

<http://vektordb.lafkon.net> redesign a free graphics repository

Christoph HAAG
LAFKON
Kitzenmarkt 8
86150 Augsburg,
Germany,
christoph@lafkon.net

Abstract

At <http://vektordb.lafkon.net> you can browse and download a collection of vector-based¹ images. The database first was intended as an online storage facility for *personal use*, but also allowed *use by third-party persons*. During the years the archive got recognized by a bigger audience and a redesign became necessary. This redesign process led to a broader approach towards a *Free/Libre/Open Source Software* inspired graphics repository.

Keywords

graphic design, open content, database

1 Introduction

In 2004 we² put a repository online, containing graphical symbols and illustrations in vector format.¹ Although the website was publicly browsable from the very beginning, it was not so much an *open-content* project as a *not-closed* personal archive, inspired by the ideas *Creative Commons*³ was advocating at this time. We covered our own demand for easy accessible online storage, determined by spatial separation, **plus** we invested a little time to build a web-based user interface, which made it easier to browse and access the collection.

It was not our goal to build a *service*, like a clipart library or stock art database,⁴ intended to meet the demands of a broad audience. We just stored vector graphics online, that were not necessarily universal applicable, but experiments in techniques and thematics

¹Vector graphics describes the use of geometrical primitives such as points, lines, Bézier curves, and polygons to represent images in computer graphics. Inherent to vector graphics is a certain aesthetic that Jon Phillips describes with the five dominant features: Scalability, Reproducibility, Modularity, Precision and Abstraction. 1

²LAFKON is Christoph Haag and Benjamin Stephan. They are doing graphic design at the moment.

³*Creative Commons* can be credited for generating interest in the issue of intellectual property and contributing to the re-thinking of the role of the “commons” in the “information age”. 2
Although there were earlier initiatives like e.g. *copyleft_attitude* with their Licence Art Libre, *Creative Commons* were the first to attract broader attention.

⁴There are projects like the *Open Clip Art Library* that “aims to create an archive of user contributed clip art that can be freely used” 3 or the project *Neubau Welt* which “is an extensive encyclopaedia of well over a thousand pictograms of everyday objects and obsessions large and small”. 4

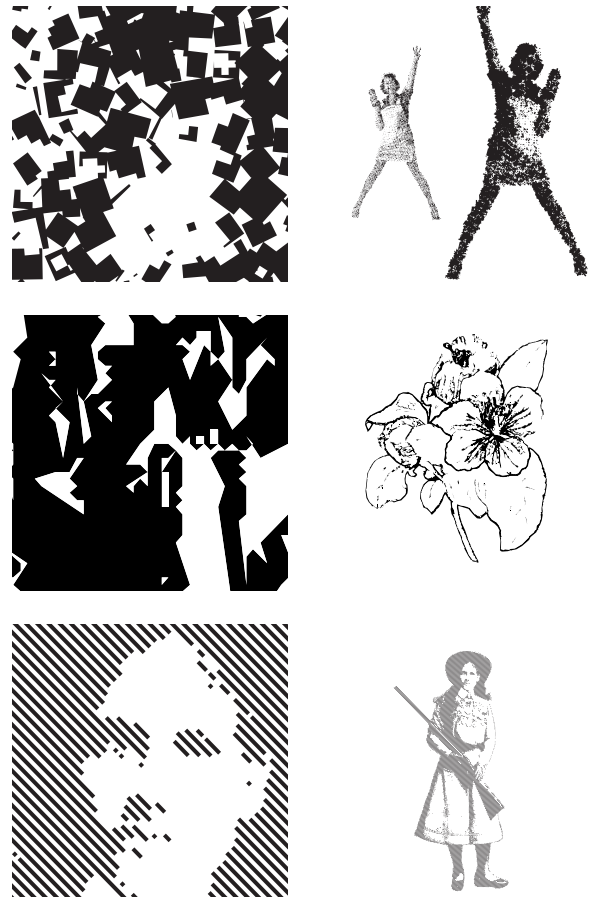


Figure 1: Different approaches to vectorization: randomize a duotone raster, autotrace -line-threshold 4, gif-dithering with 50% transparency.

Graphics were produced and used occasionally and while the repository became known to a broader audience, needs were realized and lessons learned, what made us decide to launch a major re-release in 2009.

2 What's up to change?

Several issues showed up during the years. Main subjects were technical structure, file formats, interoperability and the licensing of the artworks.

2.1 License

In 2004 we were looking for a license, that would allow people to mash-up, sample and share our graphics. We decided to choose a license, which was back then freshly launched by *Creative Commons (CC)* in collaboration with the experimental music and sound collage band [\[5\]](#) *Negativland* [\[6\]](#) :

Licensed under creative Commons Sampling License Plus+ by LAFKON 2004 

The launch of *Creative Commons Sampling License Plus* was combined with the publication of a cd-sampler, in cooperation with *Wired Magazine*, including songs from musicians like *Beastie Boys*, *Thievery Corporation*, *Le Tigre* or *Matmos* who “invited people to play with their tunes, not just play them.” [\[7\]](#)

The whole concept sounded appealing, including the possibility to transfer a mentality of collage and sampling from (electronic) music to graphic design:

The rest is the remix. Unpack the meanings, unstuff the fragments [\[8\]](#)

2.1.1 Then: Sampling License Plus 1.0

Part of a song, used in another, the use of the drum introduction from Led Zeppelin’s “When the Levee Breaks” in songs by the Beastie Boys, Dr. Dre, Eminem, Mike Oldfield, Rob Dougan, Depeche Mode and Erasure, the guitar riffs from Foreigner’s “Hot Blooded” Tone-Loc’s “Funky Cold Medina” spoken words from movies, as electronic music, the musique concrète style, based almost entirely on samples and sample-like parts, hip hop, developed from DJs repeating the breaks from songs. [\[9\]](#)

Sounds cool, did not work (for us).

CC promises an easy access to the understanding of open content licenses, but actually Sampling Plus 1.0 was nothing like this. *Creative Commons* states in its license that you “may not use this work to advertise for or promote anything but the work you create from it”. [\[10\]](#)

This is quite rare in graphic design and furthermore difficult to define. Does a website design promote anything but itself? Is a poster design an independent artwork or does it advertise for something?

The license states further:

You are free:

- To sample, mash-up, or otherwise creatively transform this work for commercial or noncommercial purposes.
- To perform, display, and distribute copies of this whole work for noncommercial purposes (e.g., file-sharing or noncommercial webcasting). [\[10\]](#)

Where starts *creatively transform* and where *copy as whole*? While *copy as whole* seems quite easy to define, *creative transformation* can be quite subjective. Is

changing colors a creative transformation? Is rearrangement creative per se. Or to provide Mona Lisa with a nice moustache?⁵

Only if your work has satisfied the definition of a transformatory work, you are allowed to use it commercially. While CC wants to easify reuse, we have a primal distinction that’s highly discussable in most cases (and we did often) and makes the adherence to the license difficult, even more with CC’s spongy definition of what is commercial and what is not.⁶ We made common⁷ mistakes, which didn’t make it easy for users to comply to the license. and often **we ourselves** could not tell how a reasonable handling would function.

The **simple** usage of our vectorgraphics, without legal borders, as CC promised, was basically **impossible**.

The fieldtest of five years maintaining the vektorDB had proven that the Sampling License Plus 1.0 does not meet our demands. Because we wanted to stick to certain preconditions, like explicit allowance of commercial usage, attribution of authors and ideally a copyleft, we neither wanted to release our works into the *public domain*⁸ nor did we want to choose a simpler, more restrictive CC license.

The possibility of commercial application is an important point, if you do not want to exclude a wide range of professional uses. Additionally the border between professional and amateurs is in such an extent fuzzy these days, that a non-commercial clause can be a major problem for *Free Licenses*,⁹ not only for the CC licenses. What bothered us most, regarding a donation to the public domain, was the fact, that it allows usage without giving the com-

⁵When Marcel Duchamps did so, it was regarded as “an example of a highly transformative work that accomplishes its transformative effect with what seems to be a minimum of added material”. [\[11\]](#)

But this happened in the world of art where different rules apply. [\[12\]](#)

⁶Qualifies the a google banner ad as *commercial use*? Do you really want to exclude this kind of users? [\[13\]](#)

⁷there is exact definition how to attribute, while the license states “You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work)” [\[10\]](#). This mistake can be found an many websites which sport a CC lifestyle icon.

⁸The Open Clip Art Library is placed into the Public Domain [\[3\]](#)

⁹*Free content is no longer a fringe movement – it is something millions of people use every day. Wikipedia, a free content encyclopedia built by volunteers, contains over 10 million entries in more than 200 languages and is among the 10 most visited websites on the planet. Moreover, its growth continues, as does its integration into search engines. Google features Wikipedia definitions in some queries, as well as through the integration of Wikipedia mirror Answers.com in the top right corner of search results. Other search engines, such as Amazon.com’s A9, Clusty.com, and Web.de have even integrated Wikipedia directly into their user interfaces.*

This success is the result of less than a decade of work. Clearly, free content is here to stay. But, in part to make uses like the above possible, free content sites like Wikipedia explicitly allow and encourage commercial use. As we will see, there are many desirable commercial uses. More importantly, however, if you choose an -NC license, your work will not be compatible with Wikipedia, Wikinews, Wikibooks, and similar free content projects which have more permissive philosophies and practices. [\[14\]](#)

munity something back, while we reckon this backflow as a crucial point for the leverage of *Free/Libre/Open Source Software*.

We decided to stick to a copyleft license, but realised soon, that while guaranteeing commercial applicability and backflow, a copyleft license poses a major problem: If you use content under a copyleft license, the license's intension is to make every addition, change or improvement available again under copyleft. Imagine somebody wants to use a graphic from the vektorDB in a book? What must be copylefted? The whole book? The layout? The modified graphic?

2.1.2 Now: Dual-Licensing

For the new licensing scheme for the vektorDB we want everybody to be able to use the graphics freely, in a sense of *Free/Libre/Open Source Software*, while at the same time we do not want to exclude and alienate people who are not so firm or ideological about *FLOSS*.

Beside our own interest in a *Free Culture Movement*, we believe that in graphic design there is definitely a use for *Free/Libre/Open Source Software* alike solutions,¹⁰ but we also recognize, that at most projects designers' work and customers' data are woven together in such an extent,¹¹ that's it's quite impossible to copyleft them. To shorten complicated discussions what's actually affected by copyleft (design,content design+content) we decided at this point, if somebody does **not** want to **spend time** (and it costs actually time to think about how, what and why copyleft affects your work and why it's worth to engage in *Free/Libre/Open Source Software*), they can **spend money** instead.

We choose a dual-licensing approach¹² that combines

¹⁰One could imagine, a demand for predefined, modifiable solutions (cf. *icons* or *pictograms*) in commercial graphic design. The time for commissioned work is often very short, what creates dependence on ready-made solutions (if there is not enough manpower to produce everything inhouse). At the moment this leads to the use of external resources e.g. *commercial stock art libraries* or a loss of quality, but it would be easy to imagine a replacement or complement through a free (*like speech and beer*) community "code" base here, similar to *Free/Libre/Open Source Software*. And there is a quite hedonistic reason. Through a *Free/Libre/Open Source Software* approach you can realize and publish projects without the restrictions that commercial projects often impose on you:

"In that world, the working programmer's normal experience includes being forced to use broken tools for political reasons, insane specifications, and impossible deadlines. It means living in Dilbert-land, only without the irony. It means sweating blood through the forehead to do sound work only to have it trashed, mangled, or buried by people who couldn't write a line of code to save their lives. If you love programming, trying to do work you can be proud of in this situation is heart-breaking. You know you could do better work if they'd just goddamn give you room to breathe." 15

¹¹basically that's a technical problem that has to be solved through a strict separation of form and content

¹²For example Sun provides its MySQL database server and MySQL Client Libraries under a dual license model designed to meet the development and distribution needs of both commercial distributors and open source projects. [...] Distributors that combine and distribute commercially licensed software with MySQL software and do not wish to distribute the source code for the under the GNU General Public License must enter into a commercial license agreement with Sun. [...]

strict copyleft¹³ with a standard copyright agreement through a usage fee for closed-source requirements.

As an example serves how MySQL, now Sun, combines an opensource license with a non-opensource license, without them violating each other. MySQL offers two licenses for its code repository.

MySQL AB sells non-GPL commercial licenses for embedded and other applications. A non-GPL commercial license allows a user to resell a product that contains MySQL code without GPLing the entire application, and it does not violate GPL rules or intentions.

17

According to the dual-licensing there are two possibilities in our case:

- You choose a free license (GPL,GFDL,CC-BY-SA) and you are free to use, to study, to make and redistribute copies, to make changes and improvements, if you share alike, according to the license.
- You choose the commercial license, pay a fee and you are allowed to use the image according to a standard copyright license.

Through charging a usage fee, commercial and closed-source requirements of a graphic designer's daily needs can be satisfied, while at the same time a **free archive**, free as in *Free Culture*, can be maintained without compromises.

And while the **free archive** grows bigger, the incentive for commercial graphic design to contribute to a *pool* of *Free Culture* will also grow bigger.¹⁴

The main difficulty in dual-licensing is, that you can only dual-license something that you own all rights for. If e.g. MySQL wants to include code contributed by the community, the contributor has to resign from his rights e.g. sign an agreement. That procedure is quite laborious and the cause why code submissions into MySQL's repository are quite rare.

In our case that does not really pose a problem, because at the moment we are more or less the sole contributor to the graphics database.

Even if we host contributions, the database is modular enough to differentiate at bottom level. A different license can be assigned to each graphic. We can have a dual license for some graphics, while others can be published exclusively under *Free Licenses*.

For developers of *Free Open Source Software* ("FOSS") applications under the GPL that want to combine and distribute those FOSS applications with MySQL software, Sun's MySQL open source software licensed under the GPL is the best option. 16

¹³a license that meets the standards of *Freedom Defined*

¹⁴*Je umfangreicher und attraktiver die öffentliche Bibliothek von GPL-lizenziertem Programmcode wird, desto größer ist auch der Anreiz für Programmierer, sich ihr zu bedienen und damit nolens volens zu ihr beizutragen.* 18

Copyleft Triple Licensing Because there exists a variety of different *Free Cultural* licenses, we are looking for a solution that guarantees compatibility with *the most common Free Licenses*.

Copyleft licenses prescribe derivative works to be republished under the exact **same** license. Therefore different licenses automatically exclude each other.

Example:

Wikipedia is licensed under the GFDL, which prescribes, that everything that is licensed under the GFDL, must stay under the GFDL.

If our graphics would be licensed solely under the CC-BY-SA (SA stands for Share-Alike and determines that derivative works must be published under the same license) it would be impossible to combine vektorDB graphics and wikipedia content, because one license would be violated anyway. Because we license our graphics under GFDL **and** CC-BY-SA, you can choose the **GFDL** in this case, without violating our terms.

Although the GFDL is widespread, it is unfortunately a license which is not easy to understand. Therefore a lot of people, who are not so familiar with a *Free Culture* approach, tend to choose different licenses, e.g. a *Creative Commons* license.

In general there are cases where one license does simply fit better (*for a special purpose*) than another.

Example:

To comply to the GFDL you have to attach a copy of the license, about 3 A4 pages. This is no problem for a book with 1000 pages, but what about a single photography?

19

To prevent a one-way-trap through exclusive single-licensing, a common practice is to choose multiple licenses.

We choose three licenses which all meet the requirements of *Freedom Defined*.¹⁵

GNU General Public License,
GNU Free Documentation License,
Creative Commons-Attribution-ShareAlike License.

With these three licenses, we keep three directions open. As soon as a licensee selects *one* license, only *one* license can apply from there on (cf. Figure 2). To prevent this, we encourage people to also triple-license resulting works, if possible. Seems complicated, but actually it isn't that bad.

2.2 Structural changes

Beside the relicensing, chance is taken to improve the structure in general.

2.2.1 Clearing contents

For a proper appliance of the licenses we cleaned the database from *possible* copyright violations. There were several files which could be acceptable in an art con-

¹⁵The definition was initiated by Erik Möller as a means to resolve ambiguity about the phrase "free content" in the context of the Wikimedia project family. It was inspired by the Free Software Definition.

20

text,¹⁶ but which were highly dubious when it comes to relicensing (cf. Figure 3). We took the chance to clean the database from copyright violation.

Non-free copyrighted material cannot be freely incorporated into one's work no matter what license one chooses. Even worse, the opposite is true: copyright owners are most likely to categorically refuse clearance for anything that will be put into free circulation because the license of the work incorporating their's would effectively relicense the former. 12



Figure 3: copyright violating graphic made for the animation movie *Trusted Computing*

2.2.2 Then: Closed-Source Fileformats

The initial fileformat for the vektordb was *Adobe's* Illustrator format (.ai). Unfortunately .ai-files were incompatible or difficult to handle with *Free/Libre/Open Source Software* and therefore excluded many users and users.

```
<BjornW> anyone here that has Adobe Illustrator and
willing to transcode a few images?
<mlinksva> BjornW, I don't, but I'm curious what sort
of image can only be read by photoshop
<BjornW> mlinksva: stupid .ai files,
which make inkscape choke. Check here:
http://vektordb.lafkon.net/index.php
<BjornW> mlinksva: pretty cool images perfect for my
presentation, but the ai files cannot be read
by inkscape, which sucks :)
<mlinksva> oh, i misread illustrator as photoshop :)
<mlinksva> those are some cool images
<BjornW> mlinksva: yeah, but they won't work on
inkscape and I have no clue why not
<BjornW> mlinksva: would be great to have these im-
ages available as svg :)
<mlinksva> BjornW, have you tried ill2svg?
http://inkscape.org/tools.php
<BjornW> mlinksva: not yet, will try now. Thanks!
```

21

Not only to stop an outflow of users, but also regarding sustainability, we change from the proprietary .ai format to open standards, centering "*distribution radically*

¹⁶Kurt Schwitters was not sued for collaging the logo of German Commerzbank into his "Merz" painting which yielded his "Merz" art. Neither did Andy Warhol receive injunctions for using Coca Cola's and Campbell's trademarks. As long as these symbols remained inside the art world, they did not raise corporate eyebrows. 12

Single Licenses

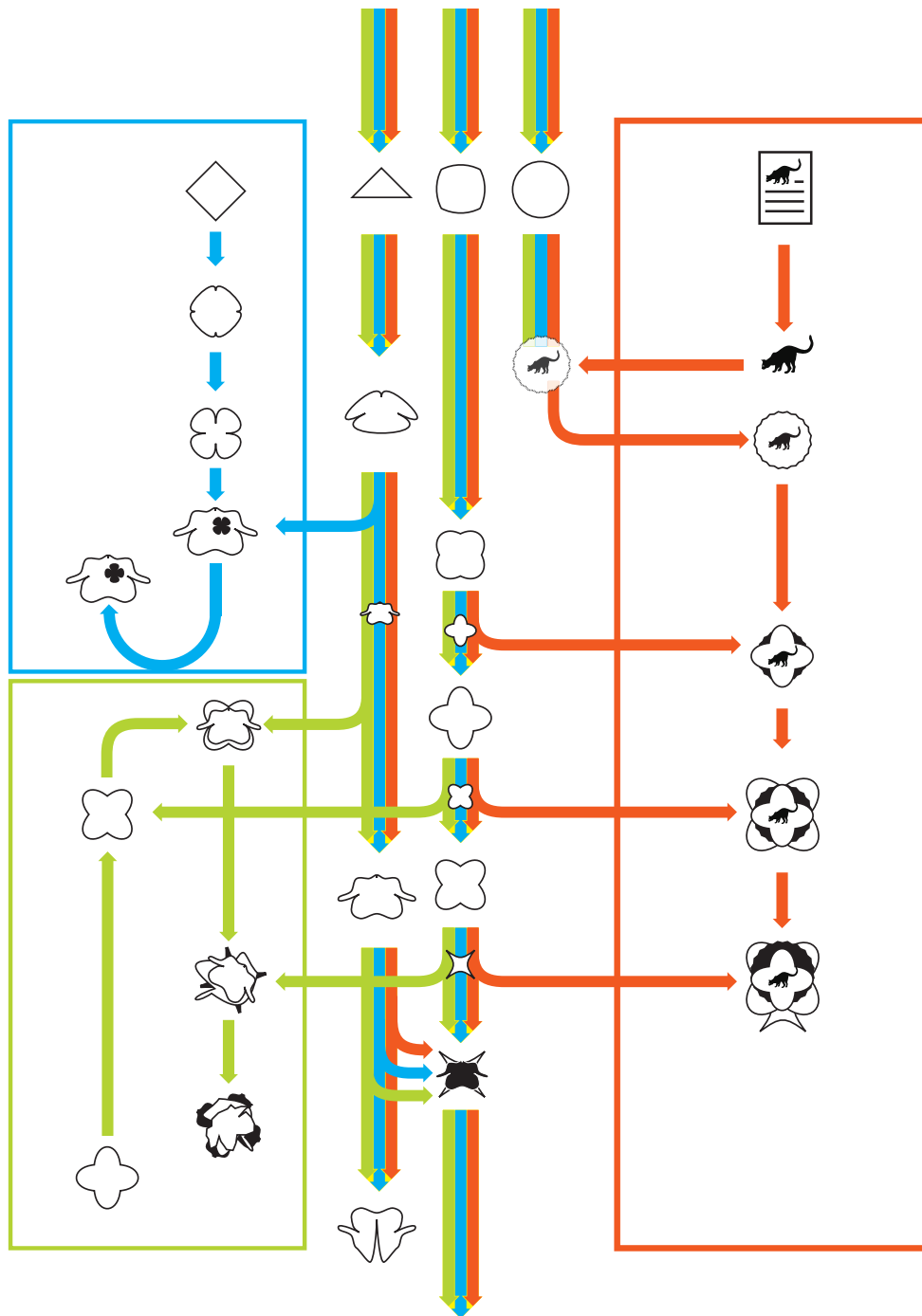
e.g. @flickr (*CC-BY-SA*),
a lot of software (*GPL*)

Multiple Licenses

e.g. vektorDB (*GPL, GFDL, CC-BY-SA*)

Single License

e.g. Wikipedia (*GFDL*)



Single-licensed content merged/combined with multi-licensed content produces single-licensed content.

We recommend to triple-license your works before merging with single-licensed content (**transformation before combination**). 3 licenses provide more compatibility for future uses.

Example:

You transform a graphic from the vektorDB (triple-licensed) to combine it with a Wikipedia-article (GFDL only). The combination of the graphic and the article **must** be licensed under the GFDL, For the transformed graphic, you as the licensor, have the right to license the graphic under GFDL **and** GPL **and** CC-BY-SA. This way it can be combined with CC-BY-SA works or included into GPLed software. The combination, by force solely under GFDL, can only be combined with GFDLed content.

Figure 2: Flow of Licenses

upon the possibility to be spread, viewed and modified with entirely free means, at the expense even of technically smoother or more integrated proprietary standard solutions.” [22]

2.2.3 Now: Open standards

All graphics have been converted to open vector formats PDF and SVG. *Portable Document Format (PDF)* was created by Adobe Systems in 1993 for document exchange. It is used for representing two-dimensional documents in a manner independent of the application software, hardware, and operating system. While formerly a proprietary format, PDF was officially released as an open standard on July 1, 2008, and published by the ISO as ISO 32000-1:2008.” [23] *Scalable Vector Graphics (SVG)* is an open standard, developed by the World Wide Web Consortium (W3C) since 1999. “*SVG images and their behaviours are defined in XML text files. This means that they can be searched, indexed, scripted and, if required, compressed.*” [24] SVG is widely supported and can be visually edited, amongst others, with the *Free Software Inkscape* on GNU/Linux, OS X and Windows. As an open standard it can be used within many applications, e.g. it can be directly imported as vector¹⁷ shapes into *Processing*¹⁸ sketches.

3 Further on: Forkable

While the old vektorDB was more or less a webfrontend, the broadened approach towards a *Free/Libre/Open Source Software inspired graphics repository* also includes the opening of the internal structure. This internal structure holds not only the graphics of the vektorDB, but is a collection of *utilities, generators and code* for the experimental treatment of graphic. The directory tree is browsable at <http://www.forkable.eu>.

The collection is a mirror of a working production environment and serves at the same time as an instrument for the documentation, traceability and opening of a working process. It is centered around hackability through human-readability¹⁹ and interconnectivity through modularity.

Regarding readability, we are in favor of plaintext notation, so files can be read, scripted and processed with the palette of common unix tools.

This is because plain text is a universal interface; that is, it can allow programs to easily

¹⁷The svg candy library by Michael Chang [25], worked at the time only with svgs exported from Adobe Illustrator. We wrote a GNU/Linux-based conversion utility, to be found at <http://www.forkable.eu/vektorconverters/v4p5.sh>. It supports also current *Processing*'s PShape()

¹⁸*Processing* is a web site, a programming environment for learning computational design, a sketchbook for rapidly prototyping [...], a new 2D and 3D graphics api and rendering engine for java, an open project started by casey reas and myself an active community of a few thousand people [...] [26]

¹⁹This attempt ranges between the desire to work with openformats and the discovery of the power of plaintext notification, respectively the power of plaintext manipulation.

interact with each other in the form of text outputs and inputs. The latter means that it is easy for humans to study, correct, improve and extend such files as well as to port (i.e., modify) them to new platforms (i.e., other combinations of operating systems and hardware).

[27]

Text as interface lays also the foundation for enough modularity to interconnect things like GNU/Linux, L^AT_EX, *Processing* and human beings.

While *Forkable* is currently on a test run there have been some projects realised. In regular intervals several scripts produce graphics as variations or replacements for different purposes. *Bashscripts* combine different functionalities e.g. unix tools and java applications written in *Processing*.

```

GRAPHICSET=minimartels
ERRORLOG=/raid/forkable/error.log
RUNLOG=run-`date +%s`.log
#RUNLOG=run.log

# CAUSE SCRIPT TO STOP ON ERROR
set -e

# START VIRTUAL XSERVER FOR PROCESSING HEADLESS #####
# Xvfb :1 -screen 0 1152x900x8 -fbdir /tmp &

# EXPORT DISPLAY FOR PROCESSING HEADLESS #####
export DISPLAY=localhost:1.0

# RUN PROCESSING SKETCH #####

APPDIR=$(dirname "$0")

LIBSKETCH=$APPDIR/i/lib/minimartels.jar
LIBCORE=$APPDIR/i/lib/core.jar
LIBXML=$APPDIR/i/lib/xml.jar
LIBCANDY=$APPDIR/i/lib/candy.jar:$LIBXML
LIBPHYSICS=$APPDIR/i/lib/physics.jar
LIBITEXT=$APPDIR/i/lib/itext.jar
LIBPDF=$APPDIR/i/lib/pdf.jar:$LIBITEXT

# MINIM #
JL=$APPDIR/i/lib/jll.0.jar
MINIM=$APPDIR/i/lib/minim.jar
MP3=$APPDIR/i/lib/mp3spi1.9.4.jar
TRIAOS=$APPDIR/i/lib/tritonus_aos.jar
TRISHA=$APPDIR/i/lib/tritonus_share.jar

LIBMINIM=$JL:$MINIM:$MP3:$TRIAOS:$TRISHA

LIBS=$LIBSKETCH:$LIBCORE:$LIBPHYSICS:$LIBPDF:$LIBMINIM:$LIBCANDY

java -Djava.library.path="$APPDIR" \
-cp "$LIBS" \
minimartels && SRUNLOG &

# AUTOMATICALLY EXIT PROGRAM AFTER X SECONDS
# IF JAVA PROCESS IS STILL RUNNING
JID='echo $!'
sleep 180;
CATCHJID='ps aux | grep $JID | sed -n '/grep!p' | wc -l'
#-----#
CATCHERROR='grep 'Stopped on' SRUNLOG | wc -l'
if [ $CATCHJID -ge 1 ]; then

kill -9 $JID

echo "-----" >> SERRORLOG
echo "" >> SERRORLOG
echo "$0 aborted on "`date`" >> SERRORLOG
echo "" >> SERRORLOG
cat SRUNLOG >> SERRORLOG
echo "" >> SERRORLOG
echo "-----" >> SERRORLOG

rm SRUNLOG
exit 1;

else
#-----#

#####
SUBJECT="Minim powered minimal Particles"
INFO=""

#####
echo "InfoKey: Title" >> metadata.txt
echo "InfoValue: Minimartels" >> metadata.txt
echo "InfoKey: Subject" >> metadata.txt
echo "InfoValue: $SUBJECT" >> metadata.txt
echo "InfoKey: Keywords" >> metadata.txt
echo "InfoValue: $INFO" >> metadata.txt
echo "InfoKey: Author" >> metadata.txt
echo "InfoValue: Christoph Haag / LAFKON" >> metadata.txt
#####

# SELECT OLDEST FILE #####

```



```

FILE='ls o/s.pdf --sort=time | tail -1'
FILENAME='echo ${FILE##*/} | cut -d "." -f 1'

# RENAME AND MOVE TO OUTPUT FOLDER #####
pdftk mnmrtcls.pdf update_info metadata.txt \
  output $FILE

# GENERATE THUMBNAIL #####
convert -resize 200 \
  -crop 150x150+20+20 \
  $FILE \
  $FILENAME.tiff

# WRITE DATE ON THUMBNAIL #####
MARK='date +%d.%B %Y %T'

convert -fill white -draw "rectangle 1,130,149,145" \
  -font helvetica -pointsize 10 \
  -fill black +antialias \
  -draw "text 3,141 '$MARK'" \
  $FILENAME.tiff \
  o/$FILENAME.gif

# GENERATE FTP COMMANDS #####
ACCESS='cat ~/.forkable/ftp.input'

echo $ACCESS > ftp.tmp
echo "put $FILE www/$GRAPHICSET/${FILE##*/}" >> ftp.tmp
echo "put o/$FILENAME.gif www/$GRAPHICSET/$FILENAME.gif" >> ftp.tmp
echo "bye" >> ftp.tmp

mv ftp.tmp ftp.input

# UPLOAD VIA FTP #####
ftp -n vektordb.kilu.de < ftp.input

rm $FILENAME.tiff
rm mnmrtcls.pdf
rm metadata.txt
rm SRUNLOG
rm ftp.input

fi

exit 0;

```

These examples should give an idea.

3.1 www.forkable.eu/generators/lacmachine

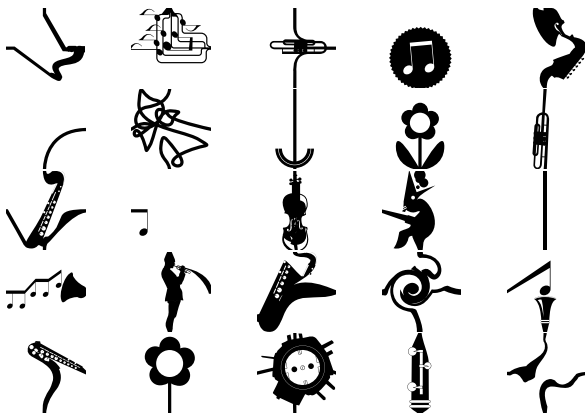


Figure 5: Parts for generative posters

In 2007 LAFKON created an graphical identity for the Linux Audio Conference 2008. Additional to flyer, program, website and logo, *generative* posters were developed.

These posters are automatically generated via a *BASH/Processing* routine. This routine selects illustrations from the vektorDB, layouts them and stores print-ready *PDFs* online. Posters are under standard copyright, while generators and modular illustrations are free to use, to study, to make and redistribute copies, to make changes and improvements.

3.2 www.forkable.eu/generators/minimartcls

minimartcls generates a simple graphic through a feedback-driven particle system.

<http://www.forkable.eu/generators/minimartcls>
<http://www.forkable.eu/stock/2007/minimartcls>

Code and graphics are online and free.

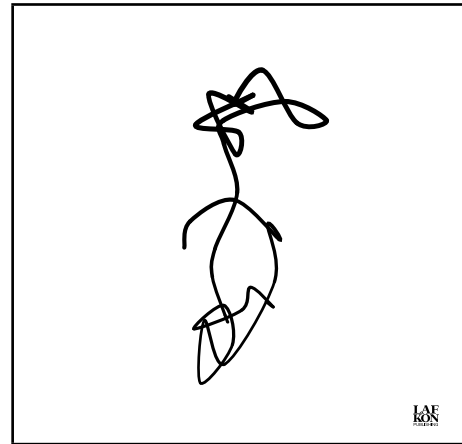


Figure 6: Automatically generated vector graphic

3.3 www.forkable.eu/generators/elytrigia

elytrigia loads vektorDB graphics, layouts them according to a growing algorithm, generates a graphic and stores it in the vektorDB.

<http://www.forkable.eu/generators/elytrigia>
<http://www.forkable.eu/stock/2008/elytrigia>

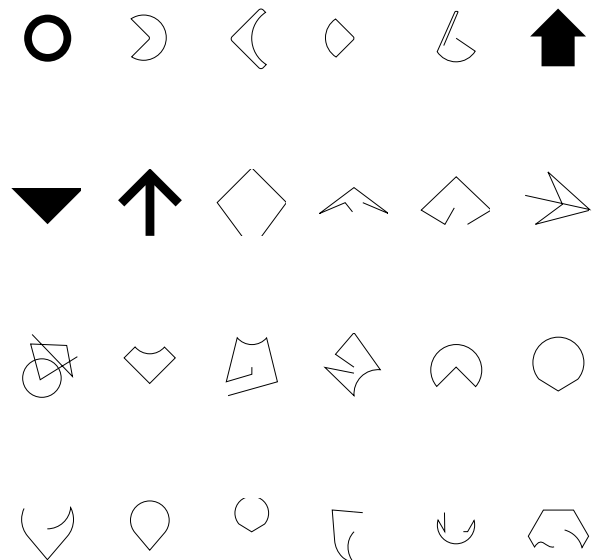


Figure 7: Scalable Vector Graphics called by the *Processing* sketch *elytrigia*

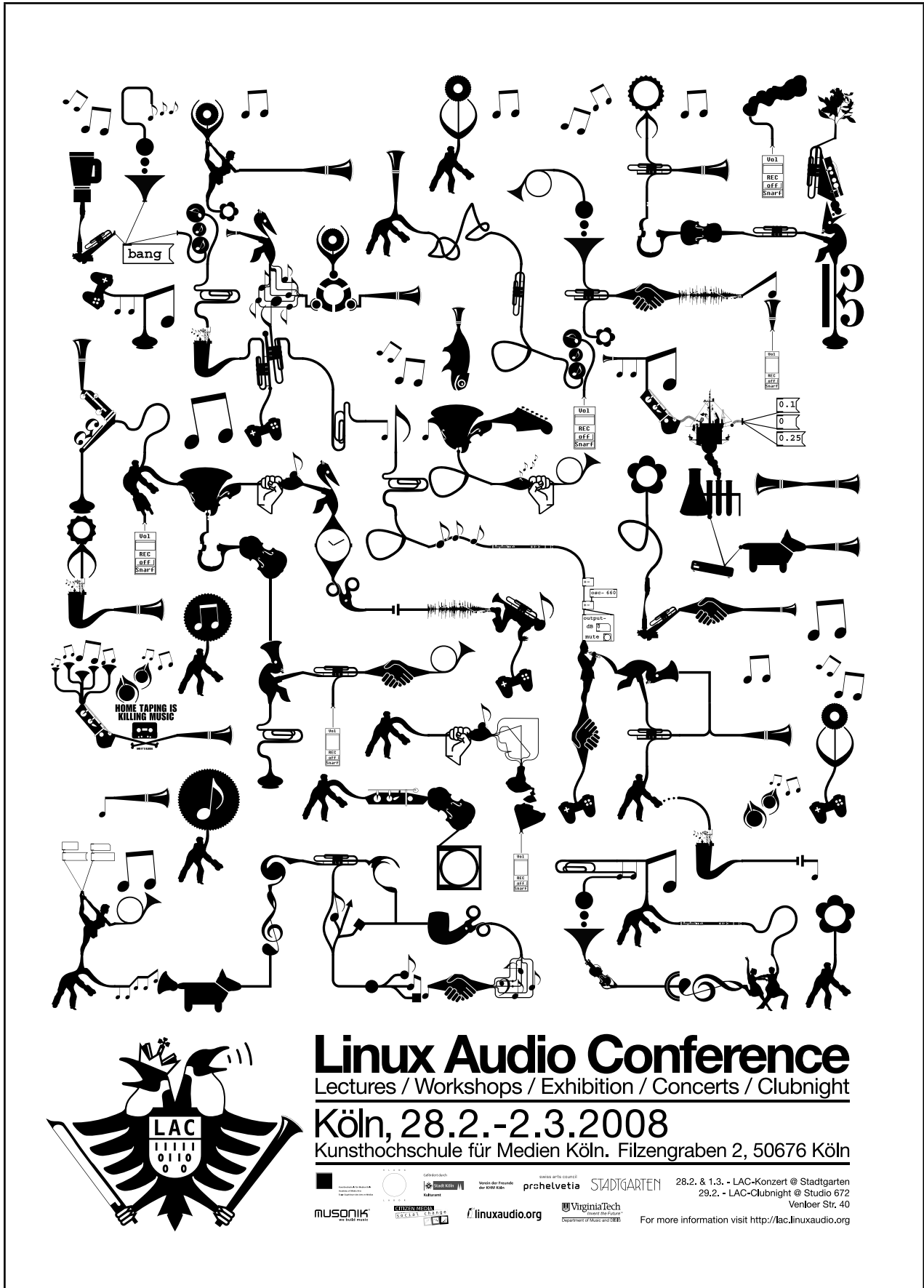


Figure 4: Generative Poster for the Linux Audio Conference 2008, one of 1000s automatically generated variations

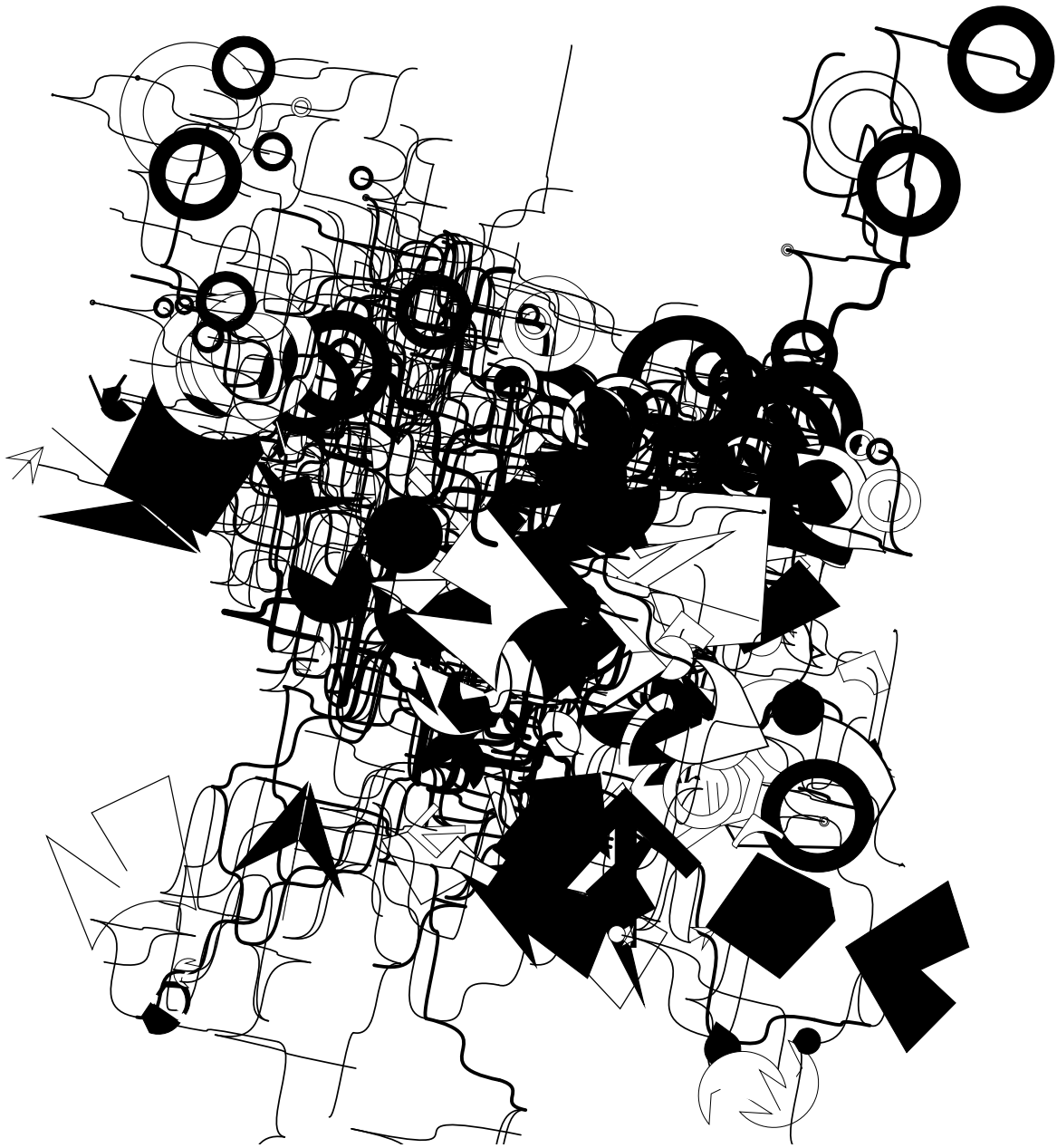


Figure 8: Graphic generated by the script <http://www.forkable.eu/generators/elytrigia>

4 Conclusion

Our primary goal is doing graphic design, not writing software or doing webhosting. But graphic design, in our perception, is closely connected with the experiment on tools.²⁰ Nowadays, if you want to experiment on tools there seems to be no way around *Free/Libre/Open Source Software*. In the spirit of experimentation our primary intension is not cloning classical graphic design tools for GNU/Linux,²¹ but learning through looking at **graphic design in a different way**.²²

5 Acknowledgements

I would like to thank Benjamin Stephan for advice and help on this paper and in general for the creative work on the vektorDB.

References

- [1] John Phillips. Vector Aesthetics. 2004. [Online; accessed 6-March-2009].
- [2] Wikipedia. Creative Commons — Wikipedia, The Free Encyclopedia, 2009. [Online; accessed 6-March-2009].
- [3] Open Clip Art Library. Open Clip Art Library, 2009. [Online; accessed 4-March-2009].
- [4] Die Gestalten. Neubau Welt, 2005. [Online; accessed 4-March-2009].
- [5] Wikipedia. Negativland — Wikipedia, The Free Encyclopedia, 2009. [Online; accessed 4-March-2009].
- [6] <http://cccip.org/blog/michael>. NEGATIVLAND SAMPLING MOVIE, 2004. [Online; accessed 04-March-2009].
- [7] Creative Commons. The WIRED CD: Rip. Sample. Mash. Share., 2004. [Online; accessed 4-March-2009].
- [8] Paul D. Miller. *Rhythm Science*. The MIT Press, 2004.
- [9] Wikipedia. Sampling (music) — Wikipedia, The Free Encyclopedia, 2009. [Online; accessed 5-March-2009].
- [10] Creative Commons. Sampling Plus 1.0, 2008. [Online; accessed 6-March-2009].
- [11] Wikipedia. Transformativeness — Wikipedia, The Free Encyclopedia, 2009. [Online; accessed 18-March-2009].
- [12] Florian Cramer. The Creative Common Misunderstanding, 2006. [Online; accessed 5-March-2009].
- [13] Erik Möller. The Case for Free Use: Reasons Not to Use a Creative Commons-NC License. In Bernd Lutterbeck, Matthias Bärwolff, and Robert A. Gehring, editors, *Open Source Jahrbuch 2006 – Zwischen Softwareentwicklung und Gesellschaftsmodell*. Lehmanns Media, Berlin, 2006.
- [14] Freedom Defined. Licenses/NC, 2009. [Online; accessed 17-March-2009].
- [15] Eric S. Raymond. Why I Hate Proprietary Software, 2008. [Online; accessed 5-March-2009].
- [16] Sun Microsystems, Inc. Commercial License for OEMs, ISVs and VARs, 2008. [Online; accessed 16-March-2009].
- [17] Nancy Cohen. The Many Faces of MySQL, 2007. [Online; accessed 3-March-2009].
- [18] Florian Cramer. Vom freien Gebrauch von Nullen und Einsen - „Open Content“ und Freie Software, 2002. [Online; accessed 20-March-2009].
- [19] Wikimedia. Commons:Lizenzen, 2009. [Online; accessed 20-March-2009].
- [20] Freedom Defined. History - Definition of Free Cultural Works, 2006. [Online; accessed 16-March-2009].
- [21] BjornW + mlinksva. IRC Protocol, 2007. [Online; accessed 5-March-2009].
- [22] Florian Cramer. What Does “Open” Mean in Digital Arts?, 2002. [Online; accessed 20-March-2009].
- [23] Wikipedia. Portable Document Format — Wikipedia, The Free Encyclopedia, 2009. [Online; accessed 6-March-2009].
- [24] Wikipedia. Scalable Vector Graphics — Wikipedia, Die freie Enzyklopädie, 2007. [Online; Stand 17. Dezember 2007].
- [25] Michael Chang. SVG Candy Processing Library, 2006. [Online; accessed 22-March-2009].
- [26] Benjamin Fry. BEN FRY - Processing, 2008. [Online; accessed 16-March-2009].
- [27] The Linux Information Project. The Unix Philosophy: A Brief Introduction, 2006. [Online; accessed 25-March-2009].
- [28] Eric S. Raymond. Basics of the Unix Philosophy, 2008. [Online; accessed 20-March-2009].
- [29] Alan D. Sokal. Transgressing the Boundaries: Towards a Transformative Hermeneutics of Quantum Gravity. *Social Text*, 46(4):217–252, 1996.

²⁰As historical reference one could mention inventions that were made for graphical reproduction, e.g. letterpress or lithography

²¹like Inkscape as a replacement for *Illustrator* or Scribus for *InDesign* (both from *Adobe*)

²²A big inspiration had been an approach that's described as the *unix way*. You could write much more here, but because that won't be done, just some buzzwords: **modularity, text as universal interface, release early, release often, avoid hand-hacking, there is no rule 6**