18 On intellectual property rights: patents versus free and open development

Alan G. Isaac and Walter G. Park

Introduction

Intellectual property rights (IPRs) encompass a broad array of legal protections, including patent rights, copyrights, trademark rights, plant breeders' rights, protection of trade secrets, industrial designs, layout designs for integrated circuits, and geographical indications (regarding the origin of goods and services). These legal protections serve diverse purposes and functions, and the institutions supporting them are also quite distinct. In this chapter we focus on the patent system.

Patents are a primary instrument by which commercial firms secure legal rights to inventions. In the United States, the core economic justification of patents is that they improve welfare by stimulating discovery, disclosure and dissemination. However, patent systems generate both economic benefits and costs. In principle, the theoretical effects of patents on innovation depend on the institutional environment and on the nature of technology. Both academics and policy makers are aware of circumstances in which patents can adversely affect innovation. For example, in some situations, the patent system has the potential to create a thicket of fragmented, overlapping property rights which raises the costs of innovation.

This chapter explores one alternative to the reliance on patents: 'free and open' development. We refer to a system that relies on free and open development as an 'open innovation' system. Open innovation systems side-step the patent thicket problem by not asserting patent rights to inventions. Open innovation systems rely on the free sharing and dissemination of knowledge. The idea that an open innovation system could successfully foster innovation may seem outlandish: where are the necessary incentives for innovation? We therefore offer a tentative discussion of how open innovation systems can work (that is, of the incentives and governance structures). We also discuss some of the advantages and disadvantages of each system in stimulating inventive activity.

As a concrete illustration of the issues, we consider the software industry. Substantial reliance on software patents is very new in this industry, and it is an industry where thicket problems seem likely (due to the nature of software inventions and patents). It is also an industry where free and open development activity has been intense. Our examination of the software industry leads us to tentative generalizations: principles and lessons that could apply to other industries and technologies. We present these in the hope of simulating further debate.

The chapter proceeds in six further sections as follows: (i) provides some brief background on the justifications of intellectual property rights; (ii) provides some background on patent law and institutions; (iii) discusses some of the economic aspects of patent protection; (iv) explores the problem of patent thickets; (v) discusses the analytics of open innovation; and (vi) concludes.

Background

From an economic perspective, property is essentially a collection of enforceable duties and privileges. We call these enforceable duties and privileges 'property rights'. Property rights are a fundamental determinant of economic activity. Institutions of exclusion, which need not be legislated or even explicit, bring into existence property and motivate the creation of property. Equivalently, these institutions of exclusion embody the current property rights that make it rational to create new property.

The economic perspective on property is close to the legal view but perhaps far from everyday parlance, where we call a piece of land, or a house, or a shirt 'property'. This common usage serves as convenient shorthand and often creates little confusion. For example, a couple who buy a residence are unlikely to be surprised that they have not acquired the right to build a toxic waste dump on it. On the other hand, they may be surprised to learn that they have also acquired a set of upkeep duties imposed by local legislation. This indicates the imprecision inherent in everyday parlance.

Confusion grounded in such imprecision may occasionally sidetrack discussions of public policy. Policy discussions of property rights often refer to legislating the protection of property or, in an even more misleading turn of phrase, the protection of property rights. From an economic perspective, such rhetoric obscures the most fundamental and challenging aspect of property rights: property is *created* by ever-changing human institutions that render feasible the maintenance of various restrictions on human behavior. Property does not exist prior to these institutions. The institutions generate the property rights and thereby the property. Public policy modifies property rights by modifying these institutions; it does not simply determine the 'protection' of pre-existing rights. Public policy in the area of property rights is a project of creation and destruction, not of discovery.

Economists are particularly interested in public policy that influences the distribution and creation of wealth. This chapter reflects that emphasis. Of course, the related question of how economic considerations influence the structure of property rights is also very interesting. The enclosure movement in England reflected the economics of the woolen industry (Weber 1923 [1981]). Native American property rights institutions changed in response to the fur trade stimulated by European colonists (Demsetz 1967). Property rights changes in response to increasing urban density in nineteenth-century America include the loss of the long-established easements of light and air (Friedman 1985, p. 413). Even now the boundaries of patentability and of the concepts of fair use of copyrighted material are shifting in response to the rise of information technology in the late twentieth and early twenty-first centuries. Clearly, changing economic conditions often lead to the creation and reallocation of property rights.

This chapter speaks of property rights as social constructs. Others speak of property rights as fundamental constituents of a metaphysical moral reality. For example, many libertarians and classical liberals assert a 'Lockean' claim that there are natural rights to our persons and tangible property, and that the basic justification of governments is to protect those rights. Such views are important for public policy, whether or not they are correct (however that might be assessed), because they determine some of the political attitudes toward property rights and thereby influence institutions and legislation. For example, whether or not it is 'self-evident' that individuals 'are endowed by their Creator with certain inalienable Rights', the notion was certainly politically influential. Metaphysical, rights-based moral reasoning influences the structure of property rights, that is, the social structures of duties and privileges that constitute property.

Justifications of intellectual property rights

It is often said that IPRs 'protect creations of the mind', but what human creation is not a creation of the human mind? IPRs are embedded in institutions that restrict the use of inventions or creative works. The rhetoric of 'protection' appears to appropriate the ancient social legitimacy of certain kinds of property rights for a different application: the right to exclude behavior that does not directly affect one's ability to enjoy a good or service.

At least since the nineteenth-century patent debates, proponents of copyright and patent protections have often used the rhetoric of natural rights and talk about the prevention of theft.[1] The extreme version claims that the fundamental Lockean argument that a person naturally owns the fruit of their labor applies even if that fruit is an idea. This view would apparently imply a radical extension of IPRs far beyond the bounds of existing institutions. For example, it would apparently allow IPRs even for abstract ideas, even mathematical theorems, which current IPR regimes do not allow. Additionally, such metaphysical claims imply no bounds to the period of ownership, and perpetual patents conflict with one of the primary pragmatic aims of the patent system.[2] While such pragmatic difficulties will certainly influence policy, they might appear irrelevant for metaphysical reasoning.

Even at the metaphysical level the argument appears problematic, however. Most obviously, it fails to explain why patent rights should be able to exclude independent development of an innovation. Furthermore, it is far from obvious that the usual Lockean reasoning can be extended to non-congestible intangibles such as intellectual property, especially when the assertion of such property rights diminishes the property rights in tangibles. The application to IPRs does not adequately ponder what 'fruits' are by right enjoyed by a creator: those of use, or those deriving from exclusion. These sets are peculiarly non-overlapping in the arena of intellectual property. Property rights in a tangible good include rights to restrict others' access because their access can interfere with the owner's enjoyment of the specific good. In contrast, IPRs are rights to restrict others' access to an instantiation of an idea, even when that access constitutes no interference of the owner's enjoyment of the creation. In the case of patents the attenuation of the rights of others is particularly obvious. The entire intent of patent protection is to restrict what others may do. Patent ownership does not even ensure the owner of a right to use the patented technology – since its use may be dependent on other patented technology – but merely grants a right to take alleged infringers to court (Thomas 2002).

The pragmatic view stresses utilitarian rather than metaphysical considerations. The core pragmatic claim for IPRs is that innovation will be undersupplied in their absence. For example, Romer (1990) suggests that innovators must anticipate a period of monopoly rents to justify the sunk costs of innovation – an argument often attributed to Joseph Schumpeter. The core pragmatic justification of IPRs therefore claims the presence of a trade-off. By means of IPRs of limited duration, society provides innovators with potential rents in order to increase the production and diffusion of innovations. Supportive claims specific to patents are that IPRs increase the disclosure and dissemination of useful knowledge: strong patent protection may reduce the need for invention secrets. Patents are sometimes characterized as a social contract, wherein patent protection is granted in exchange for the surrender of invention-specific trade secrets.

Dynamic considerations are clearly fundamental to the pragmatic view since typically innovation takes place in one period and compensation in a future period. From a static perspective, the temptation is not to provide IPRs once an innovation is developed, for there are no incentive effects left – only rents. Since uses of an idea are non-rivalrous, the optimal policy appears to be to allow free use. Thus static considerations speak against rights to exclusion. From a dynamic (repeated game) perspective, this conclusion is flawed because if innovators do not expect compensation, there may be no innovation and therefore no new ideas to freely distribute.

Intellectual property policies face a time-inconsistency problem: *ex ante* it is optimal for the authorities to promise exclusive rights (to stimulate innovation), but *ex post* it is optimal for them to deny exclusive rights (to allow free use). Brute promises are not credible: unless policy makers are institutionally constrained, innovators will disbelieve promises of legal protection. Pragmatically speaking, this suggests that policy makers cannot optimally stimulate innovation while systematically revoking IPRs.

If reputation effects were absent, a state might best promote the material standard of living by revoking all existing patent protection and then promising never to behave that way again. Of course pragmatists find this observation irrelevant: a state that revoked IPRs once could not avoid reputation effects (namely, the expectation that it would so act again). Advocates of metaphysical property rights find such proposals not just irrelevant but immoral.

IPRs are sometimes characterized as a pragmatic response to the failure of free markets to supply adequate innovation, but IPRs like other rights are simply one of the conditions that determine the nature of markets. Like property rights in tangible goods, IPRs create markets. Economists are inclined to wonder whether alternative market solutions would fail to emerge in the absence of IPRs. For example, what might prevent innovators from contracting with the potential beneficiaries of the innovation? The most obvious answers concern transactions costs and moral hazard problems, so these appear to be good areas to focus any policy discussion of IPRs.

Metaphysical and pragmatic justifications of IPRs both rely on presuppositions. Natural law justifications of IPRs rely on non-empirical claims about the essential nature of property rights. Pragmatic justifications rely on empirical claims about the net effects of IPRs on creative activity. It is important to the pragmatic justification that IPRs not only yield creative activity that would otherwise be inefficiently quiescent but that IPRs do so in sufficient quantity to justify the costs introduced by the system. For example, pragmatic justifications of patent systems assume positive effects on the discovery, disclosure and dissemination of useful knowledge that exceed the costs of the patent system in terms of the deadweight losses (deviations from marginal cost pricing) and resource costs (of operating the system).[3] Empirical assessment of this assumption is a challenge, largely because of measurement issues. There is much innovation that occurs without patent rights (particularly for innovations that are not easily imitated) and empirical analysis needs to control for this.[4] Furthermore, welfare assessments need to control for the fact that the incentives for private generation of information may in some cases be excessive, so that the patent process induces costly, duplicative resource expenditures (Hirshleifer 1971). For example, firms who could work together to more cheaply produce a non-rivalrous input might instead waste resources in a patent (winner-take-all) race.

In the United States, the legal justification for IPRs is closely related to the pragmatic economic justifications. US patent law is based in Article 1 section 8 of the US Constitution, which gives Congress the power 'To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries'. This is therefore an important criterion by which to judge the efficacy of modern intellectual property institutions.

Contemporary patent laws

This section provides a brief introduction to patent laws. We cover the basic issues of patentability and the conditions for patent grants. Two key points are central to our discussion. First, patents are intended to protect not ideas *per se* but rather innovative practical applications of ideas. Second, patents are intended to promote access to ideas and to the know-how behind the application of ideas.

In most countries, the government makes the laws governing patents, the courts interpret them, and a government patent office implements them. The patent office receives applications for patent protection and decides on whether to grant a patent. In what follows, we speak primarily to US patent law and mention practices in other countries where relevant.

What is patentable?

The distinction between ideas and applications of ideas lies at the heart of patent statutes. Patents can be awarded for new solutions to specific practical problems. Products and processes are patentable. Abstract ideas or mere suggestions are not patentable. For example, $E = MC^2$ represents an unpatentable abstract idea, but a heat conduction system that applies this principle of atomic energy is patentable. Two or more inventions can be patented that use the same idea, so long as the inventions are sufficiently distinct as to avoid infringing each other (or contain infringing material). More on this later.[5]

These considerations are evident in Section 101, Title 35 of the US Code (Patent Act): 'Whoever invents or discovers any new and useful *process, machine, manufacture, or composition of matter*, or any new and useful improvement thereof, may obtain a patent therefore, subject to the conditions and requirements of this title' (emphasis added).

In practice, distinguishing between ideas on the one hand and products and processes on the other can be a challenge, particularly in some new technological fields (such as software and biotechnology). Patent offices and the courts nonetheless must make this distinction in their decisions or rulings. Nobel prizes, academic tenure, or other rewards may provide incentives for abstract research, but the patent system does not (and is not intended to) play that role. This is reflected in international laws, which list the following exclusions from patenting (WIPO): discoveries of materials or substances already existing in nature; scientific theories or mathematical methods; plant or animal varieties, or essentially biological processes for the production of such plant or animal varieties, other than microbiological processes; schemes, rules or methods, such as those for doing business, performing purely mental acts or playing games; and methods of treatment for humans or animals, or diagnostic methods practices on humans or animals (but not products for use in such methods).

What can be granted?

The above discussion focused on subject matter that is patentable. For a patent to be granted, however, the patent applicant should demonstrate the novelty, utility, and non-obviousness of the innovation.

An invention is not novel if it has been known, practiced, used, or sold previously anywhere in the world. For example, novelty is destroyed if the inventive idea has been published in a journal, book or dissertation, or practiced by a firm, cooperative, public enterprise, or individual. Any invention that is already in the public domain cannot be patented. An invention is also not novel if a patent application for the same (or similar kind of) invention was accepted or is pending elsewhere in the world. Inventors can also forfeit the right to a patent if they display the invention in a public place (other than for the purpose of testing a prototype at a well-recognized exhibition).[6]

An invention has utility if it is capable of industrial application. This reflects the point made earlier that patent protection should not be available for purely abstract ideas or creations. Patents on genetic discoveries are particularly controversial because in many cases it is not known what functions certain genes play or what utility they provide.

The non-obviousness of an invention pertains to the inventive step (or 'quality jump') of an invention; that is, its inventive contribution (or value added) to existing inventions or technical knowledge. For example, Section 103 of the US Code does not permit a patent if 'the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains'.

Obvious inventions are not only not novel, they may infringe on existing patent rights. The inventive jump must be large enough not to infringe existing patent rights. The required distance from existing patents – often called patent 'scope' – varies substantially across jurisdictions. In the United States, patent scope is broad. In Japan, patent scope tends to be narrow: patenting of minor inventions around an existing patent is permitted. Patent holders enjoy greater market power in a system where patent scope is broad. Entrants find it easy to enter markets where patent scope is narrow.

Patent scope also helps determine the extent to which two or more inventions can build on the same idea (or on each other). In the United States, a litmus test is provided by the '*doctrine of equivalents*', which holds that inventions that substantially perform the same function, in substantially the same way, and produce substantially the same result, are the same inventions. The doctrine allows the claims of an invention to cover not only those things that are *explicitly* stated in a patent document but also those that are *implicit.* Under the doctrine, an accused is liable for infringement for using the essence of a patented invention without literally infringing it. As a weapon of litigation, the doctrine of equivalents can be exercised by patent holders to make technologically neighboring inventions liable for infringement. This pressures rival firms to distance themselves – in 'technological' space – from the rights holder.

Enablement requirements of a patent

Each patent application must usefully disclose the details of the invention. The standard of disclosure is that a person skilled in the art can replicate and use the invention (without undue experimentation). This is the '*enablement'* requirement. The inventor is required to explain not the best possible implementation but rather the best that he is aware of at the time of applying for a patent. Lack of fulfillment of these requirements is grounds for patent rejection (before the fact) or patent invalidity (after the fact). In the United States, the standard of information disclosure required by law is provided by Section 113 of the US Code (and is representative of other national laws):

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to *enable* any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same, and shall set forth the *best mode* contemplated by the inventor of carrying out his invention. (Emphasis added)

Patent applications are published 18 months after the date of application, whether or not patent rights are subsequently granted. If a patent is not granted for reasons other than infringement, the invention immediately enters the public domain. If a patent is granted, the idea immediately enters the public domain, and the invention enters the public domain when the patent expires.

From the social contract perspective, knowledge dissemination is a crucial quid pro quo for patent protection. By implication, the enablement requirement is a crucial component of the patent system.

Limitations and other conditions

Policy authorities may impose other conditions or limitations on patent rights. Examples include working requirements and compulsory licensing.

In contrast with the current US system, some jurisdictions have *working requirements*. Failure to 'work' the invention (by manufacturing or marketing it) within a specified time period may lead the patent holder to forfeit his rights to the invention or face *compulsory licensing* (that is, a mandate that the patent holder license his invention to third parties). While working requirements have been justified as forbidding the hoarding of technology, they clearly reduce the strength of patent rights by reducing the option value of patents. It restricts the right of the patent holder to choose the most opportune moment to market or manufacture his invention. For example, a patent holder might be temporarily financially unable to work the invention or might find current market conditions to be unfavorable.

Compulsory licensing may also be imposed in response to antitrust or anticompetitive actions of patent holders. This power may be used to limit patent blocking. Blocking arises when different patents cover subject matter in a way that manufacturing one good would cause an infringement of the patent rights of others. Blocking may arise if one technology is an improvement over another and where the improvement cannot be practiced without the use of the original, core technology. If the rights to the improvement and to the original technologies are owned by different patent holders, the original owner could block the improver by refusing a license or demanding 'unreasonable' terms. When the parties cannot resolve the problem through private negotiation, the parties can turn to the court system. Conflicts such as these take matters out of the sphere of intellectual property law and into that of competition law (for example, the US *Sherman Act*, Section 2).

Another anticompetitive situation that might arise is if one patent holder owns the rights to a technology that serves as an 'essential facility' for other producers in the market or downstream. In certain environments a specific computer operating system may prove to be an essential facility.[7] A similar situation arises if a technology owned by a patent holder becomes a *de facto* standard for an entire industry. In these cases, patent rights extend the market power of the patent holder considerably. Again, these matters come under the purview of competition policy.

Patents create a temporary right to exclude others from practicing an invention. While this may in some situations result in a monopoly producer or supplier of a good or input, there is nothing illegal about being a monopoly. Rather it is illegal to engage in conduct that maintains and extends market power. Determining when antitrust considerations should supercede IPRs is a difficult matter. Situations where patent holders deny access to an essential facility, block the exploitation of other technologies, or refuse to license or to deal, are cases where patent rights might come under antitrust scrutiny. In these instances, competition policies could limit the exercise of patent rights.[8]

Application: software patents

The general considerations raised above find an interesting application in the area of software-related patents. In the closing decades of the twentieth century, software production became a substantial source of economic value. In 1972 the Supreme Court ruled that mathematical algorithms are non-statutory subject matter.[9] The US Patent and Trademark Office (PTO) consequently considered computer programs to be unpatentable, like laws of nature or mathematical relationships. As a result, copyright and trade secrecy were the primary tools of software-related intellectual property protection.

This situation changed radically in the 1980s. In 1980 the Supreme Court asserted that Congress intended 'anything under the sun that is made by man' to be patentable subject matter. In 1981, the US Supreme Court asserted essentially that, although algorithms are not patentable, software that has a technical effect is patentable.[10] Critics consider this an odd ruling: the court appears to rule that combining admittedly non-statutory subject matter (the software algorithm) with known art (the curing frame) yielded a patentable 'process'.

The struggle to clarify the patentability of software led in 1996 to the PTO's *Examination Guidelines for Computer-related Inventions*, which made the possession of a 'practical application' the key criterion for patentability of the software:

> A process that merely manipulates an abstract idea or performs a purely mathematical algorithm is non-statutory despite the fact that it might inherently have some usefulness. For such subject matter to be statutory, the claimed process must be limited to a practical application of the abstract idea or mathematical algorithm in the technological arts. For example, a computer process that simply calculates a mathematical algorithm that models noise is non-statutory. However, a claimed process for digitally filtering noise employing the mathematical algorithm is statutory. (PTO 1996, Section IVB.2.b.ii.)

The result has been an increase in software patent applications as well as increased criticism of the appropriateness of some of the patents granted.

Critics believe that the courts and the PTO have tried to draw a line where none exists, since software applications are inherently a sequence of mathematical computations. The PTO and the courts, in turn, have insisted that there is a fundamental distinction between mathematical algorithms and other software-driven computer processes.

Prominent computer scientists (notably Donald Knuth and Richard Stallman) reject this view and have asserted that patents are being granted for programming techniques likely to be found in the homework of first-year computer science courses. They suggest that patent examiners fail to see the obviousness of many software-related inventions, and have overlooked much prior art in the field of software innovation. As Stallman (2000) has emphasized, there is also a sense that the PTO's limitations are being deliberately exploited by patent applicants who wrap simple and obvious ideas in complicated phrasing to make them look like patentable inventions.

A converse source of controversy concerns the relationship between software patents and the enablement requirement. As discussed above, disclosure is a quid pro quo demanded by society for the benefits provided to the patent holder. As in other industries, the availability of software-related patents is supposed to encourage inventors to disclose the underlying technical details of their inventions. Unfortunately, it remains an open question whether the intent of the disclosure requirements is fulfilled in the case of software-related patents. First, like other patents, software-related patents grant 20 years of patent protection. However, in such a rapidly developing field, many observers have argued that a shorter patent duration or a narrower patent scope may be more appropriate. Second, the disclosure of a software program's source code has not been required for satisfaction of the enablement plus best mode requirement. Indeed, applicants are 'encouraged to functionally define the steps the computer will perform *rather* than simply reciting source or object code instructions' (emphasis added). A more obvious disclosure requirement would be for a functional definition, ideally in a specified and widely used modeling language (such as the Unified Modeling Language), in *addition* to source code provision.

It may seem that the lack of a source code disclosure requirement results in an inadequate disclosure standard for software patenting.[11] However, one may argue that source code disclosure eliminates the reasonable experimentation that may be required to implement an adequately disclosed invention, so the requirement of source code disclosure would create an excessively demanding standard of disclosure for software inventions. Court rulings indicate that the enablement requirement of software patents varies by the nature of the invention and by the type of computer program needed to carry it out. If it is relatively straightforward for a programmer skilled in the art to write a computer program to carry out the invention, then the source code of the patented invention is not central to enablement. The duty of the patent holder in such cases is to be very clear, transparent and specific about what those steps are.[12] Indeed, the court rulings suggest that disclosure of the functions of the software is generally adequate because, 'normally, writing code for such software is within the skill of the art, not requiring undue experimentation, once its functions have been disclosed' (quoted from Lemley et al. 2003, p. 210). In contrast, if undue experimentation (that is, a lot of trial and error) on the part of a skilled computer programmer is required before the programmer can repeat the invention, the source code must be disclosed. For instance, in one case (*White Consolidated Industries v. Vega Servo-Control*), the court ruled that enablement was unsatisfactory because it would take a skilled programmer about two years to implement the invention described in the patent specification.

These court rulings raise a very interesting question in cases where source code is not judged as crucial to enablement. If writing the necessary code is 'normally' within the skill of the art, can the software invention still pass the test of non-obviousness? This suggests that there should be a substantial burden of proof for any patent applicant who claims non-obviousness for a software-related invention but simultaneously insists that source code provision is unnecessary. The ability of a programmer to do without the source code suggests that any skilled programmer approaching the same problem might produce the same solution: that is, it suggests that the invention is obvious to someone with normal skill in the art. Thus software patents appear to create a quandary: either the system is permitting the patents of some obvious things or it is relaxing the enablement requirement.

Some economic aspects of patents

The previous section focused on a few legal and institutional considerations of patents. We now turn to economic considerations. The theoretical literature on the economics of patents is rather large and varied, and we consider only a few salient economic considerations.

Why firms patent

The decision to patent depends on economic factors, such as market conditions, government policy, costs of patenting, and reputational, signaling and other strategic considerations. An inventor will seek patent protection for an invention if the net benefit of procuring patent protection exceeds the cost of filing for protection. We may represent this condition as $v_p - v_o > c$ where v_p and v_o denote the value of an innovation with and without patent protection, and c denotes the total cost associated with filing a patent (including, for example, the present value of any maintenance fees). The value v_o captures the best alternative to patent protection, including any steps taken to appropriate the rewards from the innovation (such as lead time, reputation and secrecy). Thus the value of a patent right is the incremental return ($v_p - v_o$). A patent application is filed when this value exceeds the (total) filing costs.

The conventional wisdom is that firms demand patent protection in order to safeguard their intangible assets, which imitators might copy and distribute at nearly zero marginal cost without incurring any sunk development costs. As we have discussed, a standard justification of patents is that they may secure gains to innovators and thereby ensure adequate incentives to innovate.

Recent survey results challenge the conventional wisdom. For example, Levin et al. (1987) report that firms do not, in general, regard patent protection as very important to protecting their competitive advantage (and thus to appropriating the returns to their investments). Alternatives such as trade secrecy, lead time, reputation, sales and service effort, and moving quickly along the learning curve are more important.

If patent protection is not important as an instrument for appropriating the returns to innovation, why do firms patent (and patent a lot)? Cohen et al. (2000) report a variety of reasons why firms patent: to block rivals from patenting related inventions, to gain bargaining chips in cross-licensing agreements, and to measure internal performance (of the firms' scientists and engineers). Their survey indicates that these factors are more important determinants of patenting than the direct protection of reserch and development (R&D) investment returns.

These survey findings should be viewed as provocative rather than definitive. First, the importance of patents (and purpose of patenting) varies by industry, being very important for chemicals and drugs. Second, the responses of firms (or their attitudes towards patents) may have been influenced by the patent regime in place: it may be easier to dismiss the importance of patent protection when it is readily obtained. Third, the responses of interviewees may not be fully comparable: the interpersonal meaning of numerical rankings is uncertain. This makes it difficult to tell whether the responses reflect differences in firm behavior or random errors. Fourth, the surveys are based solely on US firms' experiences. A similar comprehensive study for Europe and Asia would shed additional light on patenting behavior. For example, the strength of patent regimes varies much more internationally than domestically, so international data may be able to address our second concern above. Despite the methodological and theoretical problems, the available survey evidence is provocative and should stimulate further inquiry about the patent system.

A simple, illustrative model

We shall model one way that patents can be productive or counterproductive, depending on the nature of technology. We choose a model that makes no attempt to introduce dynamic or strategic considerations. It is an essentially static 'one shot game' model adapted from Bessen and Maskin (2000).

Consider a single firm that can engage in R&D. By sinking cost c_1 the firm has a probability of research success p_1. The expected value of undertaking the research is $p_1v_1 - c_1$, which is of interest only in the case where this is positive. Following Bessen and Maskin, we keep things simple by assuming that this firm can capture the social value of its innovation, so that R&D takes place if and only if it is socially desirable.

Now introduce a second firm, a pure imitator that produces an imperfect substitute for the first firm's product. We allow imitation to be costless to the imitator, and after imitation the innovating firm only captures a fraction $s < 1$ of the social value. The net payoff to R&D falls to $s\ p_1v_1 - c_1$. If this is negative, socially desirable innovation does not take place. As Bessen and Maskin note, this is the classic case for patents. Without patents, it may not pay a firm to invest even if it is socially desirable. Patent protection can ensure the ability of the first firm to capture the social value of its innovation, thereby ensuring that a socially desirable research expenditure takes place.

Coase (1960) is often cited in support of the proposal, often called the 'Coase theorem', that in the absence of transactions costs the allocation of property rights does not affect efficiency. This suggests that efficiency-based arguments for IPRs should be explicit about how the Coase theorem fails to apply. For example, in our model of the classic case for patents, an alternative to patent protection might be for the innovating firm to pay the imitating firm not to imitate. This is another arena where intellectual property law and antitrust law interact: such a negotiation may not be allowed by antitrust law.

Now suppose $sp_1v_1 - c_1 > 0$. In this case patents are not needed to induce R&D, since the firm can recoup its investment without patent rights. In this simple model, the pragmatic justification for patent protection disappears, but patent protection is neither harmful nor productive from a social welfare point of view. (In a dynamic context, however, the level of R&D investment is an endogenous decision. The expected level of profits should influence how much R&D the firm chooses to undertake.)

Bessen and Maskin (2000) note that patent protection may still prove beneficial if firm 2 is not a pure imitator but instead is symmetric in its ability to produce R&D. By assuring economic rents to one firm, patent protection can induce both firms to invest in R&D.

Let us consider a different situation. Assume $sp_1v_1 - c_1 > 0$ so that firm 1 will conduct R&D even in the absence of patent protection. But this time suppose firm 2 is also an innovator that is working on a complementary innovation. That is, the innovation has value only given access to the innovation of firm 1, so that a patent granted to firm 1 will grant a holdup right over firm 2. Firm 2 can sink cost c_2 to produce a probability p_2 of research success. The innovation has a social value v_2, which firm 2 can capture. If firm 1 does not have a patent right, firm 2's expected value of undertaking the research is $p_1p_2v_2 - c_2$. We consider the case where this is positive, so that research is worthwhile to firm 2.

If firm 1 has a patent right, then subject to the constraint that firm 1 will not lose by licensing its technology, firm 2's expected value of undertaking the research is no more than $p_1p_2[v_2 - (1 - s)v_1 - t_2] - c_2$, where firm 2 now pays a license fee to firm 1 (at least $(1 - s)v_1$) in order to produce and where t_2 is the transaction cost of negotiating a patent license. The new expected value of undertaking the research may be negative if the license fee is too high, the innovation value too low, or the transaction costs too high. The problem is of course that firm 2 cannot afford to compensate firm 1 for the decline in firm 1's ability to capture the social value of its innovation. In this case, patent protection allows firm 1 to block firm 2's innovation. Firm 2's valuable innovation is subject to 'holdup' by firm 1.

Patent thickets

Shapiro (2000) defines a 'patent thicket' as 'an overlapping set of patent rights requiring that those seeking to commercialize new technology obtain licenses from multiple patentees'. For example, he notes that a semiconductor manufacturer can potentially infringe on hundreds of patents with a single product. In such circumstances the transactions costs of negotiating licenses with many different patent holders might prove prohibitive even for a valuable commercial innovation. Shapiro argues that the perceived danger of holdup has introduced a threat to innovation that is 'of first-order significance'.

The size of the transaction costs that must be incurred to negotiate a complete set of licenses is one possible measure of the density of the thicket. These costs will vary with industry structure and also with the nature of the innovation process within an industry. However, even aside from the transaction costs dimension, patent thickets are likely to be socially costly. Antoine Cournot demonstrated that the behavior of multiple input-supplying monopolists can lower total profits and simultaneously reduce consumer welfare. Shapiro (2000) extends this analysis directly to patent thickets and argues that coordination among the input suppliers can therefore improve social welfare. Buchanan and Yoon (2000) generalize this result and show that multiple rights-to-exclude generally lead to resource underutilization, as suggested by the Heller (1998) discussion of the 'tragedy of the anti-commons'. Patent thickets introduce the possibility of a perverse outcome in the patent system: in a classic tragedy of the anti-commons, a system whose acknowledged purpose is to promote innovation may produce patent thickets that stifle it.

Where innovation tends to be highly incremental and cumulative, essential licensing for a new innovation is more likely to involve many patents. If only a few firms hold most of the fundamental patents in an industry, or if firms can form an adequate patent pool, the per patent transaction costs involved in licensing should be diminished. However, industries in which innovation is highly incremental and cumulative may prove particularly likely to see high dispersion of innovations across firms and even across private individuals.

(Software and biotechnology, for example, appear susceptible to such dispersion.) Such industries are therefore likely to develop dense patent thickets. In contemplating the possible costs of such patent thickets, one may analogically consider the field of mathematics: developments are highly incremental and cumulative, and few observers would propose burdening mathematicians with the requirement that they negotiate a license for each extant theorem used in a new proof. Are lessons from the vigorous field of mathematics, where patenting is forbidden, relevant to other areas of incremental and cumulative creative activity?

Cooperative arrangements can mitigate patent-thicket problems. For example, large manufacturers often cross-license their large present and future patent portfolios. This may seem a promising and potentially pro-competitive arrangement, but smaller and newer players may be stymied. Cross-licensing among larger corporations can promote collusion and the exclusion of competition from new entrants (Boldrin and Levine 2002). For example, semiconductor firms accumulate large patent portfolios that they then cross-license with rivals (Hall and Ziedonis 2001). This is clearly another area where patent policy and antitrust policy overlap: antitrust law historically has viewed with suspicion mechanisms for cooperation among competitors. This creates a policy dilemma: the cooperation necessary to avoid patent thickets may be precluded by antitrust concerns. Shapiro (2000) proposes that policy makers can ensure that cross-licensing is pro-competitive by attending to the nature of the patents involved: if patents licensed together are complements, not substitutes, then cross-licensing is more likely to be socially beneficial.

Patent pools offer another potential solution to patent thickets. Firms with interlocking patents may form an organization for the purpose of sharing patent rights. The organization could be an open pool, where members license patents to one another and to third parties, or a closed pool, where members license only to one another. An important superiority of patent pools over regular cross-licensing agreements among firms is that the pool acts as an entity *vis-à-vis* licensing to outsiders. In a cross-licensing agreement, one party has the right to use another's patent but not to sublicense it to a third party.

Once again antitrust considerations arise. As with any joint venture or cooperative arrangement, patent pools may generate opportunities to behave as a cartel or fix prices (for example, royalty rates on licenses). Yet patent pools may serve the public interest if they integrate complementary innovations, promote knowledge sharing, and reduce the transaction costs of negotiating licenses. Regulatory accommodation of patent pools has fluctuated over time, and current acceptability was probably influenced by the prevalence of complementary innovation in the computer industry (Lerner et al. 2003). Current law appears to reflect these issues of substitutability and complementarity: US Antitrust Guidelines permit patent pools among 'essential patents' – that is, among patented inventions that are complementary.[13] Another requirement of these guidelines is that members of a pool retain the right to license independently, which helps destabilize any collusive behavior.

Standards

The emergence of formal or informal industry standards allows for coordination of industry development efforts. Wide adoption of a standard increases the value of patents essential to implementation of the standard. If unrestricted, a firm may realize this value directly through high licensing fees or indirectly by excluding competitors from implementing the standard.

Industry groups can form standards development organizations (SDOs), which strive to reconcile the interests of intellectual property owners with the interests of others who wish to practice the standard. One common way SDOs achieve this is to require participants to disclose their patent interests in the standard and to license all patents essential to compliance with the standard on 'fair, reasonable, and non-discriminatory' terms.[14]

Technology standards promote interoperability, even in the absence of direct contact between developers. This can reduce problems of 'lock-in'. Standards may also offer an escape from the patent thicket. Technology standards with publicly known licensing terms can reduce patent thicket problems. Unfortunately, as Shapiro (2000) stresses, concerns about antitrust actions have at times left the details of these terms to be determined *ex post*, after leverage is acquired by the incorporation of the patents into the standards. Standards offer a potential for escape from the patent thicket, but standards setting bodies generally need cooperation from the antitrust institutions: *ex ante* price limitations are crucial.

Free and open standards offer these gains on a non-discriminatory basis and also provide clear *ex ante* price limitations. Free and open standards are by definition available for all to read and implement without payment. The term 'free' refers to the implied freedoms, not the price. The term 'open' refers not only to the open accessibility to the standard but also to the openness of the process, which is intended to preclude the promotion of the market power of specific vendors or groups.[15]

The benefits of open standards to industry participants, especially new entrants, are clear, but economic analyses of the incentives to actively participate in open standards bodies remain incomplete. Economists need a fuller understanding of the incentives of intellectual property holders to initiate and participate in the development of open standards, which appears to involve the private provision of a public good. If a given industry is developing open standards, a firm may of course choose active participation as a way to ensure alignment of its development process with emerging standards. This suggests that open standards movements become industry dominant if they gain 'critical mass'. In any case, is clear that open standards at times garner widespread industry support.

A well-known example of a successful open standards body is the World Wide Web Consortium (W3C), the preeminent open standards body for the Internet. Hundreds of organizations participate in the development of interoperable technologies (specifications, guidelines, software, and tools). For example, the W3C's (hypertextmarkup language) and CSS (cascading style sheets) standards affect anyone who 'surfs' the web. The W3C is also notable for its efforts to remain independent of specific vendors, who may try to co-opt an open standard through 'embrace and extend' tactics. The W3C patent policy is that its specifications must be implementable on a royalty free basis. The specific mission of the W3C Patent Policy Working Group concerns 'the growing challenge that patent claims pose to the development of open standards for the Web'. This open standards body has so far had remarkable success despite apparent vendor efforts to co-opt the standards. The widespread adoption of W3C standards has ensured a remarkable level of interoperability on the internet and has supported an explosion of competing, standards compliant technologies.

Free and open development

To economists, explaining free and open (FO) development is even more challenging than explaining open standards movements. Once again the apparently simple modifier 'free and open' proves quite complex, but we shall focus on a few key features of FO development. In FO development, enabling disclosure is readily available, and licensing to use, redistribute and modify the technology is provided gratis by the developer. (Distribution of modified technology may be restricted to ensure that the modification also remains free and open.) Technology placed in the public domain obviously satisfies these requirements.[16] However copyrighted or patented technology can also be free and open.

When FO development succeeds in producing rapid innovation, this challenges the presumptions of many intellectual property arguments. Rosenberg (1976) documents FO development in the machine tool industry, von Hippel (1988) in the scientific instrument industry and Allen (1983) in the iron industry. An interesting feature of these cases, which appears typical of FO development, is that users of technologies were actively involved in the innovation process.

The most famous FO development effort has taken place within the software industry. Software development is considered to be free and open only if the source code is readily available and freely redistributable.[17] Software produced by FO development is generally referred to as free and open source software (FOSS). FOSS development does not commit anyone to distributing source code for free; this would be inappropriate given that all known distribution methods are resource using. It does, however, mean that there are no legal restrictions on the redistribution of the unmodified open source software to others, and in practice it has meant that FOSS software has generally been provided for download without charge.

Software could be open source but adhere to few standards, open or otherwise. However, most well-known open source software projects stress adherence to open standards. To the puzzlement of many economists, this combination of open standards and open source development has generated tremendous economic value. Particularly famous open source software applications include the Linux operating system, the Apache web server, the MySQL relational database, the sendmail mail transfer agent, the Mozilla web browser, and the interpreter for the Perl programming language. These applications are all in wide use and are considered highly competitive with commercial products. They demonstrate unambiguously that high-quality, commercially important, and very innovative development can take place in the apparent absence of revenue-generating patent rights.

We shall use the term 'the open source phenomenon' to refer loosely to the success of FOSS development in producing economic value despite stringent limits on the ability of individuals and firms to appropriate the value that is being created. Economists recognize that explaining the open source phenomenon is an important project. We consider some explanations in the next section. Of immediate interest is that FO development offers clear potential for escape from the patent thicket.

A necessary condition for software to be considered 'open source' is that users and developers have the right to modify the code for their own use. Firms can therefore shield themselves from patent surprises by relying internally on proprietary or open modifications of free and open software.[18] To the extent that it can remain self-contained, FOSS development clearly eliminates the patent thicket problem.

While modified FOSS applications can be 'freely' distributed – that is, without a royalty or fee paid to the innovators of the original software program – these distributions are generally subject to one or more licensing agreements. A key aspect of these licensing agreements is the extent to which modifications of the code can be made proprietary. For example, the Berkeley Software Distribution (BSD) license allows the distribution of proprietary and closed modifications. The licensee is permitted to sell a modified program for profit (without having to reveal the source code of the modification). In contrast, under a General Public License (GPL), no proprietary rights can be asserted: distributed modifications must remain FOSS under the GPL.[19] In the early 1990s, the BSD license was the dominant form; at present, it is the

GPL. Lerner and Tirole (2002) indicate that use of the GPL appears to be declining in favor of licenses that permit proprietary modifications.

FOSS potentially offers firms an escape from the patent thicket. If the software is in the public domain, the firm can simply consider whether it pays to license patented technology given the FO alternative. The decision is similar with 'liberal' licenses such as the BSD license. However, the use of GPL'd software raises serious issues for the firm: it must consider whether this will require donation of innovations to the FOSS community beyond what will be repaid by the added insurance against stumbling into a patent thicket along with the ability to modify the GPL'd software for its own purposes.

Economics of free and open development

The open source phenomenon poses a puzzle. At first glance, it seems that economic reward has not been necessary to draw forth the factor inputs devoted to free and open software creation. Substantial economic value is being created despite the apparent absence of a price mechanism to direct this creation. Innovation appears to be substantial despite the apparent preclusion of appropriation of the returns to innovation.

Naturally this has drawn the attention of economists, who are particularly inclined to see prices as a necessary conduit of the information that can allow efficient resource allocation. Prices are a mechanism markets use to allocate resources among competing uses. Underpinning this is the role of property rights, which allows unambiguous reallocation of resources, goods and services. Together, prices and property rights provide incentives to manage resources in ways that produce economic value. Free and open development efforts appear at first glance to dispense with prices, markets and property rights in the process of value creation.

The present section addresses two issues. First, we consider some economic incentives that help explain and sustain FO development. In particular, we ask how FOSS is able to produce substantial economic value, apparently without the guidance of prices and intellectual property rights. Second, we ask whether FO development and strong IPR institutions (particularly patent rights) are mutually incompatible. In particular, we ask under what conditions IPRs adversely affect FOSS activity and under what conditions they might be complementary. While the focus of this discussion is the software industry, we attempt to draw some general lessons for FO development.

The FOSS Community

Economists are still struggling to understand the FOSS community. It is not at all clear that a single model of FO development will prove relevant to this varied community, which comprises private individuals, standards organizations, and firms of diverse sizes – all with varying degrees of involvement in FOSS development. In addition, governance structures vary from community to community. Some FOSS projects are run by strong, centralized leaders (as in the Linux project), while others are run by committees (as in Apache). Nevertheless some generalizations seem possible.

As Lerner and Tirole (2002) emphasize, FOSS innovations have not been just in the products themselves but also in the development process. FOSS development tends to take place in a collaborative organizational structure, and technically sophisticated users often provide important impetus for incremental FOSS innovation.

FOSS innovation is – and has largely been – a private sector initiative. Commercial firms – both start-ups like Red Hat Linux and behemoths like IBM – have been involved at many different levels. Private agents produce innovations using a mixture of private inputs (such as programmers' labor) and public goods (such as the public domain stock of technical knowledge). The FOSS projects are not state owned or conducted by state-owned enterprises, although they occasionally receive modest state subsidy.[20] Since FOSS products are understood to produce positive externalities, such subsidies might be amenable to traditional economic justifications. The depth of this justification is complicated when FOSS development directly competes with profit-oriented development, and it remains an active debate among policy makers and economists.

FOSS incentives

In FO development, it often seems that highly experienced labor works for free in order to provide sophisticated technical innovations at no charge. Economists are naturally interested in understanding why resources are being provided without compensation. Various proposals have been offered. Altruism is often mentioned both within the FOSS community and by observers, but it is a motivation that does not sit well with most economists. Economists generally seek more traditional motives: direct or indirect economic advantages gained by participation in FO development.

One proposal focuses on the role of users in improving extant technology. Incremental innovations that can be made at low cost by the user may have a substantial higher private valuation. For example, tinkering may pay off directly in the usability of a software application that has already been adopted. Of course, unless it is a GPL'd application, this does not explain why the innovator would eschew the assertion of property rights in the innovation. However, if one programmer's contribution leads others to invest in the project, his private valuation could increase (due to network effects) or costs decrease (due to a productivity effect owing to a higher stock of solved problems). This suggests that multiple equilibria are possible, depending on the conjectural variations held by individual programmers. Game-theoretic aspects are thus potentially important in understanding FOSS incentives.[21]

A different explanation is offered by Lerner and Tirole (2002), who focus on career signaling and peer recognition effects. A programmer's contribution could enhance his reputation or peer recognition and thus lead to better future job offers or to venture capital. This suggests that programmers have a 'signaling' incentive to contribute to an open source project. It may be easier to signal in an open source project than in a proprietary project because the open source projects offer greater visibility.

Oddly, Lerner and Tirole discount the usefulness of altruistic and intrinsic motivations while citing an interview with Linux developer Linus Torvalds that highlights these (Torvalds 1998). Torvalds in contrast discounts career signaling and peer recognition effects, saying 'it feels good to have done something that other people enjoy using'. Torvalds even addresses compensation directly: 'I want to continue to try to avoid making money directly off Linux – that keeps me focused on purely technical issues with the Linux kernel'. This interview is remarkable because the interviewer pushes hard to pinpoint incentives to produce free software that an economist can fathom, yet he ends up with responses like: 'I really don't think you need all that much *quid pro quo* in programming – most of the good programmers do programming not because they expect to get paid or get adulation by the public, but because it is fun to program' and 'The first consideration for anybody should really be whether you'd like to do it even if you got nothing at all back'. A tendency to discount such altruistic and intrinsic motivations is common in the economics literature. Exceptions include Osterloh et al. (2003).

An interesting recent development is the contribution of large, established corporations to open source products. For example, the Open Source Development Lab has investment backing from numerous large firms, including Computer Associates, Fujitsu, Hitachi, HP, IBM, Intel and NEC. IBM in particular has been strongly backing Linux as a venue for selling services, applications and hardware. While important individual participants in FOSS development do seem to contribute in response to altruistic and intrinsic motivations, other motivations will be needed to explain the involvement of commercial firms.

If a user community is attracted to open source software, possibly for reasons of security or perceived quality, then FOSS development may still provide commercial interests with lucrative soil in which to grow. For example, it may be possible to profit by providing complementary products and services including accessories (such as computer manuals or utility applications). The strategy of 'giving away the razor in order to sell blades' is a common business practice in other industries, and it may go a long way in explaining the contributions of commercial firms to FOSS development. Perhaps the best-known example of this is Red Hat, which charges for technical support on Linux-based products.

There can also be strategic considerations. Firms trying to free themselves from Microsoft's near monopoly on desktop operating systems may see a strategic advantage in promoting Linux development. Some governments as well have clearly acted out of such strategic considerations. Another possibility is that improvements in an FO product might enhance the market position of a related proprietary product. For example, Mustonen (2002) develops a model in which a monopoly producer of a proprietary software application supports GPL programs in order to achieve compatibility between its proprietary product and products that it cannot produce by itself. The monopolist gains from this strategy if sufficient network effects exist in the consumer market, so that the profits from compatibility exceed the loss in market share. Another possible motive commercial firms have to support FO development is to hurt firms that produce competing products. In contrast with predatory pricing, however, contributions to FO development cannot be repriced after successful predation. Even when there are no direct gains from freely revealing innovations, contributors to FO development may benefit if by revealing information they induce the development of desired complements to their proprietary production activities (Harhoff et al. 2002).[22] There are such a variety of reasons why the disclosure and free distribution of innovations might economically profit workers and firms that the research task of determining the most economically important is sure to prove challenging.

FO development versus IPRs?

The next set of issues concerns the relationship between IPRs, particularly patent rights, and FO development. We first discuss arguments for and against the proposition that patent rights threaten the open source movement. We then discuss how open source communities may employ the existing system of IPRs to protect their system of innovation.

Patent rights are a threat If patent rights increase the opportunity cost of participating in FO development, innovation may slow in the FO sector. If a rise in the strength of patent protection increases the returns to proprietary innovations, resources should shift away from the open source sector to the proprietary sector. Another concern is that a (non-GPL'd) project could be 'hijacked' if it were diverted into a proprietary commercial venture. Some people feel that this is what happened when AT&T began to assert copyrights over the use and distribution of Unix, after years of development in cooperation with universities and other research organizations. Another famous but less dramatic example occurred when the X Consortium, after years of encouraging volunteers making software submissions to reject the GPL, actively considered ending the free software status of the X Window System.[23]

A more substantial way in which patenting activities can harm open source activities is by fencing off certain software knowledge capital, preventing the exploitation of useful programming innovations. In addition, because the disclosure standards for software patents may not be fully enabling, access to the blueprint underlying the technical effects of software programs may be obstructed. Some firms for strategic purposes may hoard patents, never commercializing them, so as to prevent rivals from marketing close substitutes. Thus, open source projects may themselves be blocked by patent rights. The proprietors of patented technologies might demand royalties or fees that are too high (relative to the private benefits of the project) or refuse licenses altogether for strategic reasons.

A self-contained FO development project may offer a refuge from patent thickets and patent blocking. However, FO projects may find themselves unable to operate in a self-contained fashion. An FO development project may therefore find itself undermined by patent thickets. Just as with traditional development models, technology essential to a project may have been patented, and this may not be discovered until the FO development has consumed considerable resources. In sectors inclined to patent thickets, FO development faces difficulties similar to those faced under ordinary proprietary development. This shows up in struggles to maintain the integrity of the FO development process. That even FO development may face patent thicket problems may provide further support for the claim that patents are 'not appropriate for industries … in which innovations occur rapidly, can be made without a substantial capital investment, and tend to be creative combinations of previously-known techniques', as Oracle Corporation put it in a famous 1994 statement to the PTO.

Patent rights are not a threat The view that patent rights can readily coexist with FO development rests primarily on the case that the two approaches (proprietary versus free and open) have technologically separated niche markets. A related argument is that the two approaches specialize in different kinds of software technologies.

The 'niche' argument is that FOSS innovators develop customizable 'expert-friendly' products while proprietary innovators develop products for the mass market. There are several reasons for this. The programmer who participates in a FOSS project tends to be an expert at fixing bugs and adapting a program to suit his needs. The open disclosure of source code enables him to do that. The same programmer, as many have noted, has no marketing-oriented incentive to write user-friendly technical manuals. (Of course these could be written for profit under copyright protection.) Moreover, Bessen

(2001) argues that as software becomes more complex, debugging costs increase exponentially. Bessen argues that proprietary developers will therefore sacrifice complex, customizable features in favor of standardized products. Moreover, customizable products represent a small share of the market for any proprietary producer. Hence the open source community provides opportunities for particular consumers to customize the products themselves.

Evans and Reddy (2002) and Schmidt and Schnitzer (2003) discuss whether the lack of a profit motive lies behind the reason why FOSS developers do not produce goods for the mass market. The proprietary developer has incentives to undertake market research to figure out the needs of the consumers (particularly the least computer sophisticated consumers). Thus, they argue, proprietary producers better serve the end-user market, while the FOSS community better serves information technology professionals and other sophisticated users. We agree that the open source community has been extremely successful in producing expert-friendly software goods (such as operating systems, servers and programming tools) while proprietary producers have had greater success in producing user-friendly desktop applications. However, there are exceptions. FOSS development has turned more recently to the provision of user-friendly desktop applications, the best known of which is probably the OpenOffice office suite, and a variety of proprietary operating systems play important roles in many markets.

Intellectual property defenses The previous discussion focused on whether proprietary development under patent protection is likely to undermine or complement FO innovation. Patent rights appear less problematic for FO innovation when there is greater product differentiation between proprietary producers and FO developers or when FO developers work on innovations that are not patentable. When FO innovations are close substitutes for proprietary innovations, strengthened patentability appears likely to undermine FO development.

We now consider how FO communities use intellectual property laws to protect their system of innovation. For example, a natural defense against future patent thickets and blocking is to establish a robust set of documented prior art. This is widely recognized among FOSS developers, but there is disagreement about the best approaches to establish prior art. Some argue that invalid patents are being granted because of the inadequacy of the patent databases, which in turn is rooted in the extensive software development that took place before patentability, and that FOSS developers should therefore protect themselves by providing evidence of prior art to databases designed to facilitate patent search. Others argue that the opposite conclusion follows: invalid patents can be challenged, but making evidence of prior art more readily accessible also makes it easier to design a patent application that avoids claiming prior art. Patent opponents therefore have no obvious incentive to provide such assistance. The more aggressive strategy of preemptive patenting suffers from the same weakness, although it provides stronger protection of specific technology. Instead, the argument goes, the FOSS community should provide *no* assistance to patent searches beyond the full and public documentation that ensure that prior art is demonstrable. This decreases the likelihood that software-related patents can be written in ways that avoid claiming prior art, and may therefore discourage or prevent firms from patenting. Unfortunately this defense strategy could result in socially wasteful litigation.

Preemptive patenting has additional drawbacks. It can impose substantial costs on the open source community – costs that cannot be offset by license revenues. Patenting is expensive in terms of the application fees, search and examination fees, attorney fees, translation fees if filed in or from other countries, and maintenance fees (if patent holders are required to pay fees during the life of a patent to keep the patent in force). Furthermore, the patents owned by open source communities may themselves contribute to patent thicket problems. For example, if a researcher wants to build upon a patented open source innovation and develop a proprietary product, the open source community may refuse a license or issue one on restrictive terms.

Finally FOSS developers do generally rely on non-patent IPRs, such as copyrights, to protect the source code of open source projects. They also use trademarks to protect the symbols and brand names of FOSS products. O'Mahony's (2003) survey of six major FOSS projects finds that the projects utilize copyright and trademark protection, software licensing, and other legal measures such as incorporation (in order to protect collective assets). The FOSS communities do not simply forfeit their intellectual property rights: they exercise them in a specific manner. These rights protect against misappropriation and help ensure that their innovations remain free and open.

Summary

Economic exploration of the functioning of free and open development has been stimulated by economists' recognition that FO development has generated substantial economic value in the form of technological innovation and diffusion. Economists' explanations have focused on subsuming the incentives for FO development in traditional economic categories. This valuable exercise has provided many insights into economic motivations for participating in FO development, but important aspects of FO development have yet to be given traditional economic explanations.

In FO development, innovations are fully disclosed and freely redistributable. It is clear that both pecuniary and non-pecuniary rewards can motivate participants of open source projects. However the pecuniary rewards are not tied to a direct flow of licensing revenue. Innovators contribute resources to FO development without claiming revenue-generating IPRs. Modification of the innovations is unrestricted, although distribution of modifications may be restricted to keep them within the FO development community. FO development does not eschew use of the intellectual property institutions. Copyright laws, trademark rights and licensing help protect the creative assets and reputation of open source communities (against fraud, misrepresentation and 'hijacking').

The relationship between IPRs (particularly patent rights) and FO development is complex. It is conceivable that FO development could take place in a self-contained fashion, so that patent thickets elsewhere in the software industry would not hinder FO development. This is plausibly true in some niche markets. However, many FO and commercial development projects directly compete, and in these cases software patent thickets can pose quite a threat to FO development.

Conclusion

Intellectual property rights, like all property rights, are human creations. They are embodied in complex, interlocking institutions and sustained by pragmatic and metaphysical justifications. This chapter explores the IPRs embodied in patent systems. A core pragmatic justification of patents is that they foster the discovery, disclosure and dissemination of practical inventions. However, the patent system may in some cases produce complex overlapping property rights, particularly in sectors of the economy characterized by rapid, incremental and cumulative innovation. The resulting patent thickets can pose a costly barrier to the development of new technology.

Of course the phenomenon of holdup is not unique to intellectual property. Holdup also occurs in real property, as when a developer must purchase a large tract of land from different owners to produce an economic good but one of the owners holds out. Holdups are common not only when there are numerous parties involved but also when there are few parties but each has some market power.

Holdup is especially likely to be prevalent in the software industry. Software development is often a creative combination of known techniques, building on an extensive prior code base and working in concert with other programs or program components. If software patents render the code base, program components and programs all proprietary, programmers must obtain licenses from the owners of each component. This makes fertile ground for patent thickets. Moreover, software patents involve near-ideas: algorithms that produce technical effects are not always clearly separable from those that are simply mathematical algorithms. Because they protect near-ideas, software patents can potentially grant very broad powers, holding up follow-on inventions by restricting the use of near-fundamental discoveries.

There are several possible escapes from patent thickets, including cross-licensing and patent pools. Free and open standards and free and open development may also offer possible avenues for escape, and both have been used to create substantial economic value. Open innovation systems need not supplant but may operate alongside traditional patent systems. This would especially be the case for technologies that are not patentable in the legal sense (that is, in terms of novelty or non-obviousness), but there are situations where invalid patent rights (such as those for 'essential facilities' or for which prior art existed) could hold up FO development.

Despite a flurry of research activity in response to the startling successes of free and open source software, economic analyses of free and open standards and free and open development remain incomplete. Although no single explanation will encompass the diversity of free and open standards and development efforts, we suggest that free and open development holds particular promise for industries facing potentially severe holdup problems. Exploring the extent to which 'lessons' learned from free and open source software development can be applied to other industries should prove an exciting area for future research.

Notes

1. Machlup and Penrose (1950) note that many modern disputes were anticipated by the 'great patent controversy' of the 1850–1875.
2. Machlup (1958) notes a 'permanent exclusive privilege' granted in Switzerland in 1577. Machlup and Penrose (1950) observe that nineteenth-century France and Belgium produced a substantial literature arguing for 'perpetual rights in intellectual products'.
3. As Machlup (1958) emphasized, consumers must pay more for any protected innovation, whether or not that innovation was patent-system induced.
4. Machlup (1958) proposes the early automobile industry as an example. We propose other examples in our later discussion of free and open development.
5. The *Romer (1990) endogenous growth* model correctly specifies the idea versus application of ideas dichotomy. In this model, the stock of knowledge is a public good while the intermediate capital goods whose designs or blueprints are derived from research knowledge can be made proprietary.
6. In some jurisdictions, such as the United States, the inventor is given a grace period of one year to apply for a patent after a public display.
7. In other environments local telephone networks, digital subscriber lines (DSL), or central railroad switching systems might prove to be essential facilities.
8. For a history of the relationship between these two types of laws in various countries, see OECD (1989).
9. In *Gottschalk v. Benson*, the US Supreme Court invalidated a patent on a computer program to convert signals from binary-coded decimal form to pure binary form.
10. In *Diamond v. Diehr*, the US Supreme Court upheld a patent on a process for molding uncured synthetic rubber into cured products. A computer algorithm was essential to the process, since it allowed precise and timely determination of the cure time based on the temperature of the molding press.
11. Of course, source code alone can also fail to constitute disclosure, since the source can be written to intentionally obfuscate the functionality. See Lemley et al. (2003) for further discussion.
12. For example, the *Examination Guidelines for Computer-related Inventions* of the PTO

suggest that enablement could be satisfied by the patent applicant by 'outlining the significant elements of the programmed computer using a functional block diagram'.

13. See US Department of Justice/Federal Trade Commission (1995), 'Antitrust Guidelines for the Licensing of Intellectual Property', http:www.usdoj.gov/atr/public/guidelines/ipguide.htm.
14. The Joint Electron Device Engineering Council (JEDEC) failed to get adequate disclosure when it admitted Rambus Inc. to membership in 1992, and Rambus apparently modified a pending patent to encompass the standards being developed by the JEDEC.
15. A more detailed introduction can be found at http://perens.com/OpenStandards/Definition.html.
16. Inventors can always put an invention in the public domain by publicizing it while refusing to patent it, as Jonas Salk famously did with the polio vaccine.
17. The source code for a software application is human readable (with a text editor). It is a complete and implementable (for example, with a compiler) description of the software. For more detail on the definition of open source software, see http://www.opensource.org/docs/definition.php.
18. The integrity of the FOSS development process is a prerequisite for effective shielding, as illustrated by the recent suit brought by the SCO Group against IBM. The suit alleges that IBM contributed SCO trade secrets to the Linux community. IBM has denied these allegations. Many observers, noting that Microsoft recently acquired an interest in SCO, see this as an attempt to undermine the reputation of the FOSS development process in general and the Linux development process in particular. Raymond and Landley (2003) offer a very useful if perhaps tendentious analysis of the suit.
19. Implications of this may soon be tested in court. GPL considerations have become prominent in the SCO Group suit against IBM. Allegations arose that SCO had incorporated Linux code into Unix System V in violation of the GPL. Other observers claimed that since SCO had itself shipped Linux containing the allegedly proprietary code, it had effectively GPL'd the code.
20. In some regions and countries (such as the European Union, Argentina, Brazil and Peru), governments have supported open source through procurement policies (for use in government ministries or departments). Germany has been especially Linux friendly. The federal government has even directly funded improvements to Linux user interfaces. In May 2003 the city of Munich decided to switch more than 14 000 desktop computers from Windows with Microsoft Office to Linux with Open Office. In June a migration of comparable size was announced in Britain. Other countries, such as Singapore, have provided tax breaks to companies that adopt open source products (Hahn 2002, ch. 1).
21. Game-theoretic analyses of FOSS projects include Bessen (2001), Harhoff et al. (2002) and Johnson (2002).
22. Harhoff et al. (2002) describe a number of interesting examples. For instance, Technicon Corporation devised automated blood chemistry analysers. The blueprint for this equipment built on the findings of laboratory clinicians who freely revealed them via publications. Technicon thus adopted the discoveries of lab clinicians without paying any royalty to them. The lab clinics were better off nonetheless because they induced the creation of more efficient equipment that analyses blood samples; Technicon is better off because the equipment is a commercial success. Another example is the Online Public Access Catalogues (OPACs), computerized systems that enable online access to a libraries collections. Vendor sharing of OPAC source code enabled users to suggest modifications to proprietary OPAC. Manufacturers eventually adopted them, again without paying royalties to the user innovators. The manufacturers had incentives to adopt them because enough users wanted the same thing. The users in turn got a better product.
23. The outrage of the FOSS community at such developments is important to consider when assessing explanations of their activities. If career signaling and peer recognition effects were the primary considerations of these contributors, why would such *ex post* developments be viscerally resented?

References

Allen, R.C. (1983), 'Collective invention', *Journal of Economic Behavior and Organization*, **4** (1), 1–24.

Bessen, J. (2001), 'Open source software: free provision of complex public goods', Research on Innovation Working Paper <http://www.researchoninnovation.org/home.htm>, accessed 9 January 2004.

Bessen, J. and E. Maskin (2000), 'Sequential innovation, patents, and imitation', Working Paper 00-01, MIT Department of Economics, Cambridge, MA.

Boldrin, M. and D. Levine (2002), 'The case against intellectual property', *American Economic Review,* **92** (2), 209–12.

Buchanan, J.M. and Y.J. Yoon (2000), 'Symmetric tragedies: commons and anticommons', *Journal of Law and Economics,* **43**, 1–13.

Coase, R.H. (1960), 'The problem of social cost,' *Journal of Law and Economics,* **3**, 1–44.

Cohen, Wesley, Richard Nelson and John Walsh (2000), 'Appropriability conditions and why firms patent and why they do not in the American manufacturing sector', NBER Working Paper 7552, National Bureau of Economic Research, Washington, DC.

Demsetz, H. (1967), 'Toward a theory of property rights', *American Economic Review,* **57** (2), 347–59.

Evans, D.S. and B. Reddy (2002), 'Government preferences for promoting open source software: a solution in search of a problem', Working Paper, National Economic Research Associates, Cambridge, MA.

Friedman, Lawrence M. (1985), *A History of American Law,* 2nd edn, New York: Simon & Schuster.

Hahn, Robert W. (2002), 'Government policy toward open source software: an overview', in Robert W. Hahn (ed.), *Government Policy toward Open Source Software*, Washington, DC: AEI-Brookings Joint Center for Regulatory Studies, pp. 1–11.

Hall, B.H. and R.H. Ziedonis (2001), 'The patent paradox revisited: an empirical study of patenting in the U.S. semiconductor industry, 1979–1995', *Rand Journal of Economics*, **32** (1), 101–28.

Harhoff, D., J. Henkel and E. von Hippel (2002), 'Profiting from voluntary information spillovers: how users benefit by freely revealing their innovations', Working Paper, MIT Sloan School of Management, Cambridge, MA.

Heller, M.A. (1998), 'The tragedy of the anticommons: property in the transition from Marx to markets', *Harvard Law Review,* **111** (3), January, 621–88.

Hirshleifer, J. (1971), 'The private and social value of information and the reward to inventive activity', *American Economic Review,* **61** (4), 561–74.

Johnson, J.P. (2002), 'Open source software: private provision of a public good', *Journal of Economics and Management Strategy,* **11** (4), 637–62.

Lemley, Mark, Peter Menell, Robert Merges and Pamela Samuelson (2003), *Software and Internet Law,* 2nd edn, New York: Aspen.

Lerner, J. and J. Tirole (2002), 'The simple economics of open source', *Journal of Industrial Economics,* **50** (2), 197–234.

Lerner, J., J. Tirole and M. Strojwas (2003), 'Cooperative marketing agreements between competitors: evidence from patent pools', NBER Working Paper 9680, National Bureau of Economic Research, Washington, DC.

Levin, R., A. Klevorick, R. Nelson and S. Winter (1987), 'Appropriating the returns from industrial research and development', *Brookings Papers on Economic Activity,* **3**, 784–829.

Machlup, Fritz (1958), *An Economic Review of the Patent System*, Washington, DC: United States Government Printing Office, <http://www.ipmall.fplc.edu/hosted_resources/jepson/unit1/aneconom.htm>, accessed 9 January 2004.

Machlup, F. and E. Penrose (1950), 'The patent controversy in the nineteenth century', *Journal of Economic History,* **10** (1), 1–29.

Mustonen, M. (2002), 'Why do firms support the development of substitute copyleft programs?', Department of Economics Working Paper, University of Helsinki.

O'Mahony, S. (2003), 'Guarding the commons: how community managed software projects protect their work', *Research Policy,* Special Issue on Open Source Software Development, forthcoming.

Oracle Corporation (1994), 'Patent Policy', as reprinted by Gordon Irlam at <http://www.base.com/software-patents/statements/oracle.statement.html>, accessed 9 January 2004.

Organization for Economic Cooperation and Development (OECD) (1989), *Competition Policy and Intellectual Property Rights*, Paris OECD.

Osterloh, M., R. Sandra and K. Bernhard (2003), 'Open source software production: climbing on the shoulders of giants', Working Paper, Institute for Research in Business Administration, University of Zurich.

Patent and Trademark Office (PTO), US Department of Commerce (1996), *Examination Guidelines for Computer-implemented Inventions*, Washington, DC, <http://www.uspto.gov/web/offices/com/sol/og/con/files/cons093.htm>, accessed 9 January 2004.

Raymond, E. and R. Landley (2003), 'OSI position paper on the *SCO vs. IBM* complaint', http://www.opensource.org/sco-vs-ibm.html, accessed 9 January 2004.

Romer, P. (1990), 'Endogenous technological change', *Journal of Political Economy,* **98** (part 2), S71–S102.

Rosenberg, Nathan (1976), *Perspectives on Technology*, Cambridge, UK: Cambridge University Press.

Schmidt, K. and M. Schnitzer (2003), 'Public subsidies for open source? Some economic policy issues of the software market', Discussion Paper 3793, Centre for Economic Policy Research, London, UK.

Shapiro, C. (2000), 'Navigating the patent thicket: cross licenses, patent pools, and standard setting', *Innovation Policy and the Economy,* **1**, 119–50.

Stallman, Richard (2000), 'The Anatomy of a Trivial Patent', <http://lpf.ai.mit.edu/Patents/anatomy-trivial-patent.txt>, accessed 9 January 2004.

Thomas, J.R. (2002), 'Liberty and property in the patent law', *Houston Law Review Symposium,* **39** (3), 569–619.

Torvalds, Linus (1998), 'Interview with Linus Torvalds: What motivates free software developers?', <http://www.firstmonday.dk/issues/issue3_3/torvalds/index.html>, accessed 9 January 2004.

von Hippel, Eric (1988), *Sources of Invention*, Oxford, UK: Oxford University Press.

Weber, Max (1923), *Wirtschaftsgeschichte*, New Brunswick, NJ: Transaction. References are to the 1981 reprint of 1927 translation, published by Greenberg as *General Economic History*.

World Intellectual Property Organization (WIPO) (1998), *Intellectual Property Reading Material*, Geneva: WIPO.