Emacspeak —Toward The Speech-enabled Semantic WWW

T. V. Raman

URL **http://www.cs.cornell.edu/home/raman**

March 2, 2001

Abstract

Emacspeak has pioneered the speech-enabling approach to providing intelligent spoken feedback for a variety of daily computing tasks. This includes audio formatted output from World Wide Web (WWW) pages by utilizing Aural Cascading Style Sheets (ACSS). However, until recently such spoken output has been limited by presentational HTML pages optimized for visual interaction.

The WWW is presently transitioning toward a data-centric architecture; content —and its semantics— is encapsulated in XML ([W3C98]) pages designed to be served in a manner most appropriate to a given client. This opens up significant opportunities in generating high-quality spoken feedback from richly encoded WWW content. Though XML is still in its early stages of wide-spread adoption, some of the benefits to come can already be seen today. Many sites now offer access to both presentational HTML, as well as the underlying data. Examples include historical stock charts, driving directions, and other useful information.

Emacspeak now exploits the availability of such semantically encoded content to provide a richer end-user experience. This article introduces some of the data acquisition techniques used in Emacspeak and focuses on the end-user experience when interacting with such structured information.

1 Introduction

The move to XML content ([W3C98]) on the World Wide Web (WWW) takes us one step closer to a Model, View, Controller (MVC) architecture —see [Gol90, KP88]. Semantically encoded XML content —the *model*— is presented using modality-specific *viewers* and manipulated by modality-specific *controllers*.

Until recently, the primary features of WWW interaction have been determined by desktop GUI browsers. However, with the move to an increasingly mobile world of information access, additional modalities such as speech are beginning to take on an increasingly important role. Combined with this, the arrival of small display hand-held devices has caused content providers to rethink their publishing strategy; the need to serve different types of end-user clients is now driving the evolution toward a WWW where content is published in a semantically encoded XML form that is delivered to clients based on the individual characteristics of the accessing device.

Emacspeak introduced the speech-enabling approach —see [Ram96a, Ram96b, Ram97a]— to provide a complete audio desktop. With the coming of the semantic WWW, Emacspeak is now evolving to provide efficient spoken access to richly encoded WWW content.

The semantic WWW is speech-enabled by:

1. Acquiring content semantics —2.
2. Constructing high-level representations —3.
3. Using these representations to produce effective auditory user interfaces (AUI) —4.

Emacspeak also provides a rich suite of task-oriented search tools that enable the user to efficiently navigate the WWW —described in 5. We conclude the paper with a short section describing the end-user experience in section 6.

2 Acquiring Data

We speech-enable the WWW by *audio formatting* online WWW pages. This requires high-level document models —see [Ram98, Ram94]. The WWW is moving from the world of purely presentational (and often malformed) HTML to well-formed XHTML —see [BPSM00]. During this transition, data acquisition requires specialized modules that overcome problems with malformed content.

We first describe the data acquisition techniques used by Emacspeak when working with the transitional WWW. With the adoption of XML, we expect many of these specialized modules to become obsolete. However, the effort developing these modules is not wasted; working with transitional WWW content allows us to experiment with various forms of spoken interaction and navigation. Further, we gain insight in developing the next-generation content models for publishing information designed for ubiquitous, universal access.

2.1 Financial Data

Portals like Yahoo[1] offer financial data *e.g.,* stock quotes and historical stock charts. In addition to presentational HTML pages, these sites also provide the data as comma separated values (CSV files) to enable importing the data into spreadsheet programs. The CSV format also prove useful in constructing high-level representations for use in producing speech-enabled interaction.

Emacspeak constructs rich tabular structures given such financial data. Notice that though the data acquisition functions are site-specific, the constructed document models are independent of the content source. This allows Emacspeak to present a consistent *sound and feel* across the audio desktop independent of the source of the content

[1] *URL* **http://finance.yahoo.com**

being presented. The tabular representations identify row and column headers explicitly. In addition, the representation allows for nested tables, and this permits Emacspeak to later provide high-level spoken descriptions of complex semantic relationships amongst data items.

2.2 Custom Newspapers

The WWW has been a major information access boon to visually impaired users. The flood of constantly updating online news has caused us to move rapidly to a world where there is now simply too much available information! Additionally, observe that as the pace at which information is updated increases, there is a corresponding drop in the *half-life* of information. Today, online news becomes outdated almost as quickly as it becomes available —or put differently— "USA Today tomorrow is USA yesterday!".

This evolution adds an entirely new dimension to the issue of *equal* information access — in today's world, to achieve equality one needs to more than just have *access* to information— such access has to be both timely and efficient in order to ensure that one can keep pace with ones peers.

Keeping up with this flood of continuously updated information —especially when interacting using a temporal modality like speech— requires intelligent agents that can gather, filter and categorize information before presenting it in an easy to digest format. This becomes especially critical on today's WWW, given that each news site is cluttered with heavy-weight visual eye-candy that to the average user proves a nuisance —but proves a complete show-stopper to the visually impaired user attempting to access the information with traditional access solutions such as screen-readers combined with a visual browser.

Emacspeak combines with NewsClipper[2] to enable users construct personalized newspapers. After I started using Emacspeak with NewsClipper, I found that the amount of news I was able to peruse on a daily basis went up drastically; —primarily because of the time savings that result from having the Emacspeak/NewsClipper combination process complex WWW sites to provide a succinct overview of what is available. Consequently, I, as the end-user, can focus on the most interesting aspect — namely reading the news. Now, my Linux workstation constructs my custom newspaper each morning at 6:30am; the custom WWW page is rendered via speech in my living room as I have my morning coffee.

3 Data Representations

Trees and arrays are the most useful data structures in providing spoken access to structured information. Trees can be used in progressively *displaying* large amounts of structured content —for details on the use of complex tree structures in browsing mathematics, see [Ram98, Ram94]. Tabular structures are used to capture relational data that is presented using user-customizable renderings that convey both the content and semantics of each cell in the table.

[2] *URL* **http://www.newsclipper.com**

3.1 Constructing Tree Structures

The Document Object Model (W3C DOM) —[W3C00]— is a reliable source of tree structure when browsing online WWW pages. However, HTML pages on today's WWW are often malformed, and even when one has worked around this to create a usable DOM, the resulting tree often reflects the layout structure of the visual presentation —rather than the underlying logical structure.

Emacspeak uses sectioning tags such as `h1` and `h2` to generate a table of contents for the page. This proves effective in browsing large documents, *e.g.,* W3C specifications[3]. Emacspeak also provides a primitive form of DOM-based document navigation by allowing the user to move through nodes in the underlying document model —this is most useful in moving across constructs like tables. As we evolve toward well-formed WWW pages, the DOM will enable Emacspeak provide a richer end-user experience in filtering and selectively reading portions of a WWW page.

3.2 Processing Nested Tables

Present-day HTML precludes reliable separation of tables used for visual layout from tables used to encapsulate relational data. At the same time, today's WWW pages demonstrate that rather than being two discrete extremes, tables used for relational data and tables used purely for visual layout represent two end-points of a continuous spectrum —to see this, consider financial sites that use nested tables to convey complex relations amongst data *e.g.,* interest rate trends juxtaposed with loan packages from competing lenders —a good example of this is *URL* **http://loan.yahoo.com**.

4 Auditory Interfaces

Auditory interfaces are generated as modality-specific *views* of the *models* described in the previous section. These interfaces are only as effective as the underlying model; for details on the design of the ideal auditory user interface, see [Ram97a]. Here, we describe what is possible given today's high-level representations and indicate how Emacspeak plans to evolve toward the idealized AUI.

4.1 Interacting With WWW Pages

WWW pages are *audio formatted* by applying user-defined aural style sheets as specified by ACSS [Ram97b, BLLJ98]. Navigation through the page is facilitated by two forms of conceptual *table of contents* —one derived from the sectioning constructs present in the DOM, and the other made up of the list of hypertext links on the page. HTML navigational aid such as client-side image maps are presented as a list of choices with completion —the user can select amongst the choices with a few keystrokes.

[3] *URL* **http://www.w3.org/tr**

4.2 Navigating And Browsing Tree Structures

As Emacspeak evolves toward using the W3C DOM, we plan to evolve the system to provide interactive audio browsing as described in ASTER [Ram98, Ram94]. Such structured browsing allows the user to traverse the logical structure tree, filter the tree to select nodes of interest, and finally listen to the selected content.

4.3 Browsing Complex Tables

Emacspeak's user interface for navigating and browsing tables is detailed in [Ram97a]. The interface permits the user to navigate the rows and columns of the table with single keystrokes. As table navigation progresses, the contents of the current cell, row or column is spoken using a user-customizable rendering rule. This allows users to provide task-specific rendering rules that generate meaningful natural language utterances given structured tabular data.

As an example, consider the disk usage report produced by command `df` shown in table Table 1. By default, the user hears the content of each cell while navigating the table. This can be customized by having the system speak row and or column headers along with the contents of each cell –to produce an utterance of the form

/dev/hdb8 Used % 16%

the user can also supply a rendering rule for summarizing rows or columns of the table. For example, consider the row rendering rule

("disk " 0 " is" 4 " full ")

The above rendering rule provides a template for the natural language utterance to be produced for each row. In applying this rendering rule, the integers are replaced by the cell contents in the corresponding column to produce

Disk /dev/hdb8 is 16% full.

Filesystem	1k-blocks	Used	Available	Use%	Mounted on
/dev/hda5	132207	82501	42880	66%	/
/dev/hdb6	3028080	417972	2456288	15%	/disk/local1
/dev/hdb7	3028080	2028	2872232	0%	/disk/local2
/dev/hdb8	3028080	447548	2426712	16%	/disk/local3
/dev/hdb9	1589076	469600	1038752	31%	/disk/local4
/dev/hdb1	3028080	307256	2567004	11%	/home
/dev/hdb5	3028080	124896	2749364	4%	/home/httpd
/dev/hda6	2016016	1509904	403700	79%	/usr
/dev/sda4	1029032	288596	688164	30%	/mnt/ejaz
/dev/hdc	656134	656134	0	100%	/mnt/cdrom

Table 1: Emacspeak's table browsing interface allows the user to provide task-specific custom rendering rules that produce meaningful natural language utterances from structured data.

5 Task-oriented Search

Emacspeak provides a WWW search tool that is accessible with a single keystroke from anywhere on the Emacspeak desktop. We first motivate this tool by describing the use of interactive WWW sites like *maps.yahoo.com*. The remainder of this section lists the various search utilities provided by Emacspeak's websearch tool.

A few years ago, programs that could give directions given a specific address cost several hundreds of dollars and required the user to have a disk containing the map data for a specific region; today, this information is available for free at the click of a button —thanks to map portals like Yahoo (*URL* **http://maps.yahoo.com**. Though designed primarily to provide driving directions, such mapping applications are extremely useful to visually impaired users —especially given that these portals offer step-by-step textual directions in addition to graphic images.

However, the visual design of such portals sites optimized to *capture eyeballs* can leave the blind user fishing in the dark —an experience that dissuaded me from using these interactive sites until I developed Emacspeak's websearch tool.

The primary access barrier to efficiently using sites like *maps.yahoo.com* is that the form fields where one specifies location information are buried among a lot of visual gadgets on the page. This means that in addition to the obligatory world-wide wait experienced by all users, visually impaired users suffer an added hit by having to spend time locating the appropriate user interface controls on the page. Emacspeak separates the user interface for prompting the user from the act of getting the requisite data —Emacspeak therefore asks the user the appropriate questions about start and end locations before *posting* the query to the map portal. This means that the user is saved the effort of waiting for the initial page to download and render as well as the trouble of locating the appropriate user interface widgets on the page. Additionally, upon receiving the server response, Emacspeak renders the page using the W3 browser, and then focuses in on the portion of the page containing the directions. Finally the directions are spoken —once again obviating the need for the user to wade through and filter out extraneous content on the page.

Notice that the above interaction, though motivated by the needs of someone who is visually impaired, perfectly fits the profile of *functionally* blind individuals such as cell-phone users.

Emacspeak websearch tools are implemented using a *pre-processor* that prompts for search parameters, and a *post-processor* that processes the result page and speaks the relevant information —see table Table 2 on the next page for a list of the available websearch tools. In addition, some utilities like *company-news* offer additional specialized searches; thus, after the user has selected *company-news* and specified the stock tickers of interest, she can pick any one of the tasks shown in table Table 3 on page 8 to pull up the relevant piece of information. Similarly, selecting *software search* prompts for the specific software archive to search —see 4.

Key	Description
a	AltaVista Search
A	All The Web
b	BBC Archives
C-b	CS Bibliography
c	CNN Interactive
C	Company News
d	Dejanews Usenet Archives
D	Websters Dictionary
e	Encyclopedia Britannica
f	CNN FN
F	Dictionary Of Computing
g	Google WWW Index
h	HotBot WWW Index
i	Inference Search
j	Ask Jeeves
l	Lycos WWW Index
m	Driving directions
M	Merriam Webster
n	News Wire
N	Northern Light
o	Open Directory
p	Linux Packages
r	Search RedHat
R	RFB&D Catalog
s	Software Archives
w	Weather Channel
W	Search W3C
v	Vickers Insider Trades
V	VectorVest Stock Reports
y	Yahoo Index

Table 2: Search tools currently speech-enabled by Emacspeak. Speech-enabling these popular WWW utilities enables the Emacspeak user to focus on the task at hand and obtain the desired information in a timely manner.

Key	Task
c	Upgrade history
h	historical charts
n	News articles
p	Company Profile
r	Company Research
t	insider trades

Table 3: Selecting *company news* from the Emacspeak websearch tool allows the user to specify the type of information to be retrieved.

Key	Repository
sa	AppWatch
sf	Freshmeat
sp	Comprehensive Perl Archive
ss	SourceForge
st	Comprehensive TEX Archive

Table 4: Selecting *software search* from the Emacspeak websearch tool allows the user to specify the software repository to search.

6 User Experience

The WWW is an important source of information for both work and pleasure. To use this information resource effectively, it needs to be well-integrated into all aspects of daily computing. The commercial WWW is optimized to present users with pages that maximize the time spent on a given site; in contrast, productive use of the WWW requires optimizing interaction along a different dimension, namely, how quickly one returns to the task at hand after retrieving a requisite item of information. This conflict is amplified when using temporal modalities like speech.

The Emacspeak desktop allows the user to focus on the task at hand by providing the necessary utilities to quickly obtain relevant information. This has enabled the author and hundreds of Emacspeak users around the world to more effectively utilize the WWW for both work and pleasure. These tools draw on the availability of semantically encoded content; as we evolve toward tomorrow's semantic WWW, these tools —and the concomitant user experience— can be expected to get correspondingly richer.

References

[BLLJ98] Bert Bos, Häkon Wium Lie, Chris Lilley, and Ian Jacobs. Cascading style sheets(css2) specification. Technical report, World Wide Web Consortium

W3C, 1998. *URL* **http://www.w3.org/TR/REC-CSS2**.

[BPSM00] Tim Bray, Jean Paoli, and C. M. Sperberg-McQueen. Extensible hypertext markup language xml 1.0. Technical report, World Wide Web Consortium W3C, 2000. *URL* **http://www.w3.org/TR/xhtml1**.

[Gol90] Adele Goldberg. Information models, views, and controllers (software reuse). *Dr. Dobb's Journal of Software Tools*, 15(7):54, 56–59, 61, 106–107, July 1990.

[KP88] Glenn E. Krasner and Stephen T. Pope. A cookbook for using the model-view-controller user interface paradigm in smalltalk-80. Technical report, 1988.

[Ram94] T. V. Raman. *Audio System for Technical Readings*. PhD thesis, Cornell University, May 1994. *URL* **http://cs.cornell.edu/home/raman**.

[Ram96a] T. V. Raman. Emacspeak —direct speech access. *Proc. of The Second Annual ACM Conference on Assistive Technologies (ASSETS '96)*, Apr 1996.

[Ram96b] T. V. Raman. Emacspeak: A speech interface. In Michael J. Tauber, Victoria Bellotti, Robin Jeffries, Jock D. Mackinlay, and Jakob Nielsen, editors, *Proceedings of the Conference on Human Factors in Computing Systems : Commun Ground*, pages 66–71, New York, 13–18 April 1996. ACM Press.

[Ram97a] T. V. Raman. *Auditory User Interfaces –Toward The Speaking Computer*. Kluwer Academic Publishers, August 1997.

[Ram97b] T. V. Raman. Cascaded speech style sheets. In M. R. Genesereth and A. Patterson, editors, *Proc. Sixth International World Wide Web Conference*, pages 109–117, Santa Clara, CA, April 1997.

[Ram98] T. V. Raman. *AsTeR Audio System For Technical Readings*. Lecture Notes In Computer Science. Springer Verlag, December 1998.

[W3C98] World Wide Web Consortium W3C. Extensible markup language xml 1.0. Technical report, W3C, 1998. *URL* **http://www.w3.org/TR/REC-xml**.

[W3C00] World Wide Web Consortium W3C. Document object model (dom) level 2. Technical report, W3C, 2000. *URL* **http://www.w3.org/TR/DOM-Level-2**.