

Accessible User Interfaces

Milan Zamazal

Brailcom, o.p.s.

The task

To allow handicapped users to use (if possible) computers in all the important application areas.

The means

Providing accessible user interface for a set of applications covering all common *tasks* the handicapped users need to do. It's not necessary to provide an accessible user interface to every application, although more options are always welcome.

Accessible user interface can work with:

- Speech and braille output.
- Customized visual output.
- Alternative ways of input.
- ...

Different from standard UIs, but working with the same data and actions.

The Emacs way

- Emacs – complete environment: Text editor, file manager, shell prompt, MUA, News reader, Web browser, IRC client, Jabber client, CD player, ...
- Uniform interface.
- Can be completely speech enabled: Emacspeak, speechd-el.

Prerequisites

- Good speech output system (Speech Dispatcher).
- Extension mechanism, real one (Elisp).
- Uniform design (limited set of core I/O functions, applications don't abuse provided mechanisms).
- Completely Free Software.

It's easy!

We have successfully made it in the Free(b)soft project – speechd-el:

- Not much more than 1000 lines of code.
- It works.
- Absolutely no direct change of existent code.
- Lack of particular Emacs extensibility usually compensated by other features.

But ...

- On the contrary to the popular belief, Emacs is not an operating system.
- There are certain limitations (no threads, slow code execution, ...).
- Some applications are missing and unlikely to be written anytime soon, if at all (modern Web browser, office suite, ...).

So Emacs can't satisfy the needs of handicapped users completely.

Possible solutions

- Improve Emacs and write the missing applications.
- Make important non-Emacs applications accessible.

What's available

Applications built on top of uniform environments and particular wide spread libraries are good candidates for making accessible user interfaces to them.

- GNOME/GTK+
- KDE/Qt

Looks like an obvious and easy choice.

AT-SPI

- API providing common view of application elements and actions.
- The ATK library and GTK+ (and future Qt) bridge available.
- Building accessible user interfaces with the help of AT-SPI possible.

Looks promising, but ...

It doesn't work!

- Mozilla has only very rudimentary and mostly unusable AT-SPI support.
- AbiWord and Gnumeric don't provide the important information at all.
- The same applies to OpenOffice.org on Free Software platforms.

Is there anything available, that:

- works,
- has no equivalent runnable in Emacs?

What's wrong?

- Decomposition of the problem seems to be fine: application – accessible library – AT-SPI – accessible user interface.
- Are the particular components designed well enough?
- Are the applications designed well enough?

How speechd-el works

- Some of the most basic commands (basic movement commands, character insertion, killing, ...) are wrapped directly.
- Messages and minibuffer prompts are handled.
- Changes and movements are reported, in a customizable way.
- Each invoked user command cancels previous speech output and reports the command keys.
- Emacs text properties are considered.

That's basically all, no applications are modified.

What's wrong outside Emacs?

Possible problems are:

- Applications written in an unclean way – not using appropriate widgets or UI building methods? Properly written application shouldn't have to support accessibility itself.
- ATK library unfinished?
- GTK+ ATK bridge unfinished?
- Other problems?

We need to identify the problems in order to fix them.

Observations

- Application window out of the main application widget tree (Mozilla).
- The most important application area inaccessible (AbiWord, Gnumeric, OpenOffice.org).
- Text properties not reported to AT-SPI (Mozilla).
- Unnamed widgets (most GTK+ applications).
- Instability.

Accessible user interfaces

Fixing the *application – toolkit – AT-SPI* chain is absolutely necessary. The other AT-SPI side is more open, alternative accessible user interfaces can be created.

For instance:

- Application readers: Orca or gspi or Gnopernicus + gnome-speech or Speech Dispatcher.
- Emacs AT-SPI interface – can turn all applications supporting AT-SPI into Emacs applications.

Potentially useful to non-handicapped users too.

Questions – accessible user interfaces

- Why are there no important applications seriously supporting AT-SPI available?
- How to fix this situation?
- Which applications do the handicapped users miss most of all?
- How about language support?
- What accessible user interfaces would we like to have?

Questions – more general

- Which accessibility components can be shared by different environments (e.g. Speakup, command line tools, Emacs, GNOME/GTK+, KDE/Qt)?
- How can we cooperate?
- Do the handicapped users care?
- Are there the necessary resources to provide the handicapped users all the Free Software tools available?

Happy hacking!