How does software accessibility work?

How to make applications accessible?
(or rather, how to make accessible applications)

Samuel Thibault
http://brl.thefreecat.org/
http://liberte0.org/
Outline

- Introduction to accessibility
- Hardware
- Software interfaces
- Guidelines
- Tools
- Discussion
Color blindness: 8% male, 0.5% female
What is accessibility?

AKA a11y

Usable by people with specific needs

- Blind
- Low vision
- Deaf
- Colorblind
- One-handed
- Cognition (dyslexia, attention disorder, memory, ...)
- Motor disability (Parkinson, ...)
- Elderly

See Accessibility HOWTOs

- You

“Handicap” depends on the situation and is not necessarily permanent
Hardware
Hardware

- Braille input/output
- Speech synthesis
- Joysticks
  - Basically replace mouse
- Press button
  - On-screen virtual keyboard
- Eye-tracking
- ...

Don't focus on one technology

Even for a given disability

• Braille is not perfect
  – A lot of blind people can't read braille
  – Braille devices are very expensive (several k€)

• Speech synthesis is not perfect
  – Noisy environments
  – Tedious for spelling issues
Piezo braille cell

- Usually 8 dots \(\sim\) one character
- Piezoelectric effect to move up/down
Braille devices

- Serial, USB, bluetooth connection
- 12 / 20 / 40 / 80 cells, price $\approx 150 \times n$ €
Software interfaces
Dedicated software?

- **e.g. edbrowse, a blind-oriented editor/browser**
- **Generally a bad idea!**
  - Oriented to just one disability
  - Lack of manpower
    - **e.g. Web browser**
      - javascript/flash/table/CSS support?
    - **e.g. An office suite**
      - MSOffice/OpenOffice compatibility?
  - Disabled & non-disabled working together
    - Better use the same software

➡️ Better make **existing** applications accessible
Status in a few words

- Text mode is generally quite well accessible
  - But not so well suited to beginners
- Gnome quite accessible
  - Gnome 3 was however almost a restart-from-scratch
- We're late compared to the Windows world
  - We started less than a dozen years ago
  - They started a couple of decades ago
- We're Stone Age compared to the Apple world
  - Really good and integrated support
Generic methodology

Application

Abstract representation

Visual Rendering

Accessibility bus

Screen reader

Accessibility device

Registry
Abstract representation

- Window
  - Vertical container
    - Menu bar
      - File Menu
        - Open Menu Item
        - ...
      - ...
    - Horizontal container
      - Text area
      - Ok button
X accessibility, AT-SPI

X server

pixmap

pango

gtk

atk

Orca

braille, speech, ...

text

AT-SPI (bonobo/dbus)

gedit
Technically speaking

- A lot of applications are already technically accessible
  - Console
  - GTK
  - KDE-Qt4/5 ("Real Soon Now")
  - Acrobat Reader

- A lot are not
  - KDE-Qt3
  - Xt
  - Self-drawn (e.g. xpdf)
In practice

- A lot of technically-accessible applications actually aren't really usable
  - A visually-organized mess of widgets...

<table>
<thead>
<tr>
<th>First name:</th>
<th>Foo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last name:</td>
<td>Bar</td>
</tr>
<tr>
<td>Password:</td>
<td>baz</td>
</tr>
</tbody>
</table>
In practice

- A lot of technically-accessible applications actually aren't really usable
  - A visually-organized mess of widgets...
    
    First column
    - Label First Name
    - Label Last Name
    - Label Password

    Second column
    - Text Foo
    - Text Bar
    - Text Baz
In practice

- A lot of technically-accessible applications actually aren't really usable
  - A visually-organized mess of widgets...
    - Label First Name for Text Foo
    - Label Last Name for Text Bar
    - Label Password for Text Baz
In practice

- A lot of technically-accessible applications actually aren't really usable
  - A visually-organized mess of widgets...

  First column
  - Label First Name
  - Label Last Name
  - Label Password
  Second column
  - Text Foo
  - Text Bar
  - Text Baz
In practice

- A lot of technically-accessible applications actually aren't really usable
  - A visually-organized mess of widgets...

  First column
  - Label First Name
  - Label Last Name
  - Label Password

  Second column
  - Text Foo
  - Text Bar
  - Text Baz

  ➔ Screen reader “Script” for each application
Don't try to make applications accessible, just make accessible applications

Quite often just a matter of common sense from the start

Not a reason for not fixing your existing apps of course, it will just be a bit harder :)
Text applications

• Usually work really great for braille output

• Always provide such equivalent of graphical applications, e.g. based on same shared lib
  – Useful for servers via ssh too!

• The default output of screen readers is what the cursor is on
  – Works great with shell, editor, etc.
  – Doesn't work so great with semigraphical apps

→ Put the cursor appropriately!
  – Even when invisible, e.g. mutt, aumix
Graphical applications

- Design your application **without** gui in mind first
  - Logical order, just like CSS 😊
- Use standard widgets
  - e.g. *labeled* text fields
  - Avoid homemade widgets, or else implement atk yourself for them
  - Always provide alternative textual content for visual content
- Keep it simple!
  - Not only to make screen reading easier, but to make life easier for all users too!
Some pitfalls and advices
(from the accessibility howtos)

- Shouldn't *have* to use the mouse for anything
- Care of contrasts, configurable colors
- Avoid timing-based actions, or make them configurable
- No 2D organization, logical organization
- Keep it simple and obvious
- ...
Tools
Test it yourself! (textmode)

Brltty + gnome-terminal

- see doc on http://brl.thefreecat.org
Accerciser

Check that the tree of widgets looks sane and is complete

Text, notably
• Accessibility HOWTOs
  - Quite old, but still very useful advices
• Gnome Accessibility devel guide
  - For GTK applications
Discussion
This is all about freedom #0

“The freedom to run the program, for any purpose”

What about being able to use the program?

- RMS said a11y was just a “desirable feature”.
  - “Desirable” only, really?

- RMS said “this is free software, you can modify it” (freedom #1)
Why is accessibility so hard?

- Vint Cerf asked in Communications of the ACM November 2012:
  
  “Why is accessibility so hard?”

- Issues are mostly not technical, actually
A question of priority

• Should be prioritized
  – Just like internationalization
A question of who doing it

- Concerns only a small fraction of population
  - Already a hard time using computers...
  - Almost nobody with both disabilities and programming skills
  - Almost nobody with awareness and programming skills either
    → “This is free software, you can modify it” can not work.

- Support has to be integrated
  - Distributed among maintainers themselves
  - Not borne by the tiny a11y community
About bugs

- Take users suggestions into consideration
  - E.g. bracketed links in text web browsers
- Be patient with disabled people
  - It's not easy for them to use your software
  - It's even more difficult for them to explain their problems in an understandable way
    - e.g. “braille doesn't follow”

➤ Discuss!
Conclusion

- Accessibility is a concern for a lot of people
  - 10% have major concerns
  - 20% have minor concerns
- Dealing with it usually boils down to common sense
- It very often actually also helps other users
- But we need to raise awareness of this