

The AGNULA/DeMuDi distribution: GNU/Linux and Free Software for the pro audio and sound research domain

Bernardini, Nicola, Cirotteau, Damien, Ekanayaka, Free and Glorioso, Andrea
The AGNULA Project
& Media Innovation Unit - Firenze Tecnologia

ABSTRACT

AGNULA (acronym for “A GNU/Linux Audio distribution”, pronounced with a strong g) is the name of a project which has been funded until April 2004 by the European Commission (number of contract: IST-2001-34879; key action IV.3.3, Free Software: towards the critical mass). After the end of the funded period, AGNULA is continuing as an international, mixed volunteer/funded project, aiming to spread Free Software in the professional audio/video arena. The AGNULA team is working on a tool to reach this goal: AGNULA/DeMuDi, a GNU/Linux distribution based on Debian, entirely composed of Free Software, dedicated to professional audio research and work. This paper¹ describes the current status of AGNULA/DeMuDi and how the AGNULA team envisions future work in this area.

1. THE AGNULA PROJECT - A BIT OF HISTORY

In 1998 the situation of sound/music Free Software applications had already reached what could be considered well beyond initial pioneeristic stage. A website, maintained by musician and GNU/Linux² enthusiast Dave Phillips, was already collecting all possible sound and music software running on GNU/Linux architectures. At that time, the biggest problem was that all these applications were dispersed over the Internet: there was no common operational framework and each and every application was a case-study by itself.

A natural development followed shortly after, when musician/composer/programmer Marco Trevisani proposed a to a small group of friends (Nicola Bernardini, Maurizio De Cecco, Davide Rocchesso and Roberto Bresin) to create LAOS (the acronym of *Linux Audio Open Sourcing*), a binary distribution of all essential sound/music tools available at the time including website

¹This paper is Copyright © 2004 Bernardini, Cirotteau, Ekanayaka, Glorioso and Copyright © 2004 Firenze Tecnologia. It's licensed under a Creative Commons BY-SA 2.0 License (see <http://creativecommons.org/licenses/by-sa/2.0/legalcode>).

²Throughout the document, the term GNU/Linux will be used when referring to a whole operating system using Linux as its base kernel, and Linux when referring to the kernel alone.

diffusion and support. LAOS came up too early, and it did not go very far.

But in 2000, when Marco Trevisani proposed (this time to Nicola Bernardini, Günter Geiger, Dave Phillips and Maurizio De Cecco) to build DeMuDi (*Debian Multimedia Distribution*) an unofficial Debian-based binary distribution of sound/music Free Software, times were riper.

Nicola Bernardini organized a workshop in Firenze, Italy at the beginning of June 2001, inviting an ever-growing group of supporters and contributors (including: Marco Trevisani, Günter Geiger, Dave Phillips, Paul Davis, François Déchelle, Georg Greve, Stanko Juzbasic, Giampiero Salvi, Maurizio Umberto Puxeddu and Gabriel Maldonado). That was the occasion to start the first concrete DeMuDi distribution, the venerable *0.0 alpha* which was then quickly assembled by Günter Geiger with help from Marco Trevisani. A bootable CD-version was then burned just in time for the ICMC 2001 held in La Habana, Cuba, where Günter Geiger and Nicola Bernardini held a tutorial workshop showing features, uses and advantages of DeMuDi[1].

On November 26, 2001 the European Commission awarded the AGNULA Consortium — composed by the Centro Tempo Reale, IRCAM, the IUA-MTG at the Universitat Pompeu Fabra, the Free Software Foundation Europe, KTH and Red Hat France — with consistent funding for an accompanying measure lasting 24 months (IST-2001-34879). This accompanying measure, which was terminated on March 31st 2004, gave considerable thrust to the AGNULA/DeMuDi project providing scientific applications previously unreleased in binary form and the possibility to pay professional personnel to work on the distribution.

After the funded period, Media Innovation Unit, a component of Firenze Tecnologia (itself a technological agency of the Chamber of Commerce of Firenze) has decided to partly fund further AGNULA/DeMuDi developments.

AGNULA has constituted a major step in the direction of creating a full-blown Free Software infrastructure devoted to audio, sound and music, but there's much more to it: it is the first example of a European-funded project to clearly specify the complete adherence of its results to the Free Software paradigm in the project contract, thus

becoming an important precedent for similar projects in the future.

2. FREE SOFTWARE AND ITS APPLICATIONS IN THE “PRO” AUDIO DOMAIN

When describing the AGNULA project, and the AGNULA/DeMuDi distribution specifically, a natural question arises - why is it necessary or desirable to have a completely Free Software based distribution (whether based on the Linux kernel or not is not the point here) for audio professionals and research in the sound domain?

Free Software³ is the set of all computer programs whose usage and distribution licenses (think about the “EULA” or “End User Licensing Agreements”, that so many users have come to know throughout the years) guarantee a precise set of freedoms:

- The freedom to run the program, for any purpose (freedom 0);
- The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this;
- The freedom to redistribute copies so you can help your neighbor (freedom 2);
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition for this;

The most famous of such licenses is probably the GNU General Public License,⁴ which is the founding stone of the Free Software Foundation effort to build a completely free operating system, GNU (GNU’s Not Unix).

This is not the right place to describe the concepts and the history of the Free Software movement as it would deserve.

Suffice it to say that the possibility to use, study, modify and share computer programs with other people is of paramount importance to the everyday life of creators (i.e. composers), professional users (i.e. sound engineers, performers) and researchers. This distinction is of course artificial, since all of us can be creators, professional users and researchers in specific moments of our life. But this taxonomy can work as a simple tool to better understand the pros of using Free Software in everyday life and work:

³We tend to prefer this term, rather than “Libre Software”, even if the former term is inherently ambiguous because of the english term “free” — which can mean “free as in free beer” or “free as in free speech”. Free Software is, of course, free as in free speech (and secondarily, but not necessarily, as in free beer). Usage of the term “Libre Software” arose in the european context trying to overcome this ambiguity with a term, *libre*, which is correct in the french and spanish languages and is understandable in italian and other european languages. However, it’s not universally accepted as an equivalent of “Free Software” and its usage can induce confusion in readers and listeners — we therefore prefer to stick to the traditional, albeit somewhat confusing, terminology.

⁴<http://www.fsf.org/licenses/gpl.html>

- **Creators** can use tools which don’t dictate them what they *should* do, instead being easily modifiable into something that does what they *want* them to do. The non-physical nature of software makes for a very convenient material to build with; even though the creator might not have the needed technical skills and knowledge to modify the program to best suit his/her needs, s/he can always ask someone else to do it; on a related note, this kind of requests make for a potentially (and in some key areas, factually) very thriving marketplace for consultants and small businesses;
- **Professional users** have at their disposal a series of tools which were often thought and designed by other professional users; they can interact more easily with the software writers, asking features they might need or reporting bugs so that they are corrected faster (some would say “at all”). They can base their professional life not on the whim of a single company whose strategies are not necessarily compatible with the professional user’s own plans, but on a shared ecosystem of software which won’t easily disappear — if the original software authors stop maintaining the software, someone else can always replace them;
- **Researchers** can truly bend the tool they have at their disposal to its maximum extent, something which is often very hard to do with proprietary software (even with well designed proprietary software, as it is basically impossible to understand all users’ requirements in advance). They can count on computer programs which have been deeply scrutinized by a peer-review process which finds its counterpart only in the scientific community tradition⁵ as opposed to the habit of proprietary software to completely obscure the “source code” of a program, and all the bugs with it. Last, not least, for all those researchers who use software programs not as simple tools but as bricks in software development (as often happens today in computer-assisted composition and more generally in sound research) the possibility to draw from an immense database of freely available, usable and distributable computer programs can prove an incredible advantage, especially when considering the cost of proprietary computer programs and the financial situations of most research institutions nowadays.⁶

In the end, one might ask whether creativity is truly possible without control on the tools being used — a control which Free Software guarantees and proprietary soft-

⁵This is not a coincidence, as the GNU project was basically born in the Artificial Intelligence Laboratories at the M.I.T.

⁶It should be noted, however, that whilst monetary costs are of course a strong variable of all the equation, the central importance of Free Software in research is **not** related to money itself. Having free (i.e. gratis) software which is not free (i.e. not *libre*) can be an ephemeral panacea, but on the long run it simply means tying oneself and one’s own research strategy to somebody else’s decisions.

ware sometimes grants, but more often than not manipulates for purely economical reasons.

This is not an easy question to answer at all — there are many subtle issues involved, which span in the field of economics, psychology, engineering, sociology, etc, etc. The AGNULA project actually believes that creativity is very difficult without such control,⁷ but it's unquestionable that the subject would deserve a fairer treatise, through cross-subject studies able to span the whole range of fields outlined above.

3. THE AGNULA/DeMuDi FRAMEWORK

The framework of AGNULA/DeMuDi is the “classical” environment one can expect from a GNU/Linux system to run audio applications. The first component is the Linux kernel⁸ patched to turn it into an efficient platform for real time applications such as audio applications. Then the ALSA drivers⁹ allow the usage of a wide range of soundcards from consumer grade to professional quality. On top of the drivers runs the Jack¹⁰ server which allows lowlatency, synchronicity and inter-application communication. Last, not least, the LADSPA plugins format¹¹ is the standard for audio plugins on the GNU/Linux platform.

3.1. The Linux kernel

3.1.1. Is the Linux kernel suitable for audio applications?

The heart of the AGNULA/DeMuDi distribution is the Linux kernel. However, since Linux was originally written for general purpose operating systems (mainly for servers and desktop applications) as a non preemptive kernel, it was not really useful for real-time applications. Truly showing the power of Free Software, several improvements of the kernel scheduler turned it into a good platform for a Digital Audio Workstation (DAW).

To face this limitation two strategies have been adopted: the preemption patch and the lowlatency patch.

3.1.2. Preemption patch

Originally created by MontaVista¹² and now maintained by Robert M. Love,¹³ this patch redesigns the kernel scheduler and redefines the spinlocks from their SMP specific implementation to preemption locks. This patch allows the Linux scheduler to be preemptive – when an interruption of higher priority occurs the kernel preempts the current task and runs the higher priority task – except

⁷This belief has become a sort of mantra, as is stated on our t-shirts: “There is no free expression without control on the tools you use”.

⁸<http://www.kernel.org>

⁹<http://www.alsa-project.org>

¹⁰<http://jackit.sf.net/>

¹¹<http://www.ladspa.org/>

¹²<http://www.mvista.com/>

¹³<http://www.tech9.net/rml/linux>

for specific critical sections (such as spinlocks or when the scheduler is running). This strategy has proven its efficiency and reliability and has been included in the new stable releases of the kernel (2.6.x).

3.1.3. Lowlatency patch

Introduced by Ingo Molnar and improved by Andrew Morton, the lowlatency¹⁴ patch introduces some specific conditional rescheduling points in some blocks of the kernel. Even if the concept of this patch is quite simple, it imposes a very high maintenance burden because the conditional rescheduling points are spread all over the kernel code without any centralization.

3.1.4. Which patch is the best?

We test the kernel 2.4.24 with the methodology of [2].¹⁵ We used *realfeel*¹⁶ while running the Cerberus Test Control System¹⁷ to stress the kernel. 5.000.000 interrupts were generated with a frequency of 2048 interrupt per second and the scheduling latency is measured for each interrupt on a Intel Centrino 1.4 MHz with 512 Mb of RAM

The result for the non-patched kernel (see Figure 1) with a maximum latency of 48,1 ms makes this kernel not suitable for real-time application. The patches greatly improve the situation. The lowlatency patch provides better results – better maximum latency and highest percentage of lowlatency interrupts. The optimal choice seems to be the combination of both. The combination of the patches has also proven to be more reliable after a long uptime (see [2])

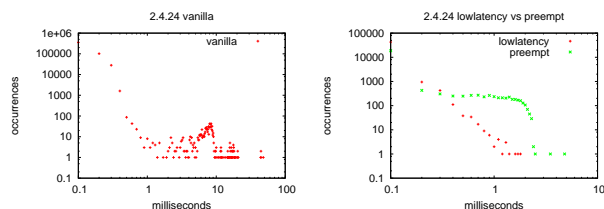


Figure 1. *Vanilla vs Lowlatency and preempt 2.4.24 scheduler latency*

Even if AGNULA/DeMuDi still provides a 2.4.x kernel some preliminary tests show that the new stable kernel (2.6.x) provides better scheduler and will therefore be very suitable for an audio platform. The preempt patch is now directly shipped with the vanilla kernel. The maximum latency measured for the 2.6.5 kernel is 0.7ms, and

¹⁴<http://www.zip.com.au/~akpm/linux/schedlat.html>

¹⁵We invite the reader to consult this paper for a more detailed explanation of how the kernel scheduler works and of the two patches.

¹⁶<http://www.zip.com.au/~akpm/linux/schedlat.html>
#amlat

¹⁷<http://sourceforge.net/projects/va-ctcs/>

	2.4.24 vanilla	2.4.24 lowlatency	2.4.24 preempt	2.4.24 both	2.6.5 preempt
$\max(L)(ms.)$	48.1	1.8	4.8	1.8	0.7
$L < 0.1ms(\%)$	90.2182	99.1168	99.5404	99.4831	99.9685
$L < 0.2ms(\%)$	97.3432	99.9679	99.9115	99.9643	99.9878
$L < 0.5ms(\%)$	99.9768	99.9976	99.9311	99.9973	99.9982
$L < 1ms(\%)$	99.9801	99.9997	99.9567	99.9998	100
$L < 10ms(\%)$	99.9983	100	100	100	100
$L < 50ms(\%)$	100	100	100	100	100

Table 1. Repartition of the latency measurements for the different kernels

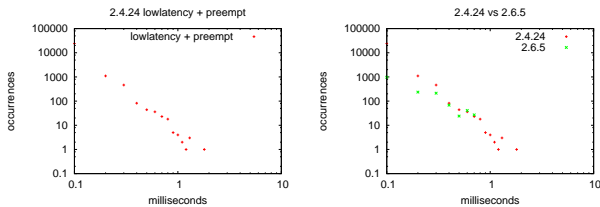


Figure 2. Lowlatency + Preempt 2.4.24 and preempt 2.6.5 scheduler latency

the percentage of interrupts being served within 0.1 ms is significantly higher than for any version of the 2.4.24 kernel.

3.1.5. Capability patch

The third patch applied to the kernel does not improve the performance of the system but allows a non-root users to use the real time *capability* of Linux. It is particularly useful to run the Jack (see 3.3) audio server as a normal user.

3.2. ALSA

ALSA (Advanced Linux Sound Architecture) is a modular software framework which supports a wide range of soundcards¹⁸ from consumer grade to professional quality. The ALSA drivers also provide an OSS/Free¹⁹ emulation to allow compatibility with legacy applications. ALSA is now the standard audio subsystem of the 2.6.x Linux kernels (replacing OSS, which was the standard throughout the 2.4.x series). ALSA also provides an API and a user space library (libasound).

3.3. The Jack Audio Connection Kit

The Jack Audio Connection Kit (*Jack*) can be considered as the real user-space skeleton of AGNULA/DeMuDi. This audio server runs on top of the audio driver (ALSA

¹⁸See the – not-so-up-to-date – soundcards matrix on the ALSA web pages to have an idea of the number of soundcards supported.

¹⁹<http://www.opensound.com/>

, OSS or Portaudio²⁰) and allows the different audio applications to communicate with each other. While other audio servers exist (aRts and esd among others), Jack is the only one which has been designed from the grounds up for professional audio usage: it guarantees low latency operations and synchronicity between different client applications. Therefore it has become a *de facto* standard for professional audio on GNU/Linux systems and the majority of the applications included in the AGNULA/DeMuDi distribution are Jack-compliant (“jackified”, to use the relevant jargon). Another reason for Jack’s success is the simple, high-level but powerful API that it provides, which has greatly facilitated the *jackification* of audio applications.

Last, not least, *Jack* also provides a master transport which allows for simultaneous control of different applications (start/pause/stop).

3.4. The LADPSA plugins

LADSPA, which stands for Linux Audio Developers Simple Plugins Architecture, is the VST equivalent on GNU/Linux systems. It provides a standard way to write audio plugins. The majority of the applications included in AGNULA/DeMuDi supports this format; since a number of high qualities plugins are available and non-compliant applications are “changing their mind”, it’s apparent how LADSPA is the “de facto” standard as far as audio plugins are concerned.

4. APPLICATIONS

AGNULA/DeMuDi doesn’t provide all the music/audio programs available for the GNU/Linux platform; the goal is to provide a thought-out selection of the “best” ones, allowing every kind of user to choose from consumer-grade to professional-grade applications. Even after the reduction process the original list underwent, the number of applications included in AGNULA/DeMuDi (100+) obliges us to present a restricted but representative overview. A complete list of the available applications is included either in the distribution itself or online²¹.

²⁰<http://www.portaudio.com/>

²¹<http://www.agnula.org/Members/damien/List/view>

Sound Editors The choice of the sound editors included in AGNULA/DeMuDi illustrate the versatility of the distribution: it goes from the complex but extremely powerful *Snd*²² to the user friendly and straightforward *audacity*²³ for the time domain. Frequency domain edition is possible with *ceres*²⁴.

Multitracker Considered as one of the major audio applications for GNU/Linux, *Ardour*²⁵ is not only an excellent multitrack recorder but it also “caused” the development of *Jack*, as the author of these two programs, Paul Davis, originally developed *Jack* to fulfil a need he had for *Ardour*. *Ecasound*²⁶ is a robust non-GUI alternative for multitrack recording.

Interactive Graphical Building Environments Free Software is very strong in this field with two well developed applications which have been enjoying a tremendous success for years: *jMax*²⁷ and *Pure Data*²⁸ (better known as *Pd*).

Sequencers Two sequencers amongst others are worth mentioning: *Rosegarden*²⁹ and *Muse*. While originally they were pure midi-sequencers, now they both have some audio capabilities which turn them into complete musical production tools.

Sound Processing Languages A wide choice of compositional languages like *Csound*, *SuperCollider*, *Common Lisp Music* are available. It may be noticed that the first two were re-licensed under respectively the GNU LGPL (GNU Lesser General Public License) and the GNU GPL during the funded lifetime of the AGNULA project.

Software synthesizers A good range of software synthesizer is provided, including tools for modular synthesis (*AlsaModularSynth*, *SpiralSynthModular*); for additive and subtractive synthesis (*ZynAddSubFX*); and dedicated synthesis/compositional languages, such as *Csound*³⁰ and *SuperCollider*³¹.

Last, not least, *fluidsynth*³² and *TiMidity++* allow sample-based synthesis. In the attempt to distribute only Free Software, a Free GUS patch, *freepat*³³ is also provided with *TiMidity++*. The patch is not complete (it still misses some instruments to cover

the General Midi map) and this raised our perception that free content (like free samples or free loops) are a crucial need in order to provide a totally Free audio platform.

Notation The last category is particularly well represented with the professional-grade automated engraving system *Lilypond*³⁴. While *Lilypond* provides a descriptive language of the musical scores, it is also a back-end for different applications such as *Rosegarden* or the dedicated notation interface *NoteEdit*.

5. PROSECUTION AFTER THE ENDING OF THE FUNDED PHASE

5.1. Why should AGNULA/DeMuDi go on?

AGNULA/DeMuDi gave rise to a fairly large interest. Users' feedback has constantly increased over the past months as well as the requests for further enhancements. Being AGNULA/DeMuDi a Free Software project, these conditions naturally favor its continuation after the end of the EC-funded period.

Even though the AGNULA team has always worked in a transparent way, documenting its progresses and welcoming external contributions, AGNULA's technical manager³⁵ and the AGNULA team felt that some actions needed to be taken to allow a successful migration to a volunteer-based effort.

For this reason, during the final phase of the project, special attention has been paid to enhancing the accessibility of its services and its infrastructure (e.g. the CVS and the building/archiving system) and the relationship with the Debian and Linux Audio³⁶ communities.

5.2. What happens to the packages?

The AGNULA team has, throughout the years, adapted and packaged a number of applications, a part of which were included in the official Debian Project; all the applications are now in the process of being uploaded on the Debian official archive and will thus become a part of the Debian Project. We hope these additional packages will help Debian grow its popularity among audio professionals, musicians and composers.

The domain of GNU/Linux audio applications is still developing very quickly, actually quicker than two years ago, when the project began; the AGNULA/DeMuDi packages will need to be constantly upgraded and maintained to prevent fast obsolescence.

³⁴<http://www.lilypond.org>

³⁵Starting from march 2003, Andrea Glorioso.

³⁶Although we did state that we would use the term GNU/Linux when referring to whole operating systems based on the Linux kernel, as is the case for current Linux Audio users' systems, the term “Linux Audio” has been traditionally used throughout the relevant community, and as such we decided not to change it.

²²<http://ccrma.stanford.edu/software/snd/>

²³<http://audacity.sourceforge.net/>

²⁴<http://www.music.columbia.edu/~stanko/About.Ceres3.html>

²⁵<http://www.ardour.org/>

²⁶<http://www.wakkanet.fi/~kaiv/ecasound/welcome.html>

²⁷<http://www.ircam.fr/jmax/>

²⁸<http://www-crca.ucsd.edu/~msp/software.html>

²⁹<http://www.rosegardenmusic.com/>

³⁰<http://www.csounds.com/>

³¹<http://supercollider.sourceforge.net/>

³²<http://www.fluidsynth.org/>

³³<http://freepats.opensrc.org/freepats/>

Unfortunately, AGNULA/DeMuDi cannot count on the same manpower it had during the funded lifetime. Therefore the AGNULA/DeMuDi team has provided and documented a development infrastructure to make life simpler to prospective maintainers. Key elements of such infrastructure are:

- Using CVS to keep track of packages' life;
- Granting direct access to the source code on which the maintainer is working, allowing anyone to contribute by enhancing and fixing a package;
- Automating some of the most frequent operations (upgrading, building, releasing) done on packages;

Beside maintaining the existing package base the above mentioned facilities were designed to promote further development and hopefully to become a reference point for Debian developers dealing with audio packages.

Specifically the GForge³⁷ collaborative environment, which was used as the basis for the AGNULA development portal³⁸ has shown obvious advantages, such as the ability to coordinate development amongst a distributed group of people; the possibility to share knowledge and experience; the lower entry level needed to actively join the project; the closer connection between developers and users, as the AGNULA/DeMuDi experience shows.

5.3. Integration in the Debian Project

When the AGNULA project started there were various reasons which determined the choice of forking the official Debian project and providing a separate distribution.

Some were technical issues, such as the need to directly control some critical part of the system (e.g. the kernel, the ALSA library, the JACK audio connection kit), and to be able to make independent decisions. Some others were Debian specific problems, such as the lack of coordination between maintainers of audio packages.

Since that time the situation has significantly evolved, and most of those reasons do not hold true anymore.

Our experience with AGNULA/DeMuDi taught us that forking Debian to provide a parallel distribution is not always the best move, and in the long run it may turn to be a rather risky choice, as it can prevent the growth of a project — the cost of maintaining a separate distribution being very high.

On the contrary, some recently born official Debian sub-projects, which are efforts to improve or specialize a specific part of Debian from the inside, are showing that being *integrated in* Debian is a good chance to *grow on the back of* Debian.

As the application domain which we refer to is developing very quickly AGNULA/DeMuDi often had to deal with brand new packages and with frequent upstream

source upgrades of existing Debian packages, which were (and are) critical to provide a distribution constantly aligned with the state of the art. Debian's release cycle is very long, and official releases (currently "woody") suffer from fast obsolescence. AGNULA/DeMuDi had to be based on the "unstable" (i.e. not released) branch of Debian, because that branch offered all needed functionalities that the official release of Debian did not provide.

On the other hand DeMuDi needs to closely track only a subset of Debian packages, and basing it directly on the unstable branch is somewhat overkill and moreover occasionally dangerous for users. A different approach could be more beneficial to AGNULA/DeMuDi as a whole.

Recently sub-projects seem to have reached the critical mass to get some more facilities from Debian itself, which is going to announce some enhancement to the archiving system which will allow an easier tracking of specific subsets of packages. This will solve some difficulties which are common to all the **Custom Debian Distributions**, as they have been named, and would be the first step toward a consistent framework for all sub-projects.

In particular the AGNULA/DeMuDi team has been actively involved in such discussions and AGNULA/DeMuDi is likely to soon benefit from such new approaches.

5.4. Debian Multimedia

Debian Multimedia is an official Debian sub-project started by Marco Trevisani (former technical manager of AGNULA/DeMuDi) whose goals are virtually identical to AGNULA/DeMuDi. The activity of the group is not as intense as AGNULA/DeMuDi, but it is constant in time, and has achieved some high quality results (e.g. good packaging for the JACK Audio Connection Kit).

The Debian Multimedia sub-project not only represents the ideal door for AGNULA/DeMuDi to enter Debian, but can also be considered a reference point for other Debian based distributions dealing with audio and multimedia (e.g. Medialinux), and it would allow to gather the various efforts under the same hat.

While AGNULA/DeMuDi had a fairly large success among the users, creating an active community around the project, it is remarkable that, beside a few cases, the same thing did not happen with respect to the developers, who generally preferred to stick to Debian.

After the funded phase of AGNULA, the team is going to progressively move all the development activity from AGNULA/DeMuDi to Debian Multimedia. In particular all the packaging effort will be carried on inside the Debian Multimedia group, while AGNULA/DeMuDi will limit its scope to the distribution mechanism (most noticeably the shipping of ISO installable images).

³⁷<http://www.gforge.org/>

³⁸<http://devel.agnula.org/>

6. CONCLUSIONS

The AGNULA project, originally funded by the European Commission, is now continuing to pursue its goal of making Free Software the best choice for audio/video professionals on a volunteer/paid basis. The history of the AGNULA project, AGNULA/DeMuDi current status and its foreseeable future have been shown, as well as the general philosophy and technical beliefs that are behind the AGNULA team choices.

The AGNULA team does believe that a positive feedback loop has been spawned between Debian and the fast evolving domain of GNU/Linux audio applications. As a matter of fact a previously weak ring in the chain between audio professionals, musicians and composers on one side and Free Software developers on the other has been significantly strengthened.

This result can be considered the basis of a future adoption of Free Software tools by people who formerly had no alternative to proprietary software, along with all the implications of such a process in the educational, social, artistic and scientific fields.

7. ACKNOWLEDGEMENTS

As the reader may expect, projects such as AGNULA/DeMuDi are the result of the common effort of a very large pool of motivated people. And indeed, giving credit to any deserving individual that contributed to these projects would probably fill completely the space allotted for this paper. Therefore, we decided to make an arbitrarily small selection of those without whose help AGNULA/DeMuDi would not probably exist. First of all, we would like to thank Richard Stallman, without whose effort Free Software would not exist at all; Linus Torvalds, who contributed the operating system we all got to depend on in the last dozen of years; and Ian Murdock who started Debian, the wonderful packaging infrastructure AGNULA/DeMuDi is based on. Then, Marco Trevisani, who has been pushing the envelope of a Free audio/music system for years, Dave Phillips, Günter Geiger, Fernando Lopez-Lezcano, François Déchelle and Davide Rocchesso: all these people have been working (and still work) on these concepts and ideas since the early days. Georg Greve, the president of Free Software Foundation Europe, has guided us through the difficulties of carrying out a rigorous Free Software project: may his patience be awarded by the warmest thanks. Other people that deserve our gratitude are: Philippe Aigrain and Jean-François Junger, the European Commission officials that have been promoting the idea that AGNULA was a viable project against all odds inside the Commission itself; Dirk Van Rooy, later AGNULA Project Officer, Marc Lemman and Xavier Perrot, patient AGNULA Project Reviewers; Luca Mantellassi and Giovanni Nebiolo, respectively President of Firenze's Chamber of Commerce and CEO of Firenze Tecnologia, for their support: they have un-

derstood the innovative potential of Free Software much better than many so-called open-source evangelists. Finally, last but not least, Anna Meo and Nicola Giosmin, close collaborators for many years in all these (and other) endeavors, deserve all our endless esteem and gratitude: without them, a lot of these achievements would have made our life much harder than it has been.

8. REFERENCES

- [1] François Déchelle, Günter Geiger, and Dave Phillips, "Demudi: The Debian Multimedia Distribution," in *Proceedings of the 2001 International Computer Music Conference*, San Francisco USA, 2001, ICMA.
- [2] Clark Williams, "Linux scheduler latency," Tech. Rep., Red Hat Inc., 2002.