INSEAD

The Business School for the World

Faculty & Research Working Paper

Component Connectivity, Team Network Structure and the Attention to Technical Interfaces in Complex Product Development

Manuel E.SOSA
Martin GARGIULO
Craig M. ROWLES
2007/68/TOM/OB
(revised version of 2007/08/TOM/OB)

Component Connectivity, Team Network Structure and the Attention to Technical Interfaces in Complex Product Development

Manuel E. Sosa*

Martin Gargiulo**

Craig M. Rowles***

We appreciate the assistance of the engineers at Pratt & Whitney Aircraft for their collaboration during the data collection of this study. We also appreciate the support of the INSEAD/Wharton Alliance Center for Global R&D. We thank the comments from participants in the innovation forum at the Carnegie Bosch Institute conference (Stuttgart, Germany, Fall 2005) and the Wharton Technology mini-conference (April, 2006). We are grateful to Jonghoon Bae, Steven Eppinger, Gokham Ertug, Bruce Kogut, Christoph Loch, and Luk Van Wassenhove for their comments on earlier drafts of this paper. We thank the associate editor and three reviewers of *Management Science* for their insightful comments and suggestions.

* Assistant Professor of Technology Management at INSEAD, Boulevard de Constance, 77305 Fontainebleau Cedex, France.

** Associate Professor of Organisational Behaviour at INSEAD, 1 Ayer Rajah Avenue, Singapore 138676, Singapore.

*** Pratt & Whitney Aircraft, East Hartford, Connecticut, USA

Abstract

The development of complex products poses substantial operational and organizational challenges to established firms. Previous research has shown that coordinating technical interdependencies is vital for the successful development of complex products. We integrate research streams in product development and organizational theory to study the determinants of the capability of teams to attend to technical interfaces in complex product development projects. We hypothesize that the connectivity among product components and teams' communication network structure significantly influence their attention to critical technical interfaces. We show that to effectively examine the role of component connectivity and team network structure is essential to recognize the dual role that teams play as *acquirers* and *providers* of technical information during the design process. We test our hypotheses by examining the network of components of a large commercial aircraft engine and the technical communication network structure of the organization that designs it. Our findings suggest that the connectivity of product components pose capacity restrictions when teams are requested to provide attention to critical design interfaces while teams with sparse communication networks are more capable of aligning their communication patterns with the design interfaces of the components they design.

Keywords: Product Architecture; Modularity; Product Development Organizations Design; Social Networks.

1. Introduction

The development of complex products poses substantial operational and organizational challenges to firms. In the operational (or product) domain, these challenges are met by breaking down complex products into systems, which may be further decomposed into smaller components (e.g., Simon 1981). The product decomposition determines the architecture of the product, which is defined by the way components interface with each other so that the product can fulfill its functional requirements (Ulrich 1995, Ulrich and Eppinger 2004). In the organizational domain, firms meet the challenges of complex product development by assigning each component to a team responsible for its design and for its integration with other components to ensure product functionality (e.g., Clark and Fujimoto 1991). The interfaces among product components define technical interdependencies among design teams, making effective coordination across interdependent teams one of the most critical challenges in complex product development (Smith and Eppinger 1997, Terwiesch et al. 2002, Mihm et al. 2003).

Although attention to technical interfaces is crucial for successful product development, teams typically ignore (or pay marginal attention to) a number of interfaces during the development process (Sosa et al. 2004). Some level of neglect is perhaps unavoidable given the cognitive and resource limitations typically faced by teams (Simon 1947, Ocasio 1997). Lack of attention to non-critical or standardized interfaces may not be ultimately significant (Sosa et al. 2004), but the neglect of critical interfaces can have serious negative consequences for firms. For example, in a study of the semiconductor photolithography alignment equipment industry, Henderson and Clark (1990) found that novel interfaces between existing components were often neglected by design teams, causing established firms to lose their leading position in the market. In a recent study in the automobile industry, Gokpinar, Hopp, and Iravani (2007) found that mismatches between product architecture and organizational structure were positively associated with quality problems. Also in the auto industry, Ford and Firestone lost billions of dollars for poorly managing the interface between the tire design and the vehicle dynamics of the Ford Explorer (Pinedo et al. 2000), and in the aerospace industry, Airbus' development of the A380

"superjumbo" suffered major delays and cost overruns due to lack of attention to some interfaces in the wiring systems at various sections of the plane's fuselage (Gumbel 2006).

Despite the importance that critical technical interfaces may have for the development of complex products, little is known about the factors that affect the capability of teams to attend to such interfaces.[1] This is the topic addressed in this paper. *Why are some teams better than others attending the technical interfaces of the components they design? Is it because of some attributes of the components they design, or because of the structure of the communication networks with other teams in the organization, or both?*

Building on research in product development focused on design iterations and modularity (e.g., Eppinger et al. 1994, Smith and Eppinger 1997, Terwiesch et al. 2002, Mihm et al. 2003; Ulrich 1995, Baldwin and Clark 2000) and on research in social network analysis on the benefits and costs of network structures (e.g., Coleman 1990, Burt 1992), we argue that the ability of a focal team to attend its technical interfaces is affected by both the connectivity of its component and by the closure of its communication network. Component connectivity captures the extent to which a component shares direct and indirect design interfaces with other components within the product. Network closure captures the extent to which the team communicates with other teams that also communicate with one another. Furthermore, we argue that the effect of component connectivity and communication network closure on attention to technical interfaces varies with the role the team plays in each specific interaction. Both product and organizational communication networks are typically directed networks. In the product network, components may receive forces, materials, energy, or information from (or send them to) other components. In the communication network, a team can seek to *acquire information* from other teams whose components can affect the team's component, or it can *provide information* to teams whose components can be affected by the team's component. As an *acquirer*, the team gathers information on in-coming technical interfaces

[1] We broadly follow the definition of *attention* provided by Ocasio (1997: 189) who defines it as "the noticing, encoding, interpreting, and focusing of time and effort by organizational decision-makers on both (a) *issues*... and (b) *answers*." Hence, in the context of this paper, attention happens when system architects identify design interfaces among the components that comprise the product and design teams interact about it.

while as a *provider*, the team provides technical information on its out-going interfaces, typically in response to a request from the affected team.

We argue that the distinction between acquirer and provider roles is fundamental to understand how component connectivity and network closure can affect the team's capability to attend to its technical interfaces. Specifically, we argue that component connectivity makes teams better at attending their in-coming interfaces and worse at attending their out-going interfaces, whereas the density (or closure) of their communication network makes them better at attending their out-going interfaces and worse at attending their in-coming interfaces. In other words, teams designing components with high connectivity are better acquirers and worse providers of information than teams designing components with low connectivity, whereas teams with dense communication networks are better providers and worse acquirers than teams with sparse communication networks. We test these ideas by analyzing the product-organization network associated with the development of a large commercial aircraft engine using probabilistic network models which incorporate both endogenous and exogenous factors (Wasserman and Pattison 1996, Robins and Pattison 2005).

This paper makes two important contributions. First, we take a network analytical perspective to integrate product development and organizational theories to explain a phenomenon that is equally important for both fields—namely, how product structure (at the component level) and informal communication structure (at the team level) affect the capability of teams to attend to technical interfaces in complex product development. Second, we show that the effect of structural properties of the product architecture and of the communication network are contingent upon the role the team plays towards other teams in the system—that is, whether the team seeks to acquire information from another team or is required to provide information to another team.

2. Framework and Hypotheses

The architecture of a product results in identifiable design interfaces between its various components (Henderson and Clark 1990, Ulrich 1995, Sosa et al. 2007). These interfaces, in turn, are the main source

of technical interdependencies between design teams, prompting coordination needs between such teams (Thompson 1967, Galbraith 1973). In the organizational domain, technical interactions between design teams are considered one of the most important and widely used coordination mechanisms to handle technical interdependencies during the development of complex products (Allen 1977, Eppinger et al. 1994, Terwiesch et al. 2002). Figure 1 depicts a network of technical interfaces between four components and a network of technical communications between six teams, four of which are in charge of designing these components and two others in charge of integration activities.

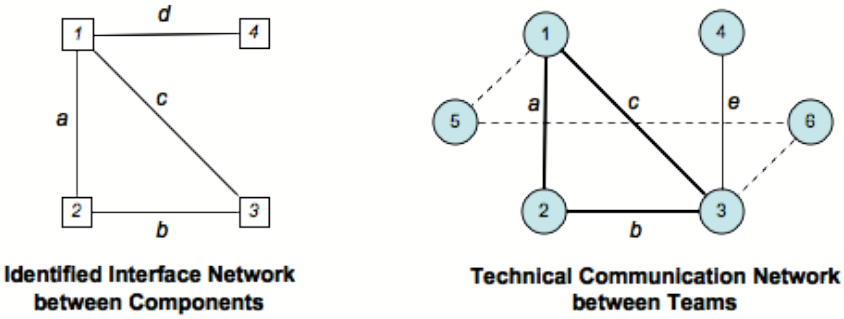

Figure 1. Hypothetical networks of components and teams

By superimposing the network that maps the identified interfaces onto the network that portrays the observable technical communication structure of the design and integration teams, we can identify five types of relationships between design teams (Figure 1). First, *matched interfaces* occur when an identified design interface between two components is matched with technical communication between the teams designing these components (relationships *a, b* and *c* in Figure 1). Second, *unmatched interfaces* occur when an identified design interface is not matched with observed technical communication between the corresponding design teams (relationship *d*). Third, *unmatched interactions* happen when the teams interact in the absence of an identified interface between their respective components (relationship *e*). Fourth, *lack of interaction* occurs when there is neither design interface nor team interaction between two teams (teams 2 and 4 in Figure 1). Finally, *external interactions* occur when teams which are not directly responsible for the design of a component interact with teams designing components. This may be the case with teams that are in charge of overseeing the integration among different aspects of the design but are not responsible for designing any specific component, as is the case with teams 5 and 6 in Figure 1.

Certainly, this is a simplified categorization of coordination relationships between teams, because we are only considering the identified (direct) design interfaces between components as the key source of technical interdependence and assuming that the coordination of those interdependencies occurs mainly through direct technical communications between the respective teams. Technical interdependencies may be also triggered by factors other than direct interfaces between products, or such interfaces may not be identified during the design process. Similarly, observed technical communication in the absence of an identified interface (our "unmatched interactions") may correspond to interfaces that become apparent only as teams work out the detailed designs of their components or may correspond to exchanges of technical information not triggered by a direct design interfaces (i.e. excess communication or communication about system-level issues). Finally, interdependencies between teams may be coordinated through alternative mechanisms, such as interface standardization, communication via computer-aided engineering tools, or through intermediary teams, which can allow organizations to handle identified design interfaces without observable technical interactions between teams.

While any study of how project organizations identify and deal with interdependencies and the corresponding coordination needs between design teams must acknowledge the multiple ways in which organizations approach these challenges, gathering exhaustive data on all such possibilities poses a daunting task. In this study, we try to deal with this difficulty by obtaining an exhaustive map of the design interfaces, as identified by the system architects, and rely on data on technical communication as reported by the teams involved in the design process to study the factors that might affect the team's ability to attend identified interfaces through technical communication. In analyzing the likelihood of observing interfaces matched with technical communication between the respective teams, we also control for the pattern of other observed relationships between the teams, as well as for the possibility that an interface was attended indirectly through an integrator team. Given the crucial role that identified interfaces have in determining technical interdependence between teams, we believe that understanding the factors affecting the likelihood that such interfaces are attended through technical communication is critical to improve the management of complex product development.

For simplicity, the links depicted in Figure 1 are not directed. However, product and organizational networks are typically directed. Components are connected through various types of directed design dependencies caused by flow of forces, materials, energy, and/or information from one component to another (Ulrich 1995, Sosa et al. 2003, 2007).[2] Teams are also linked by directed communication ties, with information flowing from a "provider" team to an "acquirer" team. In any given interaction between two teams, a team can act as an acquirer, a provider, or both towards the other team. The acquirer team typically initiates the interaction by requesting technical information from the provider (Allen 1977, Morelli et al. 1995, Terwiesch et al. 2002). As acquirer, a team has to gather technical information from the teams whose components may affect the acquirer's component. As provider, a team attends to requests for technical information from other design teams whose components can be affected by the provider team's component. Teams play these two roles effectively to the extent that they are able to align their pattern of technical communications with their pattern of design interfaces.

In this paper, we argue that the ability of a focal team to acquire information from (and provide information to) other teams is affected by both the connectivity of its component and the structure of its communication network. The effects of component connectivity and communication network structure, however, differ for each of these two tasks undertaken by teams. We will argue that component connectivity makes a team better at acquiring information on interfaces that may impinge on its own component (the acquirer role) and worse at providing information to teams that can be affected by the focal team's component (the provider role). A team with a dense communication structure (that is, a team communicating with teams who also communicate with one another) is better at providing information to other teams affected by its own component and worse at acquiring information from teams whose component might impinge on the focal team's component. We discuss the mechanisms leading to these observable effects in the next paragraphs.

[2] We use the term design dependency to refer to the various types of linkages between a pair of product components whereas we reserve the term design interface to refer to the aggregated connection (of one or more design dependencies) between such a pair of components.

Effects of component connectivity

Component connectivity is a general structural property that captures how a component shares design interfaces with other components within the product (Sosa et al. 2007). Complex products are typically broken down into sub-systems and components in order to be designed (Simon 1981). An important challenge in the design of complex products is to establish the product architecture. This is done by mapping the functions of the product to its physical components (Ulrich 1995). System architects typically establish the product architecture prior to the detailed design of the product components and their integration into the product as a whole (Ulrich and Eppinger 2004).

Establishing the architecture of the product determines its level of modularity. Modular products include a one-to-one mapping of product functions to product components, resulting in decoupled interfaces between components, whereas integral products included coupled interfaces between components (Ulrich 1995: 422). The common notion associated with modularity is that it decouples otherwise interdependent groups of elements. This decoupling increases the flexibility to adapt to external changes. Modular products are easier to change and upgrade, which facilitates product variety (Ulrich 1995). Modular systems are expected to exhibit greater rates of innovation because parallel development within modules as well as potential recombination across modules (Baldwin and Clark 2000, Ethiraj and Levinthal 2004). Modular product architectures enable the parallelism of design and testing activities (Loch et al. 2001) and modular development processes and organizations are less prone to design iterations and rework (Sanchez and Mahoney 1996, Smith and Eppinger 1997, Mihm et al. 2003). The benefits associated with modular products originate in having decoupled interfaces between components: because teams have to design components that are less connected, they have a more predictable workload.

Modularity, however, can also have drawbacks. Modular organizations tend to focus on the "local" optimization of the internal configuration of sub-systems and components, overlooking product or system-level performance (Ulrich 1995, Baldwin and Clark 2000, Ethiraj and Levinthal 2004). Conversely, products with integral architectures typically perform better on product (or system) level characteristics such as acceleration, weight, energy consumption, aerodynamics, and/or aesthetics (Ulrich

and Eppinger 2004). Teams designing components that are more connected with other components in the product (which is typical in integral architectures) may be more aware of the importance of managing the design interfaces with other components during the detail design phase, giving priority to the optimization of system-level over component (or local) level considerations.

Instead of examining the architecture of the product as a whole, we examine the connectivity of each individual component within the product, which depends on product architecture decisions. The connectivity of a component is a function of its direct and indirect interfaces with other components in the product. Component *A* has a *direct* interface with component *B* if there is a spatial constraint from *A* to *B* or a flow of forces, material, energy, and/or information going from *A* to *B*. If component *B* has another direct interface going to component *C,* then component *A* has an indirect interface with component *C,* mediated by *B*. The connectivity of a component increases with the number of its direct and indirect interfaces with other components in the product. The higher the connectivity of a focal component, the more direct and indirect interfaces it shares with other components in the product. Changes or difficulties originated in other components are likely to propagate to the focal component (Clarkson et al. 2004). This propagation, in turn, can affect the workload of the team designing this focal component in ways that are difficult to predict (Mihm et al. 2003). Conversely, components with low connectivity are less likely to be affected by changes or difficulties in other components within the product.

Building on the advantages and disadvantages associated with modular and integral architectures, we argue that component connectivity can affect the probability of the focal team attending to technical interfaces through two distinct mechanisms. The first mechanism is cognitive orientation (Grabher 1993). High component connectivity drives the team's attention away from internal design issues intrinsic to the component and towards its interfaces with other components. As the connectivity of a focal team's component increases, so does the number of technical interfaces that could affect directly or indirectly this component. This, in turn, is likely to increase the team's awareness of the impact interfaces may have on its own design activities. Insofar as component connectivity reorients the attention of the team towards technical interfaces, it should increase the team's capability to attend them.

The second mechanism through which component connectivity can affect a team's ability to attend to its technical interfaces is excessive workload. From a resource utilization viewpoint, teams can be viewed as "workstations" responsible for designing a physical or functional component of the product. A team's workload is largely determined by the characteristics of its component while the capacity of the team to process its workload determines the team's "delay" in attending to the next "job" in the design task list (Adler 1995, Loch and Terwiesch 1999). The busier the team is, the longer design tasks "wait" to be processed. Component connectivity can have an important effect on the team's workload. Components with more and stronger direct and indirect design interfaces with other components within the product have fewer degrees of freedom and thus are less flexible from a product design perspective (Clarkson et al. 2004). Highly connected components are more likely to generate highly interdependent design tasks, which are the primary source of design rework and iterations during the development process (Smith and Eppinger 1997, Mihm et al. 2003). Conversely, components with fewer and weaker interfaces are more likely to generate a predictable workload structure (Ulrich 1995, Sanchez and Mahoney 1996, Baldwin and Clark 2000). Because designing a highly connected component is more likely to generate unpredictable additional workload for the team, component connectivity should have a negative effect on the team's capability to attend to its technical interfaces. Even though experienced managers may be able to mitigate these negative effects of component connectivity by properly assigning resources to teams, we still expect to observe these negative effects due to the highly unpredictable levels of workload associated with design change propagation through *indirect* interfaces (Mihm et al. 2003, Clarkson et al. 2004).

The previous discussion indicates that component connectivity can have two opposite effects on a team's capability to attend to its technical interfaces. On the one hand, designing a highly connected component should increase this capability by reorienting the team's attention towards the interfaces of its component and away from internal design issues. On the other hand, component connectivity should decrease the team's ability to attend to technical interfaces by increasing its workload. How does the simultaneous influence of these two mechanisms shape the way in which teams allocate resources to attend to its technical interfaces? Because teams are primarily responsible for the design of their own

component, a team designing a highly connected component is likely to focus on the interfaces that can affect its own component, devoting comparatively less attention to interfaces through which its component can affect other components in the product. In other words, we expect that, as component connectivity increases, teams will respond by focusing their attention on the interfaces that can affect its component (the in-coming design interfaces) and away from the interfaces through which its component can affect the work of other teams in the project (the out-going design interfaces). That is, the team is likely to focus on its role as acquirer of information from other teams at the expense of its role as provider of information to other teams.

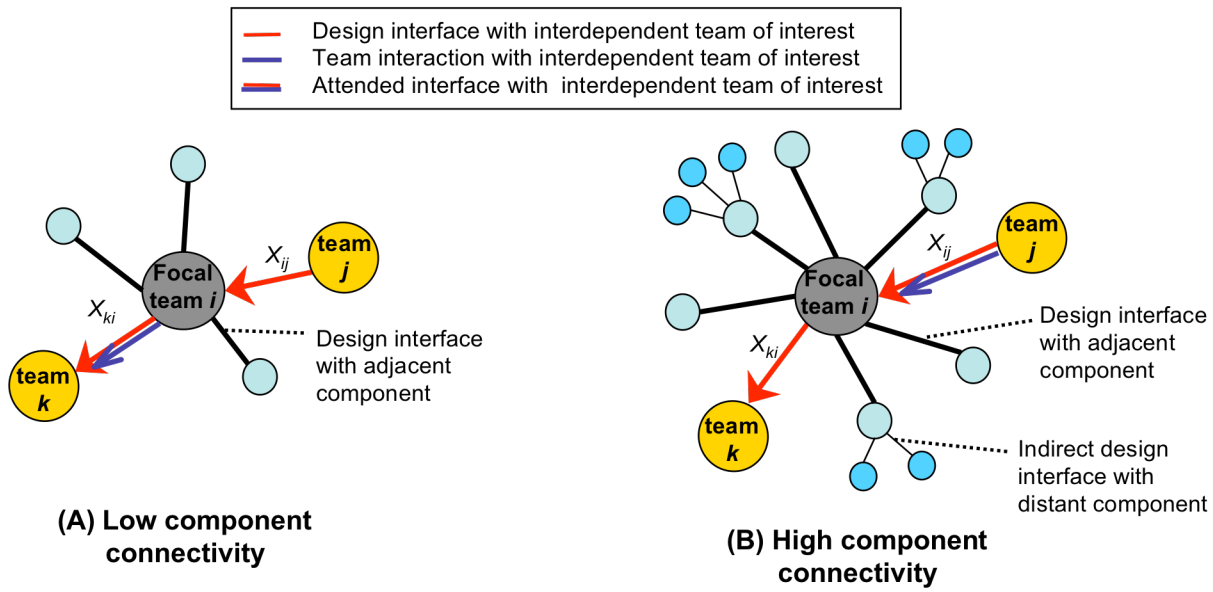

Figure 2. The effects of component connectivity

Figure 2 illustrates how component connectivity affects a team's capability to attend to its technical interfaces, holding other factors constant. As the number of direct and indirect interfaces linking the component of the focal team to other components in the product increases, the team is more likely to focus on these interfaces than on design aspects that are intrinsic to its component (cognitive orientation mechanism), but also to experience unpredictable additional workload (excessive workload mechanism). The team reacts by directing most of its resources to acquiring information on interfaces having an impact on its own component. The focal team *i* in both 2A and 2B need to acquire information from team *j* and provide technical information to team *k*. The only difference between the two sides of Figure 2 is that the

focal team *i* in 2A designs a component with fewer direct and indirect interfaces than the focal team *i* in 2B. As a result, the probability of *acquiring* technical information from team *j* to attend interface X_{ij} is greater in 2B than in 2A, yet the probability of *providing* technical information to team *k* about interface X_{ki} is greater in 2A than in 2B. Empirically, this should result in two testable predictions:

H1a: The probability that a focal team i would acquire information from team j on interface X_{ij} that affects team i's component increases with the connectivity of team i's component.

H1b: The probability that a focal team i would provide information to team k on an interface X_{ki} that affects team k's component decreases with the connectivity of team i's component.

Effects of the communication network structure

The very nature of the design process requires intense collaboration and information sharing among interdependent teams. Organization theory suggests that such interactions can be facilitated (or hindered) by the structure of the informal communication network among such teams. Research on the effects of communication networks on team performance dates back to early studies of R&D organizations that associated both intense and cohesive internal communication with high-performing organizations (Allen 1977) while also recognizing the critical role of "boundary spanners" (Tushman 1977) in linking the team with the rest of the organization. More recently, the emphasis on the role of communication networks—and network structure in particular—on team performance has gained increasing attention from scholars using social network analysis to study how the structure of a team's communication network can affect its performance (e.g., Reagans, Zuckerman and McEvily 2004). Building on social network analysis, we argue that the capability of a team to attend to technical interdependences can be affected by the social structure of its communication network. We focus on one particular property of this structure—namely, the extent to which teams that communicate with the focal team also communicate with each other. Following Coleman (1990) and Burt (2005), we label this structural property "network closure". For a focal team *i* communicating with *n* teams (team *j,* team *k, …,* team *n)*, the more teams *j, k, …, n* communicate with one another, the highest the closure of team's *i* network.

Social network theory suggests that the closure of a team's communication network can affect this team's ability to attend to its technical interfaces through two different mechanisms: collaborative environment and cognitive lock-in. First, network closure is likely to facilitate the collaboration between teams, making it easier to share information and to collaborate in complex tasks (Granovetter 1985, Coleman 1990). Densely connected networks promote information sharing among the actors in the network, foster the emergence of common norms, and facilitate the enforcement of those norms through concerns with reputation (Ahuja 2000, Reagans and McEvily 2003, Obstfeld 2005). The collaborative environment fostered by network closure can help teams attend to technical interfaces by freeing up resources for the team. This is achieved in two different ways. First, by fostering information sharing between teams, network closure facilitates the transfer of information between the teams in the network (Reagans and McEvily 2003). Second, because network closure helps teams enforce collaborative norms, teams in such network structure are more likely to receive help from other teams in the network. This mutual help creates flexible capacity within the group, which can free up resources for teams to attend to technical requests from other teams. Indeed, flexible capacity has been found as an efficient way to reduce delays in the presence of workload variability (Adler et al. 1995, Loch and Terwiesch 1999). Thus, the more a design team is embedded in a cohesive, densely connected communication network, the more it will enjoy the benefits of a collaborative environment, and the more likely it will be able to provide attention to its technical interfaces.

The second mechanism through which network closure can affect a team's capability to attend to its technical interfaces is cognitive lock-in (Gargiulo and Benassi, 2000; Grabher, 1993). It pertains to the information redundancy and rigidity associated with closely-knit communication networks (Burt 1992, 2005). Because information circulates through communication ties between actors, a team *i* that communicates with other teams *j, k…n* that also communicate with one another is more likely to receive similar—or "redundant"—information through multiple channels (Burt 1992, 2004). In addition, the amplification of the signal that results from receiving similar information through multiple channels is likely to increase the likelihood that the team would ignore, disregard, or fail to seek information coming

from outside the connected cluster (Gargiulo and Benassi, 2000). The difficulty of accessing non-redundant information and the insufficient attention paid to signals coming from outside the immediate network that can affect teams with high communication network closure can be especially detrimental for design teams developing complex products. In such cases, the full map of component interfaces among the product's components is seldom clear at the onset. Even in organizations that are well known for their high performance in developing complex products (such as the aircraft engine manufacturer studied here), managers might not be able to map out all the design interfaces between the components of the product during the project planning phase. Hence, part of the responsibilities of each of the design teams is to identify and coordinate the interfaces that can affect its own component. Insofar as network closure can hinder a team's ability to carry out these responsibilities, it should have a negative effect on the team's ability to attend to its technical interfaces.

As it was with the case with component connectivity, the closure of a team's communication network can have therefore two opposite effects on the team's capability to attend to its technical interfaces. On the one hand, the cognitive lock-in associated with a dense communication network should make it more difficult for the team to identify and coordinate the technical interdependencies of the team's component, as the team is less likely to pay attention to events beyond that network. On the other hand, the collaborative environment associated with network closure can make it easier for the team to respond to request for information from other teams. In other words, as the closure of a team's communication network increases, the team should become less effective in its role as acquirer of information from other teams, but more effective in its role as provider of information to other teams.

Figure 3 illustrates how network closure affects the capability of a team to attend to its technical interfaces (all else equal). Although the focal teams in 3A and 3B both exchange technical information with four other teams (in addition to teams *j* and *k*), the focal team in 3B has a more dense communication network around it because four of its contacts communicate with one another for technical reasons. In 3B, the closely-knit communication network structure surrounding the focal team *i* makes it more likely to respond to requests for information on its out-going interfaces, even when the requesting team *k* is not

connected to other teams in the closely-knit communication network. The focal team *i* in 3A does not enjoy the collaborative environment associated with a dense communication network that surrounds the focal team in 3B. As a result, the probability that the focal team *i* could provide technical information to team *k* is lower in 3A than in 3B. Conversely, the cognitive lock-in associated with network closure makes the focal team *i* in 3B less likely to seek information on in-coming interfaces originated in components whose teams are not part of the closely-knit network structure such as team *j*. As a result, the probability that the focal team *i* will acquire information from team *j* is greater in 3A than in 3B. Hence:

H2a: The probability that a focal team i would acquire information from team j on interface X_{ij} that affects team i's component decreases with the closure of team i's communication network.

H2b: The probability that a focal team i would provide information to team k on an interface X_{ki} that affects team k's component increases with the closure of team i's communication network.

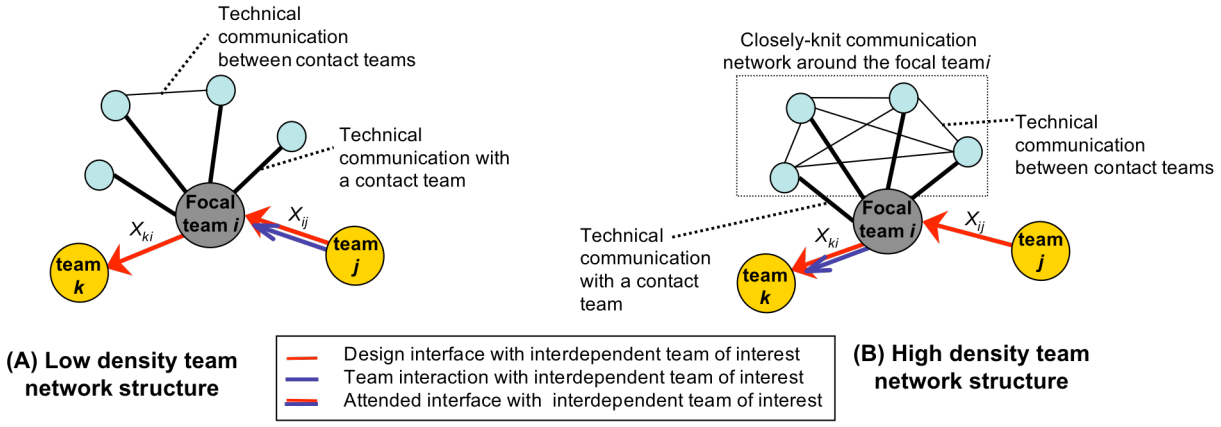

Figure 3. The effects of team communication network structure

3. Data and Methods

We test our hypotheses by studying the detailed design phase in the development of a large commercial aircraft engine at Pratt & Whitney, the PW4098 that equips the Boeing 777 two-engine aircraft. Several factors justified the selection of the project to study. First, the project was a complex design that exhibited explicit decomposition of the engine into systems, and these into components. Second, the assignment of a single development team to each component facilitated the implementation of our research approach. Third, the model studied was the most recent engine program to complete

design and development: almost all team members involved in the detail design phase were still accessible. Finally, the derivative engine studied was part of a family of engines with two new derivative engines already planned, whose development could gain directly from this study.

We captured the design interface network of the 54 components that comprised the engine and the technical communication network of the 60 teams (54 design teams and 6 integration teams) that developed the engine components and evaluated the overall engine performance (Sosa et al. 2003, 2004). We collected data from multiple sources. Product network data was constructed by interviewing several experienced engine architects. Data on communication among teams was gathered by interviewing and surveying key members of the design teams using standard sociometric questions (Marsden 1990).

Product data

Our product data capture the breakdown of the engine structure into eight systems and 54 components, as well as the five types of design dependencies among those components: spatial constraint, structural constraint due to transfer of loads, exchange of material, exchange of energy and exchange of information. Hence, each design interface (i,j) is formed by a vector of five design dependencies which capture the various types of linkages between components i and j. Each type of design dependency x between components i and j was coded as either non-critical ($x_{ij,type} = 1$) or critical ($x_{ij,type} = 2$), depending on the impact on functionality of component i due to its dependence from component j. (See appendix A for further details on the scale used to measure the criticality of design dependencies.) As a result, we define the strength of design interface (i,j) as the summation of its design dependencies. Moreover, to be able to estimate our statistical network models described in the next section we need to codify the resultant valued component network into a trichotomous network with design interfaces taking values of either *NULL* (for $design\ interface_{ij}$=0), *WEAK* (for $design\ interface_{ij} < avg.\ design\ interface$), or *STRONG* (for $design\ interface_{ij} > avg.\ design\ interface$). See Sosa et al. (2004, pp. 1680) for details and justification of such a categorization of design interfaces. Even though such a detailed map of the product architecture was not available at the onset of the project, design teams had a common understanding of the breakdown

of the engine into sub-systems and components, as well as of the relevant linkages between their components and the other engine components.

<u>*Component Connectivity*</u>. The workload associated with the detailed design of each engine component depends not only on its internal structure but also on its direct and indirect connectivity with other engine components. We devise two complementary measures of the connectivity of component *i*: *direct connectivity*, which is equal to the number of adjacent components that are connected with component *i*; *indirect connectivity*, which is a function of both the number of intermediary components and intermediary design interfaces connecting components *i* and *j* through their shortest path. We chose to use two complementary measures of component connectivity, instead of aggregating them into a single measure, to tease out empirically the effect of connectivity with adjacent components versus the effect of connectivity with distant components reached through other intermediary components.

To compute our measure of indirect component connectivity we first symmetrize and standardize our original valued design interface strength data with respect to the strongest design interface within the product. Hence, our standardized measure of design interface between components *i* and *j*, m_{ij}, is the sum of their design dependencies $x_{ij,type}$ in both directions, divided by the maximum design interface detected between any two components in the engine:

$$m_{ij} = \frac{\sum_{type} x_{ij,type} + \sum_{type} x_{ji,type}}{\max\left(\sum_{type} x_{pq,type} + \sum_{type} x_{qp,type}\right)}, \qquad i \neq j\,; \qquad p \neq q;\ \ p=[1,Num.\ of\ comps.];\ \ q=[1,Num.\ of\ comps.]$$

This measure yields a valued symmetric network of product components where $0 \leq m_{ij} \leq 1$ and $m_{ij} = m_{ji}$, with reciprocal design dependencies producing comparatively stronger technical interfaces. We found eight distinct pairs of components whose technical interface reached the maximum standardized design interface strength ($m_{ij} = 1$). To measure total (indirect) connectivity of component *i* with all other engine components with which it has no direct interfaces, we consider how component *i* connects with component *j* through intermediary components in their geodesic (i.e. the shortest path distance connecting

any two nodes in a connected graph). Our measure of indirect component connectivity is a function of our standardized measure of design interface so that indirect component connectivity diminishes with the length of the geodesic that connects any two given components. Hence:

$$Indir\_Comp\_Connectivity_i = \sum_j \sum_p (m_{ip} m_{pj}) \delta_2 + \sum_j \sum_p \sum_q (m_{ip} m_{pq} m_{qj}) \delta_3 + \sum_j \sum_p \sum_q \sum_r (m_{ip} m_{pq} m_{qr} m_{rj}) \delta_4$$

where δ_k is 1 if the geodesic distance is between components i and j is k steps, otherwise is zero. Note that the expression above has three terms because it does not capture direct connectivity between adjacent components and the maximum geodesic distance in our product network data is 4. Figure A2 (in appendix A) illustrates the difference between components with low and high connectivity in the engine studied.

Communication network data

The teams' communication network is defined by their task-related interactions with other teams during the design phase of the project. We capture task-related interactions between the 60 teams —that is, 54 design teams plus six integration teams—involved in the ten-month detailed design phase of the development process. We focused on capturing task-related interactions between design teams, referred to as "coordination-type communication" by Allen (1977) and Morelli et al. (1995). Responses were obtained from team leaders and validated (and corrected, if necessary) by at least one key member of each design team surveyed. By the time of the data collection, all but one of the team leaders were still at P&W, yet eight of them had moved to different groups in the company. We were able to survey key members of 57 of the 60 design teams. The three teams whose direct responses were missing were the interface team at Boeing (a system integration team), one team in the high-pressure compressor system group, and one team in the externals and controls system group. These teams, however, were targets of information requests by other teams in the development process, so we kept them in the network.

Presentations to describe the overall objective of and terminology used during the data collection were made in two sessions to over two-thirds of the respondents, while the other ones were briefed individually afterwards. Specific definitions of survey terms (which had been previously discussed with the program managers) were reviewed and clarified as necessary. Each respondent was asked to identify

(from the team roster for this project) the other design teams from which his team requested technical information during the detailed design phase of the project. Then we asked respondents to assess in a single scale the frequency and importance of their "peak" technical interactions during the detail design period. Consistent with Marsden and Campbell (1984), using a scale with both criticality and frequency of interactions helped respondents focus their responses on information exchanges that actually took place (i.e., the "way it was", not the "way it should have been"). Similar to previous research in technical communication (e.g., Allen 1977, Morelli et al. 1995, Lomi and Pattison 2006), we focus on the presence or absence of *significant* information exchange as the binary variable of interest. We define *significant* information exchange as the technical communication that was relevant during the design phase due to their criticality and/or frequency (See additional details in the appendix B). Based on these data, we reconstructed the informal technical communication network among the 60 teams involved in the project. The resulting communication network contains 680 nonzero communication ties among the 60 teams — that is, 19% of all possible communication ties were present.

Team's communication network. We measure two important structural attributes of the communication network structure of each design team: First, the *communication network degree* (g_i) of team i which is equal to the number of other teams in the network that are in direct contact with team i (i.e. the number of teams that receive or send technical information from or to team i). Second, the level of network closure around focal team i, or team network density, is measured considering the directed communications among the contacts of the focal team. Hence,

$$team\_density_i = \frac{\sum_j \sum_p (t_{ip} t_{pj}) \delta_{ij}}{(g_i(g_i - 1))} \qquad i=1,..,54; \qquad j \neq p=1,..,60$$

where t_{ip} is 1 if team i reported having received significant technical information from team p which also reported having received technical communication from team j. The indicator δ_{ij} is 1 if team i exchanged technical information with team j (either by providing or receiving technical information), otherwise it is zero. Finally, g_i is the communication network degree of team i as previously defined.

Attention to technical interfaces

The organizational phenomenon we study in this paper is the capability of team *i* to align its technical communication patterns with the design interface patterns of the engine component they design. Such a capability depends on the role played by team *i* as an acquirer or provider of information relevant to its interfaces with team *j*. Hence, our variable of interest is the simultaneous occurrence of directed team interactions and directed design interfaces. In our setting, 349 design interfaces out of the 569 interfaces identified by system architects were "matched" by team interactions. Moreover, 250 of the identified design interfaces (44%) were categorized as *strong* interfaces (of which 180 were attended by team interactions) while 319 design interfaces (56%) of the identified interfaces were categorized as *weak* interfaces (of which only 150 were attended by team interactions).

Control variables

In order to test how the structural properties of both engine components and design teams influence the attention to technical interfaces, it is necessary to control for other component and team attributes which can influence the occurrence of "matched" interfaces.

Component redesign. An important factor that impacts both a team's workload and the need to coordinate interfaces with other engine components is the novelty of its component. The more novel the component with respect to a prior generation, the more likely it will generate extra work for the team and the more likely it will affect the interfaces with adjacent components. On the other hand, less novel components may carry over a significant fraction of design content from the previous engine model which decrease the demand for attention to interfaces. We capture this by measuring component redesign as the percentage of actual novel design content in a component, relative to the design of this component in the previous version of the product. As it is physically impossible to determine the exact amount of redesign in a component, we relied on percentage estimates by the design teams of the amount of redesign for their respective components in comparison to the prior existing engine.

Component complexity. Another component attribute that may impact the team's workload and the level of attention placed on design issues related to the internal configuration of the component itself

(rather than its interfaces with other components) is the internal complexity of the component. We measure the complexity of components by estimating the number of distinct parts included on each component. To do so, we rely on the experience of one of the authors in the project, who is a design expert with substantial experience in similar engine programs and who also reviewed the design work for this particular project.

Team size. The effects of component connectivity on a team's ability to attend to technical interdependencies and the team's ability to engage in communication with other teams might be affected by the resources available to the team. Because managers are likely to allocate resources to teams taking into account the workload that results from characteristics of the component and its interfaces with the other engine components, the effects of component and communication network structures may be confounded with the effects of the resources available to the team. To control for this possibility, we include a four-point discrete variable accounting for the manpower resources allocated to teams. Although we were unable to collect precise data on team size—which was also variable throughout different stages of the process—we obtained a qualitative assessment of team size based on the experience of one of the authors in this project as well as other similar engine programs

Communication through integration teams. Although the two communication network variables—team network size and density—are measured on the communication network for the 60 teams in the project (that is, the 54 design teams and the 6 integration teams), design teams may differ in the extent to which they communicate indirectly through integration teams. Indeed, the role of such teams is to collect and pass technical information among design teams to determine engine-level performance in areas such as rotor dynamics and aerodynamics. In some cases, indirect communication through integrating teams may allow two design teams to attend to critical design interfaces without engaging in direct communication. If there is heterogeneity in the team's reliance on indirect communication to attend to technical interfaces, and if this heterogeneity is related to occurrence of matches of design interfaces and team interactions, then our hypotheses testing would not be accurate. To eliminate this possibility, we control for the number of teams j that team i reaches through one or more integration teams in two steps

in the communication network. Specifically, a two-step communication tie between design teams *i* and *j* occurs whenever team *i* communicates with integration team *k* and this team in turn communicates with team *j,* but there is no direct communication tie between *i* and *j*.

Table C1, included in appendix C, shows descriptive statistics and correlation coefficients of component and team level variables used in our statistical analysis.

4. Statistical Network Analysis of a Complex Product-Organizational Network

Our variable of interest is the attention to technical interfaces which is captured by the simultaneous occurrence of both directed design interfaces and directed team interactions. Modeling this variable, however, presents difficulties that require consideration. These difficulties result from the fact that ties in either network are not independent; instead, they are conditional on the presence of other ties. That is, the probability of having team *i* obtaining technical information from team *j* when component *i* depends on component *j* is not only a function of exogenous factors such as the size of the interacting teams or the level of redesign of the interdependent components. Rather, it also depends on endogenous factors such as the overall tendencies to reciprocate ties or to form transitive triangles in both the product architecture and organizational communication networks. As a result, our statistical modeling approach must appropriately account for the presence of both exogenous and endogenous factors.

Fortunately, advances in probabilistic network theory provide us with statistical models that allow us to tackle this challenge effectively (Pattison and Robins 2007). More specifically, we build exponential random graph (p^*) models of our binary network data (Wasserman and Pattison 1996, Pattison and Wasserman 1999, Robins et al. 1999, Robins et al. 2001, Robins and Pattison, 2005, Snijders et al. 2006). These models include not only endogenous effects associated with the most relevant network configurations present in our product-organizational network but also the effects of component and team attributes on the simultaneous occurrence design interfaces and team interactions.

The general statistical approach is to build probabilistic network models in which conditionally dependence among ties is allowed. Based on these types of models, Sosa et al. (2004) formulated p^*

models (Wasserman and Pattison 1986, Pattison and Wasserman 1999, Robins et al. 1999) to test how product and organizational factors associated with the ties themselves would affect the likelihood of observing a matched interface. Our work extends the models of Sosa et al. (2004) by including node-level attributes such as component connectivity and team network closure to test how they affect the likelihood of a design interface being attended by its corresponding team interaction (Robins and Pattison 2005). We carry out our statistical analysis in three steps (refer to appendix D for details on each step).

Step 1: Formulate a base *p model for our product-organization network**

To be able to test whether our observed binary network of matches interfaces contains a significantly higher number of network configurations some of which correspond to our hypothesized effects, we start with a general exponential random graph (p^*) model of our bivariate network (Pattison and Wasserman 1999, Robins et al. 1999, Robins and Pattison 2005). This model establishes that certain network configurations such as reciprocal ties or transitive triangles influence the establishment of the ties observed in our network data. Consistent with previous work in modeling exponential random graphs for social networks in technical organizations we reduce the general model to a homogeneous Markov random graph (Lomi and Pattison 2006). Hence, our base p^* model takes the following form:

$$\Pr(\mathbf{X}=\mathbf{x}) = \exp\begin{bmatrix}(\theta_1 L_1(\mathbf{x}) + \sigma_1 S_1(\mathbf{x}) + \rho_1 M_1(\mathbf{x}) + \tau_1 T_1(\mathbf{x})) + \\ (\theta_2 L_2(\mathbf{x}) + \sigma_2 S_2(\mathbf{x}) + \rho_2 M_2(\mathbf{x}) + \tau_2 T_2(\mathbf{x})) + \\ \theta_{12} L_{12}(\mathbf{x})\end{bmatrix} \Big/ \kappa \qquad (1)$$

where $\mathbf{X}$ is a random network with possible ties $X_{ij,m}$ (where m=1 correspond to ties between engine components and m=2 correspond to ties between design teams, so that $X_{ij,m}$ =1 if there is a directed tie from node i to node j); $\mathbf{x}$ is a realization of $\mathbf{X}$; κ is a normalizing quantity very difficult to calculate analytically in complex networks like ours. $L(\mathbf{x})$, $S(\mathbf{x})$, $M(\mathbf{x})$, and $T(\mathbf{x})$ are the number of ties, 2-stars (in-stars, out-stars, and mixed stars), mutuals, and triangles in the product and organizational networks, respectively. Our model also includes the network statistic $L_{12}(\mathbf{x})$ which captures the number of "matches" of design interfaces and team interactions that occur in our bivariate network. Finally, θ, σ, ρ, and τ are corresponding parameters to be estimated. Of particular importance is the θ_{12} parameter because it

captures the tendency of "matches" to occur in our network. Indeed, our statistical challenge is to test how properties of the engine components and the design teams interact with our parameter of interest, θ_{12}. In addition to our main bivariate configuration of interest, $L_{12}(\mathbf{x})$, our model controls for the presence of other bivariate configurations such as "exchanges" and "three-cycles" as well as the effects of group boundaries and sub-system modularity (Sosa et al. 2003, 2004). Refer to appendix D for details.

Step 2: Extended base *p model with component and team structural attributes**

Our base *p** formulation models the overall tendency to find "matches" of design interfaces and team interactions in our product-organization network. However, we are interested in testing how component and team attributes moderate such a tendency. Based on network models for social selection processes in which individual attributes influence the formation of network ties (Robins et al. 2001, Robins and Pattison 2005), we include interaction effects of $L_1(\mathbf{x})$, $L_2(\mathbf{x})$, and $L_{12}(\mathbf{x})$ with the control variables and the component connectivity and team network structure variables of interest. Because any dyad in a directed network like ours has an "acquirer" and a "provider", we include interaction effects with component and team attributes defined for both the "acquirer" and the "provider" of each dyad. Our hypothesis testing is then formalized by examining the significance of the parameters corresponding to interaction effects between $L_{12}(\mathbf{x})$ and component connectivity variables (for *H1*), and between $L_{12}(\mathbf{x})$ and team communication network variables (for *H2*). Hence,

H1a: $\theta_{12,component\_connectivity\_acquirer} > 0$ ***H1b:*** $\theta_{12,component\_connectivity\_provider} < 0$

H2a: $\theta_{12,team\_closure\_acquirer} < 0$ ***H2b:*** $\theta_{12,team\_closure\_provider} > 0$

Step 3: Estimate parameters of our *p* models*

Estimation of the parameters of our models is difficult because the likelihood function depends on the normalizing quantity κ, which is just too difficult to calculate for most networks (Wasserman and Pattison 1996, Pattison and Robins 2007). Although there have been significant progress in developing Markov Chain Monte Carlo Maximum Likelihood Estimation (MCMCMLE) procedures (e.g. Snijders et al. 2006), the application of these methods to multiplex networks with directed valued relations as the one

we will consider below is still under development. As a result, we use the maximum pseudolikelihood estimation (PLE) procedure elaborated by Wasserman and Pattison (1996) to estimate (albeit approximately) the parameters of our p^* models. This approach is consistent with recent work that estimates p^* models of complex multiplex organizational networks like ours (Lomi and Pattison 2006).

We estimate two sets of p^* models. First, we consider design interfaces to be equally critical and therefore both the team communication and the component interface networks are formed of binary ties. With these models we simply test how the connectivity of the components and the communication network structure of the teams moderate the net propensity to align design interfaces and team interactions. Second, we consider the criticality of design interfaces and estimate p^* models with a product-organization network with trichotomous ties among the engine components (NULL, WEAK, and STRONG design interfaces). This second set of models allows us to examine how the hypothesized effects are contingent upon the level of criticality of design interfaces. Pseudolikelihood estimates (with approximate standard errors) of the parameter of interest of these two sets of models are shown in Tables 1 and 2, respectively. Because the dependent variable of our models is the probability of reproducing the observed product-organization network based on the parameters included in the models, a positive parameter indicates a positive tendency for the associated configuration to occur in the observed network, whereas a negative parameter suggests a negative tendency to observe such a configuration.

First, we report the overall tendency in our data to align design interfaces and team interactions: $\boldsymbol{\theta_{12}}$, in Table 1 for binary relations, and $\boldsymbol{\theta_{weak2}}$ and $\boldsymbol{\theta_{strong2}}$, in Table 2 for the two levels of design interface strength considered in our analysis. Then, we show the interaction effects with the control variables (in Table 1 only) and with the structural variables associated with our hypothesized effects. Note that for each interaction effect reported, our models also include interaction effects with the component interface and team communications networks (not shown here for space limitations) so that the parameters estimated indicate the *net* effect on the propensity to observe an attended interface.

Results

Table 1 presents models with the effects for parameters that may affect the likelihood of observing "matches" of design interfaces and team interactions, disregarding the differences in strength across design interfaces. We discuss the results of the full model (1.4), referring to partial models whenever appropriate. Our data displays a baseline tendency for "matches" to occur (parameter $\boldsymbol{\theta_{12}}$), although this is not statistically significant after including the effects of our main variables. An interface is less likely to be attended if the "acquirer" team in the dyad designs a more internally complex component, as indicated by the negative parameter $\theta_{12,complexity\_acquirer}$. Yet, component complexity actually makes an interface more likely to be attended if the team designing the complex component is the provider ($\theta_{12,complexity\_provider} > 0$), although this effect is significant only in models where the connectivity of the component is held constant (Models 1.2 and 1.4). This pattern of results suggests that the internal complexity of components biases the attention of teams towards internal design issues and away from their interfaces. Teams designing complex components may still pay attention to interfaces when they are required to do so by other teams, but they are more likely to neglect interfaces affecting their own component.

Our analysis offers only partial empirical support for *H1a*. We find some evidence suggesting that the likelihood that an interface would be attended increases with the number of direct interfaces of the acquirer team in the dyad ($\theta_{12,direct\_connect\_acquirer} > 0$ in Model 1.2, but not significant in the full model). As for *H1b*, the results in Table 1 do not offer empirical evidence to support it. That is, teams designing less connected components were *not* more likely to provide more attention to their out-going interfaces.

Models 1.3 and 1.4 include the interaction effects with team communication network structure. The parameters for direct contacts indicate that the number of teams a focal team acquires information from (or provides information to) does not affect the likelihood that the team will attend to its interfaces. Supporting *H2a*, an interface is less likely to be attended if the team playing the "acquirer" role is embedded in a dense communication network ($\theta_{12,\boldsymbol{team\_density}\_acquirer} < 0$). Yet, an interface is more likely to be

attended if the "provider" team is the one embedded in a dense network ($\theta_{12,\textbf{team\_density}\_provider} > 0$). This effect is not significant in Model 1.3 ($p < 0.15$) but becomes significant at the .05 level in the full model.

The analysis reported in Table 1 does not distinguish among type of interfaces. Yet, it is reasonable to presume that the attention to an interface is likely to be affected by the strength of that interface. Attending to strong design interfaces is more likely to be critical for the teams (and for the product) than weak ones. To explore whether the hypothesized effects are contingent on the criticality of the interfaces, we conducted the analysis distinguishing between weak (below average) and strong (above average) interfaces. The results are reported in Table 2 (interaction effects with the control variables are included but not reported in the table; see Appendix D for details).

Model 2.1 shows that the overall tendency to align communication patterns with interfaces is considerable stronger for strong interfaces ($\theta_{\textbf{strong2}} > \theta_{\textbf{weak2}}$), which is consistent with the idea that, facing capacity constraints, teams are more likely to focus on strong interfaces (Sosa et al. 2004). In line with *H1a,* Model 2.2 shows that strong design interfaces are more likely to be attended when the "acquirer" team in the dyad designs a component with direct interfaces ($\theta_{strong2,\ \textbf{direct\_connect}\_acquirer} > 0$). Strong interfaces, however, seem less likely to be attended if the provider team in the dyad designs a component with many direct connections ($\theta_{weak2,\textbf{direct\_connect}\_provider} < 0$; $p < 0.15$). These effects, however, disappear after the effects of team communication network structure are included in Model 2.4.

The effects of indirect connectivity deserve special attention. The indirect connectivity of the acquirer team in the dyad does not affect the likelihood that the interface would be attended, regardless of the strength of the interfaces. However, the more the provider team in the dyad designs a component with strong indirect connections to other components, the lower the likelihood that the interface would be attended ($\theta_{strong2,\textbf{indirect\_connect}\_provider} < 0$), which is in line with *H1b* ($p < 0.16$ in Model 2.2 and $p < .05$ in Model 2.4). The effect is not significant for weak interfaces. This, however, does not indicate that strong interfaces are less likely to be attended than weak interfaces. Rather, and more importantly, it indicates that strong interfaces are more sensitive than weak interfaces to the unpredictable complications that

might result from indirect component connectivity. Moreover, the fact that the observed negative effect is due to indirect rather than to direct component connectivity provides evidence in support of the excessive workload mechanism associated with component connectivity. Indeed, indirect connectivity is more likely to cause unpredictable workload. This, in turn, should strain the team resources, rendering it less capable to provide technical information when requested.

In line with *H2a,* the negative and significant coefficient $\theta_{weak2,\textbf{team\_density}\_acquirer}$ (Models 2.3 and 2.4) suggests that teams with less dense communication network are better at attending weak in-coming design interfaces, but these teams are not better at acquiring information on strong in-coming interfaces. This suggests that weak design interfaces, which have a higher risk of being unattended by team interactions, would be even less likely to be attended if the acquirer team in the dyad has a dense communication network structure. Supporting *H2b*, the full model shows that the denser the network of the provider team, the more likely an interface will be attended, regardless of its strength ($\theta_{weak2,\ \textbf{team\_density}\_provider} > 0$ and $\theta_{strong2,\ \textbf{team\_density}\_provider} > 0$). This pattern of results is consistent with the cognitive lock-in and the flexible capacity mechanisms we associated with network closure. Indeed, cognitive lock-in is more likely to affect weak interfaces, which is confirmed by our analysis. Flexible capacity, in turn, should equally allow a team to attend requests for information on its out-going interfaces, regardless of their strength, because the provision of technical communication is typically triggered by a request from another team. Density simply makes the team more capable of responding to those requests

5. Discussion and Conclusions

The main question we address in this paper is: *What makes some teams better than others on aligning their cross-team interactions with the design interfaces of the product components they design?* This is particularly relevant in complex product development, where coordinating the effort of many design teams developing the components that form the product pose important organizational challenges.

In the product domain, we found evidence that suggests that the internal complexity of the component limits the capability of its design team to attend the in-coming design interfaces that affect the

component. Yet, we found little support to the prediction that such bias towards internal design issues would be mitigated by the connectivity of the component itself, even for strong interfaces. This last result has important managerial implications. It suggests that the extent to which a component shares interfaces with many other components may not suffice to steer the attention of the team towards those interfaces. Rather than expecting that the attention of teams will naturally focus on the interfaces of their component, product managers should actively help teams—especially those designing internally complex components—to pay attention also to the interfaces of their components. Qualitative evidence from the PW4098 project supports this finding. The design of some complex components like the fan hub or the high-pressure compressor [HPC] fixed stators, whose lead times were driven by engineering analysis and by internal part definition leading to long production cycles (e.g., major castings and custom forgings of super-alloys) were managed the "old way"—that is, with infrequent but sufficient communication rather than using concurrent engineering practices. This approach of course increased the risk of not been up to date on all the developments taking place during the design execution in other engine components.

We also argued that component connectivity would increase the unpredictable workload for the team, which would limit the capacity of the team to act as a provider of information of its out-going design interfaces. We found empirical support for such a prediction: indirect component connectivity is likely to hinder the capability of teams to provide attention to *strong* (out-going) design interfaces ($\theta_{strong2,\mathbf{indirect\_connect}\_provider} < 0$ in Table 2). The excessive workload associated with high component connectivity does not negatively impact the attention to weak design interfaces (perhaps because these are already more likely to be overlooked by design teams), but it does reduce the likelihood of providing technical information on strong interfaces. This has implications for the management and allocation of resources to design teams. Typically, managers allocate and manage team resources based on the design tasks assigned to the team (Adler et al. 1995, Loch and Terwiesch 1999). These, in turn, are typically associated with the design of the component itself and its most relevant in-coming interfaces (Eppinger et al. 1994, Terwiesch et al. 2002). Our results suggest that managers may need to ensure also that teams

have enough capacity and incentives to handle the requests for technical information on critical out-going interfaces they are likely to receive from other teams. Our results show that this is particularly troublesome when teams that are unexpectedly busy due to the effect of indirect design interfaces get "heavy" technical requests from other teams due to strong out-going interfaces. Facing such an unexpected overload, teams seem to give less priority to requests for technical information coming from other teams, which can in turn cause further complications and rework in the design process.

On the organizational domain, our results strongly suggest that teams with a sparse communication network structure are less likely to overlook in-coming interfaces. Although attention to strong in-coming interfaces seems to be unaffected by the structure of the communication network, the cognitive lock-in of a closely-knit network significantly increases the risk of ignoring weak in-coming interfaces. This result is consistent with the organizational literature that suggests that sparse networks rich in "structural holes" (Burt 1992, 2004) provide better access to information, whereas dense networks leads actors to neglect finding out what happens beyond that network (Gargiulo and Benassi 2000). Our study suggests that this neglect affected particularly weak interfaces, whose very nature makes less likely to be noticed by design teams. Such neglect, however, may still be consequential. In the project we studied, overlooking some weak design interfaces resulted in small reductions in performance or durability of the affected components and sub-systems, which could in turn cause significant warranty or service expenses over the life of the product. For example, if a critical component (like a turbine airfoil) fails to reach its life expectancy, it could mean additional unplanned engine removals for maintenance. For an engine like the PW4098, a single unplanned engine removal could add up to a $150,000 incremental cost to the customer.

Communication network closure, however, may be beneficial for teams because the flexible capacity created by such network structure renders the team more capable of providing technical information to other teams on interfaces that can affect those teams' components. This finding is consistent with previous studies that highlight the role of network cohesion in facilitating knowledge transfer (Reagans and McEvily 2003) and collaboration in technical organizations (Allen 1977, Ahuja 2000, Obstfeld

2005). More generally, our results on the organizational side pose a dilemma for managers of complex development projects: On one hand, teams with sparse communication networks are better at acquiring technical information on weak in-coming design interfaces, which are more likely to being overlooked in the design process. On the other hand, teams with a dense communication network are better at providing information on the technical requests they get from other teams. While project managers may have limited capacity to influence the shape of communication networks, our results suggest that they should try to encourage teams whose components have many in-coming weak interfaces to build sparse networks, perhaps using integrator teams as catalysts for tie formation.

As Simon (1981) suggested, complex systems are difficult to understand because the behavior of the whole depends in non-trivial ways on how its elements interact. By studying how organizations align their interactions with the product architecture they design we have learned that *indirect* ties matter significantly: Indirect interfaces beyond adjacent components and indirect interactions among the contact teams of the focal team influence significantly the capability (of the focal team) to attend the interfaces of the component it designs. Moreover, such influences are not trivial because depend on the role as acquirer or provider of information that the focal team plays during the design process. This has important implications for both theory and practice. It also raises a number of interesting questions: What is the role of alternative coordination mechanisms on the effective attention to technical interfaces? How do component connectivity and team network structure relate to important design decisions such as outsourcing? How do product and organizations co-evolve over time? By fruitfully combining insights from operations management and social networks, we believe that we have delineated a path that can help addressing these important questions. Appendices are included in an online supplement.

7. References

Adler, P., A. Mandelbaum, V. Nguyen, E. Schwerer. 1995. From project to process management: An empirically based framework for analyzing product development time. *Mgmt. Sci*. 41(3): 458–484.

Ahuja, G. 2000. Collaboration networks, structural holes, and innovation: A longitudinal study. *Admin. Sci. Quaterly*. 45: 425–455.

Allen, T.J. 1977. *Managing the Flow of Technology*. Cambridge, Mass.: MIT Press.

Baldwin, C.Y., and K.B. Clark. 2000. *Design rules: Volume 1: The power of modularity*. MIT Press.

Burt, R.S. 1992. *Structural Holes, The Social Structure of Competition*. Harvard University Press.

—, 2004. Structural holes and new ideas. *American Journal of Sociology* 110: 349–399.

—, 2005. Brokerage and Closure: An Introduction to Social Capital. Oxford University Press.

Clark, K.B., and T. Fujimoto. 1991. Product Development Performance: Strategy, Organization and Management in the World Auto Industry. Cambridge, Mass.: Harvard Business School Press.

Clarkson, P.J., C.S. Simons, and C.M. Eckert. 2004, Predicting change propagation in complex design, *Journal of Mechanical Design*, **126**(5):765-797.

Coleman, J.S. 1990, *Foundations of social theory*. Cambridge, Mass.: Belknap Press.

Eppinger, S. D., D.E. Whitney, R.P. Smith, D.A. Gebala. 1994. A model-based method for organizing tasks in product development. *Research in Engineering Design* 6(1): 1–13.

Ethiraj, S.K., and D. Levinthal. 2004. Modularity and innovation in complex systems. *Mgmt. Sci*. 50(2).

Galbraith, J. R. 1973. *Designing Complex Organizations*. Reading, Mass.: Addison-Wesley Publishing.

Gargiulo, M. and M. Benassi. 2000. Trapped in your own net? Network cohesion, structural holes, and the adaptation of social capital. *Org. Sci.* 11(2): 183–196.

Gokpinar, B., W. Hopp, and S. Iravani. 2007. The impact of product architecture and organizational structure on efficiency and quality of complex product development. *WP Northwestern University*.

Grabher, G. 1993. The weakness of strong ties. The lock-in of regional developmente in the Ruhr area. in Gernot Grabher, (Ed.): *The Embedded Firm*. London and New York: Routledge.

Granovetter, M.S. 1985. Economic action and social structure: The problem of embeddedness. *American Journal of Sociology* 91: 481–510.

Gumbel, P. 2006. Trying to untangle wires. *Time*. October 16 (European edition), p. 36-37.

Henderson, R., and K. Clark. 1990. Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms. *Admin. Sci. Quart*. 35(1): 9–30.

Loch, C.H., and C. Terwiesch. 1999. Accelerating the process of engineering change orders: Capacity and congestion effects. *J. Prod. Innov. Mgmt*. 16: 145-159.

—, —, and S. Thomke. 2001. Parallel and sequential testing of design alternatives. *Mgmt. Sci*. 45(5).

Lomi, A., and P. Pattison. 2006. Manufacturing relations: An empirical study of the organization of production across multiple networks. *Org. Sci*. **17**(3) 313-332.

Marsden, P.V. 1990. Network data and measurement. *Annual Review of Sociology*. **16** 435-463.

—, and K.E. Campbell. 1984. Measuring tie strength. *Social Forces*. **63** 482-501.

Mihm, J., C. Loch, A. Huchzermeier. 2003. Problem-solving oscillations in complex engineering projects. *Manag. Sci*. 46(6): 733-750.

Morelli, M.D., S.D. Eppinger, R.K. Gulati. 1995. Predicting technical communication in product development organizations. *IEEE Trans. Eng'g. Management* 42(3): 215-222.

Obstfeld, D. 2005. Social networks, the *Tertius iungens* orientation, and involvement in innovation. *Admin. Sci. Quart.* 50: 100–130.

Ocasio, W.1997. Towards an attention-based view of the firm. *Strat. Mgmt. J*.18:187-206.

Pattison, P. E., Robins, G.L. 2007. Probabilistic network theory. In Rudas, T (Ed.), *Handbook of Probability Theory with Applications*. Sage Publications (forthcoming).

—, and Wasserman, S. 1999. Logit models and logistic regressions for social networks: II. Multivariate relations. *British J. of Math. and Stat. Psychology.* **52** 169-193.

Pinedo, M., S. Sehasdri, and E. Zemel. 2000. The Ford-Fireston case. *Teaching case*. Department of Information, Operations, and Managment Sciences, Stern School of Business, NYU.

Reagans, R., E. Zuckerman, and B. McEvily 2004 How to make the team: Social networks: Social networks vs. demography as criteria for designing effective teams. Administrative Science Quarterly, 49: 101-133.

—, and B. McEvily. 2003. Network structure and knowledge transfer: The effects of cohesion and range. *Admin. Sci. Quart.* 48: 240–267.

Robins, G.L., Elliott, P., & Pattison, P. 2001. Network models for social selection processes. *Social Networks,* **23**, 1-30.

Robins, G., and Pattison, P. 2005. Interdependencies and social processes: Dependence graphs and generalized dependence structures. P.J. Carrington, J. Scott, S. Wasserman, eds. *Model and Methods in Social Network Analysis*. Cambridge University Press, New York, 192-214.

—, Pattison, P., and Wasserman, S. 1999. Logit models and logistic regressions for social networks, III. Valued relations. *Psychometrika.* **64** 371-394.

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

Simon, H.A. 1947. Administrative Behavior: A Study of Decision-making Processes in Administrative Organizations. Macmillan, Chicago, IL.

—, 1981. *The Science of the Artificial* (2nd ed.). Cambridge, Mass.: MIT Press.

Smith, R.P., and S.D. Eppinger. 1997. Identifying controlling features of engineering design iteration. *Management Science* 43(3): 276–293.

Sosa, M.E., S.D. Eppinger, and C.M. Rowles. 2003. Identifying modular and integrative systems and their impact on design team interactions. *Journal of Mechanical Design* 125(2): 240–252.

—. 2004. The misalignment of product and organizational structures in complex product development. *Mgmt. Sci.,* 50(12): 1674–1689.

—. 2007. A network approach to define modularity of components in complex products. *ASME Journal of Mechanical Design (forthcoming).*

Snijders, T., P. Pattison, G. Robins, and M. Handcock. 2006. New specifications for exponential random graph models. *Sociology Methodology*.

Terwiesch, C., C. H. Loch, A. De Meyer. 2002. Exchanging preliminary information in concurrent engineering: Alternative coordination strategies. *Org. Sci*. 13(4): 402–419.

Thompson, J.D. 1967. *Organizations in Action*. New York: McGraw-Hill.

Tushman, M.1977. Special boundary roles in the innovation process. *Admin. Sci. Quart*. 22: 587-605.

Ulrich, K.T. 1995. The role of product architecture in the manufacturing firm. *Res. Policy* 24:419–440.

—, S.D. Eppinger. 2004. *Product Design and Development* (3rd ed.). New York: McGraw-Hill.

Wasserman, S. and P. Pattison. 1996. Logit models and logistic regression for social networks: I. an introduction to Markov graphs and p*. *Psychometrika* **61** 401-425.

Table 1. PLEs of *p models (dichotomous relations)**

	Parameters	Model 1.1	Model 1.2	Model 1.3	Model 1.4
Base association effect	θ_{12}	2.994** (1.328)	2.608 (1.698)	4.194 (2.748)	2.262 (3.024)
Interaction effects with control variables	$\theta_{12,redesign\_acquirer}$	.476 (.491)	.689 (.527)	.206 (.523)	.209 (.564)
	$\theta_{12,redesign\_provider}$	-.042 (.492)	-.085 (.530)	.189 (.514)	.211 (.556)
	$\theta_{12,complexity\_acquirer}$	-.435*** (.153)	-.648*** (.169)	-.536*** (.168)	-.753*** (.183)
	$\theta_{12,complexity\_provider}$	.203 (.143)	.334** (.166)	.257 (.156)	.355** (.173)
	$\theta_{12,teamsize\_acquirer}$	.087 (.208)	-.026 (.230)	-.152 (.241)	-.210 (.257)
	$\theta_{12,teamsize\_provider}$	.084 (.195)	.060 (.224)	.330 (.227)	.378 (.246)
	$\theta_{12,integrators\_acquirer}$	-.009 (.019)	.000 (.022)	.003 (.023)	.005 (.025)
	$\theta_{12,integrators\_provider}$	.011 (.017)	-.002 (.020)	-.002 (.021)	-.009 (.022)
Interaction effects with component connectivity variables	$\theta_{12,direct\_connect\_acquirer}$		.068* (.041)		.065 (.047)
	$\theta_{12,direct\_connect\_provider}$		-.036 (.037)		-.016 (.042)
	$\theta_{12,indirect\_connect\_acquirer}$		.012 (.012)		.019 (.013)
	$\theta_{12,indirect\_connect\_provider}$		-.014 (.013)		-.016 (.013)
Interaction effects with team communication network structure variables	$\theta_{12,team\_contacts\_acquirer}$			-.001 (.043)	-.015 (.048)
	$\theta_{12,team\_contacts\_provider}$			-.001 (.043)	.032 (.046)
	$\theta_{12,team\_density\_acquirer}$			-.041** (.019)	-.045** (.020)
	$\theta_{12,team\_density\_provider}$			.026 (.018)	.047** (.019)
	# of parameters	53	65	65	77
	G^2_{PL}	1899.049	1756.061	1828.875	1679.159

Table 2. PLEs of *p models (trichotomous design interface relation)**

	Parameters	Model 2.1	Model 2.2	Model 2.3	Model 2.4
Base association effects	θ_{weak2}	.667 (1.481)	.466 (1.827)	1.976 (3.203)	.376 (3.429)
	$\theta_{strong2}$	5.261*** (1.891)	3.779 (2.453)	5.113 (3.954)	2.333 (4.389)
Interaction effects with component connectivity variables	$\theta_{weak2,direct\_connect\_acquirer}$		.052 (.043)		.057 (.050)
	$\theta_{strong2,\ direct\_connect\_acquirer}$		.121** (.056)		.006 (.067)
	$\theta_{weak2,\ direct\_connect\_provider}$		-.062 (.039)		-.049 (.045)
	$\theta_{strong2,\ direct\_connect\_provider}$		-.033 (.050)		-.001 (.056)
	$\theta_{weak2,\ indirect\_connect\_acquirer}$		-.001 (.014)		.010 (.015)
	$\theta_{strong2,\ indirect\_connect\_acquirer}$		.026 (.019)		.032 (.021)
	$\theta_{weak2,\ indirect\_connect\_provider}$		-.011 (.015)		-.010 (.015)
	$\theta_{strong2,\ indirect\_connect\_provider}$		-.026 (.018)		-.038** (.019)
Interaction effects with component connectivity variables	$\theta_{weak2,\ team\_contacts\_acquirer}$			-.024 (.048)	-.042 (.054)
	$\theta_{strong2,\ team\_contacts\_acquirer}$			.104 (.064)	.164** (.077)
	$\theta_{weak2,\ team\_contacts\_provider}$			.025 (.048)	.053 (.052)
	$\theta_{strong2,\ team\_contacts\_provider}$			-.034 (.058)	-.006 (.064)
	$\theta_{weak2,\ team\_density\_acquirer}$			-.054** (.022)	-.059** (.023)
	$\theta_{strong2,\ team\_density\_acquirer}$			-.024 (.026)	-.024 (.028)
	$\theta_{weak2,\ team\_density\_provider}$			.035* (.020)	.046** (.021)
	$\theta_{strong2,\ team\_density\_provider}$			.028 (.027)	.057* (.030)
	# of parameters	94	114	114	134
	G^2_{PL}	2186.590	1985.589	2098.263	1887.149

Approximate standard errors between parentheses

* < 0.10 ** < 0.05 *** < 0.01 (Two-tailed *approximate* statistical inference)

Europe Campus
Boulevard de Constance,
77305 Fontainebleau Cedex, France
Tel: +33 (0)1 6072 40 00
Fax: +33 (0)1 60 74 00/01

Asia Campus
1 Ayer Rajah Avenue, Singapore 138676
Tel: +65 67 99 53 88
Fax: +65 67 99 53 99

www.insead.edu

INSEAD

The Business School
for the World