

4th Annual Conference of the EPIP Association – 24-25 September 2009

Conference Paper

Open source software, patents and royalty-free licensing

The role of intellectual property rights (IPRs) in promoting innovation and, via it, collaborating to the achievement of knowledge society and economy has been at the core of the intellectual property (IP) discourse over the last decades, which have been experiencing a shift in terms of IP paradigm from a fully proprietary regime towards a more open and flexible way of managing IPRs. The development of the OS software is exemplificative of such shift. While the protection of software under the traditional IP law is premised on the assumption that granting an exclusive right encourages creativity and innovation, the OSS movement has emerged precisely in reaction to the perceived failure of traditional IP law to do just that. The OSS offers, thus, an alternative model at the levels of both production and distribution of computer programs via a licensing regime that encourages access to computer information and as well as to any development enabled by the fact of having had access (i.e., in copyright law terms: derivative works). However, OSS licensing is not the only successful example. The Creative Commons (CC) licences offer a similar, even more flexible regime to apply to creative works which, in absence of the choice for a CC licence, would be applied the traditional ‘all rights reserved’ copyright protection. Lastly, the open approach is starting to influence the world of patents as well; few initiatives have been undertaken to adopt a use of patents aiming at fostering innovation by sharing innovation.

In the following sections, mention will be done of the innovative features of the OSS in relation to its development, distribution and legal regime as they are the starting point of any discourse on an open management of IPRs. Such innovative features are worth analysis as they make the OSS a tool capable of contributing to the creation of a society and an economy based on knowledge according to the Lisbon strategy. In the second place, the OSS represents a competitive advantage for those countries that have not become (yet) leaders in the development of proprietary software. The critical role of the legal framework in supporting and fostering the open licensing of computer software will be then analysed. Considerations will be formulated in relation to open licensing via exclusive rights other than copyright (such as patents). Although in its early phase,

traces of patent licensing used not in the traditional proprietary form but in a more open way can already be found and ought consideration.

As well as proprietary and open licensing on computer software can cohabit – and to certain extent take reciprocal advantage¹ – the same may envisage for works protected under patents or even trademarks. However, since licences are legal tools in need of being consistent with the legal framework, the latter play an important role in facilitating or preventing open management of IPRs.

A Premise: IP protection and OSS

Before dealing with the innovative features of OSS, a premise is necessary in relation to how computer programmes are protected under IP law.

Even though historically thought for esthetical creations, since the '80s in the US, and '90s in the EU, copyright law has been the main tool for protection of computer programmes whenever they meet the required criterion of creativity. OSS does not differ in this respect from any other proprietary software. Far from rejecting the rules of copyright law, the OSS movement relies on the application of these rules to set their own open terms of use for protected OSS.

Thus, the difference lies in the way copyright protection is used to enable the creation and development of OSS. Creation and development that are deemed to be much more efficient and fast since the source code of the computer programs is accessible by users/developers whenever they want make amendments or contribute to further development.

However, copyright law is not the solely tool to protect computer programmes. The grant of patents on software has been a problematic issue worldwide for the last two decades. Today, patent protection is regularly granted in the US, provided that the invention produces a concrete, useful and tangible result and that it is new and non-obvious. In the EU, the European Patent Office (EPO) grants patents on computer implemented inventions (CIIs), provided that the claimed subject-matter, besides meeting the usual requirement, has technical character. Very dissimilarly from

¹ The coexistence of OSS and proprietary software generates advantages not only for users but also per entrepreneurs that are capable of combining different business models within their commercial strategy , as the IBM0s experience demonstrate. See, i.e., Martin Campbell-Kelly and Daniel D. Garcia-Swartz, Pragmatism not Ideology: IBM's Love Affair with Open Source Software, 2008. Available at SSRN: <http://ssrn.com/abstract=1081613>.

copyrights, OSS developers have consistently taken the position that software patents generally impede innovation in software development and that software patents are inconsistent with the OSS mechanisms. In other works, a full patentability of software would stifle OSS innovation.

With this in mind, consideration can be turned to the innovative features of OSS in relation to its development, distribution and legal regime.

OSS and Knowledge Society

As to the way OSS is created (and, thus, innovation develops), OSS is developed by Internet-based communities of programmers who voluntarily collaborate to develop software that they or their organizations need.

Mentioning the development of the Apache web server is probably the best way to understand how an OSS project works. Beginning of '90s, an undergraduate student, Rob McCool, while working at and for the National Center for Supercomputing Applications (NCSA) developed a very rough, initial version of a web server and posted the code on the Internet so that other users at other sites could download, use, modify and further work on it. In mid-94, when McCool left NCSA, a small group of webmasters took on the task to continue developing what he had initiated. This core group of 8 users gathered together all documentation and issued a consolidated patch, which is the version that evolved into Apache, whose 1.0 version was officially released – after extensive user feedback and modification – on December 1, 1995. In the space of 5 years since its release – and after many modifications and improvements contributed by users – Apache became the most popular Web server software and gained many industry awards for excellence. In 1999 the foundation was created by the people that had taken over the project development.

Nowadays the Apache Software foundation structure is much complex, but it still works as an Internet-based community where many categories of interested people access and participate: from simple users (who just download and use the last versions of the software, so called 'lurkers'), to contributors (who participate to the mailing list, provide information, criticise), to the categories of 'maintainers' who take responsibilities for the project management. In complex project, or when there are many projects linked together under the same brand, such as in the ASF case, there are just

few trusted developers, one for each project, that authorise new version of the code to be added, and who consolidate all the changes into the updated or upgraded version.

Why describing Apache development and the ASF functioning? Because, in the market of web servers, more than 50% of websites use Apache as web server (data from the Netcraft survey September 2008), whereas about 34% of sites use IIS (Microsoft) and about 6% use GWS (still, a customized version of Apache).

What is new in the case of Apache? It is not only that, from a technological point of view, the Apache web server is considered to be a significative innovative and successful product indeed, but it is also that the practices of OSS projects constitute a novel and innovative alternative to conventional models of innovation. Such alternative presents interesting hints and challenges to prevailing views regarding how innovations should be developed and how organisations should form and operate.

The innovation models conventionally known are, on the one hand, the private investment model, wherein innovation is encouraged and rewarded by the granting of patents to innovators who thereby realize private profits. On the other hand, the collective action model, which tends to operate in science, consists in the relinquishing of the developed knowledge as a public good by unconditionally supplying it to a ‘common pool’. In this model, the problem of incentives to innovate and free-riding are substantial, though there are reputational mechanisms as well as state monetary subsidies to make them still operative.

The alternative model adopted within the OSS realm consists of the compound of the private investment and collective action models of innovations. Such ‘hybrid model’ has been termed by eminent scholar² the ‘private-collective’ innovation model since the ‘inventors’ – the OS developers – achieve proprietary rights on their creation but decide to freely release them. While individual motivations to adhere OSS projects are outside the scope of this work.³

² Eric von Hippel and Georg von Krogh, *Open Source Software and the “Private-Collective” Innovation Model: Issues for Organization Science*, 2009. Available at <http://ssrn.com/abstract=1410789>.

³ A thorough analysis of motivations in Paul A. David and Joseph S. Shapiro, *Community-Based Production of Open Source Software: What do we know about the developers who participate?*, 2008. Available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1286273.

OSS and Knowledge Economy

Yet the OSS phenomenon is more than collective development of innovative software. There are companies that make relevant profits, such as Red Hat, Caldera, and SUSE (Novel) through the ‘distribution’ of the Linux operating system. The question arises, thus, as to how the move from collective development – which takes place for free – to commercial distribution – which generates revenues – occurs. The missing ring of the chain to be analysed are the services related to the OSS that are offered against payment by so called ‘commercial distributors’ of OSS.

As a matter of fact, commercial distributors’ revenues are generated by:

1. Providing the product on CDs rather than as an online download;
2. Providing support (and training) services to enterprise customers;
3. Upgrade and customize services (including the software itself).

In practice, the role of distributors is basically that of selecting among the several versions of the same OS product (if we consider the Linux operating system, we understand that choosing the version to install is not an easy task for an uninitiated); add complementary software, make a package and sell it (in certain cases, some distributors still give this for free), install and customize the package for enterprise customers, train their employees and offer as many services as possible. Sometimes, a shield against IP-related liability is also included in the ‘package’. This is, for example, the case of the Novell-Microsoft agreement that offers a shield to Novell’s customers against the risk of violating Microsoft’s patents. The choice of the OS product on which building the business is of the utmost importance and the competition into the markets of the relevant products (such as the Linux operating system) is fierce. As a matter of fact, should these OSS companies survive on the sole sale of OS software, they would bankrupt since a free version is always available. Should they survive on the sale and services of OS products which are niche products, they would bankrupt as well.

Commercial distribution is however even more than this. By distributing software whose source code is open, i.e. accessible and modifiable by any enterprise-client, distributors transfer much more than a product to their clients. They transfer knowledge. This generates two effects. On the one hand, the enterprises are capable of further developing the OSS in order to adapt it to the different needs that may arise – or, at least, to maintain their IT systems independently. On the other hand, enterprise-clients

are free to choose the maintainer that they prefer without being locked-in with the software house that provides and installs the software (and that is the sole to have access to its source code), as it typically occurs in proprietary distribution.

OSS and 'legal innovation'

Such innovative processes of collective creation and distribution are possible since a different way of using copyright to protect OSS is adopted. Precisely, the protection regime is common to the proprietary software and lies, as above mentioned, into copyright law, while the licensing regime is the main innovative feature of the legal regime of OSS, as it enables both the collective process of innovation and the commercial distribution of OSS above described.

Thus, the current licensing system of OSS constitutes 'legal innovation'⁴ in terms of using copyright licensing to enable online communities of OSS developers to innovate and commercial distributors to operate a new business model.

In order to understand the difference between copyright licensing in the traditional proprietary regime and in the OSS system, the traditional commercial exploitation of proprietary software must be mentioned. Source codes of commercial computer programmes are typically kept proprietary (only the machine readable object code is released) and undisclosed in order to hamper competitors from producing rival products. By contrast, the OSS movement advocates that source code should be freely accessible and available. For this reason, in OSS projects, the source code is distributed along with the object code so that it can be studied, improved, and modified by other programmers and users. That does not mean that users of OS programs can do what they want with OSS as they have to respect, as well as in the traditional proprietary distribution, the licence terms imposed by the licensor.

Different terms of licensing conditions are adopted within proprietary or open licensing. In order to understand such difference, the case of the General Public Licence (GPL) may help. This licence, initially written by Stallman (end of '80s) for the GNU project,

⁴ For a broader definition of 'legal innovation' see Thomas S. Ulen, Nuno Garoupa, *The Market for Legal Innovation: Law and Economics in Europe and the United States*, University of Illinois Law and Economics Research Paper No. LE07-009, 9 (available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=972360) (arguing that there have been a large number of innovations in legal scholarship in the U.S. legal academy over the past 25 or so years and very few from legal scholars in other parts of the world).

related to the project of developing a complete OSS operating system. The operating system was actually developed when the GNU project met the kernel written by Linus Torvalds and the result was Linux (or better GNU-Linux). The version 3.0 of the GPL was released in June 2007. It has been written with the consultancy of Eben Moglen, Intellectual Property law professor at Columbia Law School in NY.

The GPL provides that:

1. Software distributed under a GPL must be released in a way that makes possible the access the source code, which is the core of the software. This is not the case when the software is licensed under the traditional proprietary copyright regime ‘all right reserved’ where only the object (or binary or compiled) code is accessible (yet still protected through copyright) and the source code is kept secret.
2. Beside access to the source code, all licensees (developers, commercial distributors, client-enterprises) also acquire the right to use it, modify it, and further distribute it as well as derivative works as long as they adhere to the terms and conditions set by the GPL (or other OSS licence). This is not the case in traditional copyright licensing where the lawful licensee can just use/run the product/software. Even if the licensee had access to the source code to modify it and adapt it to her needs, this activity would be prevented by copyright law as exceptions to copyright protection in proprietary protection and distribution are limited indeed (and even reverse engineering, where technically feasible, is subject to very narrow conditions).
3. Since the source code is available online, how to avoid its appropriation by third parties? How to avoid that open source software is morphed into proprietary, not open anymore, software? By introducing into the GPL the so called ‘viral clause’, according to which a licensee may distribute derivative works as long as the same GPL is used. Anytime a derivative software comprises material that was accessed under a GPL licence, such work must be released under the same conditions. Such clause has been criticized to the extent that a Berkeley Software Distribution (BSD) licence has been released not comprising the viral effect ⁽⁵⁾.

What are the effect of the legal innovation promoted through the OSS licensing regime?

⁵ Microsoft actually used parts of a BSDed code into XP and some other product. Absent the viral clause, Microsoft could distribute its product through the traditional ‘all rights reserved’ licensing model it always adopted for software, the only condition being that it acknowledged that it had used a BSDed product.

Even more interesting is the case of Apple whose whole operating system relies on an OS project that allows to close the source code (or, better, it does not impose any viral condition). The innovation added by Apple is in the users’ interface, where actually Apple does not have competitors.

In the first place, legal innovation lies in the introduction of a licensing regime that enables collective development of software and, at the same time, avoids free-riding problems, such as those that would be encountered whether the OSS software programs were released to the public domain and copyright on such works were not asserted. This is the reason why OSS development could not exist without copyright law. In other words, the GPL (or any equivalent copyleft licence) establishes a licensing system that implements and enables the private-collective model of innovation above mentioned. Such distribution system permits the release of knowledge to the ‘common pool’ while preventing free-riding. It basically combines the ‘best of both worlds’ by keeping knowledge appropriable and exploitable to the conditions that it stays open.

In the second place, it enables a new business models undertaken by new intermediaries that in turn increases competition within the software market.

In the third place, the OSS licensing fosters innovation even on the side of licensees that are not typical developers – therefore do not participate to the Internet-based communities – but are clients in the commercial distribution. Although being a profitable and innovative business model, by adopting OSS licensing, commercial distribution, on the one hand, makes possible for enterprise-clients to be innovators as well, and on the other frees them from lock-in effects typical of proprietary distribution. The fact that the enterprises adopting OSS are capable of independently choosing to maintain their own IT system or to devolve its maintenance to the party they prefer has the further effect of fostering competition in the market of services for OSS distribution.

The role of the legal system: from facing critical issues ...

What is in the picture then? There is that innovation may follow paths that are not those thought by legislators and legal systems are not always as innovative as the models of innovation developed within the market and by the market.

The approach of the legal system towards the OSS phenomenon has been initially that of preventing processes that were unknown and not *a priori* decided, and eventually adjusting them. The initial, strong scepticism towards the enforceability of OSS licences (and all open licences), although nowadays almost fully outdone, is the sign of such slowness to adapt to changes that the Internet generates. By contrast Internet users and other communities are very prompt to implement successful models outside their initial context (as the creative commons phenomenon can show).

Enforceability issues arise when a GPLed (or otherwise openly licensed) product is used by a licensee outside the scope of the licence – i.e., the terms and conditions set by the licensor are not complied with – and the licensor seeks to stop such behaviour by taking legal action.

Lack of compliance to an OSS licence is manifold. It can occur that the source code is appropriated and ‘closed up’ (for example merged with new code and released in a proprietary way, such as ‘all rights reserved’) or is patented by free-riders. Alternatively, the terms of licence can be violated with regard to the distribution of derivative works; most times is the ‘viral clause’ not to be complied with, which produced again a ‘closing up’ effect. On the other hand, OSS developers can be challenged for having infringed patented software (such as in the well-known case SCO v IBM) by using and elaborating code that is protected (so called ‘opening up’ of closed source code).

The first point to face when dealing with the enforceability issue concerns the ownership of OSS. Given the collective development process, holders of rights need to be identified so to have prospective plaintiffs and defendants of any legal action. The Free Software Foundation (FSF) recommends all OSS developers to assign copyright ownership of their work to the FSF as this can enforce the licence better in case of infringement. Other solution adopted is to assign the copyright to the project. Each project, thus, governs the issue of ownership by assigning it to the FSF or to the project maintainer. Where the ownership is not assigned to the FSF, it can still act on all violations that are reported to it, or at least offer assistance to any copyright holder that wishes to do so.

Secondarily, when facing the issue of validity of OSS licences, it is worth mentioning that within the OSS movement there is a strong mechanism of ‘informal enforcement’. Therefore, whenever the non compliance with the OSS licences occurs between two parties both ‘involved’ in the OS movement (such as two developers, or an OS project and an OSS distributor), the problem tends to be solved through an informal procedure. This starts with a report that is turned in to a specific office of the FSF in charge of monitoring violations and of receiving complains. Then, the office will further investigate and contact the alleged infringers who most of the time are pleased to follow advice and correct any mistake they have incurred in.

The cases of non compliance with the licence that involve parties not member of the OSS movement are those that have raised more problems in terms of enforceability via courts. Although the already mentioned initial scepticism, current case law tends though

to affirm the enforceability of OSS licences (even those more ‘restrictive’ such as the GPL) under copyright law in terms of copyright infringement.

Two main decisions can be mentioned as they show the general acceptance of the OS licensing. In 2006, in *Netfalter v Sitecom*, the Court of Frankfurt upheld the first instance decision in favour of Weller, maintainer of an OSS project. Three software of that project were used by Sitecom and no mention was done of the fact the Sitecome’s product contained software licensed under GPL. Both courts concluded that the distribution of the software without complying with the conditions of the GPL constitutes an infringement of copyright. The copyright was held by Weller as the maintainer of the project. In a similar way, in August 2008, the Federal Circuit – the Unites States Court of appeals – vacated and remanded a lower court’s decision that held that the violation of a GPL licence was breach of contract. The Federal Circuit held that a breach of conditions of a GPL is rather copyright infringement. What is more important is the court’s opinion on the importance of the OSS agreements since they facilitate efficiency and innovation.

At the moment more concern relies in the cases of ‘opening up’ or patent infringement by OSS developers and OSS distributors, such as in the case, recently concluded, of *SCO v IBM*. SCO claimed that it owned part of the UNIX kernel code which is used by IBM on the machines running Linux distributions. It then threatened to sue every corporate Linux-user for copyright infringement, claiming that any Linux user must have purchased a licence from it. After years of litigation, that involved also Novel, in 2007, the district court of Utah concluded that the rightful owner of the copyright covering the Unix operating system is Novel, and rejected SCO’s claim.

Despite the result of *SCO v Novel*, the legal battle that took place gives the idea of the implications of software patents for the OSS. The foundation of OSS lies in the possibility for developers and users to share parts of the source code, to use it, to modify, to redistribute it. This freedom is severely curtailed whenever a new piece of code ends up fulfilling the same function as that of a patented invention. In fact, patent law generally protects the functionality of a computer program and not its expression. By conferring on its owner the exclusive right to manufacture, use, distribute the patented invention, the existence of a patent actually prevents any other computer programmer from independently developing a piece of software with a comparable functionality, even if the new software does not reproduce the lines of code of the patented software. In such circumstances, the risk of using a potentially infringing code would be too high as well as the costs to ask authorizations.

... to fostering legal innovation

The fact that software patents – more precisely CIIs – have been perceived as an obstacle to the development of the OSS is exemplified by the struggle for the so called software patent directive (more exactly CII directive). However, there is no point in condemning CIIs as, on the one hand, they are granted under a constant and to a certain extent consistent interpretation, at least as long as the EPO is concerned, and, on the other hand, OSS seems to have developed despite software patents. Moreover a way around is likely to be found by some private initiatives that make a strategic use of patents in defence of the Linux operating system.

For example, the PatentCommons initiative aims at defending OSS from patent infringements through the establishment of a database of patent pledges and covenants. PatentCommons act as a facilitator by collecting and making available patent owners' commitments to not, under certain terms and conditions, assert their patent rights against third parties who are engaging in activities that might otherwise give rise to a claim of patent infringement. Ownership of patents remains thus with right holders who decide what patents and to what conditions not to enforce them. Most commitments include the following:





1. The beneficiaries who may rely on the commitment;
2. The formal grant of permission allowing the beneficiaries to use patents under specified terms and conditions;
3. The permitted uses identifying the scope of permissible activities and purposes for which the patents can be used without risk;
4. A defensive termination provision identifying conduct that, if engaged in by a beneficiary, gives the patent holder the right to enforce the patent;
5. A reservation of rights that allows the patent owner to terminate the commitment if a beneficiary fails to comply with the statement of permitted use or engages in conduct triggering a defensive termination provision.

All commitments are available through the PatentCommons website by type depending on key characteristics.⁶ The PatentCommon library can also be searched by patent title, abstract, type of patent, patent number, or assignee. At the moment the library comprises commitments adopted for about 500 different patents by 16 assignees (included the OIN). Supported by the Linux foundation, PatentCommons has the precise goal of making OS developers' lives easier by offering shield to legal actions for patent infringements.

A step forward is done by another private initiative, the Open Invention Network_(OIN), which adopts a different system to enable patents to be openly shared in a collaborative environment and to be used to facilitate the advancement of applications for, and components of, the Linux operating system. More in detail, the OIN has been established by both OSS distributors – such as Red Hat and Novel (which have a clear commercial interest in having Linux unchallenged) – and more traditional IT companies, such as IBM, NEC, and Philips as an 'intellectual property company' that 'promotes a positive, fertile ecosystem for Linux, which in turns drives innovation and choice in the global marketplace. This helps ensure the continuation of innovation that has benefited software vendors, customers, emerging markets and investors'.⁷ In order to do this OIN acquires patents and make them available on a royalty-free basis to companies, institutions or individuals that agree not to assert their patents against the Linux System. OIN has a pool of about 100 patents (donated or acquired) and a standard licence agreement whose main conditions are as follows:

1. OIN grants to licensee a royalty-free, worldwide, nonexclusive, non-transferable license of its patents to make, have made, use, import, and distribute any products or services, included those activities that in absence of the licence would be inducement to infringe or contributory infringement (or infringement under any other analogous legal doctrine in the applicable jurisdiction);
2. Licensees grant to licensees a royalty-free, worldwide, nonexclusive, non-transferable license of their own patents for making, having made, using, importing, and distributing any Linux System. In other words they commit not to assert their patents against the Linux operating system or certain Linux-related applications;

⁶ From http://www.patentcommons.org/resources/about_commitments.php#type:

-  Commitments identify specific patents
-  Commitments do not identify specific patents
-  Commitments cover open source licensed software
-  Commitments cover specific standards or technology

⁷ <http://www.openinventionnetwork.com/about.php>

3. Licences last until the last to expire of the OIN patents or licensees' patents, unless earlier terminated;

4. Patents subject to the OIN licence cannot be assigned or licensed unless the assignment or license is made subject to the terms of the OIN licence (similar to a 'viral effect' clause).

Through the indirect commitment not to assert their patents against the Linux operating system or certain Linux-related applications, licensees grant back a licence to allow that their patents to be used within OSS projects by other OIN licensees. In addition they commit to require their assignees or licensees not to assert the assigned or licensed patents in relation to OSS projects.

Similarly to PatenCommons, OIN is thought to create an environment more favourable for OSS projects and developers and make them coexist in absence of a legislative intervention which OS activists would identify in the adoption of a statutory provision that grants automatic and royalty-free licenses to any OS developer/user and to all OS projects (for example 'software patent holders shall offer a compulsory royalty-free licence to OS developers and projects, and a compulsory – but not royalty-free – licence to commercial distributors'). OIN though is something more than a repository of pledges and covenants as it sets a standard licence agreement enabling licensees to use its patents for any purpose. Such condition, offered in order to defend the Linux environment, constitutes a step towards a shared use of patents but should not be confused with a use of patents as open as the open use of copyright that is done under the OS licences or under the Creative Commons licences. In terms of legal innovation then the initiatives now mentioned are far from reach the open use of copyright licensing as they are limited to refraining patent owners from asserting their patents under certain conditions.

A more innovative use of patents (not only as a strategic defence but as a proactive tool for open innovation) is adopted within the BiOS project through the Biological Open Source licenses. As a matter of fact biology is likely to be a domain where a growing number of companies and public research institutions agree to terms of sharing technology, improvements, and safety data. Within the biology field, the BiOS licences constitute the most known legal (and allegedly enforceable) framework to enable the sharing of the capability to use patented and non-patented technology, which may include materials and methods, within a dynamically expanding group of those who all

agree to the same principles of responsible sharing, a ‘protected commons’.⁸ Those who join a BiOS ‘concordance’ agree not to assert IP rights against each others’ use of the technology to do research, or to develop products either for profit or for public good. However, adherents not only comply not to sue each other, they also agree on sharing all improvements derived through the use of the technology that is offered royalty-free for use in research or in creating products, by anyone in any country, based on a legally binding agreement to the following elements. BiOs licences are many and vary depending on the object they transfer (technology, material, data etc.) and on the object been patented or not patented.

More innovative is likely to be the GreenXchange project that Science Commons, launched at the beginning of 2009 but effective at the beginning of next year. The project, funded by Nike, Best Buy, Yahoo!, Mountain Equipment, Co-Op, IDEO, nGenera, 2Degrees, Salesforce.com, University of Washington, Outdoor Industry Alliance, and coordinated by Creative Commons, aims stimulating innovation by increasing research use and rights through a ‘some rights reserved model’ for patent licensing that should become the standard commercial patent licensing for sustainability purposes.

The tools to achieve such goal are the academic research exemption (in need of being reconstructed in the US since case law repealed it) and a system to extend that exemption to corporate research via initially private contracts used to construct a commons for use in sustainability and later on a clearing house.

GX strategy is twofold. First, to promote basic research, the patent owner commits to non-enforcement of patents against users engaged in basic non-commercial research (called “the Research Non-Assertion Pledge”). Second, the patent owner may adopt a standard patent license to enable use of specific patents chosen by the patent owner for applications related to sustainability, which can include some commercial uses. In return, users may be asked to provide attribution, and when applicable, pay a standard licensing fee.

The primary differences between these efforts and the GX are the types of nonassertion pledges and licenses offered. The Eco-Patent Commons and Linux Patent Commons both use patent pledges exclusively, and these pledges allow for some commercial uses of the pledge patents. By contrast, the “Research Non-Assertion Pledge” of GX is targeted specifically for non-commercial research, while the standard sustainability

⁸ The underpinning philosophy is that of OSS: information released in the public domain becomes easily appropriable and can be subtracted from the public domain; whereas information released to a protected commons remains open.

licenses offer a path to commercialization, involving attribution and payment of fees, which are not a feature of other programs. The GX is unique in leveraging the ‘some rights reserved’ approach of Creative Commons and applying it to patents. In addition, the GX is technologically different from other patent sharing efforts. By using standard formats to describe both patents and rights, GX enables integration with major search engines, software systems, content creation systems, and the creation of patent gap analysis.