

The Adriane desktop for the sight impaired

NO BARRIER

Jack Simanzlik, photocasa.com



The Adriane audio desktop system delivers Linux to sight-impaired users.

BY KLAUS KNOPPER

Linux has many beautiful desktops, including rotating cubes, wobbly and burning windows, and mouse gestures for semi-automatic functions that do terribly smart things. Unfortunately, from the beginning of mouse-oriented computing, the graphical desktop has been designed for users who work within a visual context.

Consider, for example, a nicely configured KDE desktop with tidily sorted icons. If you switched off your monitor, would you be able to start your email client and read your mail? Most users wouldn't even be able to find the program icon without visual feedback, yet this is how a graphical desktop "looks" to blind people.

Some software manufacturers sell "accessibility add-ons" for graphical desktops, which creates the impression that the entire problem is suddenly fixed by buying more – and more expensive – software. Adding speech feedback for

menus, interactive elements, and keyboards does indeed make it possible for a blind person to operate a program they could not use before, but because the program's interactive interfaces are still designed for a graphical "overview," it is still difficult and tiresome to search for a specific button or function.

Many of the tools and extensions necessary for working with disabilities such as blindness, low vision, and motor or mental disabilities are already present in open source – and are part of virtually all GNU/Linux distribution repositories. The Adriane project [1] draws these technologies together into a single audio desktop system for the sight impaired.

Accessibility vs. Equalization

Many blind computer users don't even have a choice of what software to use in a work environment. This decision is often made by an employer who has to meet the legal requirement of providing workplace accessibility. The most common decision is simply that "everyone in the company must use the same software" or some extension of it. This choice is often made without considering the actual opinion or needs of the sight-impaired employee. A proprietary PC desktop system for a blind person

costs about EUR 7,000 to 12,000, and the person who sets up the system assumes that anything this expensive *has got to be good*. But is it?

Employers who make rash decisions about barrier-free software don't just create very ineffective and frustrating workspaces, they also lose a great amount of productivity by grossly underestimating the user's capabilities. A blind person using the right hardware and software can work at least as fast and sometimes faster than a seeing person with the standard GUI. They can read, understand, and remember a full page of text in just a few seconds – while ordinary users are still determining which part of the web page is an advertisement. They can find the perfect deal and win eBay auctions before you have even found the clicking path for placing a bid – that is, if the working environment is designed to match the user's capabilities rather than as a default "accessibility bridge" for a vision-oriented layout.

Introducing Adriane

The command line is the most effective interface for working with computers because it offers a direct way of giving commands that will make the computer do exactly what you want. A direct, text-

oriented interface is focused on the content, not on layout and visual intuition.

Even with the early DOS-based systems, it was possible to display screen text to peripherals connected through serial or parallel ports, such as printers, hardware speech synthesizers, or Braille devices. Graphical desktops have, unfortunately, made life somewhat more complicated for users with disabilities.

When the Adriane team was researching the topic, working closely together with blind computer users (even beginners) – and blind software developers – we came to the conclusion that software user interfaces should adapt to the user’s capabilities, rather than forcing the user to adapt to an interface that was never intended to support blind people.

The name ADRIANE (Audio Desktop Reference Implementation And Networking Environment) describes a user interface that does not require a monitor or visual output at all but still provides an easy-to-use, step-by-step linear user interface sorted in menus according to the user’s preference. Instead of reinventing the wheel, we integrated techniques that already exist in GNU/Linux, such as screen readers, text-to-speech, Braille drivers, keyboard navigation, and programs that can be fully controlled in non-graphical mode (Figure 1).

Admittedly, entirely by chance, “Adriane” is also the first name of my wife, who was the first beta tester and co-developer from the user perspective and who, naturally, was very skeptical about usability for beginners, having had very uncomfortable experiences with computer systems for the blind. At the time she learned Braille, you definitely needed a seeing person on hand to install and work with Braille devices, and using the Internet with available tools was entirely out of question. The most a blind user could do was text processing with complicated hotkeys.

The Adriane menu sorts the most common tasks into a flat menu structure, with no commands to remember or type. The first line says, “Enter for help, arrow down for next menu,” which is a good starting point when you encounter a non-visual interface for the first time. By special request from more experienced blind computer users and programmers, we later added a *Shell* item to the first menu. In general, the menu

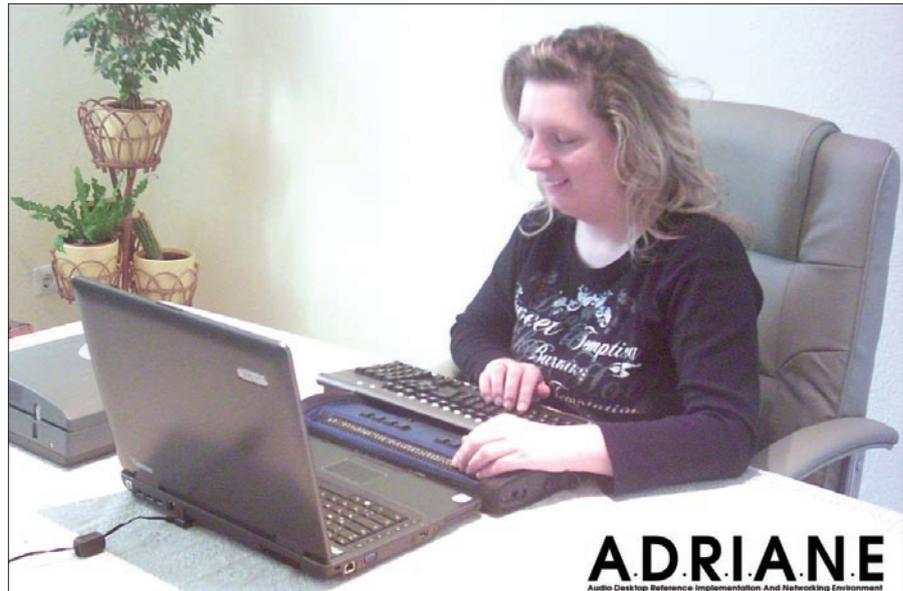


Figure 1: Adriane supports a complete desktop environment for the sight-impaired user.

is easily extensible, and it works in text mode as well as in graphical mode, thanks to dialog and Xdialog, as shown in Figure 2.

The utilities bundled with Adriane, including tools such ELinks, Mutt, Irssi, MPlayer, and Sane/OCROPUS, cover the most popular end-user activities and run easily in the text console together with the screen reader. One specialty that has not appeared in proprietary barrier-free systems yet is the option to read and send SMS text messages with a cell-phone – without any special software on the phone itself. As you can imagine, it is impossible to read or answer a text message with only the small display as feedback if you cannot see it. With GSM, the Adriane user can download SMS messages to the computer and answer them with the use of an editor and a

normal keyboard instead of the cell-phone digit keys.

Some of the programs and components used in Adriane are described in the following sections. Here, I will focus on features for blind and vision-impaired users. Additional tools that support motor or mental disabilities, including speech recognition, sticky keys, head-mouse navigation, and more do exist but won’t be covered in this article.

Braille Devices and Screen Readers

Letting the computer speak a line of text and displaying the text on a Braille device are the most common ways for blind people to learn what’s written on the computer screen. A Braille device, sometimes just called a “line” because of its restricted vertical dimensions, is a

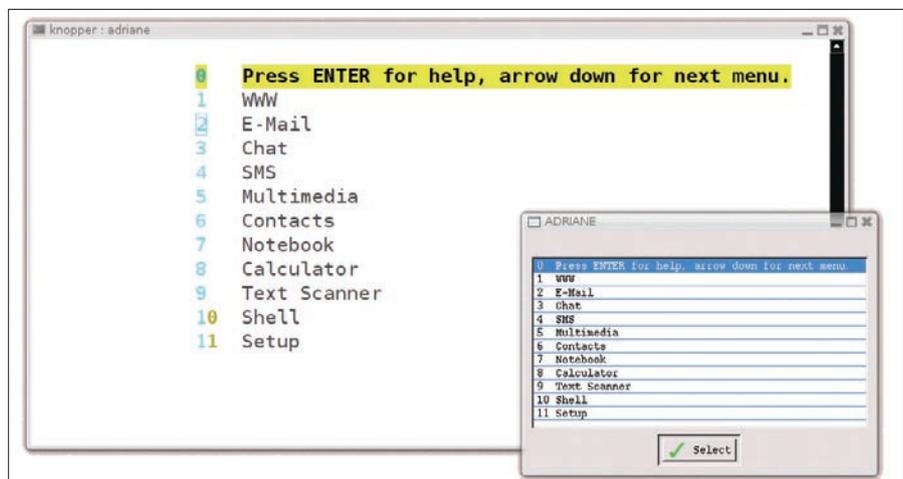


Figure 2: Adriane’s simple menus are easy to navigate and customize.

tactile display consisting of six or eight dots per letter, which can be read by touch, at least if you know Braille. Figure 3 shows part of the 1:1 translation from Braille to the English alphabet. Every language uses a different translation table, though, and with are no special symbols for numbers, letters a-j are used, sometimes prefixed by a “number symbol,” for digits 0-9.

Along with a few other more specialized options, two main screen readers and drivers for operating Braille devices and speech in Linux are brltty [2] and SUSE Blinux (SBL) [3].

Brltty is probably the best-known Braille interface driver, whereas SBL's strength lies in extending Braille- and text-to-speech support with profiles for individual applications, which makes it possible to individually customize which parts of the screen and text are displayed. SBL also allows users to navigate on the screen with Braille device keys, as well as with keyboard-only navigation, which is why SBL is the primary screen reader for the Adriane system. Adriane employs the seldom used Caps Lock key for keyboard navigation and functions for SBL, as shown in Table 1.

Using these few navigation key combinations, you can discover line-by-line what's on the computer screen, even if the monitor is turned off or you have no means of viewing it. Although it is not possible to have an overview of programs, menus, and buttons all at once, the user can still access any information line and jump to specific parts of a page using keystrokes.

Orca [4] is a screen reader that is written in python for graphical applications and uses the Gtk2 toolkit and the Assistive Technology Service Provider Interface (AT-SPI). The Orca screen reader sends text labels from menus, buttons, and mixed-text areas (such as the main panel of a web browser) to a Braille device and speech synthesizer. Orca also has a magnification feature, which did not work reliably in our tests.

Orca makes it possible to work with OpenOffice 2.3 and up with audio and Braille, provided the user knows all the keyboard abbreviations necessary to activate functions that are normally selected with the mouse. Orca not only reads the apparent “visual” text, but it also provides tooltips and meta informa-

⠠ A 1	⠠ J 0	⠠ S
⠠ B 2	⠠ K	⠠ T
⠠ C 3	⠠ L	⠠ U
⠠ D 4	⠠ M	⠠ V
⠠ E 5	⠠ N	⠠ W
⠠ F 6	⠠ O	⠠ X
⠠ G 7	⠠ P	⠠ Y
⠠ H 8	⠠ Q	⠠ Z
⠠ I 9	⠠ R	

Figure 3: The Braille English alphabet.

tion such as font family and rendering, formfield element types, and so on. Although Orca is being developed primarily for use within the Gnome desktop, it works fine with all window managers as long as the individual application supports AT-SPI. Applications that support AT-SPI include Firefox, OpenOffice, the multiprotocol chat client pidgin, and even (partly) Gimp. If you are not starting Orca from within Gnome, you need to set a few environment variables to activate AT-SPI with Gtk2 applications:

```
export SAL_USE_VCLPLUGIN="gtk"
export GTK_MODULES="gail:atk-bridge"
orca &
soffice document.odt
```

Orca has plugins for both brltty and SBL, so the same Braille driver that works in text mode can also be used for graphics mode.

Although Orca makes it possible for a blind person to work with primarily mouse-oriented programs, a graphical interface with many buttons or menus in a single GUI is not optimal or efficient for non-graphical usage. Working with graphical interfaces is still slower and

Table 1: Navigating with the Caps Lock Key

Caps Lock-arrow up	Read previous line
Caps Lock-arrow down	Read follow-up line
Caps Lock-page up	Read from top of page
Caps Lock-page down	Read to bottom of page
Caps Lock-f/s	Speak faster/slower
Caps Lock-l/p	Switch language

more complicated for blind and vision-impaired users than for those with vision. The real disaster happens when the program is minimized or the program window loses focus because of another application or pop-up. The window then becomes inaccessible to the screen reader until it gets the focus again, and to the user, it is even more “invisible.” Unless you know how to tell the window manager how to get applications de-icified and focused (Alt + Tab for some), it is unclear to a user without visual control whether the program just lost focus and disappeared or whether the program or screen reader crashed because of a software error. Therefore, the primary choice of interface for blind computer beginners is still the text console, which never loses focus and always provides “fullscreen mode” for each program.

The screen reader sends text to a Braille device or speech synthesizer, but input is still typed on an ordinary keyboard. Although a blind person cannot see what's actually written on each key, you might have noticed that every keyboard, even yours, has small bumps on the *f* and *j* keys that provide some locational orientation for a blind person. This also applies to telephones, on which the number 5 usually is marked. When preparing a keyboard for a blind person, technicians, with the use of a soldering iron, often mark additional keys with touchable dots, which gives additional orientation.

Text to Speech

To get the computer to read text displayed onscreen, you need a text-to-speech synthesizer. The linguistic theory behind how to make high-quality spoken text from written text fills a few books by itself. A few phonetic rules tell how written text (combinations of letters and syllables) is pronounced correctly, and programs can track several thousand exceptions that occur frequently. Just concatenating sounds leads to incomprehensible gibberish; therefore, high-quality text-to-speech tools pick larger (*unit selection*) or smaller (*diphone* or *half-syllable synthesis*) parts of prerecorded speech from a huge database. Recording and labeling prerecorded text means a huge amount of work; unfortunately, the result is seldom covered by a license that allows unrestricted distribution.

Festival [5] is a sophisticated speech analyzer and synthesizer, but creating a speech database and ruleset for it is not easy because Festival uses a lisp-like syntax and requires a diphone database of about 3000 audio text snippets, cut and extended by pitchpoints as glue. Only a few free recorded voices are available for festival so far. Mbrola, a binary-only “free of cost for non-commercial use” speech synthesizer, managed to get a lot of speech database contributions, but its license does not allow free distribution for any purpose, and it is not compatible with open source licenses. Some proprietary text-to-speech systems are available, but the focus here is on open source and free software.

The best choice for Adriane is eSpeak [6], which has a small memory and CPU usage footprint, speaks more than 30 languages, and is easily extensible. eSpeak has an entirely synthetic approach, with no recorded or natural-sounding voices included, so it sounds somewhat “robotic”; on the other hand, it’s free of any proprietary claims.

For coordinating text-to-speech resources, Speech Dispatcher [7] is now part of many accessibility add-ons. Speech Dispatcher can interrupt the output of a long text with a shorter message of higher priority, then return to the initial text afterwards, optionally in different voices or sounds. SBL and Orca (in their current versions) both take advantage of Speech Dispatcher to provide speech capabilities to different programs.

All accessibility back ends, such as Speech Dispatcher, kbdsniffd (the keyboard navigation driver), and SBL (the screen reader), are started by *adriane-screenreader* in the correct order.

Magnification and Color

Although you might think that 3D window managers such as Compiz Fusion are not good for vision-impaired people, Compiz actually contains a few useful accessibility extensions for users with low vision. The focus-tracking full-screen magnifier is a practical tool that I haven’t seen anywhere else but in Compiz. You might already know about the built-in resolution switch of Xorg: By hitting Ctrl + Alt + + or Ctrl + Alt + -- on the numeric keypad, you can change the resolution and scroll on the virtual screen with the mouse. Although this feature

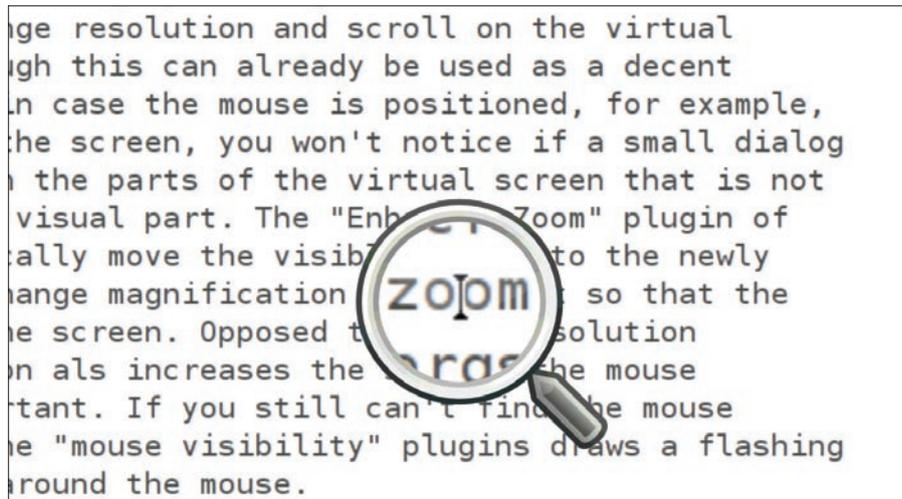


Figure 4: A Compiz plugin lets you magnify the text around the mouse cursor.

can already be used as a decent magnifier, if the mouse is positioned, for example, at the top left corner of the screen, you won’t notice if a small dialog window appears in part of the virtual screen that is not currently magnified. The Compiz Fusion Enhanced Zoom plugin will move the visible screen to the newly focused window, and it will change magnification to zoom out so that the window frame fits the screen. Unlike the Xorg resolution switcher, this magnification also increases the size of the mouse pointer. If you still can’t find the cursor location on your screen, the “mouse visibility” plugin draws a flashing and rotating ring of fire around the cursor.

A second magnification plugin only enlarges an area around the mouse pointer, which might be preferable if you like an overview of the entire screen with magnification of details at the same time (Figure 4).

For certain types of colorblindness, it is possible to exchange certain colors with others, choosing from various tables, or just inverting the entire color table (which is also a practical feature for presentations in which contrast is insufficient).

Accessible Browsers

ELinks, an “experimental” fork of the Links browser, provides some features

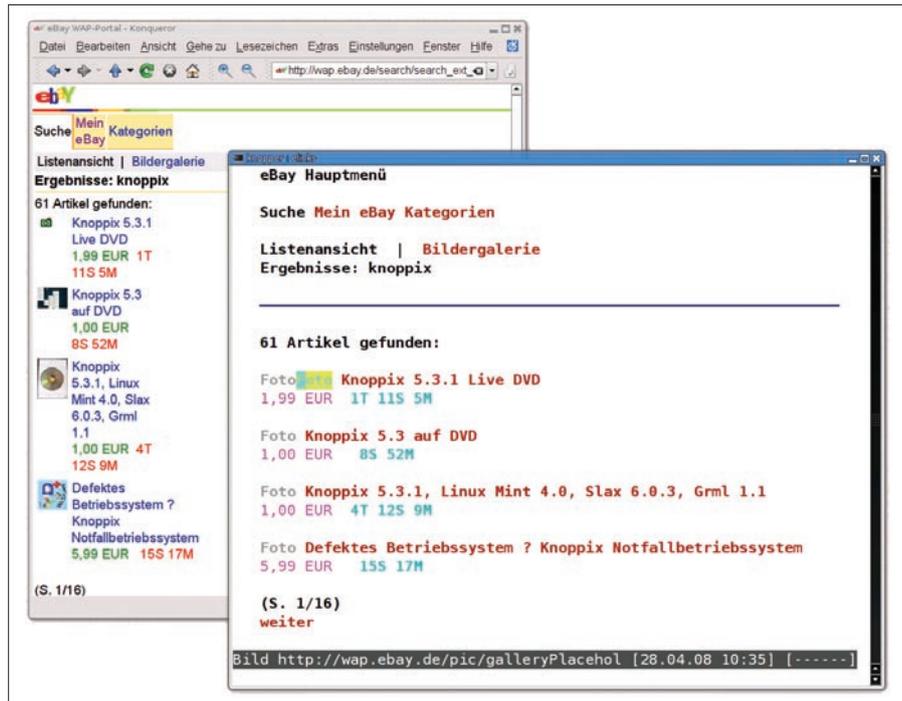


Figure 5: eBay’s WAP portal on Konqueror and ELinks.

that are practical for a text-oriented interface [8]. ELinks supports cascading style sheets and JavaScript, which allows you to enter some web pages that refuse to work without JavaScript support. SBL is set up so that ELinks reads only text marked by `<a href>` tags by default, and no plain text in between. This approach allows quick navigation by just browsing and following links first; finally, after reaching the desired page, the user can have the screen read in its entirety.

Graphical buttons or non-text symbols simply cannot be displayed as a “letter,” therefore creating barriers for users with no vision. “Captchas,” pictures with barely recognizable text that is supposed to be typed into an input field to ensure the input is coming from a human, are an example of how to create artificial barriers involuntarily. Although ELinks doesn’t have much chance of circumventing artificial barriers, it provides some tools for accessing “invisible” form elements, as well as sending a form without a submit button.

Figure 5 shows a comparison between Konqueror and ELinks, both using the less overloaded WAP version of eBay [9].

Pictures and graphical elements in a web page can still be “seen” in a text-only environment, if a textual description is available through some form of meta-information (“textarea,” “submit button,” etc.) or titles and labels, for which the ALT parameter inside the

`` tag is used. If no description is present, pictures are “invisible” unless some kind of Optical Character Recognition (OCR) software discovers written text in the picture.

ELinks can call a framebuffer image viewer like `fbi` to display pictures on the text console – as graphics for a seeing helper. In the same way, videos can play on the frame buffer with `MPlayer`, so full multimedia support does not rely on Xorg being active all the time.

Text Recognition

Also, you can use OCR software to convert paper mail into digitally readable text. For quite a while, `GOOCR` was the only free software tool that converted scanned pictures into text by recognizing letters.

Now Google has started a new open source project called `OCROpus` [10], which consists of a layout analysis that separates a page of scanned text into consecutive areas or columns, as well as an OCR engine based on `Tesseract` that does recognition and probability-based enhancement of text. The development version of `OCROpus` already produces very good results in most cases, so you can use the combination of `OCROpus` with the `Sane` scanning tool [11] to scan and read letters and books.

Getting Adriane

The Adriane system is available on the Knoppix Live CD or DVD starting from

version 5.3 and up via the *adriane* boot option. Remastering with Adriane as a default option is possible by changing `boot/isolinux/isolinux.cfg`.

With SBL 3.2.1, SBL author Marco Skambraks has added an additional keyboard daemon for onscreen navigation, so the formerly required *keyboard sniffer* kernel patch is no longer needed; the *uinput* linux kernel module is sufficient. Therefore, Adriane should now also be installable as an add-on for Debian by just installing the packages as described on the Adriane homepage [1].

After having fixed the Xorg configuration that came pre-installed on the Asus EeePC, which was not set up for composite with AIGLX, we experimented (Figure 6) and created a bootable SD flash with Adriane, Orca, and Compiz Fusion.

Together with an additional marked USB keyboard and (for users with low vision) a sufficiently large TFT display, putting Adriane on bootable flash memory stick makes quite an inexpensive, accessible, and portable workspace.

Conclusion

Although the Linux community provides a lot of great helpers, tools, and toolkits to support users with disabilities, just getting all of these programs to work together nicely usually means a lot of work, unless they come preinstalled and preconfigured. Adriane builds all the necessary tools into a single, easy-to-use audio desktop. ■



Figure 6: Adriane brings barrier-free computing to the low-cost EeePC laptop.

INFO

- [1] Adriane Project: <http://knopper.net/knoppix-adriane/index-en.html>
- [2] brltty: <http://mielke.cc/brltty/>
- [3] SUSE Blinux: http://en.opensuse.org/SUSE_Blinux
- [4] Orca: <http://live.gnome.org/Orca>
- [5] Festival: <http://www.cstr.ed.ac.uk/projects/festival/>
- [6] eSpeak: <http://espeak.sourceforge.net/>
- [7] Speech Dispatcher: <http://www.freebsoft.org/speechd>
- [8] ELinks: <http://elinks.or.cz/>
- [9] WAP version of eBay: <http://wap.ebay.com>
- [10] OCROpus: <http://code.google.com/p/ocropus/>
- [11] Sane: <http://www.sane-project.org/>