Social Accessibility: A Collaborative Approach For Improving Web Accessibility

Yevgen Borodin[‡]
borodin@cs.sunysb.edu

Shinya Kawanaka[†]
shinyak@jp.ibm.com

Hironobu Takagi[†]
takagih@jp.ibm.com

Masatomo Kobayashi[†]
mstm@jp.ibm.com

Takashi Itoh[†]
jl03313@jp.ibm.com

Chieko Asakawa[†]
chie@jp.ibm.com

[‡]Dept. of Computer Science, Stony Brook University, Stony Brook, NY 11794, USA

[†]IBM Tokyo Research Laboratory, 1623-14 Shimotsuruma, Yamato, Kanagawa, 242-8502, Japan

ABSTRACT

This chapter challenges the assumption that website owners are the ones responsible for the accessibility of web content. Web designers and developers have been notorious for not following the official accessibility guidelines. At the same time, the amount of user-generated web content makes it practically impossible to ensure web accessibility in a centralized fashion. However, the popularity of social computing opened the venue for collaborative approaches. This chapter overviews the applications of social computing to web accessibility and introduces Social Accessibility – a collaborative framework that brings together end-users and volunteers to create external accessibility metadata. In making the Web accessible, the Social Accessibility approach bypasses content owners, thus, considerably reducing the time for accessibility renovations. In addition, the centralized metadata can be used to educate web designers and developers in how to create accessible content, while providing a central point for collaborative accessibility verification.

INTRODUCTION

The Web is playing an important role in our lives, as it has become an infrastructure vital to our society. However, in its evolution from single-author text-based web pages to interactive web applications with user-generated content, the Web has become less accessible to people with vision impairments due to the carelessness of the content providers and the use of a wide variety of web technologies focusing primarily on improving visual interaction.

Web content has been traditionally under the control of the site owners, and, therefore, according to the present view on Web accessibility, site owners should be the ones bearing the responsibility of making their content accessible. Nowadays, however, the content is exceedingly generated by end-users, who are posting it using content-sharing services, such as forums, blogs, etc., in the volume which can be hardly controlled by the site owners.

Highly interactive interfaces, built with technologies such as AJAX and Flash, further exacerbate the accessibility problems. While interactive web sites can enhance the user experience by offering rich interactivity and responsiveness of the web application, they pose serious challenges not only to assistive software such as screen-readers [13] used by blind people, but also to spiders crawling and indexing the Web, software tools that help users aggregate and filter information, provide custom views [25], automate repetitive tasks [24, 26], etc.

From the compliance perspective, web designers and developers have to embed sufficient accessibility metadata into their content. For example, alternative text is required for screen-reader users, and structural metadata (e.g., indicating headings and lists) is key to making content navigable for users with visual and other impairments. Unfortunately, the accessibility metadata is often inadequate in both quality and quantity. Site owners are not able to give higher priority to ensuring the accessibility of their websites than to keeping up with their business and technology trends; hence, visual attractiveness of websites remains their primary focus.

Even when the site owners are willing to make their sites compliant with accessibility guidelines, making the websites fully accessible requires specialized knowledge. At the same time, only the end-users can reliably assess the usability and accessibility of web sites. However, user involvement in improving web accessibility is currently very limited. The general consensus among users is that reporting problems to site owners is of limited utility and no effective feedback loop exists to correct accessibility problems. There is, therefore, a clear need for a new framework that could involve the end-users and accelerate the accessibility renovations of web sites.

Recent years have seen a surge in social networks (e.g., Facebook, MySpace, LinkedIn), which have proven effective at bringing together users with common interests. Social networks, in turn, made possible a variety of collaborative approaches, such as ManyEyes [27], ESP game [28], to name a few.

The Social Accessibility project [29], featured in this chapter, is taking a similar approach – it applies social computing strategies to enable accessibility metadata authoring because a collaborative approach can drastically reduce the burden on site owners, while shortening the creation time for accessible Web content. Collaboration through a social network allows end-users to report accessibility problems, while any web users can create metadata both manually and automatically. The social networking infrastructure facilitates discussions and brings together people from around the world.

BACKGROUND

The Social Accessibility (SA) Project enables collaborative authoring of accessibility metadata, which is the extra information added to the original documents to make them more accessible. The use of metadata in improving the Web accessibility is very broad and is covered by several W3C guidelines and standards. Some representative examples include alternative text describing images, labels for form elements; and ARIA markup indicating the semantic roles of dynamic content. The important feature of the accessibility metadata is that it can be used by wide range of software tools from with screen readers to search engines.

In general, there are two types of metadata: internal metadata that is embedded into documents (web pages) and external (stand-off) metadata that is stored separately (but associated with the original documents). The important distinction is that the internal metadata can only be authored with the appropriate permissions, while the external metadata can be created by anybody and does not require the involvement of content owners.

The main challenge in using the external metadata is in the on-the-fly association of specific metadata with the content it describes. Anchoring metadata to a specific part of a document is, therefore, the key to effective use of external metadata. Various research projects have focused on automatic or semi-automatic creation and adaptation of external metadata to improve accessibility through transcoding the original documents.

Transcoding is often used to modify the presentation of content without modifying the originals. Transcoding for Web accessibility is a category of approaches that make existing web pages accessible on the fly. The technology is still not widely used, in spite of its huge potential to improve the accessibility of web content, primarily due to the workload of metadata authoring, which has not been manageable until the introduction of collaborative approaches.

The SA project uses a metadata infrastructure, *Accessibility Commons (AC)*, whose goal is to integrate and share the metadata produced by various research projects, assistive technologies, and individuals. The AC seeks to enable this needed and ambitious goal through a flexible schema for representing a common metadata repository and a method for integrating metadata of disparate types.

Metadata Authoring Approaches

Transcoding with external metadata has great potential as a new approach for creating a more accessible Web environment by supplementing the insufficient internal metadata. However, the workload of authoring has prevented it from providing major real-world benefits to users. We classify the approaches to reduce the authoring time and effort as follows.

Fully-Automated Generation

Automatic transcoding techniques can transform content without any additional information by using various inference techniques such as content analysis [20], differential analysis [21], and so on. These automatic methods have an advantage in coverage, since they can deal with any content on the Web, but the accuracy of their inferences can be problematic. Mechanisms to add supplementary manual metadata are needed for practical deployments. WebInsight [4] is an example of this approach. The system infers alternative texts for images by automatically combining the results of OCR with text-based content analysis and human-authored metadata. The system is also characterized by its use of manual metadata as a last resort after exhaustive automatic processing.

Semi-Automated Authoring

Some types of annotations are difficult to create by using fully-automated approaches, e.g., states of Rich Internet Applications (RIAs). In the traditional static-web paradigm, each page represents a state reachable through static links easily identifiable in the HTML source code. On the other hand, in RIAs the states are implicit and are determined by the user actions and the ensuing changes that occur in web pages as a result of those actions. The discovery of the states, transitions, and the information hidden in those states can improve RIA accessibility to web spiders, screen readers, and other tools that need to retrieve the information. Fully-automated approaches for discovering such states are not feasible [31], however, a semi-automated approach guided by users (even without them realizing it) can be used to create and share external metadata describing dynamic content and its behavior. The collaborative crawling approach can be used to automate the discovery of dynamic content and metadata authoring.

Manual Annotations

The knowledge obtained by users from explorations of complicated web content can be a source of metadata. For example, blind users can find the starting position of the main content in a page by exploring the page and marking this position for other users. Some commercial screen readers have functions to register alternative texts for images (e.g., JAWS [13]). Hearsay [6][20] has more advanced functions to allow users to add metadata (labels) in combination with an automatic analysis function. Users can easily select an appropriate label from the candidates. End-user annotation, although more accurate than automated annotation, is time consuming.

Improvement of Centralized Authoring (Template Matching)
Site-wide Annotation [22] aimed to reduce the workload by combining template matching algorithms and a metadata management tool called Site Pattern Analyzer (SPA). A snapshot of a target site would be crawled by the tool in advance, and then the tool visualizes the correspondences of each item of metadata with each page on the screen. This mechanism allowed creating the metadata for an entire newspaper site in 30 hours. In spite of the improvements, the workload for metadata maintenance was still excessive and prevented adoption by the site owners as a practical way of making their rapidly evolving content accessible.

Improvement of Centralized Authoring (Styling Information)
SADIe [12] is characterized by its annotation mechanism based on CSS (Cascading Style Sheet) information. One of the well-established trends in Web design is CSS-based styling, since it provides flexibility in design, reduces the cost of managing visual layouts, and even improves accessibility by separating the logical structure of the content from the design of the page. This system takes advantage of that trend to reduce the workload of metadata authoring by associating semantics with the styling components. The main limitation in applying the technique is that it only supports sites with well-organized styling information. Pages on the site should have logical structures and the styling units (such as headers and navigation bars) should be sufficiently logical to segment each page.

RELATED WORK
The work related to the Social Accessibility includes collaborative authoring techniques, transcoding, database integration, accessibility of Rich Internet Applications, and numerous accessibility research projects and products that generate and use metadata, as well as projects in other domains that deal with the problem of data integration among applications.

Improving Web Accessibility

Collaborative Authoring
Collaborative document authoring is an area with a long history (e.g. [15]). The largest success in this area is the wiki [16], and this technology has yielded such fruits of global collaboration as the Wikipedia. In spite of the successes of collaborative authoring, it has rarely been applied in the accessibility area. One of the recent projects is for collaborative "caption" authoring of multimedia content. The We-LCoME project is aimed at building accessible multimedia e-learning content through collaborative work on a wiki system [9, 10]. We-LCoME and Social Accessibility run in similar directions, using collaborative authoring for accessibility. Another example is the Google Image Labeler [11]. This is a system to build accurate textual descriptions of images through a game. The goal of the project is to improve the accuracy of Google Image search, but the generated metadata could potentially be used for accessibility.

Transcoding
Transcoding for web pages originally developed to adapt web pages for mobile devices [3] and to personalize pages [18]. Then, the technique was applied to transform inaccessible Web content into accessible content on the fly, forming a new category of technology, "Transcoding for Web accessibility". [1] is a survey article including history and methods. Transformation techniques can be divided into two major types, one for automatic transcoding and the other for metadata-based (or annotation-based) transcoding. Automatic methods have clear accuracy limitations, and therefore external metadata is needed for usably accessible transformation results, especially for people with severe disabilities, such as blindness. However, the external metadata approach has problems with metadata authoring.

A recent research challenge in the transcoding area is dynamic Web applications including AJAX techniques. The aiBrowser has a metadata mechanism to dynamically convert AJAX and Flash-based dynamic content into accessible formats [19]. AxsJAX [7] is a technology to make AJAX applications accessible by using JavaScript descriptions as a kind of metadata. Access Monkey [5] also uses JavaScript to transcode content.

Database Integration
The Web domain and the Life Science domain are two of the most active domains in integrating databases. Since these domains have many resources to handle (such as web pages or genomes) and since those resources are often stored separately for each project, there are strong demands for data integration and data exchange.

The Semantic Web (http://www.w3.org/2001/sw) is a project initiative of the W3C to integrate and exchange Web resources. Web developers can use metadata to specify titles, publishers, meanings, and other semantic roles. The metadata is described in a Resource Description Framework (RDF - http://www.w3.org/RDF) or using the Web Ontology Language (OWL - www.w3.org/TR/owl-features). By adding such metadata, applications handling RDF or OWL can interpret the meaning of Web resources, and they can also handle resources with similar meanings.

For example, if two online banking websites have the same metadata, one application can use them equally well even though they may use different visual layouts or structures. Since the metadata is written in one format, it is not necessary to convert the data format, so the data exchange is relatively easy.

In the Life Science domain, integrating genome databases is an active area. YeastHub [32] is a project aiming to integrate many databases of yeast genomes. In the past, each yeast genome project has had its own database for storing its yeast genome data. Users can now search for yeast genome data in the YeastHub, and the results are tables or RDF that combines the data stored in the separate databases. However, since the data formats are usually unchanging and since the stored data is easy to convert, the data integration is relatively easy.

In contrast, Accessibility metadata does not have any fixed format. The formats vary from application to application, and can be tables, XML, scripts, etc. When integrating accessibility metadata, it is challenging to support the many formats.

Accessibility of Rich Internet Applications

Most RIAs are currently accessible only to users visually interacting with the dynamic content. If web developers properly exposed states and transitions of their websites, screen-readers, crawlers, and tools for information filtering [14] and automation [24, 26] would be able to interact with the rich content. Unfortunately, web applications are built with a variety of technologies and toolkits, many of which make RIA web sites partially or completely inaccessible. Until recently, there have been two disjoint efforts trying to improve the accessibility of dynamic content by either manual or automatic authoring of metadata.

Manual Approaches

The use of W3C standard for Accessible Rich Internet Applications (ARIA) [33] was one of the first attempts to make RIAs accessible. ARIA markup is intended to be used by screen-readers to improve accessibility of web applications to blind people. ARIA metadata can be embedded into web pages and can be used to describe live areas, roles, and states of dynamic content. Unfortunately, most of the dynamic content available today does not implement ARIA standard. Also, web developers are unlikely to follow ARIA consistently, for they have not followed other accessibility guidelines.

ARIA can be also supplied as part of reusable components or widgets; for example, Dojo Digit (http://dojotoolkit.org/projects/dijit) provides ARIA-enabled widgets and a toolkit to build custom accessible widgets. However, Digit is only one of many available toolkits, and web developers continue creating inaccessible custom widgets of their own. ARIA can also be applied through transcoding. To illustrate, Google's AxsJAX [7] allows web developers to use JavaScript to inject ARIA metadata into existing applications. However, AxsJAX scripts have had to be, so far, created manually.

Automated Approaches

To date, the only known approaches to automatic collection of information from web applications have been crawling RIA web sites statically or crawling RIAs by opening them in a web browser [28]. Regrettably, both of these approaches have certain limitations and cannot be used to make RIAs fully accessible.

The majority of search engines index RIAs by statically crawling web sites and extracting text from the HTML source code. With such crawling, one cannot effectively infer the implicit state model of the web site. The results of indexing can be enhanced by content providers explicitly exposing textual data to web spiders, e.g. through meta-tags. However, content providers are not always aware of how to properly use meta-tags to make content accessible to web crawlers.

An alternative to the static crawling can be opening RIAs in an embedded web browser and simulating various user events on all objects to expose the resulting system events and hidden content. For instance, AJAX applications crawling is described in [28, 34], where diff algorithms are used to detect the changes. Dynamic changes can also be identified by combining a diff algorithm with HTML DOM mutation event listeners, as described in [30]. Hypothetically, embedded crawling could automate metadata authoring. However, a crawler built with the embedded browsers often cannot access all content, and consumes substantial machine-time, while suffering from state explosion [28], irreversibility of actions (requiring that transitions be retraced from the start state), latency between actions and reactions (especially, in AJAX applications), and inability to access password-protected web sites.

Research Projects and Screen Readers

This section summarizes the metadata that is already in use by the existing accessibility research projects and products. A thorough understanding of the existing metadata helped to inform our decisions and strategy, and hopefully ensures the Accessibility Commons will remain relevant as new projects and products are developed.

aiBrowser

The aiBrowser [19] is a multimedia browser for visually impaired people. The browser transcodes HTML documents and Adobe Flash (www.adobe.com/products/flash) on the client side to provide alternate content that is more accessible for visually impaired people. The transcoding is done using metadata described in XML. The metadata describes how to combine HTML elements and Flash objects to generate more accessible alternate content. In the metadata, XPath expressions are used to specify HTML elements and Flash queries are used to specify Flash objects. In addition, the aiBrowser allows users to add manual annotations for headings and alternative text. If the aiBrowser were to use a common repository, it could share its metadata and user annotations to provide alternative text and heading tags to people using other technologies.

HearSay

The HearSay non-visual Web browser [6, 20, 35-37] uses various content analysis techniques to improve Web accessibility. Among them are: context-directed browsing for identification of relevant information in web pages [37], language detection [35], concept detection [36], etc. HearSay uses the results of the automated analyses to annotate Web content. For example, the context-directed browsing algorithm inserts a "start" label, instructing the browser to begin reading the page from a specific position. The HearSay browser has a VoiceXML-based dialog interface, which interprets the labels and provides facilities for navigating, editing, and creating manual labels. The labels can be stored in personal or shared repositories. The use of uniform metadata and a shared repository allows other applications to benefit from the labels created in HearSay. At the same time, future HearSay users will have access to metadata created by a wider pool of blind Web users.

WebInSight for Images

WebInSight for Images [4] provides alternative text for many Web images to improve their accessibility. To make this alternative text, WebInSight uses contextual analysis of linked web pages, enhanced Optical Character Recognition (OCR), and human labeling. The alternative text strings are stored in a shared database referenced by an MD5 hash of the image and the URL of the image. The stored alternative text is supplied as users browse the Web. When a user visits a webpage for the first time, WebInSight attempts to create alternative texts by doing contextual analysis and OCR. If these options fail, the user can request human labeling. By combining the alternative text into a common database, users will be more likely to experience the benefits.

Site-wide Annotation

Site-wide Annotation [22] is a research project to transcode entire websites by annotating them. The metadata of the site-wide Annotation uses XPath expressions. The system checks for elements matching the expressions and transcodes the web pages based on the metadata. This allows transcoding an entire website with a small set of metadata. If this metadata can be created and shared by users, a larger number of websites could be transcoded for better Web accessibility.

AxsJAX

AxsJAX [7] is an accessibility framework to inject accessibility support into Web 2.0 applications. Currently the main targets of AxsJAX are Google applications such as GMail and Google Docs. AxsJAX scripts use Greasemonkey, a bookmarklet, or run directly in Fire Vox [38], a screen reader implemented as a Firefox Extension. AxsJAX identifies elements to which ARIA markup should be provided using XPath. Currently, these associations are distributed to users in the form of scripts. More tools could benefit from the semantic knowledge encoded in these scripts if they were stored in a more flexible and semantically-accessible common repository.

Accessmonkey

Accessmonkey [5] is another common scripting framework that Web users and developers can use to collaboratively improve Web accessibility. The goal is to enable both Web users and developers to write scripts that can then be used to improve the accessibility of web pages for blind Web users. An example script provided by the Accessmonkey demonstrates how WebInSight for images can be implemented in this framework in order to provide alternative text for Web users as they browse and to suggest alternative text for Web developers trying to improve their pages.

Structural Semantics for Accessibility and Device Independence (SADIe)

SADIe [12] is a proxy-based tool for transcoding entire websites as opposed to individual pages. It relies on ontological annotations of the Cascading Style Sheet (CSS) to broadly apply accurate and scalable transcoding algorithms. Only by explicitly enunciating the implicit semantics of the visual page structure (groups, components, typographic cues, etc.) can we enable machine understanding of the designers' original intentions. These intentions are important if we wish to provide a similar experience to visually impaired users as to fully sighted users. SADIe can be regarded as a tool for the site-wide reverse engineering of web pages to achieve design rediscovery [39].

JAWS

JAWS is one of the most popular screen readers. It has a labeling feature, which allows users to provide alternative text for images or flash objects. The latest version of JAWS can make use of WAI ARIA [33], a World Wide Web Consortium (W3C) internal metadata standard, to improve the accessibility of dynamic content.

SOCIAL ACCESSIBILITY

Social Accessibility is collaborative framework for making existing content accessible by using the power of the open community. In the current framework, developers have the primary responsibility to make content accessible by embedding accessibility metadata into the content. There is no systematic feedback loop from users to developers, even though only the users have the ability to assess the real usability. The Social Accessibility approach changes the landscape by welcoming the open community as authors of external accessibility metadata.

Different users with a variety of accessibility needs can participate in Social Accessibility by reporting their evaluations of the usability of content. Any open community member (any Web user) can help make any content or service accessible through collaboration with other community members. Whenever a user reports difficulties in some content, the volunteers can discuss, create, and publish of the accessibility metadata for all users who face the same problem. For the website owners and developers, the reported issues can be regarded as the results of volunteer-based global usability testing by real users of the site. The created metadata can also be regarded as volunteer-based consulting for accessibility improvements. In other words, the goal is to make a system of collective intelligence for end users, volunteers, site owners, and everyone who has an interest in the accessibility of the Web.

The basic principle of the approach is that anyone, developers, users, or even open community members, will be able to improve the accessibility of any content on the Internet by collaboratively authoring the accessibility metadata. This approach is a combination of Web accessibility technology (external metadata) and social computing strategy (collaborative authoring). The collaborative authoring is a method to build tangible knowledge presentations among a group of people. This approach will fill the missing link of external metadata by applying collaborative authoring methods.

A number of technical challenges have to be overcome in order to enable collaboration. It is critical to design usable authoring tools and collaboration services. The authoring tools should be usable enough to allow non-technical and accessibility-novice volunteers to join in. Collaboration services should help participants to work together and provide motivation for contributing to the activity. Also the design of metadata will define the flexibility of the parallel authoring. The accuracy and generality of metadata are also important technical challenges.

The SA framework can also support the ARIA-style markup and provide the interface for manual labeling of: live areas, relations between web objects, object roles, etc. The ARIA-style markup will allow applications, such as crawlers and screen-readers, to identify and correctly handle dynamic content, as well as identify states and transitions in RIA applications. Although the use of a shared repository can facilitate manual metadata authoring, a scalable automated approach, such as collaborative crawling, can offer significant help to both Social Accessibility users and volunteers.

COLLABORATIVE CRAWLING

Instead of invoking all possible actions on web content to discover dynamic content and states of RIAs, the collaborative crawling approach delegates this to computer users, with the expectation that, eventually, the users will discover all allowable actions (transitions) and experience all possible system reactions. This approach allows volunteers to create ARIA metadata while performing their regular browsing activities, acting in a way as “distributed spiders,” crawling the Web and discovering dynamic content.

By analyzing user actions and system reactions on a given web page, it is possible to automatically infer ARIA metadata for live areas (e.g., dynamic stock ticker) and actionable objects (e.g., draggable), identify relationships between objects (e.g., hover the mouse to open menu), and even infer element roles (e.g., slider). Observing multiple users performing the same action will only improve inference confidence.

The derived metadata can be then shared through SA and used by other applications, e.g., web spiders to intelligently crawl RIA sites. To avoid possible violation of privacy, the database may store only website addresses and the locations of objects within web pages, using a variety of addressing schemes, such as XPath, URI, etc., as discussed in [14]. However, since collaborative crawling can be anonymous, in some scenarios with no obvious security threats, e.g., for unsecured websites in the public domain, the dynamically changing content can also be committed to the database for further indexing.

DISCUSSION

Conflicts and Broken Metadata

The metadata accumulated in AC database may accrue conflicts. For example, two different versions of alternative text may be supplied for the same image. Also, the repository may contain broken metadata, e.g., due to changes in the target web pages. Currently the SA infrastructure does not detect any conflicts or broken metadata, returning all metadata corresponding to the query and leaving metadata filtering to the client. The techniques need to be developed for discarding or fixing the broken metadata. With a large number of users, metadata errors can be quickly identified and reported.

As for metadata conflicts, techniques have to be developed for metadata ranking and filtering either on the client or server side. Performing filtering on the client side allows the client applications to choose the appropriate strategies based on user preferences, context analysis, etc. In addition, since this server provides the metadata, a client can determine who created the metadata, and, if a reputation system is available, then the client can choose the metadata supplied by the most reliable author. Alternatively, if a client knows which author is an automatic analysis engine, the client can give such metadata lower preference relative to human-authored metadata. Also, the client can choose the latest metadata, which may fit the current page. Of course, these kinds of information can be provided with this infrastructure.

Spam Metadata

The possibility of spam attacks, submitting lots of meaningless or broken metadata, exists in this infrastructure. It may be possible to reduce the damage by introducing some protection mechanisms, such as a limited number of queries per second.

Performance of Database

Low latency performance of the database is crucial for the infrastructure. It is expected that millions of metadata records will accumulate in the common repository, and this growth may negatively affect the performance of the repository. Users dislike unresponsive systems. We chose to use domain names as an index for the database. Therefore the performance may depend on how much metadata exists in a domain. For the Site-wide Annotation [22], 245 annotations files were used to transcode USAToday.com, yielding reasonable performance. Since the granularity of the AC metadata is finer than that of Site-wide Annotation system, a larger amount of metadata may be recorded in each domain. However, considering the improvements of hardware and networks, we believe that the infrastructure will be able to process user queries with acceptable response times. In addition, if the client caches metadata, the processing time can be drastically reduced, because the client can query the difference between cached metadata and the latest metadata. Also, since metadata queries can be processed in parallel, it is easy to enhance the infrastructure as required.

Necessary Skills for Metadata Authoring

A successful collaborative system should require minimal technical skills and contribution from its users, while providing maximum benefits. The current pilot system requires supporters to have minimal knowledge about accessibility, and they can easily learn about the tools and services. The collaborative crawling approach goes further, requiring no work on the part of the user whatsoever.

Implications for Site Owners

The system will reduce the burdens on site owners through the power of the community, but it does not mean they should ignore accessibility issues. The centralization of the metadata will allow the SA framework to encourage site owners to pay more attention to accessibility and hopefully renovate their sites to be more accessible. Site-renovation work is too often reduced to the task of fixing the errors reported by automatic accessibility checkers. SA system can change that by automatically organizing and delivering the accessibility information to content providers as a suggestion to be incorporated into websites as internal metadata. The user request process can be regarded as volunteer-based global usability testing sessions by real users. The products of the collaborative authoring process: metadata, discussions, and site-specific rules for metadata will be invaluable information for effective renovations by site owners. When they renovate their sites for greater accessibility, they can actually know exactly how and why the supporters fixed their pages.

Appropriateness of Collaboration Methods

Four types of collaboration tools are integrated into the SA pilot system: instant messaging, discussion threads, Wiki, and email. Among these tools, the most commonly used was instant messaging. One of the reasons is that the authoring process usually starts at the same time when it is triggered by a new user request. Periodically, they organized the result of discussions into Wiki pages for future reference. We believe that the importance of asynchronous collaboration will be increased as more supporters participate in the activities, especially when supporters worldwide start collaborating. We also found that metadata authoring requires consensus on the rules used in annotation. For example, heading levels for search results should be the same across a site, so supporters need to discuss which heading level will best fit with the surrounding information. According to these requirements, we are planning to integrate the collaboration methods more tightly by adding some automation functions.

Security and Privacy

When a user reports an error, a screen image of the browser and the reading position is automatically captured and sent to the server. This function is crucial for supporters to understand the problems faced by the user. However it creates security and privacy concerns. If a user reports an error in a page that is showing personal information, such as a personal profile or a bank account, the information would be disclosed to the supporters. To address this concern, some improvements are planned. For example, when a screen is captured, all of the input forms (text inputs, radio buttons, etc.) will be blacked out before submission to the server. It is also planned to block the capture of secure pages (using https).

Effectiveness of Incentives

We interviewed the participants and all of them agreed on the importance of the incentive system. They mentioned that the ranking of supporters on the portal page motivated them to remain active on the system. They also pointed out some unfairness in the point assignment scheme. For example, metadata with well considered wildcards can cover a large number of pages, but it is harder to create such metadata. As far as the points are concerned, that broadly useful metadata still counts as "one metadata item" in the current incentive scheme. Some other supporters commented that the most effective rewards are the appreciative comments from the end users. We are considering these points and discussing with the participants how to design a better evaluation mechanism.

CONCLUSION AND FUTURE WORK

In this chapter we discussed several collaborative approaches to improving web accessibility, the quality and quantity of accessibility metadata and limitations on user participation. In order to reduce the burden on site owners and shorten the time to improved accessibility, we introduced Social Accessibility – a framework that can make the Web more accessible by gathering the power of the open community. The approach is characterized by collaborative metadata authoring based on user requests. Any Web user with a disability can report their accessibility problems to the Social Accessibility service and any Web user can volunteer to fix the accessibility problems without modifying the original content. We also discussed the collaborative crawling approach that can improve accessibility of Rich Internet Applications for screen readers, web crawlers, and other software tools that need to interact with dynamic web content.

With the growing popularity of social computing, the Social Accessibility approach has the potential to grow into a worldwide collective intelligence for Web accessibility, and contribute to changing the access environments of users with disabilities worldwide.

REFERENCES

[1] Asakawa, C. and Takagi, H. 2008. Transcoding. In *Web Accessibility: A Foundation for Research*, S. Harper and Y. Yesilada, Ed. Human-Computer Interaction Series. Springer-Verlag.

[2] Berry, L., Bartram, L., and Booth, K. S. 2005. Role-based control of shared application views. In *Proc. the 18th Annual ACM Symposium on User interface Software and Technology*. UIST '05, 23-32.

[3] Bickmore, T. W. and Schilit, B. N. 1997. Digestor: Device-independent access to the World-Wide Web. In Proc. the 6th Int. World-Wide Web Conf, 655-663.

[4] Bigham, J. P., Kaminsky, R. S., Ladner, R. E., Danielsson, O. M., and Hempton, G. L. 2006. WebInSight: making Web images accessible. In *Proc. the 8th int. ACM SIGACCESS Conf. on Computers and Accessibility*. Assets '06, 181-188.

[5] Bigham, J. P. and Ladner, R. E. 2007. Accessmonkey: a collaborative scripting framework for Web users and developers. In *Proc. the 2007 int. Cross-Disciplinary Conf. on Web Accessibility (W4a)*. W4A '07, vol. 225, 25-34.

[6] Borodin, Y., Mahmud, J., Ramakrishnan, I. V., and Stent, A. 2007. The HearSay non-visual Web browser. In *Proc. the 2007 int. Cross-Disciplinary Conf. on Web Accessibility (W4a)*. W4A '07, vol. 225, 128-129.

[7] Chen, C., and Raman, T. V. 2008. AxsJAX: A Talking Translation Bot Using Google IM. In *Proc. the 2008 int. Cross-Disciplinary Conf. on Web Accessibility (W4a)*. W4A '08, 105-107.

[8] Farzan, R., DiMicco, J., Millen, D. R., Dugan, C., Geyer, W., and Brownholtz, E. 2008. Results from deploying a participation incentive mechanism within the enterprise. In *Proceeding of the Twenty-Sixth Annual SIGCHI Conf. on Human Factors in Computing Systems*. CHI '08, 563-572.

[9] Ferretti, S., Mirri, S., Roccetti, M. and Salomoni, P. 2007. Notes for a Collaboration: On the Design of a Wiki-type Educational Video Lecture Annotation System. In *Proc. the IEEE Int. Workshop on Semantic Computing and Multimedia Systems (IEEESCMS' 07)* IEEE Computer Society, 651-656.

[10] Ferretti, S., Mirri, S., Muratori, L. A., Roccetti, M., and Salomoni, P. 2008. E-learning 2.0: you are We-LCoME!. In *Proc. the 2008 int. Cross-Disciplinary Conf. on Web Accessibility (W4a)*. W4A '08, 116-125.

[11] Google Image Labeler; see http://images.google.com/imagelabeler/

[12] Harper, S., Bechhofer, S., and Lunn, D. 2006. SADIe:: transcoding based on CSS. In *Proc. the 8th int. ACM SIGACCESS Conf. on Computers and Accessibility*. Assets '06, 259-260.

[13] JAWS, Freedom Scientific Inc.; see http://www.freedomscientific.com/

[14] Kawanaka, S., Borodin, Y., Bigham, J. P., Lunn, D., Takagi, H., and Asakawa, C. 2008. Accessibility Commons: A Metadata Infrastructure for Web Accessibility, In *Proc. the Tenth Int. ACM SIGACCESS Conf. on Computers and Accessibility.* Assets '08.

[15] Leland, M. D., Fish, R. S., and Kraut, R. E. 1988. Collaborative document production using quilt. In *Proc. the 1988 ACM Conf. on Computer-Supported Cooperative Work*. CSCW '88, 206-215.

[16] Leuf, B., and Cunningham, W. 2001. The Wiki Way: Quick Collaboration on the Web. Addison-Wesley Professional.

[17] Lui, S., Lang, K., and Kwok, S. 2002. Participation Incentive Mechanisms in Peer-to-Peer Subscription Systems. In *Proc. the 35th Annual Hawaii Int. Conf. on System Sciences (Hicss'02)*-Volume 9 (January 07 - 10, 2002). HICSS. IEEE Computer Society, 302.2.

[18] Maglio, P. and Barrett, R. 2000. Intermediaries personalize information streams. *Commun. ACM* 43, 8 (Aug. 2000), 96-101.

[19] Miyashita, H., Sato, D., Takagi, H., and Asakawa, C. 2007. aiBrowser for Multimedia - Introducing Multimedia Content Accessibility for Visually Impaired Users, In *Proc. the Ninth Int. ACM SIGACCESS Conf. on Computers and Accessibility.* Assets '07, 91-98.

[20] Ramakrishnan, I. V., Stent, A., and Yang, G. 2004. Hearsay: enabling audio browsing on hypertext content. In *Proc. the 13th int. Conf. on World Wide Web*. WWW '04, 80-89.

[21] Takagi, H. and Asakawa, C. 2000. Transcoding proxy for nonvisual Web access. In *Proc. the Fourth int. ACM Conf. on Assistive Technologies*. Assets '00, 164-171.

[22] Takagi, H., Asakawa, C., Fukuda, K., and Maeda, J. (2002) Site-wide annotation: reconstructing existing

pages to be accessible. In *Proc. the Fifth int. ACM Conf. on Assistive Technologies*. Assets '02, 81-88.
[23] XML Path Language; see http://www.w3.org/TR/xpath
[24] Gilly, L., Harber, E., Matthews, T., Lau, T., CoScripter: Automating & Sharing How-To Knowledge in the Enterprise, *in SIGCHI conference on Human factors in computing systems. 2008*, ACM: Florence, Italy.
[25] Nichols, J., Lau, T. Mobilization by Demonstration: Using Traces to Re-author Existing Web Sites. *in IUI 2008*, Gran Canaria, Spain
[26] Borodin. Automation of Repetitive Web Browsing Tasks with Voice-Enabled Macros. ASSETS 2008.
[27] F. B. Viegas, M. Wattenberg, F. v. Ham, J. Kriss, and M. McKeon. Many eyes: A site for visualization at internet scale. *Proceedings of Infovis*, 2007.
[28] Von Ahn, L., And Dabbish, L. 2004. Labeling images with a computer game. In *Proc. Conf. on Human Factors in Computing Systems*, 319.326.
[29] Takagi, H., S. Kawanaka, and M. Kobayashi. Social Accessibility: Achieving Accessibility through Collaborative Metadata Authoring. *in Proceedings of ASSETS. 2008*, ACM: Halifax, Canada.
[30] Borodin, Y., Bigham, J., Raman, R., Ramakrishnan, I.V., What's New? – Making Web Page Updates Accessible. in *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*. *2008*, ACM: Nova Scotia, Canada
[31] Ali Mesbah, Engin Bozdag, and Arie van Deursen., Crawling AJAX by Inferring User Interface State Changes. *2008, ICWE*, Yorktown Heights, NY, USA
[32] K. H. Cheung, K. Y. Yip, A. Smith, R. deKnikker, A. Masiar, and M. Gerstein. YeastHub: a semantic web use case for integrating data in the life sciences domain. In Bioinformatics 21, 1 (Jan. 2005), 85-96. 2005.
[33] WAI ARIA. http://www.w3.org/TR/wai-aria/
[34] Frey, Gianni. Indexing AJAX Web Applications. ETH Department of Computer Science, Institute of Computational Sciences (2007). Available from: http://e-collection.ethbib.ethz.ch/view/eth:30111
[35] Y. Borodin, et. al. Towards One World Web with HearSay3. In W4A'08: Proceedings of the 2008 international Cross-Disciplinary Conference on Web Accessibility, 128-129. 2008.
[36] J. U. Mahmud, Y. Borodin, and I. V. Ramakrishnan. Assistive Browser for Conducting Web Transactions. In IUI '08: Proceedings of the International Conference on. Intelligent User Interface. 2008.
[37] J. U. Mahmud, Y. Borodin, and I. V. Ramakrishnan. Csurf: a context-driven non-visual web-browser. In WWW '07: Proceedings of the 16th International Conference on World Wide Web, 31-40. 2007.
[38] Fire Vox: A Screen Reading Extension for Firefox http://firevox.clcworld.net/.
[39] E. J. Chikofsky and J. H. C. II. Reverse engineering and design recovery: A taxonomy. In IEEE Software, 13-17. 1990.