

# What's the Web Like If You Can't See It?

Chieko Asakawa  
Tokyo Research Laboratory, IBM Japan  
1623-14 Shimo-tsuruma  
Yamato-shi, Kanagawa-ken, Japan  
+81-46-215-4633  
chie@jp.ibm.com

## ABSTRACT

Awareness of Web accessibility is spreading all over the world among Web designers and developers, due to regulations such as the US law called Section 508 and guidelines like the W3C WCAG. We now see various Web accessibility adaptations on the Web. For example, we see increasing use of alternative texts for images and skip-navigation links for speed.

However, we sometimes find inappropriate ALT texts and broken skip-navigation links, even though they are present. These pages may be compliant, but they are not accessible or really usable. We analyzed such problems and found that some sites only try to comply with regulations and guidelines, but without understanding the needs underlying Web accessibility. We concluded that Web designers and developers should experience the real problems faced by people with disabilities so they can create truly accessible and usable pages. There was no practical way for them to experience disabilities.

In this paper, after giving an overview of the historical progress of voice browsers, we discuss how much and how well the Web accessibility has progressed by analyzing real world improvements to existing sites. We then describe why the "disability experience" helps give a better understanding of the Web accessibility guidelines and regulations. Some tools like Home Page Reader and aDesigner are available to let designers experience blind users' perspective of usability. Finally, we discuss the future direction of our accessibility efforts.

## Categories and Subject Descriptors

K.4.2 [Social Issues]: Assistive technologies for persons with disabilities, H.3.4 [Online Information Services]: Web-based services, H.5.2 [User Interfaces]: Evaluation/methodology, Graphical user interfaces (GUI), Screen design (e.g., text, graphics, color), Standardization, Style guides

## General Terms

Design, Reliability, Human Factors, Standardization, Verification..

## Keywords

Web accessibility, blind, voice browser, screen reader, non-visual

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

W4A at WWW2005, 10th May 2005, Chiba, Japan  
Copyright 2005 ACM 1-59593-036-1/05/05...\$5.00.

usability.

## 1. INTRODUCTION

The Web is becoming a valuable information resource for people with vision disabilities. Now they do not always need to ask for sighted people to look for the information they need. They can obtain any type of information by themselves from all over the world whenever they need it by accessing the Web. Self-talking browsers [1] and screen readers [2, 3] have been developed and improved along with the evolution of the Web to help blind people obtain information from the Web.

Since the late 1990s, however, the Web accessibility has been becoming a serious issue for people with disabilities. The Web is becoming much more visual and visually complex documents and interfaces such as JavaScript and Flash are integrated into Web pages. Some of these visual components are not accessible using self-talking browsers and screen readers. To solve such accessibility issues, various efforts are underway worldwide. The World Wide Web Consortium's Web Accessibility Initiative (WAI) published the Web Content Accessibility Guideline (WCAG) [4] in 1998. The U.S. government has Section 508 of the Rehabilitation Act [5] that requires all federal agencies' electronic and information technology to be accessible to people with disabilities.

The awareness of Web accessibility has been broadly spread all over the world. There is no doubt that regulations and guidelines related to accessibility have been helping make the Web accessible. Web designers and developers need to provide accessible content when they are asked to make it compliant with these guidelines and regulations. For example, Section 508 provides 16 check items. They need to check if their pages are compliant with these 16 check items in order to comply with Section 508. It would be a time-consuming task if they had to do this manually. Web accessibility evaluation tools [6, 7] are now available in the market to effectively check the compliance.

As a result, we now see various Web accessibility adaptations on the Web. For example, we see increasing use of alternative texts for images. All the guidelines and regulations state that every image needs to provide alternative text. The existing evaluation tools can easily check if an image has alternative text. If there is none, the tool automatically outputs an error message with the line number in the corresponding HTML file where the image should provide an alternative text.

We also see increasing use of skip-navigation links. Section 508 states that a method shall be provided to permit users to skip repetitive navigation links. Similar rules are also used in other guidelines. One of the known techniques to skip repetitive navigation links is the insertion of a skip-to-main content link at

**Table 1. My personal experiences in the context of the Web's growth**

	Personal (and IBM internal)	Other activities
1993 Mar		Lynx browser (text-based Web browser) version 2.0a was released [16]
1993 Apr		The Mosaic browser is released
Early 1994	My first Web access experience using a DOS screen reader (IBM Screen Braille) with LYNX through telnet access to a UNIX server.	
Late 1994	A Japanese text-based BBS provider (Asahi Net) started a service to convert Web pages into text through their BBS service (Jun). I started to access the Web by using this system with DOS screen readers.[9][18]	World Wide Web Consortium was created
1994 Dec		Netscape Navigator v1.0 was released.
Early 1996	Developed PAL (macro language for IBM ScreenReader/2) for Web Explorer (IBM's Web browser on OS/2) [9]	
Mid-1996	Developed PAL for Netscape Navigator v2.0 on OS/2 [10][19]	
Early 1997	Prototyped first standalone talking Web browser "Web Reader" [19][20]	
1997 May		W3C Web Accessibility Initiative (WAI) First Technical Meeting [17]
1997 July	First prototype of Home Page Reader was completed, and I succeeded in buying a cake from an online shopping site.	
1997 Oct	Home Page Reader Version 1.0 (Japanese version only) was released [12]	
1998	(heavy work of translating HPR into English)	
1999 Jan	Home Page Reader Version 2.0 (first English version) was released [13]	
1999 May	Prototyped the table navigation interface [14]	
1999 Aug	Home Page Reader Version 2.5 was released (Japanese, U.S. English, U.K. English, German, French, Italian, Spanish)	
2002 Aug		Jaws 4.5 with advanced Web navigation functions

the top of a page. However, existing tools are not able to check the existence of skip-navigation links, including skip-to-main content links [8].

Even though they are present, however, we sometimes find inappropriate ALT texts and broken skip-navigation links. These pages may be compliant, but they are not accessible or really usable. We analyzed such problems and found that some sites only try to comply with regulations and guidelines, but without understanding the needs underlying Web accessibility. We concluded that Web designers and developers should experience the real problems faced by people with disabilities so they can create truly accessible and usable pages.

In this presentation, I'll first give an overview of the historical progress of the user experience for those who use self-talking browsers and screen readers by referring to my own experiences with voice browsers, and then discuss how much and how well the Web accessibility has progressed by analyzing real world improvements to existing sites. I'll then describe why the "disability experience" helps give a better understanding of the Web accessibility guidelines and regulations. Some tools like Home Page Reader and aDesigner are available to let designers experience blind users' usability. Finally, I'll discuss the future direction of our accessibility efforts.

## 2. HISTORICAL PROGRESS OF BLIND USERS' EXPERIENCE

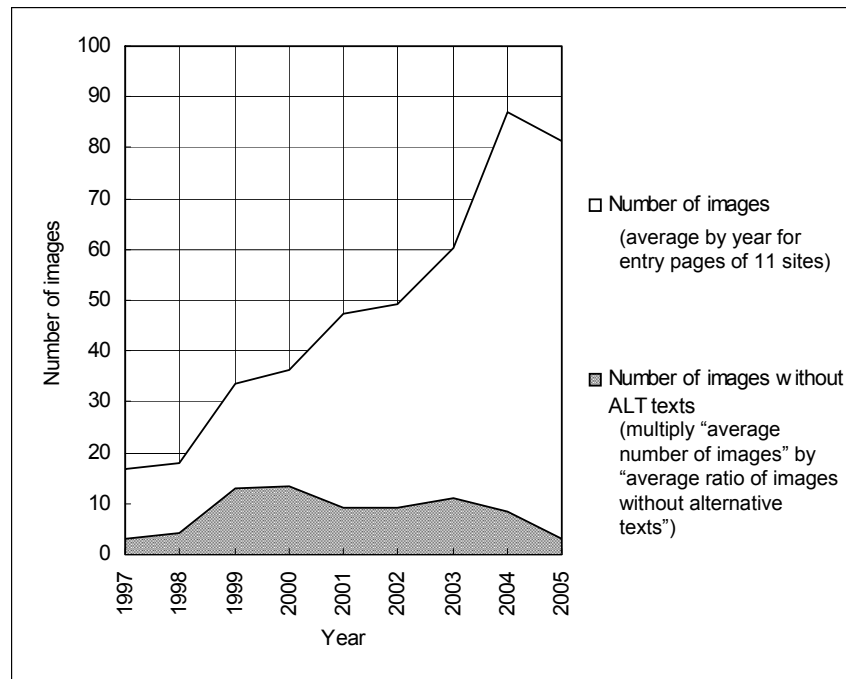
It was quickly obvious that the Web could be invaluable to blind people as an information resource, so the Web adaptation

technologies for them have been pushed along with the improvement of the Web's own technologies. In this section, the history of blind users' experiences on the Web is reviewed by referring to my own experience.

### 2.1 User Experience

Table 1 shows some historical background related to nonvisual Web access, and referring to my own experience with existing screen reading technologies as well as to the development of our voice browsing system.

Lynx, a text-based Web browser, which was developed in 1992, was very accessible for screen reader users, since it could be controlled with just a keyboard. By pressing the cursor right or left keys, for example, users could move to the next or previous link texts. I was able to access the Web for the first time using Lynx in 1994, with a DOS screen reader through telnet by accessing a UNIX server. In the same year, a Japanese text-based BBS provider started a text-based Web browsing service through their BBS service, and I could also access the Web by using their service with the DOS screen reader [9, 10]. The Lynx interface was very accessible, but it was not very easy for end-users to set up the environment for Lynx at home. On the other hand, there was no need to set up anything new for the BBS Web browsing service. The experienced blind users were already used to accessing the BBS services. However, the interface was quite different from the "Web browsing" experience, similar to the command sequence of e-mail. This could not provide an authentic Web browsing experience to blind users [9]. In addition, neither of these approaches was available with plugins, and multimedia



**Figure 1. Increasing the use of images and transition of the number of images without alternative texts.**

data was completely inaccessible with these environments. In addition, the form interface such as used for a Web search engine was not intuitively usable and I could barely use Web search engines.

It was in 1996 that I first experienced a GUI-based Web browsing system [9] using the IBM Screen Reader/2 (SR/2) [11]. In the same year, I also accessed the Web using Netscape Navigator with SR/2. [10] It was drastically easier and very intuitive to navigate through Web pages, to access a Web search engine, and so on, using only a number keypad interface that was developed for SR/2.

In setting up such an accessible environment in the office, it helped that I have better than average access to helpful sighted people at work, including support for our IT environment.

I thought if a similar user interface could be provided on end-users' environments even at home, the Web would be a valuable information resource for the blind people, and not only for experienced blind users, but even for novice blind users. Our research goal is to help blind people be more independent and to improve the quality of their lives using IT.

These wonderful experiences led me to push for the development of a self-talking browser very like a GUI-based Web browser for sighted users. At that time, there was no screen reader or self-talking browser available in the market that provided a similar user experience to that of sighted users. The "similar experience" does not mean the same interface using a mouse and a display. It means a similar experience in intuitively navigating through Web pages and in enjoying surfing the Net even without using eyes. There were still problems. Screen readers, for example tried to read the text information on the screen from left to right. If a table tag was used, the read information became like useless

information. This also happened to such two-dimensional information as frames, forms, and so on.

To overcome these problems, our team followed an approach to analyze the HTML tags rather than just reading the displayed information, and also synchronized with the Netscape Navigator. In this way, plugins for Netscape could be handled and the HTML tags could be read logically and structurally. I also decided to use the number-pad on the keyboard for the navigational commands. All of the commands necessary for surfing the Net could be handled only by pressing a key or a key combination on the number pad.

It was in July of 1997, that I first succeeded in buying my favorite cake through an online shopping site using the prototype Home Page Reader (HPR). HPR became an IBM Japan product in October, 1997 [12]. Then it became an IBM worldwide product [13] in 1999 with a table-navigation function [14] from the IBM Accessibility Center in Austin. Currently, HPR 3.04 is available with much richer functions using Internet Explorer's Document Object Model (IE DOM) API [15].

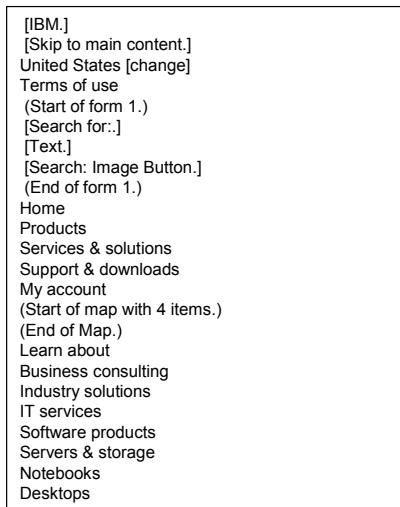
Later, other screen readers started following a similar approach to HPR's by accessing the IE DOM, and one of them also provided advanced Web navigation functions, including a table navigation function, in 2002. Currently, screen readers are providing very useful functions for nonvisual Web access. At least for Web content created in accessible way (see Section 4), blind people are able to access the Web using various kinds of screen readers and self-talking browsers.

### 3. HISTORICAL PROGRESS OF THE WEB ACCESSIBILITY

Currently, user-side technology has become quite accessible and usable, however, the content itself is becoming inaccessible using



(a) Sighted



(b) Blind  
(Home Page Reader v3.04)

Figure 2. Example of large difference between sighted and blind.

various types of visual techniques. Some of the issues cannot be solved with user-side technologies and it is necessary to ask Web designers and developers to provide accessible content. Various guidelines, such as WCAG have therefore been published to indicate the checkpoints for designers. Also, regulations have become effective in strongly pushing them to create accessible content. Here, I'll talk about how much and how well the Web accessibility is improving based on these efforts by focusing on two major accessibility problems for nonvisual Web access.

### 3.1 Insertion of alternative texts

The use of images is rapidly increasing. Figure 1 shows the annual average number of images used in one page for 11 tested sites over the past 9 years. This shows that the number has increased more than four times during this period. If there is no alternative text for these images, blind users will lose a lot of the information that is presented by using images. Alternative texts are extremely important for them to obtain this information from the Web and to navigate through Web pages. All the guidelines and regulations mention providing alternative texts when images are used in a page.

Figure 1 also shows the number of missing alt texts in a page. It increased from 1997 to 2000. However, it decreased suddenly in 2001 from 13.2 images (38%) in 2000 to 9.1 images (19%) in 2001. Investigating the data, we found that all of the images on 5 tested Federal Agency sites had alt texts. US Section 508 became effective in June, 2001. This regulation apparently affected these numbers. In 2004, it appears that private companies also started inserting alt texts. The number of images without ALT text decreased to 3.1 images (7%) in 2005. This was affected by one of the news sites that had only inserted alt texts for 27% of the images (missing alt ratio 73%) in 2004, but that site had improved to 94% (missing alt ratio 6%) in 2005. The use of images has increased steadily, but the ratio for missing alt texts has been decreasing quite consistently among these test sites since 2000. This indicates that the Web accessibility is improving drastically.

### 3.2 Insertion of tags for intra-page navigation

These days, Web pages are becoming larger and visually more complex using layout tables, images, colors and other visual effects. Thus, one page contains more various types of information than ever before. This means that blind users need to spend more time on one page to extract the information they need, since voice browsers (in this discussion, voice browser means both self-talking browsers and screen readers) present information serially using a text-to-speech (TTS) engine.

Figure 2 (b) represents what the Web is like if you can't see it, and Figure 2 (a) is the usual browser view for sighted people. The blind view is just presented linearly and there are no navigational cues such as sighted people use, since sighted people can easily recognize the main content by a large image and large fonts. Therefore, if a page contains a lot of information, for example, it takes a long time to find the main content of the

page. This navigation issue is one of the most severe usability issues for the blind.

There is a technique to solve such navigational problems[21, 22]. By inserting an intra-page link at the top of a page that takes you directly to the main content area, blind users can reach the main content area by just using their command to click that link. The link is called a "skip-navigation link" or a "skip-to-main link". It saves a lot of time and it greatly improves their usability. The link can be inserted invisibly by using an invisible image link so as not to affect the visual layout. Heading tags are also very helpful for blind users to navigate through a page, since current voice browsers provide a heading jump function. Voice browser users, for example, can jump to each heading position by pressing "H" key in major screen readers or voice browsers. Section 508 states that a method shall be provided that permits users to skip repetitive navigation links [5]. Similar rules are also effective in other guidelines.

Figure 3 shows the average time to reach the main content area (article title area) in 2004[8] and 2005 for 12 major news sites (6 from the U.S. and 6 from Japan). From each site, 5 to 25 article pages were randomly selected. Each page was evaluated by aDesigner, using the function to calculate the reaching time from the top to the clicked position. In 2004, only two sites out of 12 provided heading tags or skip-navigation links, however, in 2005, it had improved to eight sites with such tags. It is obvious that the bar in the figure is drastically shortened. This means that more sites now pay close attention to Web accessibility.

## 4. IMPORTANCE OF UNDERSTANDING THE REAL ACCESSIBILITY ISSUES

In the previous section, we learned that the Web accessibility has been improving along with the deployment of the guidelines and regulations. Of course, this is still going on and there are still very many sites that do not pay attention to accessibility, so the Web accessibility adaptation efforts should be continued.

At the same time, now is the time for us to consider the real usability that goes beyond compliance. While I was navigating through Web pages, I noticed that there are many mistakes in understanding the guidelines. Now I'll say a little about my related experiences and the importance of going beyond compliance.

#### 4.1 Inappropriate alt texts

While analyzing the number of images and missing alt texts for Figure 1, various kinds of inappropriate texts were found, such as "spacer gif", "click here", "image" and so on. It reminds me of a funny experience. One day, when I accessed my favorite page, I noticed that the page had suddenly become too noisy with lots of useless information, such as "shadow", "space", "horizontal line" and so on. It took me much more time than usual to find the necessary information. I checked with my sighted colleagues to find out what was happening. By comparing the real image and the corresponding alt text, it soon became clear what had happened. "Shadow" was the alternative information for a gray image with 5 pixels by 30 pixels, which was used for a shadow effect for photos. "Space" was the alternative information for a little spacer image used to create margins between areas. This page had not provided alternative texts for any images before that time. However at that point, they started providing ALT texts for all images, probably knowing that ALT texts should be filled in to improve Web accessibility. If the Web developers used one of the accessibility evaluation tools, it would report that there was no error for the alt texts in this page. Literally, it should be considered as a compliant page. However, it is obvious that these alt texts are not usable or helpful, but actually interfere with blind users' browsing.

In such cases, it is much better to use a null string or a space character as the alt text, so the voice browsers can ignore the images with such meaningless alt texts (see <http://www.w3.org/TR/WCAG10-HTML-TECHS/#link-text-images>). If the Web developers knew about the underlying necessity of the alt texts, this page could be more usable without wasting time. This suggests that the accessibility evaluation tool should provide functions to evaluate the real usability as well as functions to let users experience the real users' behavior.

#### 4.2 Broken skip-navigation link

As discussed in Section 3.2, the use of skip-navigation links is gradually increasing. However, sometimes nothing happens when I select the link. Here are some examples of the problems we discovered:

1. missing destination anchor  

```
<a href="#content"> skip to main content </a>
```

There is no anchor named "content" in the page.
2. anchor name mismatch  

```
<a href="#content">skip to main content </a>
```

```
<a name="maincontent">
```

"content" and "maincontent" are mismatched.
3. wrong HREF  

```
<a href="content"> skip-to-main content </a>
```

This is not an intra-page link, since the # is missing.
4. missing readable text  

```
<a href="#content" alt="skip to main content">
```

```
</a>
```

There's no readable text after the "a" tag,

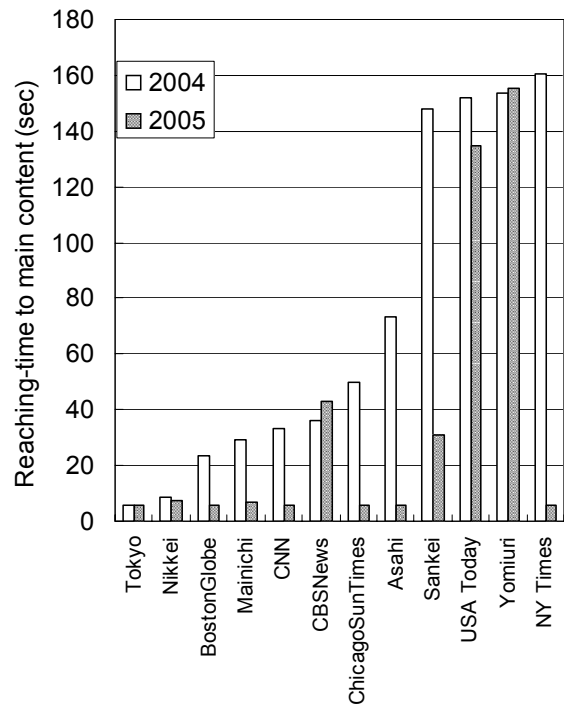


Figure 3. Comparison of average reaching time to main content area in article pages in 2004 (March) [8] and 2005 (March).

but the author of this page somehow thought the ALT would work.

Figure 4 shows the ratio for the broken-navigation links from 6 sites (1 from an IT company, 1 from a NPO, 1 news site, and 3 from federal agencies) in 2004 [8]. By crawling 1,000 pages on the IT company site, we found that 6% of the skip-navigation links were broken. By crawling 1,140 pages on the news site, we found that 5% of the skip-navigation links were broken. The tested sites have actually been paying above average attention to accessibility, so that they do provide skip-navigation links on their sites. However, these problems indicate that maintaining the quality of accessibility while updating the content is very difficult. This suggests that the Web accessibility evaluation tool should provide a high level of checking functions to help those who try to go beyond compliance. In addition, it should provide a function to notify Web developers who have not provided any skip-navigation links in their pages. Learning the necessity of skip-navigation links will help them understand the real usability of voice browser users.

### 5. DISABILITY EXPERIENCE

As discussed in Section 4, some guidelines are misunderstood or only understood literally. It will be effective for Web developers to learn why such checkpoints are defined in the evaluation process. When they find out, for example, that an image should be used with its alternative text, it will be very helpful if they understand how it is used by people with disabilities. This will also help strengthen their motivation to do so. Since the Web accessibility adaptations should be continued throughout the Web

life cycle, it is very important to train them in an appropriate way to maintain the skills.

In this section, we will briefly introduce ways to evaluate the accessibility while experiencing disability using Home Page Reader and aDesigner.

### 5.1 Home Page Reader

W3C/WAI recommends evaluating by using assistive technology software [4] in order to provide highly accessible content. HPR is listed as one of these tools. It was originally developed for blind users, but it is also used for Web developers to experience the blind users' usability [23].

First, sighted Web designers can listen to the all the readable information in a page, just as the blind users hear it. All the spoken information is also presented in the text view, so they can make sure that the visually available information in the graphics view corresponds to the information in the text view. In addition, the corresponding spoken words or positions are highlighted in the graphics view and the text view.

While listening to a page, they will hear some strange information. For example, if there is no alt text for an image link, HPR reads a part of the URL to present blind users with some information to make it clickable. But often this does not provide meaningful information. If it reads "ts\_ppt/2860381fc.html", blind users cannot figure out which page will be opened. When Web developers learned how blind users navigate through links by using a keyboard, they will recognize the importance of these alt texts.

Another example is inappropriately repeated ALT texts, when they will hear meaningless words over and over, such as "spacer, spacer, spacer, ...". When they check the highlighted positions in the graphics view and the text view, they will find that an image for space has "spacer" as its ALT text and understand that meaningless ALT texts should not be described.

When a table tag is used for layout purpose, it sometimes confuses the reading order. HPR reads the table information a cell by cell horizontally. If the information is laid out vertically on the

screen, it reads the wrong cell relative to the Web author's intention. Hearing this, the Web author will recognize that limitation of a layout table.

While listening to a page, they will notice that it takes a long time until the main content is read, and that it takes much more time to hear all of the information in a page when the page is larger. They then understand why the proper use of heading tags and skip-navigation links is important. They can learn to check this by pressing the heading navigation key, since it moves to the heading positions, one after another. They can also test the effectiveness of the skip-navigation link by clicking on the link. This should jump to the main content position by skipping the repetitive navigation links and start reading the main content. If it is broken, however, it just stays at the top of the page and becomes a useless link.

Once sighted users learned the navigation methods with HPR, they will understand how their pages are accessed by voice browser users. This experience will help them in learning more deeply about the accessibility guidelines and regulations, not just reading them literally.

### 5.2 aDesigner

The aDesigner is not an assistive technology program, but rather it is a disability simulator, which was developed for Web developers to help them ensure that their pages are accessible and usable for people with vision disabilities [8, 24]. With HPR, they can evaluate the accessibility with the senses of vision and hearing while using keyboard commands, but with aDesigner, they evaluate it with their own eyes and using the mouse. The aDesigner simulates not only voice browsers' usability, but also low vision users' views, including senior citizens [25]. Here, we will focus on the voice browser aspects.

One of the most advanced functions in aDesigner is to visualize the reaching time to get to every part of a page with voice browsers, using the background color gradation (Figure 5). The darker it becomes in the visualized view, the more time it takes to get there. The reaching time is calculated based on simulating the voice browser basic commands. This is affected by the existence of skip-navigation links, heading tags and so on. If there are no tags for quick navigation (Figure 5 (b)), the background color will be gradually darker and it will be completely blacked out at some point before the bottom of the page. The completely black part indicates that it takes more than 90 seconds to get there. By clicking the mouse on any part of the page, the approximate time to get to the clicked position will be shown. When there is a skip-navigation link or heading tags in a page, the page will be lighter at the heading tag positions or the target position of the skip-navigation link (Figure 5 (c)). The heading information itself is shown by a sky blue background color. If there are several heading tags in a page, the page becomes much lighter (Figure 5 (d)).

The aDesigner thus helps designers to provide heading tags and a skip-navigation link, if there are no such tags in the target page. Once they understand the accessible visualized view, they will be intuitively able to check if a tested page is easy to navigate. The program also checks if a skip-navigation link is broken. If it is broken, there will be no lighter section in the visualized view. Web developers are then able to fix the problems.

The aDesigner's advanced ALT-text checking function of course checks if there is alt text for an image, but it also heuristically checks if the text is appropriate as alternative text. If it is "spacer",

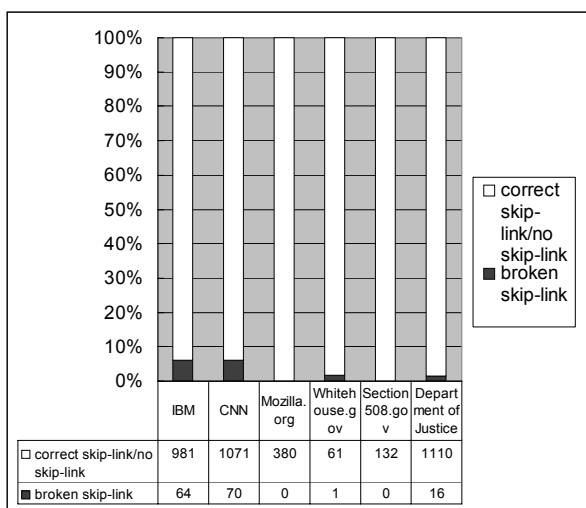
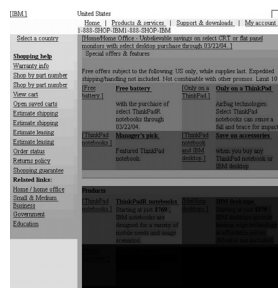


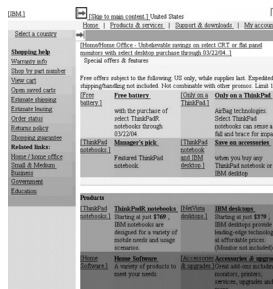
Figure 4. Checking results of skip-links in accessible sites (excerpt from [8])



(a) Original



(b) Visualization (Inaccessible)



(c) Visualization (With skip-link)



(d) Visualization (With headings)

Figure 5. Examples of aDesigner's Reaching-time Visualization (excerpt from [8]).

or "image", for example, it warns users to check and recommends inserting null strings or a space character as the ALT text for a non-significant image.

We just gave an overview of some of the major functions of aDesigner. There are several distinct advantages in using aDesigner. First, it provides functions for Web developers to learn about both compliance and usability in one tool by allowing disability simulations. Second, it only simulates the disability experience, so sighted developers can easily start using it, since it is GUI-based application and there is no need to be trained in the use of an application like HPR. It also checks the guidelines and regulations, so even novice developers can work for accessibility.

## 6. CONCLUSION

In this paper, the historical progress of blind user's experience on the Web was first reviewed by referring to my own experiences with voice browsers. Even though the Web is an extremely useful tool for blind users to obtain information, it has been becoming much visual and thus inaccessible to blind users. Therefore, the social activities involving accessibility guidelines and regulations to improve the Web accessibility were briefly introduced. Then we analyzed how much and how well the Web accessibility has been improved by referring to real sites in the world. This showed that the guidelines and regulations played an important role in spreading awareness of Web accessibility. We saw an increased use of ALT texts and skip-navigation links. However, it was also observed that some designers only pay attention to the letter of the guidelines and regulations, but without understanding the real importance. We see various types of inappropriate ALT texts and

broken skip-navigation links. We concluded that Web developers should experience how people with disabilities access the Web to realize truly accessible and usable sites. HPR and aDesigner were then introduced as tools to evaluate the accessibility while experiencing the disability.

There are still many accessibility issues that are not easily solved, however. The world will be gradually changed by the evolution of technologies and the growing awareness of accessibility in society. The future challenge is to create much more advanced tools for Web developers to learn and evaluate the Web accessibility, while experiencing the real users' behaviors. Even though the disability experience by Web developers is invaluable, it is extremely challenging work to automatically simulate the behavior of a person with a disability person's behavior. Ideally, various types of user models should be provided, such as an experienced user model, a novice model, a child model, and so on. After one of the models is selected, a virtual user should appear and perform some task, simulating a real user's behavior. To realize such functions, there will be two major technical difficulties. One is to model a user type and the other is to evaluate the high level of accessibility while considering usability, which has not been provided by the existing accessibility evaluation tools. The effects of new technology should be also considered in such efforts, since new content always uses new technologies. For example, the tools should deal with DHTML, Java applets, Flash, and so on. By continuing our efforts together with people who pursue the same goals, we will be able to make the impossible possible.

## 7. REFERENCES

- [1] Asakawa, C., Itoh, T., User Interface of a Home Page Reader, In *Proceedings of the 3rd international ACM SIGCAPH conference on Assistive Technologies (ASSETS '98)*, pp. 149-156, 1998.
- [2] JAWS, Freedom Scientific Inc.; see <http://www.freedomscientific.com/>
- [3] Window-Eyes: GW Micro, Inc.; see <http://www.gwmicro.com/>
- [4] W3C Web Content Accessibility Guidelines 1.0; see <http://www.w3.org/TR/WAI-WEBCONTENT/>
- [5] Section 508 of the Rehabilitation Act; see <http://www.section508.gov/>.
- [6] Bobby, Watchfire Corporation; see <http://bobby.watchfire.com/>.
- [7] LiFT; see <http://www.usablenet.com/>
- [8] Takagi, H., Asakawa, C., Fukuda, K., Maeda, J., Accessibility Designer: Visualizing Usability for the Blind, In *Proceedings of the 6th international ACM SIGACCESS conference on Assistive Technologies (ASSETS 2004)*, pp. 177-184, 2004.
- [9] Asakawa, C., Enabling the Visually Disabled the Use the WWW in a GUI Environment, *IEICE Technical Report, HCS96-29*, pp. 39-44, 1996. (in Japanese)
- [10] Asakawa, C., A Web Reader for the Visually Disabled, In *Proceedings of International Workshop on Human Interface Technology 97 (IWHIT97)*, pp. 71-74, 1997.
- [11] Thatcher, J. Screen Reader/2: Access to OS/2 and the graphical user interface, in *Proceedings of the First Annual*

- ACM Conference on Assistive Technologies ASSETS '94* (Nov. 1994), pp. 39-47.
- [12] Asakawa, C., Aiming at information barrier free - referring the experience with Home Page Reader research and development -, *Journal of Human Life Engineering*, Vol 4, No 2, 2003. (in Japanese)
- [13] Laws, C., Asakawa, C., IBM Home Page Reader: The Voice of the World Wide Web, In *Proceedings of Technology and Persons with Disabilities conference (CSUN 1999)*, 1999.
- [14] Asakawa, C., Itoh, T., User Interface of Non-visual Table Navigation Method, In *Proceedings of The ACM CHI 1999 Conference on Human Factors in Computing Systems*, pp. 214-215, 1999.
- [15] Laws, C., Presentation-based Web Accessibility Evaluation System, Design and Implementation Challenges and Trade-offs, In *Proceedings of the Universal Access in HCI (UAHCI 2005)*, 2005.
- [16] Raggett, D., A history of HTML, in *Raggett on HTML 4*, Addison Wesley, 1998. (see <http://www.w3.org/People/Raggett/book4/ch02.html>)
- [17] W3C Web Accessibility Initiative (WAI) First Technical Meeting Agenda; see <http://www.w3.org/WAI/References/970522/agenda.html>
- [18] History of Asahi-net; see <http://www.asahi-net.co.jp/history.html> (in Japanese)
- [19] Asakawa, C., A Web Reader: An Internet Access System for the Blind, In *Proceedings of the Worktech21*, pp. 35-40, 1997. (in Japanese)
- [20] Asakawa, C., Kitamura, K., Itoh, T., A Study of Nonvisual Web Access Using Voice Input and Output, *IPSJ Technical Report SLP17-10*, pp. 53-58, 1997. (in Japanese)
- [21] Asakawa, C., Takagi, H., Annotation-based Transcoding for Nonvisual Web Access, In *Proceedings of the International ACM SIGCAPH Conference on Assistive Technologies (ASSETS 2000)*, pp. 172-179, 2000.
- [22] Takagi, H., Asakawa, C., Fukuda, K., Maeda, J., Site-wide Annotation: Reconstructing Existing Pages to be Accessible, In *Proceedings of the International ACM SIGCAPH Conference on Assistive Technologies (ASSETS 2002)*, pp. 81-88, 2002.
- [23] Testing Web accessibility with Home Page Reader; see <http://www-306.ibm.com/able/guidelines/web/webhprtest.html>
- [24] aDesigner, IBM Corporation; see <http://www.alphaworks.ibm.com/tech/adesigner>
- [25] Maeda, J., Fukuda, K., Takagi, H., Asakawa, C., Web accessibility technology at TRL, In *IBM Research and Development Journal*, Vol. 48, No. 5/6, 2004.