World Wide Web (2012) 15:61–88
DOI 10.1007/s11280-011-0130-8

Web accessibility guidelines

A lesson from the evolving Web

Simon Harper · Alex Q. Chen

Received: 2 December 2009 / Revised: 4 April 2011 /
Accepted: 6 April 2011 / Published online: 27 April 2011

Abstract The World Wide Web (Web) is in constant evolutionary change. This evolution occurs along many fronts and is led by infrastructure developers, Web designers, technologists, and users. These multiple stake–holders ensure that the Web is a heterogeneous entity, not just in the nature of the content, but in the technology and agents used to deliver and render that content. It is precisely this heterogeneity which gives the Web its strength and its weakness. A weakness in technology adoption leading to an increasing disconnect between the actual user experience and the expected experience of the technology stakeholders. We are interested in the human factors surrounding the evolution of the Web interface; and believe that the wait is always too long for new accessibility recommendations, guidelines, and technology to be adopted. In this case, we describe a ten-year longitudinal study comprising approximately 6,000 home pages. From this study we conclude that as a ‘rule-of-thumb’ mainstream technology is adopted at about 15% within the first three years, incremental version releases are adopted at about 10% within the first three years. However, sites which are most popular often exhibit enhanced adoption rates of between 10 and 15% over the same period. In addition, we see that accessibility guidelines are mostly ignored with only a 10% adoption rate after more than ten years. From this we infer that, for maximum accessibility adoption, guidelines might be supported and reflected in mainstream specifications instead of remaining only as a separate document.

Keywords Web accessibility · visual impairment · measurement · evolution

S. Harper (✉) · A. Q. Chen
School of Computer Science, Information Management Group, University of Manchester,
Kilburn Building, Oxford Road, Manchester M13 9P, UK
e-mail: simon.harper@manchester.ac.uk

1 Introduction

The World Wide Web (Web) is a heterogeneous set of technologies,[1] recommendations, and guidelines which are in constant, and combinatorial, evolutionary change.[2] The Web is seemingly self regulatory in that there are so many stake–holders that their combined interaction makes Web evolution somewhat unpredictable. While the heterogeneity of the Web is one of its major strengths, in that it is adaptable and flexible, it also one of its major failings in that there is no control of the predicted outcome. With homogeneous technologies (such as C, Java, SQL, and even scripting languages such as perl and Ruby) there is a degree of developer led conformance, and this conformance is useful for consistency across the user experience. However, the Web cannot guarantee this conformance and therefore the interactive experience cannot be assured.

We are interested in the human factors surrounding the evolution of the Web interface and the Human Computer Interaction (HCI[3]) aspects that exist in large scale heterogeneous systems. We believe that the wait is always too long for new accessibility recommendations, guidelines, and technology to be adopted. Indeed, waiting for User Agent adoption and then Web designer use, means that it can take considerable time for the user experience to become as originally intended. Further, it also takes significant time for the failings of the mainstream technology design to become apparent when released into a domain with so much scope for use other than as was intended. This work stems from an anecdotal assertion that:

> Accessibility guidelines and recommendations are adopted at a significantly lower rate than mainstream Web recommendations and technologies.

It is the aim of this paper to support this assertion with empirical work and deductive reasoning. In this case, we need to understand the rate of technology, recommendation, and guideline adoption if we are to support our assertion. We have created an experiment utilising the Internet Archive[4] to understand these factors. By performing this study we answer the questions: ‘Do we rely on technology or guideline adoption?’, ‘Do we need technical intervention?’, and ‘Should we be led by users, by engineers, or by past history?’ In this way we are able support our anecdotal assumptions and thereby better guide future research giving us a clear rationale and domain to place new and continuing work.

In this paper we first outline our research domains of Web Interaction and Accessibility (see Section 2), and follow these with related Web evolution work (see Section 2.3). We then describe our rationale (see Section 3) and methodology (see Section 4) in which we present a ten-year longitudinal study comprising

[1] Here we use core technology and core standards interchangeably.

[2] Here, we take the Oxford English Dictionary definition of Evolution, being ‘Of, relating to, or of the nature of a process of natural or gradual origination, transformation, or development’, by no means do we assert that this change is necessarily positive for all parties affected by that change.

[3] Or Computer Human Interaction (CHI).

[4] http://www.archive.org/: Founded to build an Internet library, with the purpose of offering permanent access for researchers, historians, and scholars to historical collections that exist in digital format.

approximately 6,000 home pages; the code and an accompanying technical report are openly available [10]. Next we present our results and analysis (see Section 5); again, all pages analysed and an accompanying technical report are openly available [11]. We then discuss this analysis (see Section 6) and determine that core standards take around three years to gain a 15% coverage, that de-facto[5] technology releases are adopted at the point of need, and that guidelines are mostly ignored. This analysis enables us to deduce that if guidelines were to be subsumed into mainstream technical specification of Web based languages, accessibility adoption would increase. Finally, we conclude (see Section 9) that without a perceived benefit for users, there is little technology adoption.

2 Background

Into the mix of infrastructure, guidelines, technology, and recommendations come Web designers mostly using a very complicated set of tools in a very non-standard, often unintended, and heterogeneous way. In this case the only point where we can truly understand how the Web page will be rendered, presented, and interacted with is at the moment it is displayed on the specific client-side user agent (or assistive technology) chosen by the user. This makes understanding, and testing for, Web accessibility conformance much more difficult. In addition, while work has been undertaken at the intersection of Web accessibility and Web evolution there has been little work of the kind undertaken here and there are no sources which specifically would enable us to support our anecdotal assertions. Studies to understand the extent of accessibility and assist in addressing these failings have been undertaken [24], but, these studies have only considered how changes in Web design have affected accessibility over time, and suggested possible solutions to address these problems without a full scale accessibility re-evaluation.

2.1 Web accessibility

Web accessibility aims to help people with disabilities to perceive, understand, navigate, interact with, and contribute to the Web [37, 43]. The importance of the Web for information dissemination is indisputable but the dominance of visual design on the Web leaves visually disabled people at a disadvantage [5, 6]. While Assistive technologies, such as screen readers,[6] usually provide basic access to information, the richness of the Web experience is still often lost. In particular, traversing the Web becomes a complicated task since the richness of visual objects presented to their sighted counterparts are neither appropriate nor accessible to visually disabled users [4, 30, 34, 42]. Indeed, access to, and movement around documents has long

[5] Latin meaning ‘by [the] fact’; a technology (such as a standard) that is found in the common experience and is created or developed without regulation or reference to a standard. In computer science this often means that the technology has become a standard by use as opposed to via a standards authority diktat.

[6] A *Screen reader* is an application that can be used to read content (e.g., Web page or software) in audio.

been considered an important and major issue in the Web design and usability field [12, 20]. The commonly used slang phrase 'surfing the Web' implies rapid and free access, pointing to its importance among designers and users alike. It has also been long established [5, 6] that this potentially complex and difficult access is further complicated, and becomes neither rapid nor free, if the user is visually disabled. Web accessibility, in broad view, depends on several different components of Web development and interaction working together, including Web software (tools), Web developers (people) and content (e.g., type, size, complexity, etc.). The W3C[7] Web Accessibility Initiative (WAI) recognises these difficulties and provides guidelines for each of these components [8, 23, 46, 47]. There are also other organisations that have produced guidelines (e.g., IBM, RNIB, AFB, Adobe, etc.) but the WAI guidelines are more complete and cover the key points of all the others. There is however, no homogeneous set of guidelines that designers can easily follow. Furthermore, the introduction of the Web Content Accessibility Guidelines (WCAG) version 2.0 has changed the conformance landscape. In this case, WCAG 2.0 has four principles, 12 guidelines, 61 testable success criteria and a number of techniques for making Web content even more accessible. Although WCAG 2.0 comes with a number of useful techniques for conformance testing, most of the test procedures suggested represent manual evaluation and repair approaches [1]. This means that automated validation and repair, with developers asserting the conformance of aspects which are not machine testable, may well be no longer acceptable.

2.2 Web interaction

Web Interaction focuses on improving technologies that provide interaction with the Web. This is led by the W3C's Interaction Domain,[8] which is responsible for developing technologies that shape the Web's user interface [41]. These technologies mainly include (X)HTML,[9] which is the markup language that started the Web, Cascading Style Sheets (CSS), which provides a mechanism for adding presentation style to Web pages, Scalable Vector Graphics (SVG), which can be used to create two-dimensional graphics in XML, etc. Development in these technologies determine how people browse the Web, and how they author Web content [48]. Therefore in any effort to support Web accessibility it is crucial that features and limitations of these technologies are clearly stated. As part of the W3C's Interaction Domain, the Multimodal Interaction Working Group seeks to extend the Web to allow users to choose an effective means to interact with Web applications through the modes of interaction best suited to their needs and device (visual, aural and tactile). This activity is focused on providing use cases and requirements analyses which are important resources for future Web design. In the Web accessibility field there are also best practice efforts which mainly include developing tools to ensure

[7] The World Wide Web Consortium (W3C) is the premier body for the formulation of Web Standards.

[8] http://www.w3.org/Interaction

[9] eXtensible HyperText Markup Language, used in this paper to also include the HyperText Markup Language (HTML).

accessibility, such as validation, transformation and repair tools. Validation tools analyse pages against accessibility guidelines and return a report or a rating [26]. Repair tools, in addition to validation, try to repair the identified problems. Although there has been extensive work in the degree and development of validation, repair and transformation tools, automation is still limited [25]. While it is likely that there are certain accessibility issues that cannot be fully automated (e.g., checking the quality of alternative text provided for images), these tools still provide incomplete automation and complex outputs.

2.3 Web evolution

Work already exists which looks at other forms of Web evolution from the perspective of communities and social groupings [35] or from the viewpoint of information retrieval with regard, specifically, to the lifespan of individual pages [14, 16]. Studies in the domain have attempted to investigate many differing aspects of the evolving Web. These have focused on attempts to quantify the size [22], rate, nature, and extent [36] of changes to Web resources [18]; to observe the rate of change of the Web; and to develop exponential probabilistic models of the intervals between the changes of a Web-page [7]. This work has focused on the identification and analysis of various design choices for incremental search engine crawlers [14]. These studies found that 69% of all accesses were images, 20% was (X)HTML, and the rest were applications. They found that 23% of Web pages changed daily, approximately 47% of Web pages changed within four months, and about 30% took more than four months to change. They conclude that the rate and nature of change to the Web resources, frequency of access, and information life-span mainly depends on the content type and top-level domain, but not the size. Therefore, by building space-saving technology such as CSS and XML into Web pages, network traffic can be reduced and download times decreased. Further attempts to understand the dynamic nature of the Web in the context of search engines have been conducted [16]. This work finds that by understanding evolution, increased effort can be concentrated on crawling and indexing pages which have changed [35]; this is similarly the case for Content Management Systems [32]. These studies find that coverage of different search engines varied by around 20% of the Web indexed: 76.2% for Google; 69.3% for Yahoo!; 61.9% for MSN; and 57.3% for Ask/Teoma. Indeed, the Web seems to be highly volatile; from 1999 to 2000, about 60% of URLs disappeared, thereafter between 2001 and 2002, about 30% disappeared in four months alone [45], but, more than 70% of pages had a lifespan of more than one month. Indeed, work between 2000 to 2003 further supports this notion of change, finding that Web site designs became increasingly graphical in nature, reliant on browser scripts, and less consistent [27].

3 Predicting the future

Creating an accessible Web is a difficult goal mainly due to the complex interplay of the Web's infrastructure; guidelines; technology; and content. Indeed, it is this final point of Web content design, along with the view of the designer, which often creates the most difficulty when we wish to understand Web evolution in the context of Web accessibility. It is often the case that technology or guidelines are created to address perceived problems with access to the Web, however there is often little thought as to the time-scale that these solutions will take, or how these solutions may evolve in the future especially when we consider their non-standard use.

Research in the Web accessibility field looks at understanding the way users, often visually disabled users, interact with the Web and from this understanding enable technological solutions which are inserted between the user and delivered Web content. These solutions are normally situated on the client interface or as part of a proxy system. These technical platforms, or prototypical experiments, are useful in two ways. Firstly, by encoding a researchers hypotheses into these platforms a statement is made regarding the predicted outcome of the user experience, and by testing users with these technologies these predicted experiences can be ratified, thus they enabled the research findings to be tested. But secondly, and in the 'real-world' at least, more importantly, they enable a better user experience than the one which would have been available if the browser alone provided the interaction.

To better guide the research community and understand the context of the research output we see the need to support our initial assertion:

> Accessibility guidelines and recommendations are adopted at a significantly lower rate than mainstream Web recommendations and technologies.

However, we are often faced with the comment that research solutions are no more than temporary fixes, and that once new technology and guidelines are adopted these problems will be solved. This argument also completely ignores the fact that interdisciplinary understanding of human behaviour in the context of Web interactions may go beyond the knowledge of the technologists or guideline designers. Indeed, we would go further and suggest that research technology is vital as a demonstrator of leading edge user agent technology too. The reality is that we have two, often opposing, viewpoints on Web accessibility issues: (1) There is the viewpoint which suggests technological recommendations and guidelines will enable Web accessibility, and that these will render assistive client-side modifications superfluous or even harmful;[10] and (2) there is the opposing view that technological interventions must be created because recommendations and guidelines are often too slow to be adapted, and in some cases are not adopted at all.

Our intention then, is to understand the way that the Web has evolved over the last ten years in an attempt to support our arguments: that (1) Based on past trends, technological interventions, encapsulating leading edge research, must be created and allowed to evolve as new findings are discovered; and that (2) Research technology is useful for a similar amount of time as a user agent version; after all we do not consider the most up-to-date version of Mozilla Firefox to be a 'temporary fix'.

In these ways, we see that by understanding the way that the Web *has* evolved we can predict the way that the Web *will* evolve.

[10] See the many discussions on the British Computer Association of the Blind (BCAB) http://www.freelists.org/archive/bcab and the various WAI http://www.w3.org/WAI/ mailing lists.

4 Methodology

For our purposes an understanding of Web evolution can be based along two tracks: Web Archives,[11] and Web Archaeology.[12] In our work we use archival methods, tied to the Internet Archive, while supporting the archival trends with a ground truth. From this viewpoint we can define the following complementary terms:

Archive An archive is a set of Web pages placed into a protected, or ring fenced, store at or near the time they were created, with the intention of using them at a later day for some unspecified process or purpose. The archive however, suffers from some sociological aspects because there needs to be a methodology for placing the Web pages into the archive and this normally means there is an associated perceived importance.

Archaeology Web pages outside of a defined archive, in the wild—if you will, which happened to be preserved in a much older state than the rest of the current Web. These pages are often found in the so-called long-tail [2, 19] and are in many cases not linked to a large number of other 'current' pages. The pages tend to be forgotten remnants created in bespoke and individual cases as opposed to parts of large corporate Websites or Web sites that are currently maintained. They are therefore useful in contextualising an archive view and maybe thought of as a kind of social history.

Ground Truth Ground truth is a term used in cartography, meteorology and a range of other remote sensing techniques in which data are gathered at a distance. Ground truth refers to information that is collected 'on location'.[13] The collection of ground truth data enables calibration of remote-sensing data, and aids in the interpretation and analysis of what is being sensed. Here we borrow the term changing the spatial context for the temporal one. In this way we can compare our archive trends with a ground truth for current wild Websites.

4.1 Data collection

For our work, we build a Web robot which uses the Internet Archive as its page corpus. Web robots, are software applications that run automated tasks over the Internet. Typically, bots perform tasks that are both simple and structurally repetitive, at a much higher rate than would be possible for a human editor alone.

We select our pages by using site rankings from 'Alexa' to pick the top twenty sites that have been valid over the last ten years. Next we build a randomised corpus of five-hundred sites culled from 'Google Open Directory'. We then take snapshots in

[11] An archive refers to a collection of historical records, and also refers to the location in which these records are kept.

[12] The recovery, documentation, analysis and interpretation of material remains and environmental data.

[13] In remote sensing, this is especially important in order to relate image data to real features and materials on the ground (http://en.wikipedia.org/wiki/Ground_truth).

time between 1998 and 2008 and analyse each home-page for each site against a set of analysis criteria; each snapshot representing one sample. In addition, we confirm our archive findings by a process of 'ground truthing' in which we increase the sites analysed and collected as a snapshot of the current 'wild' Web. In this case we took a June 2008 sample spanning the Alexa top 500 sites along with a further random 5000 from Google Open Directory. Comparing this sample to our archived sample allowed us to see if the trend in the archived data continued into the wild present.

It is unnecessary to build a corpus of Web pages which represents a greater number per samples than our standard 520. According to Netcraft[14] there are 75,200,000 active Websites. Testing 10,000 Websites, this means that only 0.0013% of the active Web would be tested. Indeed, we would need to test 752,000 Websites per sample to achieve just a 1% testing. Luckily, the level of confidence in statistical tests of external experimental validity is absolute and not relative to the sample size. This means that even with such seemingly small sample sizes we can still achieve a confidence interval of 5/95 making a '*p*' value of 0.05. The key factor is good sampling; selecting a representative sample of pages found on the Web. A sample must reflect the population: major categories should be represented and this can be narrowed down further by choosing sites which are of a typical representation (sites with large numbers of visitors, and longevity). By choosing the top sites from Alexa along with a random sampling from Google Open Directory we ensured that we worked with a set of pages which reasonably represent usage.

4.2 Analysis criteria

Our analysis was focused on understanding the similarities and differences in core, de-facto, and accessibility guidelines such that a comparison can be made between them. In this way differences between mainstream core and de-facto standards can be compared to accessibility guidelines. This comparison is required to support or confound our assertions. In this case, each home page was analysed against the following criteria (contained within simple regular expressions, these can be found in full in the related technical report [10]):

Core Technology Here we wanted to build a picture of core technology adoption against the release timeline (see Figure 1). We focused on (X)HTML and CSS usage in an attempt to answer the question 'What is the uptake of core technology with regard to the Web evolution scenario?' We investigated these aspects by capturing the (X)HTML version built into the DOCTYPE header of each Web page. We also parsed the (X)HTML and sectioned the tags into specific (X)HTML versions, if the DOCTYPE was not available, or the DOCTYPE definition did not match the elements used, we took the (X)HTML elements as being definitive. In this way we validate the conformance level with regard to markup specifications and grammars. In addition, we detected the presence of CSS styling at the

[14] http://news.netcraft.com/archives/2009/01/16/january_2009_Web_server_survey.html

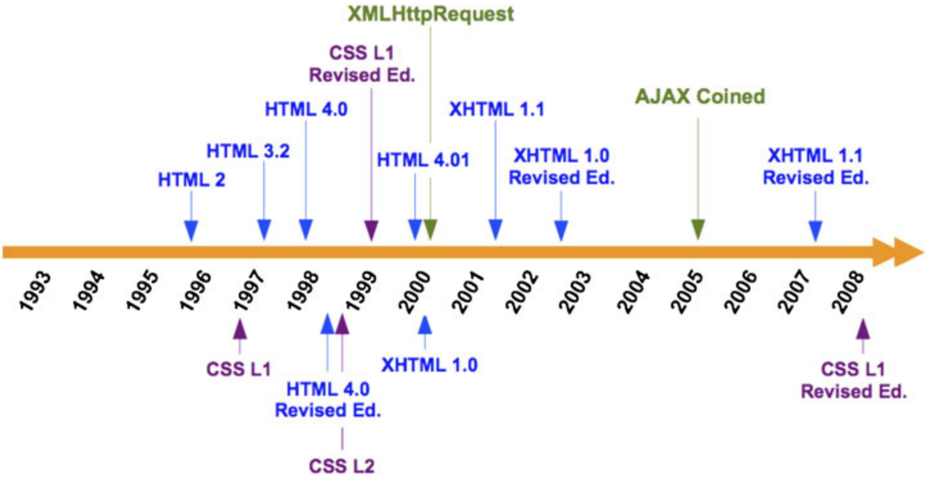

Figure 1 Web standards milestones

document and inline levels, and the presence of external CSS files attached to the document.

De-Facto Technology In this case we wanted to know what the patterns of de-facto technology adoption look like and how they relate to core technology. We therefore chose to investigate scripting technology and dynamic or asynchronous updates. Even though JavaScript has been available for some time we expected other competing scripting technologies to be at least as popular. To detect the usage of client-side scripting, we parsed the the "<script>" element looking at both the "type" and "language" attributes. In the case of dynamic updates we used two techniques: (1) the detection of "XMLHttpRequest"[15] in the JavaScript (both inline and in linked files); and (2) the usage of the HTML "iframe" element.

Accessibility Guidelines In this case we looked at WAI WCAG v1 (AAA, AA, A) and 508. We investigated these by an analysis of: (1) The hyper-links to validation sites; (2) The various image accreditation badges; (3) A combination of AAA, AA, A, and 508 characters; or (4) The presence of the word 'accessibility'—present anywhere within that last 100 characters of a page to account for statements and links to intra-site pages describing policy. Remember we are not interested in whether the guidelines have actually been followed as opposed to the perception that the

[15] Is is a common method of supporting Asynchronous JavaScript and XML (AJAX), which is a group of interrelated Web development techniques used for creating interactive Web applications or rich Internet applications.

designers think them important enough to list on their Web pages.

In this way, it could be said that this is an approach to measure how 'trendy' accessibility is or how accessibility aware the developers are, and if accessibility is important enough to consider notarising its presence on the page; as opposed to conformance to strict accessibility measures. Indeed, we consider this method to be reasonably optimistic as no testing was actually performed for this study; just the desire to be accessible on the part of the developer was enough to register a hit. In addition there are currently no automated tools that can yet check for conformance to all guideline checkpoints; for example, checkpoint 1.1 states that the author should 'Provide a text equivalent for every non-text element' however there is no automated way to test equivalency, only the presence, or not, of alternative text. In this case human evaluation is needed and so a manual analysis of accessibility for every site would be impractical. However, it may be that even if a company uses WCAG they will not make a claim on their Websites due to possible legal implications or other social issues; therefore, 'using' does not represent 'conforming'.

However, to investigate the historic data in this regard is beyond the scope of this paper, especially due to changes in the guidelines and the validation technology,[16] post 2008. In this case we decided to check accessibility conformance using of our 2008 Alexa Top 20 using AChecker, EvalAccess, and WAVE, and then manually check each page when we found conformance warnings. This enabled use to speculate about possible errors across the data sets while maintaining our optimistic approaches because we expect the top twenty Websites to be professionally developed, by well informed teams, representing large organisations, with budgets for testing and validation.

By analysing these three key areas we were able to understand, and plot, the adoption of technology both in leading edge sites and in general over the Web.

5 Findings

Our results and analysis are divided in to Core, De-Facto, and Guidelines section, and comprise graphs depicting the percentage of each set of sites using differing

[16] Web Content Accessibility Guidelines (WCAG) 2.0 became a W3C Recommendation as of 11 December 2008. Changes to the guidelines did not occur between these two dates. However WCAG 2.0 became active in 2008 forcing a change to the validation engine's technology to cope with these changes.

technologies over a 10 year period. From these graphs we can more easily understand the trends present as the Web evolves.

5.1 Core technology

Core technology is split into six graph timelines representing HTML 2 (see Figure 2) through to XHTML 1.1 strict (see Figure 3) along with CSS (all versions—see Figure 4). The solid line shows the top 20 sites over time and gives us a slightly erratic record of usage, however, when taken in concert with the random 500 (the dashed line) and the June 2008 ground truth we can see that a number of clear patterns emerge. We can see that HTML 2 maintains an average 20% share of HTML used trending down to 15% towards the present. HTML 3 (see Figure 5) on the other hand, never seemed to enjoy a full scale adoption and trends down from 12% to 4% over the 10 year period. While it is difficult to infer information pre-1998, when taken in combination with HTML 2 and 3, we can see drop-off in HTML 2 and 3 uptake followed by an increase in HTML 4 usage post 1998. HTML 4 (both versions—see Figure 6) seem to have enjoyed most successes maintaining a 70% share until starting to trend down to 55% from July 2006. The downward trend can be explained by the rapid take-up of XHTML 1.0 trending up (see Figure 7), to 20%, by the same date and showing a marked upward trend to 30% by June 2008. XHTML 1.1 strict on the other hand shows little adoption over the random 500/random 5000 and the top 20 and top 500 with between 1 and 2% over the last three years. We account for the small spike in our XHTML figures pre 2000 by early adopters reformulating HTML 4 to XML (ie XHTML); proposed as a working draft from as early as December

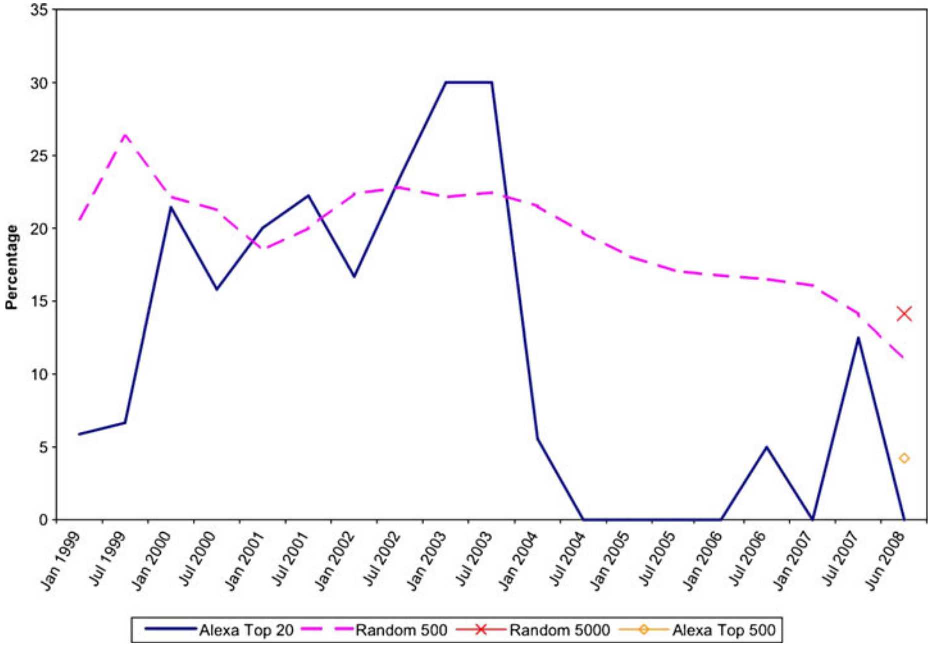

Figure 2 HTML 2 (Lines represent Archive, *circle* and *cross* represent Ground Truth)

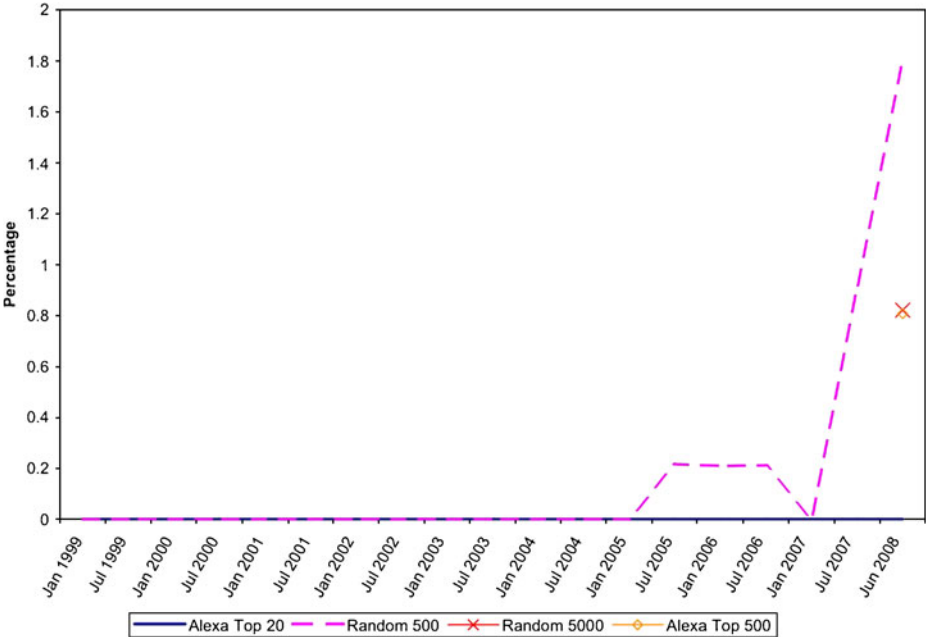

Figure 3 XHTML 1.1 Strict (*lines* represent Archive, *circle* and *cross* represent Ground Truth)

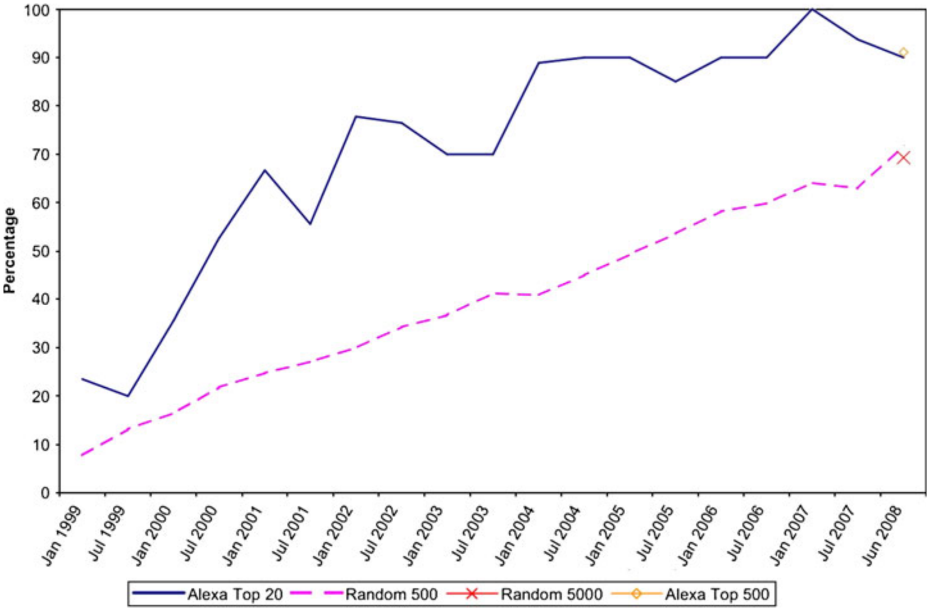

Figure 4 CSS (all versions—*lines* represent Archive, *circle* and *cross* represent Ground Truth)

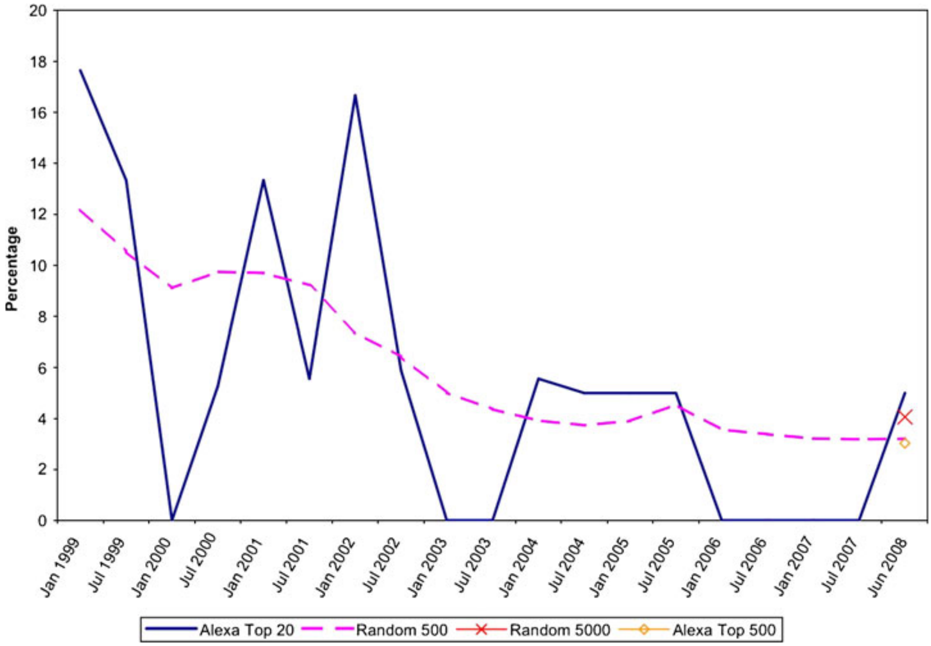

Figure 5 HTML 3 (Lines represent Archive, *circle* and *cross* represent Ground Truth)

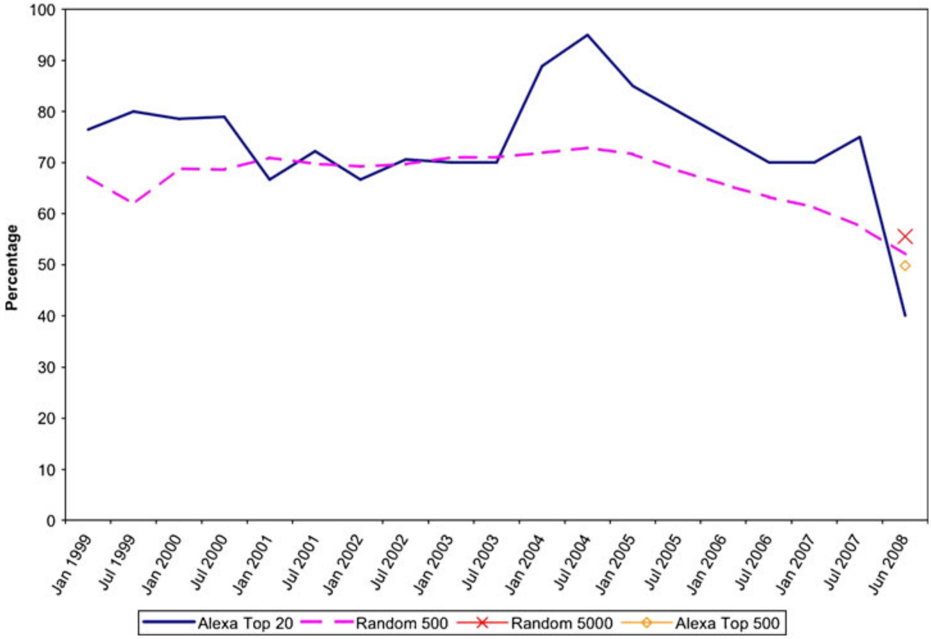

Figure 6 HTML 4 (versions 4 and 4.01—*lines* represent Archive, *circle* and *cross* represent Ground Truth)

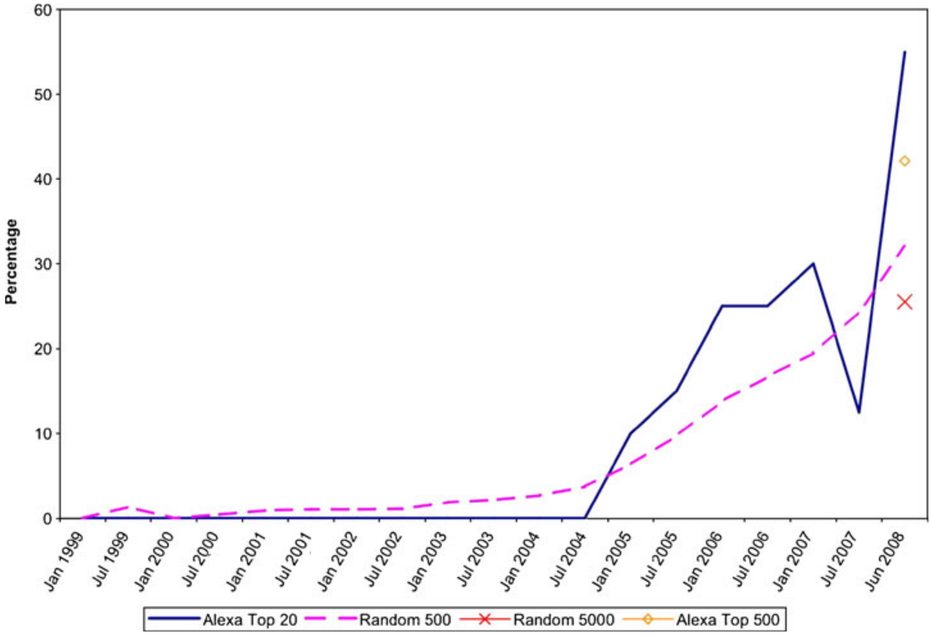

Figure 7 XHTML 1.0 (*lines* represent Archive, *circle* and *cross* represent Ground Truth)

1998 [40]. Finally, we can see CSS adoption has steadily increased from 8 to 60% over the last ten years with the top sites adopting the technology far faster to an average of 90% adoption across the premier sites.

5.2 De-Facto technology

First introduced by Microsoft Internet Explorer in 1997, standardised in HTML 4.0 Transitional, allowed in HTML 5.

De-Facto technology adoption, as characterised by script and AJAX usage, can be seen to be quite widespread over the archived and current sites. Indeed, most sites seem to have some form of scripting and adoption of the iFrame and, latterly, XMLHttpRequest seems to be on the rise. Here we include the iFrame as it was first introduced by Microsoft Internet Explorer in 1997 rapidly becoming a de-facto standard, and a W3C standard in HTML 4.0 Transitional, allowed in HTML5. JavaScript (see Figure 8) trends slightly up but over the last 10 years maintains at an average of around 80%, however there is a slight disparity between 70–75% when using the larger sample size. We see VBScript also maintains an even trend but at the other end of the scale maintaining between 1–2% (see Figure 9). The iFrame, while technically not de-facto post HTML 4.0, can be classed as non-standard when used within the dynamic/asynchronous framework. In this case we see a great trend upwards from 5 to 35% usage in just five years among the top sites with lower usage among the random 500/5000 sites, peaking at 20%. The most telling story is however that expressed by XMLHttpRequest. This technology, originating from Microsoft in the late 1990's, languished unused until AJAX was coined in 2005. Now with a more

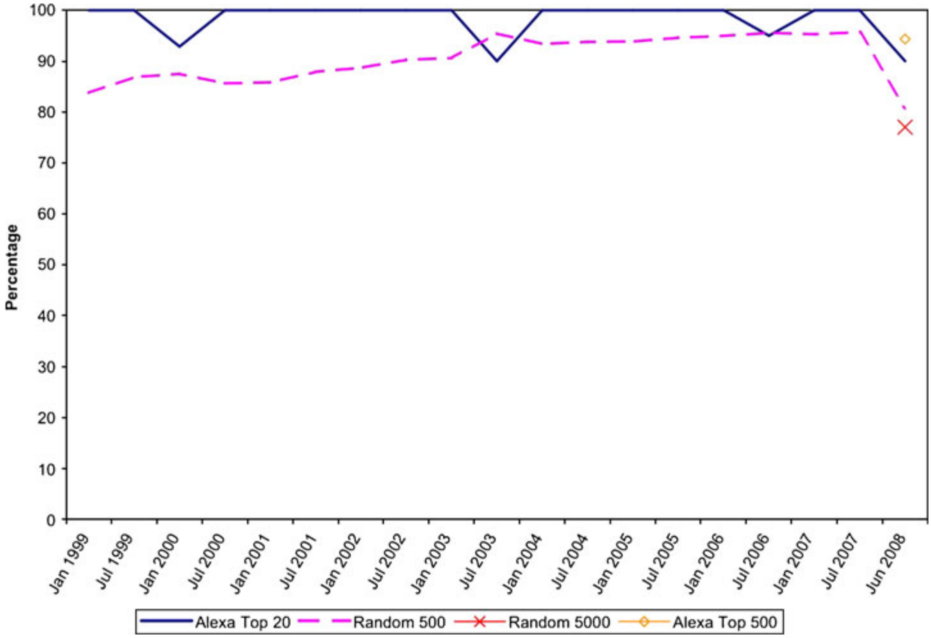

Figure 8 JavaScript (all versions—*lines* represent Archive, *circle* and *cross* represent Ground Truth)

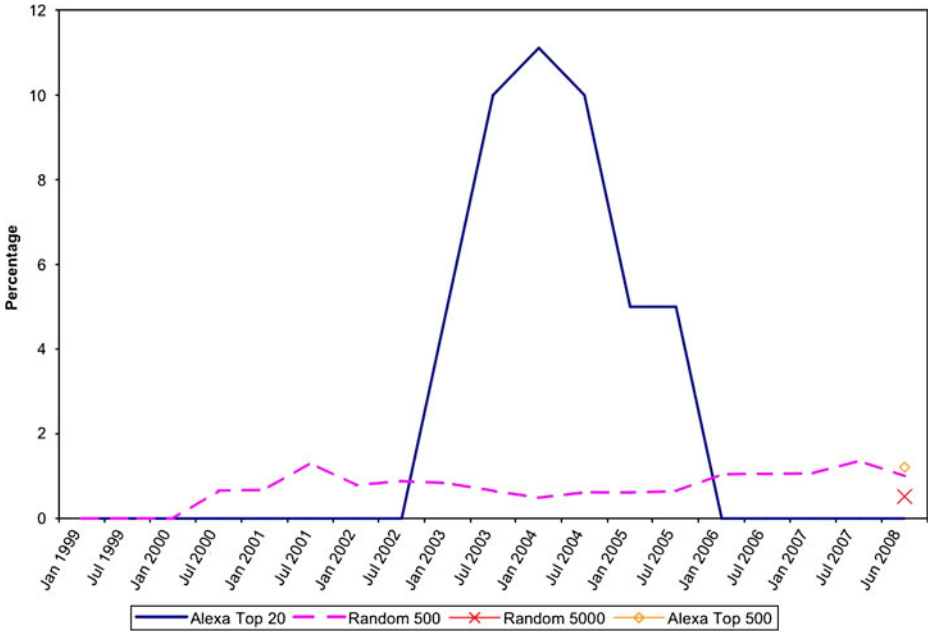

Figure 9 VBScript (all versions—*lines* represent Archive, *circle* and *cross* represent Ground Truth)

‘open’ use XMLHttpRequest has seen steady adoption to 20% in three years over the top 20/500 sites (Figures 10 and 11).

5.3 Guidelines

With regard to guidelines we can see that these are characterised by the only long standing formalised guidelines standardised by the W3C; the Web Content Accessibility Guidelines version 1.0. Here we can see that the guidelines are slowly trending upwards from around 2–10% over the ten-year period with the ground truth figure being at best 4%. Therefore, even an optimistic analysis, showing an attempt at guideline compliance, is only 10%. We say guideline compliance because we have not checked each page for compliance to the WCAG standard, but rather, whether the site developer(s) have made any kind of statement about accessibility, even if it is to explain why it has not been implemented. We want to be deliberately optimistic here, firstly because actually checking for accessibility, even with automated tools can be incredibly subjective (eg. an ‘alt’ attribute for an image may be present and filled with gibberish, this would pass an automated accessibility test but would not be accessible), and secondly, because we wanted to gauge if developers had actually thought about accessibility enough and in such a comprehensive way as to actually mention it on their home pages. Therefore, this data is about pages that make some statement about accessibility, whether or not they actually implement a truly accessible policy is another matter entirely.

However as we have previously mentioned ‘using’ does not represent ‘conforming’. In this case we decided to check accessibility conformance using of our 2008

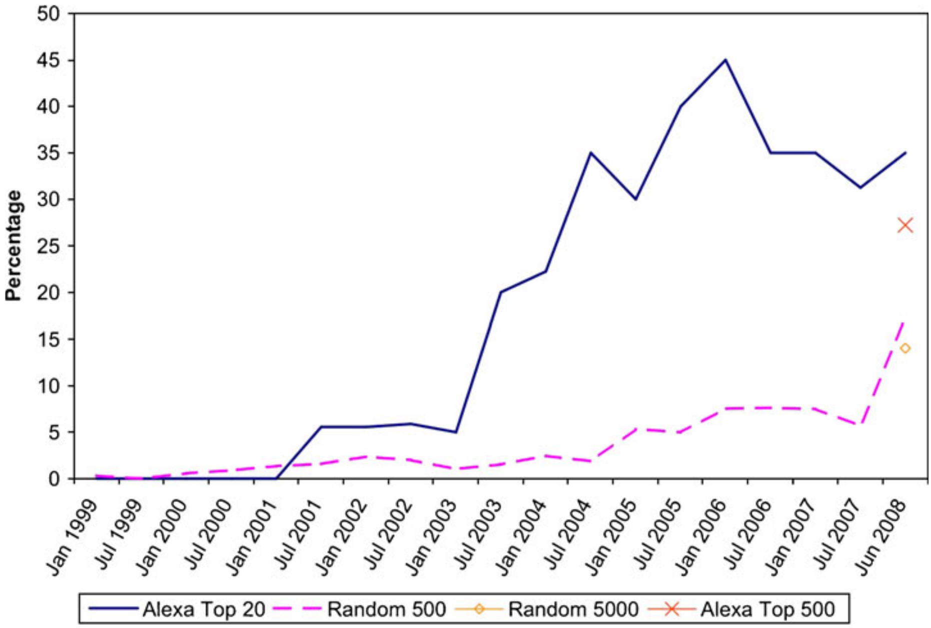

Figure 10 AJAX/iFrame (*lines* represent Archive, *circle* and *cross* represent Ground Truth)

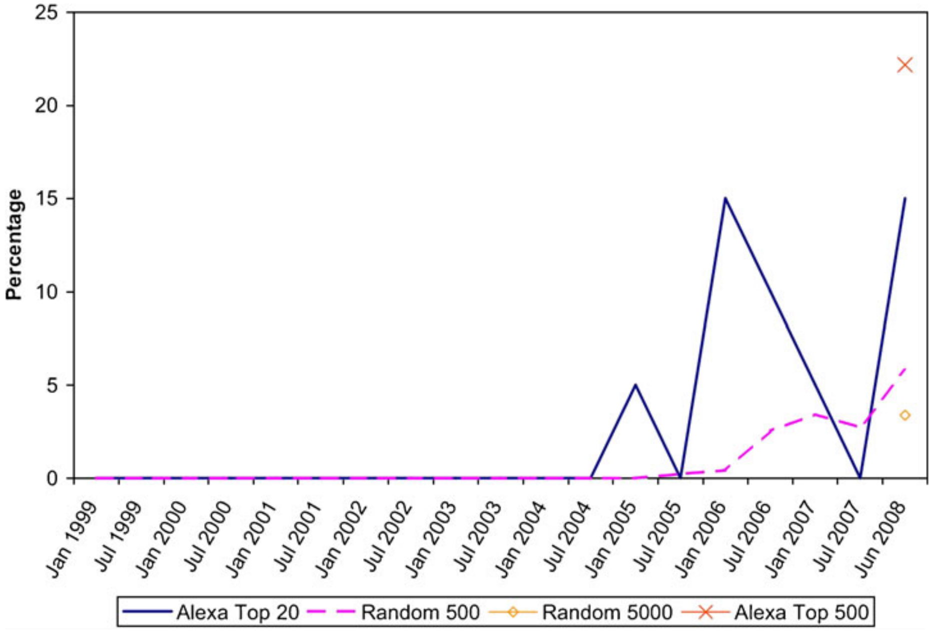

Figure 11 AJAX/XMLHttpRequest (*lines* represent Archive, *circle* and *cross* represent Ground Truth)

Alexa Top 20 using AChecker, EvalAccess, and WAVE, and then manually check each page when we found conformance warnings (see Table 1). As we can see the picture is incredibly confused with no two checkers accurately agreeing on the number of known problems, errors, or warnings. In this case we decided that further evaluation would be required to fully determine if there where errors present. Therefore we decided that the presence of at least one zero score on any row prompted a manual evaluation to uncover errors which could not automatically be evaluated; agreement between all checkers of at least one validation error meant a manual evaluation was not required.

In this case we can see that there is no one Website in the Alexa top 20 that does not show at least an agreement that an error is present between all three checkers, or that the presence of a zero error along with a warning, resolves to an instance of an actual error when manually examined. The Website with the most conformance to accessibility standards is the 'bbc.co.uk' site showing just one validation error, when manually checked. This being ten instances of checkpoint 6.3; described as: "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 [Priority 1]". However, on closer examination, we found that the BBC's proprietary Betsie [33] system will handle this inaccessibility thereby enabling the organisation to claim accessibility conformance to WCAG 1.0 priority 1. Interestingly, this site is the only one in our top twenty which discussed its accessibility conformance.

We can now speculate that at a minimum Web Accessibility Evaluation checkers are wildly variable in their ability to identify errors, and that in most cases once a

Table 1 WCAG Conformance | Alexa Top 20 | June 2008 | AChecker figures are described in terms of 'Known Problems', WAVE in terms of 'Errors', and EvalAccess in terms of 'Errors/Warnings'

Site	AChecker		WAVE	EvalAccess			Manual
	WCAG 1.0	WCAG 2.0	WCAG 1.0	WCAG 1.0 (1)	WCAG 1.0 (2)	WCAG 1.0 (3)	WCAG 1.0 (1)
yahoo.com	31	32	1	0/68	83/131	27/134	2 – CP 5.1, 5.2
google.com	21	11	0	0/22	24/54	2/69	2 – CP 5.1, 6.3
live.com	35	23	0	0/28	33/52	2/63	2 – CP 5.1, 6.3(3)
msn.com	23	3	1	0/152	8/292	1/358	3 – CP 2.1, 6.3(12), 8.1
myspace.com	65	36	17	6/245	36/245	1/192	N/A
facebook.com	26	19	10	0/83	38/105	1/269	3 – CP 2.1, 5.1, 6.3(10)
microsoft.com	159	44	2	0/274	100/249	0/241	2 – CP 1.2, 2.1
ebay.com	34	14	12	10/168	31/210	14/301	N/A
aol.com	110	55	8	0/340	75/427	0/509	2 – CP 2.1, 6.3(45)
mail.ru	84	148	86	3/540	202/537	87/588	N/A
sina.com.cn	Error	Error	Error	13/1181	2205/3276	7/4817	N/A
amazon.com	44	30	7	9/453	61/578	33/652	N/A
go.com	68	64	43	31/203	69/168	18/232	N/A
craigslist.org	186	14	0	0/9	3/208	3/397	2 – WCAG 1.0 CP 5.1, 5.2
bbc.co.uk	80	10	4	0/155	2/710	8/700	1 – CP 6.3(10)
uol.com.br	122	102	70	52/392	97/482	1/600	N/A
espn.go.com	73	46	12	6/397	58/623	1/783	N/A
rakuten.co.jp	331	373	8	1/1230	836/1268	210/747	N/A
rambler.ru	67	155	33	19/343	147/452	57/606	N/A
naver.com	194	20	Error	0/439	173/571	1/666	3 – CP 5.1, 5.2, 6.3(27)

Numbers in brackets '(x)' denote the number of instances of a particular error. The presence of at least one zero score on any row prompted a manual evaluation to uncover errors which could not automatically be evaluated—'CP' denoting the CheckPoint which failed. Agreement between all checkers of at least one validation error meant a manual evaluation was not required—signified by 'N/A'. The word 'Error' within a table element denotes that the checker suffered a non-recoverable systems failure during the evaluation of that particular page—we assume this is because there where more errors than could be handled by the software warning has prompted a manual evaluation for errors, at least one error is found to exist. Continuing our thread of optimism however, we may still suggest that when evaluated only 5–10% of Websites would pass both an automatic and deeper manual check for WCAG 1.0 priority 1 accessibility. This finding falls roughly within the boundaries of our automated text processing results and seems to suggest that, in the vast majority of cases, Websites which are accessible state that they are somewhere on the page, those Webpages that are not accessible cannot make such a, possibly libellous, claim and so do not.

6 How has the Web evolved?

To recap, our intention when undertaking this work is to understand the way that the Web has evolved over the last ten years in an attempt to understand Web accessibility research directions. In this case, we do not expect to directly contribute to the Web Evolution effort for its own sake, but rather, need to understand the evolution of infrastructure, guidelines, and recommendations to support our accessibility work. By understanding the way the Web has evolved we can attempt to predict how it will evolve, and as such guide future efforts in supporting Web Accessibility, and understanding human behaviour in the context of Web interaction (Figure 12).

6.1 Technology life-span

How long will a technology we around for once deployed in the Web? From our results we can see that even core technologies long since out dated (such as HTML 2) still have a presence on the Web which far surpasses that which we would expect.

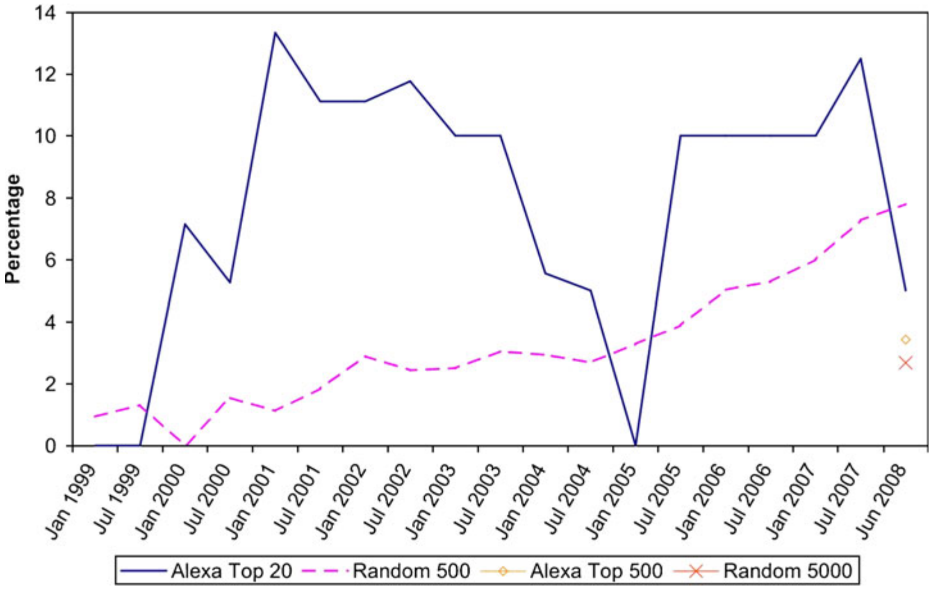

Figure 12 Web content accessibility statements

HTML 2 still has between a 5 and 15% deployment over ten years after its successor was released. Indeed, it also seems that even when core technology versions are not adopted on a large scale such as HTML 3 they still represent a 5% deployment even if technical enhancements are created. This may be because the pages reside in the so-called long-tail, whereby few changes are made to Web-sites which are running without problems. It may also be because new technology would require an infrastructure upgrade at the server side, or for business reasons of rebuilding large systems with new Web interfaces.

We would suggest that there is a cost to change and while systems are running efficiently it may be sub-optimal to change them. This is especially the case when in an ‘open standards’ environment where indecision as to the next version of the standard may cause changes to not be made immediately; especially in rapidly changing environments such as the Web. We can see that this level of life-span is also the case with de-facto standards such as scripting languages. Even though JavaScript is dominant over the Web, VBScript still maintains a 1–2% deployment especially over sites not in Alexa’s top 500. We can see that if user agents were to stop supporting HTML 2 and 3 (now long out of date) approximately 15% of the Web would be adequately rendered. This has serious implications for Web Standards Organisations and technology developers alike.

Interpretation

– Rapidly changing, incremental updates to core technology standards hinder the uptake of those core technology specifications (*inferred from the interplay of HTML 4/4.01/XHTML 1.0*);
– Duplicate or highly overlapping specifications[17] hinder the uptake of both specifications, developers preferring to wait for a single coherent outcome (*inferred from HTML adoption ‘skipping’ version 3*); and
– Now out-dated technical standards still represent around 15% of the Web (*inferred from the continued presence of HTML 2*).

6.2 Adoption times

Looking at our results in more detail we can see that adoption times vary across technologies and across core and de-facto technology specifications. This can be best seen by a study of HTML 4.x, XHTML 1.0, and CSS adoption. HTML 4 adoption took a very short amount of time to reach a 70% adoption rate (approximately two years) however we can see that there maybe issues that account for this. Firstly, there was a strong impetus to adopt early due to the failure of HTML 3 to address all developers’ issues. In addition, a number of previously de-facto ‘Netscape’ standards were introduced in to HTML 4. Finally, it is likely that HTML 2 was used the first time a new site was created (back in the mid 90’s) while HTML 4, most likely, represented an incremental release for a developer with sound technical reasons for making the adjustments.

The relationship between HTML 4.x and XHTML 1.0 is more telling. In this case we see adoption times of three years to reach a 10% deployment moving upwards

[17] As it seems the W3C are currently in the process of doing with XHTML 2 and HTML 5.

to five years for a 30% deployment. While adoption times for the top sites are higher and faster the difference between them is not as marked as may be previously imagined; indeed, there is only a 10% different between the top 500/random 5000 of the ground truthed June 2008 snapshot. CSS (level 1 released 17 Dec 1996 level 2.1 is in candidate recommendation since 19 July 2007) is also useful for understanding adoption times. Here we can see a 15% adoption after three years increasing to 55% after six years. In this case we can see some trend level correlation between XHTML 1.0 and CSS 1. We believe the faster adoption rates for CSS level 1 is due to this being the first release of the new technology, where as XHTML 1.0 was an incremental release and developers may have waited longer before moving from 4.01 (XHTML 1.0 is the XML compliant version of HTML 4.01).

Finally, let us turn to AJAX introduction rates as a gauge of de-facto standards differences. Here we can see a large variance between top site early adopters and the rest of the Web. This is the case because the technology is more difficult to implement than HTML and because the concept of Web 2.0 and AJAX takes longer to percolate down to the general Web, in addition, the technology is very specific and is not suited to all types of Web-sites and tasks being performed on them. Notice how even though XMLHttpRequest was invented in the late 1990s by Microsoft it did not enjoy widespread adoption until there was a compelling technical argument for its use, in this case the adoption of AJAX at the start of 2005. Even in this case adoption rates are still at around 8% after three years. This is similar to the more easily implemented iFrame (released as part of the HTML 4.01 specification in December 1999). Again, uptake is limited until first Google starts advocacy for it's mapping and calendaring offerings and then as result of AJAX.

Interpretation

- Technology adoption occurs when developers see that there is a good reason to make a technical change (*inferred from HTML 4.0 CSS 1.0 adoption rates*);
- Large differences do not exist between adoption of core and de-facto technology, although core technology has a compelling technical reason for adoption built in, technology which become de-facto standards have to make this technical case with every new release (*inferred from XHTML and AJAX adoption rates*); and
- As a 'rule-of-thumb' new technology moves at about 15% within the first three years, incremental releases move at about 10% within the first three years. However, sites which are most popular often exhibit enhanced adoption rates of between 10 and 15% over the same period (*inferred from the interplay of different (X)HTML versions, CSS, and AJAX, over the top 20 sites and the random 500 sites*).

6.3 Core/de-facto differences

De-facto standards do not differ much in their adoption rates than do core standards, however, there do seem to be some caveats to this assertion. Firstly, without a very strong use case new technology will never become a de-facto standard. This can been in the graph for both iFrames and XMLHttpRequest, in which the technology appeared first but was not used until other drivers were introduced to the Web use scenario. Secondly, we see a much steeper adoption trend, for technologies which become de-facto, in the early adopter sites than for core technology. This suggests that ‘cool’ technology is adopted very rapidly, and is mostly de-facto, as opposed to slow incremental core technology which often begins to incorporate successful technology that has become de-facto. Technologies which peak also do not necessarily become de-facto as in the case of VBScript which is not an open standard and is only natively supported by Internet Explorer. Compared with JavaScript we can see that this open standard (even though it has many versions) maintains hegemony over scripting languages because it is built as standard into most desktop browsers.

Interpretation

- Open technologies have a better chance of moving to de-facto technologies and standards because user agent manufacturers then have the ability to build the technology directly into the browser (*inferred from JavaScript and VBScript differences*);
- If there is only one technology, which represents a significant technical advance (such as XMLHttpRequest), the Web may make it a de-facto standard even if it is a closed standard (*inferred from AJAX adoption rates*); and
- If a technology is to become de-facto, it needs a stronger use case than does a core technology or standard (*inferred from a comparison of iFrame and XMLHttpRequest adoption rates*).

6.4 Guidelines/standards differences

There is a large difference between adoption rates found at the standards/technology level and those at the guidelines level. Here we see adoption rates over ten years (WCAG 1.0 became a recommendation in May 1999) of between 6–10% for the general Web and of no more than 14% for the early adopters; this supports other work in the area [17, 38]. While these figures may be to do with developers not mentioning accessibility compliance or policy on their sites it seems unlikely that this is the general case if the site actually does meet accessibility guidelines. Indeed, Web developers often cite barriers such as lack of time, lack of training, lack of managerial support, lack of client support, inadequate software tools, and confusing accessibility guidelines as a rational for non-accessibility compliance [28]. It seems far more likely that guidelines are not adopted at the same rate as standards, echoing our outcomes: ‘Technology adoption occurs when developers see that there is a good reason to make a technical change’ and underscoring the need for a strong use case. In both cases ‘mainstream’ development does not see accessibility as providing either a technical or fiscal advantage; even though that perception is known to be incorrect.[18] It will be interesting to test this with an analysis of ‘MobileOK’ [39, 49, 50] after the guidelines have been adopted and allowed to evolve for three years. In any case we can see that guideline adoption can only be achieved under the circumstances stated above if there becomes a good technical reason for their adoption, or if the developers are required to follow them by law, as is becoming the case with government and educational Web-sites.

[18] http://www.microsoft.com/enable/microsoft/takemura.aspx, http://www.Webcredible.co.uk/user-friendly-resources/Web-accessibility/benefits-of-accessible-Websites-1.shtml, http://www.w3.org/WAI/bcase

Interpretation

- Accessibility Guideline adoption has stagnated at less then 10% over ten years (*inferred from guideline conformance statements*);
- There must be a rationale, either fiscal, technical, or legal, for developers to adopt and implement guidelines (*inferred from the disparity between technical recommendations and accessibility guidelines*); and
- Adoption of the AJAX concept has been possible due to the technical and supposed fiscal advantages, however, a reliance on 'doing the right thing' has not as yet advanced guideline adoption to any significant level (*inferred from the disparity between AJAX and guideline adoption rates*).

7 How will the Web evolve?

In general we can see that Web Accessibility conformance is low. This may be because as a community, our Web accessibility solutions are failing to gain acceptance within the wider cadre of Web practitioners. Guidelines do not seem to be adopted by the Web development community; validation and conformance tools seem not to be widely used, authoring tools often do not routinely include significant accessibility testing suites. Dynamic content is on the rise, and Web interaction is becoming increasingly complicated as the visual look and feel becomes ever more complex. In this environment, enhanced conformance to the new Web Content Accessibility Guidelines (WCAG) version 2.0 [9] is unlikely to proceed at any greater levels of adoption than WCAG 1.0. Although WCAG 2.0 comes with a number of useful techniques for conformance testing, most of the test procedures suggested represent manual evaluation and repair approaches [1]. This move to a 'human in the loop' scenario is the correct thing to do, but from past history, adoption is low even without this extra encumbrance. It does seem clear that certainly WCAG 2.0 is a major improvement on the version one guideline set. Even though early WCAG 2.0 Drafts had a number of critical opponents the final guideline sets look significantly more understandable and usable than the old; in part due to changes made after these vocal criticisms. However, the landscape is further complicated by the rise of Web 2.0 technologies and the semantics of their access. But, we must remember that WCAG 1.0 was technology specific. As technologies changed it became very difficult to use WCAG 1.0 to make sites accessible. WCAG 2.0 is quite different and can adapt to new technologies as they evolve.

The rise in use of AJAX and XMLHttpRequest suggest that the current interaction model of the Web, as used by assistive technologies, is changing. It also suggests that this change is starting to increase at such a rate that without timely and prompt action visually disabled users will be barred from using this 'new' Web. Current access technology assumes a static interaction model, and expects that once the audio rendering has been performed no other changes to content already spoken will occur. This assumption is fundamentally flawed when dealing with AJAX sites and technologies. This is because the applications embedded through the static content are constantly and dynamically updating. Accessible Rich Internet Applications (ARIA and now WAI-ARIA) [21] defines new ways for functionality to be provided to assistive technology [15], and thereby tries to make these updates visible in 'live regions' [44], in addition suggesting how updates are handled. While there are a number of AJAX/JavaScript frameworks available, Google has built Access-Enabling AJAX (AxsJAX) [13] which is a 'library for enhancing the accessibility of Web 2.0 applications'. Simply, AxsJAX combines AJAX and WAI-ARIA at the outset and so any AJAX applications built with the library will automatically include WAI-ARIA; and will therefore be accessible [13, 31]. While ARIA may provide a programmatic solution many problems still exist centered around the likelihood of widespread ARIA adoption, retrofitting legacy AJAX sites, and user choice on the client-side [51].

Further changing the landscape, HTML5 development is on the rise. Originally, HTML5 was created from the 'WHATWG' splinter group (circa 2004) formed by the user agent manufacturers; Apple, Opera, and Mozilla. WHATWG stands for 'Web Hypertext Application Technology Working Group' and therein lays the clue, HTML5 is mostly about creating a declarative hypertext language in support of application like functionality. Indeed, directly quoting WHATWG: 'after a W3C workshop. Apple, Mozilla and Opera were becoming increasingly concerned about the W3C's direction with XHTML, lack of interest in HTML and apparent disregard for the needs of real-world authors. So, in response, these organisations set out with a mission to address these concerns and the Web Hypertext Application Technology Working Group was born.' Pressure on the W3C culminated in the formation of the 'HTML working group', chartered in March 2007, with the aim of moving WHATWG proposals into a W3C standard. HTML5 is made to be dynamic and interactive, to incorporate many of the aspects of AJAX into a holistic format focused on moving the Web to and application level construct; currently ARIA is not included in the specification.

8 Arguments for extending technical recommendations

In our opinion, this paper should be the motivation for subsuming guidelines into technical language specifications. We can see that technical conformance is indeed the fastest method for take up of new Web technologies, but optional aspects such as guidelines have a very low take up unless there becomes a technical rationale for their introduction. Lobbying, evangelising, and even legal means do not seem to have influenced guidelines adoption and so until something major changes there still seems to be little reason for technologist to adopt them. In this case, we believe that a better way of encouraging conformance is to make guidelines part of the technical specification where possible.

XHTML validity (and HTML for that matter) is mainly about conformance to the Document Type Definition (DTD) of the (X)HTML specification, and in detail, is an extensible mark-up language (XML) document that requires only the facilities described as mandatory in the specification, which are: (1) It must conform to the constraints expressed in of the approved DTDs; (2) The root element of the document must be HTML; (3) The root element of the document must contain an XMLNS declaration for the XHTML namespace; and finally (4) The public identifier included in the DOCTYPE declaration must reference one of the approved DTDs. In this case the question must be 'can we express guidelines in a technical way so as to include them as part of the technical specification?'

While minor aspects, such as the presence of alternative text in images, of Web accessibility are captured as part of the technical specifications of progressive languages such as (X)HTML, this is by no means rich enough, or complete; table summary attributes are not required. Indeed, we would suggest that a DTD is not expressive enough but that the, common, XML Schema[19] or XPath specifications may be [1, 29]. With some additional effort and even a cursory analysis of WCAG 2.0 guidelines we can see very real scope for capturing many guidelines in a technical way [3] and placing these guidelines into the technical specification, such that XHTML or HTML language adoption will require *all* specifications to be met for validation against one of the XML specifications.

It is however, worth noting that the development of Web content and the acquisition of Web tools is 'self regulatory' except for certain significant segments of the community. Specifically, there are regulations in the US and other countries that require those (employees or contractors) developing their governments' Web sites for access to data and information by the public to adhere to standards derived from the earlier version of the WCAG. Further, there are very few programs for Web application implementers either in academia or in the private (trade school and in-house training) sectors that emphasise usability, let alone accessibility, properly. Generally these issues are reserved for special or advanced courses or 'optional' topics rather than integrated throughout student training in design, development, implementation, and testing as they should be. If they are treated as special topics, then usability and accessibility will always be considered as after-thoughts or 'something that would be nice to have when the resources are available,' if at all, by the implementer community. In this case the integration of guidelines into the specification would be one way of implicitly ensuring a move from the current model of specialised or optional training into an accessibility integrated throughout all approach.

Finally, we must acknowledge that extending the technical recommendations is not a panacea, indeed features of existing technical specifications are not fully supported by developers, browsers and other user agents. If accessibility were included in the technical standards themselves it is not certain that adoption rates would be improved. Further, as each specification is created by a different group there may be different versions of individual access guidelines appearing in different technical standards making implementation confusing. In this case, inclusion of accessibility guidance within each technical specification showing how the technology-agnostic WCAG 2.0 guidelines can be implemented within that technology may be possible. Care would need to be taken that the guidance does not change WCAG 2.0 provisions by adding or subtracting requirements since this would lead to fragmentation and reverse recent harmonisation work.

9 Conclusions and future work

Web accessibility aims to help people with disabilities to perceive, understand, navigate, and interact with, as well as contribute to, the Web. There are millions of

[19] XML Schemas express shared vocabularies and allow machines to carry out rules made by people. They provide a means for defining the structure, content and semantics of XML documents.

people who have disabilities that affect their use of the Web. Although Web guidelines direct designers and authors to best practice, currently, most Web sites have accessibility barriers that make it either difficult or near impossible for many people with disabilities to use these sites. To build applications and content that allows for heterogeneity, flexibility, and device independence is incredibly difficult, incredibly challenging, and incredibly necessary. Designing and building Web documents are perhaps the most important parts of the Web accessibility cycle. Only by expecting a clean technical design can the access technology developer create approaches which fully interact with the content of the document. The W3C has recognised the importance of correctly engineering the document by recommending a separation between content as well as structure, encapsulated within (X)HTML mark-up, and its presentation, using CSS. While this separation does provide the building blocks for increasing access to Web based documents, additional guidelines, techniques, and best practice have been introduced to facilitate a document's increased accessibility.

By understanding that standards and technology adoption exhibits a three-year lead to on average attain 15% deployment while guideline adoption is only 10% over ten years, we can see that these standards would best be adopted if they are built directly in to the technical specification. In addition, we suggest that interim technology is required to bring the current interaction requirements inline with the available technology. More than this, as a three-year time lag is easily the length of time required for a new full browser version to be released and so this interim technology is no more temporary than any other piece of Web technology.

Further work is however need to answer the following questions:

- Is the technology maintained or are there peaks and troughs;
- When, and how, does one technology overcome another;
- Do validation and repair tools make a difference to the take-up of guidelines and recommendations; and
- Can we see any socio-technological aspects which effect infrastructure, recommendations, or guidelines?

Web accessibility has previously been characterised by the correction of our past mistakes to make the current Web fulfil the original Web vision of access for all. If we do not pre-empt the future by highlighting new aspects of Web development that may be useful in the accessibility domain, or may need remedial accessibility attention we are condemned to continually repeat our past mistakes.

Acknowledgements This work was undertaken as part of the SASWAT project funded by EPSRC (EP/E062954/1) and conducted in the Human Centred Web Lab part of the School of Computer Science at the University of Manchester (UK).

About: http://hcw.cs.manchester.ac.uk/research/saswat/

Reports: http://hcw-eprints.cs.man.ac.uk/view/subjects/saswat.html

References

1. Aizpurua, A., Arrue, M., Vigo, M., Abascal, J.: Transition of accessibility evaluation tools to new standards. In: W4A '09: Proceedings of the 2009 International Cross-Disciplinary Conference on Web Accessibililty (W4A), pp. 36–44. ACM, New York (2009). doi:10.1145/1535654.1535662

2. Anderson, C.: The Long Tail, revised and updated edition. Hyperion, New York (2008)
3. Arrue, M., Vigo, M., Abascal, J.: Including heterogeneous web accessibility guidelines in the development process. Lect. Notes Comput. Sci. **4940**, 620–637 (2008). doi:10.1007/978-3-540-92698-6_37
4. Asakawa, C.: What's the web like if you can't see it? In: W4A '05: Proceedings of the 2005 International Cross-Disciplinary Workshop on Web Accessibility (W4A), pp. 1–8. ACM Press, New York (2005). doi:10.1145/1061811.1061813
5. Asakawa, C., Lewis, C.: Home page reader: IBM's talking web browser. In: Closing the Gap Conference Proceedings (1998)
6. Brambring, M.: Mobility and orientation processes of the blind. In: Warren, D.H., Strelow E.R. (eds.) Electronic Spatial Sensing for the Blind, pp. 493–508. Dordrecht, Lancaster (1984)
7. Brewington, B.E., Cybenko, G.: How dynamic is the web? Comput. Networks **33**(1–6), 257–276 (2000)
8. Caldwell, B., Cooper, M., Jacobs, I., Reid, L.G., Vanderheiden, G.: Web Content Accessibility Guidelines 2.0. W3C (2008). http://www.w3.org/TR/WCAG20/
9. Caldwell, B., Cooper, M., Reid, L.G., Vanderheiden, G., White, J.: Web Content Accessibility Guidelines 2.0. WWW (2008). http://www.w3.org/TR/WCAG20/
10. Chen, A.Q., Harper, S.: Web Evolution—Code and Experimental Guide. Technical report, University of Manchester (2008). URL http://hcw-eprints.cs.man.ac.uk/75/
11. Chen, A.Q., Harper, S.: Web Evolution—Method and Materials. Technical report, University of Manchester (2008). URL http://hcw-eprints.cs.man.ac.uk/74/
12. Chen, C.: Structuring and visualising the www by generalised similarity analysis. In: Proceedings of the 8th ACM Conference on Hypertext and Hypermedia, pp. 177–186. ACM Press, New York (1997)
13. Chen, C.L., Raman, T.V.: Axsjax: a talking translation bot using google im: bringing web-2.0 applications to life. In: W4A '08: Proceedings of the 2008 International Cross-Disciplinary Conference on Web Accessibility (W4A), pp. 54–56. ACM, New York (2008). doi:10.1145/1368044.1368056
14. Cho, J., Garcia-Molina, H.: The evolution of the Web and implications for an incremental crawler. In: VLDB '00: Proceedings of the 26th International Conference on Very Large Data Bases, pp. 200–209. Morgan Kaufmann, San Francisco (2000)
15. Cooper, M.: Accessibility of emerging rich Web technologies: Web 2.0 and the semantic Web. In: W4A '07: Proceedings of the 2007 International Cross-Disciplinary Conference on Web Accessibility (W4A), pp. 93–98. ACM, New York (2007). doi:10.1145/1243441.1243463
16. Fetterly, D., Manasse, M., Najork, M., Wiener, J.L.: A large-scale study of the evolution of Web pages. Software: Practice and Experience **34**(2), 213–237 (2004). doi:10.1002/spe.577
17. Disability Rights Commission: The Web: Access and Inclusion for Disabled People. Tech. rep., Disability Rights Commission (DRC), UK (2004)
18. Douglis, F., Feldmann, A., Krishnamurthy, B., Mogul, J.: Rate of change and other metrics: a live study of the world wide web. In: USITS'97: Proceedings of the USENIX Symposium on Internet Technologies and Systems on USENIX Symposium on Internet Technologies and Systems, p. 14. USENIX Association, Berkeley (1997)
19. Duffy, B.E., Turow, J.: Key Readings in Media Today: Mass Communication in Contexts. Routledge, New York (2009)
20. Furuta, R.: Hypertext paths and the WWW: experiences with Walden's paths. In: Proceedings of the 8th ACM Conference on Hypertext and Hypermedia. ACM, New York (1997)
21. Gibson, B.: Enabling an accessible Web 2.0. In: W4A '07: Proceedings of the 2007 International Cross-Disciplinary Conference on Web Accessibility (W4A), pp. 1–6. ACM, New York (2007). doi:10.1145/1243441.1243442
22. Gulli, A., Signorini, A.: The indexable Web is more than 11.5 billion pages. In: WWW '05: Special Interest Tracks and Posters of the 14th International Conference on World Wide Web, pp. 902–903. ACM, New York (2005)
23. Gunderson, J., Jacobs, I.: User Agent Accessibility Guidelines 1.0. World Wide Web Consortium (W3C) (1999). http://www.w3.org/TR/WAI-USERAGENT/
24. Hackett, S., Parmanto, B., Zeng, X.: A retrospective look at website accessibility over time. Behav. Inf. Technol. **24**(6), 407–417 (2005)
25. Harper, S., Goble, C., Stevens, R.: Traversing the Web: mobility heuristics for visually impaired surfers. In: Catarci, T., Mercella, M., Mylopoulos, J., Orlowska, M.E. (eds.) Proceedings of the Fourth International Conference on Web Information Systems Engineering (WISE'03), pp. 200–209. IEEE Computer Society, Los Alamitos (2003)

26. Ivory, M., Hearst, M.: The state of the art in automating usability evaluation of user interfaces. ACM Comput. Surv. **33**(4), 470–516 (2001)
27. Ivory, M.Y., Megraw, R.: Evolution of Web site design patterns. ACM Trans. Inf. Syst. **23**(4), 463–497 (2005). doi:10.1145/1095872.1095876
28. Lazar, J., Dudley-Sponaugle, A., Greenidge, K.D.: Improving Web accessibility: a study of webmaster perceptions. Comput. Hum. Behav. **20**(2), 269–288 (2004). doi:10.1016/j.chb.2003.10.018. URL http://www.sciencedirect.com/science/article/B6VDC-4BRKMR8-B/2/08cd39063d4902227cf6033cf824aca4 . The Compass of Human-Computer Interaction
29. Leporini, B., Paternò, F., Scorcia, A.: Flexible tool support for accessibility evaluation. Interact. Comput. **18**(5), 869–890 (2006). doi:10.1016/j.intcom.2006.03.001
30. Lowe, D., Hall, W.: Hypermedia and the Web: An Engineering Approach. Wiley, New York (1998)
31. Lunn, D., Harper, S., Bechhofer, S.: Combining sadie and axsjax to improve the accessibility of Web content. In: W4A '09: Proceedings of the 2009 International Cross-Disciplinary Conference on Web Accessibililty (W4A), pp. 75–78. ACM, New York (2009). doi:10.1145/1535654.1535672
32. Mckeever, S.: Understanding web content management systems: evolution, lifecycle and market. Ind. Manage. Data Syst. **103**(9), 686–692 (2003)
33. Myers, W.: BETSIE: BBC Education Text to Speech Internet Enhancer. British Broadcasting Corporation (BBC) Education (2007). http://www.bbc.co.uk/education/betsie/
34. Nielsen, J.: Usability Engineering. Morgan Kaufmann, San Mateo (1994)
35. Ntoulas, A., Cho, J., Olston, C.: What's new on the Web?: the evolution of the Web from a search engine perspective. In: WWW '04: Proceedings of the 13th International Conference on World Wide Web, pp. 1–12. ACM, New York (2004). doi:10.1145/988672.988674
36. O'Neill, E.T., Lavoie, B.F., Bennett, R.: Trends in the evolution of the public Web (1998–2002). D-Lib Magazine **9**(4) (2003). Retrieved 11 October 2006 from http://www.dlib.org/dlib/april03/lavoie/04lavoie.html
37. Paciello, M.: Web Accessibility for People with Disabilities. CMP Books, CMP Media LLC (2000)
38. Petrie, H., Hamilton, F., King, N.: Tension, what tension?: website accessibility and visual design. In: W4A '04: Proceedings of the 2004 International Cross-Disciplinary Workshop on Web Accessibility (W4A), pp. 13–18. ACM, New York (2004). doi:10.1145/990657.990660
39. Rabin, J., McCathieNevile, C.: Mobile Web Best Practices 1.0 (2005). http://www.w3.org/TR/mobile-bp/
40. Raggett, D., Boumphrey, F., Altheim, M., Wugofski, T.: Reformulating HTML in XML (W3C Working Draft 5th December 1998). WWW (1998)
41. Raskin, J.: Looking for a humane interface: will computers ever become easy to use? Commun. ACM **40**(2), 98–101 (1997). doi:10.1145/253671.253737
42. Takagi, H., Asakawa, C., Fukuda, K., Maeda, J.: Accessibility designer: visualizing usability for the blind. In: ASSETS'04 pp. 177–184 (2004)
43. Thatcher, J., Burks, M., Heilmann, C., Henry, S., Kirkpatrick, A., Lawson, B., Regan, B., Rutter, R., Urban, M., Waddell, C.: Web Accessibility, Web Standards and Regulatory Compliance. Springer, Berlin (2006)
44. Thiessen, P., Chen, C.: Ajax live regions: chat as a case example. In: W4A '07: Proceedings of the 2007 International Cross-Disciplinary Conference on Web Accessibility (W4A), pp. 7–14. ACM, New York (2007). doi:10.1145/1243441.1243450
45. Toyoda, M., Kitsuregawa, M.: Extracting evolution of web communities from a series of web archives. In: HYPERTEXT '03: Proceedings of the Fourteenth ACM Conference on'Hypertext and Hypermedia, pp. 28–37. ACM, New York (2003)
46. Treviranus, J., McCathieNevile, C., Jacobs, I., Richards, J.: Authoring tool accessibility guidelines 1.0. In: World Wide Web Consortium (W3C) (2000). http://www.w3.org/TR/ATAG10/
47. Treviranus, J., McCathieNevile, C., Jacobs, I., Richards, J.: Authoring Tool Accessibility Guidelines 1.0. WWW (2000)
48. Velasco, C.A., Verelst, T.: Raising awareness among designers accessibility issues. In: SIGCAPH Comput. Phys. Handicap., pp. 8–13 (2001). doi:10.1145/569320.569323
49. W3C-MWBP: W3C mobileOK Checker. W3C (2008)
50. W3C-MWBP: W3C mobileOK Scheme 1.0. W3C (2008)
51. Zucker, D.F.: What does ajax mean for you? Interactions **14**(5), 10–12 (2007). doi:10.1145/1288515.1288523