Is Your Web Page Accessible? A Comparative Study of Methods for Assessing Web Page Accessibility for the Blind

Jennifer Mankoff*, Holly Fait, and Tu Tran *****

* HCI Institute	** Exploratorium	***SIMS
CMU, Pittsburgh, PA	San Francisco, CA	UC Berkeley, Berkeley, CA

ABSTRACT

Web access for users with disabilities is an important goal and challenging problem for web content developers and designers. This paper presents a comparison of different methods for finding accessibility problems affecting users who are blind. Our comparison focuses on techniques that might be of use to Web developers without accessibility experience, a large and important group that represents a major source of inaccessible pages. We compare a laboratory study with blind users to an automated tool, expert review by web designers with and without a screen reader, and remote testing by blind users. Multiple developers, using a screen reader, were most consistently successful at finding most classes of problems, and tended to find about 50% of known problems. Surprisingly, a remote study with blind users was one of the least effective methods. All of the techniques, however, had different, complementary strengths and weaknesses.

Author Keywords: Assistive Technologies, Disability, Evaluation, Web Accessibility.

ACM Classification Keywords: H.5.2 Evaluation/ methodology; H.5.4 Hypertext/ Hypermedia: user issues; K.4.2 Social Issues: assistive technologies for persons with disabilities

INTRODUCTION

Web accessibility involves making web content available to all individuals, regardless of any disabilities or environmental constraints they experience. This paper presents a comparison of different methods for finding accessibility problems affecting users who are blind. Our comparison focuses on techniques that might be of use to Web developers without accessibility experience, a large and important group that represents a major source of inaccessible web pages.

As of 1995, there were 8.1 million Americans with visual impairments [15], 1.3 million of whom were blind [2]. As of 1999, 196,000 people over the age of 15 with a "severe limitation in seeing" were reported to have access to the Internet, and half of those were considered regular computer users [15]. In the years since those statistics were published, the number of regular computer and web users who are blind has only increased.

Yet many web pages are still inaccessible. A study of 50 most popular websites found that more than half were only partly accessible or inaccessible [23]. Often, web sites are so inaccessible that blind users simply cannot access all of the information available to sighted users on the web.

Problems blind users experience while using the web range from mere annoyances that cause them to waste time and effort (*e.g.* poorly named links) to critical issues that force them to abandon a task, or get sighted help (*e.g.* important text displayed only in a graphic, form fields with incorrect or missing labels and names). A typical example of a critical accessibility problem that is generally viewed as innocuous or simply annoying by a sighted user is the use of popup windows.

To address the issues of discrimination that inaccessible technologies and information pose to users with disabilities, Section 508 of the U.S. Rehabilitation Act of 1973 was set forth in 1998 to ensure that users with disabilities have access to federal information technologies and properties. Section 508 laid out a set of requirements that all federal web sites must adhere to and greatly heightened awareness of accessibility issues in web design. Similar legislation exists in Europe and a small number of other countries (see http://www.webaim.org/coordination/law for a review).

Despite the importance of web accessibility, most sites remain partly or totally inaccessible [23]. We believe this is due in part to a combination of web developers having little or no accessibility experience, and lack of accurate information about the best ways to quickly and easily identify accessibility problems with web sites. Automated tools [1, 14, 26] and design guidelines [27] fail to create fully accessible sites [13], because they require accessibility expertise on the part of the developer beyond what the majority of developers currently possess, and are unable to detect all problems [9]. A successful, but more expensive alternative to these approaches is a lab study. However, user testing with special populations can incur greater time and monetary costs due to special arrangements for testing, and requires additional expertise on the part of the developer [7, 8]. Further, lab studies with real users tend to happen towards the end of the iterative design cycle, a time when large accessibility problems might be ignored to ship a product or release a site on time.

CHI 2005, April 2-7, 2005, Portland, Oregon, USA.

While the problems with various existing evaluation techniques have been noted in the past [3, p. 344-347], web site designers have access to little empirical evidence when deciding what techniques to use. The goal of this paper is to compare a number of techniques that designers could use to test the accessibility of their websites. We discuss both the quantity and type of problems that different techniques are able to find. We compare expert review, use of guidelines, automated accessibility evaluation tools and remote studies with users who are blind, using a lab study with blind users as a source of baseline data. We chose these evaluation methods because we believe they were most often used in practice and because of their lightweight nature.

Our results show that website developers are best served by asking multiple developers to evaluate a site with the help of both a monitor and a screen reader, using an expert review method. We found that multiple evaluators using a combination of a screen reader and monitor were most consistently effective, finding about 50% of known problems. However, certain classes of problems were missed under these conditions, and other classes of problems could be found as easily using other, simpler techniques.

In the following sections, we discuss why existing methods are not effective in finding accessibility problems. We then present the studies we conducted to compare methods, and discuss the relative merits of different methods.

RELATED WORK

As discussed in the introduction, the most common methods currently used to evaluate the accessibility of a web page include use of automated tools, design guidelines and user studies, or combinations of these. This section discusses how those methods work, and why they are not fully effective for developers without accessibility experience in identifying accessibility issues with web pages.

Automated tools for accessibility testing validate the HTML associated with a web page using accessibility guidelines to create a report of problems for that page. Products like Bobby [1], LIFT [14] and the World Wide Web Consortium (W3C) Markup Validation Service [26] are publicly available to validate sites for accessibility problems. Automated checking tools require a small time commitment, but do not necessarily result in more accessible sites. In fact, Ivory and Chevalier show neither automated tools for accessibility, nor guidelines alone, are adequate for insuring accessibility for disabled users [13]. In a study of a number of automated accessibility testing tools (including Bobby, LIFT and W3C Validator) they measured the accessibility and usability of a group of sites with users and then asked a group of web developers with no reported formal web accessibility training to change the web sites (some using automated tools, others using none). They then asked a number of additional users to evaluate the usability and accessibility of the revised sites. Ivory and Chevalier found that the majority of the tools they evaluated did not help designers create more usable or accessible sites, and that the guidelines embedded in the tools may not necessarily improve the usability and accessibility of the sites more than when experienced web designers (with varying levels of accessibility experience) rely on their own expertise. Problems with automated tools include the length and detailed nature of reports that "make them difficult to interpret, particularly for non-expert Web developers" and that "accessibility guidelines require developers to fully understand the requirements of each guideline" [19]. Additional problems include the limited number of accessibility problems automated tools can find without manual inspection [9] and tools that report that sites have major accessibility problems when they are in fact sufficiently accessible [19].

Various multi-step, iterative processes for producing accessible web sites have been presented [19,22]. Unfortunately, no formal reviews of these methods have been done. Our work complements these processes by helping to advance the understanding of which techniques might best complement each other in an iterative process. Additionally, where developers cannot dedicate the time to conduct both iterative accessibility *and* iterative usability evaluations, our data can help them balance cost and benefit in considering different techniques.

User testing is a common usability method proven effective for finding accessibility problems. The Nielsen Norman Group has published extensive findings on the nature of user studies with participants with disabilities, and how to run them [7, 8]. Such studies are quite effective because they find the problems actual users have with an interface. Unfortunately, user testing with special populations is often beyond the expertise or financial resources of a typical web developer, and is more time consuming than other methods.

Although several recently published books aim to help web developers address accessibility [3,17,24], none of them provide much guidance on how to evaluate accessibility. Both Thatcher *et al.* [24], and Paciello [17], discuss automated tools in depth, and also recommend browser feature manipulation (such as turning off *javascript* or images). They briefly suggest, in a single paragraph each, that the developer use screen readers or other accessibility technology to learn more about a disabled person's experience, or that the developer bring in disabled professionals to judge pages. Clark's chapter on "Certification and testing" recommends avoiding automated tools, and instead testing with real (disabled) users. He argues that "at the very least you need your own adaptive technology and, preferably, you need to include actual disabled users" He discusses the difficulty of using screen readers ("nondisabled people are not very good at pretending to be disabled"), but also argues that testing with disabled users is "essentially impossible in practice." He lists several difficulties that may preclude testing with disabled users, including the difficulty of finding potential users, and the accessibility of the testing location. His conclusion is that "there is no immediately obvious or attainable solution for the problem of testing Websites with actual disabled users." He suggests the hope that outside consultancies will fill this

gap. It is because of these kinds of difficulties that we set out to develop a better understanding of what is possible with the admittedly limited techniques reasonably available to the typical web developer.

In summary, there is little agreement about the best methods for evaluating web pages for accessibility. In the absence of other options, developers are often advised to use automated tools, despite their known flaws. Little is known about the pros and cons of other possible methods. Some comparisons have been done between automated tools [9,13] and between automated tools, guidelines and self evaluation [18], and other techniques have been studied individually [*e.g.* 6,7]. Our work expands on this past work in that we directly compare a wide range of techniques, including techniques that involve external evaluators such as other developers or blind users.

BASELINE STUDY

We began by gathering baseline data on the accessibility of four websites (Table 1), *via* a lab study with blind users. Our goal was to catalog all of the significant problems that blind users encountered when using these websites. This would give us a baseline data set, as well as information about problem severity measured in terms of impact on real users.

The tools most often employed to give blind users access to graphically displayed information are screen readers and Braille displays. Due to issues of cost and fluency, Braille displays are used far less than screen readers. Therefore, we focused our investigation of web accessibility evaluation methods on screen readers and screen reader users.

Screen readers are an assistive technology that allow blind users to hear what sighted users see on their computer monitors. With respect to using the Internet, screen readers parse the HTML of each page visited and read aloud what is presented on the page. Screen readers support a number of specific key commands to browse web pages, find information, enter information into forms, and to read content. For example, each time a screen reader user types TAB, the next link on the page is read to her.

Method

We recruited 5 blind adult computer users, ranging in age from 19-52, with varying education levels (high school to graduate studies). Participants were all legally blind and used only their screen readers for output information about the web pages. All participants used JAWS®, a common, commercially available screen reader. Participants had varying levels of experience using JAWS for browsing the web. One user had less than 2 years experience, two users had 3 to 6 years of experience, and two had more than 6 years of experience. Participants reported using the web to access e-mail (5 of 5), to shop online (4 of 5), to retrieve information (5 of 5) and to read the news (5 of 5).

The participants were asked to attempt one task on each of the four sites, each with differing levels of difficulty (easy to difficult, as rated by the researchers based on blind participants' task completion time and success). Tasks were presented to participants in a randomized order to decrease any learning effects. Tasks included finding the price for oatmeal on an accessible grocery page, finding a bus from one location to another, signing up for a computer class and finding names on a specific page (GUIR site). These tasks represent common daily web tasks, including using forms, searching and shopping. Our tasks were modeled after those used in [7] to identify a well-distributed selection of accessibility problems, and our sites included two commercial sites, one site local to our institution, and one internally developed site.

Table 1: The web pages and tasks for the pages each blind participant was asked to complete. Later, the web developers used these tasks to guide their reviews of the sites.

Web Site	Task	Difficulty
Minneapolis Metro Transit	**Bus**: Use the Trip Planner to enter the information to find a bus from the Mall of America to the intersection of Hennipen Ave and Lake St. W. in Minneapolis, on April 15th at noon.	Difficult
GUIR Home page	**Find Names**: How many faculty members have names starting with J in the GUIR research group?	Medium
Internal class reg. page	**Register**: Register for a class on JavaScript. Fill in all of the spaces and click on the SUBMIT link. (Make up a credit card number and expiration date).	Medium
Albertson's	**Grocery**: Use the accessible site to find the best price for plain oatmeal.	Easy

While they executed the tasks, participants were asked to think aloud, and to mention any problems they had. When participants wanted to make a comment, we asked them to pause the voice output from their screen reader, and to continue it after their comment. Sessions were recorded on video, and all of the web pages and links visited were recorded, as well as what and how information was entered within the web sites.

We reviewed our data, singling out problems that specifically affected the accessibility of a web page or site. For example, the accessibility problem, "Pop-up. Had to ask for sighted help to continue" was included in our final set of problems, while the comment "Their slogan is annoying" was excluded. We also excluded problems that occurred when a participant forgot a JAWS® command because this could not be addressed by a web site developer. Additionally observed problems (unmentioned by participants) were recorded. For example, on one occasion, a participant did not understand that he had encountered a problem due to a pop-up window, while on another occasion, a participant entered a date in an incorrect format without realizing that it would cause a form error. We ranked problems by severity, based on their effect on the participant and his or her ability to complete the task (1, least severe to 5, most severe). As a last step, we grouped like problems (for example, almost all of the participants encountered problems due to a pop-up window in one web site). We counted this as a single problem with that site.

Results

The Albertson's site had 7 unique accessibility problems (severity ranging from 1 to 4), the GUIR site had 3 unique problems (severity ranging from 1 to 4), the Minneapolis MetroTransit site had 10 unique problems (severity ranging from 1 to 5) and the class registration page had 9 unique problems (severity ranging from 1 to 4).

COMPARATIVE STUDY

Before beginning our comparative study, we conducted a number of pilot studies to determine exactly what methods to compare. Because we could find little detail about accessibility testing methods outside of automated tools, we wanted to select specific approaches that were likely to be successful and relatively straightforward for website developers to apply.

First, we explored the feasibility of a sighted developer using a screen reader to test websites. We wanted to discover how to best train developers to use a screen reader, and to get a general feel for whether or not using a screen reader could improve the number of accessibility problems a developer finds. In our first tests, we asked graduate students with web development experience to learn to use a screen reader without using their monitors for visual feedback about what the screen reader was conveying aurally. We found that learning to use a screen reader well enough to evaluate a web page with a monitor turned off required 20-40 hours of practice. However, with the monitor turned on, practice time could be reduced to 10-15 minutes (on average). This is partly because participants had to learn far fewer screen reader features, meaning they might have an inaccurate view of the problems with the websites. However, the presence of the monitor also helped participants to identify some problems they might otherwise have missed, by allowing them to see if the audio output matched the screen output. For example, they could see when text that should have been read was not. Without the monitor they might not have noticed the missing text.

We also briefly tested the effectiveness of using guidelines for evaluation purposes. However, we quickly discovered some important pitfalls to using guidelines in this situation. First, the complete set of available guidelines regarding website accessibility is simply too great to be reduced to a usable number of heuristics. For instance, in [7], over 100 typical errors regarding accessibility problems are listed. The corpus of guidelines, heuristics and rules we collected were simply too diverse and too numerous to create a usable set of everyday accessibility heuristics that encompassed the relevant issues. Second, developers in our pilot study used the heuristics exclusively, and did not point out other problems related to guidelines that were not directly suggested by a heuristic, thus missing more subtle problems (or problems that were not included in the heuristic list). Other problems existed with the usability of the guidelines that have also been verified by other researchers [6].

Based on these initial tests, we eliminated a variant of heuristic evaluation from our list of techniques, and we always used a monitor with the screen reader. We compared the following methods: *Expert Review* – expert review of tasks by website developers (no or little accessibility experience); *Screen Reader* – expert review of tasks with the help of a screen reader and monitor by website developers (no or little accessibility experience); *Automated* – automated review by Bobby ; *Remote* – expert review by remote, experienced, blind computer users.

Method

Our experiment had four conditions, Expert Review, Screen Reader, Automated, and Remote, corresponding to the four techniques we were testing. For the first two conditions, we recruited 17 web developers, with 2 to 8 years of professional web development experience and a median of 40-80 hours per month spent developing web pages. Participants were between 20 and 55 years old and had education levels ranging from high school to graduate studies. Our participants all had little or no accessibility experience. None of the participants had developed accessible websites. Three had taken a class that covered some accessibility issues; five had never taken a class, looked at accessibility guidelines of any kind, watched a blind user, or used an automated tool. Others fell in between this range of activities.

We chose to use developers with little or no accessibility training for two reasons. First, we wanted to avoid any confounding factors regarding experience with designing for accessibility. Second, we wanted to select developers who were representative of the general web developer community.

The web developers were divided randomly into two groups, each assigned to one of the two conditions. In the end, there were 8 participants in our Screen Reader condition and 9 participants in our Expert Review condition. The Automated condition was completed using Bobby 4.0 [1] run by the authors. We chose Bobby because it is a well-established, popular tool for testing web site accessibility. For the Remote condition, we recruited 9 experienced JAWS™ users who were blind. These users were recruited from a mailing list with the stated purpose of providing blind volunteers to help with the specific task of evaluating web site accessibility.

In each condition, participants (or tool) tested the same tasks used in our baseline study (see Table 1). In the first condition, Expert Review, participants were introduced to the Web Accessibility Initiative's Priority 1 Web Content Accessibility Guidelines 1.0, referred to as WCAG hereafter, put forth by the W3C's accessibility initiative. The WCAG guidelines are a standard for web accessibility compliance that covers everything from unlabelled images to the inaccessibility of new technology such as flash animations. Priority 1 guidelines are the key guidelines that must be addressed to meet basic accessibility standards. Participants were then told to complete each task and look for accessibility problems. In the second condition, Screen Reader, participants were introduced to the same WCAG guidelines, and given a screen reader tutorial and practice time that lasted 15 minutes on average (they were allowed as

much time as they wished). They were given a sheet with common screen reader commands that they could use during the study. We asked them to complete each task, with the screen reader running and the monitor on.

In both conditions, participants were asked to only review those pages that fell on the path of the tasks the blind users in our baseline study were given (approximately 3 pages per site). Sites were assigned in random order. Participants were asked to evaluate each site as though they were evaluating a colleague's or friend's site for accessibility issues. Participants reported aloud any problems they found that might cause a blind user to have trouble with the page, and we recorded them. We asked for clarification only if they reported something too vaguely (*e.g.* "The name field is broken", was reported, and we asked, "which one?"). This approach also closely approximates an informal expert review [21], with the caveat that our participants were neither accessibility experts nor domain experts, but rather were web design experts. Additionally, after each task was completed, we asked participants to review the list of problems they had generated, make any additions, and assign severities to each item.

In the Automated condition, we tested each of the sites using Bobby 4.0. Bobby is an on-line tool that tests web pages for accessibility. Bobby returns a list of accessibility guidelines that either have or may have been violated.

In the final condition, Remote, we asked participants to complete each task using a screen reader, from their own computer, email us a list of problems they found, and tell us if they were unable to complete any tasks.

RESULTS

Developers took between 1 and 2 hours total to complete the expert reviews of all four sites, although no time limit was given.

We coded problems into two main categories. The first category was *WCAG*, problems that mapped onto the WCAG

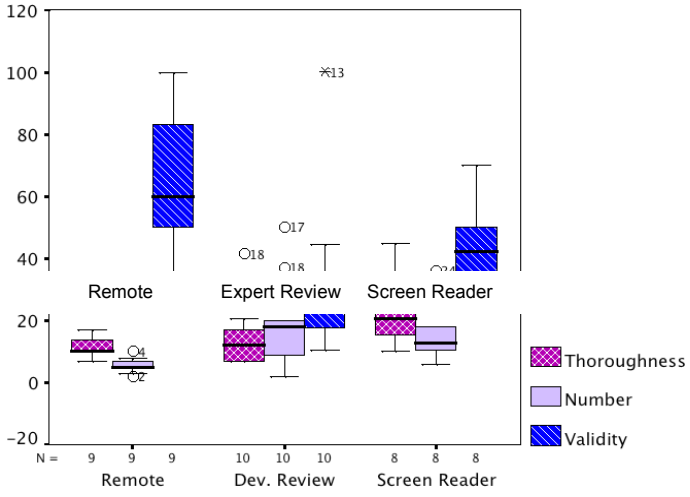


Figure 1: Boxplots showing developer scores on validity, thoroughness, and number of problems found.

priority accessibility guidelines. Note that we did not use structured problem reports [5], but coding into both categories was done twice, independently by two members of our group. Disagreements were resolved by a single arbitrator. By combining all WCAG accessibility problems found in all of our data sets, we generated a set of known WCAG accessibility problems used for calculating percentage of problems found. While WCAG problems are an important measure of success, they do not necessarily represent well the actual problems that a blind user will encounter when using a site in practice (our baseline corpus of Empirical problems). Thus, our second category of problems was *Empirical accessibility problems,* problems that matched those found in our baseline study. The baseline study provided the complete set of empirical problems used for calculating percentages.

Interestingly, there was no strong correlation between the WCAG priority of a problem and the severity assigned to the same problem by developers, or the severity assigned by developers and the severity derived from our baseline study. Additionally, simply meeting WCAG priority 1 guidelines was not sufficient to address the most severe problems found in our empirical data set.

We broke our analysis down into two main hypotheses, stated here as null hypotheses.

1. No condition will be more effective at finding accessibility problems than any other condition.

2. The types of accessibility problems found in each condition will be the same.

Hypothesis 1: *No condition will be more effective at finding accessibility problems than any other condition*. As illustrated in Figures 1, 2 and 3, this hypothesis was false. Some experimental techniques found accessibility problems more effectively than others.

Effectiveness can be thought of as a combination of two metrics, *thoroughness* (what portion of the actual problems are found) and *validity* (a measure of false positives – reported problems that are not real problems [20]).

We compared the performance of each condition on effectiveness, validity, and number of problems found. Note that because the Automated condition had only one data point, we did not include it in our statistical analysis. The median validity and number of problems reported by experimenters in each condition, calculated against our empirical data set, is summarized in Figure 1. There were significantly different distributions among the three conditions, using the Kruskal-Wallis test, for validity (X^2 = 10.755, p=.005) and number of problems reported (X^2 = 10.722, p=.005), but not for thoroughness (X^2 = 7.082, p=.029). Pairwise comparisons using the Kruskal-Wallis test (p<.05) revealed that participants in both the Remote and Screen Reader conditions scored significantly better than participants in the Expert Review condition for validity (median of 60 and 42, respectively *vs*. 23). Additionally,

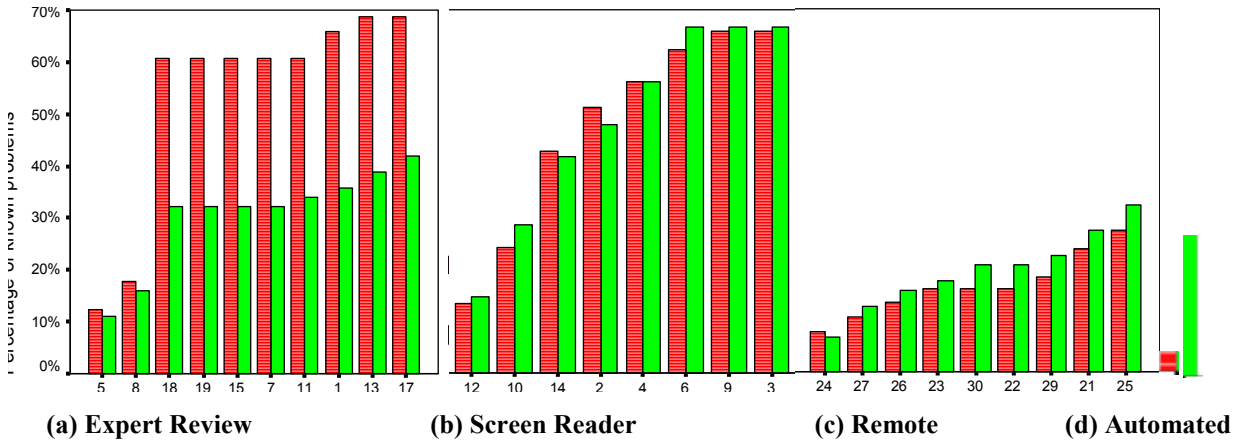


Figure 2: Performance of increasing numbers of evaluators in the (a) Expert Review (b) Screen Reader (c) Remote and (d) Automated conditions. Striped, dark red bars are cumulative percentage of empirical accessibility problems found, while solid, light green bars are cumulative percentage of W3C accessibility problems found. Height is percentage between 0% and 70%.

participants in the Remote condition reported significantly fewer problems than participants in the Screen Reader condition (median of 5 *vs.* 13). It should be noted that these statistics are limited in value by our experimental design, which did not include falsification testing [25] or asymptotic testing [11].

Thoroughness

As illustrated in Figures 1, 2 and 3, the different experimental techniques varied considerably in their thoroughness. Thoroughness was calculated as the percentage of known accessibility problems (from our empirical data set) found by each evaluator. Problems that were not due to accessibility (such as problems caused by unfamiliarity with a screen reader or misunderstanding of the experimental instructions) were excluded from the analysis. We also excluded problems that affected usability but were not specifically about accessibility.

Participants in the Screen Reader condition had the highest average thoroughness (22.84), the Automated condition had the lowest thoroughness (3.49) and participants in the Expert Review and Remote conditions fell in between (14.48 and 11.33, respectively).

It should be noted that not all participants in the Remote conditions were able to complete all tasks. Four of the nine participants failed to complete the *bus* task, and one of the nine failed to complete the GUIR task. All Remote participants completed the other two tasks. However, this may have artificially reduced their thoroughness on one of the worst offenders (the bus task).

In evaluation methods like the ones we were comparing, high variance in the number of problems found by individual evaluators is not unusual [12]. Thus, we analyzed the total number of problems found by combinations of multiple evaluators in each condition. This approach is similar to the use of multiple evaluators that is typical of Heuristic Evaluation [16]. We counted the percentage of unique WCAG accessibility problems, and the unique Empirical accessibility problems found by groups of evaluators in each condition.

In Figure 2, the striped, dark red bars represent the percentage of empirical accessibility problems found by evaluators in each condition, averaged across tasks, while the solid, light green bars represent the percentage of WCAG accessibility problems found by each group of evaluators in each condition. Each group was formed by randomly adding one evaluator to the previous group.

Note that the Screen Reader condition reached over 50% for both types of problems at 4 users, and is the only one that scored above 50% for finding both Empirical and WCAG accessibility problems. Expert Review, in contrast, did equally well after 3 participants at finding Empirical accessibility problems, but far worse at finding WCAG problems (barely above 40% after 10 participants). The Remote and Automated conditions fared worst of all, not even reaching 30% of known problems in either condition. The Automated condition found only 2.5% of Empirical problems and 26% of WCAG problems.

Figure 3 shows the performance of groups of five evaluators on each task, for the two different types of accessibility problems we tracked. We randomly selected four different groups of 5 from each condition, counted the number of unique problems each group found, calculated the corresponding percentage of known problems, and averaged the results for each condition/task. For the automated tool, we used the percentage of known problems it found. Note the reliably high performance of the Screen Reader groups, which had or tied for the highest score in every task, for both types of problems, and found over 40% of known problems in almost every task in both conditions.

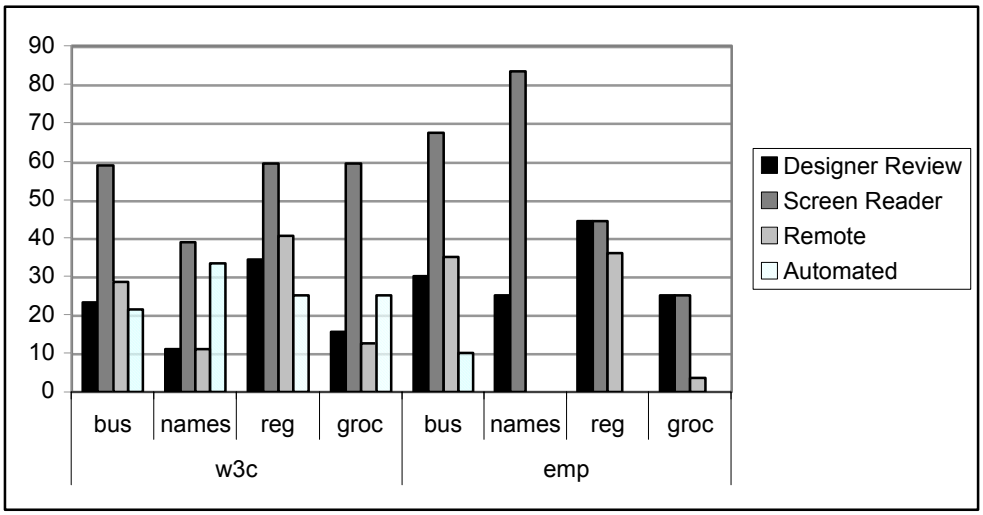
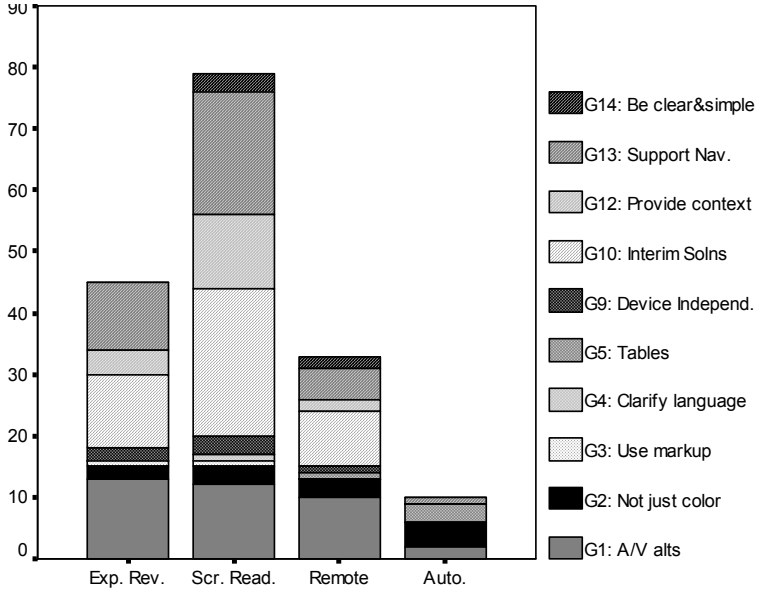


Figure 3: Combined performance of 5 randomly selected participants (in Expert Review, Screen Reader, and Remote) and one tool (Automated), on each task:
Bus (*bus*, difficult task)
Find Names (*names*, medium task)
Register (*reg*, medium task)
Grocery (*groc*, easy task).
Five is generally accepted as a good number in this context [16], and the data in Figure 2 supports that. Note the high scores of the Screen Reader group, which had or tied for the highest score in every task, for both types of problems.



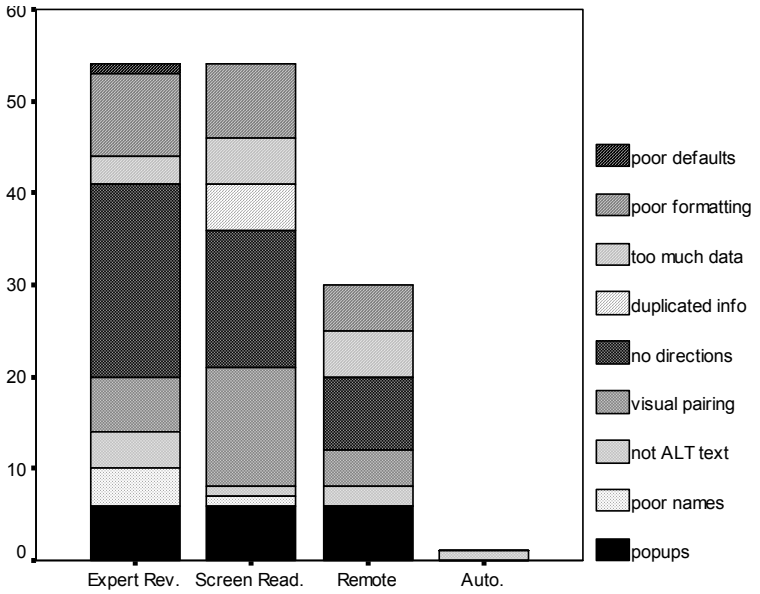


Figure 4: The different categories of problems found in each condition. Rectangle size indicates the number of problems found in that category. **(top)** The 14 major categories of WCAG accessibility guidelines. **(bottom)** Nine categories derived from our empirical study.

Validity

Validity was calculated as the percentage of problems reported by each evaluator that matched known problems (from our empirical data set). For this portion of the analysis, we included all problems that were reported, including screen reader errors and usability problems.

Participants in the Remote and Screen Reader conditions had the highest average validity scores (66 and 42, respectively), the Automated condition had the lowest validity score (2.4), and the Expert Review condition fell in between (32). Recall that the Remote and Screen Reader scores were not significantly different, but were both significantly higher than the Expert Review scores.

Hypothesis 2: *The types of accessibility problems found in each condition will be the same.* As illustrated in Figure 4, this hypothesis was false. Within the two general categories of accessibility problems discussed in Hypothesis 1, some types of problems were found only in certain conditions.

Figure 4 (top) and (bottom) shows the number of unique problems recorded across all evaluators in each condition in problems recorded (on average) by users in each condition in each of the 14 WCAG priority 1 accessibility guidelines (top) and each of 9 categories we defined based on our empirical study (bottom). The WCAG guidelines we used were the priority 1 categories put forth in the WCAG 1.0 guidelines [27]. The Empirical categories were created based on our empirical data set.

The Screen Reader and Remote conditions include problems from the largest number of WCAG categories (nine), while the Expert Review and Screen Reader conditions include problems from the largest number of empirical problem categories. Note that there were slight variations in the number of participants in the different conditions (10 in Expert Review *vs.* 8 in Screen Reader *vs.* 9 in Remote). The top performing Screen Reader condition had the *fewest* participants. Below we discuss each categorization in turn in more depth:

WCAG categories: Only participants using a screen reader (Remote; Screen Reader) found problems in G14 (Be clear and simple). Only the Screen Reader group found G4 (Clarify language), while the Automated and Remote conditions both identified issues related to tables (G5). The Screen Reader and Expert Review conditions both found problems with lack of organizational markup (G3) not found in the Remote and Automated conditions. All four conditions found the particularly obvious problems in the Audio/Visual alternatives category (G1), misuse of bold text (G2), and navigational problems (G13) but those are the only categories that are present in all four conditions.

Empirical: There was much more overlap here than in the WCAG analysis. However, the Screen reader condition was the only one to find problems in the duplicate info category (white, diagonal lines), while the Expert Review condition was the only on to find problems in the poor defaults category (black, diagonal lines). The Remote condition found a smaller variety of categories than the other two, missing duplicate info, poor names, and poor defaults, and the Automated tool did particularly badly.

We asked the developers for qualitative feedback regarding the evaluation technique they were asked to use. Participants in the Screen Reader condition indicated that they would use a screen reader to evaluate sites and that it helped them. Interestingly, participants were over-confident regarding their success in finding accessibility problems.

Discussion

Our results tell us that the Screen Reader technique performed best on both components of effectiveness: on average, participants in the Screen Reader condition were more thorough, while participants in both the Remote and Screen Reader performed statistically equivalently, and better than other conditions, on validity (reported the fewest false problems). Additionally, the Screen Reader technique performed as well as other techniques at finding a variety of types of problems (Hypothesis 2), although it did leave out some types that other techniques found. Below, we discuss conclusions about specific techniques in more detail.

Looking further at our first hypothesis, because of the high variation in performance at finding problems among evaluators, we focused our analysis of thoroughness on understanding the value of combining the findings of multiple evaluators. In that case, participants in the Screen Reader group were still most consistent in finding both WCAG accessibility problems and empirical problems. Groups of five or more participants were able to find 50% or more of known problems of both types, across tasks (Figure 1). While 50% is not a high number (a survey of studies of similar techniques applied to usability shows heuristic evaluation performing in the high 80% range, for example [11]), it is far better than the performance of automated techniques [13,9], or than not running a study at all. When we looked at each specific task, groups of five participants in the Screen Reader condition found as many problems or more problems than groups of five participants in any other condition. In the absence of a screen reader, Expert Review fared very well at finding known Empirical problems (above 50%). In contrast, the Remote and Automated conditions found under 20% of Empirical problems even when up to five evaluators' results were combined.

Considering validity, there was no statistically significant difference between participants in the Remote and the Screen Reader conditions. It should be noted that due to the size of our baseline study, and lack of falsification testing [25], it is possible that we may have labeled some real problems as false positives, making our results for validity artificially low. However, it intuitively makes sense that both of the high scoring conditions involved the use of a screen reader, and we expect that falsification testing would not significantly change our overall conclusions.

With regard to our second hypothesis, no evaluation technique stood out above the rest. However, participants in the Screen Reader condition found at least as many different

categories of problems as participants in any other condition. Participants in the Expert Review condition found one category not found in other conditions (poor defaults). The Remote and Automated conditions were the only conditions to raise WCAG issues regarding tables (interestingly, perhaps because tables used in the sites tested were structural but not visible, we did not find any table-related Empirical problems with these sites).

Overall, the automated condition performed particularly badly. While this was not unexpected [6,9,13,19], our results world likely have improved if we had studied how developers *interpret* those results [13], rather than directly measuring what Bobby was able to find.

We originally expected remote experts to fare far better, especially at finding Empirical problems, than they did. When compared to the lab conditions, the remote participants reported far fewer details and tended to leave out more minor problems. There are multiple possible explanations for this. One possibility is that they were *too* expert at screen reader use. Our lab study included a wider range of experience than we found among the experts who were members of the volunteer evaluator mailing list. Thus, the experts may have simply been more successful at completing tasks than our lab study participants. Another confound is the limited data returned by the remote participants, probably because the technique was structured to rely solely on self reporting. We believe that further refinement of the Remote technique could improve its performance.

CONCLUSIONS

For accessible websites to become more ubiquitous, website developers must have access to lightweight evaluation techniques that support iterative design. However, website developers without accessibility experience face many challenges in testing for accessibility problems. Lab studies are difficult and expensive to run, and if a developer has little accessibility experience they may lead to depressing results late in the design cycle. As an alternative, a plethora of automated tools have been created, with limited success [13,9]. Other techniques, such as using a screen reader, or conducting a remote study with blind users, have been discussed in the literature, but they have not been studied, and few details about how to apply them or what to expect from them are known.

Our results were surprising – not all of the techniques performed as well, or as poorly, as we expected. Multiple evaluators using a combination of a screen reader and monitor were most consistently effective at finding both empirical and WCAG accessibility problems. The analysis in this paper provides guidance about the best way to apply lightweight techniques, and which to avoid. Additionally, our results can help web developers to decide among different lightweight methods, based on the categories of problems they deem most important.

In general, we found that no single evaluator or tool could be counted on to find a high percentage of accessibility problems of any type (WCAG or Empirical). However, *multiple* evaluators, working independently, performed better than individuals. They are most reliable when reviewing sites using a screen reader, but even a simple expert review of specific tasks by website designers searching for accessibility problems was successful in finding some of the most critical accessibility problems (those that led to observed problems in our baseline lab study). Additionally, we found that use of a screen reader significantly reduced the number of false positives reported by individual evaluators.

Developers who do not have access to multiple evaluators might choose to use an automated tool to find WCAG problems, although other researchers have reported that some accessibility expertise is required to interpret the results appropriately [6,19]. Automated web accessibility testing tools and guidelines alone are inadequate for web designers with little accessibility training [6,9,13,19].

Our data on the use of expert, remote blind screen reader users is inconclusive. Evaluators in this condition reported few false positives, but were not very thorough. It is possible that this technique could be further improved if it were modified to encourage better reporting.

The categories of problems not found in the Screen Reader condition include issues with the use of tables raised by the Automated tool and Remote experts, and choice of defaults raised by the Expert Review evaluators. It seems likely that modifying the Screen Reader method to explicitly request feedback on these issues, or combining it with a quick pass of an automated tool, could address these gaps.

The methods we present here are clearly not perfect, as developers found at most 50% of the accessibility problems actually present in our tasks. However, they are all lightweight, an important property during early stages of design. We are not recommending these methods as a substitute for full-fledged user studies involving users with disabilities. Like other lightweight methods, they can introduce false positives, and will not find every problem. However, also like other lightweight methods, they can help developers, particularly developers with little accessibility experience, to find and fix problems at the early stages of design, before they become entrenched.

FUTURE WORK

Appropriate methods for comparing evaluation techniques are an active area of study [5,11,20,25], and techniques such as fasification testing [25], asymptotic testing [11], and structured problem reports [5] could all further improve our analysis.

In terms of expanding the work, we hope to modify and improve our remote testing technique. We are also interested asking accessibility experts to conduct an expert review, and comparing the problems they generate to those found by our users. Lastly, we hope to expand our study to include other disabilities besides blindness.

ACKNOWLEDGEMENTS

Thanks to Carol Pai and Tony Yu Tung Lai for their help on many aspects of this project. Thanks as well to all of the

people who participated in our experiments. Finally, thanks to our reviewers for their thoughtful and helpful comments. This work was supported by NSF grant IIS-0209213.

REFERENCES

1. Bobby Worldwide, http://bobby.watchfire.com/bobby/html/en/index.jsp
2. Bureau of the Census, "Survey of Income and Program Participation," 1994-95, http://www.census.go v/hhes/www/disable/dissipp.html
3. Clark, J., *Building accessible websites,* New Riders, IN, 2002.
4. Cockton, G., Lavery, D. and Woolrych, A., "Inspection-based evaluations," *The Human Computer-Interaction Handbook,* Jacko, J. A. and Sears, A. (editors), pp. 1118-1138.
5. Cockton, G. and Woolrych, A., "Understanding inspection methods: Lessons from an assessment of heuristic evaluation," In A. Blandford and J. Vanderdonckt, (Eds.), *People & Computers XV*, Springer-Verlag, pp. 171-192, 2001
6. Colwell, C. and Petrie, H., "Evaluation of guidelines for designing accessible web content," In C. Bühler & H. Knops (Eds), *Assistive technology on the threshold of the new millennium,* IOS press, 1999.
7. Coyne, K. P. and Nielsen, J., "Beyond ALT Text: Making the web easy to use for users with disabilities," Nielsen, Norman Group, October, 2001, Available at: http://www.nngroup.com/reports/accessibility/
8. Coyne, K. P. and Nielsen, J., "How to conduct usability evaluations for accessibility: Methodology guidelines for testing websites and intranets with users who use assistive technology," Nielsen Norman Group, October, 2001, Available at: http://www.nngroup.com/reports/accessibility/testing/
9. Diaper, D. and Worman, L., "Two falls out of three in automated accessibility assessment of world wide web sites: A-prompt vs. Bobby," In Johnson, P. and Palanque, P. (Eds.), *People and Computers XVII*, Springer-Verlag.
10. Gray, W.D, and Salzman, M.C., "Damaged merchandise? A review of experiments that compare usability evaluation methods," *Human-Computer Interaction,* **13**(3):203-261, 1998.
11. Hartson, H. R., Andre, T. S. and Williges, R. C., "Criteria for evaluating usability evaluation methods," *International Journal of Human Computer Interaction*, **13**(4):373-410, 2001.
12. Hertzum, M. and Jacobsen, N. E., "The Evaluator Effect: A Chilling Fact about Usability Evaluation Methods," *International Journal of Human Computer Interaction*, **13**(4):412-443, 2001.
13. Ivory, M. and Chevalier, A., "A Study of Automated Web Site Evaluation Tools," Technical Report UW-CSE-02-10-01, University of Washington, Department of Computer Science and Engineering, 2002.
14. LIFT, http://www.usablenet.com/
15. National Center for Health Statistics, "National health interview survey - disability supplement," 1994 and 1995
16. Nielsen, J. and Molich, R., "Heuristic Evaluation of User Interfaces," In *Proc. of CHI'90*, pp. 249-256, ACM Press, 1990.
17. Paciello, M. G., *Web accessibility for people with disabilities*, CMP Books, KA, 2000.
18. Petrie, H. and Colwell, C., "Tool to assist authors in creating accessible web pages," In *Proc. of NTEVH 98: Telematics in the education of the visually handicapped*, 1998. http://www.snv.jussieu.fr/inova/publi/ntevh/tools.htm
19. Rowan, M., *et al.*, "Evaluating web resources for disability access," In *Proc. of ASSETS '00*, pp. 80-84, ACM Press, 2000.
20. Sears, A., "Heuristic Walkthroughs: Finding the problems without the noise," *International Journal of Human-Computer Interaction*, **9**:213-234.
21. Shneiderman, B. and Plaisant, C., *Designing the user interface, 4th Edition*, Pearson Education.
22. Sloan, D. *et al.,* "Accessible accessibility," In *Proc. of CUU'00*, pp.96-101, ACM Press, 2000.
23. Sullivan, T. and Matson, R., "Barriers to use: Usability and content accessibility on the web's most popular sites," In *Proc. of CUU'00*, pp. 139-144, ACM Press, 2000.
24. Thatcher, J. *et al.*, *Accessible web sites*, Springer-Verlag, NY, 2002.
25. Woolrych, A. and Cockton, G., "Assessing Heuristic Evaluation: Mind the quality, not just the percentages," in *Proc. of HCI 2000*, pp. 35-36, 2000.
26. W3C Markup Validation Service, http://validator.w3.org/
27. W3C Web Content Accessibility Guidelines 1.0, http://www.w3.org/TR/WAI-WEBCONTENT