
The toolkit developer's perspective on X

Owen Taylor
otaylor@redhat.com

Boston, MA
April 2004



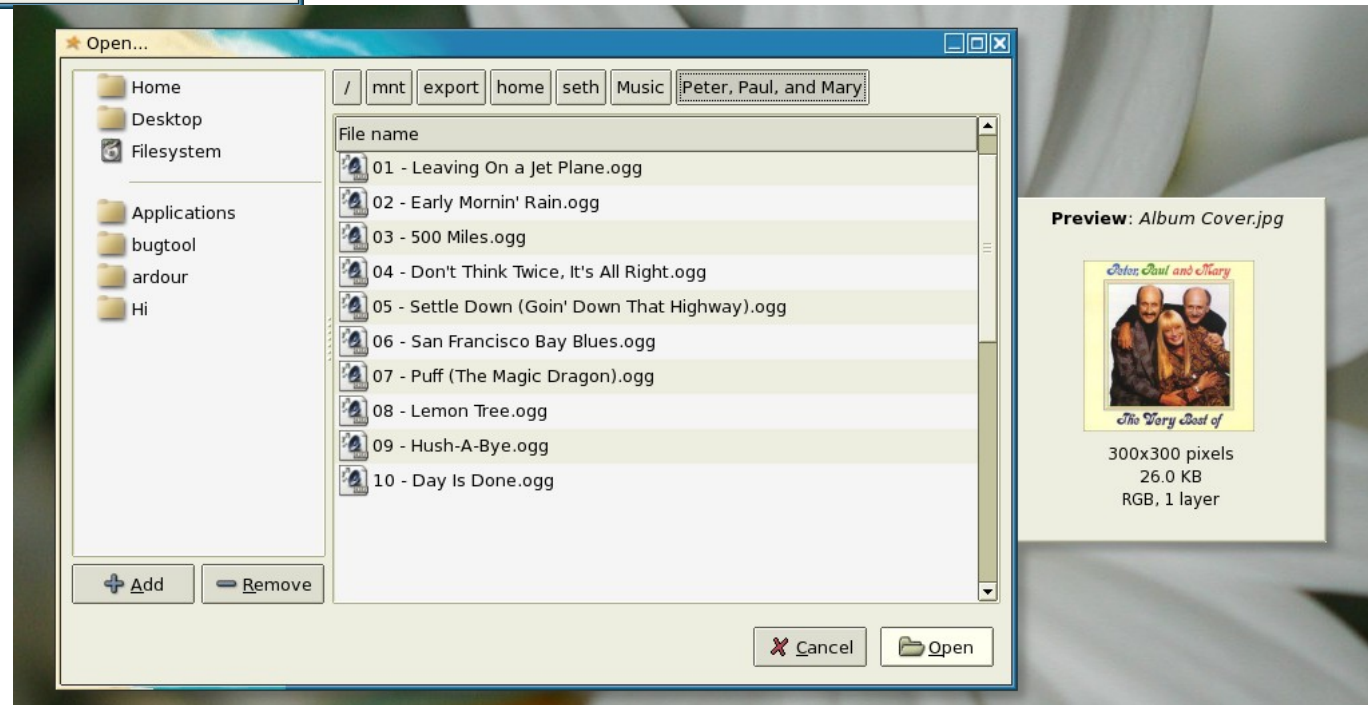
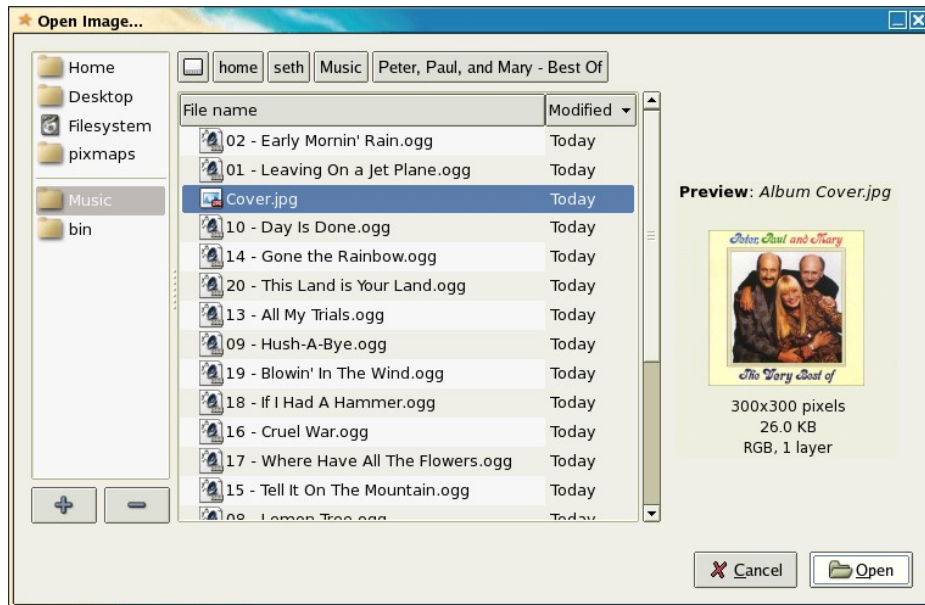
Trends in hardware

- Large screens: 21" or more
- Moderate resolutions (~5M pixels)
- Lots of memory
- Huge amounts of pixel-pushing power
- Diversity of form factor

Trends in user interface

- Movement away from strict overlapping windows
 - Popups
 - Alpha transparency
- Proritization of information
 - Computer is actively seeking information
- Explanation
 - Computer is acting on behalf of user

Window-boundary example



User experience in 5 years

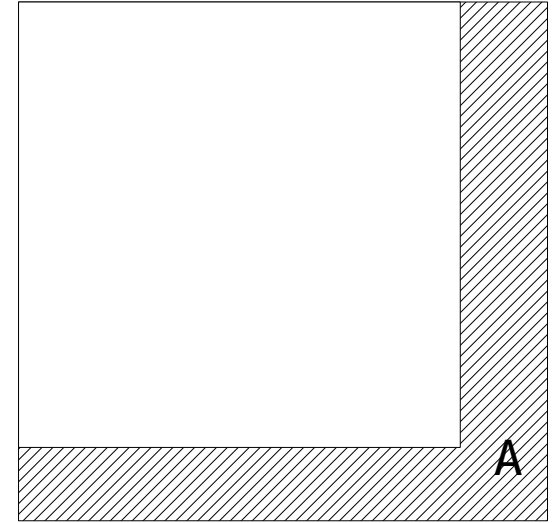
- Tight integration of multimedia
- Actions animated where appropriate
- Richly detailed desktop

User experience now

- Still many areas of improvement
- No excuse for:
 - Visible redraws
 - Inconsistent response times
 - Cut-and-paste / DND not “just working”

Example: anti-expose

- Window resized
- GTK+ marks new area “invalid”, app redraws
- Redrawn area saved with request serial (*anti-expose*)
- Area subtracted from subsequent expose event



Xft1 was a success

- Got code, user improvements, out there quickly
- Pango handled core or Xft fonts

Extensions

- Used:
 - RENDER
 - XKEYBOARD
 - MIT-SHM
 - SHAPE
 - XInput
 - RANDR
 - Xinerama
 - SYNC

Successful extensions

- Do things that apps can't do themselves
- Don't force particular architectures on apps
 - Common toolkits are cross-platform
- Provide obvious benefits to applications
- Beauty doesn't generally matter

Performance

- Most apps do very few things
 - Draw solid rectangles
 - Draw images
 - Draw text
- Need benchmarks that accurately simulate
- Point-to-point
 - How long does it take to map and redraw a menu

Conclusions

- Toolkits shield the user from the imperfections (and perfections) of the windowing
- If it isn't deployed we can't use it

Discussion Points

- Do we need new latency-reduction infrastructure?
- Should toolkits be moving toward retained-mode for rendering? (what is role of SVG?)
- What windowing system enhancements are useful?
 - Output-only windows?
 - Better input-redirection? (Grabs are annoying)
 - Cover events?