Three walls surrounding the utopia of opensource

Tadayoshi Ohkuma
President, Omoikane, Inc

* First half of 2003
* Trend Case - SCO
* Trend case - Kochi Font
* Manifesto
* Utopia of the opensource
* Possible supporter of the opensource
* Disfunctionality of local incentive
* Infringement with standardization
* Opensource as the implementation of standard technologies
* A proposed amendment
* Conclusion

----

* First half of 2003

To summarize the opensource world in the first half of 2003, along with constant increase of users and shares in server and embedded field, several opensource enterprises have failed and the limitation of application of opensource became clearer.

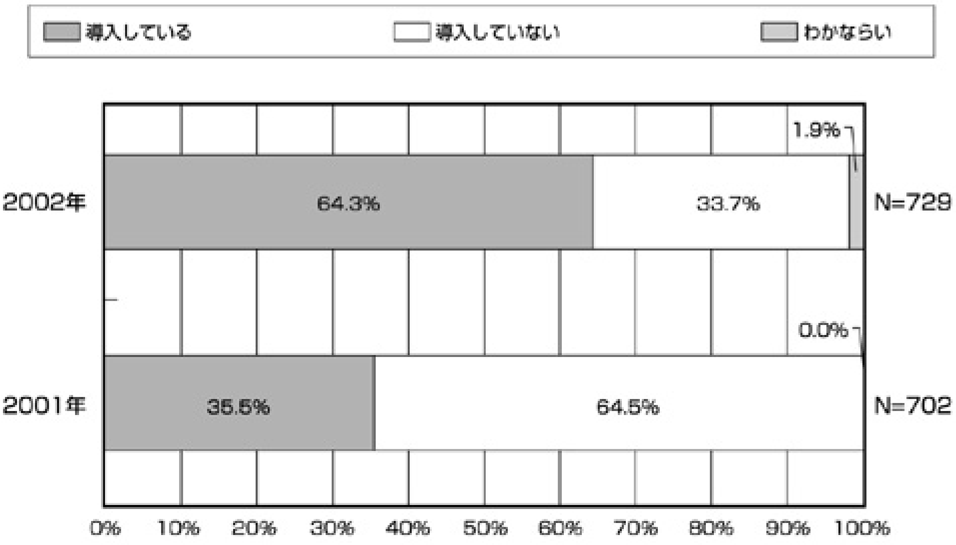
Share of Server
  an enquiry about usage of Linux server (2001-2002) [1-1-1]

  IDC Japan press release 'Domestic server share trend: OS sales share 3Q 2002'
  http://www.idcjapan.co.jp/Press/Current/20030317Apr.html
    Windows 77.7%
    Unix 12.8%
    Linux 7.3%
    other 2.2%

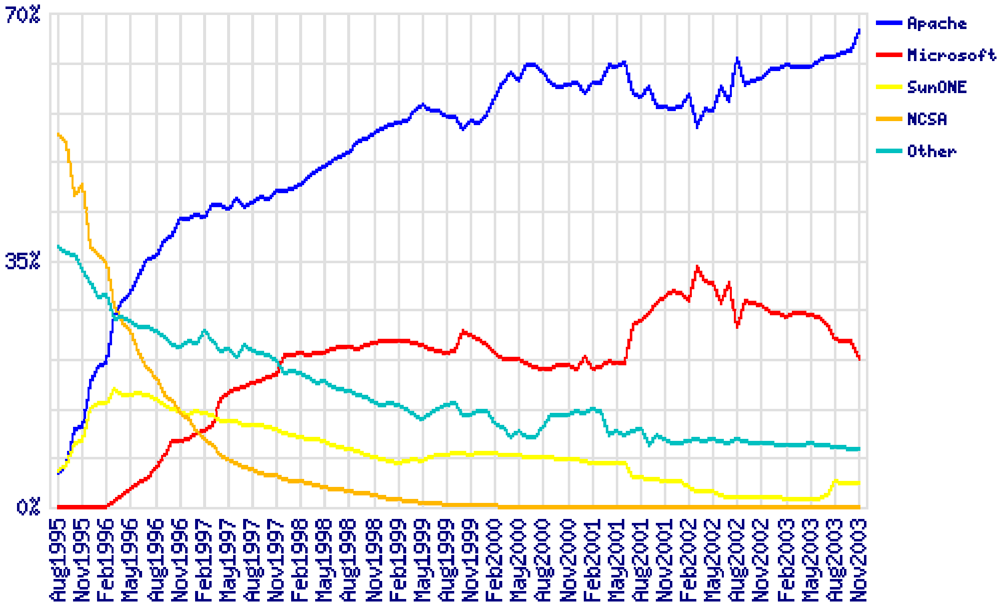
  Direct internet response research 'Netcraft Web Server Survey'
  http://news.netcraft.com/archives/web_server_survey.html
    over 40,000,000 sites
    Apache 64.5%
    IIS 23.5%

other 12%

資料1-1-1　Linuxサーバーの導入有無(2001-2002)



Linux白書2003©インプレス.Access Media International.2002



As well as pooling, a part of opensource is being consumed. This phenomenon is well observed in device driver area. Mainly because of exhaustion of fund created at net-bubble age, driver quality of Linux/XFree86 seems to become gradually poorer. For example, slow development of WLAN driver of 11b/g is often pointed. It is estimated that harewares which produced by pro-opensource company will be selected, like before Net Bubble.

* Trend Case - SCO

Although SCO case was reported a bit sensationally in Nikkei Electronics 2003/10-13, according to an enquiry to 440 IT sections of Japanese enterprise in the same article, the shock of the case seem to be limited.

- Positive	27.3%
- No effect	23.2%
- Careful	9.8%
- Don't know well	39.8%

Most readers should know about the case in detail, so I would like to explain the possible worst impact to Japanese domestic enterprises.
(Note: the estimation is based on the information available by middle of October)

SCO doesn't state the exact part of infringement. Known infringement will falls into three part of codes below:

(a) UNIX prior to SVR
(b) 4.4BSDLite1
(c) Contribution from IBM/SGI etc. eg SMP/XFS/JFS...

Code (a) impacts on every Linux user except on research and educational purpose. Known part of code are thought to be from UNIX 32V and are shared by a famous published UNIX book called Lion (Lions' Commentary on UNIX). Whether this part does infringe or not, seems to depend on either citation style, technical necessity or amount of citation.

Code (b) is the pile of code resulted from the famous law suit between BSDI and USL, and thought to be cleaned up and free to anyone under BSD license. SCO, however, stated that it is not permitted to divert into Linux.
This SCO's stance may implies that their IP does not only cover Linux but also other BSDs, which are not direct descendants from 4.4BSD. NetBSD,

OpenBSD and rarely FreeBSD come under it.

Code (c) is caused by the source code license agreement of UNIX system V. In that agreement, it was stated that derivative work from UNIX would belong to licenser, and SCO claimed that works like SMP/XFS/JFS from IBM/SGI are considered as UNIX derivative, then in turn owned by SCO.

(a)/(b)/(c) are the whole disputed part of infringement. Other one important point that relates to these is that SCO itself distributed Linux under GPL under the name of Caldera. SCO insists that GPL is not valid because the license does not reflect the will of the company.

From these classifications, impacts on enterprise will fall into two pattern, depending on whether the company has UNIX source codes and contributed to Linux.

Firstly, speaking about (c), those who are licensed UNIX source code and have a contribution to Linux, has actual possibility of losing in suit. A company having USA branch has higher possibility. This case looks like an ordinary dispute between companies rather than an opensource trouble.

Secondly for the other company and person, if Linux infringed, the legal safety basically depends on the validity of GPL. If GPL was valid as a contract, one can continue to use and distribute softwares which are permitted at the time of contract. According to this scenario, softwares prior to version 2.4.14 were distributed by SCO and seen safe. But even in that case, for softwares after version 2.4.14, Linux still has a subtle problem.

As many people criticize, it is true that the license management of Linux was somewhat loose. It might be a characteristic of Bazaar development. It is not worth to surprise if Linux contained a small amount of dead copy from a commercial UNIX.

Otherwise, it is possible that GPL will be judged invalid. In the opensource license group of SOFTiC/IPA, the validity of GPL as a contract was sometime doubted too. If GPL was invalid? According to the Japanese copyright law of article 113, an infringement only occurs when one knows that it is a piracy. Thus the impact to the most domestic users should be very limited.

Even when the infringement turns out to be true, Linux will soon be SCO-clean before the suit finishes, because SCO has to prove the extent during the suit. After all, those who actually suffers are only the group of enterprises first suited and it will not damage most of Japanese users.

The CEO of SCO, Mr. McBride said that SCO's goal was to make Linux managed a consortium like MPEG LA. It looks unreal.

As a single person among the computing society, I honestly feel like that the suit should be just judged an overuse of right. The real impact, however, seems to be the first time judgment of the validity of GPL.

It is really possible that GPL will be judged invalid because the rule of current free software usage is not so clear. Although this judgment may change the future version of GPL better, it may also cause another confusion among opensource community.

* Trend case - Kochi Font

Another intellectual property right trouble of opensource is about Kochi Font. This is a curious case because it only concern Japanese opensource community and intellectual property rights related to font are so subtle.

Kochi was created voluntarily Mr. Furukawa. It is a representative Japanese outline font and was freely distributed. Kochi is widely used and appreciated because the number of Japanese character is too many and font is seen not suitable for voluntary work.

Suddenly early this year, someone noticed and reported that a widely distributed free 32-dot bitmap font is exactly same as sample glyph shown in the several published books, including 'Shotai wo Tukuru'. As his research proceeded, it turned out that a dead copy occurred before 1989 and the distribution began via commercial BBS.

In the process of making of Kochi, this 32-dot bitmap font was a seed. Multiple people converted it and Mr. Furukawa recreate elements of typeface, balanced the typeface and created Kochi, though Kochi and original 32-dot bitmap share the balance and characteristics.

Hitachi Printing Solutions (abbreviated as Hitachi PS) and Typebank are current owner of original 32-dot bitmap font. Being informed as above, after a while, Hitachi PS announced briefly as below.

(1) About right of 'Hitachi TB 32-dot Mincho font'

The font was developed under cooperation between Hitachi and TB and all rights are reserved. No one can develop, publish and distribute fonts based on the right of 'Hitachi TB 32-dot font Mincho' without permission from both company.

(2) Usage on Linux

Although the font is already used without permission, considering and contributing to the promotion of Linux, we have decided to permit limited use of the font.

According to (2), Hitachi PS only permits non-commercial usage of the font and hence the 32-bit font is not opensource. As this announcement only refers to the dead copy and not to Kochi exactly, we asked Hitachi PS if Kochi infringes Hitachi's right or not. Hitachi PS responded 'yes it does' as below:

We have 'design rights' which are intellectual property rights being evaluated as well as software in the sense that it is a work which integrates every design over 7000 characters based on the unique philosophy and sense. Currently no law mentions about that, however, we are certain that we have some intellectual property right.

Hearing the stance of Hitachi PS, Mr. Furukawa, the author of Kochi, stopped both distributing and developing. Above is a summary of the trouble of Kochi font.

Japanese are discreet to protect typeface and establish no right with regard to 'Copyright Law' or 'Design Law' but only indirect rights with regard to 'Unfair Competition Prevention Law'.

For Kochi font, it can't be an unfair competition because the development was 10 years later from first dead copy and the quality is clearly improved from bitmap to outline.

Although Kochi doesn't cause infringement, the author said 'If there had not been a matrix which is usable enough, I could not developed Kochi. I also understand the effort of the original font of Typebank and Hitachi.' He announced the quit of development because of the claim from Hitachi PS.

A proposed protection of typeface can be seen in a web site of the Japan

Typography Association, under title of 'Typeface ni kansuru rinri kouryou (ethical principles on typeface)' and 'Nozomasii typeface houteki hogo no arikata (ideal legal protection on typeface)'. Those who requests legal protection of font doesn't seems to wish an inflexible protection, because they would have another trouble to create a derivative typeface with inappropriate protection.

Hitachi PS says 'Our work has value in the sense that designs are integrated basing on unique philosophy and sense'. But during millions of copies were distributed over 10 years and the typeface were shown in several magazines, why they couldn't find that their 'unique philosophy and sense' had been copied? Of course we understand that they invested a lot on the original font, but is such a tiny originality worth to be protected publicly?

Although both Hitachi PS and the author of Kochi act along specific industrial convention, I have to say that Hitachi PS insists too much and that the opensource pooling is very fragile.

* Trend case - Linux distribution

In the first half of 2003, two movements are observed in distribution area. One is the policy shift in development of Red Hat and the other is weak recurrence to desktop. Needless to say, Red Hat is the largest company among commercial Linux distributions and, regardless of profitability, it has a lot of power stocked via IPO.

The business of Red Hat can be characterized with strong commitment to the opensource and with internal development of distribution. For development of distribution, free rides are very easy and other relative distributions like TurboLinux owe computing resource to Red hat. (Reverse free rides can also be seen but few.)

But recently Red Hat quitted both the role of central resource of commercial distribution and the sales of package software, by leaving development of distribution to totally unknown project named Fedora Linux. Red Hat concentrated its resource to server area.

By now and in future Red Hat creates a server version of product from a matured repository of desktop version. In short, design of future version of Red Hat Linux will be determined by the external project.
This policy shift can be seen as half lose of identity of Red Hat.

From the view point of business model, this shift prove that single company can't afford to develop a distribution in any way.

Second movement is observed by several release of desktop products, including Japanese Lindows, TurboLinux Desktop and Java Desktop System from Sun Microsystems. This weak recurrence to desktop is possibly caused by reliability increase of OpenOffice and by the decline of Red Hat.

Although such a small boom of desktop was seen, strong movement of desktop will occurs only after a lot of release of commercial applications.

* Manifesto

In the November election in Japan one key word became suddenly popular - Manifesto - which is, however, already well known to old inhabitants of opensource almost over 20 years, as GNU Manifesto.

Whereas core motivation for creating opensource software is self use, some people like free software movement intentionally aims for a social revolution. 1985, at the time of founding of FSF which is the basis of free software movement, GNU declared its philosophy in the form of GNU Manifesto. Although GNU Manifesto was modified several times, it is still on GNU's web site without changing original shape.

Hearing Manifesto may remind some people of 'Manifesto of the communist' by Marx/Engels. But the author of GNU Manifesto, Richard Stallman, seems to dislike to be classified into communist. For people who know left wing economics, however GNU Manifesto should smell a communism. Here is two example paragraphs from GNU Manifesto which contains a similar point found in Marx.

For reduction of the necessary labour

GNU Manifesto:
'In the long run, making programs free is a step toward the post-scarcity world, where nobody will have to work very hard just to make a living. People will be free to devote themselves to activities that are fun, such as programming, after spending the necessary ten hours a week on required tasks such as legislation, family counseling, robot repair and asteroid prospecting. There will be no need to be able to make a living from programming.'

Marx from 'The Grundrisse' Notebook VII 1857:
'The free development of individualities, and hence not the reduction of necessary labour time so as to posit surplus labour, but rather the general reduction of the necessary labour of society to a minimum, which then corresponds to the artistic, scientific etc. development of the individuals in the time set free, and with the means created, for all of them.'

For estranged labour

GNU Manifesto:
‘...; marketing arrangements now typically used essentially forbid programmers to treat others as friends. The purchaser of software must choose between friendship and obeying the law. Naturally, many decide that friendship is more important. But those who believe in law often do not feel at ease with either choice. They become cynical and think that programming is just a way of making money.'

Marx from 'Economic and Philosophical Manuscripts' 1844:
'Thus, through estranged, alienated labour, the worker creates the relationship of another man, who is alien to labour and stands outside it, to that labour. The relation of the worker to labour creates the relation of the capitalist - or whatever other word one chooses for the master of labour - to that labour. Private property is therefore the product, result, and necessary consequence of alienated labour, of the external relation of the worker to nature and to himself.'

Besides GNU Manifesto, we also find communism in copyleft, which does not allow a one's freedom to keep source codes secret. It is a kind of ownership of information. It also clearly forbids to cipher source codes.

If we compared with the real gory struggle, tragedies and dignity would disappear, and born in individualism, GNU would looked like a comedy of the software industry. Richard Stallman, However, like Marx who grieved for own declining country which was governed by capitalism, also grieves and acts for the hackerdom which is declining with growing of the software industry.

Concluding from these characteristics, we may regard GNU as a software communist(or Marxist) movement. But that movement does neither look like the communism under the former eastern Europe nor request a violent revolution but shows a kind of reality as we already observed in the opensource movement.

People in the software industry needs to understand that there is an inevitable software communism behind the opensource, half of which consists of GNU. Following this section, we try to examine what barrier surrounds this Marx-like utopia of opensource.

One notice:
People who promotes the opensource has wide range of directions and differs in the ideal social model. The opensource is only defined by license relation and not apt to argue the social model. It is not true that most of the opensource people is from the free software movement, but, being lead by Richard Stallman, it makes the biggest effort to explain the social model. Therefore, for following arguments, no matter how left wing, we regard the free software movement as the representative of the opensource movement.

* Utopia of the opensource

Besides software communism, the goal of the opensource is to intentionally establish a layer of public software under current intellectual property system. 'intentionally' means not passively waiting for termination of copyright but trying to pool into the public resource from the first time of release.

Mixing private and public software guarantees a freedom of system software for programmer. Further more, if public software was dominant, intellectual property system would lose its function.

This is the certain aspect of the utopia of opensource from the point of view of programmer. Although this is not realistic now at 2003, in order to make the point obvious, we assume that public resource will be dominant.

In that society, the problem would be 'How and who can support the cost of software development'. Into detail, this problem splits into two parts, 'Who' and 'How'.

1. Who --- possible supporter
2. How --- disfunctionality of local incentive

In every area where once excellent opensource appears, license based business collapses and is forced to convert from high efficiency investment to quality of service. Area by area this brings us reduction of industry. It is historically inevitable. If we intend to avoid it, we would have to develop new frontier with faster speed of innovation.

But if a planned economic system was invented and the cost of software development was supported, pure software enterprise could continue and expand publicly or privately and the reduction of industry wouldn't occur.

Even when the 'possible supporter' was determined, depending on the economic system, if local incentive was gone, diversity of software would be lost. This is 'disfunctionality of local incentive'.

* Possible supporter of the opensource

As argued above, without license model, the problem would be 'How and who can support the cost of software development'. Considering possible supporter, we can introduce a classification like below.

1. volunteer
2. company/investor
3. association
4. local autonomy/government

Model #1 is typical for primitive opensource. Programmers have sacrificed their time and efforts. Primitive opensource had been pooled slowly long time this way.

Model #2 was seen in a lot of entrepreneurs' attempt ion during Net Bubble, and most of these ended in miserable failure. Precise explanation is omitted because it was already mentioned in the report of IPA's 'Opensource software no genjou to kongo no kadai ni tuite (Trend and problem of contemporary opensource)'.

Several business models were introduced 1999 in the paper 'The Magic Cauldron' by Eric Raymond, whose core 'technology' is opensource, but most of these obviously failed. With the bitter experience, we have to conclude that companies and investors can not be supporters of opensource, even in the short run. Companies and investors can only support via peripheral business. It should be said a type of corporate sponsorship for a cost of public infrastructure.

Model #3 is new. This is a model that a development is sponsored and accords with the specific interest of non-profitable association. In Japan, only the ORCA project is known, which is sponsored by Japan Medial Association. ORCA is young project yet and it is too early to determine the success of this model.

Model #4 is by local autonomies and governments. Much attention is paid for it just because people feels that the utopia of opensource depends on governmental assistance after many failures of companies and investors.

GNU Manifesto already mentions an example of governmental assistance as Software Tax. This is a taxation that people have to pay specific percent of price of computer and the government gives this to an agency. GNU Manifesto mentions NFS as an agency, but it would be thought FSF.

Actually, existing nations do neither adopt complete free competition nor complete planned economy but keep balance between freedom and welfare. Hence we can not easily refuse GNU's proposal but this doesn't seem to be realistic. In addition, current governmental investments are mainly for utilization of server and client and not for development.

As shown in this section, 'possible supporter' is an unsolved problem. If the opensource keep to spread without a novel turn, it would be highly possible that pure software industry will gradually reduce.

* Disfunctionality of local incentive

Even if 'possible supporter' was determined, under a economic system like Software Tax in GNU Manifesto, software would be only produced according to programmers' interest.

This is critical because once a segment of software market is dominated by opensource, license-based softwares can hardly survive and softwares from local incentives are rarely born.

This problem can be only solved with a sweat economic system. Again in GNU Manifesto, you can find the section titled 'Why All Computer Users Will Benefit' in which benefit for average users is explained in plain words. But it only explains efficiency of planned economics and lacks analysis whether the benefit from share exceeds diversity of software from license model.

During Net Bubble, in order to bridge customers' demand and developers' effort, mediator's business modes were attempted but all failed. If some excellent pre/post-evaluation of project was taken, government and association might be able to produce diversified softwares.

For further analyses, you need to consider both on incentive of opensource developers and on convertible opensource from public to private like BSD license software. But at least we can say that the problem of 'disfunctionality of local incentive' will certainly occur with software licensed under GPL-like license.

* Infringement with standardization

I mentioned recent two troubles on intellectual property right. Not only in regard with copyright but with patent, unintentional infringement troubles are reported periodically.

People in the computer industry may feel a kind of unfairness against SCO. That feeling is for the way of competition: once it publicly spread its technology and later it make the technology back to private, whereas it is aware of what it does.

How about Kochi? We have complex feeling that opensource side had injustice about dead copy and Hitachi PS insists its right too much.

Although these seems to be opensource specific risks, when you look at troubles of patent's infringement, you will notice that infringement of standard technology occurs in the same scheme.

T.4 Fill trouble
GIF trouble
VESA trouble
JPEG trouble

T.4 Fill trouble is the case that in 1992 Iowa University pointed out that CCITT's recommendation X.39 (namely G3 of FAX) infringed its patent acquired by a graduate student and it sued two Japanese companies. It was reported o be settled with payment of licensee fee.

GIF trouble is that once UNISYS had unofficially stated not to apply GIF patent for non commercial use in 1994 but later in 1999 it suddenly started to claim for license against web sites as far as web technology spread.

VESA trouble is that just after the VESA group finished the process of standardization in 1992, Dell Computer announced that VESA infringed its patent, whereas Dell Computer was a member of it. In VESA, members were obliged to open acquired patents. FTC soon started the trial and Dell computer settled FTC charges in 1995.

JPEG trouble is the recent one: in 1994 the joint group of ISO/ITU-T had standardized image format as the recommendations T.81 and in 2002 suddenly Forgent Networks started to claim infringement and to force license. It was reported that several companies including SONY agreed on license and Forgent Networks is expanding its business.

These four troubles all share the common characteristics: At the first stage of standardization, the technologies are seen seen/public and not claimed any right except the planned patent pooling, and after the technologies get spread, someone try to force license. This scheme is the same in case of SCO and unfortunately will be the same in the future infringement of the opensource.

Back to the view point of the opensource, if it was not a standard technology, would some problem happen? Suppose that some people or company releases new opensource software, and after a while, a third party recognizes and claims rights and compensation on it. In this case, this third party looks like act fairly and the impact on the industry is small enough.

Thus, we can say that the exact risk of unintentional infringement is not about the opensource but about standard technologies.

* Opensource as the implementation of standard technologies

As the network bases on mutual dependency of standard technology and the computer network socially grows, the impact of confusion of standard technology gets bigger and bigger. Whether claimers have justice or not, a trouble related to standard technologies pushes the knowledge worker's society into disorder. Obviously, a stabilizing device is required.

True economical value of opensource doesn't come from free of charge. As I mentioned above, whether it is changed per machine or not, is just a problem of economical model. Rather, true economical value of opensource is that it is the implementation of standard technologies. It is needless to emphasize high value of standard technology like TCP/IP or Web.

Current models to process standard technology typically falls into below.

	exactness	open	probability
De-dure standard	○	○	—
De-facto standard	—	×	◎
Consortium	○	△	×

De-dure standard is a model that official organization researches and determines strict specifications like IEEE, ITU-T, ISO or JIS. De-facto standard is a model that a standard is made as a result of competition among industry. Consortium is a model that companies entrust their technologies to the committee which determines specification balancing between elemental technologies. If we supposed the Internet standard to a derivative of de-facto, almost every contemporary standardizing process would fall into one of above three or its derivatives.

Every model has merits and demerits. De-dure standard is strict enough but sometime beaten by De-facto standard. De-fact standard is an official recognition of the present status but sometime a result of unbalanced competition. Consortium sometimes becomes an antitrust matter.

But the point is, whenever any model is chosen, opensources can function as a strong promoting device of standard. For example, OHCI specification explains how USB hub acts in C-like codes, or you can utilize existing sample implementation of PNG or TCP/IP.

One of characteristics of standard technology is that is it not brand new but made of existing technologies and thus it is very difficult to completely eliminate private property, when the technologies comes to the state-of-the-art. This is significant point for success of de-dure or consortium standard.

If we could protect the opensource which is a implementation of standard technology so that we could assure that it is free from private intellectual property, the process of standardization would be highly stable.

*** A proposed amendment**

I dare to show an amendment of the current intellectual property system, even though I am not legal professional. This amendment tries to classify the public resource as public domain according to the enough time passing. In the concrete, we need to change two articles below:

----

o article 51 of copyright law of Japan (terms of protection)

'(2) Copyright shall continue to subsist until the end of a period of fifty years following the death of the author (or following the death of the last surviving co-author in the case of a joint work; the same shall apply in paragraph (1) of the next Article), unless otherwise provided in this Section.'

Following (2), adds below.

'(3) Notwithstanding (2), copyright shall terminate when copies of work are distributed without exercising right over enough period.'

Same additions to article 52/53/54.

o article 67 of patent law of Japan (Term of patent right)

'(1) The term of the patent right shall be 20 years from the filing date of the patent application.'

Following (1),(2), adds below.

'(3) Notwithstanding (1), The patent right shall terminate when patented invention is publicly used without exercising right over enough period.'

The point is that the intellectual property right positively terminates when the property is virtually public domain. Under this amendment we can benefit the stability of public resource at the cost of property owner. He/she comes to have to keep his/her eye on it. As automatic information extraction technologies of text/image/sound mature, it becomes easier to find out infringement among public resource. In addition, the most suitable one who have to pay attention to infringement should be neither licensee nor user nor government nor any other third party but the owner himself who benefits from the property.

We need more arguments about the condition of 'public distribution/use' or 'enough period' but 'distribution via web over two years' may be one line. Only with this quantitative change of system, all cases and troubles I mentioned above can be avoided and owners still keep rights to claim compensation for damages as far as owners exercise rights. We also expect pointing out of infringement during early stages of standardization before impacts grow.

At first glance you might take this amendment as decrease of intellectual property rights, but it actually comes from free competition: 'If the owner can't notice that his/her property is widely used in softwares or products, his/her property right is just obstruction for free competition.' From the point of view of nation's strategy for industry, it is even close to propatent policy.

Further, from the beginning, the disappearing property does not accord with the purpose of intellectual property laws : to contribute to the development of culture and industry.

There might be a side effect that opensource softwares also enter public domain after 'enough period' including GPL, which contains too large legal gray zone. Thus, stabler utilization of opensource might be also expected.

* Conclusion

The utopia of opensource is surrounded by three walls.

First wall is the fact that the opensource defined by license cannot be a stable enough resource without explicit protection of law. As scale of softwares comes bigger and as technologies are more standardized, the impact of unintentional infringement gets bigger and bigger along with payment of compensation. This problem nearly comes real because the opensource becomes a promoting device of standard technology.

Beyond first wall, there are two walls yet : problem of 'possible supporter' and problem of 'disfucntionality of local incentive'. Freedom of programmer is not directly connected to social benefit. No economic device is found yet to bridge. So we have to take the ordinary conclusion that it is the best way to balance the competition between public softwares and proprietary softwares.

With a rash example amendment, I proved that the first of three walls is easily breakable by not qualitative but only quantitative changing of intellectual property system. Every time when we face the trouble like JPEG or SCO, I imagine that it is essential to protect public resource as well as private property.