Boundaries of Openness in the IT area

Michael Sonntag

Institute for Information Processing and Microprocessor Technology (FIM)

Johannes Kepler University Linz

sonntag@fim.uni-linz.ac.at

Abstract

The Magic of Open Everything: Open Source, Open Communities, Open Access is often evoked as the cure for most current problems in the IT area. E.g. proprietary software as well as bugs will die out because of Open Source (Microsoft vs. Linux; better quality through intensive reviews). Similarly, Open Communities, e.g. Wikipedia, hope for contributions of everyone, but have reduced their openness lately. And while Open Access to scientific papers possesses obvious advantages, nevertheless its success has not been very widespread to date. Obviously this "Openness" is not limitless, but some boundaries exist. Several of them are investigated in this paper categorized according to the type of the boundary: Technical, legal and economical.

1. Introduction

Openness is partially seen as the solution to everything: Open source will revolutionize software production and replace proprietary software. Similarly, open communities (loosely organized collaborative efforts where anyone may join and the results of the common goal are available for free) revolutionized for instance the encyclopaedia market: Wikipedia “destroyed” many large and reputed others (e.g. Microsoft Encarta will be closed down on 31.10.2009 [13]). Another “open” area are scholarly publications: Open access journals become more important and prominent, with funding societies increasing the pressure to publish there.

But is openness always the best or only solution? Wikipedia for instance is often criticized for its unreliability: Anyone can change (almost) any data. This has been used e.g. during elections to “modify” the biography of candidates [4]. However, every kind of openness possesses some limitations which cannot be avoided in real life. Some of them will be discussed here as examples, to show that openness in projects is not always that open as it seems to be at first. Rather it is more open than conventional approaches, but often still quite narrow and limited in other aspects. Although other, like psychological, limits exist, only the following are discussed here:

1. Technical boundaries: These can be subdivided into systemic limitations, like everyone can read the Linux kernel source code, but only few computer scientists and no laymen can actually understand it, and intentional ones. An example for the latter is the so-called Tivoization [21], where hardware is used to prevent exercising the rights coming with open source software.
2. Legal boundaries: Currently disputed is for instance the possibility of the eBook reader Kindle to read aloud electronic books by speech synthesis. Authors protested against it as an infraction

of their rights, so it can now be disabled by the eBooks publisher. Laws can require some limits to openness, so either we have to live within these fences or change them.

3. Economic boundaries: Not every “open” approach is economically feasible. Some persons might donate their work, but everyone has to make a living. As the dot-com bubble showed, a valid economical model is a requirement for all durable undertakings. This might be very small (numerous contributors of small elements for free; little general costs), but could also be quite large (like the Linux kernel: Many persons are paid by companies for contributing because these obtain an indirect benefit).

2. Technical boundaries

Some boundaries are based on technical issues: It can (or should) not be open, because then some additional features become possible.

2.1. Linux kernel

The Linux kernel [11] and many other open source software may be technically free in the sense that everyone can see the code, modify it, compile it, or suggest modifications for inclusion into the “official” version. However, very few people can actually understand the code. Even most computer scientists would have trouble understanding the source, especially at the lowest level where assembler code is intermixed with C code. Therefore the openness is in reality very often rather a theoretical openness: If you invested enough effort (and perhaps money) you could exploit the openness, but in reality you do not. This is exemplified by the Linux distributions: Although there are numerous, only very few of them are actually widespread. They closely resemble proprietary software: Instead of installing Microsoft Windows you install Suse Linux or Redhat Linux. In all three instances you have to trust the “manufacturer” to have written/assembled a valid and good package of software.

Still this is only a weak boundary: As alternative distributions do exist you can rely somewhat on the openness. It is just something done rarely and needing a lot of effort. But having the option can be extremely important. For instance, if you must use Windows 95, there is no possibility to modify it or close security holes because of the lacking source code. If using a Linux kernel version 2.4 or even older, you can do it, it’s just an economic decision whether it is worthwhile doing so – but technically it’s not impossible.

A related aspect here is trust: You have to trust a proprietary software producer that their product is secure. This might be realistic for large ones with a dedicated focus on security, but for smaller ones – who knows? But still there is at least the possibility to identify this person (see also below in section 4.4) and the option of bringing legal action in case of malfunctions. In open source software neither a specific company or person can be assessed on their trustworthiness (Novell just packages most of the code) nor exists someone to start legal proceedings against (see also 3.2), as many developers are quite "anonymous" and difficult to find and possess to few funds to pay damages or reimburse for attorney or court expenses.

2.2. Tivoization

The TiVo [18] is a hardware product for recording TV programs. It contains open source software (Linux Kernel and Gnu software), which is made available as required by the license, and anyone can modify it. However, to run any software on this specific hardware it must be signed by a cryptographic key, which is kept secret by the manufacturer. This means that the code may be (re-)used everywhere else, except on the specific device. The result is that only "original" TiVo's are in circulation. Here a loophole of the GPLv2 is exploited by providing the software, but keeping the necessary data (the private key used for signing) secret (the public key is embedded in the hardware sold). So although running open software, technical measures prohibit using one of its fundamental ideas.

While divergent opinions exist on whether this is (or should be) allowed [19], openness is obviously limited here by design. A similar measure is employed in the OLPC project [15] with the intention to prevent the use of stolen laptops. Here the restriction can be removed in a complicated way and after some delays (to allow for checking whether this specific piece of hardware was stolen). These limitations could easily be expanded to other areas, e.g. for implementing DRM schemes. Only with a matching signature (created upon selling a specific media file), including a unique number of the hardware it is intended to be used upon, it will actually be played. The advantage is that no online verification is needed, increasing privacy. The drawback obviously is, that no transfer to other devices (or only those approved by the seller) is possible anymore. In this way openness can be reduced almost to non-existence: While providing the source code (and the possibility to recreate the binary and therefore verify it is actually this code running on the TiVo) for inspection (learning, security etc.) still has some advantage for end users, none such exists in the media example: The media file might be shared with whoever you want, but nobody else could play it. This closely resembles the legal protection of protection measures, which can be used to practically abolish all free uses.

2.3. Open access journals

Open access journals (see [12] for a directory) are scientific publications which are placed online and can be accessed without payment (sometimes the author is required to pay to cover costs). One reason is, that subscriptions for similar printed journals have become so expensive, that many universities can no longer afford them. Additionally, as much of the research published in them has been funded by public money (via universities, science foundations etc), the results should also be available to the public for free. The basic premise is, that publishing on the Internet is very cheap as opposed to requiring printing machines, postage etc. to produce and distribute paper journals. So this kind of openness wasn't really possible before the Internet. But some boundaries for open access journals exist:

- If there is no income, persons responsible for it must donate their work (see 4.2). Such persons need to be found and kept for continuity. With good software support the task can be automated heavily and the role of editor is already often performed for free even for conventional journals.
- Reviewers cannot be paid for their effort. But in the scientific community this is extremely rare in any case, so no difference exists.
- Trust is harder to achieve: If a reputable publishing house starts a new journal, good quality can be assumed. But for open access journals no such trivial assessment is possible (compare section 4.4).

2.4. Security by obscurity

In general, security by obscurity is not a good idea and explicitly repudiated. This applies especially to encryption algorithms, but for all other security measures as well. However, there are situations where this is the only possibility, e.g. given a situation where potential attackers have administrator rights on the computers the software must run on by definition (already in requirements). Then the only possibility to secure this system is trying to keep as many details hidden for as long as possible and rendering access to them as expensive as can be made.

Obscurity is not seen as a valid security principle, because at some point in time the details of software programs will become known (absent hardware security, like tamper-proof modules). This is inevitable, as the program must be executed. If it can be executed, this can also be simulated and studied. Then any weaknesses that may be discovered can be exploited. Really secure are therefore only systems, where everything – or at least the algorithm or source code, but note: not the data, i.e. the keys! – is openly available, for example encryption algorithms or security procedures (like Kerberos).

The limit to openness is here introduced artificially: We know we can't keep it secret, but we try to hide it for as long as possible to achieve at least some degree of security. This limitation is only the second-best solution, but it is all that is available.

3. Legal boundaries

Boundaries may also be based on legal regulations. Sometimes these are only perceived, but nevertheless actual barriers do exist.

3.1. ASP2PHP

This is a decision from the Austria supreme civil court (OGH 16.1.2007, 4 Ob 198/06f) regarding the copyright of software programs. The basic scenario was software for a web shop written in ASP. This software was developed specifically for a single customer, but who did not receive the source code. Later he employed another company to produce a new and equivalent version in PHP. They had no access to the source code, only to the web output. Parts of the HTML code generated was reused in the new program. It was decided by the court that no copyright violation took place, as the program was not accessible (→ no source code copying) and what was taken (some trivial JavaScript) was not eligible for copyright protection. However, reusing the effort of the first company (some lines of HTML code for tables) was deemed an act of unfair competition. This decision is strongly debated, as almost a new intellectual property right was invented: Even though the program is not protected by copyright, you cannot reuse any part of it. You have to expend the same effort again, although you (or your employer) had already paid for it previously.

This is very problematic for open source, as it severely reduces possibilities for building up on other's work. This would mean that even if the source code had been available and any copyright waived, you could still have not used it to produce a competing product. If this decision continues to be valid, openness would be severely restricted: Additional clauses would be required to remove the danger of such proceedings

3.2. Liability for problems with open products

Open source severely limits the liability for problems of it, like bugs. One of the reason is, that it is usually free, the other that the actual author who introduced a specific problem can be very difficult to identify. This non-existence of any guarantees, see e.g. the BSD license [3], theoretically limits the usefulness of software severely. For instance in high-security or –reliability installations (like defence or medical systems) open source could therefore not be used. However, this boundary is comparatively weak: Closed source software comes only rarely with any kind of guarantee; see for instance the limitation of liability for Microsoft Windows [14], paragraphs 15-18. Conversely, if you buy a system including open source software, some kind of guarantee for fitness of purpose will always exist, just not by the authors of the software but rather the system integrator.

Also a similar problem exists in reverse: In many countries not all liability can be disclaimed. This can be a problem for open projects, who may find themselves under legal threats without the possibility of some kind of insurance. Commercial companies can take out an insurance and limit liability to the company, but for individual open source developers this is much more difficult.

So legally this is an actual limitation of open systems: It is difficult to identify who might be liable, and any liability is usually excluded as far as allowed. Practically there exists little difference to non-open software.

3.3. Open source licenses

Open source licenses (e.g. GPL or Creative Commons) are used often and appear very open. Seen from the legal point of view they are however rather closed and contain very hard limits. For instance it was decided that the GPL is explicitly no waiver of copyright [10]. A really "open" license is the BSD license, which roughly grants you the rights to do with the work whatever you want and is almost equivalent to placing the work in the public domain.

These boundaries can be seen best by the example of the LGPL: It allows using libraries in non-GPL software, as long as modifications to the library remain under GPL, i.e. are published. But the main software may remain closed. The Free Software Foundation, who developed it, itself actively tries to discourage [9] developers from using it, as it is too open for them! From this example it becomes apparent, that from a legal point of view open source licenses are very closed, as they set hard limits what you can and cannot do with the software, enforcing a certain philsosophy. Creative commons is a bit more open in this aspect, as it lets you choose several options, but at the core the same boundaries exist. Both just define a different set of boundaries as compared to proprietary software.

3.4. The Kindle controversy: Transferring rights

The Kindle is an eBook reader by Amazon based on open source software. Note that the Kindle itself hardly qualifies as “open”, as it depends heavily on DRM to secure the digital content. One of its features is the ability to read aloud the book (or newspaper, article etc.) through speech synthesis. Authors protested against this feature as Amazon only possesses the rights for written books, but not for audio books. The background is, that audio rights are sold separately and are usually much more expensive than eBook rights. The authors requested that this function be removed or the appropriate rights were obtained, reducing the functionality.

Legally seen this is a dubious claim because audio rights are only required for public performances, i.e. if a kindle was put on a stage and used to “read” a book to an audience (whether paying or for free would not matter). Private reading aloud, for instance to your children, is covered by fair use. A counterargument is, that the quality of the read-aloud function might not yet be useful for a public reading, but in the near future probably will be, rendering the scenario above realistic.

Related to this dispute is the concept called “Zweckübertragungstheorie” (Theory of the purpose of transfer). This theory about intellectual property rights (accepted at least in in Austria, Germany and Switzerland) states that in a contract on IPR only those rights are transferred, which are necessary for the envisaged use (absent any explicit other language). In Germany it was legally impossible (§ 31 para 4 copyright law) between 1966 and 2008 [20] to transfer the rights of uses yet unknown. This means, no rights for DVDs (contract before 1979) or the Internet (before 1994) could be transferred and therefore remained always with the authors, even when "all rights" were transferred. The reading feature of the Kindle could be seen as such a new use previously unknown. With the second part of the copyright amendment Germany changed this retroactively, so authors had the possibility to decide until 1.1.2009 whether to object to this or remain silent: Then the owner of the main rights would receive them for free. In Austria no such explicit provision exists, but the purpose theory is executed even more strictly than in Germany and usually a similar exception is assumed [17]. This leads to an uncertainty for open access content (less for open source software: new purposes for program code are unlikely!): Is it possible to “open” old content, for which an exclusive print license has been granted? And can new content be opened up so much, that currently unknown uses in the future will be included? For the Creative Commons licenses in the Austrian version ([5] at the end of section 3) such a transfer was included in doubt [16], but whether it would hold cannot be said yet.

3.5. Software patents

One reason for graphic card drivers for Linux for advanced 3D graphic cards to not be available as open source but only in binary form is the software patent problem: In this area numerous patents exist, which it is very hard to circumvent. Here open source software is severely at a disadvantage: Everyone can access the source code and therefore quite easily check, whether a specific patent is violated, for initiating legal actions. Closed software would have to be decompiled (but see obfuscation) to allow such identification. Therefore neither AMD nor nVidia publish their driver source code to reduce the danger of patent infringement proceedings.

Open source software is disadvantaged in this context in another aspect: Software patents are quite expensive, so they are no option for such developers, only for large companies acting as distributors. So they have no own patents, but are more easily pressured through those of others. This is one reason for the intense dislike of software patents by open source proponents. Only one limitation exists in the opposite direction: As open source software is freely published, it is immediately state of the art and therefore prevents any further software patents. Opposed to this closed source software is only a publication regarding the visible aspects, e.g. the process performed by it, but not its internal implementation. Therefore open source is more successful in preventing patents [6].

This difficulty is increased by the lack of special provisions for software: Copyright was modified because of the special properties of software to allow e.g. decompilation or backup copies under certain circumstances. Currently no such provisions exist at all for patents and it is not likely that they will be included (although some were proposed for the – finally failed - EU software patent directive; see article 6 on interoperability in the form amended by the parliament [7]).

3.6. (Registered) Marks

A registered mark actually must be defended or it is lost. This means, that even if it was desirable to allow some people to use it, this cannot be done. As marks are an extremely valuable part of intellectual property – they potentially last for eternity - this is a real problem. The only option available is an actual license of the mark (which might be free). This must however still be tied to the product or service the mark is registered for. If the license allows using it for other things, e.g. software from someone else performing a different action, this is equivalent to not defending it.

An example could be a service provided by some software. It might be patented or not, but it is generally known by a registered mark. If open source software is written to be a functional equivalent, it can never use the mark to describe it: This would be a violation of the mark and would have to be prosecuted. And giving a license to everyone using a certain software (as e.g. the GPL would require by analogy) would be the equivalent of loosing it: The mark would become a general descriptive term and it protection would be lost. This is a rare example where openness is legally impossible.

4. Economic boundaries

Economic boundaries are not as hard as the ones discussed above, but nevertheless very important: If it doesn't make sense to open up a project, it will remain closed. Altruism is a powerful motivation, but earning a living is as well. Although "openness" is often equated with free, the FSF [8] puts it differently: "free as in free speech, not as in free beer". This means, people might do something without compensation, but usually there is the need for remuneration in some other way. This ties in with a change from products to services: The economic background remains, only the way to achieve income shifts, like from writing and selling software to adapting, configuring and maintaining an otherwise free (often as in free beer!) software.

4.1. From product to service

Products can often be mass-produced cheaply, especially in low-wage countries (compare the problems with product piracy: If something sells, a cheap copy will be manufactured in the east very soon), and in the IT area copied, but outsourcing services like local installation or configuration is much more difficult. Similarly, a music recording (=product) can be downloaded from the Internet, but a live concert must be attended personally. This can also be compared to the transition of software from a shrink-wrap product you buy to a service you subscribe to: online activation limited in count, Software as a services (SaaS), Service oriented architecture (SoA), or subscription schemes (online role-playing games). Similar approaches are mobile phone manufacturers providing additional services, as the income from the connection itself declines continually.

One reason for this is that services are unique: They have to be performed each time again and cannot be copied as easily as products (so music subscription schemes rely on DRM to ensure this uniqueness and remain a service and not a product model!). They remove the limits of uniqueness of things, but replace them with the uniqueness of the action performed with/on it. Note that this is only correct in the current economic situation: If access to the European labour market would be opened up to everyone in the world, many services would be copied very fast and very cheaply. This danger is especially large in the IT area, where e.g. system administration could be (and al-

ready often is) outsourced to countries far away. Therefore the “openness” is actually quite limited: Who is allowed to provide the service is at least in some areas severely limited. This can also be seen with the increase in Linux certifications (see [1] for an overview of Linux certifications). Such certificates often serve as artificial restrictions for services: Unless someone holds the “XYZ” certification, he is not eligible for providing specific services (or just not hired). They impose artificial limitations, so the competition in the services area is reduced. Overall, limited physical objects are replaced by (at least at the moment; consider expert systems) limited services.

4.2. Generating revenue and the importance of large numbers

An important aspect of open projects is generating revenue. If not the whole project is supported by external gifts, some kind of income must be provided. The dominant options in this are donations and selling fan articles or advertisements. These are very tenuous and unpredictable sources of income so most open projects are constantly in dire need of money. Therefore they tend to be operationally cheap, i.e. requiring as little actual money as possible. Instead they try to attract persons donating some effort. For open source software: Donating money in the disguise of providing a patch developed in two hours of time (each worth € 50) is much more likely and attractive than giving € 100 in cash. Not going into details, one of the reason for this is taxation. You have to work many more hours than mere two to earn € 100 in cash. Additionally, when spending money it is gone, but providing a digital creative work leaves you still in its possession, albeit perhaps at a lower value for selling it. But e.g. for patches or small programs no real market exists anyway or building a business on them would be quite cumbersome: So you give away only what has no "value" to you anyway, but might be helpful for others.

But this dependence on small donations of work by numerous people shows an important boundary of openness: Unless there actually is such a large number of donators, it does not work. For example, if the Linux kernel were still developed by Linus Torvalds and perhaps three or four others alone, it would never be comparable in functionality to Windows and be completely irrelevant as an operating system. As any significant piece of software requires a lot of effort in total, large numbers of participants are required, which must be recruited, inspired, retained and replaced. As soon as this process fails, the open development will fail. While closed projects depend on direct monetary funding, open projects require large numbers of volunteers. With the number of open projects increasing, this will become very difficult. For instance starting a new open encyclopaedia competing to Wikipedia will be very difficult, as several project have already had to notice: The number of people working for free and their effort are limited, and winning them over from a competing project is very difficult. One consequence of this is that very often the first project wins, and if a large size has been reached, it is very difficult to bring down. In this way open projects are even more vulnerable to build monopolies than closed ones. Obtaining even a very large amount of money for a competing closed project is possible, e.g. with venture capital, and ways for assessing the chances of success exist. But acquiring and predicting the final and stable number of volunteers for a new open project competing with an already existing one is much more difficult. As can be seen from the example of the X-server ([2], see 1.1. The Politics) regarding the split between XFree86 and X.org), rather a divergence occurs: The code is split and two competing branches may start. In the end this is not a new development, but rather a discussion about the future direction. Real new developments from scratch become therefore very difficult in open projects, a significant limitation.

Another aspect more pertaining to open content than open source is the possibility of alternative sources of income. For instance musicians might give away their music for free and still make a living from live concerts. This, however, does not apply to all kinds of digital works: Producers of a

film have much more difficulties (cinemas are an option, but increased screen size of TVs, beamers etc. in private homes become a competition). Authors are in for real problems: Giving lectures from a book and book (printout) signings are not a real option to selling the written work itself, whether in physical or digital form. Additionally, these works are not suited to being generated by a large number of contributors. Novels might provide a good income for a writer, but it is unrealistic to expect an interesting novel or a song being written by dozens or hundreds of persons, each supplying a single paragraph, line, or melody through collaboration.

4.3. Application areas for open source software

Not every kind of software is suited for open source: For instance a software useful only for a single customer will almost never be available as open source: If it is open source and exactly matches the needs why should the customer pay for it in any way? And if it is usable for a single person only, why should someone develop it? No fame or money can be earned from it! This ties in with the pervious section: Unless a large number of contributors exist, the result is often not very useful.

Therefore open source software is usually limited to rather general software. It is not a coincidence that the "biggest" and best known open source projects are operating system (Linux) and servers (Apache webserver, Samba fileserver). However, many individual projects are built up on open source libraries or parts. This means, the general building blocks are usable for many applications (→ open source viable), but the integration and customization is "proprietary", i.e. developed for the individual customer. Because of legal requirements it might be open source too, but often not useful for others, unless a new or better building block can be extracted from it. Therefore this boundary is actually an advantage, as it ensures paid work for developers.

Regarding open content the economic balance might be directly opposite: Providing small elements (e.g. graphics) usable only for rare cases might serve as an advertisement to buy the mass-needed ones or be contracted for producing special others ones. But in contrast to software there is little need for constant service or updates of works. Only new developments (pictures, texts, movies etc) are needed and reuse of parts is much more difficult. Therefore an economic limitation exists for open content as no equivalent model currently exists. This seems to be at odds with e.g. open access journals. But it should be noted that there an economic incentive doesn't actually exist: The producers have already been paid for by public money and don't expect additional income, so the advantage should be available to the public as well. But with works paid for by private companies this does not apply: They were paid for by a single entity, and – as opposed to scholarly articles – are very often not very useful for the public anyway, only potentially for competitors.

4.4. Role of the guarantor

Somebody should guarantee for the validity and correctness of data, like the content of a scientific book or an encyclopaedia. This can either be the author (very well known expert in this area) or the publisher (reputable publishing company which ensures high quality through excellent staff and external reviews). However, if everything becomes open, there is no equivalent guarantor. Take wikipedia as an example: Everyone can add or modify content anonymously or under pseudonym. As no official review process exists, the content may be (deliberately) incorrect. Additionally a single excellent article from one person is no guarantee at all that a second article by the same person is equally reliable: The author might be knowledgeable on the first topic but not the second or the second article was defaced/modified by someone else.

Openness therefore brings often with it anonymity: This might be desirable (privacy), but removes the guarantor from the picture. Some information may be "good" or not, but it can become almost impossible to assess it. Compare this e.g. with the Linux kernel: Everyone can submit patches so it is extremely unreliable. But openness is severely restricted there, as a very small core team must approve all patches going into the kernel (and many are rejected, even if coming from well-known experts). So there the openness is reduced to re-introduce those guarantors. Inclusion is rather based on merit than on source. Similar approaches have started in Wikipedia: Some articles might be locked to prevent modifications and final deletion is only possible by administrators.

5. Conclusions

Openness is an interesting new concept, and usually based on the Internet. However, it is often misunderstood as being gratis and without restrictions or limitations. This is not true. Openness is just a different approach as opposed to the previously predominant closed one, but it has boundaries too: they are just different ones. This can best be seen in the economical limits. People still need to make a living, so income must be generated in some way – it may just be a different one than before. Overall seen it is still a decline of limitations or moving out of boundaries: These shift in open projects, but typically to the outside and only in very few and small instances new limitations are built or existing ones strengthened.

In some areas real new options arose, which were not possible before. The best example for this are open encyclopaedias. Without the Internet such a thing would have been impossible; now it is the predominant source of such knowledge. With all its shortcomings and limitations it is still a fascinating new approach: Creation and explicit representation of a large body of knowledge through a collective effort. This is the current endpoint of a change which has started several hundred years ago. Then a person could expect to make significant progress in a chosen area or discipline in their lifetime alone, but today this is much rarer, as most new developments will require numerous experts cooperating to offset the requirement of long education to reach the peak of the current knowledge even in a small area alone. Therefore cooperation becomes more important and reducing the boundaries limiting it a necessity. These limits are therefore (or will have to be) shifted to still allow the society to remain workable and stable, but reduce restrictions on and encourage collaboration.

The most important problems with open approaches are trust and economic viability, while the legal boundaries often seem to be lagging behind. Especially in some areas current stakeholders seem to be successful in even hampering this change in their own interests (see e.g. the extension of the protection for music production, which almost exclusively benefits the big media companies and extends existing boundaries into the future). However, opening up will be a necessity in the future. In which form or which direction and who profits/looses by it will yet have to be decided or become apparent from large-scale practice.

6. References

[1] Anventure: Linux Certification Questions & Answers, http://gocertify.com/faq/linuxfaq.shtml
[2] BROWNE, C.: The X Window System, http://linuxfinances.info/info/x.html
[3] BSD Newsletter.com: What is the BSD license? http://www.bsdnewsletter.com/bsd/license.html
[4] COHEN, N.: Don't Like Palin's Wikipedia Story? Change It http://www.nytimes.com/2008/09/01/technology/01link.html?_r=1

[5] Creative Commons: Namensnennung - Keine kommerzielle Nutzung 3.0 Österreich, http://creativecommons.org/licenses/by-nc/3.0/at/legalcode
[6] EFF: Patent Busting Project, Wanted: Ideaflood/Hoshiko, http://w2.eff.org/patent/wanted/patent.php?p=ideaflood
[7] FFII: Europarl 2003-09-24: Amended Software Patent Directive, http://eupat.ffii.org/papers/europarl0309/index.en.html
[8] Free Software Foundation: The Free Software Definition, http://www.gnu.org/philosophy/free-sw.html
[9] Free Software Foundation: Why you shouldn't use the Lesser GPL for your next library, http://www.gnu.org/licenses/why-not-lgpl.html
[10] LG München I 19.05.2004, 21 O 6123/04, http://www.telemedicus.info/urteile/496-21-O-612304.html
[11] Linux Kernel Organization: The Linux Kernel Archives, http://www.kernel.org/
[12] Lund University Libraries: Directory of Open Access Journals, http://www.doaj.org/
[13] Microsoft: Important Notice: MSN Encarta to be Discontinued http://encarta.msn.com/guide_page_FAQ/FAQ.html
[14] Microsoft: Windows XP Home Edition (Retail) EULA: http://www.microsoft.com/windowsxp/eula/home.mspx
[15] OLPC Bitfrost: http://wiki.laptop.org/go/OLPC_Bitfrost#P_BIOS_CORE:_core_BIOS_protection
[16] PHILAPITSCH, F.: Comments on the draft for the Austrian BY-NC-SA 3.0 license, 1.2.2008, http://creativecommons.at/draft/draft-by-nc-sa-3.0-a.pdf
[17] THIELE, C.: Verwendung von Mitarbeiterfotos auf Firmenwebsites. Wbl 2002, 397
[18] TiVo: http://www.tivo.com/
[19] TORVALDS; L.: Flame Linus to a crisp! http://groups.google.com/group/fa.linux.kernel/browse_frm/thread/2f93510ff38d9089/
[20] WACHSMUTH, R.: Urheberrecht - Übergangsregelung für neue Nutzungsarten - Widerspruch nach § 137 l UrhG sichert Rechte, http://www.proft-wachsmuth.de/Urheberrecht_Uebergangsregelung_fuer_neue_Nutzungsarten_Widerspruch_nach_137_l_UrhG.html
[21] Wikipedia: Tivoization, http://en.wikipedia.org/wiki/Tivoization