



# Freedom #0 for everybody, really?

*We may be failing by 10-20% actually*

*Well, 100% actually...*

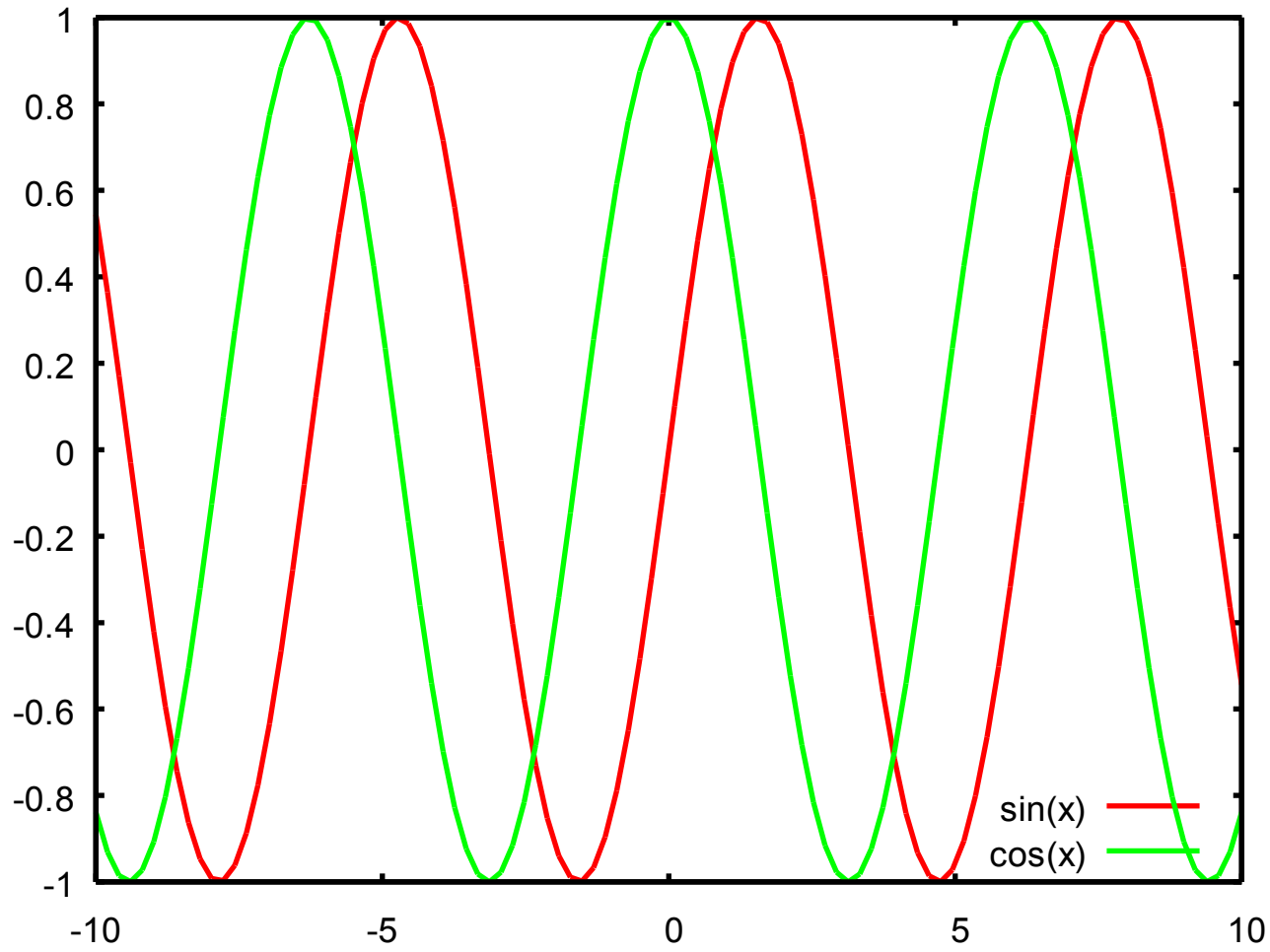
Samuel Thibault <samuel.thibault@ens-lyon.org>,  
Jean-Philippe Mengual

Slides available on <http://brl.thefreecat.org/>

<http://liberte0.org/> [liste@liberte0.org](mailto:liste@liberte0.org) (french/english?)



# Gnuplot



Color blindness: 8% male, 0.5% female



- Introduction to accessibility
- Technological overview
- The problem is *not* really technical actually
- *Discussion*
  
- More details available in LSM videos
- Links on <http://brl.thefreecat.org/>
- Notably LSM 2010, “How to make applications accessible?”



# 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

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





# 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 <sup>7</sup>



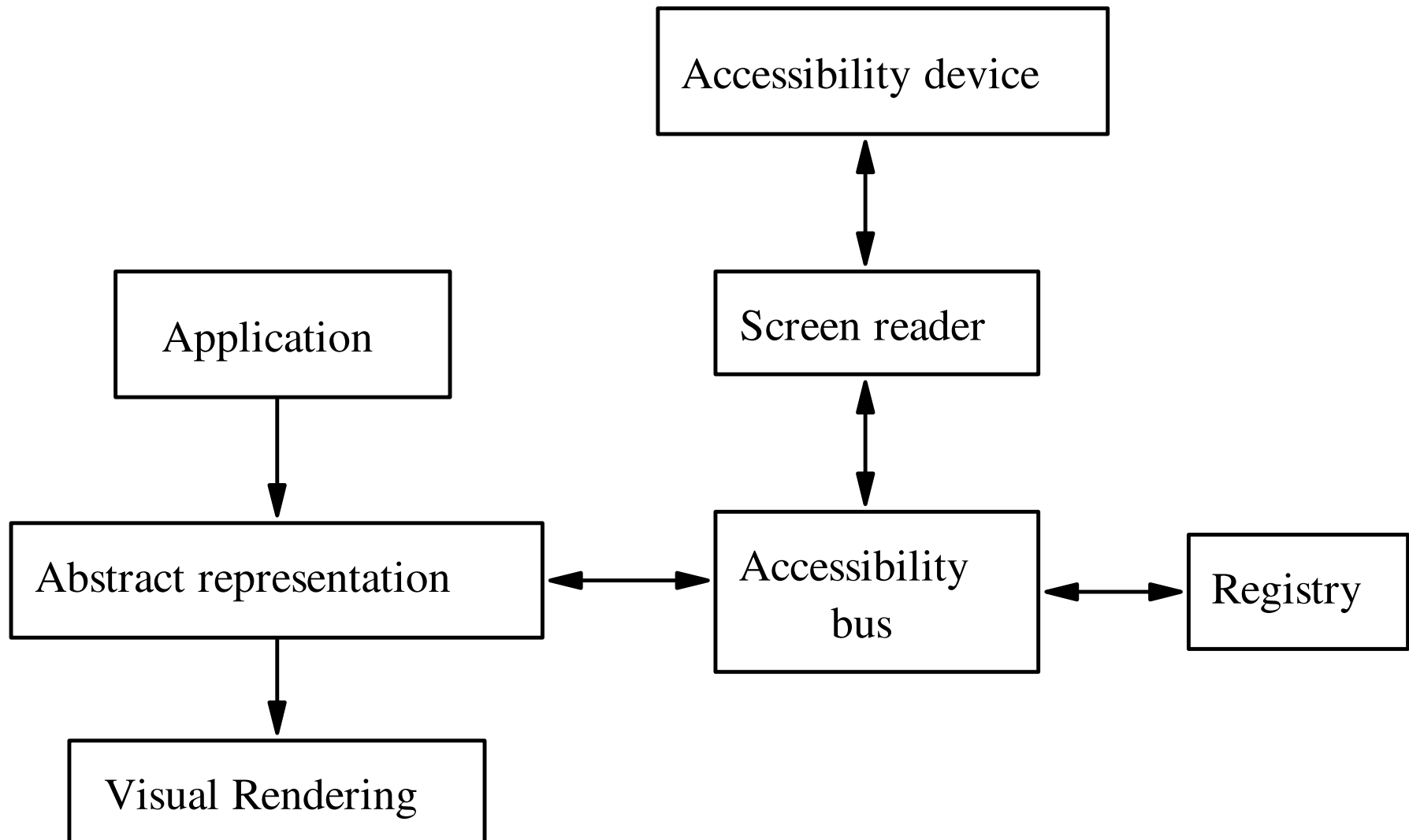
# Status in a few words

- 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



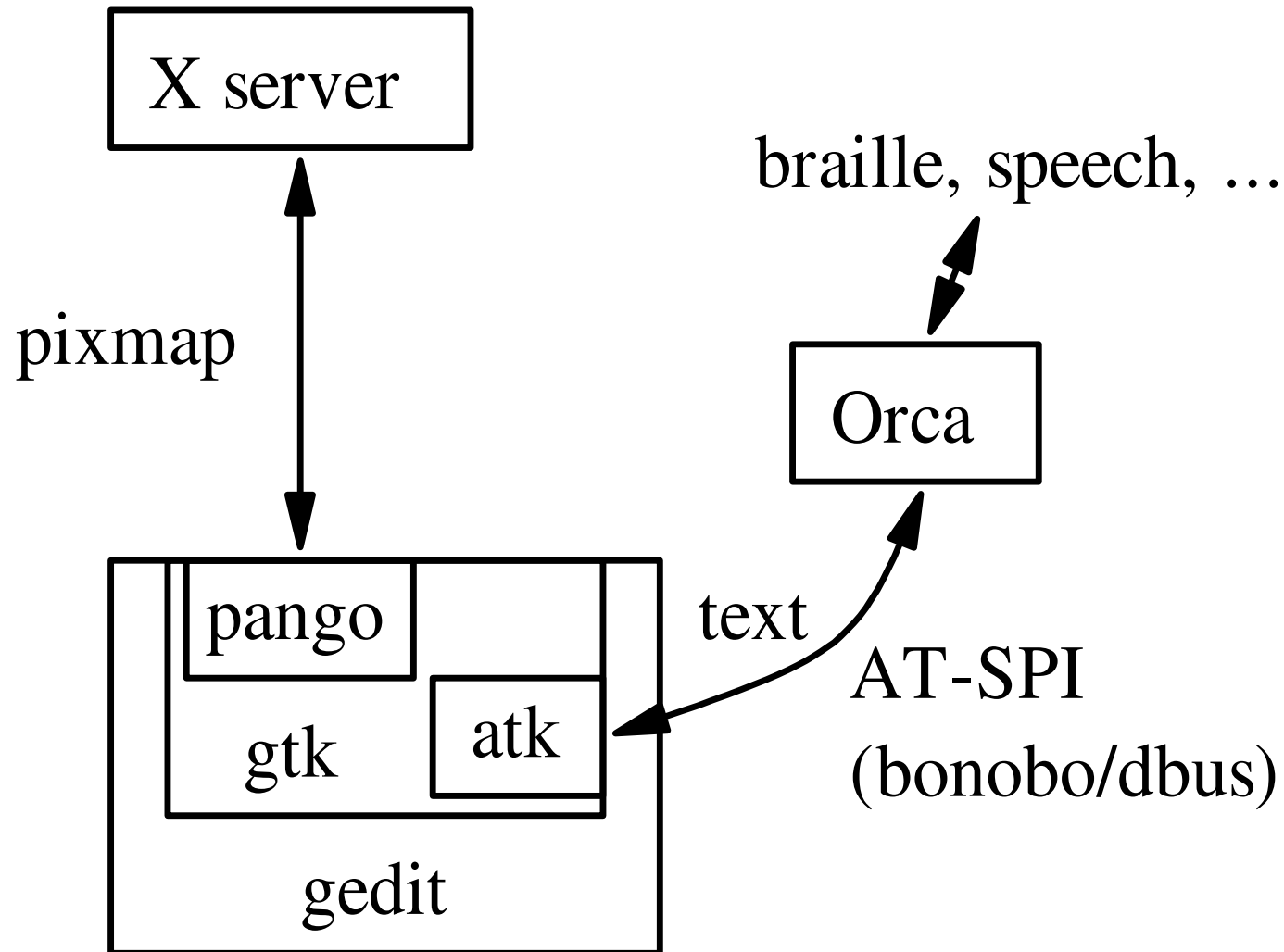


# Overview





# X accessibility, AT-SPI





# Technically speaking

- A lot of applications are already technically accessible
  - Console
  - GTK
  - KDE4 (“Real Soon Now”)
  - Acrobat Reader
- A lot are not
  - KDE3
  - 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...

First name:	Foo
Last name:	Bar
Password:	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 :)



# Opening the discussion with a few items



# 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)
  - Can. Not. Happen.



# A question of freedom

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

What about being *able to use* the program?

- #0 only a *legal* freedom?
- #0 only a *technical* freedom?
- #0 rather about *people* freedom?

30 years of GNU has pretty much given us legal and technical freedom, what about focusing on people for the next decade?



# 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 “should” or “must” in the GNU Coding Standards?
  - Evaluated on integration into the GNU project?
  - ...



# 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

# Making applications accessible?

- Extremely diverse needs
- Generic screen reading interface
  - Already supported by toolkits like gtk
  - Requires logically built applications
- Do not make applications accessible, make accessible applications
  - From the start!
- Usually benefits everybody in the end
  - Better overall design
  - Nicer for all users: colors, contrast, tab, ...

# “Accessibility of accessibility” ...

“How to make accessible applications?”

- Very-few to many communication
- Just documentation?
  - A HOWTO already exists, but mostly unknown
- Talks at conferences?
- Integrate in standard procedures?
- Testing?
  - Doesn't actually really require hardware
  - Can only be partial, but still a very good start



# Developer/User communication issues

- You all know that
- Even more difficult when disabilities come into play...
- Don't even know/understand what they are supposed to “see”
- Patience, discussion, finding intermediary, ...?

# Graal: accessibility everywhere

- Using a computer at the library, the airport, the university practice room, etc.
  - First ask admin to install & configure software?!
- Installed by default, ready for use
  - Requires very close integration
  - E.g. support in Debian Installer



# What to do?

- [accessibility@gnu.org](mailto:accessibility@gnu.org) mailing list
  - Petered out quickly
  - Revive?
- Raise awareness?
- Document?
- Integrate?





# Discussion

- Accessibility everywhere, goal of next decade?
- A question of priority
- Usually benefits everybody in the end
- A question of who doing it
- Making accessible applications?
- Accessibility of accessibility...
- Developer/user communication issues
- A question of freedom #0
- What to do?



Backup slides



# Linux Console accessibility



# 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 semi-graphical 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
- ...



# 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!



## About bugs (2)

- Try to keep in mind their disability and their consequences
  - Yes, blind users don't care that the framebuffer doesn't show up properly!
- You could even contact your local institutes for disabled people, to discuss directly with users





# Test it yourself! (textmode)

Brltty + gnome-terminal

- see doc on <http://brl.thefreecat.org>



# Test it yourself! (guis)

## Accerciser

- Check that the tree of widgets looks sane and is complete

The screenshot shows the Accerciser application window with the following components:

- Widget Tree:** A table listing widgets with columns for Name, Role, and Children. The 'File' menu bar is selected.
- Interface viewer:** A tree view showing the widget hierarchy, with 'Selection' expanded to show 'File', 'Edit', and 'View'.
- IPython console:** A terminal window showing the following code and output:

```
In [14]: acc.parent
Out[14]: <CORBA.Object 'IDL:Accessibility/Accessible:1.0' at 0x87cd2e0>
In [15]: [child.getLocalizedName() for child in acc]
Out[15]: ['menu', 'menu', 'menu', 'menu', 'menu', 'menu', 'menu']
In [16]: acc.getLocalizedName()
Out[16]: 'menu bar'
In [17]: acc.getR
acc.getRelationSet
acc.getRole
acc.getRoleName
In [17]: acc.getR
```

Path: 0 0 0



# 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