Chapter 26
Tools for Web Accessibility Evaluation

Julio Abascal, Myriam Arrue and Xabier Valencia

Abstract The objective of Web accessibility evaluation is to verify that all users are able to use the Web, this means that they can perceive, understand, navigate, and interact with it (Henry 2018a). Since the manual verification of the fulfilment of guidelines that specify accessibility requirements can often turn out to be difficult and cumbersome, it is crucial to have appropriate computer tools available to assist this activity. There exist numerous applications that perform diverse types of automatic accessibility evaluations. On the other hand, on-site and remote evaluations with users can also be supported by specific tools. Even manual evaluations may be supported by crowdsourcing-based tools. All these innovations may have crucial importance in the advancement of Web accessibility. This chapter studies the need for tools in this field, reviews the main characteristics of the tools used for Web accessibility evaluation, and reflects upon their future.

26.1 Introduction

The Web was designed to be universally accessible, which means that it can be accessed by everyone, whatever their hardware, software, language, culture, location, or physical or mental ability (Henry and McGee 2018). Nevertheless, in practice this does not always happen, mainly because websites are often designed without considering human diversity, leading to poorly designed websites containing accessibility barriers.

Research on Web accessibility ranges from web content evaluation and the design and development of tools (to support both web developers when creating accessible content and users navigating the Web), to increase the understanding of the behaviour of users on the Web. These activities include the definition of accessibility guidelines; the design of evaluation methods and assessment metrics; the development of author-

J. Abascal (✉) · M. Arrue · X. Valencia
Egokituz Laboratory of Human-Computer Interaction for Special Needs, Informatics School, University of the Basque Country/Euskal Herriko Unibertsitatea, Donostia, Spain
e-mail: julio.abascal@ehu.eus

Y. Yesilada and S. Harper (eds.), *Web Accessibility*, Human–Computer Interaction Series, https://doi.org/10.1007/978-1-4471-7440-0_26

ing and evaluation tools; the development of appropriate assistive technologies; and the study of the needs and behaviour of users when they navigate the Web.

The World Wide Web Consortium (W3C) through the Web Accessibility Initiative (WAI) is one of the main organisations promoting Web accessibility. The WAI promotes the creation of international standards that describe guidelines for the different components involved in accessible web development. One of the most widely known set of guidelines is the Web Content Accessibility Guidelines (WCAG) (Chisholm et al. 1999; Caldwell et al. 2008; Kirkpatrick et al. 2018). However, other organisations and companies have also issued accessibility guidelines: the US Government Section 508 (2018), IBM Web Accessibility Checklist (IBM 2017) or the BBC (2018) have developed their own, inspired in many cases by the W3C guidelines [Read Part 3 'Standards, Guidelines and Trends' for further details].

In order to check whether accessibility guidelines have been properly implemented into websites, it is of utmost importance to conduct accessibility evaluations not only after the website has been made available online, but also during the development process.

The availability of tools for the automatic evaluation of websites appears to be key to the advancement of web accessibility. Three types of software tools are relevant to this chapter:

- Automatic evaluation tools are applications that analyse web page code to verify the compliance of specific sets of guidelines. Even if they are limited to the evaluation of criteria that can be matched in the code, they provide the main starting point to accessible design.
- Crowdsourcing-based social accessibility contributes to methods and tools (which add accessibility metadata) to enable adequate rendering of the content. It is also a way of recruiting people for manual Web accessibility evaluation.
- Tools to support remote user evaluations achieve trustworthy accessibility evaluations by managing data from real users accessing the Web in real work environments.

26.2 Web Content Accessibility Evaluation Methods

Web accessibility evaluation can be defined as the assessment of how well a website can be used by users with disabilities (Harper and Yesilada 2008) and it is an essential process in order to check that the adopted accessibility standards have been met. Web accessibility evaluation encompasses a wide range of disciplines and skills (Abou-Zahra 2008). It can require knowledge not only about technical aspects related to web technologies, guidelines, standards and evaluation tools, but also non-technical aspects such as the involvement of end users in the evaluation process. For this reason, diverse methods are used to assess the accessibility of websites, which are usually clustered in three main categories: automated testing, manual inspection and user testing. Figure 26.1 shows a detailed taxonomy of Accessibility

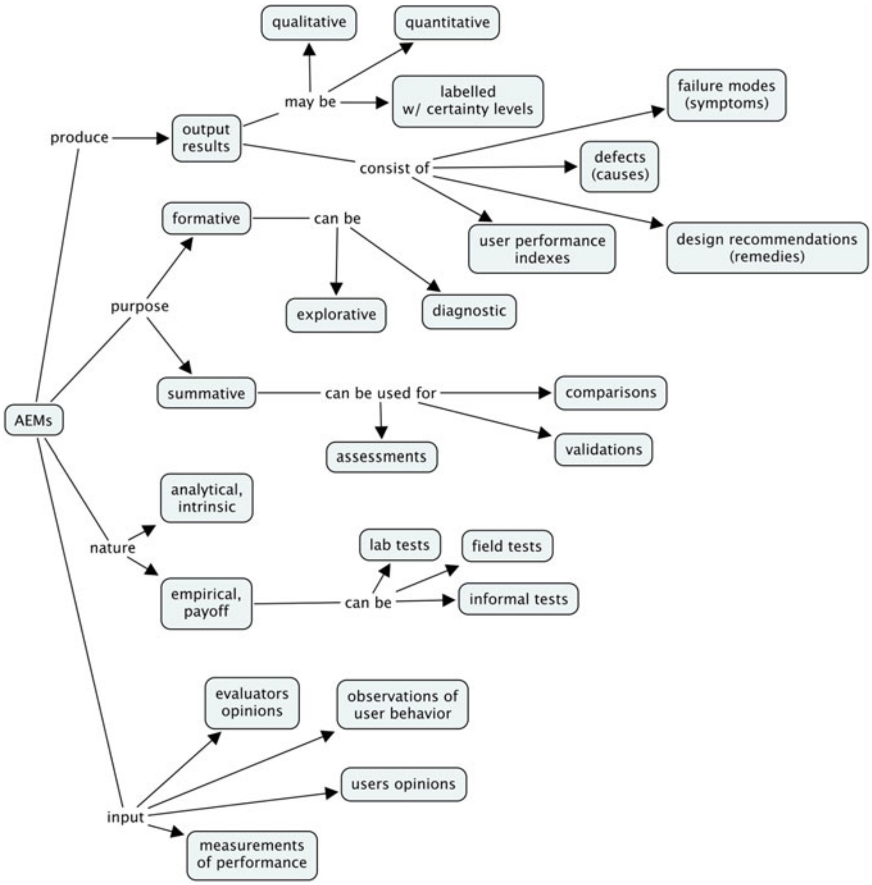

Fig. 26.1 Taxonomy of accessibility evaluation methods (Brajnik 2008b)

Evaluation Methods (AEMs), classified by purpose, type of results and type of input information used (Brajnik 2008b).

Automated Testing This kind of evaluation is performed by applications running locally or online. They analyse the page code to verify the compliance of specific accessibility guidelines. In fact, only guidelines that can be translated to code pieces can be verified by these tools. For this reason, automated testing is normally used as a first accessibility test that allows basic accessibility barriers to be detected (and sometimes fixed). In this way, they save time and resources and provide basic support to developers building and maintaining accessible websites.

Manual Inspection Accessibility inspections are conducted by expert human evaluators (Brajnik 2008a). Generally, they entail checking if a page satisfies a list of accessibility criteria. Most frequently a conformance or guidelines review is performed. This consists of checking if a website meets a set of accessibility guidelines.

For example, Barrier Walkthrough (Brajnik 2009) is a manual inspection method to evaluate Web accessibility in which the inspector identifies the frequency and severity of a list of accessibility barriers present in a website. Brajnik defines a barrier as any condition that makes it difficult for people to achieve a goal when using the website through specified assistive technology (Brajnik 2008a). The manual inspection also includes screening techniques that involve using a website in such a way that certain sensory, motor or cognitive capabilities are artificially reduced (Henry 2007). It is convenient to use manual inspection techniques throughout the design process as they can early identify potential design problems.

User Testing This kind of evaluation is usually based on informal empirical usability tests, where participants with disabilities are asked to individually perform a number of tasks. Depending on the design of the test, participants provide their feedback in various ways (e.g. concurrent or retrospective think-aloud protocols, questionnaires or interviews), and the behaviour and interactions of participants are recorded and observed by evaluators, who can then synthesise their findings. The assessment obtained from the actual experience of the users is the most reliable accessibility evaluation method [Refer to Part 2 'End User Evaluations' for more information]. Nevertheless, user testing is not easy to perform, and frequently it turns out to be slow and expensive. Support software for user testing is being produced in order to make these evaluations easier and to speed them up.

26.2.1 Methodologies for Accessibility Evaluation

Diverse methodologies for accessibility evaluation have been proposed. They vary in objective, scope, thoroughness, breadth, etc. For instance, the Website Accessibility Conformance Evaluation Methodology (WCAG-EM) 1.0 (Velleman and Abou-Zahra 2014), the TECED Accessibility Evaluation Methodology (TECED 2017) or the Accessibility Compliance and Remediation Methodology (ACRM) by AccessIT (AccessIT 2017). Many of these include an initial automatic evaluation using various (typically three) automated tools. The reason being that automatic tools can produce different evaluation results for the same page because they use different code specifications for the guidelines, and their searching/matching methods also differ. Most methodologies include manual evaluations by one or more experts. Finally, the evaluation by actual users performing real tasks is required. Nevertheless, it has not yet been fully established how these methods should be combined and organised in order to evaluate Web accessibility in a comprehensive manner (Yesilada et al. 2012).

The WAI's website contains a list of accessibility evaluation resources (Henry 2018b) and indicates different approaches for evaluating websites for accessibility. While it does not provide details on testing techniques, it contains information about general procedures and tips for accessibility evaluation in different situations, such as preliminary checks, Conformance Evaluation Methodology (WCAG-EM) (Velleman and Abou-Zahra 2014), involving users in the evaluation or the selection of automated tools, etc.

26.3 Automated Web Accessibility Evaluation Tools

The automation of the accessibility evaluation process is of particular interest in the field of accessibility evaluation research. Even if automated tools cannot replace human evaluation, they play an important role in accessibility evaluation despite their limitations. When used appropriately, they can provide developers and evaluators with support and significantly reduce the time and effort required to conduct an evaluation. Ivory and Hearst (2001) highlighted some of the advantages of using automated tools:

- The evaluation process becomes less time consuming with the consequent reduction in costs.
- The errors detected are more consistent.
- It is possible to foresee the effort needed in the evaluation process, in terms of time and financial cost.
- The evaluation scope is broadened, as it is possible to analyse diverse aspects of the interface in a shorter period of time.
- It becomes easier for inexperienced evaluators to perform usability and accessibility evaluations.
- Comparison of the suitability of different user interface design alternatives is made easier.
- It is easier to incorporate evaluation tasks during the development process.

Although useful, these tools have also some weaknesses that we must be aware of before using them. As a number of accessibility guidelines require human judgement to assess whether or not they are being met, automated evaluation tools are not able to deal with those guidelines. The diverse implementations of the search/match algorithm and the different codification of the guidelines can result in the production of false negatives (accessibility barriers that are not detected) or false positives (reported false issues). In addition, the effectiveness of the automated tools may vary depending on the number of tests implemented and how they are applied. In fact, the completeness, correctness and specificity of the results produced by a tool are used in order to measure its effectiveness (Brajnik 2004).

According to Vanderdonckt (1999), the development process of high-quality accessibility evaluation tools consists of diverse milestones. In this process, the accessibility guidelines are the essential component as they are required to:

- Collect, gather, merge and compile guidelines from available resources.
- Sort and organise the guidelines within a framework.
- Give a computational representation to the guidelines for manipulation by computer-based tools.

The effectiveness of a tool depends on the computational representation given to the guidelines. The definition of guidelines is made in natural language and may vary in its content and level of specification. The guidelines may have a different format and may be defined to a different level of detail (Abascal and Nicolle 2001; Mariage et al. 2005). The correctness of an automated tool depends on the ability of the development team to give correct computational representations to the guidelines. While several accessibility guidelines can be completely verified by computer-based tools, a number of guidelines cannot be accurately represented in a computational way because they require human judgement. For the latter, the tool can only produce recommendations for manual verification.

Therefore, most evaluation tools produce two types of results:

– Errors: when accessibility barriers have been detected in the code.
– Warnings: when specific characteristics found in the code may contain barriers, the presence of which, however, can only be verified by a human inspector.

Some tools also include general recommendations about the guidelines that cannot be searched in the code by the tool and must always be checked by human experts. Successive versions of WCAG produced by the WAI endeavoured to reduce the risk of giving incorrect computational representation to guidelines. To this end, versions 2.0 and 2.1 of WCAG contain success criteria that are testable statements, reducing the ambiguity and interpretability of guidelines in previous versions.

Specific technologies are used to define and programme automated accessibility tests. A practical option is to use XPath and XQuery expressions that can be directly applied to the evaluation of the HTML code of the web pages (Arrue et al. 2008; Luque et al. 2005). For example, the technique H36 'Using alt attributes on images used as submit buttons' of the WCAG 2.0 states that image type input elements require alternative texts to provide a functional label. Using XQuery, a test for verifying this technique can be defined as follows:

```
let $var: = doc("web_page.xml")//INPUT[@TYPE="img" and not(@ALT)
```

This query detects those image type input elements, which do not have alternative text (defined by the *alt* attribute).

26.3.1 Metrics and Personalised Accessibility Evaluation

Automatic accessibility evaluation tools report their results using metrics. A metric produces a score calculated from predefined parameters that have been counted, measured or calculated from the failures or successes detected in the selected guidelines.

26.3.1.1 Qualitative Metrics

Accessibility evaluation methods have to provide clear and useful results to inform the user whether or not a web page can be effectively used by people with diverse disabilities. The WACAG 2.0 guidelines provide three levels of conformance depending on the Web page satisfying all the Success Criteria for each level (or the provision of a conforming alternate version): A, AA and AAA. In addition to this score, most automatic evaluation tools give pointers to the code segments that fail specific checkpoints [Read Part 5 'Automatic Web Accessibility Metrics' for further information].

26.3.1.2 User-Tailored Evaluation

The qualitative score of WCAG is useful to verify whether a website legally fulfils accessibility requirements or not, but it may not be informative enough for the requirements of individuals. Systems able to provide more detailed data about the fulfilment of the WCAG 2 four principles present more detailed information about the type of barriers in a specific website which people with specific disabilities may come across. For example, the score on the *perceivable* principle can mostly be associated with sensory disabilities, while the score on *operable* principally affects (though not exclusively) people with motor disabilities. The *understandable* principle is mainly related to cognitive accessibility and the *robust* principle takes into account the technology (equipment and applications) used to access the Web, including Assistive Technology (Aizpurua 2017).

More detailed personalization of the accessibility evaluation is possible when the criteria to be evaluated can be fine-tuned. In this case, only the guidelines affecting a specific person (also taking into consideration their particular impact on the specific user) are evaluated. In this way, some tools can be tailored to particular features of specific types of users (Vigo et al. 2007b). For instance, Mauve (HIIS Lab/ISTI-CNR 2018) offers the possibility of evaluating web content using end-user-specific guidelines (such as usability criteria identified to improve web navigation for vision-impaired people). Other tools include options to select and configure the checkpoints to be verified (Arrue et al. 2008). For example, TAW includes a standalone application (CTIC 2018b) which allows the selection of checkpoints to be checked as well as the creation of new custom rules.

26.3.1.3 Quantitative Metrics

A web page that fails a specific checkpoint one in ten times is nearer to being accessible than a page that fails the same checkpoint ten out of ten times, nevertheless, both are scored equally by qualitative metrics. In order to take into account how far a web page is from being accessible, quantitative metrics have been proposed (Vigo and Brajnik 2011). These metrics take into account the relative frequency and the impact of each failure, providing a more discriminative accessibility score. Their results are useful to monitor the evolution of the accessibility of a website, for instance. Vigo et al. (2007a) used this system to create a web service called EvalBot that was able to rank by personal accessibility (using their own quantitative metrics) the first 20 websites proposed by a web search engine when answering a query.

26.3.2 Available Automated Evaluation Tools

A number of automated accessibility evaluation tools have been implemented. A comprehensive list of available web accessibility evaluation tools can be found on the W3C-Web Accessibility Initiative (WAI) website (Egger and Abou-Zahra 2016). Its advanced search functionality enables evaluation tools to be looked for according to various criteria: sets of guidelines, language, type of tool, technology, provided assistance, scope and license type.

Accessibility evaluation tools can be classified following diverse criteria:

- Free versus commercial: while a few of them are commercial tools, the majority of them are freely available.
- Platform: depending on where they run they can be classified in local or standalone applications, online services or browser extensions.
- Evaluation scope: distinguishes the evaluation of single pages, sets of pages, and complete websites.
- Evaluation only versus evaluation and repair: while most tools only provide the option to evaluate, some tools are also able to provide guidance on the repair process.
- Ways for reporting accessibility issues: the evaluation reports generated may contain step-by-step evaluation guidance, or may display information about barriers within web pages, or even modified presentations of web pages.
- Guidelines used: the standards or guideline sets they employ for evaluating accessibility.

Numerous automated accessibility evaluation tools, such as AChecker (AChecker 2011; Gay and Li 2010), or Mauve (HIIS Lab/ISTI-CNR 2018; Schiavone and Paternò 2015; Paternò and Schiavone 2015) are freely available. While other tools, such as WAVE (WebAIM 2018), TAW (CTIC 2018a), or Tenon (2018), in addition to a free version have a commercial version.

Several tools can perform accessibility checks during the development process, by installing plugins within the development tool. This is the case, for instance of aDesigner (Eclipse 2018), Continuous Accessibility Testing for Eclipse (WEBaccessibility 2018), or Accessibility checker for CKEditor 4 (CKEditor 2018). Besides the accessibility evaluation, these tools can also provide other functionalities such as simulation of disabilities (aDesigner) or assistance to repair the accessibility barriers discovered (Accessibility checker).

There are tools which can be installed as add-ons in a web browser to evaluate the accessibility of the created web pages, such as aXe (2018), which enables accessi-

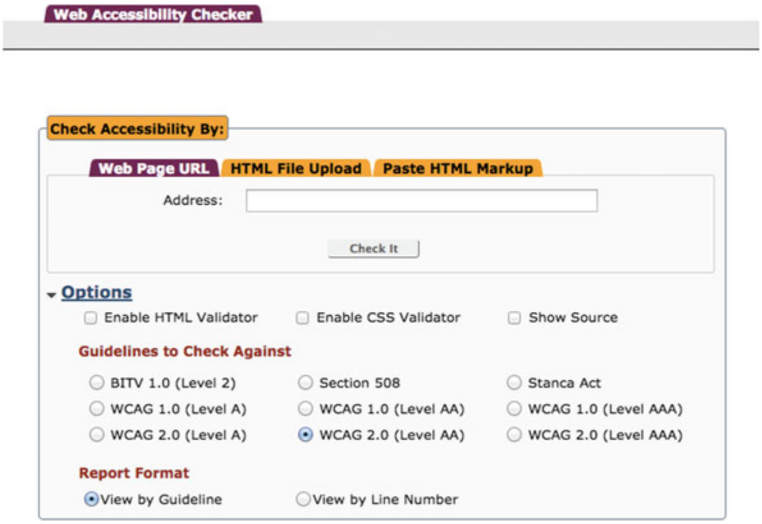

Fig. 26.2 A screenshot of AChecker evaluation tool main web page

bility evaluations to be run without sending the page code to external servers. Other tools, such as Audits (Basques 2018), are already integrated into the web browser. For example, audits can measure the accessibility of the page, its adequacy for best practices, and it can also simulate how the web page would be displayed in some mobile devices.

26.3.3 AChecker: An Example of Use of an Automated Evaluation Tool

Applying automated tools for accessibility evaluation is a simple and fast way to obtain a report containing accessibility errors and warnings. Most of the tools provide a form in which the evaluator may insert the URL of the web page to evaluate. Some tools also provide a feature to evaluate directly inserted HTML code or the code of an uploaded file. All these features are included in AChecker and can be seen in Fig. 26.2. In addition, this tool includes diverse features for configuring the evaluation process: selecting the technologies to be verified (HTML/CSS), the guidelines to be tested (BITV 1.0, Section 508, WCAG 1.0, WCAG 2.0, etc.), and the format of the report (by guideline, by line number).

The report obtained by the evaluation tools contains the necessary information to assist evaluators in understanding the detected errors/warnings as well as to repair them. Figure 26.3 shows a report obtained by AChecker. It contains links to get more

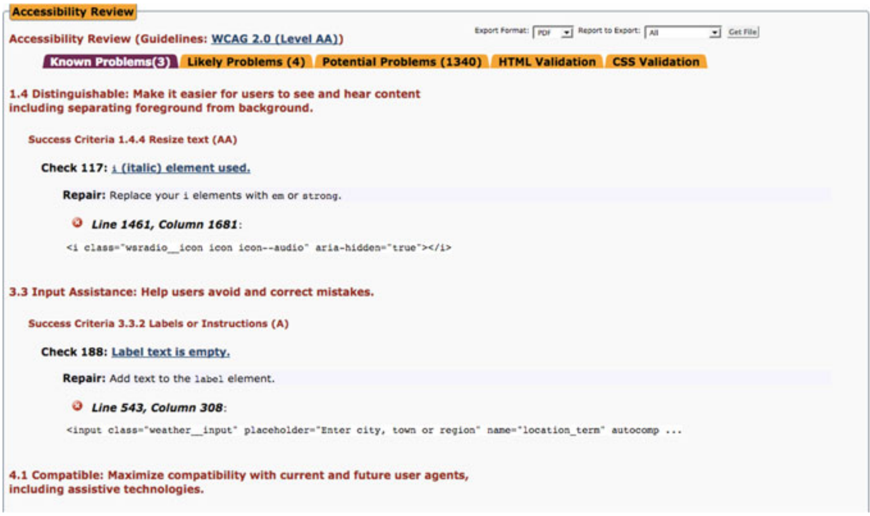

Fig. 26.3 A report from AChecker

information about the checks and the line and column number where the error was found.

26.3.4 Limitations of Automated Accessibility Evaluation

Although automated evaluation tools can assist evaluators in the evaluation process, they should not be the only method used for the accessibility evaluation of websites. As stated by the WAI (Abou-Zahra et al. 2017) 'We cannot check all accessibility aspects automatically. Human judgment is required. Sometimes evaluation tools can produce false or misleading results. Web accessibility evaluation tools cannot determine accessibility, they can only assist in doing so'.

In fact, a number of studies have been carried out in order to detect the limitations of automated evaluation. Thatcher et al. (2006) used 40 different benchmark tests to test 6 evaluation tools obtaining a failure rate between 5 and 42%. In another study carried out by Brajnik (2004) comparing the results obtained by two different automated evaluation tools, 33% of false positives and up to 35% of false negatives were detected. Vigo et al. (2013) analysed the coverage of six automated evaluation tools based on WCAG 2.0 guidelines conformance and found that, at most, 50% of the success criteria were covered.

Therefore, evaluators are often obliged to apply more than one automated tool and then to compare and aggregate the results in order to obtain better evaluation results. To aggregate and summarise the results from diverse tools can be difficult. For this reason, the WAI defined Evaluation and Report Language (EARL) (Abou-Zahra and

Henry 2011), a machine-readable format, for expressing test results. Its objective is to facilitate the processing of test results obtained from different testing tools. The EARL 1.0 version is available on the WAI/W3C website. Unfortunately, it would appear that little, if any, progress has been made since 2011.

26.4 Crowdsourcing-Based Tools for Web Accessibility

Website owners have the responsibility to make their content accessible. However, since accessibility evaluation and barrier removal are complex tasks, the rate of non-accessible content produced is constantly increasing. There are various reasons behind this: development haste, ignorance, lack of adequate tools, etc.

On the other hand, users with disabilities are aware of the barriers they find when accessing a specific website. Usually, they can report them to the owner via feedback forms or email, but even if their criticisms are taken into account, issuing a fixed version of the website usually takes time. Some authors think that involving a larger community of volunteers collaborating to fix accessibility bugs may help to increase the rate of accessible websites.

Crowdsourcing-based tools aim to reduce the burden on owners of accessibility evaluation and fixing and to shorten the time required to provide an accessibility barrier-free version by recruiting volunteers throughout the world to cooperate in barrier detection and elimination.

The underlying idea is that the inclusion within the page code of adequate metadata is a key issue to ensure web accessibility. Metadata provide structural information required by adapted browsers and screen readers to understand text structure and hierarchy (tables, headings, lists, etc.) in order to render it adequately. In addition, metadata complement information presented in inaccessible modes with alternative information in accessible modes (e.g. a text to describe a picture) (Takagi et al. 2008).[1] The availability of sufficient accessibility metadata rapidly alters the situation: the user is notified when new metadata has been added to overcome a reported barrier. When the user re-visits the page the available metadata is used to adequately render its content (Brady and Bigham 2015).

Adding semantic value to the different elements of the web is very laborious. This means correctly identifying elements of the web page and establishing what its purpose is (for example, whether it is an element of a navigation menu, or part of the main content or a decorative element, etc.) Even though there are methods that facilitate the annotation process of the elements, such as those proposed by Takagi et al. (2002) or Harper and Bechhofer (2007), the annotation process is still very expensive. However, with adequate tools, crowdsourcing can greatly help alleviate these problems since work can be divided into small tasks to be carried out by remote workers.

[1] Transcoding methods also depend on the availability of metadata [See Part 4: "User Interface Adaptation for Accessibility" for more information on transcoding].

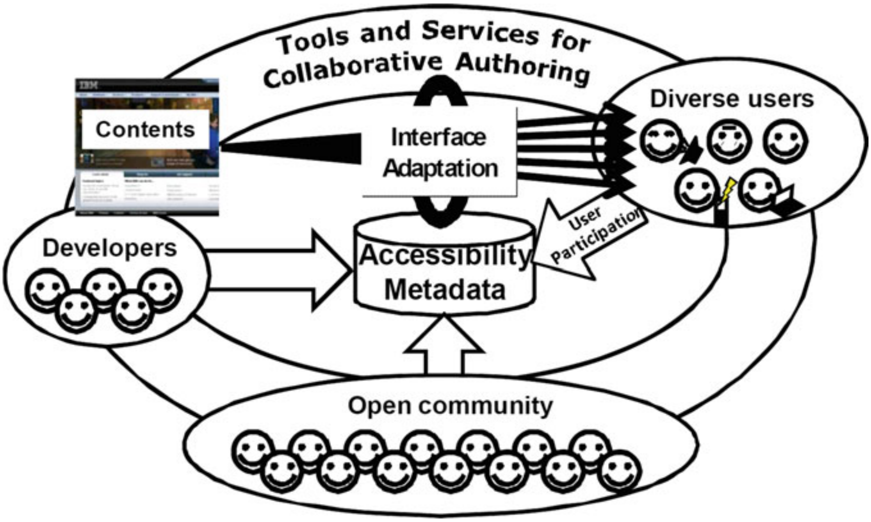

Fig. 26.4 Concept of social accessibility approach [Taken from (Takagi et al. 2008)]

Since the quantity and quality of accessibility metadata embedded in the content by web designers and developers is not sufficient, Takagi et al. (2008) propose the application of crowdsourcing methods to revert the situation. Appointing a large community of supporters would enable the following scenario: when a user finds an accessibility barrier the problem is reported to the volunteer community. The volunteers create and publish the required accessibility metadata (after a discussion if necessary). The user is prompted and then can access the page without problems.

26.4.1 Tools for Crowdsourcing-Based Social Accessibility

Following the Social Accessibility approach, Takagi et al. (2008) created a pilot system for collaborative metadata authoring aimed at achieving web accessibility.

The community was composed of three types of stakeholders: **users** with diverse accessibility needs, who reported their accessibility evaluations; **volunteers** were members of the open community, who received users' reports and collaborated to propose adequate metadata to render the reported accessibility barrier accessible; finally, **website owners and developers**, who received these metadata with a view to future accessibility improvements (Fig. 26.4).

In this way, users and volunteers were able to improve the accessibility of any content on the Internet by collaboratively authoring the accessibility metadata. In order to enable the collaboration process, Takagi et al. (2008) proposed new types of authoring tools and collaboration services following these principles:

- The authoring tools should be usable enough to allow non-technical and accessibility novice volunteers to join in.
- The collaboration services should help participants to work together in the right direction and achieve appropriate sets of metadata.
- The service should effectively motivate volunteers to contribute to the activity, and should reduce their stress load by giving them opportunities to ask questions and to get instructions.

To this end the following tools were designed:

- Some client-side code, installed in the user browser, supplied scripts to handle both problem reports and transcoding (which used the added metadata to render the web page more accessible). Since the pilot system was aimed at blind users provided with screen readers, the end user component was mainly written using screen reader scripts transparent to the users' regular Web access.
- A browser plug-in, installed in the supporter browser. This plug-in offered an extension sidebar with various functions: a popup appearing whenever a request was issued by a user; an authoring view, which had a simulation area of the reading text with a screen reader; a pushbutton to submit the created metadata to the server; and a 'page map' function to give a visual overview of the accessibility status of the page.
- Each participant was provided with a portal function that contained basic information, activity ratings, rankings among participants, pending requests, and hot sites where problems were reported or where metadata was actively being created.

Tools for collaboration were created to enable collective creation of metadata. Real-time collaboration was performed by means of a group chat. For complex requests, supporters had access to a Web form to send an email query to ask the requester (end user) for clarification. Asynchronous collaboration was carried out by means of a discussion thread associated with each request and metadata item, linked to the toolbar or the portal page. Finally, to accumulate the acquired knowledge supporters were able to describe and publish guidelines for other supporters by creating new Wiki pages in a Wiki integrated into the system.

26.4.2 Crowdsourcing-Based Web Accessibility Evaluation

Since automatic Web accessibility evaluation can only cover part of the checkpoints, manual evaluation by human experts is required for the remaining ones. To recruit enough experts to evaluate large numbers of web pages is not easy. To solve this problem, crowdsourcing-based Web evaluation has been proposed. The basic idea is to divide the accessibility evaluation into small 'minitasks' and to submit these to several evaluators working in parallel. Each mini-task is carried out by a number of evaluators. The result for each mini-task can be 'accessible', 'inaccessible' or none. The accessibility score result is obtained by the majority of the answers received.

Since expertise among evaluators can be very diverse, more complex metrics have been tried, in order to reduce the weight of non-expert opinions and to boost the importance of expert scores. For instance, Song et al. (2018), distribute a number of 'golden tasks' which have been previously rated by experts. They use the answers from the evaluators to rate their expertise and subsequently assign a weight to their scores. Additionally, these authors consider the time required to answer 'accessible' (allegedly requiring more time) or 'non-accessible' (supposedly requiring less time) to identify possible inaccurate or random answers. In this way, they achieve reasonably accurate evaluations investing half the time of similar crowdsourcing Web evaluation systems.

26.5 Support Tools for Web Accessibility Evaluations with End Users

User testing for web accessibility evaluation has some drawbacks in comparison to other evaluation methods. For example, higher cost, longer development times, difficulty of finding an appropriate sample of users representative of the target audience, etc. Moreover, setting up a user testing session requires taking into consideration many potentially challenging aspects: it is necessary to install the appropriate assistive technology type and version required by each user, to ensure certain room facilities, etc. Nevertheless, user testing is an essential part of the accessibility evaluation process if the barriers usually experienced by real users are to be accurately detected and identified. Therefore, tools to support user testing are highly desired.

26.5.1 Types of User Tests

Similarly to other experimental studies, Web accessibility evaluations with users can be carried out with supervision (moderated evaluations) or unsupervised (unmoderated evaluations).

For moderated evaluations one or more supervisors are in charge of preparing the set-up, explaining the procedures and observing or recording the participants while they are trying to perform the given tasks. Direct observation of participants enables evaluators to obtain valuable information about the barriers they face, the strategies used to overcome them as well as how they use their assistive technology. Moderated evaluations are usually carried out in a laboratory, enabling valuable information to be gathered about the participants. However, it is not always easy to recruit the appropriate number of participants with the required characteristics. Moreover, they may have difficulties to travel to the experiment location. In order to adequately carry out the accessibility evaluation, assistive technologies should be properly configured for each participant. Otherwise, they might experience barriers due to the inadequate configuration of the assistive technology, altering the results of the accessibility evaluation (Petrie et al. 2006).

Alternatively, remotely carried out evaluations mitigate some of the aforementioned issues. Participants are not required to travel to the laboratory, they have their environment already adapted to their needs and the interaction with the Web is carried out in a more naturalistic manner. Remote evaluations can also be moderated, when an evaluator supervises the tasks using communication software such as, for example, a videoconference system, screen-sharing software, etc. (Hammontree et al. 1994). However, the moderated remote evaluation also has drawbacks such as the lack of control over participant environment (noise, interruptions, etc.) or the difficulty to install and configure special software if required.

These problems are alleviated by carrying out unmoderated evaluations in which participants carry out the tasks whenever and wherever they want without any supervision. In such evaluations, multiple tests are carried out simultaneously. Data are analysed once the evaluations have been finished. In this case, the required tasks should be carefully described in order to make sure that participants have understood the test correctly. Tasks should be short enough to maintain the participant's attention, taking into consideration that, in online studies, participants' attention tends to fade after 20 or 30 min (Albert et al. 2010).

Unmoderated remote evaluations enable a wider population to be reached and are carried out in a more realistic environment. In this case, participants use their own computer as they would usually do. However, there are also several risks: the total lack of control over the evaluation environment and participant. They might suffer distractions such as, for example a telephone call, other people in the room, etc. In addition, even if the participant is focused on carrying out the tasks, it is often the case that fewer accessibility barriers or usability issues are reported (Andreasen et al. 2007).

Evidently, the large amount of valuable information about the interaction of the user with web pages, obtained through logging web usage can be used to analyse several types of accessibility barriers. However, these data have also limitations that make it difficult to detect exactly were the accessibility barriers are. For example, no information can be obtained about why a user performs specific actions or how users utilise assistive technology (Petrie et al. 2006).

26.5.2 Architecture of Tools for Remote Web User Evaluations

Depending on the required purpose and functionality, user testing tools can be installed on the computer, in a proxy (configuring the browser to redirect web requests through the tool) or inserting new code into the target page. Let us describe the most commonly used architectures, the types of data collected, and some applications.

Tools that are used to carry out remote or in situ user evaluations are usually located in one of three locations: a server, a proxy between the server and the participant device or the client (device). In the following paragraphs, the strengths and weaknesses of each approach are discussed and summarised.

26.5.2.1 Located on the Server Side

When the user evaluation tool is located on a server (Scholtz et al. 1998; Etgen and Cantor 1999; Paganelli and Paternò 2002; de Santana and Baranauskas 2015; Google 2018), participants can carry out the experimentation transparently and comfortably. Some code is added to the web pages to gather the required events. Therefore, no installation or configuration is required by participants. The only requirement is to access the test web page and to perform the pertinent tasks.

This system has a major drawback: only pages located in the same server can be analysed, hence no third party web pages can be considered. Another inconvenience is the lack of control over the evaluation. For example, if the participant navigates to a page that is out of the server, closes the navigator tab, opens a new one, etc., large quantities of data are lost affecting the quality of the evaluation.

26.5.2.2 Located on the Proxy Side

Proxy tools are located between the participants' device and the website server (Hong et al. 2001; Atterer et al. 2006; Loop11 2018). In order to perform the tasks relating to the evaluation, each participant has to use a web browser configured to access the Internet through the proxy. Alternatively, researchers can provide a link to the proxy from which participants start the test and the following web page requests are made through the proxy. All the requests made by the web browser (texts, images, videos, etc.) are first caught by the proxy, which adds the necessary code to gather user interface events before reaching the participants' device.

As opposed to tools located on servers, proxy tools make it possible to analyse any existing web page. In this way, researchers can analyse how the users navigate in different types of web pages, observe their interests, measure accessibility, etc. Nevertheless, proxy systems also have some drawbacks. When the test is made through a proxy without browser configuration, the data is lost when participants close tabs or when they type URLs manually. Adequately configuring the web browser avoids these problems but it makes the process more obstructive than when accessing from a URL. In addition, proxy systems usually have restrictions with secure web pages and dynamic content.

26.5.2.3 Located on the Client Side

These tools are located on the participants' device, usually as a browser add-on, but web browsers especially designed for user evaluations can be also found (Claypool et al. 2001; Arrue et al. 2018; TechSmith 2018). With this approach, all the web pages can be analysed. The number of events that can be gathered is larger since browser selections such as back, history, or the context menu produced by the right click can be gathered. These events are appropriate to learn the interest that a participant has for a web page, for instance if it has been saved, printed or added to the bookmarks.

As opposed to proxy or server systems with client tools there is no problem closing tabs or inserting the desired URL manually. Clients can handle dynamic content and secure connections without problems. Nevertheless, data from secure connections have to be carefully handled, no keyboard events or information about web pages elements can be gathered, as this information can be used to get personal data such as passwords, bank accounts, etc. The need to perform a previous software installation before starting the experimental session is a major drawback, proving to be rather obstructive for the participant. In all three cases, when user interface events are gathered, a transmission mechanism is needed to send the data from the browser to the server.

26.5.3 Collected Data

Various types of data that are useful to enhance web accessibility can be obtained from tools that locally or remotely carry out user evaluations.

Early tools were only able to gather log data generated by servers (Hallam-Baker and Behlendor 2018) or the path followed by the participant (Scholtz et al. 1998). These logs are generated by the server to get information about its performance or the problems that may occur. It records the request (texts, images, etc.) made by the web browser in a known format (*host, ident, authuser, date, request, status, bytes*). However, with adequate data processing methods, data useful for finding problems related to navigation within the web pages in a site can be obtained. However, data obtained from server logs is limited and no information about the user interaction in a web page can be obtained.

On the other hand, with the use of technologies, such as JavaScript or Java Applets, events generated by the cursor or keyboard can be gathered (Kacmarcik and Leithead 2018). These events provide richer information about the participant compared to server logs (Atterer et al. 2006).

The most relevant events for obtaining information about the participant are those related to the mouse (*mouseup, mousedown, mousemove*), touch (*touchstart, touchend, touchmove, touchleave*), keyboard (*keyup, keydown*) or with the web browser (back-forward buttons, contextual-menu, zoom, etc.). These events usually provide information about the position (x, y) where the event was triggered, information about elements related to the event, which button or keys were pressed to trigger the event, etc. This information can be used to calculate measurements (cursor speed, time on page, trajectory linearity, etc.) to determine when the user is having problems, to detect accessibility issues or to classify the input device used.

For example, Mackenzie et al. (2001) defined seven accuracy measures (Target Re-entry, Task Axis Crossing, Movement Direction Change, Orthogonal Direction Change, Movement Variability, Movement Error and Movement Offset) to evaluate differences between the pointing devices. Later, Kates et al. (2002) added six new characteristics (Distance Travelled Relative to Cursor Displacement, Distribution of Distance Travelled for a Range of Cursor Speeds, Sub-movements, Cursor Distance Travelled Away from the Target, Distribution of Distance Travelled for a Range of Curvatures, Distribution of Distance Travelled for a Range of Radii from the Target). With these, it is possible to detect accessibility difficulties during target selection on the web page of people with reduced mobility.

Another alternative for detecting accessibility issues is to find the common events streams (event sequence) of a page. Studying the events streams with a post hoc analysis, several accessibility issues can be revealed such as, the lack of contrast, the confusion of text with links or unexpected usage patterns (de Santana and Baranauskas 2015). Vigo and Harper (2017) on the other hand, identified four strategies that might indicate navigation problems (*quick preview, asking for help, retracing* and *quick revisitation*) and created algorithms to detect those problems during navigation. This strategy enables the provision of interventions as the problems occur, making web navigation easier.

It is also possible to obtain information about the characteristics of the participants with data obtained from events (Hurst et al. 2008; Valencia et al. 2015) or to analyse the input performance from natural interactions (Gajos et al. 2012). Gajos et al. (2012) used an algorithm to extract the deliberated point and click movements to obtain laboratory like measurements. Hurst et al. (2008), distinguished between young, young adults, older adults and people with Parkinson's disease using their input events obtained in laboratory settings. Valencia et al. (2015) instead, obtained different measures from natural web navigation that can be used to detect accessibility issues or to identify the assistive technology used by the users. Another possible use of the information obtained from events is to extract the interest of the person on a page. For instance, the time spent on the page and the amount of scrolling were found as good interest indicators by Claypool et al. (2001).

26.5.4 Remotest: A Tool to Assist Evaluations with Users

Remotest is an application that assists researchers to define experiments, manage experimental remote/in situ sessions and analyse the gathered interaction data (Arrue et al. 2018). The main objective of the evaluations is to measure the accessibility and usability of a website during use. The platform admits a wide range of experiments. Various examples are, to study the user behaviour when performing a task in different websites, to analyse and compare navigational strategies of different types of participants when interacting with the same website, to evaluate the accessibility-in-use of several websites, to gather significant information through surveys, to measure user satisfaction when using a certain web service, to analyse user performance improvement when interacting with adapted versions of original web pages, etc.

The platform is split into four modules:

- The **Experimenter Module**: helps to specify the type of experiment, tasks, stimuli, the procedure of the experimental session, data to gather and sample.
- The **Coordinator Module**: stores and manages the experiments, creates stimuli to be presented during the experimental sessions (questionnaires, task description web page, task completion alert, etc.) and stores interaction data.
- The **Participant Module**: conducts the experimental sessions following the specification established by the Experimenter and Coordinator Modules. It presents login procedures, questionnaires, task descriptions pages, etc. It also tracks the participant throughout the experimental session and collects interaction data.
- The **Results Viewer Module**: deals with the presentation of the interaction data gathered in experimental sessions. For this purpose, it implements functions for collecting the data from the Coordinator Module, structuring it in understandable blocks of events and presenting them to the experimenter through a web application. In addition several statistical values, grouped by pages, tasks or users, can be visualised.

Remotest has been used to detect accessibility barriers related to motor and vision restrictions (Valencia et al. 2015). In addition, the results provided by remotest have been used to create adaptation techniques for transcoding and to evaluate the efficacy of the transcoded pages (Valencia 2017).

26.6 Discussion

Automatic tools can help make pages more accessible but it is essential that developers must receive training which enables them to gain a better understanding of the guidelines, their application and of the users themselves. Knowing the users would also foster a greater awareness of the need for web accessibility something which is otherwise often seen as being of secondary importance.

The disparity of results from web accessibility evaluations is also a problem. Not only do the various automatic evaluation tools provide different results, the evaluations carried out by different experts rarely reach the same conclusions. Therefore, it is necessary to study what the causes for this divergence are, and what can be done to achieve more coherent results.

A simpler wording for the guidelines might facilitate their applications and unambiguous machine-readable specifications would presumably help to produce more uniform results.

Another problem is the time required to perform web page accessibility evaluations. When the evaluation is made during the development process the extra time required is minimal. But analysing existing websites can be quite laborious when a site has numerous pages. Therefore, it is necessary to explore new ways to facilitate the evaluation process. For example, automatic evaluation tools could incorporate machine-learning modules to elucidate compliance and even the arrangement of some rules that cannot currently be evaluated. For example, automatic tools could be used to find out the suitability of alternative text and even to propose a substitute text, to check the complexity of the texts, suitability of the structure, etc. It could also be complemented with crowdsourcing techniques, to evaluate the adequacy of the results obtained by machine-learning techniques.

It cannot be overlooked that in order to entirely guarantee web page accessibility, it is necessary that the pages be tested by people with disabilities. Remote evaluation tools, are a cheap option that can reach larger samples of users. Despite the advantages of remote testing, 'in situ' user tests continue to be vital for developers to be able to see how users navigate, what difficulties they come across and what strategies they use to overcome those difficulties.

26.7 Future Directions

Advances in the definition of more specific accessibility guidelines which can be matched with the page code will contribute to more effective automatic accessibility evaluation. Unambiguous guidelines would also help to decrease the discrepancy between the accessibility reports generated by the various automatic evaluators, as well as between different human experts.

The integration of automatic accessibility evaluation tools into common use for professional web design and development frameworks will facilitate the creation of accessible pages and will combat the resistance of some designers to tackling accessibility issues.

Adaptability to the specific user of the accessibility evaluation would allow a more practical approach to help each individual to find pages that do not have accessibility barriers for her or him.

Methodologies, with their supporting tools, that include the complete series of evaluation, (automatic, by experts and by users, preferably in real scenarios) would ensure higher levels of accessibility.

26.8 Author's Opinion of the Field

Web accessibility is perceived by numerous designers as requiring extra effort and producing no extra benefits. Frequently, accessibility is only considered when legal or marketing requirements impose its fulfilment. This is due to the fact that designers often lack training in accessible design, and do not have the development tools that could help them in this field.

To reverse this tendency, it is essential that designers accept accessibility as a factor of quality, and they perceive it as a step towards their quality objective. To this end, accessibility must be integrated in the same way as any other requirement within the methodologies and tools that developers use every day.

Therefore, the advancement of web accessibility is closely tied to the availability of advanced, easy to use, tools that are well integrated into the professional design and development frameworks. These tools must be effective in helping the designer to develop websites that fulfil standard web accessibility guidelines and to test them efficiently. The scope of the required tools is broad: not only for automatic accessibility verification, but also, for support to expert inspections and different types of user testing.

26.9 Conclusions

Progress in web accessibility is closely associated with the availability of tools for design, evaluation and reparation. Automatic web accessibility evaluation has proved to be an essential starting point towards obtaining accessible websites, but it is important to underline its limitations: it can only detect failure to fulfil accessibility guidelines that can be expressed in machine-readable mode. More general guidelines must be checked by human experts. Moreover, the absence of detected compliance failures does not guarantee full accessibility. An evaluation by an ample and diverse sample of real users can detect barriers which are invisible to automatic evaluators and experts.

Most accessibility guidelines focus on 'general accessibility', which could be too restrictive for particular users with specific accessibility problems. Tailoring the evaluation to the concrete needs of specific users can help them to find sites which are useful for them, even if they may not be generally accessible. This kind of personalised evaluation may also help in page transcoding for specific users.

User tests are usually conducted with a small number of users working in artificial environments. A key issue to enhance web accessibility is to be able to evaluate the existence of barriers while people are using the Web with their own equipment (frequently including well-adapted assistive technology). To this end, tools to gather usage data from remote users and to efficiently process these data are indispensable.

Acknowledgements The authors are members of the EGOKITUZ/ADIAN research team, supported by the Basque Government, Department of Education, Universities and Research under grant (IT980-16).

References

Abascal J, Nicolle C (2001) Why inclusive design guidelines? In: Nicolle C, Abascal J (eds) Inclusive design guidelines for HCI, Taylor & Francis, London, pp 3–13

Abou-Zahra S, Henry SL (eds) (2011) WAI/W3C evaluation and report language (EARL) overview. https://www.w3.org/WAI/standards-guidelines/earl/. Accessed 1 Aug 2018

Abou-Zahra S, Steenhout N, Keen L (eds) (2017) Selecting web accessibility evaluation tools. https://www.w3.org/WAI/test-evaluate/tools/selecting/. Accessed 1 Aug 2018

Abou-Zahra S (2008) Web accessibility evaluation. In: Harper S, Yesilada Y (eds) Web accessibility: a foundation for research. Springer, London, pp 79–106

AccessIT (2017) Accessibility compliance and remediation methodology (ACRM) http://www.dor.ca.gov/Access-IT/documents/ACRM%2005102017.docx. Accessed 1 Aug 2018

AChecker (2011) Web accessibility checker. https://achecker.ca/checker/. Accessed 1 Aug 2018

Aizpurua A (2017) Contributions to web accessibility: device-tailored evaluation, user-tailored interface generation and the interplay with user experience. PhD Dissertation, University of the Basque Country/Euskal Herriko Unibertsitatea

Albert W, Tullis T, Tedesco D (2010) Beyond the usability lab: conducting large-scale online user experience studies. Morgan Kaufmann, San Francisco

Andreasen MS, Nielsen HV, Schrøder SO, Stage J (2007) What happened to remote usability testing?: An empirical study of three methods. In: Proceedings of the SIGCHI conference on human factors in computing systems (CHI '07). ACM, New York, pp 1405–1414. https://doi.org/10.1145/1240624.1240838

Arrue M, Valencia X, Pérez JE, Moreno L, Abascal J (2018) Inclusive web empirical studies in remote and in-situ settings: a user evaluation of the RemoTest platform. Int J Hum–Comput Interact. https://doi.org/10.1080/10447318.2018.1473941

Arrue M, Vigo M, Abascal J (2008) Including heterogeneous web accessibility guidelines in the development process. In: Gulliksen J, Harning MB, Palanque P, van der Veer GC, Wesson J (eds) Engineering Interactive Systems EHCI 2007. Lecture notes in computer science, vol 4940. Springer, Berlin, Heidelberg, pp 620–637

Atterer R, Wnuk M, Schmidt A (2006) Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In Proceedings of the 15th international conference on world wide web. ACM, New York, pp 203–212. https://doi.org/10.1145/1135777.1135811

aXe (2018) Easy accessibility testing with aXe. https://www.axe-coreorg/. Accessed 1 Aug 2018

Basques K (2018) What's new in DevTools (Chrome 60). https://developers.google.com/web/updates/2017/05/devtools-release-notes#lighthouse. Accessed 1 Aug 2018

BBC (2018) BBC digital guidelines accessibility standards and guidelines. http://www.bbc.co.uk/guidelines/futuremedia/accessibility/. Accessed 1 Aug 2018

Brady E, Bigham J (2015) Crowdsourcing accessibility: human-powered access technologies. Found Trends Hum-Comput Interact 8(4):273–372. https://doi.org/10.1561/1100000050

Brajnik G (2004) Comparing accessibility evaluation tools: a method for tool effectiveness. Univ Access Inf Soc 3(3):252–263. https://doi.org/10.1007/s10209-004-0105-y

Brajnik G (2008a) A comparative test of web accessibility evaluation methods. In: Proceedings of the 10th international ACM SIGACCESS conference on computers and accessibility. ACM, New York, pp 113–120. https://doi.org/10.1145/1414471.1414494

Brajnik G (2008b) Beyond conformance: the role of accessibility evaluation methods. In: Hartmann S, et al (eds) Web information systems engineering WISE 2008. Lecture notes in computer science, vol 5176. Springer, Berlin, Heidelberg, pp 63–80

Brajnik G (2009) Barrier walkthrough. https://users.dimi.uniud.it/~giorgiobrajnik/projects/bw/bwhtml. Accessed 1 Aug 2018

Caldwell B, Cooper M, Reid LG, Vanderheiden G (2008) WAI/W3C web content accessibility guidelines (WCAG) 20. http://www.w3.org/TR/WCAG20/. Accessed 1 Aug 2018

Chisholm W, Vanderheiden G, Jacobs I (eds) (1999) WAI/W3C web content accessibility guidelines 10 http://www.w3.org/TR/WAI-WEBCONTENT/. Accessed 1 Aug 2018

CKEditor (2018) Accessibility checker. https://ckeditor.com/cke4/addon/a11ychecker. Accessed 1 Aug 2018

Claypool M, Le P, Wased M, Brown D (2001) Implicit interest indicators. In Proceedings of the 6th international conference on intelligent user interfaces. ACM, New York, pp 33–40. https://doi.org/10.1145/359784.359836

CTIC (2018a) TAW web accessibility test. https://www.tawdisnet/?lang=en#. Accessed 1 Aug 2018

CTIC (2018b) TAW monitor. https://www.tawdisnet/proj#c2. Accessed 1 Aug 2018

de Santana VF, Baranauskas MCC (2015) WELFIT: A remote evaluation tool for identifying web usage patterns through client-side logging. Int J Hum-Comput Stud 76:40–49. https://doi.org/10.1016/j.ijhcs.2014.12.005

Eclipse (2018) ACTF aDesigner. http://www.eclipseorg/actf/downloads/tools/aDesigner/. Accessed 1 Aug 2018

Egger E, Abou-Zahra S (2016) WAI/W3C web accessibility evaluation tools list. https://www.w3.org/WAI/ER/tools/. Accessed 1 Aug 2018

Etgen M, Cantor J (1999) What does getting WET (web event-logging tool) mean for web usability. In Proceedings of 5th human factors and the web conference, NIST, Gaithersburg, 3 June 1999

Gajos KZ, Reinecke K, Herrmann C (2012) Accurate measurements of pointing performance from in situ observations. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, New York, pp 3157–3166. https://doi.org/10.1145/2207676.2208733

Gay GR, Li CA (2010) Open, interactive, customizable, web accessibility checking. In: Proceedings of the 7th international cross disciplinary conference on web accessibility. ACM, New York, p 2. https://doi.org/10.1145/1805986.1806019

Google (2018) Google analytics. http://www.google.com/analytics/. Accessed 1 Aug 2018

Hallam-Baker PM, Behlendor B (2018) W3C extended log file format. https://www.w3.org/TR/WD-logfile. Accessed 1 Aug 2018

Hammontree M, Weiler P, Nayak N (1994) Remote usability testing. Interactions 1(3):21–25. https://doi.org/10.1145/182966.182969

Harper S, Bechhofer S (2007) SADIe: Structural semantics for accessibility and device independence. ACM Trans Comput-Hum Interact (TOCHI) 14(2):10. https://doi.org/10.1145/1275511.1275516

Harper S, Yesilada Y (eds) (2008) Web accessibility: a foundation for research. Springer, London

Henry SL (2007) Just ask: integrating accessibility throughout design. http://uiaccess.com/accessucd/. Accessed 1 Aug 2018

Henry SL (ed) (2018a) WAI/W3C introduction to web accessibility web accessibility initiative WAI (W3C). https://www.w3.org/WAI/fundamentals/accessibility-intro/#what. Accessed 1 Aug 2018

Henry SL (ed) (2018b) WAI/W3C evaluating web accessibility overview. https://www.w3.org/WAI/eval/Overview. Accessed 1 Aug 2018

Henry SL, McGee L (eds) (2018) Accessibility. https://www.w3.org/standards/webdesign/accessibility. Accessed 1 Aug 2018

HIIS Lab/ISTI-CNR (2018) MAUVE multiguideline accessibility and usability validation environment. http://mauve.isti.cnr.it/. Accessed 1 Aug 2018

Hong JI, Heer J, Waterson S, Landay JA (2001) WebQuilt: a proxy-based approach to remote web usability testing. ACM Trans Inf Syst 19(3):263–285. https://doi.org/10.1145/502115.502118

Hurst A, Hudson S E, Mankoff J, Trewin S (2008) Automatically detecting pointing performance. In: Proceedings of the 13th international conference on intelligent user interfaces. ACM, New York, pp 11–19. https://doi.org/10.1145/1378773.1378776

IBM (2017) IBM accessibility checklist. https://www.ibm.com/able/guidelines/ci162/accessibility_checklist.html. Accessed 1 Aug 2018

Ivory MY, Hearst MA (2001) The state of the art in automating usability evaluation of user interfaces. ACM Comput Surv (CSUR) 33(4):470–516. https://doi.org/10.1145/503112.503114

Kacmarcik G, Leithead T (eds) (2018) W3C UI events. https://www.w3.org/TR/uievents/. Accessed 1 Aug 2018
Keates S, Hwang F, Langdon P, Clarkson P J, Robinson P (2002) Cursor measures for motion-impaired computer users. In Proceedings of the 5th international ACM conference on assistive technologies. ACM, New York, pp 135–142. https://doi.org/10.1145/638249.638274
Kirkpatrick A, O'Connor J, Campbell A, Cooper M (2018) WAI/W3C web content accessibility guidelines (WCAG) 21. https://www.w3.org/TR/WCAG21/. Accessed 1 Aug 2018
Loop[11] (2018) Loop[11]. http://www.loop11.com/. Accessed 1 Aug 2018
Luque V, Delgado C, Gaedke M, Nussbaumer M (2005) Web composition with WCAG in mind. In: Proceedings of the 2005 international cross-disciplinary workshop on web accessibility (W4A). ACM, New York, pp 38–45. https://doi.org/10.1145/1061811.1061819
MacKenzie I S, Kauppinen T, Silfverberg M (2001) Accuracy measures for evaluating computer pointing devices. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, New York, pp 9–16. https://doi.org/10.1145/365024.365028
Mariage C, Vanderdonckt J, Pribeanu C (2005) State of the art of web usability guidelines. In: Proctor R, Vu K (eds) The handbook of human factors in web design. Erlbaum, Lawrence, pp 688–700
Paganelli L, Paternò F (2002) Intelligent analysis of user interactions with web applications. In: Proceedings of the 7th international conference on intelligent user interfaces. ACM, New York, pp 111–118. https://doi.org/10.1145/502716.502735
Paternò F, Schiavone AG (2015) The role of tool support in public policies and accessibility. Interactions 22(3):60–63. https://doi.org/10.1145/2745395
Petrie H, Hamilton F, King N, Pavan P (2006) Remote usability evaluations with disabled people. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, New York, pp 1133–1141. https://doi.org/10.1145/1124772.1124942
Schiavone AG, Paternò F (2015) An extensible environment for guideline-based accessibility evaluation of dynamic Web applications. Univ Access Inf Soc 14(1):111–132. https://doi.org/10.1007/s10209-014-0399-3
Scholtz J, Laskowski S, Downey L (1998) Developing usability tools and techniques for designing and testing websites. In: Proceedings of the 4th conference on human factors and the web, Basking Ridge, New Jersey, 5 June 1998
Section 508 (2018) Website policies. https://www.section508.gov/content/accessibility. Accessed 1 Aug 2018
Song S, Bu J, Wang Y, Yu Z, Artmeier A, Dai L, Wang C (2018) Web accessibility evaluation in a crowdsourcing-based system with expertise-based decision strategy. In: Pearson E, Sorge V (eds) W4A, ACM, pp 23:1–23:4. ISBN: 978-1-4503-5651-0
Takagi H, Asakawa C, Fukuda K, Maeda J (2002) Site-wide annotation: reconstructing existing pages to be accessible. In: Proceedings of the fifth international ACM conference on assistive technologies. ACM, New York, pp 81–88. https://doi.org/10.1145/638249.638265
Takagi H, Itoh T, Kawanaka S, Kobayashi M, Asakawa C. (2008) Social accessibility: achieving accessibility through collaborative metadata authoring. In: Proceedings of the 10th international ACM SIGACCESS conference on computers and accessibility. ACM, New York, pp 193–200. https://doi.org/10.1145/1414471.1414507
TECED (2017) Accessibility evaluation methodology. http://teced.com/services/web-accessibility/accessibility-evaluation-methodology/. Accessed 1 Aug 2018
TechSmith (2018) Morae. http://www.techsmith.com/moraehtml. Accessed 1 Aug 2018
Tenon (2018) LLC simplify your accessibility. https://tenon.io/. Accessed 1 Aug 2018
Thatcher J, Burks MR, Heilmann C, Henry SL, Kirkpatrick A, Lauke PH, Lawson B, Regan B, Rutter R, Urban M, Waddell CD (2006) Web accessibility: web standards regulatory compliance. Apress, New York
Valencia X (2017) A web transcoding framework based on user behaviour evaluation. Dissertation, University of the Basque Country

Valencia X, Pérez JE, Muñoz U, Arrue M, Abascal J (2015) Assisted interaction data analysis of web-based user studies. In: Abascal J, Barbosa S, Fetter M, Gross T, Palanque P, Winckler M (eds) Human-computer interaction—INTERACT 2015. Lecture notes in computer science, vol 9296. Springer, Cham, pp 1–19

Vanderdonckt J (1999) Development milestones towards a tool for working with guidelines. Interact Comput. Elsevier 12(2):81–118. https://doi.org/10.1016/S0953-5438(99)00019-3

Velleman E, Abou-Zahra S (eds) (2014) W3C/WAI Eval TF WAI/W3C website accessibility conformance evaluation methodology (WCAG-EM) 10. https://www.w3.org/TR/WCAG-EM/. Accessed 1 Aug 2018

Vigo M, Brown J, Conway V (2013) Benchmarking web accessibility evaluation tools: measuring the harm of sole reliance on automated tests. In: Proceedings of the 10th international cross-disciplinary conference on web accessibility. ACM, New York, pp 1–10. https://doi.org/10.1145/2461121.2461124

Vigo M, Arrue M, Brajnik G, Lomuscio R, Abascal J (2007a) Quantitative metrics for measuring web accessibility. In: Proceedings of the 2007 international cross-disciplinary conference on web accessibility. ACM, New York, pp 99–107. https://doi.org/10.1145/1243441.1243465

Vigo M, Brajnik G (2011) Automatic web accessibility metrics: where we are and where we can go. Interact Comput 23(2):137–155. https://doi.org/10.1016/j.intcom.2011.01.001

Vigo M, Harper S (2017) Real-time detection of navigation problems on the World 'Wild' Web. Int J Hum-Comput Stud 101:1–9. https://doi.org/10.1016/j.ijhcs.2016.12.002

Vigo M, Kobsa, A, Arrue M, Abascal J (2007b) User-tailored web accessibility evaluations. In: Proceedings of the 18th conference on hypertext and hypermedia (HT '07). ACM, New York, pp 95–104. https://doi.org/10.1145/1286240.1286267

WEBaccessibility (2018) Continuous accessibility testing. https://webaccessibility.com/. Accessed 1 Aug 2018

WebAIM (2018) WAVE web accessibility evaluation tool. http://wave.webaim.org/. Accessed 1 Aug 2018

Yesilada Y, Brajnik G, Vigo M, Harper S (2012) Understanding web accessibility and its drivers. In: Proceedings of the international of the 2012 cross-disciplinary conference on web accessibility. ACM, New York, pp 1–9. https://doi.org/10.1145/2207016.2207027