Two Falls out of Three in the Automated Accessibility Assessment of World Wide Web Sites: A-Prompt v. Bobby.

Dan Diaper & Linzy Worman

School of Design, Engineering & Computing, Bournemouth University, Talbot Campus, Fern Barrow, Poole, Dorset, BH12 5BB, U.K

Tel: *+44 (0)1202 523172*
Email: *ddiaper@bournemouth.ac.uk*
URL: *http://dec.bournemouth.ac.uk/staff/ddiaper/*

The results of comparing two world wide web accessibility assessment tools, Bobby and A-Prompt, is reported. The two tools were applied to a sample of 32 U.K. university web site home and search pages. Relating the tools' outputs to the Web Content Accessibility Guidelines, A-Prompt found all the guideline compliance failures that Bobby did at both priority levels 1 and 3 and some more that Bobby did not detect. At priority level 2 there was no agreement between the tools as to the compliance failures they detected.

Keywords: accessibility, world wide web, web content accessibility guidelines, assistive technologies, accessibility assessment tools, Bobby, A-Prompt.

1 Introduction

The case for universal accessibility to the world wide web is so patent that legislation already exists, e.g. in the U.S.A. and U.K., or is before various governments, to enforce universal accessibility and, indeed, to punish those who fail to meet legally determined reasonable, minimum accessibility criteria. This, of course, causes great, global confusion, for example, about whether universal accessibility is required legally, in some countries, of all web sites, or just certain types of them. By the end of 2002, for example, two cases on web accessibility in different U.S. states have been before the courts and while the cases are reasonably

similar, both involve transport company web sites (Southwest Airlines and the Metropolitan Atlanta Rapid Transit Authority), the legal judgement has gone in opposite directions in the two cases. N.B. This, of course, is how case law is developed and the legal systems, no doubt, will settle down in a few years time. What is more distressing, except to the cynical, is that anyone should fight against the concept that, ideally, all people should be able to access all of the web, i.e. universal accessibility.

Universal accessibility is an ideal, of course, and "nearly" should be inserted in front of both occurrences of "all" in the definition at the end of the paragraph above. Furthermore, the definition need not be taken to imply that performance won't be different with different web access mechanisms. Eschewing politically correct terminology, the current focus is on extending web accessibility to "those with disabilities [Waddell 1999] and those using 'non-standard' web browsing technology" [Sloan et al. 2002] and, in particular, to cater for web users who are blind or partially sighted.

There are a lot of assistive technologies [Bergman & Johnson 1995] to help the visually impaired by providing alternative output media such as synthesised speech or Braille. The World Wide Web Consortium's (W3C) Web Accessibility Initiative (WAI) web site lists about 70 assistive software tools of various types. Many of these assistive technologies rely on web pages using a standards compliant form of HTML and adhering to accessibility guidelines, notably the Web Content Accessibility Guidelines (WCAGs) produced by the W3C's WAI and, in the U.S.A., Section 508 of the Rehabilitation Act Amendments of 1998. There is considerable overlap between the WCAGs and those derived from Section 508. For example, the Access Board responsible for Section 508 indicates at priority 1 an overlap of 11 check points between the two and 5 that differ; the WCAGs are organised into check points which may be at one of three priority levels, with priority 1 check points being the most important to comply with. There are a total of 66 WCAG checkpoints, with 17, 29 and 20 at priority levels 1 to 3, respectively. Given that W3C's ambition is to be global, this paper will only concern itself with discussing the WCAGs and not with those associated with the U.S.A. centric Section 508 standards.

1.1 Web Content Accessibility Guidelines

Admittedly, not all of the WCAGs and check points are equally clear and the interpretation of some of them requires human craft skill even when assistive technologies are used by web site developers. We can only agree with Sloan et al. that:

> "Current accessibility guidelines require developers to fully understand the requirements of each guideline, the reasoning behind that guideline, and the steps to be taken to meet that guideline. Colwell & Petrie [1999] have questioned the effectiveness of the W3C WCAG in successfully helping developers create accessible resources."

Indeed, our suspicions go rather further in that we doubt that even all those involved in the WCAGs development achieve such a comprehensive understanding of all the guidelines as Sloan et al. say is required, and which we might even question as possible at all with the WCAGs as they are currently expressed. We certainly do not count ourselves amongst such few *cognoscenti* that may exist and we are thus with the vast majority of those concerned and involved with web design.

To illustrate the sort of difficulty that people, including ourselves, have with understanding and applying the WCAGs' check points, check point 12.2 states "Describe the purpose of frames and how frames relate to each other if it is not obvious by frame titles alone. [Priority 2]". Following the appropriate links on the WAI web site leads to the relevant part of their document on HTML design (WAI Note 6) which provides a worked example of a newspaper web site that has three frames described as:

> #Navbar - this frame provides links to the major sections of the site: World News, National News, Local News, Technological News, and Entertainment News.
>
> #Story - this frame displays the currently selected story.
>
> #Index - this frame provides links to the day's headline stories within this section.

While these may or may not be good examples of description links such as 'longdesc' in HTML, they self evidently contain nothing about the relationships between the frames as the #Navbar description doesn't mention the "headline stories" in #Index.

More extremely, we find compliance with WCAG number 14 "Ensure that documents are clear and simple." beyond our own, limited abilities to assure. While there are software tools such as the Clear Language And Design tool (CLAD) which can calculate a reading level to help developers with check points that have some stylistic language requirements, the numerous psychological issues pertaining to different types of document, by different authors, for different readers, in different environments, are such that we believe WCAG 14 is over ambitious on the part of the WAI. We think WCAG 14 should be dropped as part of a desirable simplification of the WCAGs that would make them more understandable and hence usable by a wider range of web developers. We suggest this because not only are the psychological, stylistic language requirements very complicated and not even agreed upon in theory, but also because the concept of simplicity is not simple and "the world is never as (ahem) simple as that" [Smith et al. 1982].

For anyone attempting to apply the WCAGs there are two major types of task: (1) finding accessibility problems; and (2) repairing them. While both these task types requires an understanding of the WCAGs, we think the latter is the easier of the two in that finding WCAG compliance failures really does need the accessibility analyst to understand the check points and be able to apply them all,

appropriately, to every web page assessed. This sort of search activity is extremely difficult to master because analysts have no absolute feedback about their own performance, i.e. they cannot know if they have detected all the possible compliance failures. Obviously different analysts, and the same one on different occasions, may be more or less thorough in their attempts to detect compliance failures. In contrast, most attempts to repair compliance failures are likely to lead to some accessibility improvement, even if the repair is less effective than it could be. We believe that assistive technologies that help web developers detect possible WCAG compliance failures should provide vital support to help with what, for most people, are very difficult detection orientated tasks.

2 Web Accessibility Assessment

To properly assess a web site's accessibility is not easy and it must be necessary to adopt an approach like that of Sloan et al.'s, which involved various teams of people and seven different types of method. Even though Sloan et al. are attempting "discount usability engineering", their method is still far too expensive, in time, money, available expertise, etc., for what are probably the majority of commercial and institutional web site owners. Our suspicion is that a human expertise shortage is the most serious problem because accessibility is a complex Human-Computer Interaction (HCI) issue that requires a fairly sophisticated understanding of people's psychology, i.e. the "rich, multiple perspectives of human thought and behaviour, which have often taken (psychologists) years to acquire" [Diaper 1989]. We suspect that most web developers come from a different, technical computing, background and therefore struggle with what are genuinely complex HCI issues. At present, however, no one recommends that web accessibility assessment tools are used alone, without a human contribution to assessment, but it is just this expertise which we think might be relatively rare amongst web developers.

To give an example that concerns us, like Sloan et al., we have found web sites where the HTML ALT text facility is not merely ignored, but abused. ALT text provides a textual description of non-textual objects, such as images, that assistive technologies for the visually impaired can use, and the WCAG, priority 1, check point 1.1 "Provide a text equivalent for every non-text element" does mean, for the visually impaired, a description of the image and not, for example, as a 'tool tip' and certainly not the same ALT text on every image, e.g. "A picture of this page." as reported by Sloan et al.

The WCAG's do allow an option of just using a "*" as a dummy ALT text, but we remain unconvinced that this option is really a good one. For example, we contacted one university web site developer responsible for their disability pages when we noticed that every image on these pages just contained "*" in the ALT texts. The reply we received was that "The images are of lesser importance." This may not be the institutional position, but illustrates our concern with how real web developers may operate, in this case using a general excuse for limiting accessibility. Furthermore, users who need ALT text descriptions have to trust the web developers' judgement that the images are indeed of "lesser importance".

Perhaps even a legitimate use of "*", if used frequently, indicates a violation of WCAG 14, regarding clarity and simplicity, if pages are cluttered with unimportant images.

Although the WCAGs do try to provide a fairly detailed explanation of what they mean by the term "equivalent" in check points such as 1.1, this is still a matter of interpretation by web developers. While the adage that "A picture is worth a thousand words." is hyperbole in that a thousand words is somewhere between two and four typed pages, just how to describe an image in its specific context on a web page is obviously problematic. A company logo might be a simple thing to put on a home page, but the range of possible ALT texts is vast: at one extreme we might have (1) "The X company logo."; or (2) we might try and describe the visual appearance of the logo; or (3) at the other extreme, one might try to summarise the corporate identity the logo purports to represent. Option (2) is a classical, insoluble problem for those who have been blind from birth and thus, while looking a superficially attractive option as a style of ALT text description, is actually the worst case for such blind people as the description will be meaningless. Option (2), however, might be of high utility to the partially sighted and those who have previously enjoyed the sense of vision. Our point here is that to use ALT text wisely requires not only skill, but probably considerable effort on the part of web developers, if they are to move beyond the tool tip like style of option (1) and significantly improve accessibility.

Since both Sloan et al. and our research investigated university web sites, then obviously we are not suggesting that such institutions have anything but the best intentions about supporting universal accessibility. That both studies do find a considerable number of WCAG compliance failures, including check point 1.1 examples, indicates that there is a genuine problem. Our guess is that when many people are involved in developing a large institution's web site, then many of them will have little expertise about accessibility issues. We also suspect that web site development is often poorly co-ordinated and that some things, like accessibility issues, fall between the cracks between different bits of development. One potential problem with web sites developed by many people is that accessibility assessment tools might indicate someone else's pages are compliant when this isn't really the case, for example, that ALT text place-holders have not been replaced with their intended text.

We strongly support the sort of approach adopted by Sloan et al., and would recommend something like it to any organisation really serious about universal accessibility. For the many organisations that are more resource limited and lack sufficient accessibility expertise, then the role of web accessibility assessment tools becomes increasingly important. Our research is grounded in the real, current world in that we have deliberately chosen to test web sites which we expect to be of a reasonable quality with regard to accessibility issues. Thus, any WCAG compliance failures that are correctly detected represent real and typical accessibility problems with web sites. We assume that the tools' developers have already tested their tools on web pages with contrived compliance failures of all sorts.

The research reported focuses on two web accessibility assessment tools, Bobby and A-Prompt, and compares their relative success. The easy victor is A-Prompt at

two of the three WCAGs' priority levels, and with a poor tie at level 2. The reason for choosing to test Bobby is that it is probably the most widely known of all the web accessibility assessment tools. We chose A-Prompt because it has a similar functionality to Bobby with respect to identifying WCAG compliance failures. An additional reason for choosing these two tools was that they were available free on the web at the time the research was conducted, in the early months of 2002.

2.1 Bobby

Bobby was first developed in 1996 by the Center for Applied Special Technology (CAST) before the development of the WCAGs. Since then, CAST has worked closely with the W3C WAI to support the testing of web sites against the WCAGs. Support for the Section 508 standards was implemented in December, 2001. In July 2002, the Watchfire Corporation "acquired Bobby from CAST and has assumed responsibility for the continuing development, marketing and distribution of the technology." (Watchfire Corporation).

Bobby is probably the most widely known of the many web accessibility assessment tools, which is one reason for choosing it. Web pages that are Bobby and A-Prompt compliant can be publicly badged, which is undoubtedly an incentive to web site owners who want to appear to be concerned with universal accessibility issues, although compliance, without intelligent, careful, human collaboration with these tools, does not guarantee genuine improvements in accessibility.

Bobby version 3.2 was used in the research described in this paper.

2.2 A-Prompt

While the interface to A-Prompt (Accessibility Prompt) looks different from that of Bobby's, both the history of this software tool and its functionality are similar to Bobby's. A-Prompt was developed by a partnership between the University of Toronto's Adaptive Technology Resource Centre (ATRC) and the Trace Research & Development Center at the University of Winconsin. Like Bobby, A-Prompt now tests web pages against both the WCAGs and Section 508 standards.

A-Prompt version 1.0.5 was used in the research described in this paper.

3 Assessing Two Web Accessibility Assessment Tools

The difficulty with creating web sites with known accessibility problems with which to test tools such as Bobby and A-Prompt is that this relies on the creativity of the test web page designer. It is highly unlikely that all of the complex accessibility problems that can arise on real web pages can be anticipated. We assume that the tools' developers have, at least, already taken this approach during the tools' development. There is thus an argument for testing the tools on real web sites where complex accessibility problems can arise and thus to test the realistic usefulness of the tools to organisations.

There are three styles of approach to assessing accessibility tools on real web pages. These, in decreasing order of effectiveness and cost, involve comparing a tool against: (1) a full web site accessibility evaluation; (2) the WCAGs applied manually to a web site; (3) other tools. The first of these would involve comparing a tool's performance to an assessment similar to that of Sloan et al. and, while able to test both how well the WCAGs themselves support accessibility as well as how well assistive tools such as Bobby and A-Prompt perform, this is an expensive approach. The second approach assumes compliance with the WCAGs does improve accessibility and evaluates how well the tools are able to detect compliance failures. The difficulty with this approach is that great expertise is required to manually apply the WCAGs and WCAG experts may not agree on every instance of a potential compliance failure.

The third approach, adopted in the research described, merely requires the application of each tool to the same set of web pages and a comparison is then made of their relative performance. The virtue of this approach is its cheapness and it is much easier than applying the WCAGs manually to a web site because, in this third approach, the accessibility analyst has only to consider whether a specific check point, detected by a tool, has not been complied with at a particular locale on a web page. Against this approach is that there is no independent check of the tools coverage of all the check points, i.e. if two tools are compared and neither detects a compliance failure that is present, then such false negatives will not be detected in the research. On the other hand, this is also a problem with the second approach where, while it is assumed that a WCAG expert will find more compliance failures than the tools to be tested, there can be no guarantee that all potential failures will always be detected by such an expert.

More generally, we believe our comparative approach provides a paradigm example of how assistive technology tools, which purport to have some common functionality, can be usefully assessed. We admit to reaching this opinion *post hoc* of our experimental analyses and the surprising results we discovered. Similar research, using a larger number of assistive technology tools, is planned and we hope other researchers will be attracted by the efficiency and cheapness of the approach.

3.1 Web Sites Assessed

While we started by examining a range of commercial and government web sites, we felt that U.K. universities (and the larger Colleges of Higher Education) provided a good set of web sites with which to fairly test the two tools. At a high level of generality, the set is relatively homogeneous in that all U.K. universities have similar, large web site requirements reflecting each university as a large institution, with many parts to it, with a variety of complex functions and services [e.g. Hales & Hazemi 1998]. The universities have a public duty, made explicit, for example, by the U.K. Joint Information Systems Committee (JISC), to take into consideration the needs of different types of users. Furthermore, given that one major class of university web site visitors will be potential students from around

the world, then users and their computer platforms are as about divers as it is possible to be.

To ensure the sites we used did indeed reflect some concern for the disabled, from more than 100 university and college web sites sampled, the 32 sites chosen all had more than three pages on their web site devoted to disability issues. These web sites were all HTML based as neither Bobby or A-Prompt claim to cope with more specialised web design environments. None of the web pages tested was badged as A-Prompt or Bobby compliant. Restricting ourselves to HTML based sites also facilitated our manual inspection of web pages.

These university web sites each contain many hundreds of pages. A web site's home page is of critical importance [Nielsen 2000] and often determines whether users proceed further into the site. Home pages are thus more likely to have had greater care taken on them than some pages buried deeper in the site. They are also more likely to contain images, tables and other non-text objects, which, for example, should have ALT texts attached to them, than many deeper pages which are often mostly text. Thus the reason for testing home pages is that they provide opportunities for potential WCAG compliance failures to occur and we might expect some attempt to have been made to make such pages as accessible as possible.

All the 32 sites provide a search facility which is either on the home page (in 14 cases) or accessed from the home page (18 cases). Using such search facilities involves rather different tasks from those associated with viewing pages so as to acquire information and thus there are likely to be some different accessibility issues arising from such different tasks. WCAG 13 "Provide clear navigation mechanisms", of course, is particularly relevant, e.g. check point 13.7 "If search functions are provided, enable different types of searches for different skill levels and preferences. [Priority 3]".

3.2 Method

Each web site's home and search pages were submitted to Bobby and A-Prompt for accessibility assessment and the WCAG check point compliance failures, at each priority level, that each tool detected was recorded.

For the rarer WCAG compliance failures, as well as a sample of the common ones, the source HTML was inspected to test for false positive results, i.e. where a failure is reported that should not have been. Provided the check points are interpreted generously, then false positives did not appear to be a problem with either tool, i.e. we could always see why the tool had reported a compliance failure, even if sometimes we might then decide that it did not require fixing, which is sometimes the appropriate action recommended by the WAI.

We already knew that A-Prompt tests at priority 1 for D-links, which are textual descriptions in addition to HTLM's 'longdesc' ones, and that Bobby does not do so. We have therefore ignored A-Prompt's detection of missing D-links when comparing the two tools.

3.3 Analysis

The possible results of the research are summarised below:

R1 A good result for the tools would be if they performed identically, detecting the same WCAG compliance failures. Such a result would not demonstrate that the tools provided complete coverage of all the WCAGs, but it would give a measure of confidence in that such consistency would mean that the WCAGs were being interpreted in the same manner and that the tools are useful to organisations for detecting real accessibility problems that do occur, even on quality web sites.

R2 A good result for one tool would be if it detected all of the WCAG compliance failures of the other tool and some more as well. We might not abandon the poorer performing tool because it may be better able to detect compliance failures that were not present in the sampled web sites, but if only one tool were to be used, then this sort of result would favour choosing the better performing one.

R3 A poorer result for both tools would be where there was little or no overlap between the WCAG compliance failures that each detected. We might not be confident about the tools, even if we elected to always use both tools on web sites in future, because this sort of result does indicate that both tools are only partially covering the complete set of WCAGs and it is possible that some lack of coverage is shared by both tools and which, of course, comparative tests between them cannot detect.

R4 Both Bobby and A-Prompt will detect some WCAG compliance failures which are present on a web site, so it is not necessary to consider the truly disastrous outcomes, provided, of course, that the web sites assessed do contain compliance failures.

There are several possible reasons why the tools might not report the presence of many types of WCAG compliance failure. Perhaps most importantly, many of the check points simply do not apply, for example, because the web pages assessed don't use moving images, audio or "new technologies" (Guideline 6) or involve "user control of time-sensitive content changes" (Guideline 7), for example.

The research reported was carried out in the early part of 2002, and while newer versions of both Bobby and A-Prompt will continue to be released, we believe our results are timely and that the issues the results raise will continue to be germane for some time to come.

3.3.1 Statistical Analyses

Non-parametric statistics are used for the usual reasons that such statistics are suitable for relatively small samples because their ordinal counting system basis

makes them immune to any effects of skew and kurtosis within the data [Siegel 1956; Miller 1975]. The Wilcoxon Match Pairs Rank Sum Test was used to compare the results from the two tools and to test for differences between sites' home and search pages where these are on different pages. The Mann-Whitney U Test was used to test for differences between home pages that incorporated a search facility (N=14) to those where search was carried out from other than the home page (N=18). As the latter involves testing two web pages, then the sum of the compliance failures detected for both pages was divided by two before comparison with the data from the pages which have search engines on their home pages. All tests were two-tailed as no predictions were made as to which tool might perform better than the other.

3.4 Results

Neither Bobby or A-Prompt differed in the number of WCAG compliance failures they detected on the eighteen home and separate search engine pages (Wilcoxon T = 39.5 and 46.5, respectively), although the Bobby result is on the border of the 5% significance level. The range of the number of compliance failures detected within these pages was 4-10 and 3-11 for Bobby and A-Prompt respectively, but the range of differences between the home and search pages was only 0-3 and 0-5, respectively. The number of ties is likely to decrease the statistic's sensitivity. Comparison of the fourteen home pages with search engines included to the eighteen where the search facilities are separated found no difference in the number of compliance failures detected (Mann-Whitney U = 106.5 and 92 for Bobby and A-Prompt, respectively). Given the lack of difference between these two styles of web page, then all subsequent analyses are on all 32 web sites tested.

Bobby found some compliance failures on every web site and A-Prompt on all but two of the sites.

Table 1 shows the average number of WCAG check point compliance failures detected per site, at each priority level, for each tool individually and the total number that are detected by both tools. N.B. the first two figures in each column do not sum to the third, "Total Different", because the same failure detected by each tool is counted only once, i.e. a perfect R1 result would make the three figures in each column identical.

	Priority 1	**Priority 2**	**Priority 3**
Bobby	0.47	3.56	2.05
A-Prompt	1.12	3.12	3.53
Total Different	1.21	6.34	3.75

Table 1. Mean number of WCAG compliance failures, per web site, detected at priorities 1, 2 and 3 for Bobby, A-Prompt and the number of different compliance failures detected by both tools.

Combining the results across all three priority levels, then A-Prompt finds many more compliance failures than Bobby (Wilcoxon: $z = 3.89$, $p < 0.001$). There are

too few failures detected at priority 1 for statistical analysis purposes and a worrying number of ties at priorities 2 and 3. Inspecting the data, A-Prompt gains its clear superiority over Bobby at detecting WCAG compliance failures at priority 3: A-Prompt detects more failures than Bobby on 29 web sites; they tie on one; and Bobby does better than A-Prompt on the remaining two sites.

Figure 1 shows graphically and numerically the average, per web site, number of check point failures detected by each tool at each priority level. The central portion of the bar represents the average number of check point failures that both tools detected and the outer portions those detected only by either Bobby, on the left, or A-Prompt, on the right: the length of the bars is consistently proportional across all three priority levels. The arrow above the bars represents the average number of check point failures per site detected by A-Prompt and the arrow below the bars, the average detected by Bobby (see also Table 1).

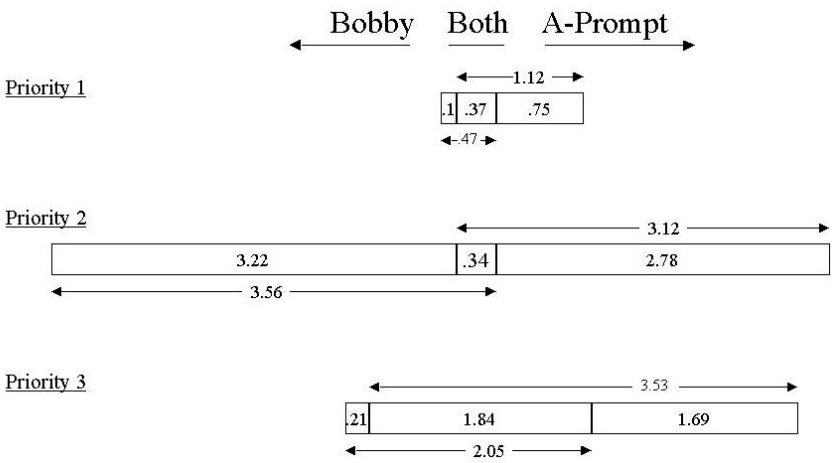

Figure 1. Mean number of WCAG compliance failures, per web site, detected at priorities 1, 2 and 3 which: only Bobby detected, on the left hand side of the bar; only A-Prompt detected, on the right hand side of the bar; and the compliance failures detected by both tools, in the centre of the bar. The arrow above each bar represents the mean number of WCAG compliance failures detected by A-Prompt and the arrow below represents this for Bobby.

Figure 2 shows the same data as Figure 1, but representing the average number of check point failures per site as a percentage of the number detected at each priority level, hence the bars are all the same length.

Given that the university web sites were selected as quality ones that publish some concern about disability issues, then it is good news for the universities that only a small number of compliance failures were found at priority 1: on average, just over one failure per site (1.21). Of the 17 check points at this priority, only three different ones were detected by the tools (17.6% coverage). The three check points were:

Check Point 1.1 Provide a text equivalent for every non-text element.

Check Point 6.3 Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative accessible page.

Check Point 8.1 Make programmatic elements such as scripts and applets directly accessible or compatible with assistive technologies.

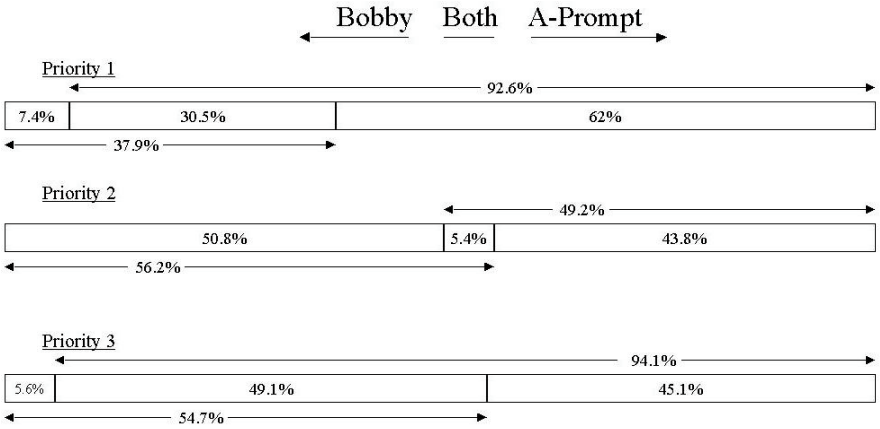

Figure 2. Percentage of WCAG compliance failures, per web site, detected at priorities 1, 2 and 3 which: only Bobby detected, on the left hand side of the bar; only A-Prompt detected, on the right hand side of the bar; and the compliance failures detected by both tools, in the centre of the bar. The arrow above each bar represents the percentage of WCAG compliance failures detected by A-Prompt and the arrow below represents this for Bobby.

At priority 1, Figure 2 clearly demonstrates a strong type R2 result in that A-Prompt detects virtually all of the check point 1.1 failures that Bobby detects and, unlike Bobby, it also detects check point 6.3 and 8.1 compliance failures. The latter two are obviously closely related, although A-Prompt sometimes detected only one of them on some sites. We were not surprised at finding a few check point 1.1 failures as sites, particularly around their home pages, tend to have a lot of images and occasionally it is easy to forget to provide the additional ALT text. Both Bobby and A-Prompt provide a valuable check that there is some text but, as discussed earlier, they don't ensure that the text is suitably equivalent, or even meaningful. Our suspicion from looking at the web sites with the "programmatic" failures detected is that these are sometimes invisible to web developers, particularly when many developers are involved and our "falling between the cracks" hypothesis is the one we advance as their most probable cause.

The number of types of compliance failure detected is small, but with quality web sites, particularly at priority 1, this is to be expected. Bobby's complete failure to detect the "programmatic" compliance failures supports using A-Prompt rather than Bobby if a site is only interested in covering priority 1 compliance and

only one assessment tool is to be used. Furthermore, A-Prompt's ability to detect D-link compliance failures, which Bobby does not detect and which were ignored in our research, also thus favours choosing A-Prompt over Bobby at priority 1.

The priority 3 results, while three times as many compliance failures were detected as at priority 1 (3.75 v. 1.21), are very similar to the priority 1 results in that A-Prompt detects all the failures that Bobby detects (taking the 5.6% in Figure 2 to be effectively 0%) and detects a similar number that Bobby fails to detect. Also like the priority 1 results, the coverage of all priority 3 check points is relatively small, with only 5 out of a possible 20 types of failure being detected. At the priority 3 level, however, we suspect that the tools are actually weaker at detecting compliance failures, not just because priority 3 is of lesser consequence than priority 1, but also, we suspect, because some of the priority 3 check points are harder to interpret, by person or machine. If only one accessibility assessment tool was to be used, then the evidence again strongly favours A-Prompt over Bobby.

The priority 2 results are quite different from those found at the other two levels. Five times as many failures are found at priority 2 compared to priority 1 (6.34 v. 1.21) and not quite twice as many as at priority 3 (6.34 v. 3.75). Between them the tools found 15 different types of check point failure out of a total possible at priority 2 of 29 (51.7%). Critically, the results show a strong type R3 result in that the two tools detected completely different types of error; only 5.4% of the failures were detected by both tools.

In summary, the pattern of the results is: type R2 at priorities 1 and 3, favouring A-Prompt; and type R3 at priority 2.

4 Discussion and Conclusion

The only way to properly assess a web site's accessibility is to undertake some approach like Sloan et al.'s, but quite a number of people were involved in a range of methods in their work and we don't think most organisations have the resources, and particularly the expertise, to assess their web site's accessibility in such a thorough way. While one might imagine organisations hiring an expert accessibility team, against such a one-off approach is that large web sites tend to evolve continuously so there is a strong case for in-house expertise to ensure that accessibility is maintained. There must be a strong temptation for organisations to rely more on accessibility assessment tools than they should because the tools are supposed to encapsulate and apply knowledge about the WCAGs and check points. Accessibility tools have their own knowledge over-heads concerning how to use the tools and, vitally, interpret their outputs. Indeed, it may well be that at present the tools' related knowledge is additional to a sound understanding of the WCAG by the tools' users, i.e. you need to know more to use the tools, not less. This still makes accessibility tools valuable as a contribution to efficiently finding possible WCAG compliance failures, which are difficult for people to always find reliably, provided that the tools do find most of them.

For anyone wishing to assess the accessibility of their web site using either Bobby or A-Prompt, then our results should give them cause for concern. Even

though A-Prompt better performs at both priorities 1 and 3, detecting virtually everything Bobby detects and quite a lot more, the results at priority 2, where both tools detect a large number of types of compliance failure but with no agreement between them, would make the sensible advice to be to use both tools. Furthermore, generalising these results, we might suggest that until further research is conducted on other tools, then the best strategy is to use as many web accessibility assessment tools as possible. While sensible, we doubt that the 'use many tools' advice will be adopted, not least because of the overheads associated with using each tool and, sometimes, there may be conflicting advice between the tools. Our comparative approach to tool evaluation is cheap by comparison to other approaches and we think it provides a paradigm example that can, and should, be extended to many such related tools.

One reason for publishing our research is that we were shocked by the results. What we had expected when we started was that we would find type R1 results, with the tools mostly agreeing but with a few exceptions, which we had expected to investigate in detail. We were interested in looking at the performance of these tools on real web sites of reasonable quality with respect to accessibility because we have assumed that the tools will have been tested on blatant examples of compliance failures during their development, i.e. we intended to investigate the utility of these tools to organisations that had probably tried to support universal accessibility and we wanted to know if the tools could help in such cases. Noting that Bobby is probably the most well known web accessibility assessment tool, then our results show that A-Prompt is a better tool at priorities 1 and 3. The priority 2 results are a mess and, we think, reflect a nascent technology. The priority 2 results suggest that the tools functionality, i.e. their compliance failure detecting abilities, needs further development and we think that the tools warrant some improvement in their usability. We also think that further work is needed on the WCAGs and check points; to us they look like something designed by a committee and could do with some simplification, perhaps by reducing the number of check points by making greater use of the many relationships that already explicitly exist between the check points. We favour abandoning WCAG 14 regarding keeping everything as clear and as simple as possible as the concept of simplicity is not simple.

Overall, if you are only going to use one web accessibility assessment tool then, based on our research, use A-Prompt rather than Bobby. Much more importantly, we don't recommend placing too much trust in either tool, particularly below priority 1.

Our research cannot stand the 'test of time' in that the WCAGs, Bobby and A-Prompt, and other web accessibility assessment tools, continue to be developed. Our research, however, represents a 'shot across the bows' to organisations who are perhaps over confident of the performance of tools such as Bobby and A-Prompt and it offers a warning to such tool developers of both the difficulty and distance they still have to cover before their tools are functionally sufficiently adequate that they can be relied on. Our comparative approach to testing is efficient, easy and cheap and we hope that others will apply it to a wider range of assistive technologies.

References

A-Prompt – http://www.aprompt.ca/ (last accessed February, 2003).

Bergman, E. & Johnson, E. [1995], "Towards Accessible Human-Computer Interaction" *in* Nielsen, J. (ed.), *Advances in Human-Computer Interaction*, 5.

Bobby – http://bobby.watchfire.com/bobby/ (last Accessed March, 2003).

CLAD – http://www.eastendliteracy.on.ca/ClearLanguageAndDesign/start.htm/ (last accessed April, 2003).

Colwell, C. & Petrie, H. [1999], "Evaluation of Guidelines for Designing Accessible Web Content", *in* Buhler, C. and Knops, H. (eds.), *Assistive Technology on the Threshold of the New Millennium*, IOS Press, 39-47.

Diaper, D. [1989], "Giving HCI Away" *in* Sutcliffe, A. & Macaulay, L. (eds.), *People and Computers V*, Cambridge University Press, 109-120.

Hailes, S. & Hazemi, R. [1998], "Reinventing the Academy", *in* Hazemi, R., Hailes, S. & Wilbur, S. (eds.), *The Digital University: Reinventing the Academy*, Springer, 7-24.

JISC – http://www.jisc.ac.uk/ (last accessed March, 2003)

Miller, S. [1975], *Experimental Design and Statistics.* Methuen.

Nielsen, J. [2000], *Designing Web Usability*. New Riders Publishing.

Siegel, S. [1956], *Nonparametric Statistics for the Behavioural Sciences.* McGraw-Hill.

Sloan, D., Gregor, P., Booth, P. & Gibson, L. [2002], "Auditing Accessibility of UK Higher Education Web Sites", *Interacting with Computers*, **14**(4), 313-326.

Smith, D.C., Irby, C., Kimball, R., Verplank, B. and Harslem, E. [1982], "Designing the Star User Interface" *Byte*, **7**(4). *Reprinted in* Baecker, R.M. and Buxton, W.A.S. (eds.), [1987], *Readings in Human-Computer Interaction: A Multidisciplinary Approach,* Morgan Kaufmann, 653-661.

Waddell, C. [1999], "The Growing Digital Divide in Access for People with Disabilities: Overcoming Barriers to Participation in the Digital Economy", http://www.icdri.org/CynthiaW/dig_div1.htm/ (last accessed April, 2003).

W3C – http://www.w3c.org/ (last accessed March, 2003).