# Linux and music out of the box

**Hartmut NOACK**
www.linuxuse.de/snd
Max-Steinke-Strasse 23
13086 Berlin,
BRD,
zettberlin@linuxuse.de

## Abstract

The perception of Linux audio software amongst so-called "average musicians" with no "geek"background is not as positive as it could be. Many musicians still think that there is not enough usability and too little Linux software available to produce an album. This proposal introduces design studies for a user interface dedicated to a Linux Audio Workstation (LAW) that I intend to build based on standard PC hardware. The interface intends to integrate all the main virtues of Linux audio software in a logical and self explanatory way, without encumbering their flexibility, to provide the same level comfort and intuitive operation as offered by suites such as Steinberg's Cubase. There will be no install-CD to be downloaded, as offered by Distros like Demudi or Agnula, but only some useful scripts and LFS-style documentation to enable people to rebuild the box itself if they wish to do so.

## Keywords

usability, integration, suite, interface, workstation

## 1   Introduction

Musicians often state that they don't know very much about computers, that they only want to use these boxes and that they have neither the time nor motivation to learn how a computer or software works. Proprietary software vendors try to adapt to such attitudes by designing all-in-one applications, such as Steinberg's Cubase, with simple-looking interfaces and automatic setup assistants that leave power usage and fine tuning to the experienced users and hide possible options and - of course - the source code from the "end user" to keep their products under control and consistent.

This is not the way that Linux audio can go, since it is open source. Free software authors, as well as the distributors, need to develop other ways to achieve usability and a trustworthiness that meets the needs of productive use. This can be done if the whole audio workstation is developed and built as an integrated combination of software **and** hardware components, that fit together and are Linux compatible down to the last screw.

Some may ask:

> is this still possible without charging the users for software ?

we say: It is! - if we find a way to charge the users for integration and support... it is our belief that the time is nigh to make people invest in free audio software development. People like Paul Davis, Taybin Rutkin or Werner Schweer should be rewarded for their work. We believe that a reasonable part of the revenue that could be generated with oss-based audio workstations should go directly to the developers of the most important software for these boxes - especially if these developers do not yet receive any reasonable financial support.

To make sure that the system as a whole works flawlessly enough for studio use, it will be necessary to setup and test each box by hand before shipping. We did so in Suse 9.3 to build the first prototype - on the second box, that we plan to build in June, it will be done with Ubuntu including packages from the Ubuntu studio metadistribution. The project does not intend to add another distribution, toolbox or file format to the already existing mass of experiments in Linux audio - we only want to help others using all the great stuff that is already there. So all scripts will be native, average bash-syntax and XML as used in xfce4. The LAW itself will be built, set up and tuned by hand, the scripts will be usable in every Linux-environment that has xfce4 and the incorporated applications.

## 2   plans and sketches

Most users coming from proprietary tools, such as Steinberg Nuendo or SEKD Samplitude, have a rather ambivalent impression of the system when looking at Linux solutions. On the one

hand they like tools such as Jamin or snd, but on the other hand they are disappointed by the lack of features that they have commonly come to expect in proprietary applications. The concept that a complex system can be built out of many little tools that provide each other with functions is not very common amongst users of proprietary systems[1].

So to add the comfort known from commercial suites to the Linux audio tools, primarily means integrating all the little, cute and powerful free tools to make them work properly together, without limiting their flexibility.

What have we left behind in our Fat32 partitions?

Applications such as Steinberg's NUENDO can be used to record a complex album without learning more then half a dozen keyboard commands (though one can use hundreds of them - if you wish...) i.e.: you only have to learn to understand the metaphors for things that you already know from hardware mixers, recorders and FX boxes, to make NUENDO work for you - you do not need to know about computers, because the GUI makes it look as if it where some space age version of the hardware that you already know.[2].

Under the hood one could find a framework made of engines, plug-in-interfaces and routing mechanisms, quite similar to jack and ladspa, the main difference is indeed the GUI: you don't need to think about it: everything in one place and all needed functions labelled more or less descriptively in more or less logical menus. Right click - double click and sometimes CTRL or Shift - that's it. There are 2 prerequisites for this type of comfort:

1. Every part of the suite works with the given drivers and its interface fits with the rest.

2. The parts do not conflict with each other

Both can be achieved with free components under Linux, but it takes a lot of effort to set it up and there is still no integrated user interface that allows truly intuitive work with the full power of available possibilities and without

unwanted surprises. The ingredients of applications such as NUENDO are available for Linux and PC hardware properly supported by Linux is also available. So the first steps to build a Linux Audio Workstation would be:

1. To design a hardware setup for a reasonable price, that fits perfectly together and is completely Linux proof.

2. To develop a set of scripts and consistent templates and presets that allow the user to switch on the full force of a jack-environment with a single click.

Commercial developers have a major advantage over Linux hackers: the proprietary Systems (i.e. MS Windows XP and Mac OSX) are consistent, applications can be monolithic and survive for years with few changes. Testing and development is easier with less diversity. If Linux software is installed as a binary package and automatically set up, then Linux will not be as stable and consistent as is required. Many developers work on Fedora or Debian and most of them massively alter/tune their systems - so that users who try to to run preconfigured applications on out-of-the-box systems from Novell or Mandriva will be confronted with the unpredictable effects of diversity. Since we cannot relay on consistency - we need to *use* the diversity and freedom of free software to prove that free development can lead to the same and even better functionality as known from commercial vendors. Whereas the basic system can safely be installed automatically, the important applications (jackit, ardour, muse and around 12 more) have to be compiled at the box with sane ./configure-flags and the whole system setup needs to be examined, adapted and thoroughly tested by humans. To limit the effort to a level that can be reached with a small financial budget and in reasonable time, there can be only 1 PC setup and maybe 1 or two laptop machines. The latter will be compromised versions because they must more often serve as office/web-machines than a PC built as a dedicated audio workstation.

To those who want to use our system without purchasing or rebuilding the whole box, we recommend using Ubuntu Linux dapper drake to install the needed components. We have also started a tutorial in the wiki of http://www.audio4linux.de, that describes every step to set up the stuff on Ubuntu dapper

---

[1]there is indeed buzz and of course PD for MS Windows - but those who know these systems well are not in desperate need for clickable wrapper-scripts in Linux....

[2]Steinberg even tries to resemble the look of real equipment by using bitmaps that look rusty and scratched - just like the stomp boxes you are familiar with...

drake, and also provides links to download the needed files.

The main goal of the native approach is stability. No bleeding-edge components, no x86_64-CPU and no beta/cvs versions of software, as far as this can be avoided. The Workstation should start out of the box up to kdm/gdm with no configuration needed (while still retaining all the configuration possibilities that the user may wish for). The user will have the opportunity of choosing between 3-4 preconfigured setups:

- Audio workstation (preset-configuration with factory support, guaranteed functionality, limited configurability)

- Audio workstation experimental (same preset-configuration as above, but with full configurability, limited support and no guarantee - this would also be available for download.)

- PC Workstation (Standard home computer with all the power of a Linux desktop system, Internet, office, graphics/layout etc.)

- Rescue system (with direct access to scripts, that reset configurations and replay backups - this may be offered at the boot screen)

Sometimes the question arises, do we make this with KDE or GNOME? Since both systems are used to provide typical desktop automation mechanisms (office, pim etc) they are not optimal for an audio workstation. An exception to this is the "PC Workstation". Whereas this may work with KDE, the audio systems and the rescue system were set up with Fluxbox, which appeared to us to be the best compromise between leanness and configurability/comfort. Experimenting with other WMs finally led us to the decision to switch to fxce. It is leaner then KDE or GNOME and as scriptable as Fluxbox, but also comes with some very useful features such as multiple menus and bitmap icons for a more descriptive interface. However, all the scripts, templates and presets can be used in any Desktop environment - templates and presets rely on fitting versions of the respective audio applications and the scripts only require bash ... KDE-base must be also installed, since the wrapper scripts utilise kdialog to display messages and konqueror is used to show HTML help pages.

## 2.1 What exists already and how we are different

There are both integrated hardware solutions with Linux and CD and/or metadistros available for installation out there. The hardware systems are designed and priced for semi-professionals and we don't know of any Linux-based audio solution that is also a decent Desktop PC. Our aim is to provide a box that can be built for about 700,- EUR and that can also serve as a Desktop/Internet computer.

The installable Distros such as Demudi, jacklab or CCRMA all have one thing in common: they work with binaries, they do little to integrate the apps and they leave the choice and setup of the hardware to the user. All these people still do great work and it is definitely not our intention to replace any of them, but rather to collaborate with them.

We are not trying to build our own little world, but wish to incorporate things that are already there and glue them together in a useful way. As mentioned before, we primarily deliver simple wrapper scripts, presets, templates and samples. These will work on any distro that has the software installed, which is called by these scripts and can handle these presets etc. On the box that we ship, the same things will run like a charm (no kidding - our concept allows intense testing of the very machine that will be delivered - so unwanted surprises will be seldom...) - if one wants to use the stuff in a similar but different environment, adaptations may be needed.

We follow a paradigm that favours a grassroots style growth of the project. So we have built a standard desktop PC with an Intel PIV 2.8 CPU 1024MB DDR-RAM and a Terratec EWX 24/96 audio card - rather average real-world hardware available for less then 700,- .

After about 8 months of intense testing and experimentation the box is now configured as follows:

- Suse 9.3 Linux system with default kernel 2.6.11.4-20a and alsa 1.0.9 out of the box

- jackit 0.100.0, ardour 0.99,. Muse 0.7.2, Hydrogen 0.9.2 compiled on the machine with Suse's gcc and libs

- recent rezound and snd wave-editors also compiled on the box

- a plethora of synths and smaller tools from Suse-RPM-repositories

We tested the setup by using it to record several multitrack sessions, composing and manipulating music with midi-driven softsynths (neither real keyboards nor external midi hardware so far) and by editing and optimising a noisy-tape-to CD - job for the complete "Ring" by Richard Wagner (plus several similar smaller jobs for string quartet and for rehearsal tapes). The setup works well and stable and provides everything we need.

The issues regarding compatibility/stability are solved so far (though it would not be wise, to actually *guarantee* full stability for all needed programs under all conditions...)

### 2.1.1 ... and what we are working on at the moment

We have begun to build the previously mentioned framework of helping elements, consisting of 4 major components:

- XFCE WM configuration scripts, which allow access to all needed features in a logical and comfortable manner

- several scripts and hacks to load complex scenarios and to provide additional help text

- a set of templates for all major applications that also involve the collaboration between them

- about 300 free licensed presets, sounds and patterns

Starting setups of several collaborating applications *could* be trivial - if all involved elements were aware of each other. Today we still face the problem of certain softsynths that can be attached to jackd via their own command line, and others that need to be called up via tools such as *jack connect*. These are not serious obstacles of course, but there is no reason not to address smaller annoyances as well as great to-dos. The communication between the several processes is particularly critical and often leads to ruin - this should be remarked amongst developers and distributors.

The website http://www.linuxuse.de/snd offers downloads of sample scripts and templates plus some help texts. More help is to be found at http://www.audio4linux.de and we are also working on an extensive user manual that can serve as an introduction to Linux audio. On the final System, in addition to the help provided by the programmers, there will be also be 3 levels of extra help for the users:

- kdialog popups explaining things that happen, ask the user for needed interaction and point to more detailed help.

- HTML help files for *every* application that explain the basics and how the whole system works. It will be possible to start scripts directly from these files, which will be shown in KDE's Konqueror. (Security people may be not that enthusiastic about the help system...)

- an online forum (PHPBB plus a simple wiki) with daily appearance of at least one developer/help author.

## 3 next steps

I would like to present a wish list to the developers and to the user community as well. Developers should improve their documentation and users should start to read it.... . Back in 2003 we had a set of experimental stuff that could be used but was limited by some crucial weaknesses, especially the lack of stability. Today we have the tools complete and solid enough to start to discover how to deploy them as perfectly as possible. To do this, there must be a vivid community of more than just about 100 users[3]

We have combined Ardour, Muse, Rezound, Hydrogen, AMS, ZynaddSubFX and about 20 tools such as Ladspa, qalsatools etc. into a setup that is powerful and usable for us as experienced Linux users, and we have made a feature-freeze from the 1st of January 2006 until the 1st of June 2006, to use the time to make the setup powerful and usable for everyone that wants to deal with music and computers.

We will then offer the complete Linux Audio Workstation, which can be used to give sound-tech people and musicians a chance to find out that Linux audio is ready to run for everybody.

## 4 Conclusions

We believe that free audio software can be an important, powerful way to make Linux visible to the public and thus to make the very concept of collaborative, open and free production of software a success. We not only believed that

---

[3]http://www.frappr.com/ardourusers lists 80 Ardourusers today, Taybin Rutkin pointed the users at the ardour-mailing lists to this page in November 2005...

Linux audio is somewhat usable now and could have a niche - we believe that within 2-3 years it can develop into a superior solution for creative sound people in the same manner as Linux has become a success in motion picture animation/CGI - production.

This can be done if it it becomes easier to use, more logically integrated, more stable and more consistent **without** hiding anything of its great opportunities from the user.

At LAC I would like to present our approach to making Linux audio usable for everyone to developers and users as well.

## 5 Acknowledgements