HARVARD | BUSINESS | SCHOOL

Modularity for Value Appropriation – How to Draw the Boundaries of Intellectual Property

Joachim Henkel
Carliss Y. Baldwin

Working Paper

11-054

Modularity for Value Appropriation –

How to Draw the Boundaries of Intellectual Property

November 2010

Joachim Henkel[1], Carliss Y. Baldwin[2]

Abstract

The existing theory of modularity explains how modular designs create value. We extend this theory to address value appropriation. A product or process design that is modular with respect to intellectual property (IP) allows firms to better capture value in situations where knowledge and value creation are distributed across many actors. We propose a theory of IP modularity based on value maximization net of transaction and agency costs. We then use case examples to extend the theory into practical settings and derive strategic recommendations and empirical predictions.

Keywords: Modularity, value appropriation, intellectual property, open innovation, design

[1]Technische Universität München, Arcisstr. 21, 80333 Munich, Germany, henkel@wi.tum.de.
[2]Harvard Business School, Soldier's Field, Boston, MA 02163, USA, cbaldwin@hbs.edu.

We are grateful to many people who shared their insights or commented on earlier drafts of the paper. Special thanks go to Sung Joo Bae, Timo Fischer, Deborah Gray, Marc Gruber, Kathrin Henkel, Mako Hill, Eric von Hippel, Michael Jacobides, Andrew King, Karim Lakhani, Deishin Lee, Alan MacCormack, Matt Marx, Phanish Puranam, Manuel Sojer, Victoria Stodden, Mary Tripsas, David Upton, Gianmario Verona and three anonymous reviewers for the Academy of Management Meeting. Thanks also to participants in seminars and workshops at Bocconi University, Harvard Business School, Imperial College Business School, London Business School, MIT, and Ross School of Business. Errors and oversights are ours alone.

INTRODUCTION

Distributed value creation through innovative product and process designs is a key fact about the modern world. Today, technologically sophisticated firms must expect innovations to come from all around: from customers; from suppliers; from complementors; and from sources so far entirely unrelated to them. In turn, a firm's inventive output may be built upon by various outside innovators. Chesbrough (2003) appropriately termed this phenomenon "open innovation."

For such decentralized innovation and value creation in large systems, it is nearly a prerequisite that parts are independent in the sense that an innovation in one part of the system will not require changes in many other parts. This condition is captured in the concept of modularity (e.g., Baldwin and Clark, 2000, 2006; Garud and Kumaraswamy, 1995; Langlois, 2002; Sanchez and Mahoney, 1996; Schilling, 2000). Modular systems are made up of components that are highly interdependent within sub-blocks (called modules) and largely independent across those sub-blocks (Baldwin and Clark 2000).

While open innovation and modularity can boost value creation, they may create serious challenges for capturing value. In order to draw in external contributors, an innovator often grants them access to knowledge covering a product or process (e.g., Foray, 2004: 165; Harhoff, Henkel, and von Hippel, 2003; Henkel, 2006; Pénin, 2007; von Hippel and von Krogh, 2003). The innovator thus gives up some of its intellectual property (IP), by waiving legal exclusion rights or by revealing formerly exclusive knowledge. But as a result, contributors may appropriate a large share of the jointly created value.

This paper addresses the above tension between creation and appropriation of value. It shows how to manage a system's modular structure in conjunction with its IP so as to reconcile opportunities for delegated and distributed innovation with a firm's need to capture value. We develop our argument by first laying out a theoretical framework and a model and then using case examples to derive a set of strategic recommendations and empirical predictions.

A central element in our argument is that the structure of a technical system is in large measure under the control of the system's architects. Therefore, the technical architecture of a product or process and a firm's strategy for creating and claiming value are inevitably intertwined. In particular, in a modular system, a firm must simultaneously decide on the technical boundaries of the modules and the IP deployed in each one. Controlling too much of the system's IP is problematic if it deters innovation by others. But controlling too little—or the wrong parts—may prevent the focal firm from capturing value for itself. Below we explain how an "IP-modular" system architecture can be designed to reconcile distributed value creation with the firm's goal of capturing value.

As a motivating example, consider the example of M-Systems, a subsidiary of the leading seller of flash memory cards, SanDisk. In 2005, the open source operating system Linux was becoming popular for mobile devices. Buyers of flash memory increasingly demanded that the memory's driver (the software that allows it to operate within the mobile device) be made publicly available as open source software to simplify development of the host systems (Kaplan, 2006; Käs, 2008: 140). However, M-Systems wanted to protect sensitive IP which was bundled with the driver. As a solution, M-Systems introduced what we call IP-modular architecture: They placed the flash management code on the physical memory device itself, thus keeping it protected via copyright and secrecy (Kaplan, 2006). They published the remaining "thin" driver as open source software. Redesigning the technical modules to be congruent with the desired structure of IP allowed M-Systems to reconcile customer demands for openness with its own value appropriation.

IP modularity, that is, the congruence between the technical and IP structure of products and processes, is not a new strategy, as we shall show. However, we believe that it is increasingly important for managers, for two reasons. First, the management of formal IP in the form of patents and copyrights is an increasingly important aspect of the management of innovation. But at the same time, there is movement in the opposite direction: open and semi-open innovation processes are steadily gaining ground in a number of industries (e.g., Chesbrough, 2003; Henkel, 2006; West, 2003). Strategies based on IP modularity can help the firm to reconcile these opposing trends.

Our study contributes to three strands of prior literature in strategy. First, it contributes to the resource-based view (RBV) of competitive advantage (Barney, 1991; Rumelt, 1984; Wernerfelt, 1984) by showing how firms can protect unique knowledge resources in alliance networks and business "ecosystems." Second, it contributes to the literature on "profiting from innovation" (Chesbrough and Teece, 1996; Teece, 1986, 2000) by spelling out conditions for appropriating value in open innovation processes. Finally, it contributes to the management-related theory of modularity by showing that modularity not only relates to value creation, but also to value appropriation.

The rest of the paper is organized as follows. In the following section, we review the literatures on modularity and value appropriation, and explain how this paper relates to prior work. Next, we lay the foundation of our analysis, formally defining the concepts of "IP status," "IP incompatibility," and "IP modularity." We go on to construct a simple decision model to show when and why IP modularity is preferred in the presence of opportunistic behavior and resulting transaction and agency costs (Williamson, 1979, 1985; Jensen and Meckling, 1976). We then present ten strategic recommendations with supporting case examples; these can serve as the basis for empirical tests. The final section provides a discussion and conclusions.

LITERATURE REVIEW

In this section we review those bodies of literature that are fundamentally related to IP modularity. We first review the management literature on modularity, open innovation, and value creation, then the resource-based view (RBV) of competitive advantage (Barney, 1991; Rumelt, 1984; Wernerfelt, 1984), and then the literature on "profiting from innovation" (Chesbrough and Teece, 1996; Teece, 1986, 2000). By and large, the prior strategy literature assumes that the structure of technical systems is fixed and thus not a target of strategic choice. This paper differs from prior work in that it envisions both modularization of the technical system and the partitioning of intellectual property as strategic possibilities and shows how they are related.

Modularity, Open Innovation, and Value Creation

Simon (1962: 477) was the first to note that modularity reduces the cost of managing the cognitive burden of complexity. In addition, modularity allows tasks to be partitioned (Gomes and Joglekar, 2008; von Hippel, 1990) and worked on in parallel (Baldwin and Clark, 2000). With task partitioning, a productive division of labor may arise between individuals within one firm (Sanderson and Uzumeri, 1995) or across firm boundaries (Baldwin and Clark, 2000; Langlois and Robertson, 1992; Sanchez, 1995; Sanchez and Mahoney, 1996; Staudenmayer et al., 2005). The existence of task modules with well-defined interfaces reduces transaction costs, thus supports outsourcing (Baldwin, 2008; Fine, 1998; Hoetker, 2006; Jacobides, 2005; Sturgeon, 2002). Changes within modules and recombinations of modules are easy given a modular architecture. (In contrast, a change of the architecture itself is difficult (Henderson and Clark, 1990).)

Importantly in our context, modularity in the design of products and processes facilitates the division of labor in the innovation process between firms (Baldwin and Clark, 2000; Langlois, 2003; Sanchez and Mahoney, 1996)—an approach aptly labeled "open innovation" by Chesbrough

(2003). In a modular system, innovators do not have to be lodged within the firm that created the modular architecture. Innovations can be introduced by firms in related markets, entrepreneurial startups, or even users (Baldwin and Clark, 2000, 2006; Franke and von Hippel, 2003; Jeppesen, 2004; von Hippel, 2001; von Hippel and Finkelstein, 1979).

However, the fact that modularity enables innovation “at a distance” from the focal firm creates implicit tension between value creation and value appropriation. The architect of a modular system risks losing the lion’s share of value to complementary innovators. For example, the modular architecture of IBM’s mainframe computer System/360 allowed the makers of plug-compatible peripherals, such as disk drives, to enter the market (Baldwin, 2008; Baldwin and Clark, 2000). The IBM PC had a highly modular architecture, but in the end most of the value was captured by Intel and Microsoft, not IBM (Ferguson and Morris, 1993). Yet MacCormack and Iansiti (2009) also point out that creating a modular architecture, with clear separations between interfaces and code, can facilitate value appropriation by allowing the firm to make interfaces public while hiding the internal code.

Almost without exception, the prior literature on modularity focuses on how modular architectures *create value*, in particular in the context of open innovation. When *value appropriation* is considered, it is in the context of stories, such as System/360 or the IBM PC. To our knowledge, this is the first systematic analysis of how firms can use modularity (in conjunction with IP) to appropriate value.

Value Appropriation

Two strands in the modern strategy literature address value appropriation directly. First is the resource-based view (RBV) of competitive advantage, whose modern formulations are due to

Rumelt (1984), Wernerfelt (1984), Barney (1986, 1991), Peteraf (1993) and others.[1] Second is the literature on how firms profit from innovation, which originated with Teece (1986).

The RBV holds that competitive advantage derives from a firm's ownership or control of "assets, capabilities, organizational processes, firm attributes, information, knowledge, etc." (Daft, 1983, quoted by Barney, 1991:101) that are valuable, rare, inimitable and non-substitutable (Barney, 1991). The ensuing rents must be protected by "isolating mechanisms" (Rumelt, 1984: 567) or "resource position barriers" (Wernerfelt, 1984: 172). When the resource is knowledge, state-sanctioned property rights (patents and copyright) and/or secrecy serve as isolating mechanisms or position barriers that enhance the focal firm's bargaining power, and allow it to appropriate the value of its knowledge resources (Lavie, 2007; Reitzig and Puranam, 2008).

The traditional RBV does not envision a world in which firms are "interconnected" and might gain by sharing or giving away knowledge to others (Lavie, 2006). Yet the growing importance of alliance networks (Gulati, 1998, Lavie, 2006) and business "ecosystems" (Iansiti and Levien, 2004, Adner and Kapoor, 2010a, b) indicates that knowledge sharing can be advantageous both for individual firms and groups of firms. But at the same time, a firm with no protected resources can only earn competitive returns, not superior returns.

Such cases call for an intermediate strategy in which some knowledge is shared while other knowledge remains protected. Modularity makes it practicable to partition knowledge into discrete "chunks," which in turn receive different IP treatments. On this view, module boundaries serve as "partitioning mechanisms," dividing the firm's knowledge into discrete subsets. Some knowledge

[1] There has been some discussion in the literature about if and to what extent the RBV addresses value appropriation. We follow more recent contributions stating that much of the RBV is indeed about value appropriation (e.g., Coff, 1999; Makadok and Coff, 2002).

has relational value (Dyer and Singh, 1998; Lavie, 2006), hence should be shared. Other knowledge has positional value (Wernerfelt, 1984), hence should be protected via IP rights and/or secrecy.

A second strand in the strategy literature asks the question how firms profit—or fail to profit—from innovation (Teece, 1986). Teece and others make a distinction between "autonomous" innovations, which can be introduced without major modifications to other parts of the system, and "systemic" innovations, which require changes to many other parts (Chesbrough and Teece, 1996; Langlois and Garzarelli, 2008; Teece, 1988, 2000). In effect, autonomous innovations arise within the modules of a modular system, whereas systemic innovations arise in integral (i.e., non-modular) systems. According to Teece (1988, 2000), the value of autonomous innovations can be captured by small firms, while systemic innovations are best pursued by large companies with complementary assets and capabilities and financial resources.

In this literature, "autonomous" and "systemic" are taken to be essential, unchangeable properties of an innovation. In contrast, the literature on modularity offers many examples where it is possible to change the structure of the system to make it more modular (subject to autonomous innovation) or integral (subject to systemic innovation). However, modularizations may be problematic for value appropriation. If the focal firm modularizes a product or process, then, according to Teece (1988, 2000), it will be possible for smaller firms to enter, innovate on modules, and compete away rents. (This is essentially what happened with IBM's System/360.) Why would a firm want to open itself up to this type of competition?

Taken as a whole, the prior literature on value appropriation seems to imply that modularizations will be problematic for value appropriation. In the context of the RBV, modularity can be seen as weakening the focal firm's "isolating mechanisms" or "resource position barriers." In the Teecian context, modularity changes the nature of innovations from "systemic" to "autonomous" and thus allows smaller firms to compete for innovation rents. However, the case of M-Systems and other cases discussed below show that, when combined with "IP modularity," product and/or process modularity may allow the focal firm to appropriate more value than it could obtain with a non-modular architecture. Below we derive necessary and sufficient conditions for this to occur.

Of course a firm's control over its own architecture is never perfect: there are always constraints and trade-offs to be considered. Indeed, it is precisely the fact that there are trade-offs that makes the concept of IP modularity, introduced in the next section, economically and managerially significant.

INTRODUCING IP MODULARITY: DEFINITIONS

We begin this section by introducing the notion of "antagonistic coupling" or "incompatibility," which is central to the concept of IP modularity. We then formalize the concept of "IP modularity."

Antagonistic Coupling: Incompatibility

Just as it has focused on value creation, the prior literature on modularity has focused almost exclusively on linkages—often called "couplings"—that encourage designers to place elements *closer together*. Typically, if design decisions with respect to element X depend on decisions with respect to Y or vice versa, it is recommended to place X and Y in the same module.

Conceptually, however, it is possible for X and Y to repel one another, in which case it is good practice to place the two elements in *separate* modules. This is what often happens with IP. Consider the following example. A software developer obtains code from two sources: from a commercial supplier, under a typical proprietary license; and from an open source repository, licensed under the General Public License (GPL). The proprietary license prohibits passing on the human-readable source code to any third party. The GPL stipulates that any user of a program derived from GPL-licensed code is entitled to obtain the source code, read it, modify it, and pass both source and binary code on to others (Free Software Foundation, 1991). Interweaving the commercially licensed code and the open source code then leads to conflicting legal requirements. Thus, the two licensed-in blocks of code exhibit, from an IP perspective, an antagonistic coupling, which we call "*IP incompatibility*." At the very least, the two blocks must not be placed in the same module.

The above example involved "*incoming" IP*, that is, IP obtained under license from others. But there can also be incompatibility in the firm's own "*outgoing" IP*. Recall the example of M-Systems described in the Introduction. The flash management software needed strong IP protection (realized by copyright and secrecy) to minimize the risk of imitation. But the driver code needed to be open—accessible to those who would integrate the flash memory into their devices. Thus, the two parts of the system exhibited incompatibility with respect to IP: the desired IP status for one was incompatible with the desired IP status for the other.

With these examples in mind, we can formulate the following general definition: *Antagonistic coupling, or incompatibility, arises between two elements when some type of interaction requires or strongly advises that they be treated in different ways.* Placing them in separate modules will be necessary, and often also sufficient, in order to allow for such different treatment.

We are now in a position to introduce "IP modularity" formally.

Formalizing the Concept of IP Modularity

To fix ideas, we begin this section with a historical vignette. In the eighteenth century, the Saxony ruler Augustus the Strong obtained a monopoly on European porcelain by the simple expedient of imprisoning the inventor of the process in a fortress in Meissen. But when the inventor was close to death, the ruler sought to protect his valuable IP by resorting to another age-old strategy—*divide et impera*, or divide and rule. He instructed the inventor to tell his secrets to two men. One learned how to make porcelain paste; the other porcelain glaze. When the inventor died, no one person knew everything about how to make Meissen porcelain products. The ruler's monopoly was safe for the time being (Gleeson, 1998).

The above is a classic example of the construction of what we are calling an "IP-modular" architecture, designed for the purpose of appropriating value (protecting the ruler's monopoly). We will use this vignette to introduce the concept of IP modularity. The central points of our theoretical framework are the following. Complex systems consist of linked elements, each of which carries a certain "IP status." The IP status of two elements may be incompatible in the sense described above. To accommodate such IP incompatibility, the respective elements should be placed in different modules. Doing so is possible since designers have a choice in defining module boundaries within the system. The resulting "IP modularity" of the system is conducive to the focal firm's value appropriation.

We expand on this outline in the paragraphs below using the Meissen case to make our arguments more concrete.

The system as a network of linked elements. The concept of IP modularity applies to *complex products or processes*, which consist of several coherent parts, called *components*. In our Meissen example, the making of porcelain paste, the shaping and firing of porcelain objects, and the glazing of the objects are all components (or steps) of the porcelain process. Each component in turn will be made up of smaller components—these would be the subprocesses and individual steps of the porcelain-making recipe. The components and their dependencies can be represented as a network, as illustrated in Figure 1. The linkages indicate dependencies of the form "a change in one component may require an adaptive change in some other component."

---- Insert Figure 1 about here ----

Complementarity between components. Essentially all components must be present for the system as a whole to have value. In particular, in a process like porcelain making, all steps must be carried out, hence, they are strong complements to each other. This was also true for M-Systems's flash management software and the corresponding "thin" driver. In general, as we shall see, strategies based on IP modularity rely on strong complementarity between the components of the system. The value of a product or process as a whole is much greater than the sum of the values of its parts.

Choice in defining module boundaries. For purposes of carrying out the work, the different components may be separated and segregated, and there are different ways of doing so. Different patterns of linkage entail different system architectures (Ulrich, 1995, Baldwin and Clark, 2000). In porcelain-making, for example, the unglazed objects could be made in one location by one set of workers, and the glazing carried out in a different location by a different set of workers. The segregated components are "modules." Note that the process of porcelain making, in common with many other products and processes, did not *have* to be organized in a modular fashion, but it *could* be carried out that way. The degree of modularity in the process architecture and, especially, the location of module boundaries were to a good extent under the control of the process designers. For illustration, Figure 1(a) exhibits an integral system turned into a modular system (b), during which process one dependency is removed and another added.

Knowledge and IP corresponding to each module. In porcelain making, not only were the steps separable, but the knowledge relating to the early and late stages of the process was separable as well. Knowledge of how to mix, shape, and firm porcelain paste did not give one knowledge of how to mix and apply glazes, and vice versa. Therefore, knowledge could be split up in ways that corresponded to the underlying modular structure of the product or process. When, furthermore, knowledge is covered by IP—because the owner has legal or de facto means of excluding others from using it, as Saxony's ruler had—then the relevant IP can also be split up according to the underlying modular structure.

A module's IP status. The IP covering a particular module defines what we call the module's "IP status." By this term we mean the legal rights to and de facto accessibility of knowledge about the module and its role in the surrounding system. In our porcelain example, the knowledge related to the "porcelain paste" module was awarded the IP status "secret to the world, known only to person A," while the "porcelain glaze" module was given the distinct status "secret to the world, known only to person B."

Who decides on the IP status of a given body (or "chunk") of knowledge? Here we must distinguish between *outgoing* IP and *incoming* IP. With *outgoing* IP, the focal firm owns the IP under consideration, and is free to award to each element the IP status it finds most advantageous. Thus it can grant to other agents bundles of IP rights and levels of de facto access, which may differ across both elements and agents. With *incoming* IP, in contrast, the IP under consideration is owned by some party external to the focal firm. The owner of the IP has determined the IP status of the knowledge under consideration, and the focal firm must accept the bundle of rights and possibilities it has been granted.

IP incompatibility. As explained above, the IP rights or de facto ability to use different chunks of knowledge may be inconsistent. Such inconsistency may arise, for incoming IP, because of legal or contractual conflicts (e.g., two licenses specify different terms of use), or, for outgoing IP, as a matter of strategy (e.g., the owner of the IP wants to share one part while keeping the other part proprietary). In the Meissen case, the ruler introduced IP incompatibility as a matter of strategy.

IP modularity. A particular module of a larger system is “IP-modular” if all of its elements have the same or compatible IP status. The module in question is then an “IP module.” If all of the modules of a system are IP modules, we say that the system as a whole is “IP-modular.” Importantly, this definition implies that a system may use different, incompatible forms of IP and still be IP-modular, as long as the incompatible “chunks” of knowledge are associated with different modules of the overall system. For example, turning back to Figure 1, let A and B refer to incompatible forms of IP. The system shown in Figure 1(a) then is not IP-modular, while the one shown in Figure 1(b) is.

IP modularity is a stronger condition than “simple” product or process modularity. However, architects exercise control over the module boundaries and interfaces of the system, and (if the firm owns the IP in question) may also exercise control over the IP status accorded to different chunks of knowledge. Thus it is often possible to design or redesign a system to be “IP-modular,” if that is viewed as a desirable goal. Such redesign is illustrated in Figure 2. Here system 2(a) is modular, but not IP-modular. Shifting the boundary creates system 2(b) which *is* IP-modular. In general, however, the IP-modular system may not be optimal from an engineering standpoint, an issue we discuss in the next section.

---- *Insert Figure 2 about here* ----

VALUING IP-MODULARITY: BASIC THEORY

In this section, we describe the costs and tradeoffs of achieving IP modularity and relate them to the economic theories of transaction and agency costs.

Costs and trade-offs

Even though the modularity of a product or process and its IP modularity are largely under the control of product or process architects, *neither type of modularity comes free*. Strict product or process modularity with well-defined boundaries and interfaces is costly to achieve and maintain (Baldwin and Clark, 2000). It was possible to organize the porcelain factory as two non-overlapping task modules with a formal procedural interface, but it would have been easier to organize the work with overlapping tasks and informal handoffs. Similarly, segregating different chunks of knowledge and IP is more expensive than leaving the knowledge and IP unseparated. In the case of porcelain making, to maintain IP modularity, the ruler had to use soldiers to keep workers in separate zones within the fortress (Gleeson, 1998). In general, the advantages of IP modularity must always be weighed against the costs of achieving an IP-modular structure and enforcing the module boundaries.

From a pure value creation standpoint, there will at any time be an optimal modular structure. However, the optimal modular structure for *capturing* value is not necessarily the same as the optimal structure for *creating* value. This tension between value creation and value capture and the need to trade off benefits and costs can be illustrated via a simple model of the focal firm's architectural decisions.

A Simple Decision Model

We assume that the focal firm controls to some extent the architecture of a complex system. For each architecture under consideration, the focal firm needs to estimate the cost, the value created by the architecture, and the share of this value the firm can appropriate.

We assume the focal firm considers three basic architectures. The "*best closed*" architecture is optimized with respect to value creation under the restriction that the firm designs all components using only its own IP and keeps this IP proprietary. The value created by the best closed system is denoted v_c. (All values are defined net of the cost of creating the system.) The "*best open*" architecture is optimized with respect to value creation irrespective of who owns the relevant IP. Here the design of some components will be allocated to outside parties by opening up the focal firm's IP and/or the firm will use external (incoming) IP in its own designs. The value created by the best open system is denoted v_o. The third architecture is *IP-modular*. As with the best open architecture, the focal firm may select certain modules to be open, license "outgoing" IP to third parties, and license "incoming" IP from others. However, it must draw module boundaries in a way that segregates incompatible chunks of knowledge. The value created by the IP-modular system is denoted v_m.

We stipulate that the best open architecture creates more value than the best closed architecture: $v_o > v_c$. This inequality is likely to hold in a complex system, because the best open system is less constrained in its design than the best closed system. Similarly, the best open system is less constrained than the IP-modular system, hence: $v_o > v_m$.

The three architectures differ not only in terms of value created, but also in terms of value captured by others. Consider first end users. For simplicity, we assume the value allocated to end users and the transaction cost of selling final goods account for the same fraction of value under each architecture. Hence the value available to be split amongst producers (including the focal firm, its suppliers, complementors and employees) equals a fraction α of the total value created.

Under all three architectures, the focal firm will incur agency costs (Jensen and Meckling, 1976). These costs include the value captured by opportunistic agents, e.g., their private benefits (Hart, 1995) and the costs of controlling opportunism, e.g., the cost of bonding and monitoring agents. In Jensen and Meckling's (1976) integrated analysis of agency, these are both costs of delegating work to others. Let the agency cost of each architecture be denoted by a_c, a_o and a_m respectively. We assume that the agency cost is highest under the closed architecture.

Under the open and the IP-modular architectures, the focal firm will also incur transaction costs vis à vis other producers (Coase, 1937; Williamson, 1979, 1985). As with agency costs, transaction costs can be split into the value captured by opportunistic transactors and the costs of setting up, monitoring and enforcing contracts (Hart, 1995). In Williamson's (1979, 1985) integrated analysis of transaction costs, these are both "costs of using the market." We assume that the closed architecture incurs no transaction costs vis à vis other producers. For the best open and the IP-modular architecture, transaction costs are strictly positive and denoted by t_o and t_m respectively.

The value V captured by the focal firm under each architecture can now be written as:

$$V_c = \alpha v_c - a_c \quad \text{(best closed);}$$

$$V_o = \alpha v_o - a_o - t_o \quad \text{(best open);}$$

$$V_m = \alpha v_m - a_m - t_m \quad \text{(IP-modular).}$$

For a firm seeking to maximize its own value, the IP-modular architecture will be preferred if and only if $V_m > max\ [V_c, V_o]$. From this fact, it follows that an IP-modular architecture is preferred if and only if:

$$\alpha v_m - a_m - t_m > \alpha v_c - a_c \text{ ; and} \tag{1a}$$

$$\alpha v_m - a_m - t_m > \alpha v_o - a_o - t_o \text{ .} \tag{1b}$$

Inequalities (1a) and (1b) are *both necessary and, jointly, sufficient* conditions for IP modularity to be a preferred architectural strategy for the focal firm. From inequality (1a), given that transaction costs, t_m, are strictly positive, IP-modularity can be preferred only if $\alpha v_m - a_m > \alpha v_c - a_c$. In other words, the IP-modular system must either *create more value* and/or *have lower agency costs* than the closed system. From the second inequality, given that $v_o > v_m$, IP-modularity can be preferred only if $a_m + t_m < a_o + t_o$. In other words, the IP-modular system must have *lower total transaction and agency costs* than the best open system. Lower transaction and agency costs in turn can arise in two ways: (1) the IP-modular system reduces the fraction of the value of the open system claimed by third parties; and/or (2) it reduces the costs of setting up, monitoring and enforcing contracts.

Figure 3 shows how the IP-modular architecture comes to be preferred. Each stack represents the value created by each of the three architectures. The grey blocks represent the value allocated to end users, transaction costs vis à vis other producers, and agency costs. The fraction of value allocated to end users is the same for each stack; transaction cost is zero for the best closed architecture; and agency cost is higher for the closed architecture than for the alternatives. The white blocks represent value captured by the focal firm. In the illustration, the IP-modular architecture has higher value created and lower agency cost than the closed architecture; and lower total transaction and agency cost than the best open architecture, which outweighs the lower level of value created. Under these conditions, the IP-modular architecture is preferred over the other two.

---- Insert Figure 3 about here ----

THEORETICAL EXTENSIONS AND EMPIRICAL PREDICTIONS

In this section we enrich the abstract theory presented above by presenting case examples from practice, showing how conditions favoring IP modularity arise in practical settings, and deriving strategic recommendations and empirical predictions. Our examples and propositions fit within the theory laid out in the previous section: In every instance, the advantage of IP modularity derives from some combination of higher value created in an open system and reduced transaction or agency costs. However, as we will show, IP modularity is utilized as an architectural strategy in a wide range of situations, and this phenomenological diversity tends to obscure the unity of the underlying theory. Our case examples are thus intended to illustrate both the unity of the theory and its scope (Eisenhart and Graebner, 2007).

We drew our case examples from many sources, including published academic case studies; quantitative analyses; histories; interviews with decision makers; and press reports. Table 1 summarizes our examples and sources.[2]

---- Insert Table 1 about here ----

To simplify the exposition, we organized the case examples and propositions under the following headings: (1) outgoing IP; (2) incoming IP; (3) the effects of uncertainty. We also framed the propositions as strategic recommendations. After presenting a case example that illustrates a particular use of IP modularity, the related proposition succinctly states under what conditions IP modularity is advantageous and explains where the module boundaries should go. Although the propositions are stated in normative terms, each one can be reframed as a positive prediction: under the assumption that firms behave optimally, IP modularity will appear where it is strategically advantageous.

---

[2] In selecting examples, we gave preference to academic case studies and quantitative analyses. However, because IP-modularity has not been a focus of academic inquiry before now, we could not always find such sources to illustrate each important use of the strategy. We included examples from non-academic sources because we wanted to understand as broadly as possible how IP modularity is used in practice. If we depended only on prior academic work and excluded all other sources, we would be guilty of "looking for keys under the lamppost." For this reason, we admitted evidence from interviews, histories and press reports, with full disclosure as to the source (see Table 1).

Outgoing IP

We described M-System's redesign of its flash memory product in the Introduction. Importantly for our argument, the redesign was costly in several ways. The company had to rewrite its software, redesign the physical memory device, and implement the new product design in their production facility. However, by simplifying the integration of its flash memory into mobile devices, the firm enabled external value co-creation by device manufacturers and open source developers. Whenever the potential for such value co-creation exists, splitting a product into relatively "open" modules that allow for value co-creation and other "proprietary" or "closed" modules that enable value capture will make sense. This leads to our first general proposition:

> *Proposition 1: [Value Co-creation] The greater the potential for value co-creation in the surrounding ecosystem, the more advantageous is outgoing IP modularity. Module boundaries should go between the open and proprietary IP.*

The benefits of IP modularity also depend on the extent to which value co-creation is distributed. Consider the case of Counter-Strike. In 1998, Valve Software (Valve) released the game "Half-Life" with the codebase designed in such a way that the core engine and the complementing code were separate modules (Jeppesen, 2004). These modules were given different IP status. Valve put the engine under a proprietary license and kept its source code secret, while it disclosed the complementing source code and granted users a broad license to modify and share it. Within eight months of release, users had used the complementary code to build a modified game, "Counter-Strike," which became far more popular than the original game. Still, in order to run the free modification, Counter-Strike players had to license and use the Half-Life core engine, which created a remarkable commercial success for Valve.

Valve Software had no idea who of the tens of thousands of buyers of Half-Life would work on the game's code, and developers like to tinker with code rather than sign legal papers. Hence, requiring contracts and nondisclosure agreements (NDAs) with potential downstream developers was not a viable option for Valve due to the excessive transaction cost of identifying contract partners. The following proposition summarizes this reasoning:

> *Proposition 2: [Distributed Co-creators] The more distributed, numerous, and anonymous the co-creators of value, the more advantageous is outgoing IP modularity.*

Having established the general linkage between outgoing IP modularity and value co-creation, we now turn to propositions that specifically address value co-creation by customers and complementors (Adner and Kapoor, 2010 a, b). Assume the focal product is sold to a systems integrator and becomes part of a larger system downstream. When this downstream system is complex, the system integrator typically assembles numerous preexisting components. To optimize system-level performance, system integrators must often adapt the behavior of individual components. Such adaptations can only take place if the system integrator knows and understands the components in some detail—thus systems integrators must "know more than they make" (Brusoni, Prencipe, and Pavitt, 2001).

However, full knowledge by the system integrator of the design of a component may compromise the supplier's IP. Again, IP modularization provides a way out of this quandary. Quite often it is possible to segment the component's design into information that is helpful to the integrator, and information that pertains only to the internal workings of the component (Brooks, 1975; Parnas, 1972). From an IP perspective, the former can be treated as relatively "open" IP and the latter as more proprietary or "closed" IP. Such architectural delineation of "open" and "closed" entails considerably lower transaction cost than drafting and monitoring a contractual delineation that refers to a non-IP-modular product. Again, the case of M-Systems provides a prime example. This leads to our third proposition:

Proposition 3: [Complexity] The more complex the downstream system, the more advantageous is outgoing IP modularization. The module boundaries should separate the IP that helps to integrate the focal component into the system from that which contributes to the component's internal performance.

IP modularity is also important when downstream user needs are heterogeneous and unpredictable, i.e., when users demand variety. One way to accommodate customers' demand for variety is for the focal firm to provide a list of features or options that customers may select. However, this approach is often very costly, and still may not cover the full range of customer needs or desires. Similarly, granting each user individual rights for the modifications he or she desires would entail high transaction cost.

Another way to address heterogeneous and unpredictable user needs is to split the product, and the related outgoing IP, into two sets of modules. Modules in the first set are unmodifiable and their IP is protected, while those in the second set are supplied in a modifiable fashion and under a more open IP status. In a similar way as with toolkits for user innovation (von Hippel, 2001), users' modifications may then range from minor changes to entirely new modules. This example gives rise to our fourth proposition:

Proposition 4: [Customization] The greater and more varied the need for downstream adaptations, the more advantageous is outgoing IP modularization. The module boundaries should separate the IP that serves as the basis for modification from the IP supporting the proprietary "core" modules.

It is worth noting the link between Proposition 4 and "platform strategies" (e.g., Gawer and Cusumano, 2002). In a "platform architecture," certain components (the platform) remain fixed, while others (the complements) are permitted to vary (Baldwin and Woodard, 2009). This strategy then permits the focal firm to give control over the design of complements to users or third parties. A platform strategy is a specific instance of IP modularization in the sense that, as MacCormack and Iansiti (2009) describe, the platform owner will often award a proprietary IP status to the core of the platform, and a more open IP status to the interfaces that allows others to link complementary components to it.

Sometimes value co-creation takes place upstream of the focal firm, i.e., through innovation by suppliers (Adner and Kapoor, 2010a, b). In order to enable innovation by suppliers, the focal firm typically needs to share knowledge, such as blueprints or parts specifications. It can protect that knowledge via NDAs and licenses, but it perforce must give up the protection of secrecy. If the transaction cost of policing and enforcing such NDAs is high, IP modularity offers a solution. By partitioning knowledge and IP, it reintroduces secrecy as a protective measure for some parts of the focal firm's overall system, strengthening the innovator's ability to appropriate value in the supply chain.

There are two situations in which IP modularity vis-à-vis suppliers deserves special attention from a strategic perspective. First, when a given supplier also sells to the focal firm's competitors, leakage of IP may result. As an example, consider the following case from the automotive industry. A supplier of braking systems sells its product to several auto manufacturers. One of its customers developed a stability control system which relied heavily on features of the braking system, in particular on its ABS (antilock braking system). From a strictly technical perspective, it would have been optimal to integrate the braking and stability systems into one module.

Suppose the automaker had gone this route and sought to codevelop an integrated braking-and-stability system in conjunction with its supplier. The supplier might then have sold two braking systems—one to the automaker who provided the IP, and one to its other customers—or, it might have sold the braking-and-stability system to all its customers. The second option was unacceptable to the automaker, which considered its proprietary stability system to be an important point of product differentiation. The first would have required the brake supplier to maintain two distinct products, forgoing economies of scale, and would still have posed the risk of inadvertent loss of critical IP.

The automaker instead developed the stability system as a separate module, which was added to the braking system during assembly. In other words, its desire to protect proprietary IP in the stability system caused the automaker to adopt an IP-modular design for its braking system. In the words of an interviewee from the auto company:[3] "*Sometimes we need to re-segment both hardware and software modules, or the modularity of the system, based more on the commercial needs of, say, protecting an in-house algorithm, than on just the most efficient design.*" These considerations lead to our fifth proposition:

> *Proposition 5: [Leakage of IP through a Common Supplier] Outgoing IP modularity is more advantageous when the focal firm shares a common supplier with competitors. Module boundaries should separate the knowledge suppliers need to build components that work in the customer's system from the knowledge contributing to features that differentiate the customer's product from the competition.*

We now turn to the second important instance of IP modularity vis-à-vis suppliers, and the related strategy of *divide et impera* (divide and rule). When a firm outsources a large portion of its production to external suppliers and/or provides critical knowledge to employees, it must be concerned that it is giving opportunistic agents precisely the knowledge they need to compete in its own markets. However, each participant only needs knowledge relevant to the modules it designs or builds (Baldwin 2008, Tiwana, 2008a, b). Thus, if the focal firm divides the work among different agents who do not exchange information with each other, critical IP can be protected. Each agent will know part of the puzzle, but none can reconstruct the whole. IP modularization in this case requires breaking up the product design or production process into

[3] Source: Interview with R&D manager (11 July 2008).

different task modules, and allocating those modules to separate, non-communicating agents. This architectural strategy was employed by the ruler of Saxony in the porcelain example discussed above.

A modern case in point is that of SMaL, a maker of the core components for very small, low-power cameras (Christensen and Anthony, 2003). SMaL sold a "camera kit" consisting of five components, two of which were the control chip and the imager chip. It would have been common practice to integrate these chips into one. However, SMaL decided against an integral design. IP considerations played a role in this decision. Separating the chips allowed SMaL to encase the control chip, which helped to conceal SMaL's cost structure from buyers and also made reverse engineering more difficult. In addition, even though SMaL eventually decided to source both chips from one supplier (whom they judged to be very trustworthy), SMaL looked into having the two chips manufactured by different suppliers to prevent one supplier knowing the design of these two key parts of the system.[4]

IP-oriented modularizations based on the logic of "divide and rule" can strengthen IP protection in otherwise weak appropriability regimes. This leads to our sixth proposition:

> *Proposition 6: [Divide and Rule] When the focal firm must rely on opportunistic suppliers and/or employees and its IP is not strongly appropriable, outgoing IP modularity is advantageous. The module boundaries should divide the total knowledge needed to make the product into separate task modules, which can then be outsourced to different suppliers or allocated to separate business units.*

Note that—in line with our general argument about complementarity between IP modules—for the "divide and rule" strategy to work, it is essential that the IP allocated to the individual task modules has little or no value independent of the focal firm's overall system.

---

[4] Sources: Interview with Maurizio Arienzo, former CEO of SMaL (10 September 2008); Christensen and Anthony (2003).

Zhao (2006) provides a very clear example of a “divide and rule” strategy based on complementarity in the context of multinational companies (MNCs) managing research and development (R&D) across international boundaries. The knowledge created through R&D cannot be protected effectively in countries with weak intellectual property rights (IPR). In our terminology, R&D projects in weak IPR countries are (process) modules with an unavoidably “open” IP status. According to Zhao (2006), multinationals address this problem by assigning to weak IPR countries projects whose results are strongly complementary with those of R&D projects conducted in the United States. In effect, the MNCs *divide* and *allocate* R&D projects so that the knowledge obtained in weak IPR countries has relatively little stand-alone value. This strategy, in turn, allows the MNCs to appropriate a greater portion of the aggregate value of their R&D investments. In support of this argument, Zhao (2006) presents evidence from patent citations that IP created in weak IPR countries has more value inside the focal firm than outside of it.

Incoming IP

We now turn to situations in which the focal firm does not control the IP status of some components of its system. However, it can control its system’s technical architecture (subject to physical and cost constraints) and use modularity to mitigate various IP-related concerns.

First, we address the risk of holdup. LaMantia, Cai, MacCormack, and Rusnak (2008) describe a software company whose entire product family depended on a central platform component. This platform contained both the company’s own code and code licensed-in from another software vendor, hence this component was not IP-modular. Furthermore, the platform was designed in an integral fashion, with the licensed-in code spread throughout the codebase. It would have been very difficult to separate the licensed-in code from the company’s own code on short notice. When the license came up for renewal, this fact would have exposed the firm to "holdup" on the part of the licensor (Williamson, 1979, 1985).

Anticipating this threat, the focal firm re-modularized the codebase, placing a newly designed platform and the licensed-in code in separate modules. Although the system required both modules, the platform module no longer contained, nor did it depend on, any licensed-in code. As a result, the focal firm's switching cost was greatly reduced, and risk of holdup was largely eliminated. This case example motivates our seventh proposition:

> *Proposition 7: [Holdup Risk] Incoming IP modularity is advantageous when the focal firm faces the risk of holdup from suppliers of incoming IP. The module boundaries should go between the firm's own IP and the incoming IP.*

Propostion 7 can be reinterpreted in terms of Williamson's concept of asset specificity (Williamson, 1979, 1985). When a system has an integral (i.e., non-modular) design, its subsidiary components depend on each other in myriad ways. In this case, the firm's own assets (its code) became intertwined with assets it did not own (the licensed-in code), and hence highly specific to them. Asset specificity increases the threat of holdup, making arm's-length transactions more costly relative to transfers within a firm. As a result, transaction cost economics recommends that highly specific assets should be owned by a single firm. However, in the example presented above (LaMantia et al. 2008), the focal firm took a different route: it reduced asset specificity by making the underlying system IP-modular.

When dealing with incoming IP, transaction cost may also be high simply because the upstream co-creators of value are distributed and numerous. If, in addition, inconsistent legal or contractual restrictions create incompatibility between different "chunks" of incoming IP *within* modules (hence, if the architecture is not IP modular), transaction cost is particularly high. This problem arose at Sun Microsystems when it moved to license key implementations of its Java programming language as open source software. The technology magazine *eWeek* quotes Sun General Counsel Mike Dillon, saying the transition was tedious and legalistic: "*Java Standard Edition contains about 6 million lines of code. […] Our legal team [of 190 lawyers] had to go over it, line by line, and look for all copyright marks and third-party involvements. Where Sun didn't have the correct licenses, we had to contact the owners, one by one, and determine the rights.*"[5].

In Sun's case, the problem was caused not by the opportunism of a single supplier, but by the sheer multitude and variety of IP sources. Such multiplicity of sources is now a common occurrence in large software-intensive systems. It may give rise to incompatibilities between different chunks of incoming IP and between the firm's incoming and own IP. In particular, the entanglement of incoming commercial IP with the firm's own IP creates a headache when, for strategic reasons, the focal firm wants to release part or all of its system as open source software to instigate external value creation.

We conclude that when a firm has many diverse sources of incoming IP, it should reduce the impact of IP incompatibilities by designing its whole system—whether it be a codebase, a product, or a process—in an IP-modular fashion. With software, doing so may even be a contractual obligation since, as discussed above, open source and commercial licenses often impose contradictory requirements on the licensee. This leads to our eighth proposition (which, we note, is the mirror image of Proposition 2):

> *Proposition 8: [Distributed ownership of incoming IP] Incoming IP modularity is advantageous when the focal firm obtains incoming IP on different terms from diverse sources. Module boundaries should be placed to ensure IP-compatibility within modules and to minimize the costs of renegotiation. In particular, it should be*

---

[5] Source: http://www.eweek.com/c/a/Application-Development/Sun-Pours-Out-Java-Cup/ (accessed 6 January 2009).

possible to change the IP status of a module without renegotiating numerous incoming IP licenses.

The effect of uncertainty

Uncertainty may compound the various sources of transaction cost that played a role so far. By creating IP-modular designs, firms gain the ability to change their IP strategy in specific parts of the system in response to new value-capture opportunities or to threat of expropriation. Thus, IP modularity creates option value.

Consider the example of the operating system Darwin, originated by Apple Inc.[6] Originally, Apple kept the code that later became Darwin proprietary, thus from an IP perspective there was no reason not to adopt an integral design. However, in 2000 Apple made large parts of Darwin's source code—but not all of it—publicly available under an open source license, to take advantage of distributed value creation.[7] According to our Propositions 1 and 2, IP modularity is desirable in such circumstances, but it is typically difficult and expensive to implement ex post. To the extent that Apple anticipated this move when designing the system, it might have built into the program's architecture the option of selectively changing the IP status of certain parts. That is, it may have performed an "anticipatory" IP modularization by creating a design that featured a high, seemingly excessive degree of modularity.

Generally, in settings characterized by high levels of technological or market uncertainty, changes in the external environment can arise after the fact that make IP-modularity desirable. As in the case of M-Systems, customers may require openness of some of the product's IP as a

[6] Source: http://en.wikipedia.org/wiki/Darwin_(operating_system) (accessed 25 March 2009).

[7] Note that Darwin was, even before its release under an open source license, largely based on licensed-in open source code. However, the respective licenses allowed keeping derived work proprietary, which is what Apple initially did.

condition of purchase. Alternatively, outsourcing of production may become attractive for cost reasons. An "overly modular" initial design allows the focal firm to respond to such changes rapidly and cost-effectively. In summary, we have:

> *Proposition 9: [Uncertainty and Outgoing IP] An "overly modular" product or process architecture creates options to capture value in the future by selectively adapting the IP status of individual modules. Such options for outgoing IP modularity are more valuable in the presence of high market or technological uncertainty.*

Also when incoming IP is subject to uncertainty, IP modularity creates option value. We have already seen (Proposition 7) that IP modularity is a way to counter a known threat of holdup. Today, however, firms are increasingly vulnerable to IP-related threats that cannot be predicted when the product or process architecture is chosen. For a complex new product or process, in fields like electronics and software, it is often impossible to identify with certainty all patents that the product might infringe. Patent "trolls" are non-producing firms that enforce patents, often acquired ones, against firms that unwittingly use the technology covered by the patent (Golden, 2007; Henkel and Reitzig, 2007; Lemley and Shapiro, 2007; Reitzig, Henkel, and Heath, 2007). A key aspect of the trolls' strategy is surprise: they wait until the IP-related product or process is successful before filing suit, thus maximizing the value of any future damages or settlement. For example, Forgent Networks collected, in less than three years, more than US$100 million in licensing fees after suing users of the JPEG standard for infringement of one of its patents (Reitzig et al., 2007:136).

One way to counter a patent infringement suit is to "design around" the patent by replacing the allegedly infringing component with a non-infringing substitute. Design arounds, in turn, are less costly when the underlying product or process is modular. As long as the infringement is confined to one (or a few) modules, those modules can be redesigned or even removed without compromising the functionality of the rest of the system.

The MPEG-4 compression standard for audio and video data is a case in point.[8] MPEG-4 is not a single technology, but rather a collection of methods and so-called "parts." For each application, MPEG-4 is implemented by generating a specific "profile" that assembles the parts ("modules" in our context) required for it. The modular structure of MPEG-4 allows firms, in defining a "profile," to leave out parts with a high perceived IP uncertainty. Furthermore, when components under uncertain IP status are encapsulated into a separate module of the product, it will be easier to replace or even drop them if it turns out that they infringe on so far unidentified patents. This leads to our last proposition:

> *Proposition 10: [Inadvertent Infringement] A product or process architecture characterized by incoming IP modularity is a partial defense against unanticipated incoming IP claims, because it gives the firm options to "design around" the claims without redesigning the whole system.*

DISCUSSION AND CONCLUSIONS

In concluding, we want to point out generalizations of the concept of "IP modularity," discuss a number of caveats, and summarize.

For clarity in exposition, we have focused on IP modularity in the products firms sell and the processes they use. However, the concept extends to organizations. "Chinese Walls" (McVea, 1993) are an illustrative example. This term refers to virtual barriers between different organizational units that prevent information exchange between these units—the advisory and trading departments of a bank, for example. These units, in our notion, constitute "IP modules"

[8] Sources: Interview with Klaus Diepold, member of the MPEG-4 standardization committee (24 July 2007); Henkel and Reitzig (2008); http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm (accessed 7 January 2009).

within the organization, and the Chinese Walls between them are module boundaries. Similarly, Liebeskind (1997:650) proposes "structural isolation" of organizational units that deal with sensitive knowledge as a mechanism to keep that knowledge secret. These units also constitute organizational IP modules.

A second generalization concerns the rights and obligations under consideration. While we focused on IP as an important source of such rights and obligations, the latter may also have other legal or contractual origins. For example, the need to clearly attribute liability in case of product failures may, in a similar way to IP, give rise to a modular architecture. In safety-critical products, modules may be segregated due to differing safety certification levels. Classified data may need to be stored and processed separately from unclassified data, again suggesting a modular architecture (Ames, 2003).[9]

We now briefly address approaches to testing our propositions empirically. First, industrial contexts differ, thus it is unlikely that all propositions could be tested in a single industry. For a group of related propositions, one must choose an industry characterized by complex products and processes, such as software, electronics, telephony or pharmaceuticals. Then, one needs to devise a way to measure the degree of modularity of artifacts, and to quantify the extent to which different components carry diverse IP status. Depending on the proposition under consideration, one must finally operationalize the respective independent variables—among others, the potential for value co-creation, the degree of distributedness of value co-creators, and the level of complexity of downstream systems.

---

[9] Also the concept of "IP incompatibility," discussed in the theory section and in Proposition 9, may be generalized. It is applicable not only to the IP-dimension of product modularity, but also to other dimensions of coupling between the elements of a complex artifact. It thus constitutes an extension of the theory of modularity in general.

Within any industry, software seems an especially suitable target of analysis. Unlike processes or hardware products, software can be analyzed in an automated fashion with respect to its modularity (e.g., MacCormack et al., 2006) and even, to some extent, with respect to the modules' IP status (Black Duck Software, 2008). One may then either assume that actors behave rationally, and take the occurrence of IP modularity as an indication of its being advantageous, or one may relate the degree of IP modularity to some measure of success in value appropriation, although this approach is typically challenging due to concerns about endogeneity.

Before closing the paper, we must emphasize a few caveats and limitations. First and foremost, modularity almost always comes at a cost. Thus IP modularity, which we have said is stricter than simple modularity, is likely to increase the cost of design, require legal clarifications, and may imply a loss of performance. As was emphasized in our simple decision model, IP strategists must evaluate these costs in relation to the benefits of IP modularity in terms of higher value capture.

Second, in some circumstances, even with a modular product or process architecture, IP modularity may not be necessary. For example, if a company's strategy for appropriating value rests heavily on ownership of complementary assets (Teece, 1986), that company may not need to protect its IP. Similarly, trust between organizations reduces the risk of opportunistic behavior, and thus may reduce the need to pursue IP modularity.

Finally, the industries and technologies covered in our case examples range from software to automobiles to porcelain. Such variety served our goal of demonstrating the breadth of application of the concept, but unavoidably limited the level of detail in our discussion. Future research can address this limitation by investigating how modules are delineated and IP partitioned for particular artifacts and in specific business ecosystems. For example, it would be enlightening to compare the technical and IP structure of the ecosystems built up around Apple's iPhone and Google's Android mobile phone.

In conclusion, in this paper we have explored the concept of IP modularity. Fundamentally, IP modularity eliminates incompatibilities between IP rights in a given module, while permitting incompatibilities within the overall system. This in turn permits a firm to "have its cake and eat it too," that is, it can reap the benefits of an open architecture while at the same time reducing the costs of opportunism on the part of suppliers, complementors and employees. In this fashion, IP modularity overcomes intrinsic conflicts between distributed value creation on the one hand, and value appropriation on the other hand.

The details of IP modularization must be determined by engineers and legal experts working together. But beyond the technical and legal concerns, IP modularity affects a firm's strategies for value appropriation in increasingly complex and fragmented technological spaces. It is an issue that deserves the attention of general managers and management scholars generally.

REFERENCES

Adner R, Kapoor R. 2010a. Value Creation in Innovation Ecosystems: How the Structure of Technological Interdependence Affects Firm Performance in New Technology Generations. *Strategic Management Journal* **31**: 306-333.

Adner R, Kapoor R. 2010b. Innovation Ecosystems and the Pace of Substitution: Re-examining Technology S-curves. *Working paper Dartmouth University*: http://mba.tuck.dartmouth.edu/pages/faculty/ron.adner/working_papers.html.

Ames B. 2003. Real-time software goes modular. *Military & Aerospace Electronics,* September **14**(9): 24–29.

Baldwin CY. 2008. Where do transactions come from? Modularity, transactions, and the boundaries of firms. *Industrial and Corporate Change* **17**: 155–195.

Baldwin CY, Clark KB. 2000. *Design rules, Volume 1: The power of modularity.* MIT Press: Cambridge, MA.

Baldwin CY, Clark KB. 2006. The architecture of participation: Does code architecture mitigate free riding in the open source development model? *Management Science* **52**: 1116–1127.

Baldwin CY, Woodard CJ. 2009. The architecture of platforms: A unified view. In *Platforms, Markets, and Innovation*, A Gawer (ed). Edward Elgar: Cheltenham, UK; 19-44.

Barney JB. 1986. Strategic factor markets: Expectations, luck, and business strategy. *Management Science* **32**: 1231-1241.

Barney JB. 1991. Firm resources and sustained competitive advantage. *Journal of Management* **17**: 99–120.

Black Duck Software. 2008. Software compliance management: Automating license compliance in the new, mixed-IP development world. White paper. Waltham, MA: Black Duck Software. http://ducks.blackducksoftware.com/~whitepapers/WP-SP-0805-UL-AA-SCM-Web.pdf.

Brooks Jr. FP. 1975. *The mythical man month: Essays on software engineering*. Addison-Wesley: Reading, MA.

Brusoni S, Prencipe A, Pavitt K. 2001. Knowledge specialization, organizational coupling, and the boundaries of the firm: Why do firms know more than they make? *Administrative Science Quarterly* **46**: 597–621.

Chesbrough HW. 2003. *Open innovation. The new imperative for creating and profiting from technology*. Harvard Business School Press: Boston.

Chesbrough HW, Teece DJ. 1996. When is virtual virtuous? Organizing for innovation. *Harvard Business Review* **74**: 65–73.

Christensen CM, Anthony SD. 2003. *Making SMaL big: SMaL Camera Technologies*. Case No. 603-116. Harvard Business School Case Services: Boston.

Coase RH. 1937. The nature of the firm. *Economica* **4**(16): 386-405.

Coff RW. 1999. When competitive advantage doesn't lead to performance: The resource-based view and stakeholder bargaining power. *Organization Science* **10**: 119–133.

Daft R. 1983. *Organization theory and design*. West: New York.

Dyer JH, Singh H. 1998. The relational view: Cooperative strategy and sources of interorganizational competitive advantage. *Academy of Management Review* **23**: 660–679.

Eisenhart KM, Graebner MA. 2007. Theory Building from Cases: Opportunities and Challenges, *Academy of Management Journal* **50**(1): 25-32.

Ferguson CH, Morris CR. 1993. *Computer wars: How the West can win in a post-IBM world*. Times Books: New York.

Fine CH. 1998. *Clockspeed: Winning industry control in the age of temporary advantage*. Da Capo Press: Cambridge, MA.

Foray D. 2004. *The economics of knowledge*. MIT Press: Cambridge, MA.

Franke N, von Hippel E. 2003. Satisfying heterogeneous user needs via innovation toolkits: The case of Apache security software. *Research Policy* **32**: 1199–1215.

Free Software Foundation. 1991. *The GNU General Public License (GPL) – Version 2, June 1991*. http://www.opensource.org/licenses/gpl-2.0.php.

Garud R, Kumaraswamy A. 1995. Technological and organizational designs for realizing economies of substitutions. *Strategic Management Journal* **16**: 93–110.

Gawer A, Cusumano MA. 2002. *Platform leadership: How Intel, Microsoft, and Cisco drive industry innovation*. Harvard Business School Press: Boston, MA.

Gleeson J. 1998. *The arcanum: The extraordinary true story*. Warner Books: New York.

Golden JM. 2007. "Patent trolls" and patent remedies. *Texas Law Review* **85**: 2111–2162.

Gomes PJ, JoglekarNR. 2008. Linking modularity with problem solving and coordination effort. *Managerial and Decision Economics* **29**: 443–457.

Gulati R. 1998. Alliances and networks. *Strategic Management Journal* **19**: 293–317.

Hart O. 1995. *Firms, Contracts and Financial Structure*. Oxford University Press: Oxford, UK.

Harhoff D, Henkel J, von Hippel E. 2003. Profiting from voluntary information spillovers: how users benefit by freely revealing their innovations. *Research Policy* **32**: 1753–1769.

Henderson RM, Clark KB. 1990. Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms. *Administrative Science Quarterly* **35**: 9-30.

Henkel J. 2006. Selective revealing in open innovation processes: The case of embedded Linux. *Research Policy* **35**: 953-969.

Henkel J, Reitzig M. 2007. *Patent sharks and the sustainability of value destruction strategies.* Working paper: available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=985602.

Henkel J, Reitzig M. 2008. Patent sharks. *Harvard Business Review* **86**: 129–133.

Hoetker G. 2006. Do modular products lead to modular organizations? *Strategic Management Journal,* **27**: 501–518.

Iansiti M, Levien R. 2004. *The keystone advantage*. Harvard Business School Press: Boston.

Jacobides MG. 2005. Industry change through vertical disintegration: How and why markets emerged in mortgage banking. *Academy of Management Journal* **48**: 465–498.

Jensen MC, Meckling WH. 1976. "Theory of the Firm: Managerial Behavior, Agency Costs, and Ownership Structure," *Journal of Financial Economics* **3**(4): 305-360.

Jeppesen LB. 2004. *Profiting from innovative user communities: How firms organize the production of user modifications in the computer games industry*. Working paper No. 2004-03. Copenhagen Business School: http://ep.lib.cbs.dk/download/ISBN/ 8778690978.pdf

Kaplan F. 2006. Opening the door for the latest NAND flash in open source mobile platforms. *LinuxDevices.com*: http://www.linuxdevices.com/articles/AT2185129745.html.

Käs, S. 2008. *Rethinking industry practice: The emergence of openness in the embedded component industry*. Pro BUSINESS: Berlin.

LaMantia MJ, Cai Y, MacCormack AD, Rusnak J. 2008. Analyzing the evolution of large-scale software systems using design structure matrices and design rule theory: Two exploratory cases. *Seventh Working IEEE/IFIP Conference on Software Architecture*: 83–92.

Langlois RN. 2002. Modularity in technology and organization. *Journal of Economic Behavior & Organization* **49**: 19–37.

Langlois RN. 2003. The vanishing hand: The changing dynamics of industrial capitalism. *Industrial and Corporate Change* **12**: 351–385.

Langlois RN, Garzarelli G. 2008. Of hackers and hairdressers: Modularity and the organizational economics of open-source collaboration. *Industry & Innovation* **15**: 125-143.

Langlois RN, Robertson PL. 1992. Networks and innovation in a modular system: Lessons from the microcomputer and stereo component industries. *Research Policy* **21**: 297–313.

Lavie D. 2006. The competitive advantage of interconnected firms: an extension of the resource-based view. *Academy of Management Review* **31**: 638–658.

Lavie D. 2007. Alliance portfolios and firm performance: a study of value creation and appropriation in the U.S. software industry. *Strategic Management Journal* **28**: 1187–1212.

Lemley MA, Shapiro C. 2007. Patent holdup and royalty stacking. *Texas Law Review* **85**: 1991–2049.

Liebeskind JP. 1997. Keeping organizational secrets: Protective institutional mechanisms and their costs. *Industrial and Corporate Change* **6**: 623-663.

MacCormack A, Iansiti M. 2009. Intellectual property, architecture, and the management of technological transitions: Evidence from Microsoft Corporation. *Journal of Product Innovation Management* **26**: 248–263.

MacCormack A, Rusnak J, Baldwin CY. 2006. Exploring the structure of complex software designs: An empirical study of open source and proprietary code. *Management Science* **52**: 1015–1030.

Makadok R, Coff R. 2002. The theory of value and the value of theory: Breaking new ground versus reinventing the wheel. *Academy of Management Review* **27**: 10–13.

McVea H. 1993. *Financial conglomerates and the Chinese wall: Regulating conflicts of interest*. Oxford University Press: New York.

Parnas DL. 1972. A technique for software module specification with examples. *Communications of the ACM* **15**: 330–336.

Pénin J. 2007. Open knowledge disclosure: An overview of the evidence and economic motivations. *Journal of Economic Surveys* **21**: 326–348.

Peteraf MA. 1993. The cornerstones of competitive advantage: A resource-based view. *Strategic Management Journal* **14**: 179-191.

Reitzig M, Henkel J, Heath CH. 2007. On sharks, trolls, and their patent prey – Unrealistic damage awards and firms' strategies of 'being infringed.' *Research Policy* **36**: 134–154.

Reitzig M, Puranam P. 2008. *Value appropriation as an organizational capability: The case of IP protection through patents*. Working paper London Business School: http://ssrn.com/abstract=957335

Rumelt RP. 1984. Towards a strategic theory of the firm. In *Competitive Strategic Management*, B Lamb (ed.). Prentice-Hall: Englewood Cliffs, NJ; 556–570.

Sanchez R. 1995. Strategic flexibility in product competition. *Strategic Management Journal*, Summer Special Issue **16**: 135–159.

Sanchez R, Mahoney JT. 1996. Modularity, flexibility, and knowledge management in product and organizational design. *Strategic Management Journal,* Winter Special Issue **17**: 63–76.

Sanderson S, Uzumeri M. 1995. Managing product families: The case of the Sony Walkman. *Research Policy* **24**: 761–782.

Schilling MA. 2000. Toward a general modular systems theory and its application to interfirm product modularity. *Academy of Management Review* **25**: 312–334.

Simon HA. 1962. The architecture of complexity. *Proceedings of the American Philosophical Society* **106**: 467–482.

Staudenmayer N, Tripsas M, Tucci CL. 2005. Interfirm modularity and its implications for product development. *Journal of Product Innovation Management* **22**: 303–321.

Sturgeon T. 2002. Modular production networks: A new model of industrial organization. *Industrial and Corporate Change* **11**: 451–496.

Teece DJ. 1986. Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy. *Research Policy* **15**: 285–305.

Teece DJ. 1988. Technological change and the nature of the enterprise. In *Technical Change and Economic Theory*, G Dosi, C Freeman, RR Nelson, G Silverberg, L Soete (eds). Pinter: London; 256–281.

Teece DJ. 2000. *Managing intellectual capital: Organizational, strategic, and policy dimensions*. Oxford University Press: Oxford, UK.

Tiwana A. 2008a. Does interfirm modularity complement ignorance? A field study of software outsourcing alliances. *Strategic Management Journal* **29**: 1241-1252.

Tiwana A. 2008b. Does technological modularity substitute for control? A study of alliance performance in software outsourcing. *Strategic Management Journal* **29**: 769-780.

Ulrich KT. 1995. The role of product architecture in the manufacturing firm. *Research Policy* **24**: 419–440.

von Hippel E. 1990. Task partitioning: An innovation process variable. *Research Policy* **19**: 407–418.

von Hippel E. 2001. Perspective: User toolkits for innovation. *Journal of Product Innovation Management* **18**: 247–257.

von Hippel E, Finkelstein SN. 1979. Analysis of innovation in automated clinical chemistry analyzers. *Science & Public Policy* **6**: 24–37.

von Hippel E, von Krogh G. 2003. Open source software and the “private-collective” innovation model: Issues for organization science. *Organization Science* **14**: 209–223.

Wernerfelt B. 1984. A resource-based view of the firm. *Strategic Management Journal* **5**: 171–180.

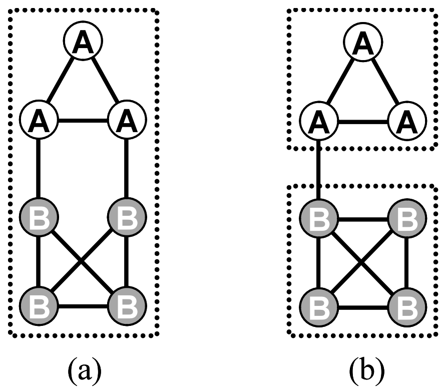
West J. 2003. How open is open enough? Melding proprietary and open source platform strategies. *Research Policy* **32**: 1259–1285.

Williamson OE. 1979. Transaction-cost economics: The governance of contractual relations. *Journal of Law and Economics* **22**: 233–261.

Williamson OE. 1985. *The Economic Institutions of Capitalism*. Free Press: New York, NY.

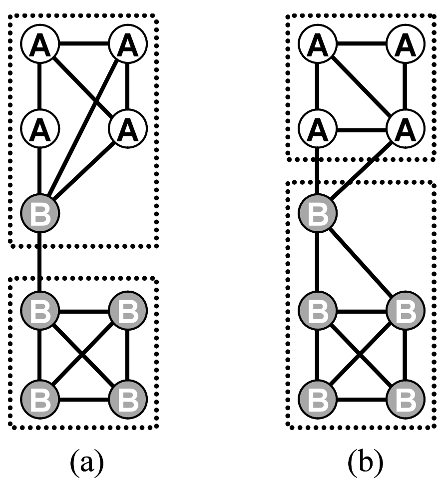
Zhao M. 2006. Conducting R&D in countries with weak intellectual property rights protection. *Management Science* **52**: 1185–1199.

FIGURES AND TABLES



Note: Elements A, B are under distinct IP status.

Figure 1. Systems with integral (a) and IP-modular (b) structure.



Note: Elements A, B are under distinct IP status.

Figure 2. Systems with modular (a) and IP-modular (b) structure.

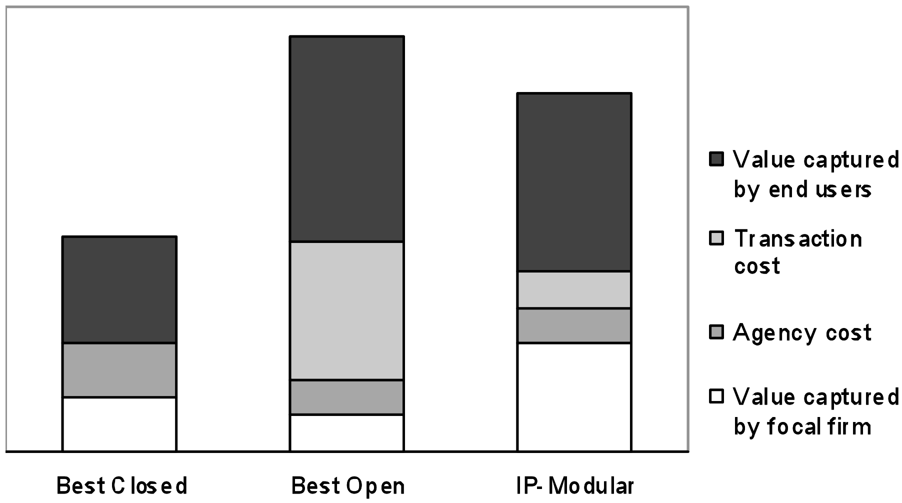


Figure 3. Comparison of the Benefits and Costs of Three Architectures

Case Example	**Source**	**Details**
Outgoing IP		
M-Systems	Case study, press reports	Käs, 2008
Counterstrike	Case study	Jeppeson, 2004
Microsoft Platform	Case study	MacCormack and Iansiti, 2009
ABS Brake System	Interview	
Meissen Porcelain	History	Gleeson, 1998
Incoming IP		
Multinational R&D	Quantitative, cross-firm analysis	Zhao, 2006
License hold-up	Quantitative, within-firm analysis	La Mantia et. al., 2008
Sun Microsystems, Java	Press reports	
Effects of Uncertainty		
Apple Darwin	Press reports	
MPEG	Case study, interview	Henkel and Reitzig, 2007, 2008
Forgent	Case study	Reitzig et al. 2007

Table 1. Case Examples and Sources