

From Assistive Technology to a Web Accessibility Service

Peter G. Fairweather

IBM T.J. Watson Research Center
Route 134
Yorktown Heights, NY 10598 USA
914 945 2138
peterf@watson.ibm.com

Vicki L. Hanson

IBM T.J. Watson Research Center
19 Skyline Drive
Hawthorne, NY 10532 USA
914 784 6603
vlh@watson.ibm.com

Sam R. Detweiler

Richard S. Schwerdtfeger
IBM Accessibility Center
11400 Burnet Road
Austin, TX 78758
512 838 2248
{sdetweil, schwer}@us.ibm.com

ABSTRACT

This paper considers different ways to enhance access to the World Wide Web for persons with sensory, cognitive, or motor limitations. Paradoxically, while complex Web architectures may seem to have inhibited accessibility, they have broadened the range of points where we can try to improve it. This paper identifies these points and evaluates the advantages and disadvantages of each. In particular, it describes a project to develop a strategy to enhance access that can be distributed across multiple control points and implemented as an aggregation of Web services.

Keywords

World Wide Web, adaptive interfaces, accessibility, Web services

INTRODUCTION

The World Wide Web provides an increasing proportion of the services that in many developed societies are necessary for full economic, cultural, and political participation. Recognition of this crucial role has fomented an insistent regulatory climate in many areas of the world that encourages and protects access to the Web by persons with disabilities [3]. The number of persons with disabilities trying to access the Web is rising, propelled mainly by the increasing proportion of the aged in populations of developed countries [10].

PROGRESS AND SEMANTIC OBSCURITY

Regulatory directives, such as "Section 508"[9] and Web design standards such as those promulgated by the W3C Web Accessibility Initiative (WAI)[12] work in concert to inform developers how to protect and enhance access to the Web. However, the increasingly rapid evolution of Web technologies makes it difficult for such regulations and standards to keep pace.

At its beginnings, the currency of World Wide Web was the simple Hypertext Markup Language (HTML) document,

retrieved by a server from an underlying file system and sent to the requesting browser. Its two-layer architecture (see figure 1) and constrained media choices largely limited to text did little more to obscure document semantics than did earlier Internet tools such as Gopher or Archie, for example. Although the aim of HTML was not to enhance the semantic structure of the information it presented, as developers inventively and idiosyncratically applied its features, it became a significant contributor to semantic turbidity.

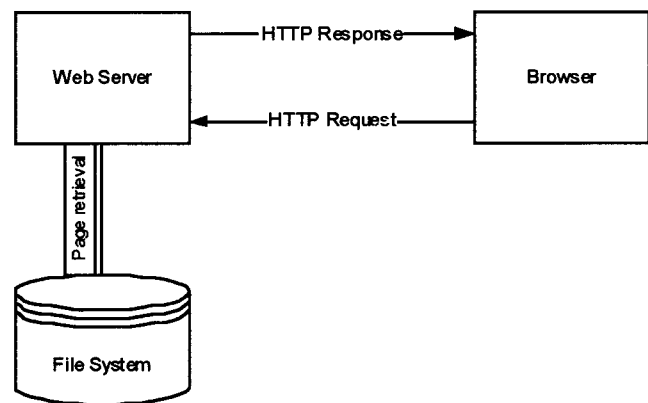


Figure 1 Simple page retrieval model from the early Web

Web architectures have diversified, supporting purposes that have moved beyond the sharing and exploration of data to include graphics and animation, video and audio, and personalized interactive rich media. With Web pages being generated by servlets, or server-side scripts, for example, or with style sheets directing their final rendering, the information models for Web applications have separated from the way they are presented (see Figure 2). This gap proves troublesome to assistive technologies trying to reconstruct an information model by operating at the user interface.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Assets 2002, July 8-10, 2002 Edinburgh, Scotland
Copyright 2002 ACM ISBN 1-58113-464-9-02/07 ...\$5.00

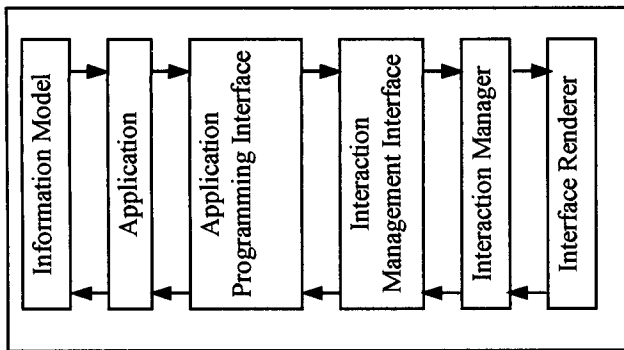


Figure 2: Example of multi-layer Web architecture

Indeed, Web developers increasingly build upon Web services, stretching even further the gap between the user experience at the interface and the underlying semantics. While increasing architectural complexity compromises access to the Web by obscuring the semantics of its information model, that same complexity may engender novel ways to enhance access by calling upon Web services.

ACCESSIBILITY CONTROL POINTS

Web pages are retrieved or assembled along a pipeline that runs from the information model resident on the server at one end to the rendering mechanisms that determine the user experience at the other (see Figure 2). A set of control points along this pipeline offer places where accessibility technologies can be applied. These are discussed in the next sections.

EXTERNAL ADAPTERS

External hardware or software that replaces or modifies customary inputs or outputs can enhance access to Web content. For example, the Archimedes Project's Total Access Processor [6] can be set up to transform a user's speech into a character stream presented to a Web page just as if it had been produced at the keyboard. Similarly, commercially available speech recognition systems typically offer a control mode for adapting other applications including those on the Web to speech input. The independence of the external adapter from the Web browser consigns it to relative ignorance about the browser's state and the information model used to render its content. For example, an output adapter attempting to transform Web content into an audio stream needs some information about the information model. Without it, the audio stream will be little more than a parroting of the organization of the visual user interface with few of the directives one normally hears in speech but rarely sees in text.

ADAPTIVE RENDERERS

Moving from outside of the Web browser to inside, we consider adaptations that can take place there. At the Web's beginnings, whatever representation the server sent to the browser largely defined user experience. Later, style sheets were introduced that allowed browsers to adapt Web content. For most Web pages, there is little semantic

information for style sheets to work on, so their scope is limited to directing the way information is presented. For example, although images serve a variety of purposes on Web pages, from illustration and backgrounds for text, to formatting spacers, they appear syntactically homogeneous, so a general Cascading Style Sheet (CSS) cannot treat them differently. For example, imagine an image on a Web page that illustrates a product and another that provides an area of dark background over which the product's name written in white text. Using a style sheet to remove all images might also make the product name invisible with its white text now showing over a white background. Because style sheets cannot analyze a page and infer semantics, their actions can produce unwanted side effects.

Web page readers, such as IBM Home Page Reader™ [6] illustrate another class of adaptive renderers that transform visual Web user interfaces into auditory ones. Operating in a semantically impoverished environment, they render content and navigation cues using the Document Object Model (DOM) of a Web page as well as some heuristics about how pages are organized (e.g., navigation bar links typically are on the upper left side or at the bottom of a page).

MULTIMODAL REPRESENTATIONS AND PROTOCOLS

Some recent proposals to enhance access advocate enriching and extending the information needed to manage an interaction with the end user. This information provides alternative ways for an interaction to be realized, depending on the user's preference. For example, VoiceXML [1] provides a means to create Web interactions with users who use voice-enabled browsers or telephones. It presages a set of rich XML-based protocols that can specify the information needed to adapt Web content to users' preferences and capabilities.

Coupled with such semantically rich representations is a set of comprehensive proposals for adaptive protocols built along the lines of the peer-to-peer, universal plug-and-play (UPnP) protocol. For example, the developing Alternative Interface Access Protocol (AIAP)[7] establishes communication between one control point dispensing an XML-based abstract representation of a document and another that transforms that representation into a concrete, accessible user interface on a particular device. The protocol will handle the discovery of control points and the negotiation of both device and user preferences and capabilities. Negotiated dynamic construction of user interfaces can be applied not only hardware devices, but to Web clients communicating with content servers.

COMPARING CONTROL POINTS

Most assistive technologies for the Web operate on the client device as external adapters or adaptive renderers at the opposite end of the pipeline from the information model. For example, a general-purpose software screen magnifier sees the screen as pixels that appear the same no

matter whether they form text or image. Similarly, a screen reader operating on HTML elements cannot recognize the cues in an online news site that could be used to organize how the material should be read aloud. Without heuristics, it follows an unchanging pattern, such as top to bottom, left to right.

Adapters tend to focus on a single disability, falling short of the needs of those with multiple limitations. For example, applications to support learning disabled readers have nothing except text-to-speech to support blind users, even though such users may have problems processing aural language. The software assumes that these users can see, in the same way that Web page readers assume that users can hear. Moving the accessibility control point away from the client opens the possibility of dynamically composing adapters to deal with a particular user's requirements instead of building a static set based on assumptions.

End users generally must purchase assistive technology for their own use and, even if it is subsidized, they usually must install and support it. Even in the workplace, those charged with maintaining information technology are most often unfamiliar with assistive hardware and software. Shifting the control point for adaptation away from the client practically ensures that the organization hosting the adaptation mechanism will bear the costs of its acquisition, installation, and support.

Neither current standards nor regulations require Web pages to be directly accessible. In fact, adapters are part of the definition of what it means for information technology to be accessible. For example, Web Accessibility Initiative recommends "[to] make programmatic elements such as scripts and applets directly accessible or compatible with assistive technologies." [2]. Following standards can only go part of the way to achieving directly accessible Web pages. The standards pursue the more practical goal of indirect accessibility where the access is mediated by assistive technology.

Multimodal representations combined with the ability to dynamically configure appropriate user interfaces will grant effective access to information technology that can make use of them. For example, manufacturers of consumer products could economically embed the means for their devices to interact with other devices to construct an interface that meets particular users' requirements. For example, a rheostat controlling room lighting and a personal digital assistant (PDA) might mutually discover one another using a protocol such as *Jini*TM [5] and construct an interface on the PDA that would allow its user to control illumination.

Multimodal representations and technologies to create accommodative interfaces will succeed as long as they contribute a small portion of the cost of production. While this is plausible for high traffic commercial Web sites, the

preponderance of Web documents cost so little to produce in the first place that the inclusion of multiple representations would seem prohibitive. Furthermore, because so many documents available on the Web are created with no anticipation of Web presentation, their authors have neither reason nor opportunity to consider representing them with different modalities.

Distributed Adaptation: Web Accessibility Service (WAS)

We have developed a prototype service to evaluate the alternative of operating over a range of accessibility control points, instead of concentrating on one. The service transforms the contents of HTTP requests and responses flowing between Web client and server so that they meet users' requirements for access. Besides making these transformations, the service also provides whatever is necessary to render the transformations appropriately on the client.

By distributing the service across several control points rather than implementing it only upon the client, we reap several powerful advantages, although not without cost. The service is provided through a Web intermediary that adopts many of the roles of both server and client. The intermediary aggregates and orders whatever discrete functions are necessary to meet users' accessibility requirements. For example, a user with the combination of hemianopsia and hemiparesis may require that both display and keyboard require adaptation to meet his or her visual and dexterity limitations. Where no single assistive technology handles both, WAS can configure a service adapting both display and keyboard [10].

Another goal of this effort involves freeing the user from the requirement of having to use a particular device equipped with assistive technology. The pervasiveness of the Web confronts users with public or shared devices that are not pre-configured with assistive technology appropriate to any particular user's requirements. WAS can adapt to user requirements anywhere insofar as the client device can be dynamically configured to support the required renderings.

The service can be broken down into four parts: (1) signon, authentication, and registration (2) user preference retrieval and update, (3) accessibility transformations, and (4) session management.

Signon, authentication, and registration

Signon, authentication, and registration operations pose a difficult bootstrapping problem: how does WAS provide access to the user before having identified that user and retrieved a record of his or her interaction preferences? The problem of providing an initial interface without knowing the user's capabilities can be solved through a succession of interactions that incrementally construct an adaptive interface [4]. The prototype signon procedure works well only for users with low-vision. In the future, it will

contribute to a layered adaptive interface that will also be able to use speech for input and output.

Mechanisms to access user preference records securely cannot use standard browser authentication mechanisms because they require users to be able to read small text and type. Consequently, WAS must provide its own secure authentication mechanism that can be adapted to different user requirements.

User Preference Retrieval

User preferences are represented in XML and define the transforms that will be applied to Web content to render it accessible to a particular user using a particular type of device. Once retrieved, the XML directly guides the application of the series of transforms (see Figure 3).

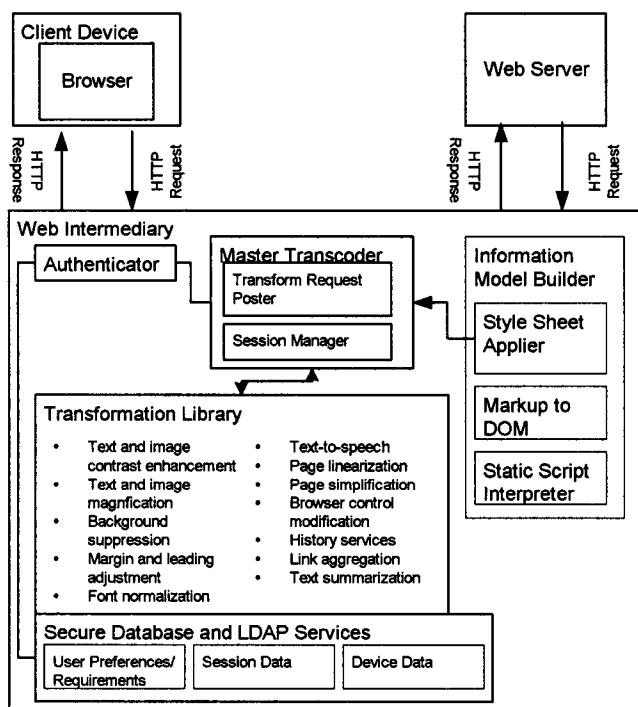


Figure 3: Functional Components of Web Accessibility Service

Accessibility Transformations

The accessibility service can be applied only to the Hypertext Transport Protocol (HTTP). Its current aim is to transform documents composed of Hypertext Markup Language (HTML) and JavaScript. Other document representations (e.g., Portable Document Format (PDF™), Flash™) are planned for future implementation.

The page is retrieved, parsed, and converted into its corresponding Document Object Model (DOM) form. Some transforms, such as those applied to text, can be executed with style sheets. Others, such as image transforms, require programmatic manipulation. Still others, such as auditory rendering or dynamic keyboard adaptation, are handled by injecting JavaScript and Java applets into the

page so that the transformations can be handled locally on the client device.

Session Management

The accessibility service introduces complexity to the task of managing Web transactions. If the transformations must be applied to a Secure Sockets Layer (SSL) connection, the service must replace the session between the user's browser and the Web server with two separate sessions and apply its transformations in between them. Because WAS transforms the transaction contents while they are unencrypted, the service itself must be operated from a secure server.

To simplify a page may entail breaking it up into a set of less complex pages among which the user may navigate. Because serving of these new pages must be handled by the service, instead of the content server, WAS must rewrite links on pages, ensure that they are resolved correctly, and appropriately negotiate cookie transactions.

CONCLUSION

We are developing an intermediary-based strategy to enhance access to the Web that relieves some of the problems that beset conventional client-based assistive technology. In particular, moving the control point for accessibility away from the client enables this service to compose adaptations tailored to a user's particular preferences and capabilities. Moreover, it transfers the cost to acquire, operate, and maintain adaptive technology to either Web content providers or Internet Service Providers (ISP). Finally, because of its multiple roles as browser, server, and transformational engine, it can co-evolve with the emerging, semantically enriched representations that will simplify adaptive access in the future.

Acknowledgements

Aries Ardi, Fran Brown, Susan Crayne, Gregg Daggett, Beth Tibbitts, and Shari Trewin contributed substantially to the conceptualization, design, and development of this system.

REFERENCES

1. Boyer, L., Danielsen, P., Ferrans, J., Karam, G., Ladd, D., Lucas, B., and Rehor, K. (2000) Voice eXtensible Markup Language (VoiceXML) version 1.0. Available at: <http://www.w3.org/TR/voicexml/#h>.
2. Chisolm, W., Vanderheiden, G. and Jacobs, I. (2000) HTML Techniques for Web Content Accessibility Guidelines 1.0. Available at: <http://www.w3.org/TR/WCAG10-HTML-TECHS/>.
3. Dardailler, D., Brewer, J. and Jacobs, I. (2001) Making the Web Accessible. In Constantine Stephanidis (ed). User Interfaces for All. Mahwah, NJ: Lawrence Erlbaum Associates.
4. Dieterich, H., Malinkowski, U., Schneider-Hufschmidt, M., and Kühme, T. (1993) State of the Art in Adaptive User Interfaces. In M. Schneider-Hufschmidt, T.

- Kühme, and U. Malinkowski U. (Eds.), *Adaptive User Interfaces: Principles and Practice*, (pp 13-47). Amsterdam: North-Holland, Elsevier Science.
5. Edwards, W. K. (2000) *Core Jini*. Upper Saddle River, NJ: Prentice-Hall.
 6. IBM Home Page Reader (2001) Available at: <http://www-3.ibm.com/able/hpr.html>
 7. NCITS (2001), National Committee for Information Technology Standards. Available at: <http://www.ncits.org/>
 8. Scott, N. G. and Galan, J. B. The Total Access System. *Proceedings of the 1998 CSUN Conference*. Available at: http://www.dinf.org/csun_98/csun98_151.htm.
 9. Section 508: The Road to Accessibility. Available at: <http://www.section508.gov/>
 10. Trewin, S. and Pain H. (1999) A model of keyboard configuration requirements. *Behaviour and Information Technology*. Special issue on Assistive Technologies for People with Disabilities. Vol. 18, No. 1, pp.27-35.
 11. United Nations (1998) The sex and age distribution of the world populations: 1996 Revision. New York: United Nations.
 12. Web Accessibility Initiative (2001) Available at: <http://www.w3.org/WAI/>

Home Page Reader is a registered trademark of IBM.

Jini is a registered trademark of Sun Microsystems Incorporated.

PDF is a registered trademark of Adobe Systems Incorporated.

Flash is a registered trademark of Macromedia.