MANAGEMENT SCIENCE
Vol. 50, No. 12, December 2004, pp. 1674–1689
ISSN 0025-1909 | EISSN 1526-5501 | 04 | 5012 | 1674
informs®
DOI 10.1287/mnsc.1040.0289

The Misalignment of Product Architecture and Organizational Structure in Complex Product Development

Manuel E. Sosa
INSEAD, Boulevard de Constance, 77305 Fontainebleau, Cedex, France, manuel.sosa@insead.edu
Steven D. Eppinger
Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, eppinger@mit.edu
Craig M. Rowles
Advance Engine Program, Pratt & Whitney Aircraft, East Hartford, Connecticut 06108, rowles@alum.mit.edu

Product architecture knowledge is typically embedded in the communication patterns of established development organizations. While this enables the development of products using the existing architecture, it hinders the organization's ability to implement novel architectures, especially for complex products. Structured methods addressing this issue are lacking, as previous research has studied complex product development from two separate perspectives: product architecture and organizational structure. Our research integrates these viewpoints with a structured approach to study how design interfaces in the product architecture map onto communication patterns within the development organization. We investigate how organizational and system boundaries, design interface strength, indirect interactions, and system modularity impact the alignment of design interfaces and team interactions. We hypothesize and test how these factors explain the existence of the following cases: (1) known design interfaces not addressed by team interactions, and (2) observed team interactions not predicted by design interfaces. Our results offer important insights to managers dealing with interdependences across organizational and functional boundaries. In particular, we show how boundary effects moderate the impact of design interface strength and indirect team interactions, and are contingent on system modularity. The research uses data collected from a large commercial aircraft engine development process.

Key words: product architecture; product development organizations; technical communication; design structure matrix; statistical network analysis
History: Accepted by Karl Ulrich, technological innovation, product development, and entrepreneurship; received July 23, 2002. This paper was with the authors $13\frac{1}{2}$ months for 3 revisions.

1. Introduction

Understanding how organizations manage the knowledge associated with the architecture of the products they design is a critical challenge for firms developing complex products. As highlighted by Henderson and Clark (1990, p. 9), "architectural knowledge tends to become embedded in the structure and information-processing procedures of established organizations." Hence, organizations dealing with novel architectures must understand how they manage the embedded knowledge of the products they currently develop. This is especially relevant in complex product development due to the large number of both physical components and design participants involved in the process. Unfortunately, methods and/or tools available to address this challenge are scarce.

Consider the typical job of a design engineer during the development of a complex product, such as an aircraft engine. Usually, design engineers are part of cross-functional design teams dedicated to specific components of the product (Robertson and Allen 1992, Pimmler and Eppinger 1994). During the design phase, the team responsible for designing an engine component (e.g., the blades of the low-pressure turbine) needs to balance the technical demands from other design teams in addition to managing its own design constraints. Usually, demands from other teams depend on the nature of the design interfaces between their components. For example, when examining the interfaces between the vanes and blades of the low-pressure turbine studied in this paper, we learned that there is a potential transfer of energy (vibration) from the vanes to the blades that needs to be avoided. Teams designing those components are expected to interact to address such an interface (see Sosa et al. 2003 for further details). In general, while managing the integration effort across design teams, managers of complex development projects typically raise the following questions: Are design teams communicating about the right things? If not, why? Are all design interfaces between product components identified and addressed during the design phase? If not, why?

This situation highlights the importance of not only identifying the interfaces between product components but also evaluating whether or not the corresponding teams interact to address those interfaces properly. Of course, it is not difficult to argue that if two components share design interfaces, the teams that design them need to interact (Thompson 1967, Galbraith 1973). However, in the development of highly complex products, it would be naïve to expect a perfect mapping between design interfaces and team interactions. Hence, we investigate the factors that prevent such an occurrence:

- Can we expect any significant misalignment of product architecture and organizational integration effort? If so, how can we uncover it?
- What factors may impact such misalignment? More specifically,
  - — Why do some design interfaces between product components not correspond to technical interactions between the design teams that develop them?
  - — Why do some technical interactions between design teams take place even though no design interface is identified between the components they design?

Investigating these questions is crucial to understand in what areas of the product and organization managers need to pay particular attention to moderate the impact of misalignments. Previous research has studied complex development efforts separately from two important perspectives: the product architecture and the organizational structure. Rather, we bring these two perspectives together to examine how and why interfaces between product components map onto interactions between teams designing them.

This paper offers two important contributions. First, we integrate two separate streams of research to investigate why misalignment of product and organizational structures occur, and hypothesize factors that impact such misalignment. This contributes to the existing literature by enhancing our understanding of technical communication patterns in organizations developing complex products. Second, we extend our research approach introduced in Sosa et al. (2003) by using a novel statistical network analysis technique (based on p^* models of Wasserman and Pattison 1996) to test hypothesized effects while controlling for dyadic and triadic tendencies typically embedded in network data. By doing so, we not only uncover misalignment of product and organizational structures, but also properly examine factors that are systematically associated with such misalignment. We provide some evidence that while certain types of misalignment can be beneficial to complex product development projects, others can be extremely costly, resulting in major rework and customer impact. Hence, it is important for managers to anticipate where misalignment is more likely to occur—that is, to distinguish which areas of the product and organization require special attention to identify critical design interfaces and ensure important team interactions.

2. Design Interfaces and Team Interactions

In the product architecture domain, we define a *design interface* between component i and component j as component i depending on component j for functionality. That is, component j either imposes geometry constraints or transfers forces, material, energy, and/or signals to component i for component i to function properly. In the organizational domain, we define *team interaction* between design team i and design team j as team i requesting technical information directly from team j during the detailed design phase of the development process. Note that our definitions for both design interface and team interaction imply a direction. That is, component i's functionality is affected by component j, and technical information flows from team j to team i.

We observe that during the detailed design phase of a complex development effort, design interfaces are the primary source of team interdependence (Mihm et al. 2003). Hence, for projects where the task structure is of the form "*team i* designs *component i*" (which is typical in complex product development), it is not difficult to argue that the existence (or absence) of a design interface between component i and component j should correspond to the existence (or absence) of technical interaction of team i with team j. These expected cases are represented by the lower-left and upper-right cells of Figure 1. The lower-right and upper-left cells represent the unexpected cases, which are the focus of our study. *Unmatched design interfaces* correspond to design interfaces that are not addressed by direct team interactions, whereas *unmatched team interactions* correspond to communication between teams whose components

Figure 1 **Mapping Design Interfaces and Team Interactions**

Team Interactions		Design Interfaces: YES	Design Interfaces: NO
	NO	***Unmatched*** *design interfaces*	***Aligned*** *absence* of interfaces and interactions
	YES	***Aligned*** *presence* of interfaces and interactions	***Unmatched*** *team interactions*

do not explicitly share design interfaces. Note that our variable of interest is whether or not design interfaces and team interactions are aligned. Hence, we are not claiming causality, but association. Although we believe design interfaces drive team interactions for the most part, we are also open to the possibility of team interactions determining some design interfaces.

Previous research has largely ignored the significant existence of unmatched design interfaces and unmatched team interactions (Brown and Eisenhardt 1995, Krishnan and Ulrich 2001). However, we believe that considering these cases is important because their existence would indicate that not all known product-related interdependences are addressed by direct technical communication, and that technical communication (where not expected) may uncover undocumented product-related interdependences.

There are two types of factors that may prevent alignment of design interfaces and team interactions. First, dynamic factors refer to how previous and future development efforts may affect the likelihood of encountering misaligned cases (Henderson and Clark 1990, Adler 1995, Terwiesch et al. 2002). Although dynamic factors are important (see online Appendix D available at mansci.pubs.informs.org/ecompanion.html), we focus this study on understanding static effects. Static factors refer to how both the current product architecture and organizational structure impact the likelihood of misalignment of design interfaces and team interactions.

2.1. Product Architecture Perspective

Previous research on product architecture has focused on the product itself. In this view, product architecture is defined as "the specification of the *interfaces* among interacting physical components" (Ulrich 1995, p. 420). Although previous work in this area has advanced our understanding of architectural knowledge and its impact on some operational aspects of the firm, the explicit link between product architecture and organizational structure has been largely neglected (see reviews by Krishnan and Ulrich 2001, Sosa et al. 2003). In this paper, we extend the product architecture literature by proposing that although most of the architectural knowledge is explicit and known by development organizations, some interfaces between components are unspecified (or even unknown) and only identified or documented during the design process itself. It then becomes important to determine where (in the product) those unidentified interfaces are likely to be, and how they can be uncovered. By simultaneously analyzing the design network of components and the communication network of design teams, we uncover those unknown interfaces and the factors that influence their occurrence, which provides us with a more complete view of the architecture of the product.

2.2. Organizational Perspective

Adopting the information-processing viewpoint, Brown and Eisenhardt (1995, p. 358) summarize that "frequent and appropriately structured task communication" results in better performing development processes. Not surprisingly, a large body of research focuses on the communication process in development organizations, much of which has studied how factors such as physical distance, organizational structures, task structures, and communication media affect technical communication (e.g., Allen 1977, Morelli et al. 1995, Van den Bulte and Moenaert 1998, Sosa et al. 2002). Yet, how product architecture relates to technical communication remains unaddressed by this stream of work.

Much work on technical communication has focused on understanding the factors that inhibit technical communication. In addition to distance, boundaries between distinct organizational groups have also been identified as an important inhibitor to communication (e.g., Allen 1977, Van den Bulte and Moenaert 1998). Conversely, there is another line of thought in the product innovation literature that focuses on team interdependence as an important driver of technical communication (e.g., Morelli et al. 1995, Adler 1995, Loch and Terwiesch 1998, Terwiesch et al. 2002). In this paper, we not only show how product architecture is an essential source of team interdependence, but we also disentangle the hindering effects of organizational boundaries from the motivating effects of product interdependence. This paper also extends the literature on product innovation by explicitly considering indirect interactions between design teams as a possible mechanism to handle certain design interfaces.

3. Understanding the Misalignment of Design Interfaces and Team Interactions

3.1. Effects of System and Organizational Boundaries

Architectures of complex products are typically decomposed into systems and components. As a result, *system boundaries* are established to cluster components so that a significantly larger proportion of design interfaces are within these boundaries (Pimmler and Eppinger 1994, Stone et al. 2000, Whitney 2004). This may impose architectural knowledge barriers that inhibit explicit identification of design interfaces across systems by the design experts (Henderson and Clark 1990, Sanchez and Mahoney 1996). Nevertheless, to develop working systems, we propose that certain design teams need to interact, which results in unmatched team interactions. This argument is consistent with the concept of ambiguity associated with complex engineering projects (Schrader et al. 1993, Pich et al. 2002). That is, due to *product ambiguity*, defined as the absence of knowledge about design variables and/or their interfaces, some design interfaces are not foreseen at the outset of the project and are only discovered after design teams work on the systems themselves. We argue that evidence of product ambiguity is more likely to occur across system boundaries because system barriers prevent the identification of some existing interfaces. Hence, for complex products, we expect a higher likelihood of encountering unmatched team interactions across system boundaries.

Complex product development also requires structuring the organization into groups of cross-functional design teams to design systems and components. As a result of this organizational breakdown, *organizational boundaries* are formed between design teams that belong to different groups of teams (Ulrich and Eppinger 2004, p. 21). Previous research on R&D management suggests that organizational boundaries between functional groups impose communication barriers that inhibit cross-team interactions even in the presence of collocation (e.g. Allen 1977, Van den Bulte and Moenaert 1998). People within these groups are subjected to organizational bonds that promote the development of a language and an identity inherent to the group in which they belong. As a result, the greater the degree of group specialization, the higher the communication barriers are across them (Tushman and Katz 1980). Accordingly, in complex product development projects, organizational boundaries are expected to significantly reduce cross-boundary team interactions. By extension to our framework, we should expect a significantly larger proportion of unmatched design interfaces across boundaries. More specifically, we expect design teams to exhibit a lower tendency to discuss cross-boundary design interfaces than within-boundary interfaces. Hence, we envision a higher likelihood of encountering unmatched design interfaces across organizational boundaries. Considering the effects of system and organizational boundaries, we posit the following hypothesis to test:

HYPOTHESIS 1. *Misalignment of design interfaces and team interactions is more likely to take place across system and organizational boundaries than within boundaries.*

3.2. Effects of Design Interface Strength

When examining design interfaces between components of complex products, research in engineering design has distinguished various types of design dependencies (such as spatial, material, and energy types) and several levels of criticality (such as required, indifferent, and detrimental) to characterize a design interface between any two components (Pahl and Beitz 1991, Pimmler and Eppinger 1994, Sosa et al. 2003). We extend this taxonomy by defining two levels of strength of a design interface. We define *weak* design interfaces as those which involve few types of design dependencies and have low impact upon the functionality of the other component, whereas *strong* design interfaces are those that involve several types of design dependencies and have high impact on the functionality of the other component.

To understand the link between design interface strength and team interactions, we refer to previous research on task interdependence. The degree of task interdependence determines the degree to which tasks require collective action (Thompson 1967). Moreover, the greater the degree of task interdependence, the greater the information requirements are between design teams (Galbraith 1973). This is consistent with research that has shown that a greater degree of task interdependence leads to greater team interaction (e.g., Adler 1995, Smith and Eppinger 1997, Loch and Terwiesch 1998). Considering that in many complex development efforts a significant proportion of the task structure directly maps onto the product structure under development (i.e., *task i* is defined as "designing component *i*"), we expect strong design interfaces to generate greater team interdependence. This should result in higher likelihood of team interaction between the corresponding design teams:

HYPOTHESIS 2. *In complex products, strong design interfaces are more likely to be aligned with team interactions than are weak design interfaces.*

Hypothesis 1 posits that organizational boundaries hinder the alignment of design interfaces and team interactions, whereas Hypothesis 2 suggests that greater component interdependence favors the occurrence of such alignment. We then ask: Which effect is stronger?

When considering the effects of system boundaries, one might claim that they not only inhibit legacy design experts from identifying all cross-boundary interfaces (Hypothesis 1), but they also inhibit design teams from properly perceiving strong design interfaces as such. This is particularly relevant in complex products due to the simultaneous presence of several types of design dependencies (such as spatial, structural, and thermal) associated with the same design interface (Pahl and Beitz 1991, Pimmler and Eppinger 1994). In addition, organizational research has suggested that design teams not only face difficult challenges when they need to search for and transfer technical knowledge across their organizational boundaries (e.g., Hansen 1999, 2002), but also tend to simplify and filter certain aspects of external information to facilitate internal problem solving

(Henderson and Clark 1990). Based on this, one could argue that, across boundaries, design teams would be more likely to underestimate the impact of certain types of design dependencies, and therefore would not be able to distinguish the difference between weak and strong design interfaces.

Previous research in development organizations, which recognizes that teams are selective when interacting across boundaries, provides the basis of the alternative reasoning for which we argue. This stream of work suggests that teams engage in cross-boundary communication to address critical interdependence (e.g., Tushman and Katz 1980, Ancona and Caldwell 1992, Cummings 2004). Moreover, Tushman (1977, p. 592) suggests that specialized gatekeepers "may not attend to all external communication areas, but may specialize in those external areas most critical to the work of their unit." This observation is consistent with recent findings from the telecommunications industry suggesting that teams are more likely to cross communication barriers imposed by geographical separation when they are highly interdependent (Sosa et al. 2002, Cummings 2004). Extending this insight to our context, we argue that teams are more likely to overcome system/organizational boundaries to address strong design interfaces.

Hypothesis 3. *Strong design interfaces are more likely to be aligned with team interactions than are weak design interfaces, even across organizational and system boundaries.*

3.3. Effects of Indirect Team Interactions

We define *indirect team interactions* as technical information flow that takes place between two teams through an intermediary design team. Research in social networks has long supported the notion of indirect communication via intermediary units (Granovetter 1973). More recently, research about knowledge sharing in a multi-unit development organization has also considered the role of indirect relations to effectively transfer technical information through intermediary teams that are close to the focal team (Hansen 1999, 2002). Although early work in R&D identified the organizational benefits of having a gatekeeper who could gather relevant information from the team's external environment and pass it to the rest of the team (e.g., Tushman 1977, Tushman and Katz 1980), indirect interactions between design teams have been largely neglected as a coordination mechanism to address their interdependence in product development organizations (e.g., Adler 1995, Terwiesch et al. 2002).

In this paper, we use the concept of indirect team interaction to hypothesize that team i, whose component has a design interface with component j, may not report direct interaction with team j because it interacts with an intermediary team (team k, which also interacts with team j) which passes the information (to team i) that would otherwise have flowed directly from team j to team i. Hence, we expect a higher likelihood of finding unmatched design interfaces between teams that communicate indirectly through intermediary teams:

Hypothesis 4. *Two interrelated components are more likely to have an unmatched design interface when their corresponding design teams have other intermediary teams through which they can indirectly communicate.*

3.4. Effects of Indirect Design Interfaces

In the product architecture domain, we introduce the notion of *indirect design interfaces* as the indirect impact of component j over component i through an intermediary component k. This definition considers the product as a set of interrelated elements (Krishnan and Ulrich 2001). Although the impact in the design process due to the propagation of product design dependencies through intermediary components has been investigated (e.g., Whitney 2004, Mihm et al. 2003), the effects on the communication patterns due to unconnected components linked through intermediary components remains unknown.

Similar to the case of indirect team interactions, we hypothesize that the existence of intermediary elements between two components that do not share a direct design interface increases the likelihood that the corresponding design teams interact, resulting in an unmatched team interaction. Considering the effects of indirect design interfaces is important because it offers an alternative explanation to our initial argument that unmatched team interactions indicate the existence of unidentified direct design interfaces between two components. That is, design teams might interact not only to address direct interfaces that had not been identified at the outset of the project (product ambiguity effect), but also to address indirect design interfaces between components not directly connected (product complexity effect):

Hypothesis 5. *Two design teams are more likely to have an unmatched team interaction when the components they design share interfaces with a common component.*

3.5. Effects of System Modularity

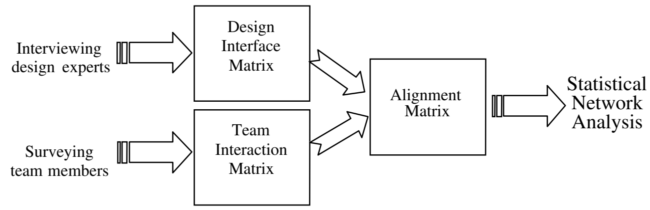
Based on how functions map onto physical components, one can distinguish modular and integral product architectures (Ulrich 1995). In Sosa et al. (2003), we extend this concept to the system level by introducing a new notion of system modularity based upon the way physical components share design interfaces across systems within a complex product. *Modular systems* are "those whose design interfaces with other systems are clustered among a few physically adjacent systems," whereas *integrative systems* are "those whose design interfaces span all, or most of, the systems that comprise the product due to their physically distributed or functionally integrative nature throughout the product" (Sosa et al. 2003, p. 240).

To study complex product architectures in terms of component interfaces, we use the design structure matrix (DSM) tool, an analytical method introduced by Steward (1981) and used by Eppinger et al. (1994) to study interdependence between product development activities. In Sosa et al. (2003), we detail our DSM approach to identify modular and integrative systems in complex products. Although that paper does not explain why misalignment of design interfaces and team interactions occur (which is the purpose of this paper), it presents limited empirical evidence showing that the effects of organizational and system boundaries described above (Hypothesis 1) are more severe between modular systems than with integrative systems. In addition, we extend the approach presented by Sosa et al. (2003) by applying, for the first time, statistical modeling techniques based on social network analysis for proper hypothesis testing using DSM data.

The organizational literature on product innovation considers products as hierarchically arranged sets of subsystems with defined interfaces (e.g., Alexander 1964). By examining the impact of the architecture of the product on the innovation process from a strategic viewpoint, this line of research suggests that the communication structure of development organizations depends on the type of product architecture they design (e.g., Henderson and Clark 1990, Sanchez and Mahoney 1996, Baldwin and Clark 2000). Previous organizational research suggests that development teams exhibit different strategies to manage their interdependences across boundaries (Ancona and Caldwell 1992). Because modular systems differ from integrative systems in the way they share design interfaces *across* boundaries rather than *within* boundaries (Sosa et al. 2003), we expect modular teams to exhibit different cross-boundary communication patterns than do integrative design teams. That is, given the physically distributed or functionally integrative nature of integrative systems (Pimmler and Eppinger 1994), integrative design teams are more likely to cross organizational boundaries than design teams that develop modular systems (Yassine et al. 2003). Because Sosa et al. (2003) tested this proposition without controlling for typical nonrandom tendencies embedded in DSM data, we posit the following hypothesis for further testing:

Hypothesis 6. *For interfaces and interactions occurring across organizational and system boundaries, misalignment of design interfaces and team interactions is more likely to occur between components that belong to different modular systems than with components that belong to integrative systems.*

Figure 2 **Our Research Approach**



4. Research Approach

This section summarizes our method of comparing and analyzing the architecture of a product with its development organization. Our approach involves four major steps (see Fig. 2):

Step 1: *Identify design interfaces*. By interviewing design experts who have a deep understanding of the architecture of the product, we identify how the product is decomposed into systems, and these are further decomposed into components. We then ask the experts to identify the types and criticality of the design dependencies between all the components. We represent this network of component dependencies in a *design interface matrix*.

Step 2: *Identify team interactions*. We identify the teams responsible for developing each of the product's components. We then survey key members of each team to capture the intensity of the technical interactions between them. We represent this communication network in a *team interaction matrix*.

Step 3: *Map design interfaces and team interactions*. We compare the design interface matrix with the team interaction matrix and capture this comparison in the *alignment matrix*. When each design team is responsible for the design of only one physical component, the alignment matrix is obtained by overlaying the identically sequenced design interface matrix and team interaction matrix.

Step 4: *Analyze the alignment matrix*. We use statistical network analysis techniques to rigorously analyze the patterns exhibited in the alignment matrix and test hypothesized effects that may systematically cause a significant misalignment of design interfaces and team interactions.

5. The Study

We applied our approach to study the detail design period of the development of a large commercial aircraft engine (Sosa et al. 2003), the PW4098 derivative engine developed by Pratt & Whitney (P&W).

5.1. Identifying Design Interfaces

The engine studied was decomposed into eight systems. Each of these systems was further decomposed into five to ten components each, for a total of 54 components. Six of the eight systems were identified as *modular systems*, whereas the other two systems (mechanical components system and externals and controls system) were recognized as *integrative systems* because of the physically distributed and functionally integrative features of their components (Sosa et al. 2003).

Five types of design dependencies were defined for the design interfaces between the physical components, and a five-point scale was used to capture the level of criticality of each dependency for the overall functionality of the component in question. These metrics are discussed at length in Sosa et al. (2003). The type and criticality of design interfaces were used to assess their strength as follows:

$$[\text{design interface strength}]_{ij} = \sum |c_{ij}^d|, \quad \text{where}$$

$d =$ dependency type $=$ [spatial, structural, material, energy, information],

$c_{ij}^d =$ level of criticality for design interface (i, j) of type $d = [-2, -1, 0, +1, +2]$.

For the 569 nonzero design interfaces documented, the mean (sd) of *design interface strength* was 4.41 (1.92). Similar to network studies that consider valued ties (Granovetter 1973, Marsden 1990), we define an indicator variable, STRENGTH_{ij}, which trichotomizes the criticality of the design interfaces:

$$\begin{aligned}
&\text{STRENGTH}_{ij} = \text{NULL} = 0\\
&\qquad \text{if } [\text{design interface strength}]_{ij} = 0,\\
&\text{STRENGTH}_{ij} = \text{WEAK} = 1\\
&\qquad \text{if } 0 < [\text{design interface strength}]_{ij} \le 4,\\
&\text{STRENGTH}_{ij} = \text{STRONG} = 2\\
&\qquad \text{if } [\text{design interface strength}]_{ij} > 4.
\end{aligned}$$

Under this categorization, we determined 319 WEAK interfaces and 250 STRONG interfaces. This is consistent with other observations of complex products in which there are fewer critical interfaces than less important ones (Smith and Eppinger 1997). Alternative definitions of STRENGTH_{ij} resulted in a skewed distribution of nonzero design interfaces, and were somewhat limited in capturing both type and criticality of the design interfaces. Results of categorical data analysis with these alternative definitions were consistent with the findings reported in this paper. We mapped the design interface data into a trichotomous design interface matrix (see Figure 3). The off-diagonal entries of the matrix are marked with either a "W" or "S" to indicate the existence of a WEAK or STRONG design interface, respectively, between two components (see Sosa et al. 2003 for details).

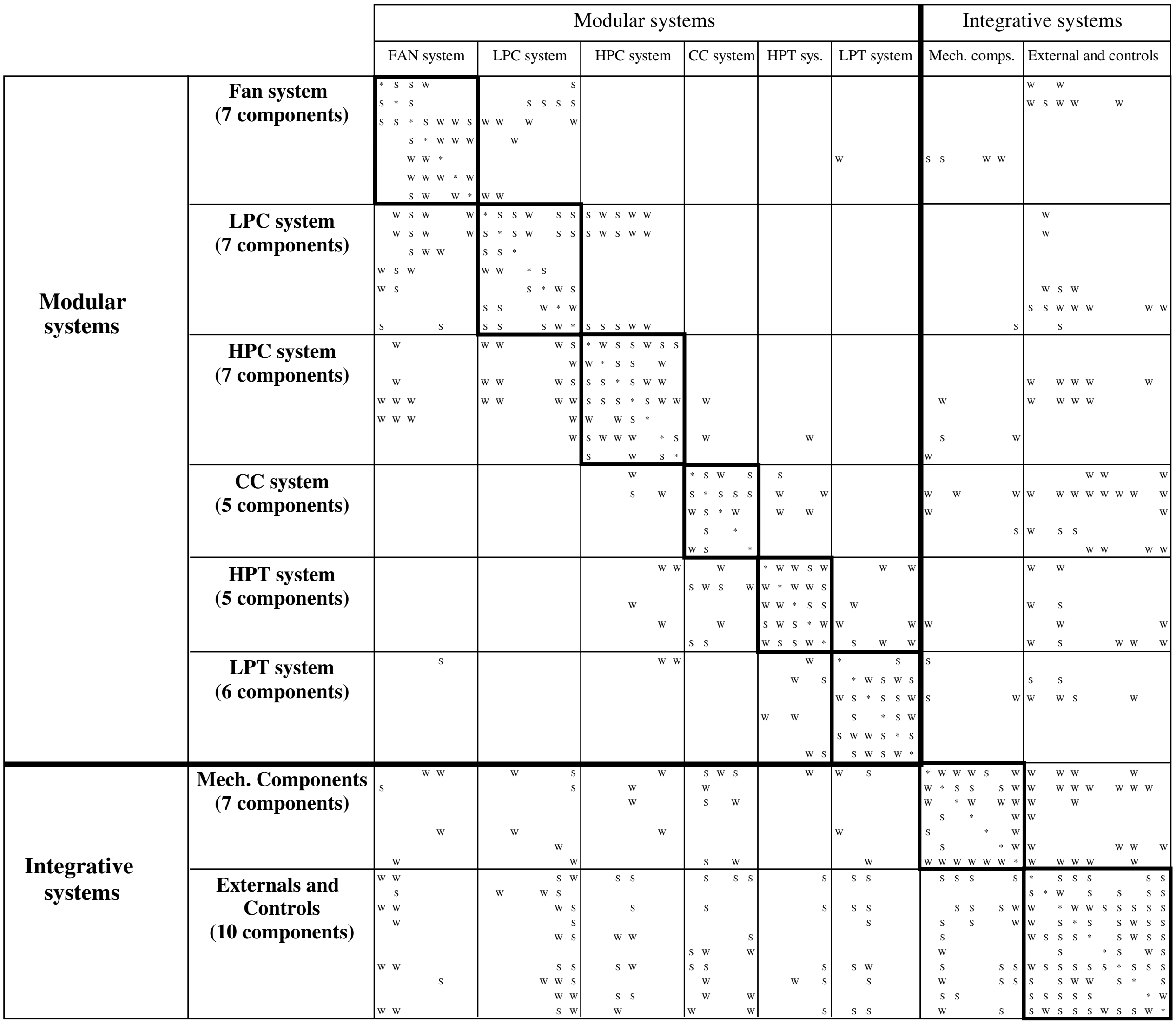
5.2. Identifying Team Interactions

The organization responsible for the development of the aircraft engine was structured into 60 design teams exclusively dedicated to the project. Fifty-four of these teams were responsible for developing the 54 components of the engine, and were grouped into eight system-design groups mirroring the architecture of the engine studied. The remaining six design teams were system integration teams, who had no specific hardware assigned to them, and whose responsibility was to assure that the engine worked as a whole. Examples are the rotordynamics and secondary flow teams.

We captured the intensity of the task-related technical interactions between the design teams involved in the development process (Allen 1977, Morelli et al. 1995). Similar to previous studies in technical communication and social networks, we surveyed key members of design teams about the peak frequency and criticality of their technical interactions (Allen 1977, Marsden 1990). We surveyed 57 of the 60 teams. We assumed reciprocal interactions for the teams whose responses were missing. Additional analysis without these components/teams was consistent with the findings reported here. We used a six-point scale that combines the frequency and criticality of each interaction into a single metric called interaction intensity. This is consistent with Marsden and Campbell (1984), who found closeness or intensity as best indicators of unobserved tie strength. More recently, Hansen (1999, 2002) combined frequency and closeness into a single metric called *interunit tie weakness*. The criticality component of our metric allows asymmetry in the interaction intensity of each pair of teams. After completing data purification, we identified a total of 680 nonzero technical team interactions within the organization, with mean (sd) intensity of 2.37 (1.42), of which 423 interactions were between the 54 component teams.

Similar to previous research in technical communication (e.g., Allen 1977, Van den Bulte and Moenaert 1998), we chose the presence or absence of *significant* information exchange as the binary variable of interest. We define *significant* information exchange as those technical interactions that were relevant during the design phase due to their criticality and/or frequency. Such interactions are captured by a nonzero interaction intensity in our scale. We organized the interaction data into a square (60×60) team interaction matrix (Figure 4), whose off-diagonal cells marked "O" indicate each *significant* team interaction revealed.

Figure 3 Design Interface Matrix



W = WEAK design interface; S = STRONG design interface.

5.3. Mapping Design Interfaces and Team Interactions

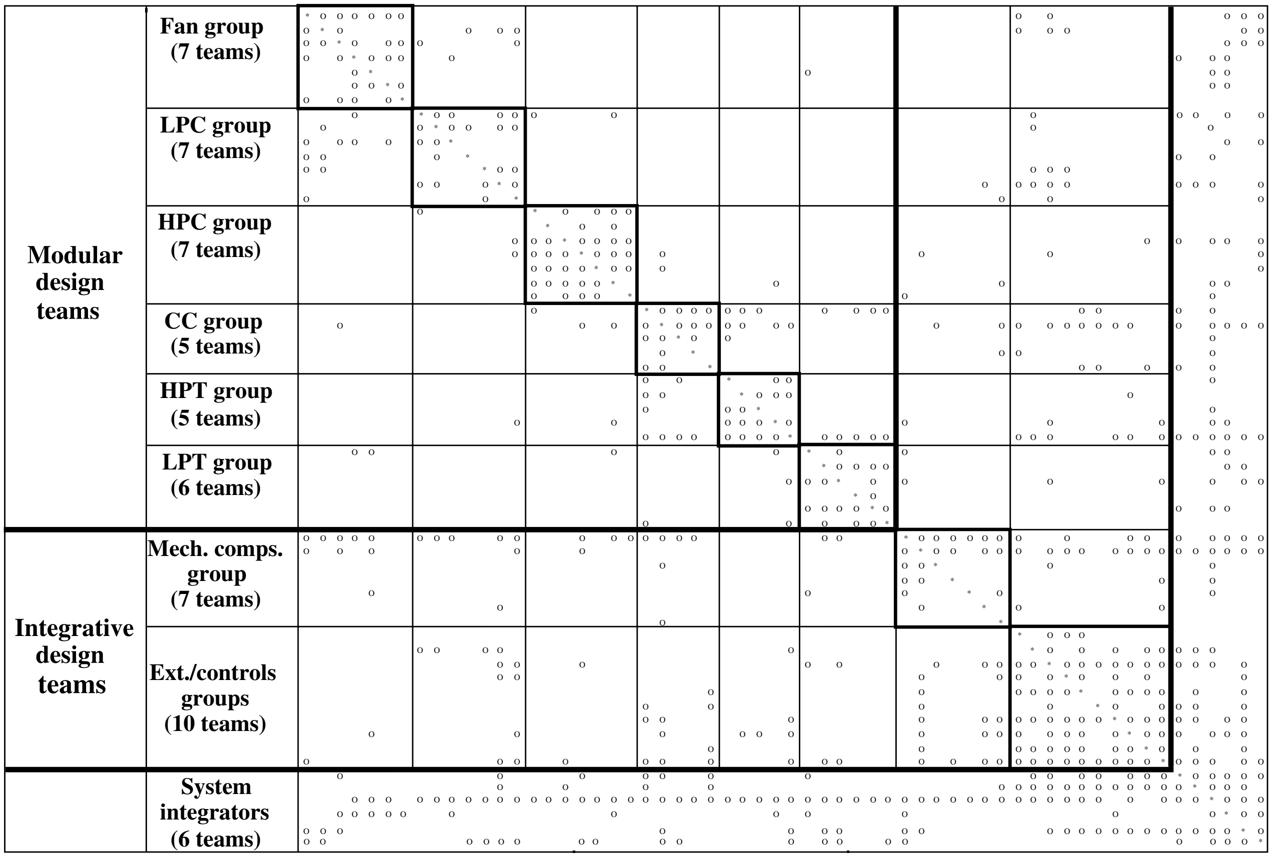
The one-to-one assignment of the 54 components to the 54 design teams allows the direct comparison of the design interface matrix with the team interaction matrix. Hence, by overlaying the design interface matrix with the team interaction matrix, we obtain the alignment matrix exhibited in Figure 5. Note that we omitted the six integration teams from this analysis. These teams interact with almost every component design team in the organization, which prevents us from inferring any particular communication pattern in which they were involved (see online Appendix D for further discussion).

The alignment matrix provides the basis for the analysis completed to test the hypotheses posed above. Figure 6 exhibits the possible states for each cell of the alignment matrix. As expected, the majority of the cases (90% of the cells) show aligned presence or absence of design interfaces and team interactions. The unexpected cases accounted for 10% of the cells; these were the 220 *unmatched design interfaces* (39% of the 569 design interfaces), and the 74 *unmatched team interactions* (17% of the 423 team interactions). A descriptive categorical data analysis tentatively supports all the hypotheses posed in §3, except Hypothesis 3 (see online Appendix A for details). Yet, for proper hypothesis testing we need to control for nonrandom tendencies typically embedded in network data.

6. Statistical Network Analysis

Similar to social network data, in our data set each component and design team appears as many times as they share interfaces or interact with others, resulting in observations that are clearly not

Figure 4 **Team Interaction Matrix (Binary)**



independent. By visually inspecting both the design interface and the team interaction matrices, we can observe strong tendencies for reciprocation of ties as well as significant concentration of ties within boundaries (i.e., clustering of ties). Other tendencies that can be present in our data are propensities of components and teams to generate or attract linkages. Such deviations from randomness embedded in our network data make our statistical analysis problematic. We tackle this challenge by considering two statistical network approaches: *log-linear* p_1 and *logit* p^* analyses. We also considered quadratic assignment procedure (QAP) (Krackhardt 1988), however, given the strong tendencies for reciprocation and clustering in our data, we found the use of QAP to be less suitable.

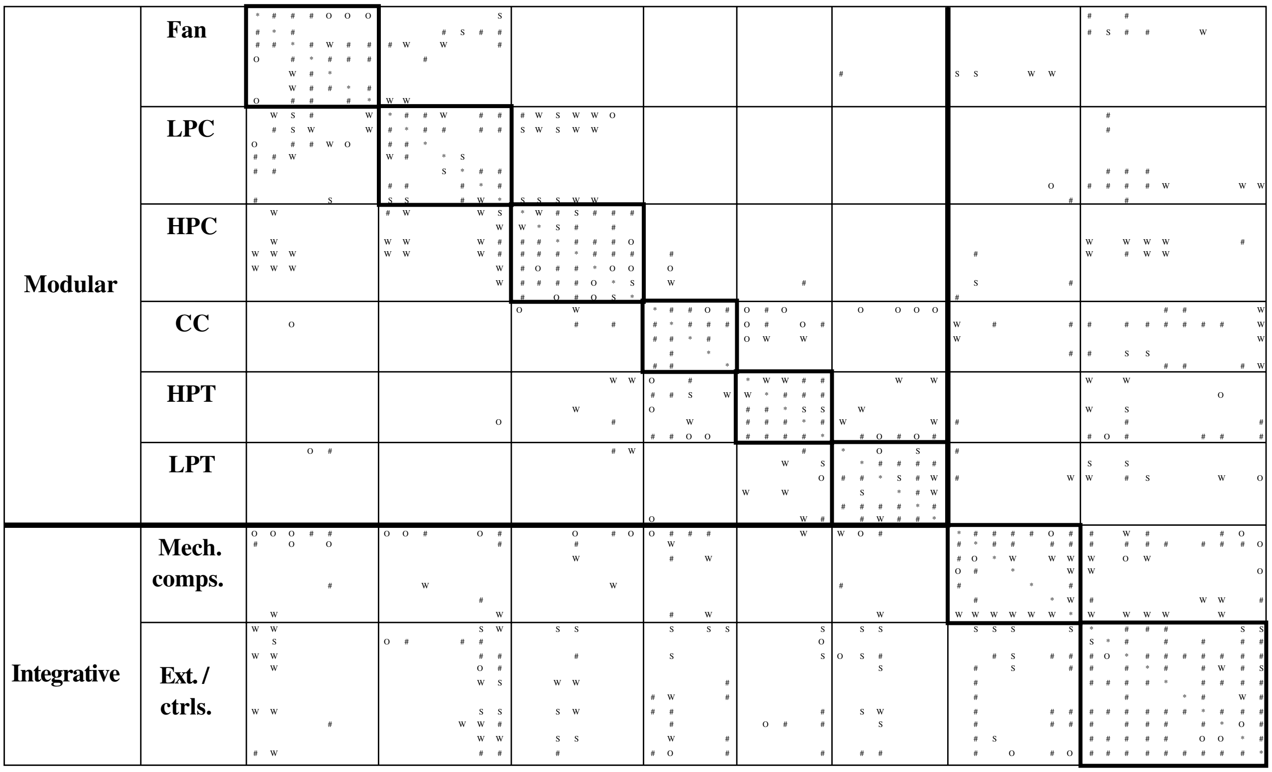
6.1. *log-linear* p_1 Analysis

We built log-linear models of the alignment matrix based on the p_1 distribution introduced by Holland and Leinhardt (1981). Similar to Van den Bulte and Moenaert (1998), in five steps we build a log-linear model for the probabilities of the dyads of our alignment matrix to test Hypotheses 1 and 6. We then construct additional *log-linear* p_1 models that consider discrete design interfaces to test the effects of design interface strength (Hypotheses 2 and 3). Although this statistical modeling approach well suits our research problem, its independence dyad assumption could be limiting and unrealistic (Wasserman and Faust 1994, Chapter 15). Because p_1 models do not explicitly handle triadic effects, we are not able to use them to test our hypotheses concerning indirect relations (Hypotheses 4 and 5). Nonetheless, we used this analysis to validate the results obtained from our main statistical approach, the *logit* p^* analysis. Details of our *log-linear* p_1 analysis are included in online Appendix B.

6.2. *logit* p^* Analysis

To address the limitations of p_1 models, a new generation of exponential family models, p^*, was developed by Wasserman and Pattison (1996). These models not only release the independent dyad assumption, but also allow researchers to formulate them in a standard *response-explanatory variables* form in which the response variable is the log odds (or logit) of the probability that a network tie is present, and the explanatory variables can be either any hypothesized network structure or network actor attribute. We describe, in online Appendix C, how we build specific members of the p^* family to model our alignment matrix and

Figure 5 Alignment Matrix



W: Unmatched WEAK interface; S: Unmatched STRONG interface; O: Unmatched Team interaction; #: Aligned presence of interface and interaction; (Blank): Aligned absence of interface and interaction; *: Diagonal elements (meaningless).

test our hypotheses. We complete our *logit p** analysis in three steps.

6.2.1. Step 1: Define Hypothesized Network Effects. Our *logit p** formulation includes basic dyadic and triadic effects typical of network data (Anderson et al. 1999, p. 46) for both design interfaces and team interactions as well as bivariate effects captured by our alignment matrix. Refer to online Appendix C for parameters and associated network statistics definitions.

To test our hypotheses, we define structural variables as ACROSS_{ij} and MODULAR_{ij} to capture whether tie ij is across boundaries and between modular systems, respectively. Note that by including ACROSS_{ij} into our models, we capture the clustering effects due to organizational and systems boundaries embedded in both design interface and team interaction matrices. Using these structural variables and the bivariate network effects described in online Appendix C, we define formal tests for our hypotheses as follows:

Hypothesis 1. $\theta_{\text{ACROSS},12} < 0$.

Hypothesis 2. $\theta_{\text{STRONG},2} > \theta_{\text{WEAK},2}$.

Hypothesis 3. $\theta_{\text{ACROSS},\text{STRONG},2} > \theta_{\text{ACROSS},\text{WEAK},2}$.

Hypothesis 4. $\tau_{221} > 0$.

Hypothesis 5. $\tau_{112} > 0$.

Hypothesis 6. $\theta_{\text{ACROSS},\text{MODULAR},12} < 0$.

6.2.2. Step 2: Estimate Parameters by Fitting Our *logit p Models to Observed Data.** Fitting a *logit p** model to data can be done (albeit approximately) by adopting the pseudo-likelihood estimation strategy discussed by Wasserman and Pattison (1996), Pattison and Wasserman (1999), and Robins et al. (1999). This approach assumes that the logits, ϖ_{ijm}, of the conditional probabilities are statistically independent. Hence, maximizing the pseudo-likelihood function is equivalent to fitting a logistic regression model to the logits, ϖ_{ijm}, using standard computing packages (we used SPSS 11.0). Note that the explanatory variables in the logistic regressions are the *difference* network statistics, that is, the change in network statistic from tie (ij, m) being present to being absent. Hence, before

Figure 6 Overall Results

Team Interactions	NO (2439)	**W** (150)	**S** (70)	(2219)
	YES (423)	$\#_W$ (169)	$\#_S$ (180)	**O** (74)
		YES (569)		NO (2293)
		Design Interfaces		

fitting any of the models, we need to pre-process the observed data to calculate the change statistic for each relational tie X_{ijm}. (X_{ijm} is the observed tie for pair (i, j) of type m, where $m=1$ for design interfaces, and $m=2$ for team interactions. We modified SPSS codes to calculate the change statistics of interest. The codes are available from the authors upon request.)

Following Anderson et al. (1999, p. 49), we assess (approximately) the statistical importance of any explanatory variable by evaluating the difference in pseudo-likelihood ratio statistics (G^2_{PL}) by referring it to the appropriate χ^2 distribution. We also evaluate (approximately) the significance of each parameter by comparing their pseudo-Wald statistics (Wald_{PL}) to the appropriate χ^2 distribution. Table 1 shows the results of fitting six dichotomous bivariate *logit* p^* models to test Hypotheses 1, 4, 5, and 6. Table 2 shows the results of fitting four trichotomous *logit* p^* models for testing Hypotheses 2 and 3. For brevity, Tables 1 and 2 exhibit parameters corresponding to alignment network effects only. (See online Appendix C for dyadic and triadic parameters associated with both design interfaces and team interactions.)

6.2.3. Step 3: Interpret Parameters from Logistic Regressions. In general, a significantly positive parameter indicates a tendency for the associated configuration to occur in the network, whereas negative parameters suggest a lack of presence. In Table 1, Model 1 includes an insignificant exchange parameter (ρ_{12}), which indicates that design interfaces are not likely to be reciprocated by team interactions (nor vice versa). Most of the improvement in fit of Model 1 is due to the significantly positive association parameter (θ_{12}), which indicates a strong general tendency for design interfaces and team interactions to be aligned.

We add the effects of group boundaries in Model 2, which shows significantly negative clustering parameters ($\theta_{\text{ACROSS},1}$ and $\theta_{\text{ACROSS},2}$) indicating, as expected, a strong tendency for both design interfaces and team interactions to be clustered within boundaries. The third-order parameter, $\theta_{\text{ACROSS},12}$, is significantly positive, which indicates, contrary to Hypothesis 1, that the tendency for design interfaces and team interactions to be aligned is stronger across boundaries. Similar to our *log-linear* p_1 results, Model 2 still shows a negative overall effect due to group boundaries. That is, Model 2 predicts that the overall probability for design interfaces and team interactions to be aligned is lower across boundaries than within boundaries (Robins et al. 1999).

Model 3 shows no statistically significant parameters associated with indirect effects. Model 4, however, shows that indirect effects are statistically significant within boundaries. More specifically, the significantly positive $\tau_{\text{WITHIN},221}$ parameter indicates that there is stronger propensity for indirect team interactions configurations to be present within boundaries (in line with Hypothesis 4). On the other hand, the significantly negative $\tau_{\text{WITHIN},112}$ parameter indicates that indirect design interface configurations are much less likely to occur within boundaries than across boundaries.

Table 1 Results of *logit* p^* Analysis (Dichotomous Relations)

Parameters	Model 1 association	Model 2 across	Model 3 indirect	Model 4 ind. within	Model 5 modular	Model 6 all
Alignment effects						
ρ_{12}	−**0.242** (2.124)	−0.224 (1.740)	−0.242 (1.970)	−0.196 (1.282)	−0.198 (1.343)	−0.166 (0.900)
θ_{12}	**2.769** (329.614)	1.917 (46.185)	2.121 (54.088)	2.203 (56.673)	1.807 (22.092)	2.227 (29.711)
Clustering effects and boundary effects (Hypothesis 1)						
$\theta_{\text{ACROSS},1}$		−**0.803** (9.864)	−0.860 (10.522)	−1.807 (20.093)	−0.771 (9.050)	−1.738 (18.194)
$\theta_{\text{ACROSS},2}$		−**1.895** (35.314)	−1.908 (35.209)	−1.659 (15.251)	−1.816 (31.969)	−1.462 (11.508)
$\theta_{\text{ACROSS},12}$		**1.013** (10.681)	1.077 (11.721)	0.977 (9.315)	1.375 (11.924)	1.265 (8.937)
Effects of indirect team interactions and indirect design interfaces (Hypotheses 4 and 5)						
τ_{221}			−**0.004** (0.004)	−0.004 (0.004)		0.000 (0.000)
τ_{112}			−**0.082** (2.732)	−0.061 (1.490)		−0.059 (1.392)
$\tau_{\text{WITHIN},221}$				**0.228** (5.746)		0.258 (7.183)
$\tau_{\text{WITHIN},112}$				−**0.330** (11.809)		−0.337 (12.004)
Effects of systems modularity (Hypothesis 6)						
$\theta_{\text{MODULAR},1}$					**0.230** (1.359)	0.196 (0.941)
$\theta_{\text{MODULAR},2}$					**0.681** (6.866)	0.701 (7.034)
$\theta_{\text{MODULAR},12}$					**0.117** (0.086)	−0.089 (0.045)
$\theta_{\text{ACROSS,MODULAR},12}$					−**0.749** (3.003)	−0.687 (2.283)
# Parameters	16	19	23	25	23	29
G^2_{PL}	2,021.192	1,984.473	1,955.779	1,944.521	1,974.203	1,934.936

Notes. Wald_{PL} statistics are shown in parentheses. For approximate statistical inference we compare Wald_{PL} against χ^2. Hence, $p < 0.1$ if $\text{Wald}_{\text{PL}} > 2.706$. Models 3, 4, and 6 also include lower-order parameters $\sigma_{\text{I},12}$ and $\sigma_{\text{O},12}$. For Models 4 and 6, we define WITHIN_{ij} to capture whether tie ij is within boundaries.

Table 2 Results of *logit p Analysis with Trichotomous Design Interface Strength**

Parameter	Model 1 (exchange)	Model 2 (clustering)	Model 3 (association)	Model 4 (assoc. across)
$\rho_{\text{WEAK},2}$	**1.105** (55.724)	0.950 (38.780)	−0.264 (2.260)	−0.187 (1.116)
ρ_{STRONG2}	**1.305** (54.383)	1.017 (30.788)	−0.470 (4.578)	−0.404 (3.234)
$\theta_{\text{ACROSS,WEAK}}$		−**1.151** (23.029)	−0.663 (6.342)	−1.090 (14.743)
$\theta_{\text{ACROSS,STRONG}}$		−**1.445** (20.305)	−0.876 (5.465)	−1.317 (10.223)
$\theta_{\text{ACROSS},2}$		−**1.409** (37.008)	−1.193 (21.145)	−1.860 (35.149)
$\theta_{\text{WEAK},2}$			**2.625** (259.769)[a]	1.718 (29.792)
$\theta_{\text{STRONG},2}$			**3.078** (231.840)[a]	2.403 (46.674)
$\theta_{\text{ACROSS,WEAK},2}$				**1.138** (10.857)[b]
$\theta_{\text{ACROSS,STRONG},2}$				**0.877** (4.938)[b]
# of parameters	28	31	33	35
G^2_{PL}	2,833.541	2,755.417	2,347.819	2,335.305

Notes. Wald_{PL} statistics are shown in parentheses. For approximate statistical inference we compare Wald_{PL} against χ^2. Hence, $p < 0.1$ if $\text{Wald}_{\text{PL}} > 2.706$.

[a] To test that $\theta_{\text{STRONG},2} > \theta_{\text{WEAK},2}$, we estimate a reduced model with a single parameter ($\theta_{1,2}$), whose $G^2_{\text{PL}} = 2{,}352.179$. Hence, $\Delta G^2_{\text{PL}} = 4.360$, $\Delta df = 1$, and $p < 0.05$.

[b] To test that $\theta_{\text{ACROSS,STRONG},2} > \theta_{\text{ACROSS,WEAK},2}$, we estimate a reduced model with a single parameter ($\theta_{\text{ACROSS},1,2}$), whose $G^2_{\text{PL}} = 2{,}335.637$. Hence, $\Delta G^2_{\text{PL}} = 0.332$, $\Delta df = 1$, and $p > 0.1$.

Model 5 includes the effects of system modularity. The model suggests that modularity does not directly influence the alignment of interfaces and interactions (i.e., insignificant $\theta_{\text{MODULAR},12}$), however, its significantly negative $\theta_{\text{ACROSS, MODULAR},12}$ parameter indicates that when considering the cases across boundaries, the pure propensity of design interfaces and team interactions to be aligned is significantly lower between modular systems (in line with Hypothesis 6). These findings coincide with our p_1 results.

Finally, Model 6 includes all the effects together. All relevant parameters are still significant with the exception of $\theta_{\text{ACROSS,MODULAR},12}$, which became slightly nonsignificant (i.e., $p = 0.131$). Further analysis not included here indicates that $\theta_{\text{ACROSS, MODULAR},12}$ becomes nonsignificant only in the presence of indirect design interface effects within boundaries ($\tau_{\text{WITHIN},112}$). As a result, we conclude that the proportion of misaligned cases is significantly greater across modular systems (Model 5), but some of those misaligned cases coincide with indirect design interfaces (Model 6). We could not cross-validate this result with our p_1 analysis due to its inability to capture triadic effects.

In Table 2, Model 2 includes clustering effects resulting in significantly negative parameters ($\theta_{\text{ACROSS, WEAK}}$, $\theta_{\text{ACROSS, STRONG}}$, and $\theta_{\text{ACROSS},2}$) confirming that both team interactions and design interfaces (at both levels) tend to be clustered within boundaries. Model 3 includes statistically significant association parameters. In line with Hypothesis 2, we found that strong design interfaces are more likely to be aligned with team interactions than are weak design interfaces (i.e., $\theta_{\text{STRONG},2}$ is significantly greater than $\theta_{\text{WEAK},2}$). Model 4 includes third-order parameters to capture whether there is a significant difference of the association effect across boundaries. We found that $\theta_{\text{ACROSS, WEAK},2}$ and $\theta_{\text{ACROSS, STRONG},2}$ are not significantly different, indicating that effects due to design interface strength do not dominate over organizational and system boundaries effects (contrary to Hypothesis 3). We obtained similar results in our *log-linear* p_1 analysis.

7. Discussion of Results

To properly test our hypotheses, we built several p_1 and p^* models of the alignment matrix. As expected, we found a strong tendency for design interfaces and team interactions to be aligned throughout the network. Even this basic result is tremendously relevant, as it suggests that managers should be able to explicitly examine the product architecture to plan for cross-team interactions when organizing design teams. However, our results also indicate that managers need to be wary of factors that may prevent a perfect alignment of interfaces and interactions.

When evaluating the effects of boundaries in our statistical models, we found that clustering effects are very strong, both in the product and organizational domains. Surprisingly, group boundaries did not hinder the alignment of design interfaces and team interactions (as hypothesized in Hypothesis 1) but significantly strengthened such alignment. Yet we observe a significantly lower proportion of aligned cases across boundaries. The reason for this apparently contradictory result is that the effects of organizational/system boundaries have two components, a *clustering* component and a *pure alignment* component, which when considered jointly result in smaller probabilities of finding aligned ties across boundaries. By disentangling clustering and pure alignment effects in our models, we found that the latter is significantly stronger across boundaries. Because clustering parameters capture the capability of product and organizational structures to group interdependences within boundaries, the denser the clusters become, the greater the opportunity for alignment within boundaries and the lesser the opportunity for alignment across boundaries. Thus, the main hindering effect of organizational and system boundaries is to lower the *expected* level of alignment across boundaries. In our study, 52% of the cross-boundary design interfaces unmatched by team interactions and 25% of the cross-boundary team interactions unmatched by design interfaces are actually better than expected, considering such strong clustering effects. As a result, managers should still *expect* (and prepare for) a significantly greater proportion of misaligned cases across boundaries.

At Pratt & Whitney, managers made special efforts to handle cross-boundary interdependences by using an integration tool called a Component Requirements Document (CRD). The purpose of this tool was twofold: to encourage design optimization by breaking down design requirements to the system level, and to encourage cross-boundary interactions by having teams regularly update the document. Even with the use of this tool there were both unmatched interactions and unmatched interfaces. For example, a team in the low-pressure turbine (LPT) group realized that they needed to meet with a Fan system team to estimate the impact of a Fan test requirement that would transmit high loads throughout the engine. However, in two other cases teams adhered to the stated requirements and did not feel the need to interact (across group boundaries) to review their interfaces. Had they done so, they would have discovered additional load transfer interfaces not explicitly defined in the CRD that were left unchecked and led to problems later. These examples illustrate the difficulty of managing cross-boundary interdependences. On the one hand, teams themselves discover unknown interfaces which are more likely to be across system boundaries, and on the other hand, known interfaces are more likely to be mismanaged when they occur across boundaries. This illustrates our need to better understand which factors moderate the alignments across (and within) boundaries.

We found that the stronger the design interface, the greater the likelihood that teams would interact (Hypothesis 2), which is in line with previous research about team interdependence. Although this result could be considered as "good news" for managers who might believe design interface strength drives the alignment of design interfaces and team interactions across organizational/system boundaries (Hypothesis 3), we did not find empirical support for this latter hypothesis. This can be interpreted as "bad news" for managers because even if cross-boundary interfaces are critical, the likelihood that they are unmatched by the corresponding interaction may be the same as if they were noncritical interfaces. Follow-up interviews with engineers in our study qualitatively corroborated these results. They confirmed that many strong cross-boundary design interfaces were perceived as weak interfaces and, therefore, no planning mechanisms were in place to address them. This indicates that managers must identify and manage critical cross-boundary interfaces without relying on their level of importance as a mechanism to improve their alignment with interactions.

We found significant evidence that the effects of indirect interactions (Hypothesis 4) exist within group boundaries. This suggests that design teams use other intermediary teams (most likely within their groups) to obtain relevant technical information. In our study, many design interfaces included spatial dependencies that were not supposed to change due to the derivative requirements of the product. Yet while the spatial dependencies were supposed to remain unchanged, those interfaces had other functional design dependencies (such as structural or thermal loadings) that did change. Teams within organizational boundaries recognized unplanned functional changes by nature of their "proximity to the action." They had interactions with common teams in their groups in which they discovered and effectively reviewed unplanned changes from other teams that affected them.

Follow-up interviews indicated that the effects of indirect design interfaces (Hypothesis 5) existed across components of some modular systems (e.g., combustion chamber (CC) and Fan systems), yet our results do not allow us to generalize such a qualitative observation throughout the product. However, we found that the propensity of finding unmatched team interactions covering indirect design interfaces within boundaries is significantly lower than across boundaries, which coincides with our qualitative observation.

When studying the effects of system modularity, we found evidence that the hindering effects of system and organizational boundaries are more severe between modular systems than with integrative systems. This result partially supports the categorical data analysis presented in Sosa et al. (2003), because although the moderating effects of system modularity are significant when controlled for basic dyad and triadic effects, they become nonsignificant when controlled for the effects of indirect design interfaces. This indicates that some of the unmatched team interactions (across modular systems) can be considered "good" misalignment cases because they occurred to resolve system-level dependencies, such as those related to the Fan test requirement. On the other hand, our results also show a significantly larger proportion of unmatched design interfaces across modular systems. This poses an important consideration for managers developing products that involve modular systems because it suggests that modularization itself may further hinder design teams' ability to handle interfaces across boundaries. In follow-up interviews at Pratt & Whitney, some teams reported to be more apt at handling integrative rather than modular system interfaces due to a tendency for those integrative interfaces to impact more than one aspect of their design. For example, the Intermediate Case team of the low-pressure compressor (LPC) group was highly dependent on detailed definition of the engine oil and secondary flow systems, for which it interacted regularly with the externals and controls team.

7.1. Impact on Performance

Many design interfaces across boundaries were unmatched by team interactions because they were either weak or perceived as weak interfaces. One reason for these unmatched interfaces is that teams across boundaries did not have opportunities for indirect interactions to communicate or discover changes associated with them. We found this to be particularly relevant for structural and thermal design dependencies. The impact of these missed weak interfaces was typically a very small reduction in performance or durability of affected components and systems. Given the 25- to 30-year product life expectancy, however, even these small performance deviations may cause significant warranty or service expense over the life of the product. Hence, the need for careful attention to all cross-boundary interfaces.

In contrast, the programmatic impact of missing strong design interfaces across boundaries could be dramatic. While the PW4098 engine development program set new industry standards in development speed and cost, there were still major setbacks during the program. Two of these resulted from design teams from different modular systems who did not capture strong design interfaces between them. The costs associated with the two unmatched interfaces were related to the time in which it took for the problems to be discovered. One caused excessive loads on assembled hardware in early development tests, resulting in severely distressed hardware and special disassembly procedures. This resulted in significant cost and delays in the program to redesign the components affected and rebuild the test engines. The other also caused excessive loads and reduced life to a critical engine component, but was not discovered until the first engines entered service. This problem cost substantially more to rectify, as it affected engines both in production and in development tests.

There were 25 unmatched team interactions across modular systems, many of which corresponded to unidentified design interfaces. Many of these were reportedly related to investigations into possible engine-level design conditions which manifested in adverse structural or thermal load transmission or insufficient pressures. Some teams were using their experience with prior generation engines to uncover new direct and indirect design interface characteristics prior to the development of tests where they would be evaluated. This type of team interaction is almost universally positive as it serves to improve product performance and reduce downstream design iterations.

8. Conclusions and Implications

Previous research has studied product architecture decisions and technical communication patterns in product development from separate viewpoints. Here we integrated these two perspectives to study and explain the misalignment of product and organizational structures during detailed design of complex development efforts. This work contributes to the product innovation literature, in both product architecture and organizational perspectives, by uncovering factors that impact the likelihood that (1) product-related interdependences are not addressed by team interactions, and (2) design teams interact despite the absence of a product-related interdependence between them.

Our results show not only that the likelihood of misalignment is greater across organizational and system boundaries, but also that weak and strong interfaces may be equally affected by boundary effects, that indirect interactions are an important coordination mechanism within boundaries, and that system modularity may prevent the alignment of interfaces and interactions across boundaries.

From an analysis viewpoint, we illustrate how to formally build statistical models based on social networks methods for proper hypothesis testing using DSM-type data. We use a novel statistical technique (the *logit* p^*, developed by Wasserman and Pattison 1996) to estimate statistical models that control for dyadic and triadic network effects. We also illustrate how the use of p^* models opens up new avenues for researchers interested in testing the effects of network structures that involve three players. Although the p^* formulation is very robust from a statistical modeling viewpoint, fitting these models to data is still done approximately. We also tested the robustness of our results by completing a log-linear p_1 analysis. Future research will benefit from ongoing efforts focusing on alternative fitting strategies of p^* models.

As in many other empirical studies that collected data in a single organization (e.g., Morelli et al. 1995, Van den Bulte and Moanaert 1998), we cannot claim the generality of our findings before completing similar studies in other types of products in different industries. However, we would expect to obtain analogous results in other projects developing complex systems and where teams are organized according to the product architecture, as we observe to be the case in automobile and aerospace industries. This study is descriptive in nature and as such, we avoid drawing explicit prescriptive conclusions.

8.1. Managerial Implications

In addition to the managerial repercussions of each result, as discussed above, this research has important implications for managers from two different perspectives. From a strategic viewpoint, Henderson and Clark (1990, p. 28) highlighted the fact that "learning about changes in the architecture of the product is unlikely to occur naturally. Learning about changes in architecture—about new interactions across components (and often across functional boundaries)—may therefore require explicit management and attention." By documenting the architecture of the product for every generation of a product family, managers can identify key differences (i.e., new or removed interfaces) between old and new architectures. By building an alignment matrix, managers have a compact and visual representation that allows them to diagnose how their organization addresses design interfaces of the product under development. Furthermore, the alignment matrix helps managers pinpoint their efforts to align team interactions with design interfaces to effectively develop distinct product architectures.

From a project management perspective, our approach helps managers integrate activities of design teams across organizational and functional boundaries. This is particularly beneficial in projects of incremental and modular innovation, in which the product architecture is well understood. Our analysis suggests that managers should focus their efforts on understanding the *causes* of unmatched design interfaces and unmatched team interactions across modular systems. These are the design interfaces most difficult to identify or be addressed by the corresponding design teams, even if they are critical design interfaces. For example, some of the 25 unmatched team interactions between modular systems in our study were critical design interfaces that had not been previously identified by design experts. As a result of our study, managers learned about these interdependencies and established dedicated design teams or formally extended the responsibility of existing teams to explicitly handle these critical cross-boundary design interfaces during the development of the next engine. These teams were also held responsible for managing the unmatched design interfaces that resulted in the problems described in §7.1. For example, two teams were formed to manage the burner profile effects on downstream components and the fan blade-out loads throughout the engine. The implementation of our approach provided a structured way to identify which design interfaces these teams would manage in future engine designs.

8.2. Research Implications

This paper opens a new stream of research on the interface of product architecture and organizational structure. By uncovering the existence of unmatched team interactions, we provide empirical evidence that product ambiguity exists, and it is more likely to be present across organizational and system boundaries. Which components are more likely to have unknown interfaces? How can managers of complex design efforts discover those unknown interfaces? We also provide evidence suggesting that teams may fail to perceive the actual criticality of their cross-boundary design interfaces. What architectural and organizational mechanisms influence teams' cognitive capabilities across boundaries? Our results also indicate that indirect interactions act as an important coordination mechanism within boundaries. We need to better understand what types and conditions of indirect interactions contribute to the performance of complex development projects, and how they can be promoted. All these questions are important and merit further research in both engineering design and management science domains.

We studied the static effects that influence the alignment of interfaces and interactions. An interesting methodological and statistical challenge for future research is to explore the evolution over time of such alignment. Are alignment matrices in a product family more likely to exhibit an increasing proportion of unmatched design interfaces and team interactions as product families evolve? Finally, this paper provides some limited examples to illustrate the importance of certain kinds of misalignment, however, further systematic research is needed to understand their performance implications. To obtain the greatest benefit from preemptively changing organizational or product structures, it is critical to understand what kinds of misalignments are most costly and why. Moreover, under what circumstances is an organizational design that mirrors the architecture of the product a good one?

An online supplement with the appendices is available at http://mansci.pubs.informs.org/ecompanion.html.

Acknowledgments

Funding of this research was provided by the MIT Center for Innovation in Product Development. The authors appreciate the assistance of the engineers and managers at Pratt & Whitney Aircraft and insightful comments by Christophe Van den Bulte, Dawn Iacobucci, William Lovejoy, Christoph Loch, Luk Van Wassenhove, the associate editor, and three anonymous reviewers. The authors also thank Philippa Pattison for insights in data analysis issues.

References

Adler, P. S. 1995. Interdepartmental interdependence and coordination: The case of the design/manufacturing interface. *Organ. Sci.* **6** 147–167.

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

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

Ancona, D., D. Caldwell. 1992. Bridging the boundary: External activity and performance in organizational teams. *Admin. Sci. Quart.* **37** 634–665.

Anderson, C., S. Wasserman, B. Crouch. 1999. A p^* primer: Logit models for social networks. *Social Networks* **21** 37–66.

Baldwin, C., K. Clark. 2000. *Design Rules: The Power of Modularity.* MIT Press, Cambridge, MA.

Brown, S. L., K. M. Eisenhardt. 1995. Product development: Past research, present findings, and future directions. *Acad. Management Rev.* **20** 343–378.

Cummings, J. 2004. Work groups, structural diversity, and knowledge sharing in a global organization. *Management Sci.* **50**(3) 352–364.

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

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

Granovetter, M. 1973. The strength of weak ties. *Amer. J. Sociology* **78**(6) 1360–1380.

Hansen, M. 1999. The search-transfer problem: The role of weak ties in sharing knowledge across organization subunits. *Admin. Sci. Quart.* **44**(1) 82–111.

Hansen, M. 2002. Knowledge networks: Explaining effective knowledge sharing in multiunit companies. *Organ. Sci.* **13**(3) 232–248.

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

Holland, P. W., S. Leinhardt. 1981. An exponential family of probability distributions for directed graphs. *J. Amer. Statist. Assoc.* **76** 33–51.

Krackhardt, D. 1988. Predicting with networks: Nonparametric multiple regression analysis of dyadic data. *Social Networks* **10** 359–381.

Krishnan, V., K. Ulrich. 2001. Product development decisions: A review of the literature. *Management Sci.* **47**(1) 1–21.

Loch, C., C. Terwiesch. 1998. Communication and uncertainty in concurrent engineering. *Management Sci.* **44**(8) 1032–1048.

Marsden, P. V. 1990. Network data and measurement. *Ann. Rev. Sociology* **16** 435-463.

Marsden, P. V., 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. *Management Sci.* **46**(6) 733–750.

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

Pahl, G., W. Beitz. 1991. *Engineering Design, A Systematic Approach.* Springer-Verlag, New York.

Pattison, P., S. Wasserman. 1999. Logit models and logistic regressions for social networks: II. Multivariate relations. *British J. Math. Statist. Psych.* **52** 169–193.

Pich, M., C. Loch, A. De Meyer. 2002. On uncertainty, ambiguity, and complexity in project management. *Management Sci.* **48**(8) 1008–1023.

Pimmler, T. U., S. D. Eppinger. 1994. Integration analysis of product decompositions, *Proc. ASME Design Theory Methodology Conf.*, DE-Vol. 68, 343–351.

Robertson, D. C., T. J. Allen. 1992. Managing CAD systems in mechanical design engineering. *IEEE Trans. Engrg. Management* **39**(1) 22–31.

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

Sanchez, R., J. T. Mahoney. 1996. Modularity, flexibility, and knowledge management in product and organization design. *Strategic Management J.* **17** (Winter Special Issue) 63–76.

Schrader, S., W. Riggs, R. Smith. 1993. Choice over uncertainty and ambiguity in technical problem solving. *J. Engrg. Tech. Management* **10** 73–99.

Smith, R. P., S. D. Eppinger. 1997. Identifying controlling features of engineering design iteration. *Management Sci.* **43** 276–293.

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

Sosa, M. E., S. D. Eppinger, M. Pich, D. McKendrick, S. Stout. 2002. Factors that influence technical communication in distributed product development: An empirical study in the telecommunications industry. *IEEE Trans. Engrg. Management* **49**(1) 45–58.

Steward, D. 1981. The design structure matrix: A method for managing the design of complex systems. *IEEE Trans. Engrg. Management* **EM-28**(3) 71–74.

Stone, R. B., K. L. Wood, R. H. Crawford. 2000. A heuristic method for identifying modules for product architecture. *Design Stud.* **21**(1) 5–31.

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

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

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

Tushman, M. L., R. Katz. 1980. External communication and project performance: An investigation into the role of gatekeeper. *Management Sci.* **26**(11) 1071–1085.

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

Ulrich, K. T., S. D. Eppinger. 2004. *Product Design and Development*, 3rd ed. McGraw Hill, New York.

Van den Bulte, C., R. K. Moenaert. 1998. The effects of R&D team co-location on communication patterns among R&D, marketing, and manufacturing. *Management Sci.* **44** S1–S19.

Wasserman, S., K. Faust. 1994. *Social Network Analysis.* Cambridge University Press, New York.

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

Whitney, D. 2004. *Mechanical Assemblies: Their Design, Manufacturing, and Role in Product Development.* Oxford University Press, New York.

Yassine, A., N. Joglekar, D. Braha, S. Eppinger, D. Whitney. 2003. Information hiding in product development: The design churn effect. *Res. Engrg. Design* **14**(3) 131–144.