



Linux Audio Usability Issues

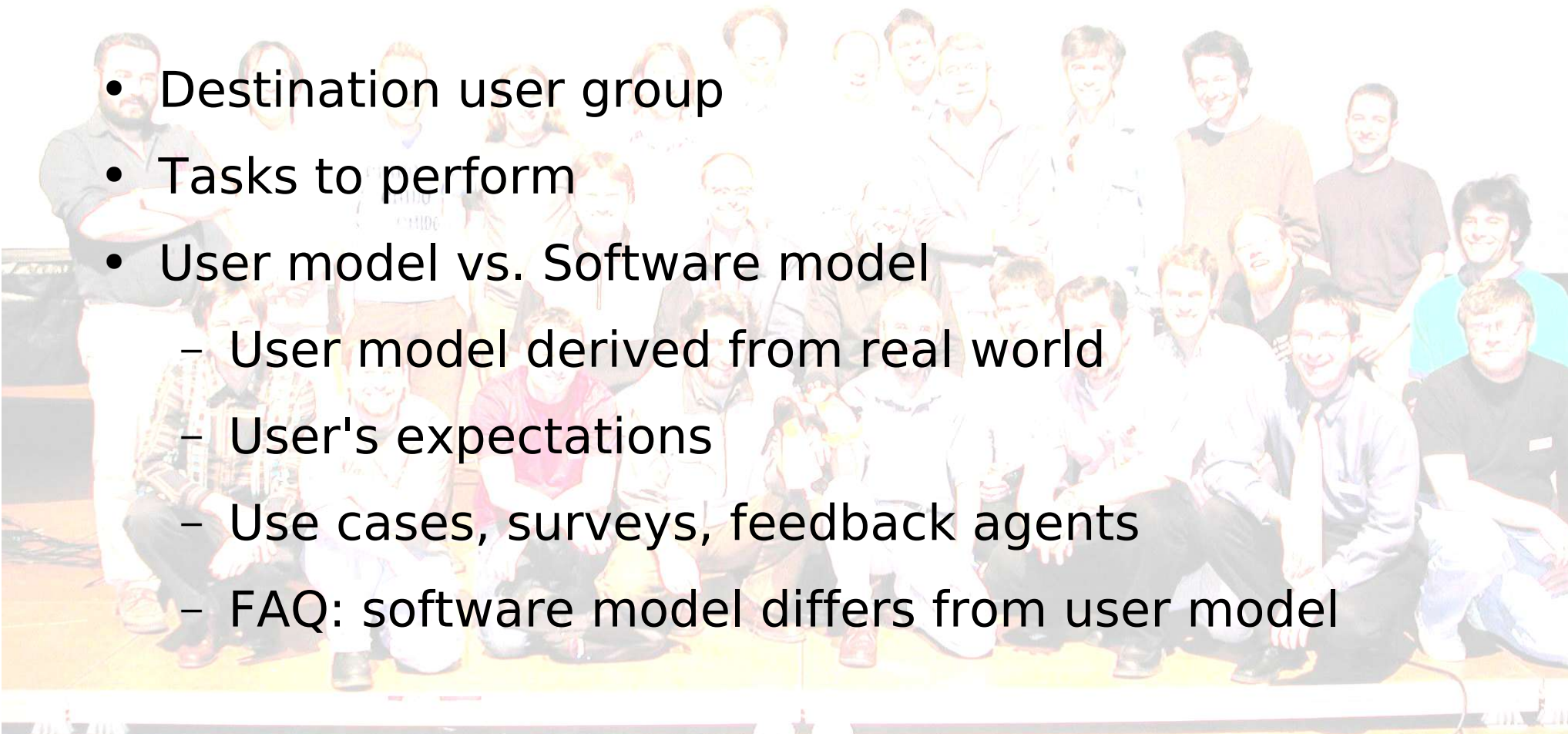
Christoph Eckert
mchristoph.eckert@t-online.de

Introducing usability ideas
based on Linux audio software

Usability Documents

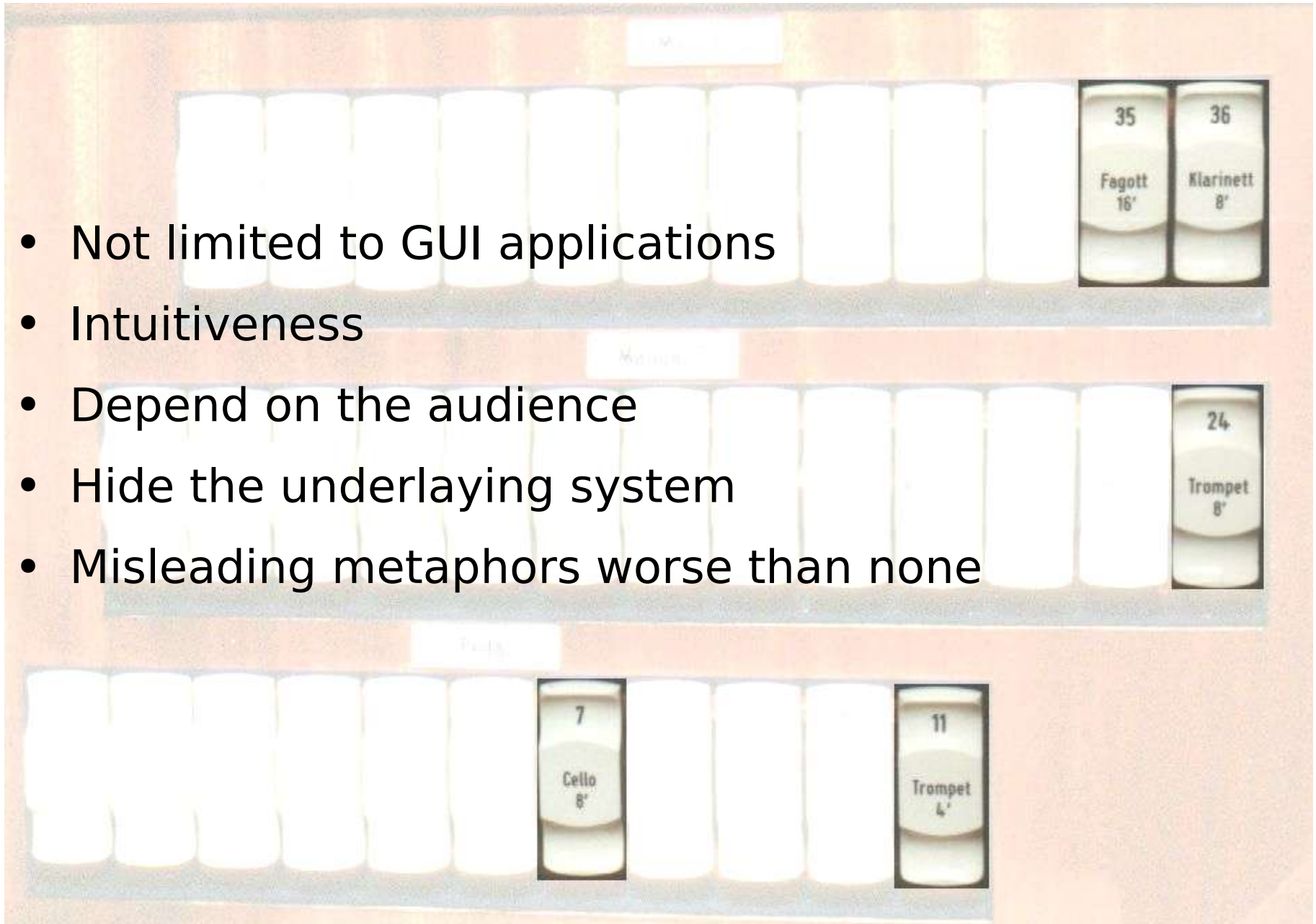
- Mac OS 8 Human Interface Guidelines
- Gnome User Interface Guidelines
- User Interface Design by Joel Spolsky
- GNU Coding Standards

1. Knowing the Audience

- 
- Destination user group
 - Tasks to perform
 - User model vs. Software model
 - User model derived from real world
 - User's expectations
 - Use cases, surveys, feedback agents
 - FAQ: software model differs from user model

2. Metaphors

- Not limited to GUI applications
- Intuitiveness
- Depend on the audience
- Hide the underlying system
- Misleading metaphors worse than none





3. Accessibility

- Citizens with physical or cognitive limitations
 - Blindness (command line options in GUI programs)
 - Movement Limitations (Keyboard shortcuts)
- Internationalisation & Localisation (i18n & l10n)
 - Translation of UI and documentation
 - Cultural and political aspects (parts of humans, flags)

Pianino

Pianino to

Zasada działania jest identyczna jak w fortepianie z tą różnicą, że rama jest pozioma a nie pionowa, co powoduje, że również sam instrument jest wyższy od fortepianu, za to znacząco od niego krótszy.

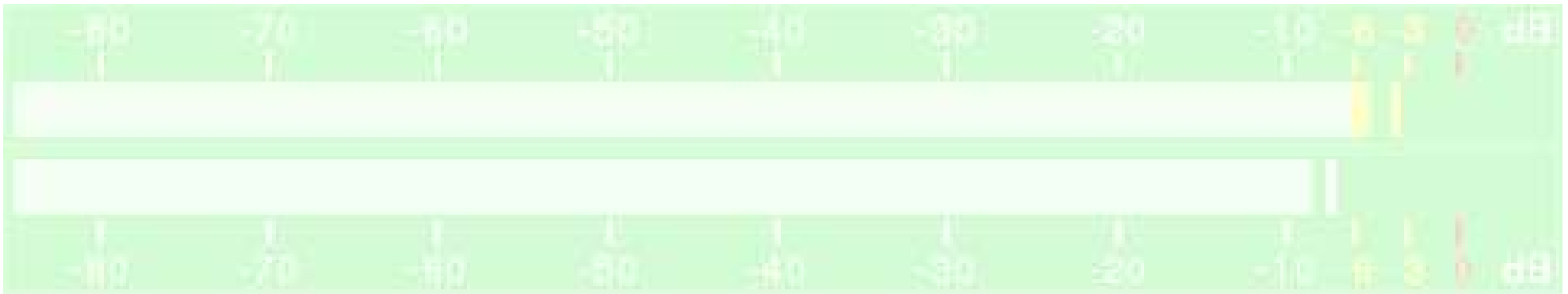
Poprzednikiem pianina było fortepiano. Za pierwszego budowniczego pianina uważa się Bartolomeo Cristoforiego, który pracował nad nim w latach 1680-1710. Konstruktorami pianin byli także Anglicy i Niemcy. Długo jednak pianino nie zdobyło popularności. Podobny do pianina był inny instrument zwany

4. Consistency

- With previous versions & file formats of itself
- With the windowmanager and other applications
 - Support for common command line options
 - GUI programs starting without command line options
 - Desktop help system integration
 - Desktop menu entries & filetype registration

5. Feedback

- Keeping the user informed
- Providing simple but clear information
- Enabling the user to solve a problem



6. Straightforwardness

- Reduced attention amount
- Software remembering things for the user
 - Commands, settings and connections
- Reducing the need to read documentation or dialogs
 - Users dislike reading documentation
 - Project members dislike writing documentation
 - Documentation often is not in sync with releases
- Tidy UI, matching the workflow (analogue synths)
- Disabling interface elements when senseless
- Reasonable preconfiguration for an immediate start
- Human numbering

7. User Control

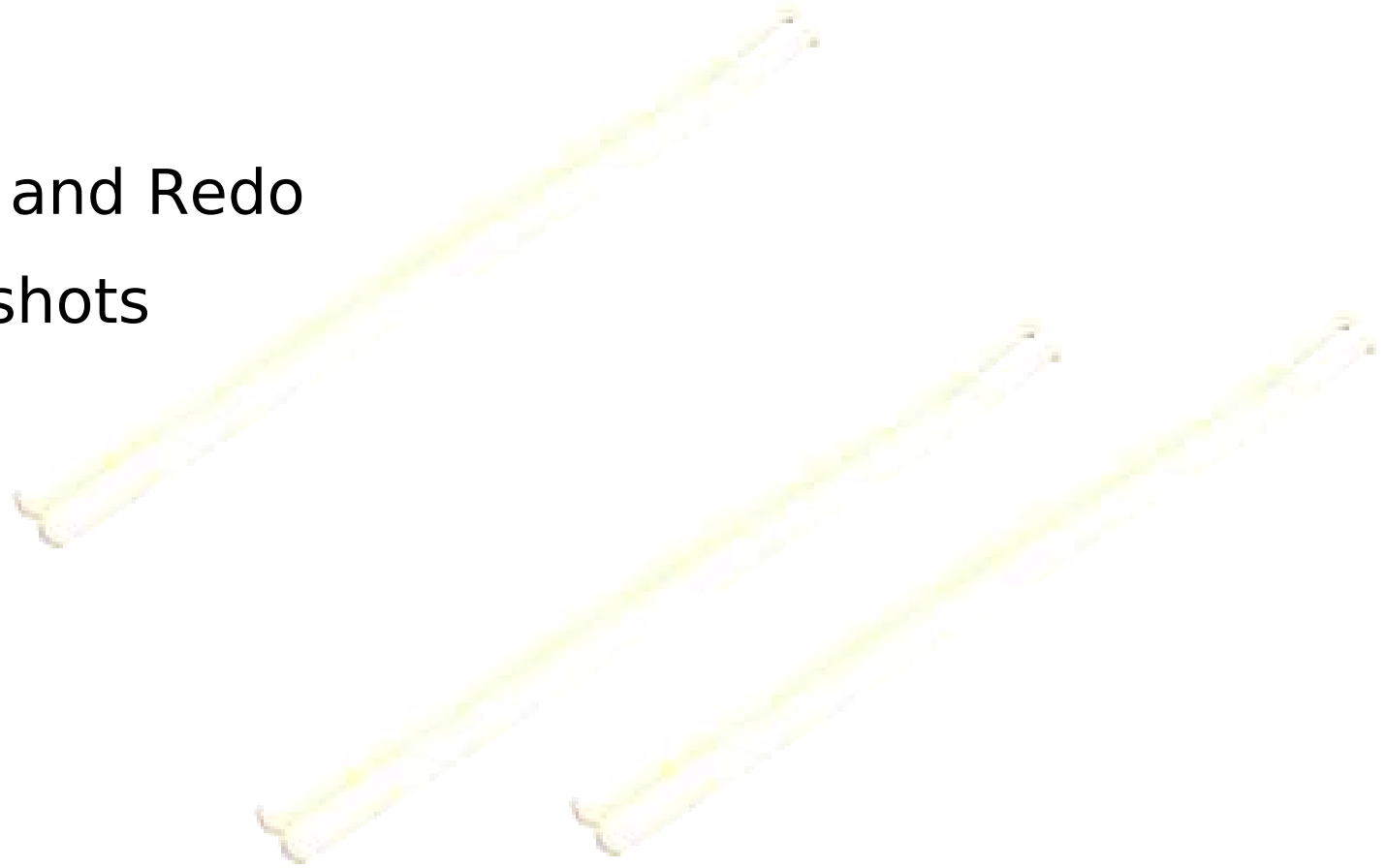
- Humans want to control their environment
- Cancelling long lasting operations
- System Configuration
 - Working Directories (Session or Preference based)
 - Audio Subsystem
 - Renaming and reordering of soundcards and ports
 - Fitting user expectations leads to a feel of control
- Indicators for debugging the setup
- Reducing the amount of decisions to be made

8. Direct Access

- Easy access to commands
- Dynamic instead of static options
- Documentation on demand
 - Tooltips
 - What's this help
 - Need to contain reasonable entries (see consistency)
- Controls for invoking external applications
- Built-in access to MIDI and audio connections
- Configurable card mixer frontends (UTF8)

9. Forgiveness

- Undo and Redo
- Snapshots





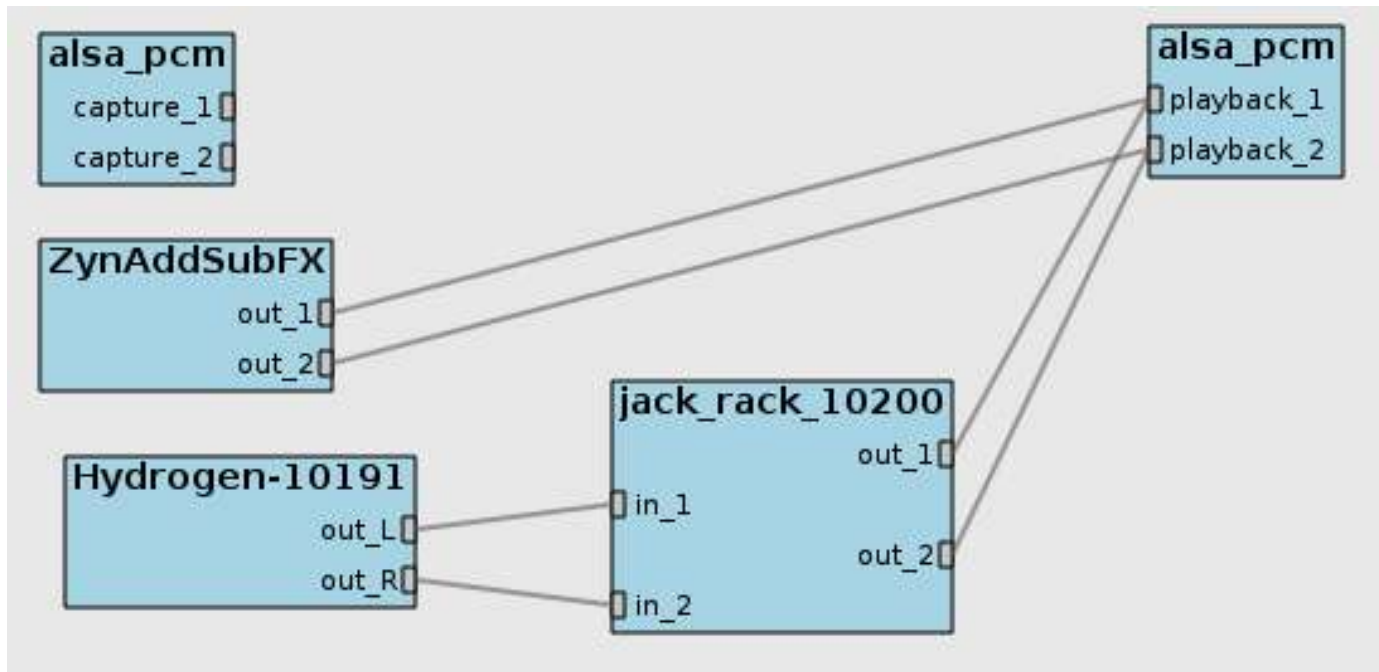
The Developer's Point of View

- Work on usability needs time - resources are limited
- Developers of free software cannot be urged to respect usability ideas
- Bigger projects tend to take care about usability ideas
- Some usability issues can be avoided at the very beginning of a project (dynamic design)
- Later introduction can be difficult (consecutive bugs)
- Technical interest vs. UI design, UI experts missing
- Bug fixing and new features vs. Usability
- Clear decision needed

```
26 jack_default_audio_sample_t jack_get_buffer (input_port, nframes);
27
28 memcpy (out, in, sizeof (jack_default_audio_sample_t) * nframes);
29
30 return 0;
31 }
32
33 /**
34 * This is the shutdown callback for this JACK application.
35 * It is called by JACK if the server ever shuts down or
36 * attempts to disconnect the client.
37 */
38 void
39 jack_shutdown (void *arg)
40 {
41 }
42
43 int
44 main (int argc, char *argv[])
45 {
46     jack_client_t *client;
47     const char **ports;
48
49     if (argc < 2)
50         fprintf (stderr, "usage: jack_simple_client <name>\n");
51     return 1;
52 }
53
54 /**
55 * Tell the JACK server to call 'process()' whenever
56 * there's work to be done.
57 */
58 if ((client = jack_client_new (argv[1])) == 0)
59     fprintf (stderr, "jack_client_new failed: %s\n",
60             strerror (errno));
61 }
62
63 /* tell the JACK server to call 'process()' whenever
64 * there's work to be done.
65 */
66 jack_set_process_callback (client, process, 0);
67
68 /* call the JACK server to call 'jack_activate()'
69 * which starts the client.  Without this call, the
70 * client just decides to stop calling us.
71 */
72
73
74 jack_port_t *input_port, *output_port;
75
76 /* display the current sample rate.
77 */
78
79 printf ("engine sample rate: %u\n",
80        jack_get_sample_rate (client));
81
82 /* create two ports */
83
84 input_port = jack_port_register (client, "input", JACK_DEFAULT_AUDIO_TYPE, JackPortIsInput, 0);
85 output_port = jack_port_register (client, "output", JACK_DEFAULT_AUDIO_TYPE, JackPortIsOutput, 0);
```

Metaphors

- Patchage as intuitive patchbay
- Patchage could also make MIDI connections
- Patchage could act as LADSPA/DSSI host



Accessibility

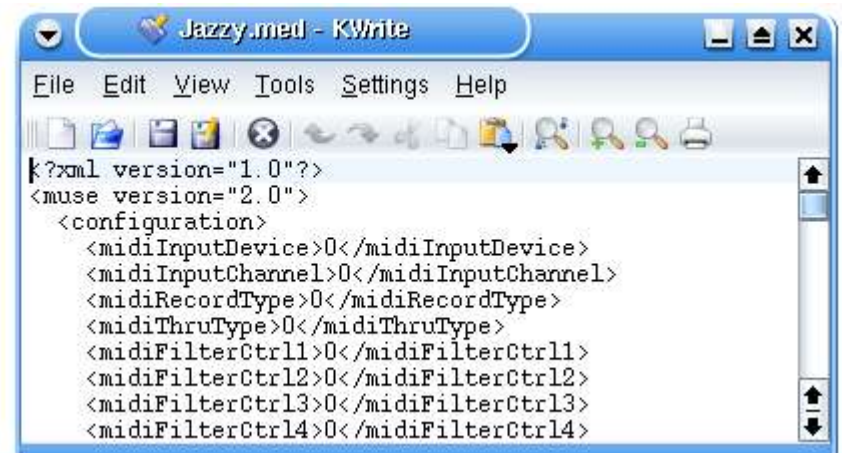
- Applications with preincluded translation and documentation template files will encourage users to contribute GUI and documentation translations



Consistency

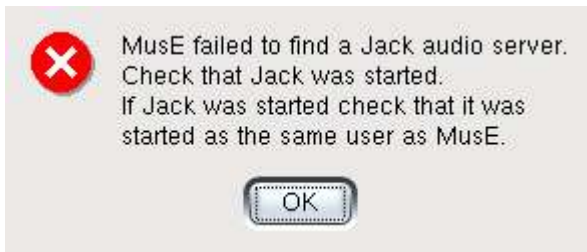
- Documentation on demand (tooltips, 'What's this')
- Menu entries
- Help entries
- Filetype registration
- Understanding common command line options
- Trying to unify command line options

```
bash-2.05b$ aeolus -J  
bash-2.05b$ ams --jack  
bash-2.05b$ mplayer -ao jack
```

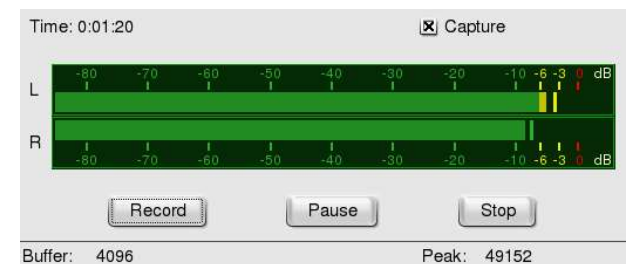


Feedback

- Clearly stating problems
- Not freezing in unexpected circumstances
- Start even if JACK is not available
- Displaying a message if the audio device is blocked
- Displaying work in progress



```
bash-2.05b# xmms Tower_Of_Integration.mp3 &  
[1] 9375  
bash-2.05b# ams --soundcard hw:0,0  
LADSPA_PATH: /usr/lib/ladspa  
loadPath: /usr/share/doc/, savePath: /home/ce/  
█
```



Straightforwardness

- Preconfiguration
 - Loading default documents
 - Easy access to example files
 - Opening last used file at startup
 - Remembering audio system
- Various output plugins
- Autodetection of the available audio subsystem
- Human numbering



User Control

- Indicator controls for debugging the setup

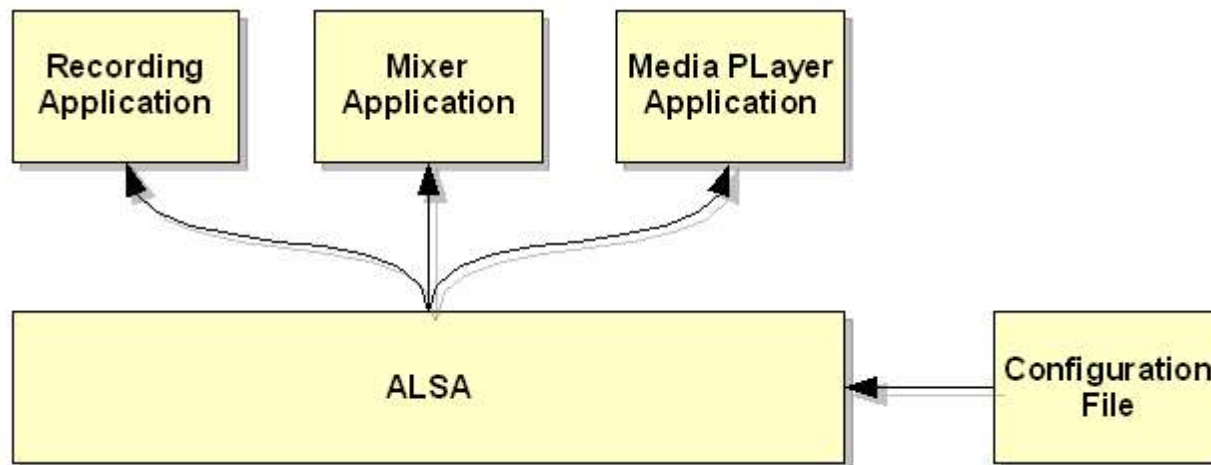


- Avoiding urging the user to make decisions



Direct Access

- Polyphony in ams only via command line switch
- Preference changes requiring restart
- Configurable external helper applications
- Configurable card frontends
- More information for mixers



Summary

- Linux is surely ready for the audio desktop
- Usability can bring free audio software to more users
- More users can lead to more project members
- Usability is not only limited to GUI design
- Usability on heterogenous systems is a special challenge
- Working on usability needs time and resources
- Working on usability needs a clear decision