HARVARD | BUSINESS | SCHOOL

The Mirroring Hypothesis: Theory, Evidence and Exceptions

Lyra Colfer
Carliss Y. Baldwin

Working Paper

10-058

The Mirroring Hypothesis: Theory, Evidence and Exceptions

Lyra J. Colfer
lyrajcolfer@yahoo.com

Carliss Y. Baldwin
cbaldwin@hbs.edu

The Mirroring Hypothesis: Theory, Evidence and Exceptions

Abstract

The mirroring hypothesis predicts that the organizational patterns of a development project (e.g. communication links, geographic collocation, team and firm co-membership) will correspond to the technical patterns of dependency in the system under development. Scholars in a range of disciplines have argued that mirroring is either necessary or a highly desirable feature of development projects, but evidence pertaining to the hypothesis is widely scattered across fields, research sites, and methodologies. In this paper, we formally define the mirroring hypothesis and review 102 empirical studies spanning three levels of organization: within a single firm, across firms, and in open community-based development projects. The hypothesis was supported in 69% of the cases. Support for the hypothesis was strongest in the within-firm sample, less strong in the across-firm sample, and relatively weak in the open collaborative sample. Based on a detailed analysis of the cases in which the mirroring hypothesis was not supported, we introduce the concept of actionable transparency as a means of achieving coordination without mirroring. We present examples from practice and describe the more complex organizational patterns that emerge when actionable transparency allows designers to 'break the mirror.'

Keywords: Modularity, innovation, product and process development, organization design, design structure, organizational structure, organizational ties

The Mirroring Hypothesis: Theory, Evidence and Exceptions

1 Introduction

Innovation is a process in which people define problems and then actively develop new knowledge to solve them (Nonaka, 1994). In the modern economy, much new knowledge takes the form of new product and process designs. Modern efforts to develop new designs in turn require the coordination of an ever wider range of disciplines, embodied in technologically and functionally sophisticated systems and artifacts (Brusoni et al., 2007).

Given the well-known challenges of coordinating complex interdependent tasks (e.g. Thompson, 1967: Galbraith, 1977: Williamson, 1971: Arrow, 1974), theorists have tended to predict (or recommend) that the formal structure of a development organization will (or should) "mirror" the design of a system under development (e.g., von Hippel, 1990; Henderson and Clark, 1990; Sanchez and Mahoney 1996; Chesbrough and Teece, 1996; Baldwin and Clark, 2000). These theories in turn lead to an empirically testable prediction: the organizational ties of a development project (e.g. communication links, geographic collocation, team and firm co-membership) will correspond to the technical dependencies in the system under development.

This basic idea has different names in different fields. In organization design, it can be seen as an application of task contingency theory to product design and development (Lawrence and Lorsch, 1967; Thompson, 1967; Galbraith, 1977; Tushman, 1979; Drazin and Van deVen, 1985; Tushman and Nadler, 1997). In computer science, it is known as Conway's Law (Conway, 1968), and it has recently been called "socio-technical congruence" (Cataldo et. al., 2008). Notably, the hypothesis predicts correspondence but does not impose a direction of causality: effects may flow from organizational structure to technical design (Henderson and Clark, 1990); from technical design to organizational structure (Chandler, 1977); or in both directions (Baldwin and Clark, 2000; Fixson and Park, 2007).

A thorough understanding of the evidence for and against the mirroring hypothesis is difficult to achieve. The relevant literature is scattered across a number of fields in management, economics, and engineering, and there are significant differences in how the hypothesis is interpreted in the different streams of literature. Accordingly this study makes two contributions. First, it formally defines the mirroring hypothesis and systematically reviews and summarizes the empirical evidence pertaining to it. Second, it synthesizes observations from studies that contradict the hypothesis to explain when and how development organizations can "break the mirror."

Our analysis proceeds as follows. In Section 2, we trace the intellectual roots of the mirroring hypothesis and then formally define it in terms of network graphs. Building on this theoretical base, in Sections 3 and 4, we systematically review the empirical evidence pertaining to the hypothesis. We consider 102 empirical studies, most of which were published between 2000 and 2009. The studies are divided into three groups by organizational level of analysis: (1) development projects within a single firm (22 studies); (2) projects spanning two or more firms (62 studies); and (3) projects undertaken by open collaborative communities (18 studies). We classify each study according to whether its results fully or partially support or fail to support the mirroring hypothesis.

Across the entire sample, the mirroring hypothesis received support (including partial and mixed support) in just over two-thirds (69%) of the cases. The hypothesis was more strongly supported in the within- and across-firm groups (77% and 74% respectively) than in studies of open collaborative development projects (39%). Cases contradicting the hypothesis in turn were of two types: (1) in four cases collocated, richly communicating groups developed designs made up of largely independent components; and (2) in twenty-eight cases independent and dispersed contributors collaborated on highly interdependent designs.

The first type of exception already has a theoretical explanation: a collocated, richly communicating group can create a system of independent components or modules by adhering to design rules that impose independence and information hiding across the technical components (Parnas, 1972; Mead and Conway, 1980; Baldwin and Clark, 2000). By contrast, the second set of exceptions poses a deeper theoretical challenge. In traditional development organizations, people have relied on face-to-face communication, physical collocation, and formal authority to coordinate highly interdependent design tasks. The paradigmatic form of organization for developing an interdependent design is a highly interactive team, working in close proximity, employed by a single firm (Allen, 1977; Clark and Fujimoto, 1991; Sanchez and Mahoney, 1996; Chesbrough and Teece, 1996; Baldwin and Clark, 2000). However, the large number of counter-examples revealed by our study raises the question, how are interdependent design decisions and tasks coordinated in the absence of face-to-face communication, physical collocation, and formal authority?

To answer this question, in Section 5, we take a closer look at the twenty-eight cases in which independent contributors developed highly interdependent designs. We find that in all such cases the contributors (1) had compatible motivations and no severe conflicts of interest with respect to the ultimate use of the design; (2) worked in or created a framework that gave them expectations of good faith and some protection from harmful actions by other contributors; and (3) worked to maintain a shared understanding or "common ground" with respect to the design (Clark, 1996; Srikanth and Puranam, 2007). Common ground was sometimes created using analogues of traditional organizational ties—e.g., electronic communication, temporary collocation, and status-based authority. But we also find that independent contributors often coordinated their efforts implicitly by using development tools that made the design-in-progress both transparent and actionable for all members of the group.

The concept of *actionable transparency*, discussed in Section 6, captures the extent to which everyone with an interest in improving a given design has the right and the means to act on it. The concept traces back to Zuboff (1988), who observed that when digital archives constitute a near-perfect surrogate for the activities that generated them, access to those archives provides observers with "universal transparency" into what others are doing (pp. 315, 356-361). *Actionable* transparency requires not just that people can access and make sense of source materials, but that they can also act upon an evolving design. We argue that actionable transparency can serve as a substitute for direct ties between designers, and in its presence, more complex relationships between system design and organizational structure may emerge in lieu of genuine mirroring.

Section 7 summarizes the findings of this paper and concludes.

2 What Is the Mirroring Hypothesis?

The literature that pertains to the mirroring hypothesis commonly draws on two distinct sources for its motivation: (1) the literature on organization design and organizations as complex systems (e.g., Thompson, 1967; Galbraith, 1974; Weick, 1976) and (2) the literature on product design and products as complex systems (e.g. Alexander, 1964; Parnas, 1972, 1978; Ulrich, 1995). Both traditions make use of the concept of modularity described next.

Scholars in the organization-design tradition usually attribute the concept of modularity to Simon (1962, 1981), who used the parable of Hora and Tempus to illustrate the advantage of partitioning a complex problem into parsimoniously linked sub-problems:

> Hora ... put together subassemblies of about ten elements each. ... Hence, when Hora had to put down a partly assembled watch…he lost only a small part of his work, and he assembled his watches in only a fraction of the man-hours it took Tempus. (Simon, 1981, p. 188)

By partitioning the watch into subassemblies, Hora made it easier to cope with the complexity of creating a watch. At the system level, Simon called the property of being divisible into loosely linked subsystems "near-decomposability."

Scholars in the product-design tradition take inspiration from Simon, but also refer to works by Alexander (1964) and Parnas (1972, 1978). Like Simon, Alexander (1964) argued that it is easier to cope with the complexity of a large-scale problem if it is decomposed into parsimoniously linked sub-problems. Parnas (1972, 1978) stressed the benefits of parallelism, arguing that it is easier to split development work across a group if people can work independently and in parallel. To support parallelism, Parnas encouraged developers to avoid sharing assumptions and data. Specifically, he contended that every developer's task assignment, or product module, should be "characterized by its knowledge of a design decision that it hides from all others" (1972: p. 1056).

Information hiding is a fundamental principle underlying the mirroring hypothesis. With information hiding, each module is informationally self-sufficient, hence can be designed independently of the rest of the system. This means that independent individuals, teams or firms can separately design different modules, yet the modules will work together as a whole (Baldwin and Clark, 2000). As a result, there will be a one-to-one mapping from designers (or design teams) to modules (Parnas, 1972, 1978). This one-to-one mapping in turn allows us to formally state the mirroring hypothesis in terms of a structural correspondence between two networks, one technical and one organizational.

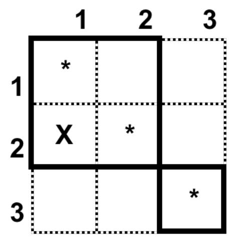
Following Simon (1981), we define the elementary components of a design as decisions about design parameters, or, equivalently, the tasks of making such decisions. The *technical architecture* of a development project captures "what depends on what" in terms of these design decisions or tasks. It is the scheme by which the system's functions and sub-functions are allocated to distinct components (nodes in a network), plus a description of the dependencies (links) between components (Suh, 1990; Ulrich, 1995; Baldwin and Clark, 2000; Whitney et. al., 2004). Design component dependencies are relationships of the form "if the design of Component 1 changes, the design of Component 2 may need to change as well" (Parnas 1972, 1978; Baldwin and Clark, 2000).

The *division of labor* captures "who does what." It is the scheme by which the design decisions (or tasks) are allocated to people or teams (nodes), along with the organizational ties (links), such as communication, collocation, or firm co-membership, between those people. Finally, the *division of knowledge* captures "who knows what." It is the scheme by which design-relevant information is apportioned to designers or teams (nodes) and transferred among them (links).

The mirroring hypothesis posits that, in the design of a complex system, the technical architecture, division of labor and division of knowledge will "mirror" one another in the sense that the network structure of one corresponds to the structure of the others. To visualize what this means, we can represent the technical and organizational networks using matrices.

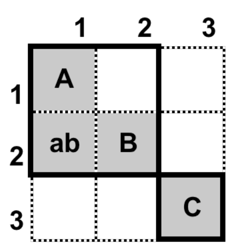
For the technical architecture, a *design structure matrix* (DSM) shows the network of dependencies among the design components (Baldwin and Clark, 2000; see also Donald Steward, 1981, and Eppinger, 1991). Consider a product with N component design decisions. To construct its DSM, we let the rows and columns of an $N \times N$ matrix denote those decisions. Then we set the values in the matrix to capture the dependencies between design decisions: the value of matrix cell d_{ij} is '**x**' if and only if a change in j may require an adaptive change in i. For example, figure 1 depicts a design with three components such that Component 2 depends on Component 1, and there are no other dependencies. The heavy lines in the figure group the components according to their dependencies.

Figure 1 A Design Structure Matrix



The DSM captures dependencies among components in a design, but it does not show who does what, nor how they coordinate their work. We introduce the *labor-knowledge structure matrix* (L-KSM) for these purposes. (It would be more precise to represent the interactions of people and knowledge in two matrices, but to simplify the exposition, we have collapsed them into one.)[1] The L-KSM begins by assigning a person or team to each design task and labeling each cell on the main diagonal accordingly. For example, for the design depicted in figure 1, suppose Alice designs Component 1, Bob designs Component 2, and Carol designs Component 3. In a separate 3x3 matrix with similar rows and columns, we place the uppercase letters A, B and C along the main diagonal as shown in figure 2. These task assignments indicate which agents have primary knowledge of which components: Alice knows about the design of Component 1, Bob knows about 2, and Carol knows about 3.

Figure 2 A Labor-Knowledge Structure Matrix (corresponding to the DSM in Figure 1)



The mirroring hypothesis predicts that, given the dependencies between their design components, Bob and Alice will share one or more explicit organizational ties that enable them to share the requisite

[1] Cataldo et al. (2006) and Sosa (2008) presented a different approach to capture the division of labor. We use the approach above because it allows us to illustrate the mirroring hypothesis as simply as possible.

design information. We denote the (presumptive) presence of an organizational tie between Alice and Bob, by placing the notation "ab" in Alice's column and Bob's row. By comparison, Carol shares no design component dependencies with Alice or Bob, thus the mirroring hypothesis predicts no organizational ties for her. More generally, the mirroring hypothesis predicts that the DSM and L-KSM will have entries in the same cells. The structure of one will correspond to the structure of the other, as can be seen by overlaying figure 2 on figure 1.

Formally defining the mirroring hypothesis in terms of the correspondence between two networks, one technical and one organizational, allows us to specify an ideal test of the hypothesis. In the ideal test, researchers would collect data on the dependencies between the design components and separately collect data on task assignments and organizational ties between designers. Measurable organizational ties might take the form of (1) firm co-membership; (2) geographic collocation; or (3) formal and informal communication. With such data in hand, the researchers could then test to see whether the presence (or absence) of a technical dependency was correlated with the presence (or absence) of a given type of organizational tie. High correlation between technical dependencies and organizational ties would be evidence in favor of the mirroring hypothesis: low correlation evidence against the hypothesis.

Of the studies we surveyed (discussed below), three provided data that permitted this "ideal type" of test (Sosa et. al., 2004; Kratzer et. al., 2008; Bird et. al., 2008). All other studies fell short of the ideal in some way. In the absence of "ideal tests", we might conclude that there is little or no empirical evidence for or against the mirroring hypothesis. However, we believe that this interpretation is overly conservative. As described below, we were able to identify 102 separate studies, in a wide range of settings, in which both design dependencies and organizational ties were observed and their correspondence assessed in a rigorous quantitative or qualitative fashion. Each study constitutes an observation of a case in which the mirroring hypothesis might or might not hold. Thus, even though there are few ideal tests, there is a large and widespread body of empirical evidence pertaining to the hypothesis. We believe it is worthwhile to examine this evidence critically, summarize it statistically, and identify commonalities and differences among the separate studies. We turn to this task next.

3 Empirical Evidence on the Mirroring Hypothesis

In this and the following section, we report the results of a survey and synthesis of the empirical evidence pertaining to the mirroring hypothesis. We first describe how we obtained and analyzed our sample and present an overview of the results. We then discuss in greater detail the findings for three subgroups corresponding to three different levels of organization.

3.1 Sample Selection and Methodology

To identify our sample, we first conducted an electronic search for relevant scholarly works across the full digital publication lives of 19 major journals spanning multiple disciplines. These journals are listed in Table 1. To cast a wider net for the nascent literature on open collaborative development, we repeated the search using the ACM Portal's Guide to Computing Literature. We used a broad set of keywords to perform the search: these appear in Table 2.

Table 1 Journals included in the initial search for empirical evidence

Academy of Management Journal	*Journal of Management Studies*
Academy of Management Review	*Journal of Product Innovation Management*
Administrative Science Quarterly	*Management Science*
California Management Review	*Managerial and Decision Economics*
Harvard Business Review	*Organization Science*
IEEE Engineering Management Review	*Organization Studies*
IEEE Software	*Research in Engineering Design*
IEEE Transactions on Engineering Management	*Research Policy*
Industrial and Corporate Change	*Strategic Management Journal*
Industry and Innovation	

Table 2 The keyword string used in the initial search for empirical evidence

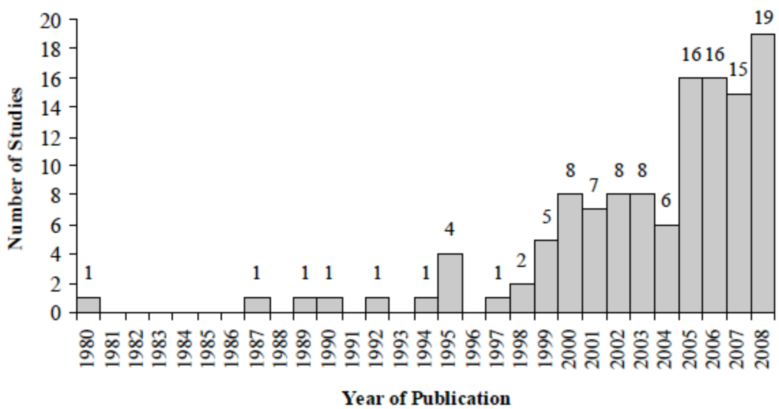
((modular OR modularity OR "product architecture") AND "division of labor") OR
((modular OR modularity OR "product architecture") AND "organizational structure") OR
((modular OR modularity OR "product architecture") AND "industry structure") OR
((modular OR modularity) AND "open source") OR "task partitioning"

We filtered the original search results in two steps. First, we examined titles and descriptions to remove duplicates, "non-results" (e.g. bibliographies, editors' commentaries, book reviews), unsuitable articles (e.g. pure theory works, directions for future research, experiments and simulations, technical "white papers"), and unrelated articles (e.g. empirical studies in the natural sciences). Next, we examined the abstracts of all 141 remaining candidates and removed another 35 due to lack of direct relevance (e.g. the article investigated organizational structure or product architecture but not both).

After this initial filtering, we expanded the sample in two ways. First, in a snowball type search, we scanned the retained articles for relevant studies that were not yet included. We also added studies with which we were familiar. The resulting sample included 130 articles and books, but only 129 separately classified studies, since some works contained more than one study and others contained duplicate discussions of the same study. Of these, 127 were scholarly works; two were books describing the personal experiences of the chief architect of a development project (Mead and Conway, 1980; Colwell, 2006). We believe the two studies based on personal experience constitute valid empirical observations, but our results would not be significantly affected by their exclusion.

Figure 3 shows the distribution of works by publication year. It shows a substantial growth in academic interest in the mirroring hypothesis, with the mean number of works per year more than doubling from 7.4 in 2000-2004 to 16.5 in 2005-2008.

Figure 3 The distribution of works by publication year[2]



We reviewed the full contents of the 129 studies in the surviving set. We found that 27 did not have sufficient data to correlate technical dependencies with organization ties, and dropped these from the sample. As indicated, three studies contained evidence to support an "ideal type" of test; 99 studies fell short of the ideal, but still provided evidence on the degree of correspondence between technical dependencies in the design and the organizational structure of the development organization. Thus our sample comprises 102 unique development projects. Although large, the survey is not exhaustive: some relevant studies undoubtedly escaped our net. Nevertheless, the sample is broad-based and (we believe) representative of scholarly empirical work through the middle of 2009.

We examined and coded the studies' research designs and results to facilitate cross-comparisons. In particular, we noted whether the methodology employed was primarily quantitative (e.g., statistical correlation of design dependencies with organizational variables such as outsourcing/insourcing) or qualitative (e.g., interviews or surveys of perceived correlations between design dependencies and organizational variables). A detailed description of the data, organized by study, is available as an online

[2] Since we collected the data mid-way through 2009, we do not include 2009 in the figure. The sample included nine scholarly works published between January 2009 and May 2009.

appendix <link>. Works included in our final sample are indicated by an asterisk (*) in the references section of this paper.

As indicated, we divided our sample into subgroups according to the level of organization being studied: within firm, across firm, and open, community-based projects. Of the 102 studies, more than half (61%) were across-firm studies, and the remainder split almost evenly between the within-firm (22%) and open collaborative (18%) groups.

We evaluated each study in terms of the question, "*Does the study provide empirical support for the mirroring hypothesis?*" We classified studies using the following categories:

- "*Yes*" denotes that a study's results uniformly supported the mirroring hypothesis—that is, it either (1) showed strong congruence between organizational ties and technical dependencies; (2) found performance advantages associated with congruence; or (3) found performance disadvantages associated with lack of congruence.
- "*No*" denotes that a study's results contradicted the mirroring hypothesis. This category was further divided into two subcategories as follows.
  - Type 1: Contributors with rich organizational ties (e.g., face-to-face communication, collocated, belonging to the same within-firm team) successfully developed a modular technical system with largely independent parts; and
  - Type 2: Contributors with few or no organizational ties (e.g., communicating under constraints, physically and temporally separated, employed by different firms) successfully collaborated on a highly interdependent design.
- "*Partial*" denotes that a study's results provided partial support for the mirroring hypothesis. These cases, which mostly appeared in the across-firm subset of the sample, tended to show that mirroring is not an absolute yes-or-no proposition. For example, a company might modularize its technical designs, but not go so far as to split the corresponding business units into separate firms. Or a firm operating as a systems integrator might adhere to a strict modular partitioning of design responsibilities, but still promote information sharing (not information hiding) across the modules.
- "*Mixed*" denotes that some parts of the study supported the mirroring hypothesis and others challenged it.

3.2 Overview of Results

Table 3 summarizes the results of our analysis broken down by level of organization and methodology. It shows that, across the entire sample, mirroring is a common pattern but by no means universal. Looking at the last two columns of the table, if we count the "Partial" and "Mixed" categories as essentially supportive of mirroring, then a little more than two-thirds (69%) of the studies in the sample supported the hypothesis. Support is strongest in the within-firm group, with 68% clearly supportive and 10% partial or mixed. It is less strong in the across-firm group, with 47% clearly supportive and 28% offering partial or mixed results. Finally, in the open collaborative group, only 28% of the studies clearly support the hypothesis; 11% provide partial support; and 61% contradict it.

Table 3 Results broken down by level of organization and methodology

	Quantitative		Qualitative		Both		All	
	#	%	#	%	#	%	#	%
Within Firm								
Yes	6	60%	6	75%	3	75%	15	68%
Partial	1	10%	0	0%	0	0%	1	5%
Mixed	1	10%	0	0%	0	0%	1	5%
No	2	20%	2	25%	1	25%	5	23%
Total	10	100%	8	100%	4	100%	22	100%
Supportive	8	80%	6	75%	3	75%	17	77%
Across Firm								
Yes	13	57%	9	38%	7	47%	29	47%
Partial	6	26%	3	13%	5	33%	14	23%
Mixed	1	4%	2	8%	0	0%	3	5%
No	3	13%	10	42%	3	20%	16	26%
Total	23	100%	24	100%	15	100%	62	100%
Supportive	20	87%	14	58%	12	80%	46	74%
Open Collaborative								
Yes	3	33%	0	0%	2	29%	5	28%
Partial	1	11%	0	0%	1	14%	2	11%
Mixed	0	0%	0	0%	0	0%	0	0%
No	5	56%	2	100%	4	57%	11	61%
Total	9	39%	2	100%	7	100%	18	29%
Supportive	4	44%	0	0%	3	43%	7	39%

Importantly, the results within each group do not change significantly if we break down the studies according to the methodology employed. In the across-firm group, qualitative studies provide somewhat less support for the hypothesis than quantitative and mixed studies, but, given the small numbers involved, the distributions are not significantly different. Thus we conclude that the results are robust with respect to methodological differences.

Thirty-two (out of 102) studies in our sample did not support the mirroring hypothesis. In four cases, contributors with rich organizational ties created modular systems made up of independent components. However, the bulk of the exceptions (28 out of 32) were cases in which contributors with few observable organizational ties worked collaboratively to perform highly interdependent design tasks. Such cases constituted all of the exceptions in the across-firm and open collaborative subsets of our sample. Table 4 summarizes these results:

Table 4 Summary of results: The "non-supportive" category is split between Type 1 (contributors with rich organizational ties develop a modular system with independent components) and Type 2 (contributors with few organizational ties make highly interdependent design contributions)

	Within-firm		Across-firm		Open Collaborative		All	
	#	%	#	%	#	%	#	%
Supportive of the Hypothesis	17	77%	46	74%	7	39%	70	69%
Not supportive of the Hypothesis	5	23%	16	26%	11	61%	32	31%
Split of Non-supportive Cases								
Type 1	4	80%	0	0%	0	0%	4	13%
Type 2	1	20%	16	100%	11	100%	28	88%

4 Results within Each Group

This section describes our detailed findings within each group.

4.1 Within-firm Results

In the within-firm group of studies, developers are employees of a single firm, thus by definition share one organizational tie. Accordingly this literature focuses on the patterns of communication and collocation within a product development organization. The basic contentions are that the organizational and geographic boundaries of teams should coincide with the technical boundaries of product components and that communication patterns between individuals and teams should coincide with the dependencies between components.

The within-firm literature generally assumes that developers are compatibly motivated but cognitively bounded—that is, it stresses the problem of coordination over cooperation. The organization's key problem is that developers will inevitably experience unintended information asymmetries as a consequence of their bounded abilities as information processors (March and Simon, 1958). Communication then becomes a central concern. If developers fail to communicate on critical design issues, there are two ways to fix the problem: (1) redesign the product to reduce the need for communication (e.g., by standardizing component interfaces); or (2) redesign the organization to facilitate communication (e.g., by collocating developers). Either way, there should be an isomorphic relationship (mirroring) between developers' communication patterns and the technical architecture of the product under development (Conway, 1968; Henderson and Clark, 1990).

As indicated, the mirroring hypothesis was strongly supported in the within-firm studies in our sample. Indeed, a notable fact about this group of studies is that, in virtually all cases, the authors' intent was not to test the hypothesis per se. Instead the studies focused on how to achieve and maintain congruence (mirroring) between product and organization (e.g. Mead and Conway, 1980; Baldwin and Clark, 2000: Ch. 7; Cleidson et al., 2004; Colwell, 2006; Amrit and van Hillegersberg, 2008). The cases supporting the hypothesis showed that mirroring is desirable, but usually difficult to achieve and maintain (e.g., King, 1999; Herbsleb and Mockus, 2003; Cataldo et al., 2006; Cataldo et al., 2008; Gokpinar, 2007; Henderson and Clark, 1990; Morelli et al., 1995; Herbsleb and Grinter, 1999a, 1999b; Ovaska et al., 2003; Sosa et al., 2004; Amrit and van Hillegersberg, 2008; Srikanth and Puranam, 2007, 2008).

One study offered partial support for the hypothesis. In a study of 207 acquisitions by large firms, Puranam et. al. (2009) found (consistent with mirroring) that acquisitions with technologies directly relevant to the acquirers' product line were more likely to be integrated into the acquirer's organization, whereas acquisitions with "stand-alone" technologies were more likely to continue as separate business units. However, when there were high levels of common knowledge (measured by overlapping patents) between target and acquirer, this correlation was significantly weaker. Thus common knowledge appeared to reduce organizational-technical mirroring to some extent.

A study by Sosa et. al. (2004) offered mixed support for the hypothesis. Here, consistent with mirroring, the dependencies among technical subsystems generally corresponded to the patterns of communication between subsystem development teams. However, contra the hypothesis, it was sometimes necessary for teams developing independent subsystems to consult with each other over system-wide issues. This study was one of the three "ideal-type" studies, which directly compared the network structure of design dependencies and organizational ties. The microscopic nature of observations in this study may have led to more complex and nuanced findings than would arise in studies with less precise data.

The within-firm group also provided five clear exceptions to the mirroring hypothesis. Four were what we call "Type 1," that is, cases in which a collocated, highly interactive team within a single firm designed a modular system made up of independent components. For example, Lehnerd (1987) described how, through an intense collaborative effort across functional units, Black and Decker redesigned its product line as a family of modular products assembled from standardized components. In the realm of software, MacCormack et. al. (2006) described how a single, interactive and collocated team within Netscape redesigned the Mozilla browser in a way that significantly reduced the codebase's internal interdependencies and increased its modularity. Such efforts require rigid adherence to design rules to prevent the modularity of the system from breaking down through bilateral negotiations and coordination between module teams (Baldwin and Clark, 2000).

Finally a study by Srikanth and Puranam (2007, 2008) constitutes the lone "Type 2" exception in the within-firm sample. This study looks at software development undertaken by geographically dispersed developers. In 3 of 22 projects, there were few cross-location dependencies, consistent with the mirroring hypothesis. However, contra mirroring, 19 projects exhibited high levels of interdependence across geographic locations. Cross-location interdependencies could not be eliminated because of organizational rigidities (e.g., programmers could not be relocated), technological legacies, and in some cases the newness of the system under development.

4.2 Across-firm Results

In the across-firm literature, developers are distributed across two or more firms, thus by definition, some lack the organizational tie of firm co-membership. This literature seeks to determine whether the distribution of design work across firms corresponds to the underlying pattern of technical dependencies. Specifically, do firm boundaries coincide with module boundaries in the technical architecture? And are formal transactions located where technical dependencies are sparse (Baldwin, 2008)? The theoretical arguments supporting the hypothesis proceed as follows: To make interdependent contributions to a product's design, people must exchange information openly and resolve design conflicts efficiently (Monteverde, 1995). Two well-known schools of thought, transaction cost economics and the knowledge-based theory of the firm, suggest that conflict resolution and information exchange are more easily accomplished within a firm than across firms.

Transaction cost economics (TCE) focuses on the resolution of conflicts of interest. It assumes that the key problem for organizations is to align incentives so that exchanges can take place with minimal risk of opportunistic behavior like hold up or reneging (Williamson, 1985). Such holdups are most damaging when there is a high level of dependency between upstream and downstream agents, i.e., when one or the other lacks an outside option. This in turn argues for allocating interdependent design decisions to agents within a single firm. The TCE approach also stresses the ability of hierarchical organizations (firms) to use formal authority to resolve conflicts (Williamson, 1999).

In contrast, the knowledge-based theory of the firm (KBT) assumes that the key problem for organizations is to facilitate flows of information and assemble requisite stocks of knowledge. Relative to markets, firms have superior capacity for central planning (Alchian and Demsetz, 1972) and rich contextual communication (Arrow, 1974; Monteverde, 1995). For example, Nickerson and Zenger (2004) argue that, when problems are fully decomposable, several independent firms can efficiently work in parallel, coordinated by markets. However, as the sub-problems become interdependent and non-decomposable, it is more efficient to bring the search process within the purview of a single firm. Marengo and Dosi (2005) use a formal model of search on an NK landscape (Kaufmann, 1993) to arrive at similar conclusions.

Notably, although they emphasize different organizational problems, the transaction cost and knowledge-based perspectives arrive at the same conclusion vis a vis mirroring. Effective across-firm collaboration, it is argued, requires a modular technical architecture with information hiding. Clearly delineated task boundaries and codified interface standards are needed to make formal transactions and third-party dispute resolution cost-effective. For this reason, it is more efficient to consolidate interdependent design tasks within a firm than to distribute them across firms (Teece, 1996; Baldwin and Clark 2000: Ch. 14; Langlois, 2002; Baldwin, 2008). Thus, consistent with mirroring, the boundaries of firms will correspond to (mirror) the modules in the technical architecture. (Causality may run in either direction, from technical architecture to firm boundaries or from firm boundaries to architecture.)

Recent contributions to the across-firm theoretical literature do not challenge the mirroring hypothesis, but rather add new concerns to prior theoretical arguments. For example, Veloso and Wolter (2008) point out that obsolescence risk and the need to preserve outside options create forces causing fragmentation: as a result, firms and industries experiencing modular or radical innovations may have a propensity to break apart. Helfat and Campo-Rebado (2009) show that vertically integrated firms may choose to stay integrated even when the underlying technical system is modular, if they anticipate that the designs will become reintegrated at a later date.

Across-firm studies form the largest group in our sample, numbering 62. Of these, 47% fully supported the hypothesis; 23% offered partial support; and 5% provided mixed support. Of the studies supportive of the hypothesis, one important subset involved macroscopic, longitudinal analysis of an industry or industry segment. These longitudinal studies in turn told two stories. In one, the focal industry initially consisted of vertically integrated firms developing technically integrated products. Then a new modular architecture emerged and became the "dominant design" (Abernathy and Utterback, 1978; Tushman and Murmann, 1998; Murmann and Frenken, 2006). The structure of the industry swiftly changed to mirror the new product architecture, with different firms focused on developing different modules of the product. The products examined in these studies included stereo systems (Langlois and Robertson, 1992), computers (Langlois and Robertson, 1992; Baldwin and Clark, 2000: Ch. 1, 14-15), bicycles (Galvin and Morkel, 2001), autos (Ro et al., 2007), banking products and services (Jacobides, 2005; Consoli, 2005), and fantasy role-playing games (Lecocq and Demil, 2006). The second story runs the opposite way: the focal industry initially consisted of specialist firms developing modular components. Then a new integral "dominant design" emerged. The structure of the industry subsequently changed to mirror the integral product architecture, with individual firms developing integrated products.

Firms lacking the capacity to integrate were often forced to exit the industry. The products examined in these studies included bicycle drivetrains (Fixson and Park, 2008) and building facilities (Cacciatori and Jacobides, 2005).

Other studies supportive of mirroring focused on more microscopic levels of analysis, including individual sourcing decisions (e.g., Monteverde, 1995; Novak and Eppinger, 2001; Parmigiani, 2007), supplier-alliance decisions (e.g., Gulati et al., 2005; Sahaym et al., 2007; Tiwana, 2008a, 2008b), outsourcing (e.g., Wüllenweber, 2008), and product innovation events (e.g., Shibata et al., 2005). These studies generally found that greater levels of component complexity and interdependence were correlated with higher probabilities of vertical integration and/or poorer performance among "outsourcers" than "insourcers."

Fourteen studies (23%) provided partial support for the mirroring hypothesis. These in turn fell into two subgroups. In the first, mirroring was observed between the technical architecture of the system and the division of labor, but not between divisions of labor and knowledge (e.g., Prencipe, 1997, 2000; Takeishi, 2001, 2002; Brusoni, 2005; Brusoni et al., 2001; Brusoni and Prencipe, 2006; Tokumaru, 2006; Kapoor and Adner, 2009). As Brusoni et. al. (2001) observed, some firms "know more than they make." Hence, studies in this group support the mirroring hypothesis insofar as they show groups of firms relying on modular divisions of labor to develop modular products, but they also qualify the hypothesis by showing that firms may depend, not on information hiding, but on information sharing, to coordinate their joint efforts.

The second subgroup included studies where firms responded to product modularity by shifting their internal organizational structure in the direction of mirroring, but did so within pre-existing boundaries (e.g., Hoetker, 2006; Jacobides and Billinger, 2006; Gomes and Joglekar, 2008; Parmigiani and Mitchell, 2009). These studies support the mirroring hypothesis in that they show firms responding to greater product modularity with greater organizational modularity, but they also show that firms do not equate the decision to modularize with decisions to divest or outsource. A firm may create a technical architecture with few dependencies across modules and few ties between organizational units, but still keep all units beneath the same corporate umbrella.

Three across-firm studies (5%) fell into the "Mixed" category. These all provided evidence that technical modularity reduced the need or the frequency of cross-firm communication. However, each case also revealed alternative mechanisms of coordination across firms. In one case, Grunwald and Kieser (2007) found that cross-firm communication was reduced by product modularization (consistent with mirroring), but also by systems that helped developers locate and share information. In a second case, Kratzer et. al. (2008) found that the technical architecture changed over time, but patterns of communication were less flexible, hence the degree of mirroring was only moderate in each phase of development. Langner and Seidel (2009) also found temporal instability in patterns of mirroring: the focal firm developed highly modular technical specifications to support contract negotiations with external suppliers (consistent with mirroring), however, once the contract was awarded, the buyer-supplier pair switched to a pattern of rich, frequent information sharing across firm boundaries.

Finally, sixteen studies (26%) were not supportive of the mirroring hypothesis. All exceptions involved separate firms that collaboratively co-designed interdependent systems or sub-systems. The products examined in these studies included autos (e.g., Clark, 1989; Helper et al., 2000; Takeishi, 2001; Mikkola, 2003; Sako, 2004), aircraft and related products (e.g., Argyres, 1999; Miller et al., 1995), building facilities (e.g., Barlow, 2000), software (e.g., Staudenmeyer et al., 2005; Srikanth and Puranam, 2007, 2008), machines and systems for manufacturing (e.g., Bonaccorsi and Lipparini, 1994), computers and computer sub-systems (e.g., Scott, 2000), and semiconductors (e.g., Appleyard et al., 2008). We describe how these firms managed their collaborations in Section 5 below.

4.3 Open Collaborative Results

In an open collaborative development project, product design information, such as software source code, is placed in the public domain. Different individuals and firms contribute voluntarily to the design, according to their own private needs and interests; they self-select their contributions without relying on managers or market prices to guide them (Raymond 1998, 2001; Benkler 2002; von Hippel and von Krogh, 2003; Weber 2004; Lakhani and von Hippel, 2009).

In the open collaborative setting, most developers lack the organizational ties of firm co-membership and collocation. Their opportunities for face-to-face communication are highly restricted or sometimes non-existent. Because open collaborative developers are highly decentralized and geographically dispersed, scholars have predicted (consistent with mirroring) that they will design modular systems made up of largely independent components (e.g. Moon and Sproull, 2000; Kogut and Metiu, 2001; Benkler, 2002; Gacek and Arief, 2004; Fitzgerald, 2005, 2006; Fitzgerald and Ågerfalk, 2005; German, 2005; Lerner and Tirole, 2001, 2002; Osterloh and Rota, 2007). In addition, the mirroring hypothesis predicts that contributors will tend to specialize on particular modules and not range over all parts of the system.

The most striking finding in the open collaborative portion of the sample was the small number of studies that clearly supported the predictions of the mirroring hypothesis (5 studies or 28%) versus the larger number that clearly challenged it (11 studies or 61%). The numbers are too small to be definitive, but the evidence to date indicates that many open collaborative groups do not design particularly modular systems. Furthermore open collaborative groups do not as a general rule divide labor and knowledge along the lines defined by the technical system's modular architecture. Thus the groups do not display classic mirroring (based on information hiding), even when they use a modular architecture for the technical system under development.

The most interesting cases in this branch of the literature involved large groups working together to develop large systems. In these cases (15 out of 18) there was usually some measure of product and task modularization, but it was neither uniform nor complete. Instead, the most common pattern was one in which the developer community self-organized into a “core” group that made numerous large and small contributions across the entire system and a peripheral group that made smaller contributions within the more modular parts of the system (e.g., Mockus et al., 2000, 2002; Koch and Schneider, 2002; von Krogh et al., 2003).

In summary, the mirroring hypothesis was upheld in the majority of studies in our sample. However, the cases contradicting the hypothesis call for further investigation. This is our task in the next section.

5 When and How Do Organizations Break the Mirror?

As indicated, the correspondence between technical dependencies and organizational ties predicted by the mirroring hypothesis can break down in two ways. On one hand, contributors with rich organizational ties may develop a modular product consisting of largely independent components. On the other hand, contributors with highly constrained organizational ties may make interdependent contributions to the design of a single technical system or sub-system.

The first set of exceptions to the mirroring hypothesis has theoretical antecedents in the engineering and product design literatures. Creating a modular system can be an explicit design goal for an individual, team, or firm. Descriptions of how to create modular systems are found in works by Bell and Newell (1971), Parnas (1972, 1978), Mead and Conway (1980) and others. Basically, the designers must have prior knowledge of all implicit or potential dependencies in the technical system to be designed. They can then systematically remove dependencies between modules by establishing design rules (Baldwin and Clark, 2000: pp. 63-86). Thus in theory nothing prevents a tight-knit team or an individual with adequate knowledge from creating a design made up of largely independent components. (In practice, strict adherence to design rules may be hard to achieve.)

The second and larger set of exceptions, in which independent and dispersed contributors make highly interdependent contributions to the same design, raises deeper theoretical questions. As discussed in Section 2, the conventional approach to coordinating interdependent design tasks relies on a tight-knit organizational structure involving rich and unstructured interpersonal communication, physical collocation, and authority-based planning and oversight. Scholars generally perceive some or all of these organizational conditions to be essential for effective coordination. Nevertheless the evidence in the studies we reviewed shows that in a significant percentage of cases, people can and do make highly interdependent design contributions without these organizational supports.

When we looked more closely at the data, we were able to identify three conditions common to all exceptions of the second type. First, the developers had mutually compatible motivations and essentially no major economic conflicts of interest with respect to the design. Second, they operated within frameworks that gave them reason to expect good faith and protected them from malicious or unintentional harm by co-contributors. Last but not least, they established and maintained a common ground from which they shared design information and acted on the evolving design. We discuss each of these conditions in sections below.

5.1 Compatible Motivations

Independent contributors collaborating on highly interdependent design problems exhibited similar motivations for doing so. In all cases, the participants faced resource constraints and thus had the need or a keen desire to access capabilities or resources they did not have themselves. Additionally, in some cases, the participants faced a non-decomposable technological problem, while in others, the participants were developing a brand-new technology whose inherent interdependencies were unknown.

Facing constraints on resources or resource mobility, individual designers and managers were motivated to work across firm or organizational boundaries to gain access to external capabilities. For example, Henkel (2006) found that firms contributed privately developed source code to an open-source software product (embedded Linux) in the hope of drawing complementary contributions from others in the form of bug fixes, code improvements, and code maintenance. In another case, Srikanth and Puranam (2007, 2008) discovered that managers of software projects (at two firms) faced a plethora of resource mobility constraints. Developers were geographically dispersed and could not be relocated. Yet a client might require developers to collaborate with third-party vendors, or impose strict security provisions regarding which developers could access its resources. Such constraints precluded a clean mapping between the technical architecture of products and the geographic and administrative boundaries of the teams working on them.

In addition, independent collaborators working on interdependent designs did not have strong economic conflicts of interest. Sometimes in the open collaborative setting, the participants were competing for a prize, but in such cases, the prize was something of low economic value, for example, a T-shirt (Gulley, 2001, 2004; Gulley and Lakhani, 2009).

Some cases involved a non-decomposable technological problem (e.g., Miller et al., 1995, Argyres, 1999; Parmigiani and Mitchell, 2009). The development of the B-2 “Stealth” Bomber (Argyres, 1999) is a case in point. Because no one firm had all the capabilities and resources needed to develop the aircraft independently, the US Air Force awarded separate contracts to five different firms. However, due to the exacting technical requirements associated with achieving "stealth," the designs of the different firms had to fit together in exceptionally complex and precise ways—thus, the technology inherently "demanded" richly connected contributions.

Other cases involved new—and thus uncertain and poorly understood—technologies (e.g., Brusoni et al., 2001; Takeishi, 2001, 2002; Mikkola, 2003; Appleyard et al., 2008; Garud and Munir, 2008). The ongoing development of "next-generation lithography" (NGL) technology in the semiconductor industry (Appleyard et al., 2008) is an example. Industry leaders, such as Lucent, IBM, and Intel, formed multiple, firm-spanning consortia that advanced novel NGL technologies by coordinating "an unprecedented level of interorganizational cooperation" (p. 419).

5.2 Frameworks Supporting Good Faith

In virtually all exceptional cases, we found that collaborating contributors interacted within frameworks that enhanced expectations of good faith and protected the contributors from malicious harm. The frameworks included formal contracts, long-term relationships, and open-source licensing schemes. These frameworks helped independent contributors cultivate opportunities for mutual gains from collaboration by clarifying the terms of exchange and also by fostering expectations of reciprocal good faith. In some cases, the frameworks did not exist at the time the collaboration began, but grew into place over time. In effect, participants invested in risky alliances in the expectation that their investments would pay off. (See Scott, 2000 and Gilson et. al., 2009 for examples.)

In our sample, contracts came in two forms: lateral–dyadic and hierarchical–multilateral. In the lateral–dyadic form, pairs of firms signaled their intentions to behave in good faith by contracting with one another directly. In such cases, relationship-specific investments and experiences generally fostered expectations of good faith. The supplier relationships of Honda, Nissan, and Toyota (Sako 2004) are cases in point. The automakers cultivated their suppliers' willingness to share knowledge by investing their own resources in their suppliers' capabilities: they created teaching programs to replicate their in-house capabilities at suppliers' facilities, and they set clear rules for sharing the resulting financial gains cooperatively. In another case studied by Gilson et al. (2009), Apple and SCI entered a contract committing the two firms to jointly assess the feasibility of each other's innovations. The contract's enforceable portion consisted of a three-year commitment by Apple to purchase parts manufactured by SCI to remunerate SCI for purchasing Apple's manufacturing facilities. However, the contract also contained unenforceable sections, which, the authors argue, were designed to build the firms' knowledge of each other's capacity for good faith collaboration and dispute resolution (Gilson et al., 2009: p. 466).

In the hierarchical–multilateral form of contracting, multiple contributors signaled their intentions to behave in good faith by contracting with a common sponsor or system integrator. For example, recall the case of the B-2 "Stealth" Bomber discussed above (Argyres, 1999). To meet their respective contractual obligations to their military client, the firms involved voluntarily cooperated with each other.

While relationship-specific investments and experiences play an important role in open collaborative projects (see, for example, Mockus et al., 2000 and von Krogh et al., 2003), open-source licensing affords further support for cooperation in that setting. For example, Garud et al. (2008: pp. 357-358) described the role played by the General Public License (GPL) in the growth of Linux as an open collaborative project:

> A key development was Torvalds' decision to release the kernel under [the terms of the] General Public License (GPL), which mandated that any user modifying or adding to the source code would have to make their own contributions available to everyone else (Stallman 1999). This decision…facilitated the establishment of a stable "generalized exchange system" (Kollock 1999) in which people both contributed to and benefited from the assistance of others.

In other words, the terms of the license provided "the legal and cultural basis" (p. 359) for the cooperation that followed.

5.3 Common Ground

To build on each other's work in the context of an interdependent design, developers require a shared understanding, or "common ground" (Clark, 1996; Srikanth and Puranam, 2007, 2008), regarding (1) how to codify and interpret product design information and (2) what channels and protocols to use for exchanging and discussing design information. In other words, contributors need to be able to "speak the same language" in terms of how they explain and interpret designs. They also need to be able to anticipate when, where, and how to look for design information from their counterparts.

In some cases in our sample, contributors lacked sufficient common ground at the outset of the design process, but subsequently invested time and effort to develop it. The development of the windshield wipers sub-system of the Chrysler Jeep Cherokee (Mikkola 2003) is an example. Chrysler initially outsourced the development of wipers using a modular design approach with explicit interface specifications. The supplier, which had not worked with Chrysler before, developed its first design in isolation based on the specifications. Unfortunately, the supplier did not understand some of the specifications, the design failed, and the supplier had to start over "from scratch" (p. 449). In the second (and successful) design effort, the two firms acted to increase the flow of technical information across their boundary: "Face-to-face meetings…daily phone calls, etc. became a habit" (p. 450). Sharing technical information in this way allowed the engineers at both firms to develop and maintain a common, accurate understanding of the evolving product design.

In the exceptional cases, we found that the mechanisms used by collaborating groups to share design information fell in two classes: (1) those that were analogues of traditional organizational ties; and (2) those that constituted a genuine departure from tradition. The analogues to traditional organizational ties included broadband electronic (instead of face-to-face) communication, temporary physical collocation, and status-based planning and dispute resolution. These devices created a set of direct, though informal organizational ties between designers. (For examples, see in the across-firm case: Clark, 1989; Bonaccorsi and Lipparini, 1994; Miller et al., 1995; Andersen, 1999; Barlow, 2000; Grimaldi and Torrisi, 2001; Mikkola, 2003, Sako, 2004; Brusoni, 2005; Herbsleb et al., 2005; Staudenmayer et al., 2005; Dibiaggio, 2007; Grunwald and Kieser, 2007; Gilson et al., 2009; in the open collaborative case: Mockus et al., 2002; Elliot and Scacchi, 2004; Garud et al., 2008).

In other cases, however, independent contributors did not develop as many direct ties, but coordinated their efforts implicitly, via shared product design and development tools. In these cases, transparency and direct involvement with the design served as a partial or complete substitute for direct ties between designers. (For examples, see in the across-firm case: Argyres, 1999; Andersen, 1999; Scott, 2000; Herbsleb et al., 2005; Grunwald and Kieser, 2007; in the open collaborative case: Mockus et al., 2000; Yamauchi et al., 2000; Gulley, 2001, 2004; Gutwin et al., 2004).

In the next section, we explain how design transparency and especially actionable transparency can break the correspondence between technical architecture and organizational ties that lies at the heart of the mirroring hypothesis.

6 Actionable Transparency

In *The Age of the Smart Machine*, Shoshana Zuboff (1988) observed that the increasingly information-based nature of industrial work has radically increased its "transparency." She argued that, when auto-generated archives constitute a near-perfect surrogate for the activities that generated them, access to those archives provides "universal transparency" into what others are doing (pp. 315, 356-361). In effect, anyone with access to the archives can "see" what's going on without the benefit of direct input or assistance from others. There are in fact three different levels of transparency that collaborating groups can achieve. *Material transparency* denotes the mere disclosure of information. *Conceptual transparency* requires not only that contributors can access the information, but also that they can make sense of it (cf. Wenger, 1990). Finally, *actionable transparency* requires not just that they can make sense of it, but that contributors can act on the design itself (cf. West and O'Mahony, 2008).

Clearly, material transparency is necessary but not sufficient for conceptual transparency (von Krogh et al., 2003; West and O'Mahony, 2008). Material and conceptual transparency do not imply actionability, however. Just because a potential designer understands a design doesn't mean she can act on it. One level of actionability gives developers the right and means to customize their own private copies of the design. A second level of actionability allows developers to combine their individual changes with others', in near real-time, while at the same time guarding against design conflicts and catastrophes. Thus the concept of actionable transparency captures the extent to which *everyone* with an interest in improving the design has the right and means to act on both his or her own copy and the master copy of the design.

It is helpful to consider some examples of actionable transparency in practice. In the across-firm literature, the B-2 "Stealth" Bomber project mentioned above provides one example (Argyres, 1999). The firms involved in the development process shared a system with five key features: (1) it stored all their design information in a shared database; (2) it allowed developers to view all design changes in near-real time; (3) it performed automated checks to ensure all data met rigid rules for how to codify design information; (4) it performed automated batch updates to ensure that all instances of a part were promptly and simultaneously updated; and (5) it enabled the use of automated tools for prototyping. Together these features allowed teams working separately at five different firms in different locations to develop highly interdependent part designs by relying primarily on the system itself to synchronize their efforts.

Another example of actionable transparency—this time in an open collaborative setting—is the MATLAB programming contests described by Gulley (2001, 2004) and Gulley and Lakhani (2009). Each contest requires entrants to use the MATLAB programming language to write a software program to solve a specific problem (e.g., calculating a Fibonacci number) as quickly as possible. However, the contests also have the feature of actionable transparency: programmers can see each other's code, adopt it, and modify it with no explicit constraints. As Gulley (2004: p. 20) explained:

> [T]he unusual feature of this contest is that contestants submit code that is immediately scored, ranked, and displayed for all to see. In fact, ... the contest is specifically designed to encourage participants to steal each other's code.

(Contestants do have norms as to what constitutes "fair play," but these do not preclude using code created by others.) Observing the contest in action, Gulley found that contestants leveraged actionable transparency to make rapid, iterative changes to each other's work. In turn, the winning entry of each contest ultimately manifested the "*tangled effort of dozens of people*" (Gulley, 2004: p. 23; emphasis added).

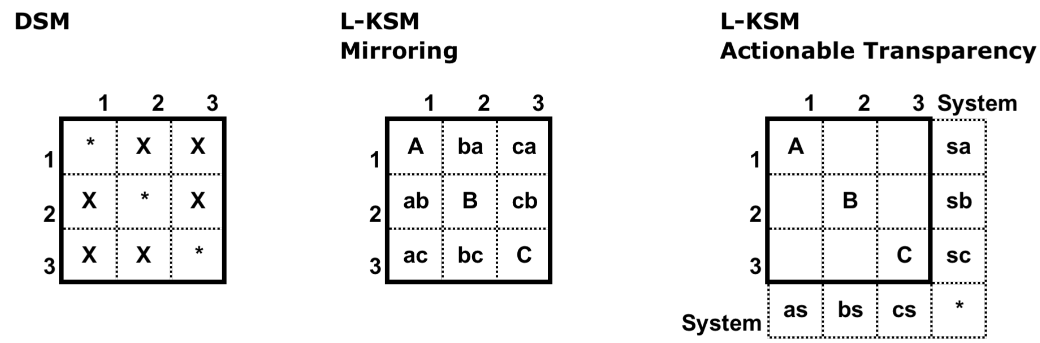
In practice, perfect actionability is rare: almost always, some controls are in place regarding who can change the master design. The need for such controls is reduced when there are automated checks to prevent malicious or unintentional damage, and backtracking schemes to allow previous states of the design to be recaptured. But even with controls in place, actionable transparency supports a level of interdependent collaboration that goes far beyond what is attainable via conceptual transparency alone.

What does actionable transparency achieve that conceptual transparency cannot? As indicated at the start of this paper, innovation is a process in which people define problems and then draw on their stores of knowledge and generate new knowledge to solve those problems (Simon, 1981; Alexander, 1964). Much design-relevant knowledge is tacit and initially inaccurate (Nonaka, 1994), consisting of conjectures of the form: "if I change the design this way, these things will happen." "This way" and "these things" are generally tacit hunches, which are not well articulated even in the mind of the designer (Bucciarelli, 1994). Yet if a design is actionably transparent, conjectures of this type can be tested, evaluated, and new conjectures generated quickly and efficiently. There is no need to make the conjecture comprehensible to another person or persuade someone else that a new idea is worth trying. Interactions between the designer and the design (embedded in a system of archives and test suites) are all that is needed to generate a new trial and new knowledge.

Moreover, if a technical system is actionably transparent to several or many designers, experiments can go on in parallel and concurrently. The designers can then learn about and use each other's changes via the system itself. (See Lakhani and von Hippel, 2009 on "optimistic concurrency" in open-source development.) This form of concurrent, recombinant experimentation creates a very rapid and powerful "generator-test" cycle (Simon, 1981) or "variation-retention-selection" cycle (Campbell, 1969). Such cycles lie at the core of most evolutionary processes, including those found in theories of organizational change and evolutionary economics (Nelson and Winter, 1982; Anderson and Tushman, 1990; Tushman and Rosenkopf, 1992; Nelson, 1995). Thus, in effect, actionable transparency is a form of "facilitated variation" that speeds up design evolution, increasing rates of innovation and improvement for the system as a whole (Baldwin and Clark, 2000; Kirchner and Gerhart, 2005).

Development groups that achieve actionable transparency through systems that provide immediate access to the design, rapid testing and feedback, and near-real-time updating do not require mirroring between the organizational structure and technical design to coordinate their joint efforts. To see this, it is helpful to return to the matrix representations of design and organizational structure introduced in Section 2. In figure 5, the leftmost matrix shows a hypothetical technical architecture in the form of a DSM. The system is fully interdependent, hence the DSM is fully filled in. Three agents, A, B and C, each design one component. The second matrix shows the organizational ties needed if the design process were coordinated via mirroring. In order to resolve their design dependencies, each agent must give and receive information to and from every other agent, hence the structure of the L-KSM mirrors the structure of the DSM exactly. By comparison, the third matrix shows the structure of organizational ties if the design is made actionably transparent by embedding it in a shared information system. Each agent then gives and receives information to and from the system. The agents don't need to interact directly because the system informs them—automatically and promptly—about what others have done. In this case, the organizational ties (the L-KSM) do not mirror the technical architecture (the DSM) at all. (Note: we do not assume that the technical architecture causes the organizational structure or vice versa. If the technical architecture comes first, it can be supported by either form of organizational structure; if the organizational structure comes first, either type of L-KSM can give rise to a fully interdependent DSM.)

Figure 5 Mirroring vs. actionable transparency



The cost of coordinating an organization or team is sometimes equated with its *communication complexity*, that is, the number of messages that must be passed between members in the course of getting the work done (Brooks, 1975). Mirroring and actionable transparency in a shared system have very different implications for communication complexity. Given a team of n agents working on a fully interdependent design, the cost of coordination via mirroring is $n^2 - n$. For the same number of agents and design structure, the cost of coordination via actionable transparency falls to $2n$: each agent only needs to manage his or her information exchanges with the system.

In the presence of actionable transparency, it is common for more complex organizational patterns to emerge in lieu of genuine mirroring. We describe these patterns below.

6.1 Pseudo-alignment

In an instance of *pseudo-alignment*, independent contributors develop a modular product without modularizing their division of labor. At a macro level of analysis, an instance of pseudo-alignment may appear to support the mirroring hypothesis: a decentralized and distributed group will create a product consisting of largely independent components. However, upon closer inspection, pseudo-alignment is not genuine mirroring. Instead of using the product's modular architecture to delimit their tasks and knowledge (as advised by Parnas, 1972), the contributors use modularity to divide the design into relatively small and independent sub-problems. At that point, because of actionable transparency, any member of the group may advance a solution to any sub-problem. Thus if one traces "who does what" within each module, instead of a proper modular division of labor with clear boundaries between the work of each contributor, one finds what Gulley (2004) called "the tangled efforts of dozens of people," within each module of the system.

6.2 Core/Periphery Alignment

In an instance of *core/periphery alignment*, two organizational patterns coexist: there is a small "core" group that (contra mirroring) makes contributions throughout the system, and a larger "periphery" of contributors who make contributions confined to individual modules. The two groups differ in the way they use the system's modularity. Core contributors rely on modularity to reduce the cognitive complexity of their design tasks: However, they do not confine their contributions to specific modules, but range throughout the system. By contrast, peripheral contributors generally target specific features that they care about, which are well-encapsulated within modules, hence easy to modify without triggering changes throughout the rest of the system. Accordingly, the contribution patterns of the core group tend to exhibit pseudo-alignment, while the contribution patterns of the periphery group tend to look more like genuine mirroring.

While the patterns of pseudo-alignment and core-periphery alignment are especially common in the open collaborative setting, we found examples elsewhere. For instance, many of the geographically distributed projects that Srikanth and Puranam (2007, 2008) studied appear to be instances of pseudo-alignment. Likewise, several other authors (e.g., Prencipe, 1997, 2000; Takeishi, 2001, 2002; Brusoni et al., 2001; Brusoni and Prencipe, 2006; Tokumaru, 2006; Kapoor and Adner, 2009) studied an organizational pattern that is a special case of core/periphery alignment, in which the core group is a single firm, or "systems integrator" (Brusoni et al., 2001), that uses a mix of "firm-like" mechanisms and actionable transparency to shepherd the efforts of module suppliers.

7 Conclusion

In this paper we have attempted to present a comprehensive view of the mirroring hypothesis. The hypothesis predicts that there will (or should) be a correspondence between the dependencies in the technical architecture of a complex system and organizational ties between the system's designers. We began by formally defining the hypothesis, and then reviewed the empirical evidence for and against it at three levels of organization: within firm, across firms, and in open, community-based projects. Across all levels, we found much evidence to support the hypothesis, but also a significant number of exceptions that challenged the hypothesis. Support for the hypothesis was strongest in the within-firm sample, less strong in the across-firm sample, and relatively weak in the open collaborative sample.

Exceptions running counter to the mirroring hypothesis took two forms: In four cases, tight-knit teams developed modular systems made up of independent design components. In 28 cases, contributors with few observable organizational ties made highly interdependent contributions to the design of a single technical system or sub-system. The first set of exceptions can be explained using the theory of design rules. For the second set, we identified three conditions common to all cases: (1) compatible motivations; (2) frameworks supporting expectations of good faith; and (3) a shared understanding of the evolving design. We then looked at how the shared understanding was achieved. It was sometimes achieved by creating "virtual" organizational ties: electronic (instead of face-to-face) communication; temporary collocation; and informal and status-based authority. However, a shared understanding can also be achieved and maintained via *actionable transparency*. Development organizations that use actionable transparency often exhibit more complex organizational patterns than simple mirroring, such as pseudo-alignment and core-periphery alignment.

This study is limited by sample and methodology. In terms of the sample, we cast a wide net by surveying numerous publication outlets, and adding works known to us or brought to our attention. However, it is impossible to construct a truly comprehensive sample. The second limitation lies in the fact that our methods of classification were necessarily subjective and more coarse-grained than the studies themselves. Many studies provided complex and nuanced views of the relationship between technical architecture and organizational structure. We are only able to describe the most common patterns, not all patterns.

This study opens up several lines of future research. From a theoretical perspective, the concept of actionable transparency and its relationship with modularity needs to be better understood. Up to now, many scholars have assumed that modularity of a technical system and mirroring necessarily go hand-in-hand. However, as discussed in Section 6, modularity of the technical system can also support high degrees of actionable transparency. A modular architecture reduces the cognitive burden on contributors, making problems smaller and encapsulating them so that changes to one part of the design do not trigger a host of secondary and tertiary changes. When combined with systems that support an ongoing shared understanding of the evolving design, modular architectures can lead to new relationships between the structure of the technical system and the organizational ties between developers. Investigating these relationships is a promising avenue for future research.

In conclusion, the traditional solution to the challenges of coordination and cooperation in development organizations has been to create mirroring between the structure of design and the organizational ties between developers. We have shown mirroring is common in practice, but not universal. Independent, dispersed individuals and firms can successfully collaborate on highly interdependent development tasks if they have compatible motivations and expectations of good faith and can maintain a shared understanding of the evolving design. Actionable transparency can sustain an ongoing shared understanding of a design amongst far-flung contributors, thus is an important means of collaboration in the digital age.

Acknowledgements

An earlier version of this paper appeared as a chapter in the first author's doctoral dissertation, "Three essays on the structure of technical collaboration." The authors extend special thanks to the dissertation committee, Margo Seltzer and Karim Lakhani, for their longstanding support. They also thank Peter Murmann, Mary Tripsas, Michael Tushman, and Eric von Hippel for detailed comments on a previous draft. They gratefully acknowledge the financial support of HBS Division of Research and Doctoral Programs. The authors alone are responsible for errors, omissions, and faulty reasoning.

References

Abernathy, William J. and James Utterback (1978) “Patterns of Industrial Innovation,” *Technology Review* 80:41-47.

Alchian, Armen A., Harold Demsetz. 1972. Production, information costs, and economic organization. *The Amer. Econom. Rev.* **62**(5) 777-795.

Alexander, Christopher. 1964. *Notes on the Synthesis of Form.* Harvard University Press, Cambridge, MA.

Allen, Thomas J. 1977. *Managing the Flow of Technology: Technology Transfer and the Dissemination of Technological Information within the R&D Organization.* MIT Press, Cambridge, MA.

*Almeida, Paul, Jaeyong Song, Robert M. Grant. 2002. Are firms superior to alliances and markets? *Organ. Sci.* **13**(2) 147-161.

*Amrit, Chintan, Jos van Hillegersberg. 2008. Detecting coordination problems in collaborative software development environments. *Inform. Systems Management.* **25** 57-70.

*Anderson, Paul H. 1999. Organizing Internat. technological collaboration in subcontractor relationships: an investigation of the knowledge-stickiness problem. *Res. Policy.* **28**(6) 625–642.

Anderson, Philip and Michael L. Tushman (1990) "Technological Discontinuities and Dominant Design: A Cyclical Model of Technical Change, *Administrative Sciences Quarterly*, 35: 604-633.

*Appleyard, Melissa M., Clara Y. Wang, J. Alexander Liddle, John Carruthers. 2008. The innovator’s non-dilemma: the case of next-generation lithography. *Managerial and Decision Econom.* **29** 407-423

*Argyres, Nicholas S. 1999. The impact of information technology on coordination: evidence from the B-2 “Stealth” Bomber. *Organ. Sci.* **10**(2) 162-180.

Arrow, Kenneth (1974). *The Limits of Organization.* Norton, New York.

Baldwin, Carliss Y. 2008. Where do transactions come from? Modularity, transactions, and the boundaries of firms. *Indust. and Corporate Change.* **17**(1) 155-195.

*Baldwin, Carliss Y., Kim B. Clark. 2000. *Design Rules: The Power of Modularity*. MIT Press, Cambridge, MA.

*Baldwin, Carliss Y., Christoph Hienerth, Eric von Hippel. 2006. How user innovations become commercial products: a theoretical investigation and a case study.” *Res. Policy.* **35**(9) 1291-1313.

Baldwin, Carliss Y. and Eric von Hippel (2009) "Modeling the paradigm shift: from producer innovation to user and collaborative innovation," Harvard Business School Working Paper 10-038 (November).

*Barlow, James. 2000. Innovation and learning in complex offshore construction projects. *Res. Policy.* **29**(7-8) 973-989.

Bell, C. Gordon and Allen Newell (1971). *Computer Structures: Readings and Examples,* New York, NY: McGraw-Hill.

Benkler, Yochai. 2002. Coase’s penguin, or Linux and the nature of the firm. *Yale Law J.* **112**(3) 369-447.

*Bird, C., D. Pattison, R. D’Souza, V. Filkov, P. Devanbu. 2008. Latent social structure in open source projects. *Proc. 2008 ACM SIGSOFT Internat. Sympos. on Foundations of Software Engrg.* Atlanta, Georgia.

*Bonaccorsi, Andrea, Andrea Lipparini. 1994. Strategic partnerships in new product development: an Italian case study. *J. of Product Innovation Management.* **11**(2) 134-145.

*Bowman, we. T., Holt, R. C. 1998. Software architecture recovery using Conway’s Law. *Proc. 1998 Centre For Advanced Studies on Collaborative Research Conf.* Toronto, Ontario.

Brooks, Frederick P. 1975. *The Mythical Man Month: Essays on Software Engineering.* Addison-Wesley, Reading, MA.

*Brusoni, Stefano, Andrea Prencipe, Keith Pavitt. 2001. Knowledge specialization, organizational coupling and the boundaries of the firm: why do firms know more than they make? *Admin. Sci. Quart.* **46**(4) 597-621.

*Brusoni, Stefano, Luigi Marengo, Andrea Prencipe, Marco Valente. 2007. The value and costs of modularity: a problem-solving perspective. *Eur. Management Rev.* **4**(2) 121-132.

Brusoni, Stefano. 2005. The limits to specialization: problem solving and coordination in "modular networks." *Organ. Stud.* **26**(12) 1885-1907.

Bucciarelli, Louis L. (1994) *Designing Engineers*, Cambridge MA: MIT Press.

*Cacciatori, Eugenia, Michael G. Jacobides. 2005. The dynamic limits of specialization: vertical integration reconsidered. *Organ. Stud.* **26**(12) 1851-1883.

Campbell, Donald T. (1965) "Variation and selective retention in socio-cultural evolution". In: Herbert R. Barringer, George I. Blanksten and Raymond W. Mack (Eds.), *Social Change In Developing Areas: A Reinterpretation Of Evolutionary Theory*, pp. 19-49. Cambridge, Mass.: Schenkman.

*Cantamessa, Marco, Maurizio Milanesio, Elisa Operti. 2006. Correlation between design structure matrices to study the structure of the value chain. H. A. ElMaraghy, W. H. ElMaraghy, eds. *Advances in Design.* Springer, London.

*Cataldo, Marcel, James D. Herbsleb, Kathleen M. Carley. 2008. Socio-technical congruence: a framework for assessing the impact of technical and work dependencies on software development productivity. *Proc. 2008 ACM-IEEE Internat. Sympos. on Empirical Software Engrg. and Measurement.* Kaiserslautern, Germany.

*Cataldo, Marcelo, Patrick A. Wagstrom, James D. Herbsleb, Kathleen M. Carley. 2006. Identification of coordination requirements: implications for the design of collaboration and awareness tools. *Proc. 2006 Comput. Supported Cooperative Work Conf.* Banff, Alberta.

Chandler, Alfred D. 1977. *The Visible Hand: The Managerial Revolution in American Business.* Harvard University Press, Cambridge, MA.

Chesbrough, Henry, David J. Teece. 1996. When is virtual virtuous? Organizing for innovation. *Harvard Bus. Rev.* **74**(1) 65-74.

*Chesbrough, Henry. 2003. *Open Innovation: The New Imperative for Creating and Profiting from Technology.* Harvard Business School Press, Boston, MA.

Clark, Herbert H. 1996. *Using Language.* Cambridge University Press, Cambridge, England.

*Clark, Kim B. 1989. Project scope and project performance: the effect of parts strategy and supplier involvement on product development. *Management Sci.* **35**(10) 1247-1262.

Clark, Kim B. and Takahiro Fujimoto (1991) *Product Development Performance: Strategy, Organization and Management in the World Auto Industry*, Boston, MA: Harvard Business School Press.

*Cleidson R. B. de Souza, David Redmiles, Li-Te Cheng, David Millen, John Patterson. 2004. How a good software practice thwarts collaboration: the multiple roles of APIs in software development. *Proc. 2004 ACM SIGSOFT Internat. Sympos. on Foundations of Software Engrg.* Newport Beach, CA.

*Collinson, Simon, David C. Wilson. 2006. Inertia in Japanese organizations: knowledge management routines and failure to innovate. *Organ. Stud.* **27**(9) 1359-1387.

*Colwell, Robert P. 2006. *The Pentium Chronicles: The People, Passion, and Politics Behind Intel's Landmark Chips.* John Wiley & Sons, Hoboken, NJ.

Conner, Kathleen R. and C. K. Prahalad (1996) "A Resource-based Theory of the Firm: Knowledge vs. Opportunism," *Organization Science*, 7(5): 477-501.

*Consoli, Davide 2005. The dynamics of technological change in UK retail banking services: an evolutionary perspective. *Res. Policy.* **34**(4) 461-480.

Conway, Melvin. 1968. How do committees invent? *Datamation.* **14**(4) 28-31.

*Dibiaggio Ludovic. 2007. Design complexity, vertical disintegration and knowledge organization in the semiconductor industry. *Indust. and Corporate Change.* **16**(2) 239-267.

Drazin, Robert and Andrew H. Van de Ven (1985) "Alternative Forms of Fit in Contingency Theory," *Administrative Sciences Quarterly*, 30(4):514-539.

*Elliott, Margaret S., Walt Scacchi (2004). Free software development: cooperation and conflict in a virtual organizational culture. Stefan Koch, ed. *Free/Open Source Software Development*. Idea Group, Hershey, PA.

Eppinger, Steven D. 1991. Model-based approaches to managing concurrent engineering. *J. of Engrg. Design.* **2**(4) 283-290.

Fitzgerald, B. 2005. Has open-source software a future? B. Feller, S. Hissam, K. Lakhani, eds. *Perspectives on Free and Open-source Software*. MIT Press, Cambridge, MA.

Fitzgerald, B. 2006. The transformation of open source software. *MIS Quart.* **30**(3) 587-598.

Fitzgerald, B., Ågerfalk, P. 2005. The mysteries of open-source software: black and white and red All over? *Proc. 2005 Hawaii Internat. Conf. on System Sciences*.

*Fixson, Sebastian K., Jin-Kyu Park. 2008. The power of integrality: linkages between product architecture, innovation, and industry structure. *Res. Policy.* **37**(8) 1296-1316.

Gacek, Cristina, Budi Arief. 2004. The many meanings of open source. *IEEE Software.* **21**(1) 34-40.

Galbraith, J. 1977. Organization design: an information processing view. *Interfaces.* **4**(3) 28-36.

*Galvin, Peter, André Morkel. 2001. The effect of product modularity on industry structure: the case of the world bicycle industry. *Indust. and Innovation.* **8**(1) 31-47.

*Garud, Raghu, Kamal Munir. 2008. From transaction to transformation costs: the case of Polaroid's SX-70 camera. *Res. Policy.* **37**(4) 690-705.

Garud, Raghu, Sanjay Jain, Philipp Tuertscher. 2008. Incomplete by design and designing for incompleteness. *Organ. Stud.* **29**(03) 351–371.

German, D. 2005. Software engineering practices in the GNOME project. B. Feller, S. Hissam, K. Lakhani, eds. *Perspectives on Free and Open-source Software*. MIT Press, Cambridge, MA.

Gilson, Ronald J., Charles F. Sabel, Robert E. Scott. 2009. Contracting for innovation: vertical disintegration and interfirm collaboration. *Columbia Law Rev.* **109**(3) 431-502.

*Gokpinar, Bilal, Wallace J. Hopp, Seyed M. R. 2008. The impact of product architecture and organization structure on the efficiency and quality of complex product development. Working paper, Northwestern University, Chicago, IL.

*Gomes, Paulo J., Nitin R. Joglekar. 2008. Linking modularity with problem solving and coordination efforts. *Managerial and Decision Econom.* **29**(5) 443-457.

*Grimaldi, Rosa, Salvatore Torrisi. 2001. Codified-tacit and general-specific knowledge in the division of labour among firms: a study of the software industry. *Res. Policy.* **30**(9) 1425-1442.

*Grunwald, Roman, Alfred Kieser. 2007. Learning to reduce interorganizational learning: an analysis of architectural product innovation in strategic alliances. *J. of Product Innovation Management.* **24**(4) 369-391.

*Gulati, Ranjay, Paul R. Lawrence, Phanish Puranam. 2005. Adaptation in vertical relationships: beyond incentive conflict. *Strategic Management J.* **26**(5)415-440.

*Gulley, Ned. 2001. Patterns of innovation: a web-based MATLAB programming contest. In *CHI 2001 Extended Abstracts on Human Factors in Computing Systems.* Seattle, Washington.

*Gulley, Ned. 2004. In praise of tweaking: a wiki-like programming contest. *Interactions.* **11**(3) 18-23.

Gulley, Ned and Karim R. Lakhani (2009) "The Determinants of Individual Performance and Collective Value in Private-Collective Software Innovation," Harvard Business School Working Paper (November).

*Gutwin, Carl, Reagan Penner, Kevin Schneider. 2004. Group awareness in distributed software development. *Proc. 2004 ACM Comput. Supported Cooperative Work Conf.* Chicago, Illinois.

*Haefliger, Stefan, Georg von Krogh, Sebastian Spaeth. 2008. Code reuse in open source software. *Management Sci.* **54**(1) 180-193.

Helfat, Constance E. and Miguel Campo-Rembado (2009) "Integrative Capabilities, Vertical Integration, and Innovation over Successive Technology Lifecycles," manuscript, September.

*Helper, Susan, John Paul MacDufie, Charles Sabel. 2000. Pragmatic collaborations: advancing knowledge while controlling opportunism. *Indust. and Corporate Change.* **9**(3) 443-488.

*Henderson, Rebecca, Kim B. Clark. 1990. Architectural innovation: the reconfiguration of existing product technologies and the failure of established firms. *Admin. Sci. Quart.* **35**(1) 9-30.

Henkel, Joachim. 2006. Selective revealing in open innovation processes: the case of embedded Linux. *Res. Policy.* **35**(7) 953-969.

*Herbsleb, J., Grinter, R. 1999a. Architectures, coordination, and distance: Conway's Law and beyond. *IEEE Software.* **16**(5) 63-70.

*Herbsleb, J., Grinter, R. 1999b. Splitting the organization and reintegrating the code: Conway's Law revisited. *Proc. 1999 Internat. Software Engrg. Conf.* Los Angeles, CA, 85-95.

*Herbsleb, James D., Audris Mockus. 2003. An empirical study of speed and communication in globally distributed software development. *IEEE Trans. on Software Engrg.* **29**(6) 481-494.

*Herbsleb, James D., Daniel J. Paulish, Matthew Bass. 2005. Global software development at Siemens: experience from nine projects. *Proc. 2005 Internat. Software Engrg. Conf.* St. Louis, MO

*Herraiz, Israel, Gregorio Robles, Juan José Amor, Teófilo Romera, Jesús M. González Barahona. 2006. The processes of joining in global distributed software projects. *Proc. 2006 Internat. Workshop on Global Software Development for the Practitioner.* Shanghai, China.

*Hoetker, Glenn, Anand Swaminathan, Will Mitchell. 2007. Modularity and the impact of buyer-supplier relationships on the survival of suppliers. *Management Sci.* **53**(2) 178–191.

*Hoetker, Glenn. 2006. Do Modular Products Lead to Modular Organizations? *Strategic Management J.* **27**(6) 501-518.

*Howard, Mickey, Brian Squire. 2007. Modularization and the impact on supply relationships. *Internat. J. of Oper. & Production Management.* **27**(11)1192-1212.

*Jacobides, Michael G. 2005. Industry change through vertical disintegration: how and why markets emerged in mortgage banking. *Acad. of Management J.* **48**(3) 465-498.

*Jacobides, Michael G., Stephan Billinger. 2006. Designing the boundaries of the firm: from "make, buy, or ally" to the dynamic benefits of vertical architecture. *Organ. Sci.* **17**(2) 249-261.

*Kapoor, Rahul, Ron Adner. 2009. What firms make vs. what they know: how firms' production and knowledge boundaries affect competitive advantage in the face of technological change. *Atlanta Competitive Advantage Conf.* Available at http://ssrn.com/abstract=1353550

Kauffman, Stuart A. (1993) *The Origins of Order*, New York: Oxford University Press

*King, Andrew. 1999. Retrieving and transferring embodied data: implications for the management of interdependence within organizations. *Management Sci.* **45**(7) 918-935.

Kirschner, Marc and John Gerhart (2005) *The Plausibility of Life* (New Haven, CT: Yale University Press).

*Kittur, Aniket, Robert E. Kraut. 2008. Harnessing the wisdom of crowds in Wikipedia: quality through coordination. *Proc. 2008 ACM Comput. Supported Cooperative Work Conf.* San Diego, CA.

*Koch, Stefan, Georg Schneider. 2002. Effort, co-operation and co-ordination in an open source software project: GNOME. *Inform. Systems J.* **12**(1) 27-42.

Kogut, Bruce, Anca Metiu. 2001. Open-source software development and distributed innovation. *Oxford Rev. of Econom. Policy.* **17**(2) 248-264.

Kollock, Peter. 1999. The economies of online cooperation: gifts and public goods in cyberspace. Marc A. Smith, Peter Kollock, eds. *Communities in Cyberspace*. Routledge, London.

*Kratzer, J., Hans G. Gemuenden, Christopher Lettl. 2008. Revealing dynamics and consequences of fit and misfit between formal and informal networks in multi-institutional product development collaborations. *Research Policy* 37 1356-1370

Lakhani, Karim R. and Eric von Hippel (2009) "No Managers Required: A case study of collaborative innovation using managerial toolkits" MIT Sloan School of Management Working Paper.

*LaMantia, Matthew J., Yuanfang Cai, Alan D. MacCormack, John Rusnak. 2008. Analyzing the evolution of large-scale software systems using design structure matrices and design rule theory: two exploratory cases. *Proc. 2008 IEEE/IFIP Software Architecture Conf.*

Langlois, Richard N. (2002). "Modularity in Technology and Organization," *Journal of Economic Behavior and Organization,* 49(1):19-37.

*Langlois, Richard N., Paul L. Robertson. 1992. Networks and innovation in a modular system: lessons from the microcomputer and stereo component industries. *Res. Policy.* **21**(4) 297-313.

*Langner, Benedikt, Victor P. Seidel. 2009. Collaborative concept development using supplier competitions: insights from the automotive industry. *Journal of Engrg. and Tech. Management.* **26**(1) 1-14.

Lawrence, Paul R. and Jay W. Lorsch (1967) "Differentiation and Integration in Complex Organizations," *Administrative Science Quarterly*, 12(1):1-47.

*Lecocq, Xavier, Benoît Demil. 2006. Strategizing industry structure: the case of open systems in a low-tech industry. *Strategic Management J.* **27**(9) 891-898.

*Lehnerd, Alvin P. 1987. Revitalizing the manufacture and design of mature global products. B.R. Guile, H. Brooks, eds. *Technology and Global Industry: Companies and Nations in the World Economy.* National Academy Press, Washington, D.C.

Lerner, Josh , Jean Tirole. 2001. The open source movement: key research questions. *Eur. Econom. Rev.* **45** 819-826.

Lerner, Josh, Jean Tirole. 2002. Some simple economics of open source. *J. of Indust. Econom*. **52** 197-234.

*MacCormack, Alan, John Rusnak, Carliss Baldwin. 2008. Exploring the duality between product and organizational architectures: a test of the mirroring hypothesis. Working Paper, Harvard Business School. Boston, MA.

*MacCormack, Alan, John Rusnak, Carliss Baldwin. 2006. Exploring the structure of complex software designs: an empirical study of open source and proprietary code. *Management Sci.* **52**(7) 1015-1030.

March, J., Simon, H. 1958. *Organizations.* Wiley, New York.

Marengo, Luigi and Giovanni Dosi (2005) "Division of Labor, Organizational Coordination and Market Mechanism in Collective Problem Solving," *Journal of Economic Behavior and Organizations* Elsevier, 58(2):303-326.

McCord, Kent R., Steven D. Eppinger. 1993. Managing the iteration problem in concurrent engineering. Working Paper, Massachusetts Institute of Technology, Cambridge, MA.

*McIvor, Ronan, Paul Humphreys, Trevor Cadden. 2006. Supplier involvement in product development in the electronics industry: a case study. *J. of Engrg. and Tech. Management.* **23**(4) 374-397.

*Mead, Carver, Lynn Conway. 1980. *Introduction to VLSI Systems.* Addison-Wesley, Reading, MA.

*Mikkola, Juliana Hsuan. 2003. Modularity, component outsourcing, and inter-firm learning." *R&D Management.* **33**(4) 439-454.

*Miller, Roger, Mike Hobday, Thierry Leroux-Demers, Xavier Olleros. 1995. Innovation in complex systems industries: the case of flight simulation. *Indust. and Corporate Change.* **4**(2) 363-400.

*Miozzo, Marcela, Damian Grimshaw. 2005. Modularity and innovation in knowledge-intensive business services: IT outsourcing in Germany and the UK. *Res. Policy.* **34**(9) 1419-1439.

*Mockus, Audris, Roy T. Fielding, James D. Herbsleb. 2000. A case study of open source software development: the Apache server. *Proc. 2000 Internat. Software Engrg. Conf.* Limerick, Ireland.

*Mockus, Audris, Roy T. Fielding, James D. Herbsleb. 2002. Two case studies of open-source software development: Apache and Mozilla. *ACM Trans. on Software Engrg. and Methodology.* **11**(3) 309-346.

*Monteverde, Kirk. 1995. Technical dialog as an incentive for vertical integration in the semiconductor industry. *Management Sci.* **41**(10) 1624-1638.

Moon, J., Sproull, L. 2000. Essence of distributed work: the case of the Linux kernel. *First Monday*. **5**(11). URL: http://firstmonday.org/issues/issue5_11/moon/index.html

*Morelli, Mark D., Steven D. Eppinger, Rosaline K. Gulati. 1995. Predicting technical communication in product development organizations. *IEEE Trans. on Engrg. Management.* **42**(3) 215-222.

Murmann, Johann Peter and Koen Frenken (2006) "Toward a Systematic Framework for Research on Dominant Designs, Technological Innovations, and Industrial Change," *Research Policy* 35:925-952.

Nelson, Richard R. (1995) "Recent Evolutionary Theorizing about Economic Change," *Journal of Economic Literature,* 33: 48-90.

Nelson, Richard R. and Sidney G. Winter (1982) *An Evolutionary Theory of Economic Change,* Cambridge, MA: Harvard University Press.

Nickerson, Jack A. and Todd R. Zenger (2004) "A Knowledge-based Theory of the Firm—The Problem-Solving Perspective," *Organization Science*, 15(6): 617–632

Nonaka, Ikujiro. 1994. A dynamic theory of organizational knowledge creation. *Organ. Sci.* **5**(1) 14-37.

*Novak, Sharon, Steven D. Eppinger. 2001. Sourcing by design: product complexity and the supply chain. *Management Sci.* **47**(1) 189-204.

*O'Sullivan, Alan. 2006. Why tense, unstable, and diverse relations are inherent in co-designing with suppliers: an aerospace case study. *Indust. and Corporate Change.* **15**(2) 221-250.

Orton, J. D., Weick, K. 1990. Loosely coupled systems: a reconceptualization. *Acad. of Management Rev.* **15**(2) 203-223.

Osterloh, Margit, Sandra Rota. 2007. Open source software development—just another case of collective invention? *Res. Policy.* **36**(2) 157-171.

*Ovaska, Päivi, Matti Rossi, Pentti Marttiin. 2003. Architecture as a coordination tool in multi-site software development. *Software Process: Improvement and Practice.* **8**(4) 233-247.

*Parmigiani, Anne, Will Mitchell. 2009. Complementarity, capabilities, and the boundaries of the firm: the impact of within-firm and interfirm expertise on concurrent sourcing of complementary components. *Strategic Management J.* Forthcoming.

*Parmigiani, Anne. 2007. Why do firms both make and buy? An investigation of concurrent sourcing. *Strategic Management J.* **28**(3) 285-311.

Parnas, David L. 1972. On the criteria to be used in decomposing systems into modules. *Communications of the ACM*. 15(12) 1053-1058.

Parnas, David L. 1978. Some software engineering principles. Hoffman, D. and D. Weiss, eds. *Software Fundamentals: Collected Papers by David L. Parnas.* Addison-Wesley, Boston, MA.

*Paulson, James W., Giancarlo Succi, Armin Eberlein. 2004. An empirical study of open-source and closed-source software products. *IEEE Trans. on Software Engrg.* **30**(4) 246-256.

*Prencipe, Andrea. 1997. Technological competencies and product's evolutionary dynamics a case study from the aero-engine industry. *Res. Policy.* **25**(8) 1261-1276.

*Prencipe, Andrea. 2000. Breadth and depth of technological capabilities in CoPS: the case of the aircraft engine control system. *Res. Policy.* **29**(7-8) 895-911.

Raymond, Eric S. 1998. The cathedral and the bazaar. *First Monday.* **3**(3). URL: http://www.firstmonday.org/issues/issue3_3/raymond/index.html

Raymond, Eric S. 2001. *The Cathedral & the Bazaar Musings on Linux and Open Source by an Accidental Revolutionary.* O'Reilly, Cambridge, MA

*Ro, Young K., Jeffrey K. Liker, Sebastian K. Fixson. 2007. Modularity as a strategy for supply chain coordination: the case of U.S. auto. *IEEE Trans. on Engrg. Management.* **54**(1) 172-189.

*Rusovan, S., Lawford, M., Parnas, D. 2005. Open source software development: future or fad? B. Feller, S. Hissam, K. Lakhani, eds. *Perspectives on Free and Open Source Software*. MIT Press, Cambridge, MA.

*Sahaym, Arvin, H. Kevin Steensma, Melissa A. Schilling. 2007. The influence of information technology on the use of loosely coupled organizational forms: an industry-level analysis. *Organ. Sci.* **18**(5) 865-880.

*Sako, Mari. 2004. Supplier development at Honda, Nissan and Toyota: comparative case studies of organizational capability development. *Indust. and Corporate Change.* **13**(2) 281-308.

Sanchez, R., J. Mahoney. 1996. Modularity, flexibility, and knowledge management in product and organization design. *Strategic Management J.* **17** 63-76.

*Sanderson, Susan, Mustafa Uzumeri. 1995. Managing product families: the case of the Sony Walkman. *Res. Pol.* **24** 761-782

*Schilling, Melissa A., H. Kevin Steensma. 2001. The use of modular organizational forms: an industry-level analysis. *Acad. of Management J.* **44**(6) 1149-1168.

*Scott, Judy E. 2000. Facilitating interorganizational learning with information technology. *J. of Management Inform. Systems.* **17**(2) 81-113.

*Shibata, Tomoatsu, Masaharu Yano, Fumio Kodama. 2005. Empirical analysis of evolution of product architecture: Fanuc numerical controllers from 1962 to 1997. *Res. Policy.* **34**(1) 13-31.

Simon, Herbert A. 1962. The architecture of complexity. *Proc. of the American Philosophical Soc.* **106**(6) 468-482.

Simon, Herbert A. 1981. *The Sciences of the Artificial, 2nd Ed.* Cambridge, MA: MIT Press.

*Sobrero, Maurizio, Edward B. Roberts. 2001. The trade-off between efficiency and learning in interorganizational relationships for product development. *Management Sci.* **47**(4) 493-511.

*Sobrero, Maurizio, Edward B. Roberts. 2002. Strategic management of supplier-manufacturer relations in new product development. *Res. Policy.* **31**(1) 159-182.

*Sobrero, Maurizio, Stephan Schrader. 1998. Structuring inter-firm relationships: a metaanalytic approach. *Organ. Stud.* **19**(4) 585-615.

*Sosa, Manuel E. 2008. A structured approach to predicting and managing technical interactions in software development. *Res. in Engrg Design.* **19**(1) 47-70.

*Sosa, Manuel E., Steven D. Eppinger, Michael Pich, David G. McKendrick, Suzanne K. Stout. 2002. Factors that influence technical communication in distributed product development: an empirical study in the telecommunications industry. *IEEE Trans. on Engrg. Management.* **49**(1) 45-58.

*Sosa, Manuel, Steven Eppinger, Craig Rowles. 2004. The misalignment of product architecture and organizational structure in complex product development. *Management Sci.* **50**(12) 1674-1689.

*Spinellis, Diomidis. 2006. Global software development in the FreeBSD project. *Proc. 2006 Internat. Workshop on Global Software Development for the Practitioner.* Shanghai, China.

*Srikanth, Kannan, Puranam, Phanish. 2007. Coordination in distributed organizations. Working paper, available at SSRN: http://ssrn.com/abstract=939786

*Srikanth, Kannan, Puranam, Phanish. 2008. Co-Presence not communication: how ICT's really enable software services offshoring. Working paper, available at http://ssrn.com/abstract=928072

Stallman, Richard. 1999. The GNU operating system and the free software movement. Chris DiBona, Sam Ockman, Mark Stone, eds. *Open Sources: Voices from the Open Source Revolution*. O'Reilly & Associates, Sebastopol, CA.

*Staudenmayer, Nancy, Mary Tripsas, Christopher L. Tucci. 2005. Interfirm modularity and its implications for product development. *J. of Product Innovation Management.* **22**(4) 303-321.

Steward, D. 1981. The design structure system: a method for managing the design of complex systems. *IEEE Trans. on Engrg. Management.* **28**(3):71-84.

*Sturgeon, Timothy. 2002. Modular production networks: a new American model of industrial organization. *Indust. and Corporate Change.* **11**(3) 451-496.

Suh, Nam (1990) *The Principles of Design*, Oxford UK: Oxford University Press.

*Takeishi, Akira. 2001. Bridging inter- and intra-firm boundaries: management of supplier involvement in automobile product development. *Strategic Management J.* **22**(5) 403-433.

*Takeishi, Akira. 2002. Knowledge partitioning in the interfirm division of labor: the case of automotive product development." *Organ. Sci.* **13**(3) 321-338.

Teece, David J. 1996. Firm organization, industrial structure, and technological innovation. *J. of Econom. Behavior and Organ.* **31** 193-224.

Thompson, James D. 1967. *Organizations in Action: Social Science Bases of Administrative Theory.* McGraw-Hill, NY.

*Tiwana, Amrit. 2008. Does interfirm modularity complement ignorance? A field study of software outsourcing alliances. *Strategic Management J.* **29**(11) 1241-1252.

*Tiwana, Amrit. 2008. Does technological modularity substitute for control? A study of alliance performance in software outsourcing." *Strategic Management J.* **29**(7) 769-780.

*Tokumaru, Norio. 2006. The organizational evolution of innovative activity in the US semiconductor industry: technological specialization and diversification. *Econom. of Innovation and New Tech.* **15**(6) 591-603.

Tushman, Michael L. (1979) "Work Characteristics and Subunit Communication Structure: A Contingency Analysis, *Administrative Science Quarterly*, 24:82-98.

Tushman, Michael L. and Murmann, J. Peter (1998) "Dominant designs, technological cycles and organizational outcomes" in Staw, B. and Cummings, L.L. (eds.) *Research in Organizational Behavior*, JAI Press, Vol. 20.

Tushman, Michael L. and David A. Nadler (1978) "Information Processing as an Integrating Concept in Organizational Design," *Academy of Management Review*, 3: 613-624.

Tushman, Michael L. and Lori Rosenkopf (1992) "On the Organizational Determinants of Technological Change: Toward a Sociology of Technological Evolution," *Research in Organizational Behavior*, 114:311-347.

Ulrich, Karl T. 1995. The role of product architecture in the manufacturing firm. *Res. Policy.* **24**(3) 419-44.

von Hippel, Eric, Georg von Krogh. 2003. Open source software and the "private-collective" innovation model: issues for Organ. Sci. *Organ. Sci.* **14**(2) 209-223.

von Hippel, Eric. 1990. Task partitioning: an innovation process variable. *Res. Policy.* **19**(5) 407-418.

*von Krogh, Georg, Sebastion Spaeth, Karim R. Lakhani. 2003. Community, joining, and specialization in open-source software innovation: a case study. *Res. Policy.* **32**(7) 1217-1241.

Weber, Steven. 2004. *The Success of Open Source.* Harvard University Press, Cambridge, MA.

Weick, Karl. 1976. Educational organizations as loosely coupled systems. *Admin. Sci. Quart.* **21** 1-19.

Wenger, E. 1990. Towards a theory of cultural transparency: elements of a social discourse of the visible and the invisible. Unpublished doctoral dissertation, University of California, Irvine.

West, Joel, Siobhan O'Mahony (2008) "The Role of Participation Architecture in Growing Sponsored Open Source Communities," *Industry and Innovation*, 15(2) 145-168.

Wheelright, Steven C., Kim B. Clark. 1992. *Revolutionizing Product Development.* The Free Press, New York.

Whitney, Daniel E., Edward Crawley, Olivier de Weck, Steven Eppinger, Christopher Magee, Joel Moses, Warren Seering, Joel Schindall, David Wallace (2004) "The Influence of Architecture in Engineering Systems," Engineering Systems Monograph, MIT, Cambridge, MA (March); available at http://esd.mit.edu/symposium/monograph/ (viewed 7/29/07).

Williamson, Oliver E. (1985). *The Economic Institutions of Capitalism*, New York, NY: Free Press.

Williamson, Oliver E. (1991) "Comparative Economic Organization: The Analysis of Discrete Structural Alternatives," *Administrative Science Quarterly,* 36(2):269-296.

Wolter, Claudio and Francisco Veloso (2008) "The Effects of Innovation on Vertical Structure: Perspectives on Transaction Costs and Competences," *Academy of Management Review,* 33(3):586-605.

*Wüllenweber, Kim, Daniel Beimborn, Tim Weitzel, Wolfgang König. 2008. The impact of process standardization on business process outsourcing success. *Inform. Systems Frontiers.* **10**(2) 211-224.

Zuboff, Shoshana. 1988. In the Age of the Smart Machine: the Future of Work and Power. Basic Books, NY.