

EXPLORING A BASIC ILLITERATE WEB ACCESS SYSTEM:

Discussion and demonstration of technical concepts, and pointers to future research

IDRC Study/Acacia Initiative

Prepared by m-powa, Midrand, South Africa

February 1997

Index

- ☐ Introduction
- ☐ Executive Summary
- ☐ Basic Illiterate Web Access System
- ☐ Alternatives
- ☐ Additions to the Basic System
- ☐ Research Agenda
- ☐ Summary

- ☐ Appendix

Introduction

The technical developments of the last few years have given rise to an unprecedented opportunity for illiterate people. Because of the near-universal adoption of the World Wide Web as a publishing medium, information is now broadly available and accessible. More importantly, this human-oriented information is in a format that can be read by a machine, and thus easily translated or enhanced. It has become feasible for the first time for illiterate people to consider having the same access to information as that taken for granted by literates.

Similar opportunities arise regarding other impairments of the faculties required for reading - visual, language, or motor. Much work has been done to address the needs of blind people, for example, in operating a computer, and some newer work is available pertaining to the specific problem of accessing the Web (see Appendix for representative examples). Where the greatest problems of illiteracy lie, in the developing countries, is however where the resources for the technical work applicable to these very problems are least available.

South Africa straddles the two domains (highly developed and underdeveloped) and it is perhaps natural that it would be from here that the first initiatives for projects in the area of illiterate-empowerment via technology should come. We have found that such a project is quite feasible in terms of existing technology, and yet has not been done before. It is therefore an idea with great potential for development.

Below we propose a basic system structure for illiterate web access and later we discuss additional possibilities and ideas for practical implementation.

*Once the web-based version of this document is completely developed, it will provide examples of the look-and-feel (and sound) of voice-enabled pages, have page enhancements, and have links to some of the information sources upon which the report is based.

Executive Summary

The research was intended to investigate the presentation of Internet information in audio format in order to make it available to all members of society, at all levels, and has brought

about the development of some interesting concepts. The main question it presents as opposed to answers however, is does the use of audio extend the availability of Internet-based information and can illiterate people and non-English-speakers gain access to the wealth of information available to those in the literate First World?

While the international community is certainly looking at the development of audio content on the Internet the focus is not on providing existing content for illiterate people in a format that they can relate to and understand. Internationally, the focus is on presenting audio information for "radio" broadcasts or presenting information to the visually impaired. The challenge is to develop a system, that can be utilised by all, whether they are illiterate or not.

In reviewing the developments in the field of internet audio, we have found technologies available that will be useful to greater and lesser degrees. Our evaluation of these is given below. It is followed by the outline of a purpose-designed system for illiterate Web-page access, for we have found that there is no development directed specifically at that goal.

Audio and speech compression and transmission tools: These are abundant and greatly improve the quality and practicality of delivering audio over the Internet. They are used in multimedia pages, internet radio stations and internet 'phones. They are a necessary component of any proposed system, but they serve only to add general audio capabilities. Specialised software is required to apply these capabilities to the function of text-to-audible-speech conversion.

Specific aids for visually impaired: An example of specialised software mentioned above, these tools can read parts of a page aloud, but the page is extensively stripped and reformatted to suit the needs of visually-impaired persons. The result is not suitable for those who can benefit from visual information or as an all-purpose (literate/illiterate) tool.

Speech synthesis and text readers: Speech synthesis is a promising means to obtain efficient delivery of audio, and is expected to become, but is not yet, available for Web-page applications as are the audio compression technologies mentioned above. Speech synthesis is used in conjunction with general text reader software.

Text readers can read a chosen portion of text on the screen. With some customisation of the speech synthesis pronunciation, they would be useful as a first approximation to the proposed system, since no further development would be required to use them to read a page. However their use would be limiting in the long term, as is described presently. They are also a little more complex to use than our proposed system.

There are two ways of developing this type of system; either server based or client based. This research proposes a server based approach in order to simplify the development as much as possible and to ensure that the standard browsers (i.e. Netscape or Internet Explorer) can be utilised to access the data. This method will facilitate the creation of audio assisted data so that users in community based access points can access any existing information. If they are literate, they will not need to utilise the additional functionality of the presented data but the access will remain the same for all. There will be no additional expertise required at the community level. This approach could also act as the start to the presentation of information in local languages.

We believe a custom-designed system is likely to be, at worst, more useful than exclusive use of off-the-shelf tools, and at best it will be entirely necessary. We have proposed that the custom design be concentrated in a special server (computer on the internet) that will intercept requests to view pages and transparently add text-to-speech functionality.

Although it is possible to consider customising the web browsers that will be used to gain illiterate access, a great deal of the existing browser functionality would need changing, resulting in far greater development costs. An example of the implications of custom browser

development can be seen in the behaviour of the pwWebspeak browser for the visually-impaired.

It dispenses with all graphical information in the page, since that is not required, and thus can concentrate exclusively on extracting and speaking words. If it had been necessary or desirable, as it is in the present case, to preserve graphical information, the idea of a custom browser would not have been used. It would have required duplication of much of the functionality of high-end browsers, which would have made its development costs soar, and would be an immense reinvention of a very prominent wheel, namely Netscape and Microsoft products.

There is a need for a simple, quick way to implement the system in its first phases, because the questions of how it will best be used cannot be answered until the system can be tested in the field. The precise advantage of a custom system and its fine-tuning to the specialised needs of the communities, will be lost if it cannot be cheaply prototyped and tested. Added to this is the great likelihood that different features will be needed for different communities (for example illiterate communities whose first language is English, and those for whom it is not). Using a server-based solution it is easy to achieve rapid prototyping because of the fact that additional Web-page features will be inserted into existing web-pages. Different servers will also be able to serve different communities, with the localisation of the server being performed by local developers. This is possible because the server software will operate within an environment defined by international standards and publicly available tools.

Besides faster and cheaper development cycles, a number of other related advantages are applicable in the server-based system:

Other functionality. While a simple text-to-speech function can be achieved without the need for specialised servers or browsers, the weakness of off-the-shelf systems becomes apparent as soon as any refinements are found desirable. For example it might be found that it is too distracting for the user to have to highlight a portion of the screen before playing it. It might be desirable for the words to be highlighted as they are pronounced. Special buttons that speak instructions on how to use them would be desirable, etc. All these modifications would be straightforward on a server-based system and require little or no user retraining or software updating. They would be unavailable on an off-the-shelf system, or at least subject to the initiatives of the commercial developers, or requiring adoption of new ensembles of software components.

More powerful tools. Other additional functionality could well be desirable. Dictionary lookup of words, intelligent word extraction or language translation are some examples. These would require powerful software. It is not feasible to supply such software to the majority of end users because of licensing costs and high-performance hardware requirements. With a server, a single powerful machine can provide sophisticated benefits to low-end users.

Cross-platform. By avoiding non-standard browser modifications and sticking to data standards, the server system will maintain its goal of adding specialised behaviour to web pages without tying users down to ageing technology, and will work for anyone, anywhere.

The only real way to test these concepts is to establish pilot studies. These studies need to be tested in conjunction with existing initiatives for community based information services to test the types of information required, who is utilising this information and who is paying for this information delivery.

The recommendation of this report is therefore a three-stage development procedure. Stage one is to obtain initial reactions to spoken web pages, from a sample of the intended users. We would use existing text-reading software and speech-enabled web browsers in human-assisted or monitored environments. Stage two would be a prototype server that will clarify implementation issues and allow broader and more comprehensive testing. Stage three

will be a redesigned system taking into account the lessons learned.

Basic Illiterate Web Access System

The literacy server

The system is composed of a single physical component: A "literacy server". A server is central computer which performs a service for peripheral "clients" or users (in our case, users are illiterates who wish to access the Web). By implementing the literacy software on a server, the need for special software or user procedures is reduced or eliminated. As the project develops, more and more functionality can be provided without problems of updating client software.

The literacy server will be interposed between the web client (the browser software such as Netscape Navigator or Microsoft Explorer) and the web server (the site which the user wishes to access). This interposition is termed "proxying" and there are publicly available resources for creating customised proxy servers (see Appendix: The Apache Server). Every Web page requested by the user will be processed by the literacy server before being passed on.

Speaking pages

The second aspect of the system is not so much a component as a software strategy. We refer to the specific processing that is to be performed by the literacy server. Our proposed strategy is to add the following basic functionality to the web pages, which after processing, will be visually similar to the original: When the user passes the mouse pointer over the words on the page (or touches them using a touch screen), they will be audibly pronounced by the computer. The functionality will be added through the use of two technologies: Browser programming and speech production. Browser programming refers to the use of built-in scripting languages (JavaScript) and code (Java applets) to add interactivity and special functions. In this case, it would be used to make the text sensitive to the mouse movements and pass information to the speech production unit. Since JavaScript and Java are open and ubiquitous standards, they represent the most cost-effective and efficient way to do the special web pages.

Speech synthesis is chosen as the mostly likely means to produce the audible output. The alternative is prerecorded words. Speech synthesis is far more efficient than recorded sound. The efficiency is an important issue in Web system design, and refers to the amount of data that must be transferred. Recorded sounds (a recording of each word on the page) might take so long to reach the user (through a slow connection to the internet) that they render the system unusable. Another important consideration is the cost of recording and storage of all the possible words, especially if a number of alternative languages are to be provided. That said, speech synthesis has the disadvantage of inaccurate pronunciation. It is also not quite ready for implementation in the browser environment as discussed below. In the "Alternatives" section, the viability of recording sounds is discussed in the light of some available technologies. In order to achieve speech synthesised web pages, two components are required: Speech synthesis software and a speech synthesis browser plugin. Speech synthesis software is chosen over speech synthesis hardware as being cheaper while providing quality adequate and appropriate to a prototype or exploratory project. If high quality hardware synthesis can be justified it can be exchanged for the software component at a later stage. The software runs in the native OS environment (DOS/Windows, MacOS, or UNIX). There are freely/widely available software packages for all three environments (see Appendix: for example Creative TextAssist for DOS/Windows, PlainTalk for Macintosh, and rsynth for UNIX).

Using TextAssist as a representative example we have found the basic low-cost software synthesis to be surprisingly intelligible. The pronunciations may be programmed to some degree, allowing localisation and non-English language dictionaries to be compiled. Other

speech systems are perhaps more sophisticated but certainly more expensive.

*—active content-----

Click here to evaluate speech pronounced by TextAssist.

*-----

However the speech synthesis must take place in the browser software environment, which is separate from the native OS. For this reason a plugin linking the browser to the external synthesis software is required. Such a plugin does not yet exist for the PC (Windows) or UNIX environment, but there is one for the Mac called Talker (see Appendix). The plugin effectively provides Netscape with the basic vocal functionality, required for the illiterate access application, to web pages viewed by Macintosh users. That is, it is being presently used to cause a web page to read itself aloud automatically.

It might be thought then that no literacy server or development is required at all, and that the available software could be used without further effort. We do not believe this to be the case for the following reasons:

- 1. The interposing literacy server with its programmable additions to the web pages will provide a great deal of added value to the system, in that it will allow ongoing improvement, tailoring and enhancement of the behaviour of the voice-enabled pages. See the "Additions" section below. Without it, the only improvements will be those initiated by the developers - MVP Solutions and Apple
- 2. The server will also offer a perfect opportunity to add custom designed content, educational resources, etc. as discussed in the "Research" section below.
- 3. There is no PC-based plugin as yet, and Apple Macintosh computers are higher-end and less flexible than PC compatibles. The current difficulties of Apple as a company mean that long term planning reliant on their technology is risky.
- 4. Pending full evaluation of Talker, we cannot be sure that it can provide the required functionality (mouse sensitive playback etc.)

On the other hand, the Talker software will provide a perfect vehicle for a preliminary study into the appropriateness and effectiveness of the illiteracy web access concept, as discussed in "Research" below. Its existence also implies that a PC-based plugin is not far off. It is therefore important to note that it would not be prudent to embark upon development of such a plugin on our initiative. We would almost certainly find our efforts made redundant within months. Nevertheless the plugin is almost certainly an essential requirement a speech synthesis-based solution (the alternatives are briefly covered in a special section later on).

The conclusion that we reach is therefore to adopt a recorded sound-based approach in the short term and for evaluation purposes. There is no need for synthesis for plugin software and so the project can proceed with development of the server. When the plugin becomes available it will be a simple and natural step to integrate the speech synthesis. Meanwhile, as discussed in the "Alternatives" section below, the recording technologies are likely to be very viable for all preliminary phases.

*—active content-----

This paragraph has been "illiterate-enabled" in simulation of the effect of the proposed system. Simply move the mouse over its words to hear them spoken. The illiteracy server will create pages like this automatically. This demonstration gives an accurate idea of what can be done

using today's recorded speech technology.

* -----Please set your browser preferences to remove underlining of links, to better simulate the proposed effect.--

Alternatives

Custom browser vs. custom server

For a customizable system, the only alternative to the server-based approach is a custom-written web browser to be obtained and used by every person who wishes to have illiterate access. The development effort for such a project is far greater than for a proxy server, since browsers must run on a variety of platforms and must keep up to date with Web technology changes (see note on Java below). There is really no competing with Netscape and Microsoft in this arena. The "speaking browser for the blind" noted in the Appendix (see pwWebSpeak) was probably feasible only because it made no attempt to preserve normal visual browser functions. It extracts the text from a page, formats it for voice playback and discards almost all visual information.

Augmented screen reader vs. augmented browser

Rather than look for a browser plugin that integrates it with speech synthesis software, it is possible to consider modifying or enhancing a synthesizer or screen reader package (see Appendix) such that it integrates with the browser. For example Creative TextAssist can be used to read each word in a web page by correctly operating the mouse and the application. The actions required are tedious however (user must highlight words manually). Any solution of this nature would be likely to involve a number of piecemeal workarounds. It would not have the elegance and integration of a synthesis-aware browser. Additional functionality would require updating an ensemble of small and large programs which would need to be redistributed to every user. This is in contrast to the centralised nature of a system consisting of a browser (augmented with plugins) and enhanced pages created by a literacy server. Typically the plugin would be supplied initially to the user, and later enhancements to the system would not likely require updates to the plugin.

Recordings vs. synthesis

If recorded sounds are to be used instead of synthesized speech, for reasons of speech accuracy or implementation ease, then efficient ways to store and transmit large amounts of sound data will be necessary. Of the possibilities reviewed, those which stand out are RealAudio 3.0 and Toolvox, both Netscape plugins (which means they work on all popular platforms and browsers).

RealAudio is designed for the efficient transmission of speech over low bandwidth internet connections. It achieves high compression of raw sound data (from 1:70 to 1:17 in our preliminary trials on speech data, depending on desired quality). In addition a RealAudio server can be purchased which "stream" data, meaning the file size is less relevant, sounds starting to play soon after transmission begins. Using basic RealAudio would be free of charge, but streaming would require purchase of software for the central server site. RealAudio also allows fine programmatic control of sequence of playback, which is important in this application - the plugin must have JavaScript handles for programmability.

ToolVox is a compression system designed specifically for speech, and thus is very efficient (claimed 1:50) for speech files. In addition Netscape browsers themselves have native streaming formats which could not be tested due to their very recent release. Microsoft servers have streaming products. However RealAudio is the best known and close to a de facto standard.

With the above compression ratios, recorded sounds could be a practicable alternative to speech synthesis. However this matter can only be settled satisfactorily by experimenting with the available technologies. Some further such work is required.

Additions to the Basic System

The system would be open to expansion in an incremental fashion. It would evolve into a comprehensive site with multiple avenues of education and information access. According to the success of the basic system, the following additional features could be considered.

Translation

By changing the speech-production dictionary, a simple single-word language translation facility may be implemented. Pointing at an English word would cause an utterance of a native-language translation of the word. This technique can also be used to localise the accent of the English pronunciation, which can be in wide variance with the generic computerised American pronunciation and may considerably impede comprehension.

*—active content-----

Click here to hear how customisation of speech
synthesis can give different accents.

Move the mouse over these words to hear them
defined in the Soho language.

*-----

A separate text-based definition dictionary could be implemented. A window would appear containing a more detailed explanation of the desired word, synonyms, translations and alternative translations, etc. The definition text itself would of course be self-pronouncing.

*In the demonstration above, click the words to see them defined in Soho language.

More sophisticated translation software, that could correctly translate whole sentences and paragraphs with correct grammar, is probably beyond the scope of this project. Although quite a lot of work has been done to produce translation aids for those who work in the field, the output usually contains significant inaccuracies. A system which would increase comprehension rather than confuse is likely to be extremely expensive, incorporating highly sophisticated artificial intelligence algorithms. Before investing in development with such systems, the usefulness of a basic illiterate web access system would need to be thoroughly researched through prototyping and field evaluation. See "Research" below.

Visual augmentation of pages

Users are going to need help not only reading words but using the browser software, the buttons and menus of which usually rely on text to explain themselves. The literacy server can most easily address this problem by adding "talking" buttons for the most common browser functions directly onto each page. This can be done with straightforward Browser programming.

Custom educational content

Independent of the development of the literacy server, it will be possible (likely very necessary) to provide custom web pages designed as tutorials and references for the illiterate

users. Being custom created pages they will incorporate the ability to read themselves aloud and interactively guide the users into learning how to use the system. Such content is currently in development at m-powa.

Links to relevant sites

Bringing together an index to the most useful and relevant content for the user base would be a natural role for the literacy server site to fulfil. Links to other sites would be gathered and organized into a growing resource. Users would contribute the sites they found meaningful, and the beginnings of an internet community would be formed.

* Click here to view an example page containing demonstration

visual augmentation, custom content and links to other sites.

Research Agenda

The preceding sections answer the question of what practical steps to take to begin to enable illiterate access to the Web. They describe an essentially simple and quite feasible system. It can be developed with currently available technology, at a relatively low overhead cost, making it suitable to function as a prototype.

We would therefore outline a research agenda to be undertaken as follows:

Create a prototype system:

Manually create a representative range of examples of what the literacy server would output, including recorded and synthesized examples. Evaluate in field tests and in consultation with educational experts, in academic and school environments, and with reference to government and private sector guidelines. Based on feedback, implement a literacy server to allow comprehensive testing -Evaluate the prototype system using live (public) web pages:

Consultation with other pilot projects to design appropriate testing protocols Field testing to comprise observation of and discussion with illiterate subjects

- Address amongst others the following issues:
- Relevance of available content
- Usefulness of system in wider context of internet-based activity, for example how much use is a web-page enabling technology without an e-mail reading facility, and how much use is reading in the absence of writing (input) facilities.
- Issues pertaining to the system's user interface.

Summary

There currently exists a real possibility of enabling a new level of empowerment to be attained by otherwise severely disadvantaged people. The possibility, of opening up the world's information sources to illiterate people or those at a language disadvantage in an increasingly Anglicised first world, is so new that its consequences cannot be accurately predicted without concrete measurements. Fortunately the technology that is involved may be cheaply prototyped to allow in-depth investigation of the possibilities. Chief amongst the questions to be answered is whether the substantially strange new world opened up to the illiterate will be perceived as relevant and have beneficial applications. The new channels available for communication will need to be allowed to evolve, flourish and engender new relationships between information consumers and information providers before their full impact can be assessed. Clearly however it is well worth the effort to research the proposed system. The consequences of

inaccessibility in the undeveloped world to the rapidly accelerating changes in the developing world are widening gaps that spell disaster for all involved.

Appendix

Technology survey: links to excerpts from information sources

Excerpts:

Click on Normal links to see a local copy of web page.

Click on Bold links only if you are connected to the Internet, to reach a public website.

Text-to-speech technology

Screen Readers

Telesensory has software for Windows and DOS that allows, amongst other features related to blind access, audio reading by character, word, line or window.

Frontier Computing is based in Canada and caters for the visually impaired. It acts as a source for software. Text-to-speech software provided is as follows:

- Arctic - Makers of WinVision, Business Vision
- GW Micro - Makers of Vocal-Eyes, Window-Eyes (has visual word highlighting feature)
- Biolink - Makers of Protalk
- Henter-Joyce - Makers of JAWS
- IBM - Makers of IBM Screen Reader (tailored for use with IBM software - e.g with Web Explorer, has voice changes on hyperlinks).

The above site's home page is an example of a site which has been tailored to be friendly to blind people - it is meant to be clear to both sighted and blind people using text-to-speech devices or other aids.

Productivity Works have a speaker program, pwWebSpeak designed specifically to read web pages - that is, it read the HTML and thus understands more than English text. Again the emphasis is on aid the visually impaired, so some of the intelligence is likely unnecessary. Their product however would be a good starting point for developing a similar one tailored to aid literacy. An upgrade planned for March '97 and called WebReader is due which will be tailored to dyslexics and partially sighted and work with Microsoft Internet Explorer (thus supplying fuller featured web-browsing). Foreign language (European) versions are being developed. It is recent software, shown by review in CNN from Feb 2.

Public Speech Synthesis software for UNIX is available with source code- this could be a basis for custom designed browsers or plugins that need to talk

PC-based Software speech synthesis: Creative Labs TextAssist (part of software pack with SoundBlaster cards), Digital's DECtalk PC, Productivity Works' SoftVoice

Mac-based software speech synthesis: Apple has a freely available software speech synthesizer which, together with an Mac-based Netscape plugin called Talker by MVP solutions, allows web pages to read themselves aloud. This could not be directly evaluated in time for the present report, but it is a most important component and will be examined as soon as possible.

Web Access for Visually Impaired

The Swiss Eidgenossische Technische Hochschule (ETH) has a web service for blind users. With no special software required, the ETH site is used as an intermediate processor of web pages. Any web page is delivered after being fed through the site and reformatted for use with the user's existing voice-synthesized screen reading package.

Machine Enhanced Literacy

Project LISTEN at CMU is project to teach literacy through machine speech and voice recognition of stories. It is a custom program with custom material, not web-based or applicable to external content.

Translation technology

The WHO has investigated machine translation and found that a fully automatic approach is not cost effective.

A Swahili Translation system is available as a windows application. It uses a word-for-word-based system enhanced by some learning capacity. It therefore could serve as a useful evaluation for what may realistically be achieved, since it matches our proposed possibilities in language group (African rather than European) and translation strategy (word-based).

There are no references relating to Southern African language machine translation.

Proxy Web Server

The most straightforward way to create a custom proxy web server would be to use Apache. This is the most widely used web server in the world, it is free, it is efficient, and it has an API (Application Programm Interface) which allows third party modules to be developed and "plugged in". A proxy module already exists, its source code is available and thus the path to developing a customised proxy server is straightforward (although probably not trivial). The wide popularity and continued development of Apache make it the premier public UNIX web server package.

EXPLORING A BASIC ILLITERATE WEB ACCESS SYSTEM

| - Index - |