Emacspeak: A Speech Odyssey

T.V Raman

Thursday, August 1, 2024

1 Dedication: To My Guiding Eyes Aster, Hubbell and Tilden

2 Key Insights

1. User interface is a means to an end.
2. Open Source is essential for discovering new interaction paradigms.
3. This is not mere idealism. Openness is a key enabler for creating user journeys that were not envisioned by a system's designers.
4. Emacs and TEX are good exemplars. They permit maximal freedom when seen from the viewpoint of user extensibility and creativity. TEX enabled Audio System For Technical Readings (AsTeR); Emacs enabled Emacspeak.
5. Rapid, reliable task completion is the most important metric and trumps secondary items such as eye-candy — the latter only leads to bloat as evinced by the HTML Web.
6. Having a well-identified problem when designing a system is paramount.
7. *Usability* is important, but to matter, the system needs to be *useful* first.
8. *Ease of use* by itself is often *marketing* hype.
9. Useful systems are fun to learn and give back more than what you put in with respect to time and effort.
10. A steep learning curve in and of itself is not to be feared — it can be fun to learn and gets you farther faster.
11. True empowerment: Ensure that the user grows continuously.

3 Emacspeak — The Complete Audio Desktop

1. Emacspeak, started in September 1994, was released as Open Source in April 1995.
2. The goal was to create a system for daily use that doubled as a research work-bench for developing an auditory interface.
3. Speech and auditory output would be treated as first-class citizens.
4. The time felt right with respect to building a system that enabled eyes-free access to the emerging Web.
5. Emacspeak At Twenty was published in September 2014 and traced the evolution of the project.
6. Now, this article gives a birds-eye overview of the last 10 years by loosely following the logical structure of the *Turning Twenty* paper.
7. In the process, we identify the dreams that have come to pass as well as the expectations that have failed to materialize — **both** attributable to developments in the larger Internet eco-system.
8. But never fear, though some of these may be superficially disappointing, they likely herald the nature of bigger and better things to come!
9. As a proof-point, in 1994, I could not have imagined the impact that the world of Internet-centered computing and the accompanying information revolution would have on the state of information access.
10. Conversely, I boldly (and incorrectly) predicted that the arrival of mobile devices and mainstream speech interfaces would herald the move to a Web of information where there would be a clean separation between application back-ends and various client-specific front-ends. See Specialized Browsers and The Web, The Way You Want. Distinguished Lecture Series, UW Oct 2007.
11. The above still makes sense from the view of *scalable* software architecture. However the rapid growth of the Web economy has also resulted in an even faster race to the bottom where applications continue to be built and re-built every few years for *the next best thing* — welcome to the *write once, debug everywhere* world all over again!

12. Case in point; today we have smart phones, smart watches and smart speakers, but each of these require targeted front-ends if one wishes to bring the riches of the Internet to them.

13. So the larger the Web gets, the fewer devices it becomes available on — a classic downward spiral.

Share And Enjoy — The Best Is Yet To Come!

3.1 How To Read This Document

1. I recommend reading the *Turning Twenty* paper to get a full overview.

2. Then, read this paper a section at a time, while referring back to the parallel section in the *Turning Twenty* paper to understand how things have evolved.

3. Make sure to skim or deep-dive into the references in both papers.

4 Using UNIX With Speech Output — 2024

1. In 2024 *UNIX* equates mostly to various Linux distributions, and from the Emacspeak perspective, they are all made *mostly* equal.

2. Variations do exist and running bleeding-edge distributions can come with issues, *e.g.*, unstable versions of the underlying audio infrastructure.

3. Yes, 30 years and counting, Linux Audio is still a work in progress though I hope Pipewire will be the last of these tidal shifts.

4. Linux is moving to Wayland and expect that transition to be choppy.

5. Native applications are mostly gone bar the shouting. In this context, where most users access things through a *mainstream* Web browser, Emacspeak users access *everything* through Emacs.

6. The above when done right is hugely empowering; when done badly, it's extremely limiting — see later sections of this paper on the continuing evolution of the Web.

5 Key Enabler — Emacs And Lisp Advice

1. Advice in Emacs as implemented in `advice` is rock-solid.
2. There is a newer `nadvice` that is part of Emacs that Emacspeak does not use.
3. There are no plans to migrate to `nadvice` since that is a lot of busy work in my view and any such migration would be difficult to test for correctness.
4. The classic *advice* package may be removed from Emacs at some point in the future, but never fear; it'll be bundled with Emacspeak if that becomes necessary. This is a feature of Free Software and is a great example of what that *Freedom* entails.

6 Key Component — Text To Speech (TTS)

1. Speech output — especially unencumbered text-to-speech — is just as much a challenge as it was 30 years ago.
2. In the bigger picture, early instances of using TTS for voice assistants has driven the industry toward *natural sounding* voices.
3. The above sounds attractive on the surface, but a price we have paid is the loss of fine-grained control over voice parameters, emotion, stress and other supra-linguistic features.
4. I believe these to be essential for delivering good auditory interfaces and remain optimistic that these will indeed arrive in a future iteration of speech interaction.
5. Things appear to be coming full circle, Emacspeak started with the hardware Dectalk; now, the Software Dectalk is increasingly becoming the primary choice on Linux — see this Readme for setup instructions.
6. Viavoice Outloud from Voxin is still supported. However, you can no longer buy new licenses. If you have already purchased a license, it'll continue to work.
7. The Vocalizer voices that Voxin now sells *do not* work with Emacspeak.
8. The other choice on Linux is ESpeak which will hopefully continue to be free — albeit of much lower quality.

9. The future as ever is unpredictable and new voices may well show up — especially those powered by on-device Large Language Models (LLMs).

10. On non-free platforms, there is usable TTS on the Mac, now supported by the new SwiftMac server for Emacspeak.

7 Emacspeak And Software Development

1. *Magit* as a Git porcelain is perhaps the biggest leap forward with respect to software development.

2. New completion frameworks such as *company* and *consult* come a close second in enhancing productivity.

3. Completion strategies such as *fuzzy* and *flex* provide enhanced completion.

4. Effective Suggest And Complete In An Eyes-free Environment explains the higher-level concept involved in defining such strategies.

5. The ability to introspect code via `eglot` turns Emacs into a powerful and meaningful IDE — I say meaningful because this brings the best features of an integrated development environment while leaving behind the eye-candy that tends to bloat commercial IDEs.

6. Packages like *transient* enable discoverable, rapid keyboard access to complex nested-menu driven interfaces.

7. Ergonomic keybindings under `X` using xcape to minimize chording has been a significant win in the last two years.

8. Jupyter is the generalization of IPython notebooks to *Julia*, *Python* and *R*. The news here isn't all good; IPython notebooks are well-designed with respect to not getting locked into any given implementation. However in practice, front-ends depend on Javascript in the browser.

9. Consequently, Emacs packages for IPython Notebooks *e.g.*, package `ein`, are no longer maintained.

10. Developing in higher-level languages continues to be very well supported in Emacspeak.

11. The re-emergence of Common Lisp in the last 20 years, thanks to asdf and quicklisp as a network-aware package manager and build tool has once again made Lisp development using Emacs `Slime` a productive experience.

12. In 2022, I updated Audio System For Technical Readings (AsTeR)— my PhD project from 1993 — to run under `SBCL` with a freshly implemented Emacs front-end.

13. So now I can listen to Math content just as well as I could 30 years ago!

8 Emacspeak And Authoring Documents

1. Package `org` is to authoring as `magit` is to software development with respect to productivity gains.

2. `Org` has existed since circa 2006 in my Emacs setup; but it continues to give and give plentifully.

3. Where I once authored technical papers in *LaTeX* using `auctex`, used `nxml` for HTML, *etc.*, I now mostly write everything in `org-mode` and export to the relevant target format.

4. Integrating various search engines in Emacs makes authoring content extremely productive.

5. Integrated access to spell-checking (`flyspell`) dictionaries, translation engines, and other language tools combine for a powerful authoring workbench.

6. Extending `org-mode` with custom link types enables *smart note taking* with hyperlinks to relevant portions of an audio stream — see article Learn Smarter By Taking Rich Hypertext Notes.

9 Emacspeak And The Web In 2024

1. Package `shr` and `eww` arrived around 2014. But in 2024, they can be said to have **truly** landed.

2. 2014 also marked the explicit take-over of the stewardship of the HTML Web by the browser vendors from the W3C — I say explicit — because the W3C had already thrown in the towel in the preceding decade.

3. This has led to a Web of content created using the assembly language of divs, spans and Javascript under the flag of HTML5 — the result is a tangled web of spaghetti that everyone loves to hate.

4. In this context, see Tag Soup, Scripts And Obfuscation: How The Web Was Broken for a good overview of HTML's obesity problem.

5. For better or worse, the investment in XML and display-independent content is now a complete write-off at least on the surface.

6. So what next — wait for the spaghetti monster to show up for lunch? Humor aside that monster may well be called AI — though whether today's Web gives that monster life, indigestion, constipation, dysentery or hallucinations is a story to be written in the coming years.

7. I say *on the surface* above because The welcome re-emergence of `ATOM` and `RSS` feeds is perhaps a silent acknowledgement that bloated Web pages are now unusable even for users who can see.

8. Package `elfeed` has emerged as a powerful feed-manager for Emacs.

9. Emacspeak implements `RSS` and `ATOM` support using `XSLT`; those features now shine brighter with mainstream news sites reviving their support for content feeds.

10. Browsers like Mozilla now implement *content filters* — a euphemism for scraping off visual eye-candy and related cruft to reveal the underlying content. These are now available as plugins, (see RDRView for an example). Emacspeak leverages this to make the Web more readable.

11. Package `url-template` and `emacspeak-websearch` continue to give in plenty, though they do require continuous updating.

12. Web APIs come and go, so that space is in a state of constant change.

13. The state of web applications is perhaps the most concerning from an Emacspeak perspective, and I do not see that changing in the short-term. There are no incentives for Web providers to free their applications from the tangled Web of spaghetti they have woven around themselves.

14. But as with everything else in our industry, it is precisely when something feels completely entrenched that users rebel and innovations emerge to move us to the next phase — so fingers crossed.

10 Audio Formatting — Generalizing Aural CSS

1. Audio formatting with Aural CSS support is stable, with new enhancements supporting more TTS engines.
2. Support for parallel streams of TTS using separate outputs to left/right channels is a big win and enables more efficient interaction.
3. Support for various Digital Signal Processing (DSP) filters enables rich auditory effects like binaural audio and spatial audio.
4. Soundscapes implemented via package `boodler` makes for a pleasant and relaxing auditory environment.
5. Enabling virtual sound devices via Pipewire for 5.1 and 7.1 spatial audio significantly enhances the auditory experience.

11 Conversational Gestures For The Audio Desktop

1. Parallel streams of audio, combined with more ergonomic keybindings are the primary enhancement in this area.
2. Parallel streams of speech, *e.g.*, a separate notification stream on the left or right ear help increase the band-width of communication.
3. Notifications can thus be delivered without having to stop the primary speech output.

12 Accessing Media Streams

1. Emacspeak support for rich multimedia is now much more robust.
2. Emacs package `empv` is a powerful tool for locating, organizing and playing local and remote media streams ranging from music, audio books, radio stations and Podcasts.
3. This makes media streams from a large number of providers ranging from the BBC to Youtube available via a consistent keyboard interface.
4. This experience is augmented by a collection of *smart* content locators on the Emacspeak desktop, see the relevant blog article titled smart media selectors.

13 Electronic Books— Ubiquitous Access To Books

1. Emacspeak modules for *Epub* and *Bookshare* continue to provide good books integration.
2. There are *smart* book locators analogous to the locators for media content.
3. Emacspeak speech-enables `Calibre` for working with local electronic libraries.

14 Leveraging Computational Tools — From SQL And R To IPython Notebooks

1. This area continues to provide a rich collection of packages.
2. Newer highlights include `sage` interaction for symbolic computation.
3. Emacspeak speech-enables packages `gptel` and `ellama` for working with local and network LLMs.

15 Social Web — Mail, Messaging And Blogging

1. This is a space that is definitely regressing.
2. The previous decade was marked by open APIs to many social Web platforms.
3. Over time these first regressed with respect to privacy.
4. Then they turned into wall-gardens in their own right.
5. Finally, the Web APIs, other than the kind embedded in Javascript have started disappearing.
6. Looking back, the only *social* platform I now use is Blogger for hosting my Emacspeak Blog, it has a somewhat usable API, albeit guarded by a difficult to use *OAuth* interface that requires signing in via a *mainstream* browser.
7. IMap continues to survive as an open email protocol, though its days may well be numbered.

8. The dye is already cast with respect to mere mortals being able to setup and host their email — witness the complexity in setting up the Emacspeak mailing list in 2023 vs 1993!

9. This is an area that is likely to get worse before it gets better, thanks to the spammers — more's the pity, since Internet Email is perhaps the single-most impactful technology with respect to leveling the communications playing field.

10. The disappearance of APIs mentioned above also means that today the only usable chat service on an open platform like Emacspeak is the venerable Internet Relay Chat (IRC).

16 The RESTful Web — Web Wizards And URL Templates For Faster Access

1. This area continues to thrive — either because of — or despite — the best and worst efforts of application providers on the Web.

2. Twenty years on (this feature originally landed in 2000) Emacspeak has a far richer collection of filters, preprocessors and post-processors that enables ever-more powerful Web wizards. See the relevant chapter in the Emacspeak manual for the automatically updated list of **URL Templates**.

17 Mashing It Up — Leveraging AI And The Web

1. Developing solutions by combining various API-based services on the Web has all but disappeared, unless one is willing to commit fully to the Javascript-powered Web hosted in a Web browser, something I hope I never have to accept.

2. So for now, I'll keep well away and count my blessings.

3. The next chapter of the *mash-up story* may well be based around *Generative AI* using LLMs. In effect, LLMs trained on Web content define a *platform* for generating content mash-ups. The issue at present is that they are just as likely to produce *meaningless mush* — something that may get better as the field gets a handle on cleaning up Web content.

4. Notice that we are now back to the previously unsolved problem of cleaning up the HTML Web — with LLMs, we'll just have an order of magnitude more documents than the 2^W postulated by Beyond Web 2.0, Communications Of The ACM, 2009.

18 The Final Word — Donald E Knuth (DEK)

- The best theory is inspired by practice. The best practice is inspired by theory.
- The enjoyment of one's tools is an essential ingredient of successful work.
- Easy things are often amusing and relaxing, but their value soon fades. Greater pleasure, deeper satisfaction, and higher wages are associated with genuine accomplishments, with the successful fulfillment of a challenging task.
- Computer Programming Is An Art.

The best example of the above is of course Knuth's TEX — work that was motivated by his own dissatisfaction with the tools available to him at the time for typesetting his magnum opus — The Art Of Computer Programming (TAOCP). It is something I've looked up to ever since my time as a graduate student at Cornell.

The Emacspeak Speech Odyssey outlined in this paper is, in some small measure, my own personal experience of the sentiments he expresses.

–T. V. Raman, San Jose, CA, August 1, 2024.

19 References

1. User Interface is a means to an end, DDJ 1997.
2. GNU Emacs
3. Knuth's TEX
4. Audio System For Technical Readings
5. Announcing Emacspeak: April 1995
6. Emacspeak At Twenty

7. The Web, The Way You Want. Distinguished Lecture Series, UW Oct 2007
8. Specialized Browsers
9. An Ode To Emacspeak: The Best Is Yet To Come
10. Software Dectalk on Github
11. Dectalk setup instructions
12. Effective Suggest And Complete In An Eyes-free Environment
13. Common Lisp: asdf
14. Common Lisp: Quicklisp
15. Soundscapes on the Emacspeak Audio Desktop
16. RESTful Web
17. Ergonomic keybindings
18. Minimize chording with XCape
19. Learn Smarter By Taking Rich Hypertext Notes
20. Tag Soup, Scripts And Obfuscation: How The Web Was Broken
21. Readable Web Pages: RDRView
22. smart media selectors
23. Beyond Web 2.0, Communications Of The ACM, 2009
24. Emacspeak Manual: URL Templates
25. Beautiful Code An overview of the Emacspeak architecture, O'Reilly Media, 2007.
26. The Art Of Computer Programming (TAOCP)