can be presented in (X)HTML should use (X)HTML format rather than PDF format because (X)HTML is more accessible.

As for Flash, our results show that user agents except Home Page Reader read accessible Flash well.

These results show that both PDF and Flash can be accessed by Japanese user agents to some degree if they are made accessible.

As for (X)HTML, authors are encouraged to use structure markups of (X)HTML because there are user agents that can make use of these markups. It must be emphasized that in Japan there are user agents that cannot make use of structure markups of (X)HTML technology. Our survey shows that discussion of baseline depends greatly on capabilities of user agents.

7.4 Applicability of UAAG 1.0 to Japanese User Agents

There was no UAAG 1.0 test file that could not be applied to Japanese user agents. We found, however, that some test files did not provide test functions because of grammar errors, ambiguous procedures, etc[6]. Evaluation of checkpoints was difficult if no test file was prepared for that checkpoint. We also found that the checkpoints of UAAG 1.0 were difficult to understand because of its abstract representation. We sometimes could not definitively decide if a user agent met a checkpoint.

[6] These problems were reported to UAWG.