Understanding the Effects of Product Architecture on Technical Communication in Product Development Organizations

Manuel E. Sosa
Steven D. Eppinger
Craig M. Rowles

August 2000
Working Paper Number 4130

Prof. Steven D. Eppinger
Sloan School of Management
Massachusetts Institute of Technology
Cambridge, MA 02142-1347
eppinger@mit.edu
Understanding the Effects of Product Architecture on Technical Communication in Product Development Organizations

Manuel E. Sosa
Department of Mechanical Engineering
Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

Steven D. Eppinger
Sloan School of Management
Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

Craig M. Rowles
Advance Engine Program
Pratt & Whitney Aircraft, East Hartford, Connecticut

ABSTRACT
Effective communication in product development organizations is widely recognized to be a key element of product development performance. Furthermore, management of product architecture knowledge by the development organization provides important competitive advantage for established firms facing architectural innovation. This research studies how the combination of product architecture and organizational structure determines technical communication in development teams. By documenting and analyzing both the design interfaces between the components that comprise a product and the technical interactions between the teams that design each of these components, we learn how the architecture of the product and the layout of the organization drive development team interactions. Several hypotheses are formulated to explain the unexpected cases when: 1) known design interfaces are not matched by team interactions, and 2) observed team interactions are not predicted by design interfaces. We test the hypothesized effects due to organizational and system boundaries, and design interface strength. Hypotheses are tested using both categorical data analysis and log-linear network analysis. The research is conducted using data collected describing a large commercial aircraft engine development process.

Keywords: Product Architecture; Design Interfaces; System Integration; Team Interactions, Organizational Structure, Technical Communication.

1 Please address all the correspondence to this author. 50 Memorial Drive. E53-347. Cambridge, MA 02142. Tel: (617) 253-0468. Email: eppinger@mit.edu
1. Introduction

This paper introduces a method to understand to what extent the technical interactions between design teams are determined by both the product architecture and the structure of the organization which designs it. We apply our method to studying the development process of a large commercial aircraft engine. The objective of our study is not only to predict technical interactions between design teams, but also to understand which factors associated with both the architecture of the product and the organizational structure need to be taken into consideration to enable technical communication. This research effort aims to provide insights to improve planning of large development projects where the architecture of the product is known in advance.

This work is motivated by the crucial importance of product development in today's businesses and the need to improve our understanding of the communication process in development organizations. Much has been written about improvement of product development processes and in particular about the role of effective communication in product development teams. Allen (1977) initiated a stream of research to investigate how effective internal and external communications stimulate the performance of development organizations. Clark and Fujimoto (1991) related successful development in the auto industry to intensive communication between upstream and downstream activities. Wheelwright and Clark (1992) emphasized the need to improve technical communication when and where it certainly improves project performance.

Henderson and Clark (1990) conducted one of the very few studies focused on the coupling of product architecture and organizational structure. They introduced a framework to study the effects of product architecture innovation in established firms' development organizations. They suggested that architectural innovation threatens established firms not only because they are slow in recognizing novel architectures, but also because their development organizations possess architectural knowledge specific to the established product architecture.

Much of the research on technical communication focuses on how factors such as physical distance, organizational structures, task structures, and use of communication media affect technical communication (e.g., Allen 1977, Griffin and Hauser 1992, Morelli et al. 1995, McDonough III et al. 1999, Sosa et al. 2000b). Morelli et al. (1995) compare the actual communication network of a development organization with a predicted communication network.
based on the task structure of the project, however their method does not explicitly capture the product architecture. More recently, Van den Bulte and Moenaert (1998) address the problems associated with network data by using network analysis to study the communication network of a development organization before and after collocation. To the best of our knowledge, previous work has not considered the mapping of the communication network with other type of networks such as a product architecture network. An important contribution of this paper is our novel method to study not only the overlap but also the mismatch between the communication network associated with the development organization and the design interface network associated with the product architecture being developed.

**Designing Complex Products Requires Product Decomposition and Product Integration**

This paper addresses the problem of understanding technical communication in complex product development. We focus on the development of complex products, such as automobiles, computers, or aircraft engines. The general approach when developing complex products is to decompose the product into systems and, if the systems are still too complex, to decompose these into smaller sets of components (Alexander 1964, Simon 1981, Smith and Browne 1993, McCord and Eppinger 1993, Pimmler and Eppinger 1994, Eppinger 1997). Consequently, product architecture is defined as the scheme by which decomposed elements of a product are arranged into sets of components in order to meet its functional requirements (Ulrich 1995, Ulrich and Eppinger 1995).

From an organizational viewpoint, design teams are commonly organized around the architecture of the product. In most technical products we can observe a clear mapping between the product architecture and the development organization which designs it (McCord and Eppinger 1993, Pimmler and Eppinger 1994). Large development projects may involve the efforts of hundreds or even thousands of team members. A single team does not design the entire product at once (it is too complex). Rather, many teams develop the components, or systems, and work to integrate all of these components to create the final product (von Hippel 1990).

An important challenge faced by development organizations is product integration (Iansiti 1998). Design teams face two important levels of integration during the development of complex products: Function-level integration takes place within each cross-functional design team when they have to coordinate efforts in order to design their respective components. System-level integration takes place across design teams in order to integrate the components (designed by
each team) to assure the product works as an integrated whole. Furthermore, we distinguish two types of system-level integration efforts:

- Within-group system-level integration effort, which usually takes place between teams that design components of the same system.
- Across-group system-level integration effort, which usually takes place between teams that design components belonging to different systems.

To summarize, complex products are decomposed into systems, and these systems are further decomposed into components. The arrangement of these physical sets of components defines the architecture of the product. Similarly, development organizations are usually split into design teams that develop each of the components that comprise the product. Figure 1 illustrates the main research question we want to investigate: How do the architecture of a product and the system-level integration efforts between the design teams map into each other?

![Diagram](image)

**Figure 1. Research Question**

Within this context, we are particularly interested in answering the following questions:

- How accurately can we predict coordination-type communication by analyzing the coupling of product architecture and the structure of the development organization?
- Why do some design interfaces between components not correspond to technical interactions between the teams that design them?
Why do design teams that develop independent components still engage in technical interaction?

2. Research Method

This section describes our novel method of comparing the architecture of a product with the development organization which designs it. Our approach involves three steps:

1) Capture the product architecture. 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 further decomposed into components. We then ask them to identify the design interfaces between the components required for their functionality. We represent the product architecture in a design interface matrix.

2) Capture the development organization. We next identify the design teams responsible to develop the product’s components. We then survey key members of each team to capture the frequency and importance of the technical interactions between them, and thus assess the technical communications of the development organization. We represent the system-level integration efforts of the development organization in a team interaction matrix.

3) Compare the product architecture and the development organization. Finally, we compare the design interface matrix with the team interaction matrix to answer the research questions posed above.

We applied our approach to study the detail design period of the development of a large commercial aircraft engine. Several factors justified the selection of the project to study. First, the project chosen was a complex design that exhibited explicit decomposition of the engine into systems, and these into components. Second, the way the development team was organized around the architecture of the product facilitated the implementation of our approach. Third, the model studied was the most recent engine program to complete design and development, and almost all team members involved in the detail design development phase were still accessible. Finally, the engine studied was part of a family of large commercial engines with two new derivatives planned whose development programs had the potential to gain directly from this analysis. For more details about the project description and data collection refer to Rowles (1999).
2.1. Capturing the Product Architecture

The engine analyzed was decomposed into eight systems (see Figure 2). Each of these systems was further decomposed into five to ten components each. Six out of the eight systems (the fan, the low-pressure compressor, the high-pressure compressor, the burner/diffuser, the high-pressure turbine, and the low-pressure turbine) exhibited characteristics of a modular architecture in which the interfaces between their components were clearly defined with their adjacent components \textit{(modular systems)}. On the other hand, the components of the other two systems (the mechanical components system and the externals and controls system) were physically distributed throughout the engine exhibiting characteristics of an integral architecture \textit{(integrative systems)}. Components such as the main shaft and the external tubes are examples of these types of distributed components within the integrative systems. In total, the engine was decomposed into 54 components grouped into these eight systems (Sosa et al. (2000a) provide details of the analysis supporting this categorization into modular and integrative systems).

![Figure 2. Eight Systems of a Large Commercial Aircraft Engine](image)

After documenting the general decomposition of the product, we proceeded to identify the interfaces between the 54 components of the engine. Researchers in engineering design (Suh 1990, Pahl and Beitz 1991) have modeled functional requirements of product design in terms of exchanges of energy, materials, and signals between elements. Based on a method proposed by Pimmler and Eppinger (1994), we distinguished five types of design dependencies to capture the design interfaces between the physical components:

- **Spatial** dependency indicates a functional requirement related to physical adjacency for alignment, orientation, serviceability, assembly, or weight.
- **Structural** dependency indicates a functional requirement related to transferring loads, or containment.
• **Energy** dependency indicates a functional requirement related to transferring heat energy, vibration energy, electric energy, or noise.

• **Material** dependency indicates a functional requirement related to transferring airflow, oil, fuel, or water.

• **Information** dependency indicates a functional requirement related to transferring signals or controls.

After design interfaces were identified, we captured the level of criticality of each dependency for the overall functionality of the component in question. Using the five-point scale used by Pimmler and Eppinger (1994) we capture the level of criticality as:

- **Required (+2):** Interface is necessary for functionality.
- **Desired (+1):** Interface is beneficial, but not absolutely necessary for functionality.
- **Indifferent (0):** Interface does not affect functionality.
- **Undesired (-1):** Interface causes negative effects, but does not prevent functionality.
- **Detrimental (-2):** Interface must be prevented to achieve functionality.

We mapped the design-interface data into a square (54x54) design interface matrix. (The design interface matrix can be described as a special form of design structure matrix (DSM). For a formal introduction to DSM refer to Steward (1981) or Eppinger et al. (1994)) The identically labeled rows and columns name the 54 components of the engine, and their sequencing follows the front-to-back physical arrangement of the systems within the engine. Each off-diagonal cell of the matrix contains a vector of five values representing the degree of criticality of the five types of design dependency for a single design interface. Hence,
$A_{54,54} = \text{Design Interface Matrix}$

$$a_{ij} = \begin{bmatrix}
  c_{ij}^{\text{spatial}} \\
  c_{ij}^{\text{structural}} \\
  c_{ij}^{\text{energy}} \\
  c_{ij}^{\text{material}} \\
  c_{ij}^{\text{information}} 
\end{bmatrix}$$

where,

$c_{ij}^{d}$ = criticality of the interface of type "d" between components "i" and "j", for overall functionality of component "i"

$c_{ij}^{d} = [-2,-1,0,+1,+2]$  
$c_{ij}^{d}$ is undefined for $i = j$

$A_{54,54} = \text{Design Interface Matrix (binary)}$

$a_{ij} = 1$ if $|a_{ij}| > 0$

$a_{ij} = 0$ if $|a_{ij}| = 0$

where $|a_{ij}| = \sum_{d} c_{ij}^{d}$

For graphical simplicity, Figure 3 shows a binary version of the design interface matrix. The off-diagonal elements of the matrix are marked with an "X" for each pair of components that shares at least one design interface (any non-zero level of criticality). Reading across a row corresponding to a particular component indicates the other components with which it has interfaces. The diagonal elements are meaningless and are shown to separate the upper and lower triangular portions of the matrix. Note that the matrix is not completely symmetric with respect to its diagonal due to the fact that each row captures the dependencies necessary for one component's functions.

The boxes along the diagonal indicate the eight system boundaries. Marks inside the boxes represent design interfaces between components of the same system, whereas marks outside the boxes indicate interfaces between components of different systems. Light boxes throughout the matrix enclose the cross-boundary design interfaces between any two systems. The first six systems in the matrix correspond to the six modular systems, while the last two systems correspond to the two integrative systems. Note that the integrative systems have design interfaces with components in every system of the engine. (For details refer to Sosa et al. (2000a).)
2.2. Capturing the Development Organization

The organization responsible for the development of the aircraft engine was divided into sixty design teams. Fifty-four of these teams were grouped into eight system-design groups mirroring the architecture of the engine described above. Each of those teams was responsible for developing one of the 54 components of the engine. The remaining six design teams were system integration teams, which had no specific hardware assigned to them and whose responsibility was to assure that the engine worked as a whole. Examples of the system integration teams are the rotordynamics team and the secondary flow team.

We capture the system-level integration efforts (both within groups and across groups) of the organization by measuring the intensity of the technical interaction between the design teams involved in the development process. This method is similar to the approach used by McCord and Eppinger (1993). To measure the intensity of each team interaction, we asked at least two key members from each design team to rate the frequency and criticality of their technical interactions with each of the other teams during the detailed design phase of the engine development project. We used a six-point scale that combines the frequency and criticality of
each interaction into a single metric. (For details refer to Sosa (2000).) The criticality component of our metric allows asymmetry in the interaction intensity of each pair of design teams. That is, interaction intensity is measured from the respondent's point of view, and we surveyed both parties of each dyad to obtain a bilateral view of each interaction.

Previous researchers (Allen 1986, Morelli et al.1995) have defined various types of technical communications in development organizations. We focused our efforts on capturing task-related interactions between design teams (coordination-type communication). We explicitly asked respondents not to report consultation-related or skill-development-related interactions (knowledge-type communication) nor motivation-related or creativity-related interactions (inspiration-type communication).

We organize the team-interaction data in a square (60x60) team interaction matrix. The identically ordered labels of the rows and columns of this matrix contain the names of each of the design teams. Each cell in the matrix contains the interaction intensity reported by each team. Hence,

\[
T_{60,60} = \text{Team Interaction Matrix}
\]

\[t_{ij} = \text{team interaction intensity \([0,5]\) reported by team "i" about its interaction with team "j".}\]

\[t_{ij} \text{ is undefined for } i = j\]

\[
T_{60,60}^b = \text{Team Interaction Matrix (binary)}
\]

\[t_{ij}^b = 1 \text{ if } t_{ij} > 0\]

\[t_{ij}^b = 0 \text{ if } t_{ij} = 0\]

Figure 4 shows a binary team interaction matrix with off-diagonal cells marked "O" to indicate each non-zero team interaction revealed. Reading across a particular row indicates with which other teams the surveyed team interacted.

The 60 design teams are organized into groups which mirror the product architecture structure. As shown in Figure 4, associated with the six modular systems are corresponding groups of design teams. Similarly, the two integrative systems have their two corresponding groups of design teams. Finally, there are six system integration teams that are not responsible for designing any specific engine's component but they are in charge of integrating all the components into a whole. The boxes along the diagonal indicate the organizational boundaries of the eight design groups. Marks inside the boxes indicate within-boundaries team interactions, which we associate to within-group system-level integration effort. On the other hand, marks
outside the boxes indicate cross-boundaries team interactions, which we associate to across-group system-level integration effort.

![Team Interaction Matrix (Binary)](image)

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

### 2.3. Comparing Product Architecture and Development Organization

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. Sosa (2000) presents an algebraic model that allows one to perform this comparison in the general case when the assignment is not one-to-one. Figure 5 shows how, by overlapping the design interface matrix with the team interaction matrix, we obtain the resultant matrix. The resultant matrix is exhibited in Figure 6.
3. Hypotheses

The resultant matrix provides the basis for the analysis completed to answer our research questions. Figure 7 exhibits the four possible outcomes for each cell of the resultant matrix. Two positions in the 2x2 matrix shown in Figure 7 represent the expected cases in which either design interfaces are matched by team interactions ("#" cell), or absence of team interactions corresponds to lack of design interfaces ("blank" cell). However, the two unexpected cases ("X" and "O" cells) are far more interesting. In the "X" cell we find the cases in which design
interfaces are not matched by team interactions. In the "O" cell we find the cases in which team interactions were not predicted by design interfaces.

<table>
<thead>
<tr>
<th>Team Interaction</th>
<th>NO</th>
<th>X</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>#</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>NO</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 7. Four Possible Values of Each Cell of the Resultant Matrix**

While we expect the majority of the cells of the resultant matrix to contain blank and "#" cells, we will focus our analysis on the unexpected cases. This paper focuses on understanding the occurrence of the two types of unexpected cases by studying the effects due to design interface strength, organizational and system boundaries, and system modularity.

### 3.1. Effect Due to Design Interface Strength

Research suggests that a greater degree of design interdependence leads to greater communication. Allen (1997) claims that the degree of interdependence between engineers' work is directly related to the probability that they engage in frequent technical communication. At the task level, Smith and Eppinger (1997) use the strength of task interdependency to identify the sets of activities requiring many design iterations to complete their work in a coordinated manner. Loch and Terwiesch (1998) use an analytical approach to suggest that communication frequency increases with the level of dependence. These results are consistent with the empirical evidence presented by Adler (1995) and the numerical approach presented by Ha and Porteus (1995). More recently, Sosa et al. (2000b) showed that communication frequency increases with the degree of interdependence, independently of the communication media used. Therefore, we expect to find empirical support for the following hypothesis:

_H1: Weak design interfaces (i.e. non-critical and few dependencies) are less likely to be matched by team interactions than are strong design interfaces (i.e. critical and multi-dependency)._
likely to occur than across organizational boundaries. People within such boundaries are subjected to organizational bonds that promote the development of a language and an identity inherent to the group. Indeed, Allen (1977) found higher probability of engineers (in R&D organizations) engaging in technical communication when they share organizational bonds. More recently, Sosa et al. (2000b) showed that higher communication frequency is found in pairs that share organizational bonds independent of the communication media used.

System boundaries are defined by the way components comprise systems. Such boundaries may impose architectural knowledge barriers which inhibit explicit identification of cross-system design interfaces by the design experts. Nevertheless, in order to develop working systems, the teams learn of their needs to interact and do so. This results in team interactions that are not predicted by the design interfaces. Hence, we should expect a higher percentage of unknown design interfaces across system boundaries.

Having described the effects of organizational and system boundaries, we expect the following hypothesis to hold true:

H2: Team interactions are less likely to correspond to design interfaces (the "#" cell of Figure 7) when these occur across (organizational/system) boundaries. More specifically,

H2a: When considering the cases with design interfaces only (the YES column of Figure 7), design interfaces across organizational boundaries are less likely to be matched by team interactions than are design interfaces within organizational boundaries.

H2b: When considering the cases with team interactions only (the YES row of Figure 7), team interactions across system boundaries are less likely to be predicted by design interfaces than are team interactions within system boundaries.

3.3. Effects due to System Modularity

Sosa et al. (2000a) define modular and integrative systems, and study the differences between handling design interfaces across only modular systems versus handling design interfaces with integrative systems. Indeed, they found empirical support to the hypothesis that the effects due to organizational/system boundaries are statistically significant different for interactions between modular systems than for interactions with integrative systems. On the
other hand, Sosa (2000) did not find empirical support to the hypothesis that the effects due to design interface strength are statistically significant different for interfaces between modular systems than for interfaces with integrative systems.

Since this paper is focused on understanding the factors that may explain the existence of the unexpected cases (the "X" and "O" cells of Figure 7), we want to explore whether system modularity has a direct effect on the way design teams handle design interfaces. Hence, we want to explore the following hypothesis:

**H3: The proportion of design interfaces and team interactions that correspond to each other (the "#" cell of Figure 7) is statistically significant different for modular systems versus integrative systems. More specifically,**

**H3a: When considering the cases with design interfaces only (the YES column of Figure 7), the proportion of design interfaces between modular systems that are matched by team interactions is statistically significant different than the proportion of design interfaces with integrative systems that are matched by team interactions.**

**H3b: When considering the cases with team interactions only (the YES row of Figure 7), the proportion of predicted team interactions between teams that design modular systems is statistically significant different than the proportion of predicted team interactions with teams that design integrative systems.**

### 4. Categorical Data Analysis

Figure 8 summarizes the binary results shown in the resultant matrix (Figure 6). As expected, the majority of the cases (90% of the cells) are the cases when known design interfaces were matched by team interactions (349 "#" cells), or the cases with no design interfaces and no reported team interactions (2219 blank cells). The unexpected cases accounted for 10% of the cells; those were the cases when known design interfaces were not matched by team interactions (8%, or 220 "X" cells), and the cases when reported teams interactions were not predicted by design interfaces (2%, or 74 "O" cells).

<table>
<thead>
<tr>
<th>Design Interfaces</th>
<th>NO (2439)</th>
<th>X (220)</th>
<th>(2219)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Team Interactions</strong></td>
<td>YES (423)</td>
<td># (349)</td>
<td>O (74)</td>
</tr>
<tr>
<td></td>
<td>YES (569)</td>
<td>NO (2293)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Design Interfaces</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 8. Overall Results**
Among the 569 design interfaces, we found that 61% of those interfaces were matched by team interactions. Among the 423 team interactions, we found that 83% of those team interactions were predicted by design interfaces. Additionally, of the 2293 cases in which no design interfaces were known, 97% did not report team interactions. Finally, of the 2439 cases in which no direct team interactions were reported, 91% did not correspond to design interfaces.

The unit of analysis used in this section is the cell of the resultant matrix. Since the resultant matrix is the combination of both the design interface matrix and the team interaction matrix we make the following assumptions regarding the randomness of the data:

- Independence of error between the matrices. Since the data documented in the design interface matrix were provided by the design experts, and the data documented in the team interaction matrix were provided by key design team members it is fair to assume that the sources of error in these two matrices are independent of each other. Hence, we assume that systematic patterns in the resultant matrix are not the result of correlation between measurement errors in the matrices.

- Independence among the cells of the team interaction matrix. We assume that the data collected in the team interaction matrix follow a Bernoulli probability distribution (statistically independent cells) with estimated constant probabilities of 0.148 (i.e. 423 team interactions out of 2862 cells in the team interaction matrix). We based this assumption on the fact that team members surveyed were part of one team only. We acknowledge that this assumption is just an approximation of reality given the strong deviation from randomness exhibited by social networks (and evidenced in Figure 4).

- Independence among the cells of the design interface matrix. Similarly to the previous assumption, we assume that the data collected in the design interface matrix follow a Bernoulli probability distribution (statistically independent cells) with estimated constant probabilities of 0.199. Even though, the data collection was completed independently for each component of the engine, we expect to encounter the same types of deviation from randomness presented in social networks.

In the next section we present a log-linear model based on techniques used to analyze social networks in order to relax the last two assumptions described above, and therefore validate the results presented in this section.
Before testing the hypotheses posed in section 3, we first test the nominal null hypothesis that "a team interaction is independent of whether there is a design interface associated to it". Under this null hypothesis the probability distribution for the resultant matrix is also a Bernoulli probability distribution with an estimated constant probability that predicts the cases where a design interface is matched by a team interaction to be equal to 0.029. As expected, the $\chi^2$ obtained when testing the nominal hypothesis equaled to 1222, which is remarkably greater than the critical value of 6.635 (for one degree of freedom and $\alpha = 0.01$), therefore we strongly reject the nominal null hypothesis stated above.

4.1. Testing the Hypothesized Effects (H1, H2, and H3)

In order to illustrate how the hypothesized effects are tested using classical categorical data analysis, we describe how we test the effects due to design interface strength. We define the strength of a design interface by the number and level of criticality of the design dependencies as follows:

$$[\text{design interface strength}]_{ij} = \sum_{d=\text{dependency type}} \left| c_{ij}^d \right|$$

where,

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

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

To test hypothesis H1, we categorize the 569 design interfaces (YES column of Figure 7) according to the following two criteria:

- **First criterion**: Whether a design interface is matched by a team interaction or not.
- **Second criterion**: Whether a design interface is either weak (design interface strength $\leq 4$) or strong (design interface strength $> 4$). Since the average design interface strength is 4.4, we use 4 as the cut-off point between weak and strong design interfaces.

We display the cross-classification of the sample in a contingency table (Table 1) used to perform a chi-square test of independence. The test resulted in a $\chi^2$ of 21.385, exceeding the critical value of 6.635 (for one degree of freedom and $\alpha = 0.01$). Hence, we reject the null hypothesis that matching a design interface by a team interaction is independent of the strength of the design interface. More specifically, of the 319 weak design interfaces, 53% were matched by team interactions, whereas of the 250 strong design interfaces, 72% were matched by team
interactions. Therefore the empirical evidence supports hypothesis H1 that weak design interfaces are less likely to be matched by team interactions than strong design interfaces.

Similar chi-square tests of independence were completed to test the hypothesized effects about organizational and system boundaries (H2), and system modularity (H3). The results summarized in Table 2 show that the data support both H2a and H2b, but do not support either H3a or H3b.

Table 1. Chi-square Test of Independence. Effect of Design Interface Strength.

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th>Expected number (fraction) of design interfaces matched by team interactions</th>
<th>Expected number (fraction) of design interfaces not matched by team interactions</th>
<th>Actual number (fraction) of design interfaces matched by team interactions</th>
<th>Actual number (fraction) of design interfaces not matched by team interactions</th>
<th>( \chi^2 ) of design interfaces matched by team interactions</th>
<th>( \chi^2 ) of design interfaces not matched by team interactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weak design</td>
<td>319</td>
<td>191.176 (61.34%)</td>
<td>127.824 (38.66%)</td>
<td>169 (52.98%)</td>
<td>150 (47.02%)</td>
<td>3.633</td>
<td>5.763</td>
</tr>
<tr>
<td>interface</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(strength ≤4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strong design</td>
<td>250</td>
<td>149.824 (61.34%)</td>
<td>100.176 (38.66%)</td>
<td>180 (72.00%)</td>
<td>70 (28.00%)</td>
<td>4.635</td>
<td>7.354</td>
</tr>
<tr>
<td>interface</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(strength &gt;4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>569</td>
<td>349.000</td>
<td>220.000</td>
<td>349</td>
<td>220</td>
<td>8.268</td>
<td>13.116</td>
</tr>
</tbody>
</table>

\( H_0 \): Weak design interfaces are as likely to be matched by team interactions as strong design interfaces.

\( \chi^2 = 21.385 \)

Critical \( \chi^2_{(0.99,1)} = 6.635 \)

Since \( \chi^2 > \text{Critical } \chi^2_{(0.99,1)} \), we reject \( H_0 \).

Table 2. Results of Chi-square Tests of Independence

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Sample</th>
<th>Results</th>
<th>( \chi^2 )</th>
<th>Conclusion(^a)</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1: Effect of design</td>
<td>569 design interfaces</td>
<td>47% of the 319 strong design interfaces were matched by team interactions whereas 53% of the 250 weak design interfaces were matched by team interactions</td>
<td>21.385</td>
<td>H1 is supported</td>
</tr>
<tr>
<td>interface strength</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H2a: Effect of organizational boundaries</td>
<td>569 design interfaces</td>
<td>81% of the 231 within-boundary design interfaces were matched by team interactions whereas 48% of the 338 cross-boundary design interfaces were matched by team interactions</td>
<td>63.101</td>
<td>H2a is supported</td>
</tr>
<tr>
<td>H2b: Effect of systems</td>
<td>423 team interactions</td>
<td>90% of the 208 within-boundary team interactions were predicted by design interfaces whereas 75% of the cross-boundary team interactions were predicted by design interfaces</td>
<td>15.517</td>
<td>H2b is supported</td>
</tr>
<tr>
<td>boundaries</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H3a: Effect of system</td>
<td>569 design interfaces</td>
<td>60% of the 247 design interfaces between modular systems were matched by team interactions whereas 62% of the 322 design interfaces with integrative systems were matched by team interactions</td>
<td>0.068</td>
<td>H3a is not supported</td>
</tr>
<tr>
<td>modularity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H3b: Effect of team</td>
<td>423 team interactions</td>
<td>79% of the 189 team interactions between modular design teams were predicted by design interfaces whereas 85% of the 234 team interactions with integrative design teams were predicted by design interfaces</td>
<td>2.335</td>
<td>H3b is not supported</td>
</tr>
<tr>
<td>modularity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(^a\): The null hypothesis is rejected when \( \chi^2 \) is greater than the critical \( \chi^2_{(0.99,1)} = 6.635 \)
4.2. Combined Effects: Organizational Boundaries and Design Interface Strength

This section is focused on studying the joint effects of design interface strength (H1) and organizational boundaries (H2). We found that the portion of strong design interfaces within organizational boundaries is statistically significant greater than the portion of weak design interfaces within organizational boundaries. Similarly, the portion of weak design interfaces across organizational boundaries is statistically significant greater than the portion of strong design interfaces across organizational boundaries (for details refer to Sosa (2000)).

This result suggests that we test the null hypothesis that the effect due to organizational boundaries is homogenous throughout the data (for both weak and strong design interfaces). We also need to test the null hypothesis that the effect due to design interface strength is homogenous throughout the data (for both within-boundary and across-boundary design interfaces).

We performed chi-square tests of homogeneity whose results are summarized in Table 3. We found that for the cases within organizational boundaries, the portion of strong design interfaces matched by team interactions was statistically significant greater than the portion of weak design interfaces matched by team interactions, which is in line with hypothesis H1. However, for the cases across organizational boundaries we could not reject the null hypothesis that weak design interfaces are as likely to be matched by team interactions as strong design interfaces, which is contrary to hypothesis H1. We also found that for both weak and strong design interfaces, the likelihood that a design interface is matched by a team interaction is greater when it is within organizational boundaries.

As a result, we conclude that the effects of organizational boundaries are more severe than the effects of design interface strength. That is, we found empirical support for hypothesis H2a throughout the data (for both weak and strong design interfaces). On the other hand, the data support hypothesis H1 within organizational boundaries only, while across organizational boundaries design interface strength makes no statistically significant difference on whether or not design interfaces are matched by team interactions.
Table 3. Results of Chi-square Tests of Homogeneity

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Sample</th>
<th>Results</th>
<th>$\chi^2$</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1 &amp; H2: Effects of organizational</td>
<td>231 within-boundary design</td>
<td>87% of the 135 strong design interfaces were matched by team interactions</td>
<td>8.778</td>
<td>H1 supported within</td>
</tr>
<tr>
<td>boundaries controlling for design</td>
<td>interfaces</td>
<td>whereas 72% of the 96 weak design interfaces were matched by team interactions</td>
<td></td>
<td>organizational</td>
</tr>
<tr>
<td>interface strength</td>
<td></td>
<td></td>
<td></td>
<td>boundaries</td>
</tr>
<tr>
<td></td>
<td>338 cross-boundary design</td>
<td>54% of the 115 strong design interfaces were matched by team interactions</td>
<td>2.501</td>
<td>H1 not supported</td>
</tr>
<tr>
<td></td>
<td>interfaces</td>
<td>whereas 45% of the 223 weak design interfaces were matched by team interactions</td>
<td></td>
<td>across</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>organizational</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>boundaries</td>
</tr>
<tr>
<td>H1 &amp; H2: Effects of design interface</td>
<td>319 weak design interfaces</td>
<td>72% of the 96 within-boundary design interfaces were matched by team interactions</td>
<td>19.685</td>
<td>H2 supported for</td>
</tr>
<tr>
<td>strength controlling for organizational</td>
<td></td>
<td>whereas 45% of the 223 cross-boundary design interfaces were matched by team interactions</td>
<td></td>
<td>weak design</td>
</tr>
<tr>
<td>boundaries</td>
<td></td>
<td></td>
<td></td>
<td>interfaces</td>
</tr>
<tr>
<td></td>
<td>250 strong design interfaces</td>
<td>87% of the 135 within-boundary design interfaces were matched by team interactions</td>
<td>34.558</td>
<td>H2 supported for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>whereas 54% of the 115 cross-boundary design interfaces were matched by team interactions</td>
<td></td>
<td>strong design</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>interfaces</td>
</tr>
</tbody>
</table>

a: The null hypothesis is rejected when $\chi^2$ is greater than the critical $\chi^2_{(0.05),6} = 6.635$

5. Log-linear Analysis

Research in social science has shown that social network data (such as those documented in the team interaction matrix) possess strong deviation from randomness. More specifically, previous research (Holland and Leinhardt 1981) shows that social networks exhibit several types of dependence such as tendency toward reciprocation, tendency toward expansiveness (i.e. to generate interactions) and tendency toward attraction (i.e. to attract interactions). Additionally, the design interface matrix (Figure 3) and the team interaction matrix (Figure 4) suggest the presence of "within-system" and "within-group" effects, respectively.

In this section we build upon statistical techniques used in social network analysis to develop a log-linear model that allows us to test the effects of organizational/system boundaries and the effects of system modularity while controlling for reciprocation, differential expansiveness, differential attraction, and within-system and within-group tendencies. This model is a dyadic interaction model, which uses the natural log of probabilities as the basic modeling unit. Specifically, we estimate a model of the form:

$$\ln \left[ P(\text{component } i \text{ depends on component } j \text{ and team } i \text{ reports interaction with team } j) \right] = F(\text{overall mean, tendency of component } i \text{ to generate design interfaces to other components, tendency of component } j \text{ to depend upon other components, overall tendency to reciprocate design interfaces, tendency of team } i \text{ to report interaction with})$$
other teams, tendency of other teams to report interaction with team \( j \), overall tendency to reciprocate team interactions, overall association between design interfaces and team interactions, effect due to system and organizational boundaries, effect due to systems modularity)

The objective of this analysis is to relax the assumptions made in the previous section regarding the independence of the cells of both the design interface matrix and the team interaction matrix. Our model is based on the \( p_1 \) distribution introduced by Holland and Leinhardt (1981).

In order to introduce the \( p_1 \) distribution, we consider the four-dimensional \( Y \)-array whose component \( Y_{ijkl} \) describes the interaction between element \( i \) and element \( j \). The third and fourth dimensions of the \( Y \)-array are binary. Hence, \( k=1 \) if element \( i \) interacts to element \( j \), and \( l=1 \) if element \( j \) interacts to element \( i \). This model specifies the probability distribution that a pair of elements (either a pair of physical components or a pair of design teams) has one of four possible dyadic relationships: mutual silence (\( Y_{i00} \)), mutual interaction (\( Y_{i11} \)), asymmetric interaction (either \( Y_{j01} \) or \( Y_{j10} \)). To determine the probability distribution of the network, the dyads are assumed to be conditionally independent, so that we multiply the dyad probability distributions to obtain their joint probability distribution.

Fienberg and Wasserman (1981) show that Holland and Leinhardt's distribution, \( p_1 \), can be expressed as follows:

\[
\ln P\{Y_{i00} = 1\} = \lambda_y
\]
\[
\ln P\{Y_{i01} = 1\} = \lambda_y + \theta + \alpha_i + \beta_j
\]
\[
\ln P\{Y_{j01} = 1\} = \lambda_y + \theta + \alpha_j + \beta_i
\]
\[
\ln P\{Y_{i11} = 1\} = \lambda_y + 2\theta + \alpha_i + \beta_i + \alpha_j + \beta_j + \rho
\]
or in shorthand,

\[
\ln P\{Y_{i00} = 1\} = \lambda_y + (k+l)\theta + k \cdot \alpha_i + l \cdot \beta_j + k \cdot \beta_j + (kl)\rho \tag{1}
\]

The parameters \( \{\alpha_i\} \) measure the expansiveness or "productivity" of the elements of the network, indicating how likely an element is to generate relational ties (non-zero cells in row \( i \) of the matrices). The parameters \( \{\beta_j\} \) measure the attraction or "popularity" of the elements of the network, indicating how likely an element is to receive relational ties (non-zero cells in column \( j \) of the matrices). The "reciprocity" parameter, \( \rho \), measures the overall tendency in the network to reciprocate interactions. The \( \theta \) parameter indicates the overall volume of interaction in the
network. Finally, the $\lambda_{ij}$ parameters are "dyadic" effects that ensure that the probabilities sum to one for each dyad (equation 1), they have no substantive meaning. For a more detailed description of these parameters refer to Holland and Leinhardt (1981).

Our approach is similar to the statistical modeling technique used by Van den Bulte and Moenaert (1998) to analyze interactions between R&D teams before and after collocation. Likewise, we complete our log-linear analysis in five steps:

1. Extend the $p_I$ model to a network with two relations (design interfaces and team interactions).
2. Aggregate physical components and design teams into groups.
3. Extend the model with association parameters that capture the underlying tendency of correspondence between design interfaces and team interactions.
4. Extend the model with structural parameters to capture the hypothesized effects of organizational/system boundaries and system modularity.
5. Estimate parameters, compute test statistics, and test the hypotheses.

**Step 1: A $p_I$ Model for Two Relations**

Fienberg et al (1985) first addressed the problem of extending $p_I$ to multiple sociometric relations. Wasserman and Iacobucci (1988) used their results as the basis to study sequential network data, and Van den Bulte and Moenaert (1998) used these models to analyze the interactions between R&D teams in two points in time. Based upon these results we develop a base log-linear model of the resultant matrix. We consider the joint distribution of both design interfaces and team interactions for a given dyad. That is, each dyad $(ij)$ of the resultant matrix consisting of elements $i$ and $j$ has 16 states. Four ($2 \times 2$) states are associated to the elements' design interface relation, and four ($2 \times 2$) states are associated to their team interaction relation, resulting in 16 states for each dyad. We assign the subscripts $(k_1, l_1)$ to describe the four states associated to the design interface relation, while the subscripts $(k_2, l_2)$ refer to the four states associated to the team interaction relation of dyad $(ij)$. The redefined Y-array has now six dimensions $54 \times 54 \times (2 \times 2) \times (2 \times 2)$, and its characteristic element can be defined as follows:

$Y_{ij k_1 l_1 k_2 l_2} = 1$ if dyad $(ij)$ behaves as described by $(k_1, l_1)$ for their design interfaces and by $(k_2, l_2)$ for their team interactions.

$Y_{ij k_1 l_1 k_2 l_2} = 0$ otherwise.
Considering the joint distribution of design interfaces and team interactions yields a log-linear model which describes simultaneously the behavior of the elements of our network according to two independent relations (design interfaces and team interactions). Hence, the base log-linear model can be written as follows:

$$\ln P\{Y_{ijkl} = 1\} = \lambda_{ij} + (k_1 + l_1)\theta_i + k_1\alpha_{ij} + l_1\beta_{ij} + l_1\alpha_{ij} + k_1\beta_{ij} + (k_1l_1)p_1 + (k_2 + l_2)\theta_2 + k_2\alpha_{2i} + l_2\beta_{2i} + l_2\alpha_{2i} + k_2\beta_{2i} + (k_2l_2)p_2$$

The parameters on this model have the same meaning as in the original \( p \) model, but applied to either design interfaces (subscript 1) or team interactions (subscript 2).

Step 2: Aggregate components and teams into groups

Fienberg and Wasserman (1981) introduced the approach of placing actors into subsets using relevant actor characteristics such that actors within a subset are assumed to behave similarly. This assumption of comparable behavior of elements within subsets has been termed stochastic equivalence (Wasserman and Weaver 1985). Assuming that elements \( i \) and \( j \) are stochastic equivalent means, in mathematical terms, that:

$$\alpha_i = \alpha_j \quad \text{and} \quad \beta_i = \beta_j$$

We operationalize the concept of stochastic equivalence by aggregating the 54 elements of the \( Y \)-array into 8 subsets according to the system boundaries of the product and the organizational boundaries of the development organization, respectively. By doing so, we obtain a much smaller \( W \)-array whose dimensions are \( 8 \times 8 \times (2 \times 2) \times (2 \times 2) \), with elements \( \{ W_{rs kl,11} \} \) to be equal to the number of dyads between groups \( r (G_r) \) and \( s (G_s) \) whose design interfaces are described by \( (k_1, l_1) \), and whose team interactions are described by \( (k_2, l_2) \). Hence,

$$w_{rs kl,11} = \sum_{i \in G_r} \sum_{j \in G_s} y_{ijkl}$$

Therefore, we can rewrite the base model specified in equation (2) as follows:

$$\ln E (W_{rs kl,11}) = \lambda_{rs} + (k_1 + l_1)\theta_1 + k_1\alpha_{1r} + l_1\beta_{1r} + l_1\alpha_{1r} + k_1\beta_{1r} + (k_1l_1)p_1 + (k_2 + l_2)\theta_2 + k_2\alpha_{2r} + l_2\beta_{2r} + l_2\alpha_{2r} + k_2\beta_{2r} + (k_2l_2)p_2$$

Step 3: Extend the base model with association parameters

The base model specified in equation (4) assumes that design interfaces and team interactions are two independent relations of the same network of elements. We consider second-
order interaction effects between design interfaces and team relations to capture the association between the design interface matrix and the team interaction matrix. We adapt the description of these effects provided by Wasserman and Iacobucci (1988) to our context as follows:

\( \theta_{12} \) = parameter measuring tendency toward conformity across relationships. That is, component \( i \) depends on component \( j \), AND team \( i \) reports interaction with team \( j \).

\( \rho_{12} \) = parameter measuring tendency toward flow reversal. That is, component \( i \) depends upon component \( j \), AND team \( j \) reports interaction with team \( i \).

Since the \( \theta_{12} \) parameter reflects the overall tendency toward positively associated design interfaces and team interactions (the "#" and the "blank" cells of Figure 7), we expect this parameter to be significantly positive. On the other hand, the \( \rho_{12} \) parameter reflects the overall tendency toward flow reversal, that is, how likely it is that component \( i \) depending on component \( j \), influences team \( j \) to interact with team \( i \). Given the relatively small number of "X" and "O" in the resultant matrix (Figure 6), we do not expect \( \rho_{12} \) to be significantly different than zero.

After extending the model with the second-order interaction parameters described above, the base model can be written as follows:

\[
\ln E(W_{rs,k_1,k_2}) = \lambda_{rs} + (k_1 + l_1)\beta_1 + l_1\alpha_{1r} + l_2\beta_{1r} + l_1\alpha_{1s} + k_1\beta_{1s} + (k_1l_1)\rho_1 + (k_2 + l_2)\beta_2 + k_2\alpha_{2r} + l_2\beta_{2r} + l_2\alpha_{2s} + k_2\beta_{2s} + (k_2l_2)\rho_2 + \theta_{12} + \rho_{12} \tag{5}
\]

**Step 4: Extend the model with structural parameters**

To explicitly represent organizational and system boundary effects, we define the following indicator variable:

ACROSS = 1 if elements (i.e. component and team) \( i \) and \( j \) are in the different groups \((r \neq s)\)

ACROSS = 0 if \( r = s \)

By expanding the dimension of the W-array with ACROSS as the seventh dimension, we can estimate the parameter associated to the second-order interaction terms ACROSS x \( k_1 \), and ACROSS x \( k_2 \), due to symmetry of the W-array identical to ACROSS x \( l_1 \) and ACROSS x \( l_2 \), respectively. These terms capture the within-system and within-group effects exhibited in both the design interface matrix and team interaction matrix. Indeed, we expect these terms to be significantly negative indicating that it is less likely to encounter design interfaces across system boundaries and team interactions across organizational boundaries.

We define another indicator variable to include the effects due to system modularity into the model. Hence,
MODULAR = 1 if both components of a dyad belong to modular systems (r < 7 and s < 7).
MODULAR = 0 if one of the components of a dyad belongs to integrative systems (r ≥ 7 or s ≥ 7).

Since the interaction term \( k_1 \times k_2 = l_1 \times l_2 \) captures whether or not design interfaces are matched by team interactions, the third-order interaction effect that defines \( \theta_{ACROSS,k1,k2} (= \theta_{ACROSS,l1,l2}) \) captures whether the occurrence of dyads across boundaries with design interfaces matched by team interactions is significantly less than the occurrence of dyads within boundaries with design interfaces matched by team interactions. Hence, a formal hypothesis testing of H2 can be specified as follows:

\[
H2: \theta_{ACROSS,k1,k2} < 0
\]

Similar rationale is followed to define the parameter associated to the third-order interaction term MODULAR \( x k_1 \times k_2 \) (due to symmetry of the W-array identical to MODULAR \( x l_1 \times l_2 \)). Hence, a formal hypothesis testing of H3 can be specified as follows:

\[
H3: \theta_{MODULAR,k1,k2} \neq 0
\]

Finally, we estimate the parameter associated to the fourth-order interaction effect MODULAR \( x ACROSS \times k_1 \times k_2 \) (due to symmetry of the W-array identical to MODULAR \( x ACROSS \times l_1 \times l_2 \)). \( \theta_{MODULAR,ACROSS,k1,k2} (= \theta_{MODULAR,ACROSS,l1,l2}) \) captures whether the effect due to organizational/system boundary is significantly different for modular systems than for integrative systems. We expect this fourth-order interaction effect to be statistically significant smaller than zero, which corresponds with fewer cross-boundary design interfaces (matched by team interactions) between modular systems than with integrative systems (in line with the results presented by Sosa et al. (2000a)).

**Step 5: Fitting the model to data, computing test statistics, and testing hypotheses**

We use standard iterative proportional fitting computer programs for contingency tables (we used SPSS) to fit the model described by (5) to data. It is important to mention that the \( G^2 \) statistic obtained from commercial statistical software applications is incorrect, and the correct value has to be calculated using the Y-array. The reason for this is that the unit of analysis is still the dyad rather than the group of dyads (for details see of Fienberg and Wasserman (1981), p. 181).

As described in step 4, ACROSS and MODULAR expanded the dimensions of the W-array, but they are just indicator variables and do not increase the number of states of the dyad. They are completely defined by the independent states \( r \) and \( s \), hence we define structural zeros
(when using SPSS) for the dyads where ACROSS = 1 and \( r = s \), and for the dyads ACROSS = 0 and \( r \neq s \). Similarly, we define structural zeros for the dyads where MODULAR = 1 and \( r \geq 7 \) or \( s \geq 7 \), and for the cases where MODULAR = 0 and \( r < 7 \) and \( s < 7 \).

Table 5 shows the estimates of the parameters for five log-linear models with their respective likelihood-statistic \( G^2 \) and the number of degrees of freedom. The first model (independent) does not include the association parameters between design interfaces and team interactions. The second model (base) includes \( \theta_{12} \) which substantially improve the goodness-of-fit of the independent model (\( \Delta G^2 = 943.66, \Delta df = 1 \)). Including \( \rho_{12} \) did not significantly improve the model fit and therefore it is excluded from the base model. However, including the second-order interaction effects with ACROSS greatly improves the goodness of fit of the base model (\( G^2 = 3068.3, df = 5689 \)). The inclusion of these effects resulted in statistically significant negative parameters indicating, as expected, that smaller portion of design interfaces and smaller portion of team interactions take place across boundaries (see Model 3). Model 3 and Model 4 include the third-order interaction parameters that test hypothesis H2 and H3, respectively. Model 3 includes a statistically significant negative \( \theta_{ACROSS,1,k2} (= \theta_{ACROSS,11,12} \) parameter indicating that design interfaces matched by team interactions are less likely to take place across boundaries (supporting H2). When adding second-order and third-order interaction effects with MODULAR the log-linear model does not significantly improve its goodness-of-fit (see Model 4), resulting in insignificant parameters. Therefore, we could not reject the null hypothesis that \( \theta_{MODULAR,k1,k2} (= \theta_{MODULAR,11,12} \) is zero, which correspond with the results obtained in the previous section (and contrary to hypothesis H3). Finally, model 5 includes the fourth-order interaction parameter \( \theta_{MODULAR,ACROSS,k1,k2} (= \theta_{MODULAR,ACROSS,11,12} \), which resulted to be statistically significant negative confirming the results reported by Sosa et al. (2000a) about how cross-boundary design interfaces matched by team interactions are less likely to occur between modular systems.
Table 5. Results of Fitting Base Model to Data

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Model 1 (Independent)</th>
<th>Model 2 (Base)</th>
<th>Model 3 (ACROSS)</th>
<th>Model 4 (MODULAR)</th>
<th>Model 5 (FINAL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters for the design interface matrix</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \beta_{IFAN} )</td>
<td>0.4321</td>
<td>0.3647</td>
<td>0.3772</td>
<td>0.2630</td>
<td>0.3152</td>
</tr>
<tr>
<td>( \beta_{LPC} )</td>
<td>0.2090</td>
<td>0.3687</td>
<td>0.3244</td>
<td>0.2679</td>
<td>0.2660</td>
</tr>
<tr>
<td>( \beta_{HPC} )</td>
<td>-0.0119</td>
<td>0.1134</td>
<td>0.0634</td>
<td>0.0202</td>
<td>0.0010</td>
</tr>
<tr>
<td>( \beta_{BD} )</td>
<td>-0.0171</td>
<td>-0.2261</td>
<td>-0.1600</td>
<td>-0.3054</td>
<td>-0.2240</td>
</tr>
<tr>
<td>( \beta_{HPT} )</td>
<td>-0.5652</td>
<td>-0.5774</td>
<td>-0.5480</td>
<td>-0.6449</td>
<td>-0.6096</td>
</tr>
<tr>
<td>( \beta_{LPT} )</td>
<td>-0.0770</td>
<td>-0.0474</td>
<td>-0.0524</td>
<td>-0.1373</td>
<td>-0.1190</td>
</tr>
<tr>
<td>( \beta_{MC} )</td>
<td>-0.2566</td>
<td>-0.1597</td>
<td>-0.1756</td>
<td>0.0709</td>
<td>0.0080</td>
</tr>
<tr>
<td>( \beta_{EC} )</td>
<td>0.2869</td>
<td>0.1638</td>
<td>0.1710</td>
<td>0.4655</td>
<td>0.3627</td>
</tr>
<tr>
<td>( \beta_{IFAN} )</td>
<td>-0.7415</td>
<td>-0.6637</td>
<td>-0.7052</td>
<td>-0.7554</td>
<td>-0.7959</td>
</tr>
<tr>
<td>( \beta_{LPC} )</td>
<td>-0.0671</td>
<td>0.1406</td>
<td>0.0970</td>
<td>0.0335</td>
<td>0.0371</td>
</tr>
<tr>
<td>( \beta_{HPC} )</td>
<td>0.0509</td>
<td>0.2175</td>
<td>0.1838</td>
<td>0.1068</td>
<td>0.1184</td>
</tr>
<tr>
<td>( \beta_{BD} )</td>
<td>-0.0956</td>
<td>-0.4275</td>
<td>-0.3586</td>
<td>-0.5252</td>
<td>-0.4031</td>
</tr>
<tr>
<td>( \beta_{HPT} )</td>
<td>0.4363</td>
<td>0.3355</td>
<td>0.3748</td>
<td>0.2171</td>
<td>0.3232</td>
</tr>
<tr>
<td>( \beta_{LPT} )</td>
<td>-0.3861</td>
<td>-0.2278</td>
<td>-0.2301</td>
<td>-0.3330</td>
<td>-0.3088</td>
</tr>
<tr>
<td>( \beta_{MC} )</td>
<td>0.3176</td>
<td>0.1866</td>
<td>0.1819</td>
<td>0.4910</td>
<td>0.3682</td>
</tr>
<tr>
<td>( \beta_{EC} )</td>
<td>0.4856</td>
<td>0.4384</td>
<td>0.4565</td>
<td>0.7650</td>
<td>0.6611</td>
</tr>
<tr>
<td>( \beta_1 )</td>
<td>-1.0653</td>
<td>0.1633</td>
<td>0.1608</td>
<td>0.6128</td>
<td>0.2875</td>
</tr>
<tr>
<td>( \rho_1 )</td>
<td>3.9891</td>
<td>3.3992</td>
<td>3.2644</td>
<td>3.3657</td>
<td>3.2504</td>
</tr>
</tbody>
</table>

Parameters for the team interaction matrix

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Model 1 (Independent)</th>
<th>Model 2 (Base)</th>
<th>Model 3 (ACROSS)</th>
<th>Model 4 (MODULAR)</th>
<th>Model 5 (FINAL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters for the design interface matrix</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \beta_{IFAN} )</td>
<td>0.2778</td>
<td>0.24615</td>
<td>0.2534</td>
<td>0.0947</td>
<td>0.1667</td>
</tr>
<tr>
<td>( \beta_{LPC} )</td>
<td>0.0061</td>
<td>-0.2359</td>
<td>-0.2124</td>
<td>-0.3694</td>
<td>-0.3228</td>
</tr>
<tr>
<td>( \beta_{HPC} )</td>
<td>-0.0313</td>
<td>-0.1505</td>
<td>-0.1421</td>
<td>-0.2860</td>
<td>-0.2594</td>
</tr>
<tr>
<td>( \beta_{BD} )</td>
<td>0.0008</td>
<td>0.2441</td>
<td>0.3118</td>
<td>0.1040</td>
<td>0.2735</td>
</tr>
<tr>
<td>( \beta_{HPT} )</td>
<td>-0.3079</td>
<td>-0.0763</td>
<td>-0.0386</td>
<td>-0.2085</td>
<td>-0.1246</td>
</tr>
<tr>
<td>( \beta_{LPT} )</td>
<td>-0.0197</td>
<td>0.0719</td>
<td>0.1062</td>
<td>-0.0779</td>
<td>0.0039</td>
</tr>
<tr>
<td>( \beta_{MC} )</td>
<td>-0.3880</td>
<td>-0.3505</td>
<td>-0.4727</td>
<td>0.0235</td>
<td>-0.1839</td>
</tr>
<tr>
<td>( \beta_{EC} )</td>
<td>0.4619</td>
<td>0.2511</td>
<td>0.1943</td>
<td>0.7199</td>
<td>0.4467</td>
</tr>
<tr>
<td>( \beta_{IFAN} )</td>
<td>-0.5182</td>
<td>-0.2435</td>
<td>-0.3679</td>
<td>-0.4126</td>
<td>-0.5145</td>
</tr>
<tr>
<td>( \beta_{LPC} )</td>
<td>-0.1838</td>
<td>-0.3645</td>
<td>-0.3571</td>
<td>-0.5355</td>
<td>-0.4639</td>
</tr>
<tr>
<td>( \beta_{HPC} )</td>
<td>-0.1618</td>
<td>-0.3141</td>
<td>-0.3214</td>
<td>-0.4867</td>
<td>-0.4242</td>
</tr>
<tr>
<td>( \beta_{BD} )</td>
<td>0.2766</td>
<td>0.5727</td>
<td>0.6643</td>
<td>0.3838</td>
<td>0.6278</td>
</tr>
<tr>
<td>( \beta_{HPT} )</td>
<td>0.3068</td>
<td>0.2731</td>
<td>0.3596</td>
<td>0.0838</td>
<td>0.3201</td>
</tr>
<tr>
<td>( \beta_{LPT} )</td>
<td>-0.5070</td>
<td>-0.3641</td>
<td>-0.4016</td>
<td>-0.3369</td>
<td>-0.5236</td>
</tr>
<tr>
<td>( \beta_{MC} )</td>
<td>0.4624</td>
<td>0.4074</td>
<td>0.4579</td>
<td>0.9309</td>
<td>0.6942</td>
</tr>
<tr>
<td>( \beta_{EC} )</td>
<td>0.3248</td>
<td>0.0327</td>
<td>-0.0341</td>
<td>0.5730</td>
<td>0.2841</td>
</tr>
<tr>
<td>( \theta_1 )</td>
<td>-1.0619</td>
<td>-0.1969</td>
<td>-0.3419</td>
<td>0.5552</td>
<td>-0.0984</td>
</tr>
<tr>
<td>( \rho_1 )</td>
<td>3.5191</td>
<td>2.3742</td>
<td>1.9946</td>
<td>2.2971</td>
<td>1.9442</td>
</tr>
</tbody>
</table>

Second-order association parameter

| \( \theta_2 \)     | 3.1070*               | 3.1120         | 3.0876           | 2.7775            |                 |

Third-order interaction parameters

| \( \theta_{ACROSS112} \) | -0.1595               | -0.5517        |                 |                 |
| \( \theta_{ACROSS112} \) | -1.0191               | -1.4690        |                 |                 |

Fourth-order interaction parameter

| \( \theta_{MODULARACROSS112} \) | -0.9450*              | -0.9450*       | -0.0544*         | 0.4768           |
| \( \theta_{MODULARACROSS112} \) | -0.3354*              |                 |                 |

Goodness-of-fit

| \( G^2 \) | 5242.96               | 4299.30         | 3061.56          | 3468.25          | 3030.81         |
| \( df \)  | 5692                  | 5691            | 5688             | 5688             | 5684            |

a: The unconstrained model against which significance was assessed is model 1. Hence, \( \Delta G^2 = 943.66, \Delta df = 1, p < .001 \). Model 2 does not significantly improve when \( \rho_1 \) is added to the model \( (G^2=4298.64, df=5690) \).
b: The unconstrained model against which the hypothesis \( (H2) \) is tested includes the second-order parameter with ACROSS \( (G^2=3068.3, df=5689) \). Hence, \( \Delta G^2 = 6.74, \Delta df = 1, p < .01 \).
c: The unconstrained model against which the hypothesis \( (H3) \) is tested includes the second-order parameters with MODULAR \( (G^2=3468.3, df=5689) \). Hence, \( \Delta G^2 = 0.13, \Delta df = 1, p > .1 \).
d: The unconstrained model against which the significance was assessed includes second and third order interaction terms with both ACROSS and MODULAR \( (\Delta G^2 = 29.95, \Delta df = 1, p < .001) \).
6. Discussion and Conclusions

The research method presented in this paper provides a useful approach to investigate the coupling of the product architecture and the development organization. Our approach involves three steps. 1) capture the product architecture by documenting design interfaces, 2) capture the integration effort of the development organization by documenting team interactions, and 3) couple the product architecture with the development organization by comparing design interfaces with team interactions. This method is particularly applicable to projects where the architecture of the product is well understood and the development team is organized around the product architecture.

The usefulness of our approach is evidenced by the fact that it allows one to study both the association and the mismatch between design interfaces and team interactions. The analyses presented in this paper have focused on explaining the mismatch between design interfaces and team interactions. We have contributed to an understanding of what drives technical communication in product development organizations by formulating and testing several hypotheses to explain the cases when: 1) known design interfaces were not matched by team interactions, and 2) observed team interactions were not predicted by design interfaces. More specifically, our analyses provide the following important results:

1. There is a remarkably strong association between design interfaces and team interactions. System-level integration efforts, reflected by coordination-type communications between design teams, are driven by the architecture of the product to be designed. Indeed, 83% of the coordination-type communications were predicted by design interfaces.

2. The probability that a design interface does not correspond to a team interaction depends on several factors. In this paper we present empirical evidence showing that part of the mismatch between design interfaces and team interactions may be due to the existence of various levels of criticality and multiple dependencies of the design interfaces (design interface strength), and the existence of communication barriers associated with organizational and system boundaries. Additionally, Sosa (2000) formulates and tests several other hypothesized effects such as, design interface type, design interface redesign, indirect team interactions, secondary design interfaces, which add further insight to comprehend the mismatch between design interfaces and team interactions.
3. When considering the joint effects of organizational boundaries and design interface strength, we found that the barriers to communication imposed by organizational boundaries are more severe than the barriers to communication associated with weak design interfaces.

4. We distinguish two types of system architectures — modular and integrative systems. (Refer to Sosa et al. (2000a) for details.) We found that a mismatch between design interfaces and team interactions is equally likely to occur between modular systems as with integrative systems. However, our log-linear analysis confirms the results reported by Sosa et al. (2000a) about how the effects of organizational and system boundaries are more severe between modular systems than with integrative systems.

6.1. Managerial Implications

These results suggest that managers may be able to better use understanding of product architecture to design organizational structures effectively, which facilitate coordination-type communications and thus improve the product integration process. This further suggests that managers may be able to improve product development performance by effectively selecting team members to deal with specific critical design interfaces and by outlining organizational boundaries to foster critical technical team interactions. It is important to understand that greater effort is needed to identify and handle cross-boundary design interfaces due to the effects of system and organizational boundaries.

While the effects of organizational boundaries partially explained the large proportion of design interfaces not matched by team interactions, the effects of system boundaries were highlighted by the existence of team interactions that were not predicted by design interfaces. Such empirical evidence provided great benefits to the organization where our approach was implemented. In particular, the development organization responsible for the design of the next engine model assigned a design team that would handle those critical cross-boundary design interfaces that had not been recognized before by the design experts.

From a product innovation viewpoint, the project we studied is a mix of modular and incremental innovation. However, the lessons learned through this study may help development organizations to address architectural innovation. By documenting the architecture of the product in a design interface matrix for every generation of product family, novel architectures can be
quickly identified. Furthermore, by documenting the interactions between the design teams (team interaction matrix) to compare them with the potential interactions provided by the design interface matrix provides a systematic way to evaluate how development organizations manage architectural knowledge, a critical issue for firms facing architectural innovation (Henderson and Clark 1990).

6.2. Limitations of the Study

By studying the coupling of the architecture of an aircraft engine and the development organization that designed it we have gained important insights about how the architecture of a complex product drives the technical interactions of its development organization. While we cannot claim the generality of our findings before completing similar studies in other types of products in different industries, we would expect to obtain analogous results in other projects developing complex systems and where the development teams are organized according to the product architecture.

Even though the one-to-one mapping between the product architecture and the organizational structure greatly facilitates the implementation of our approach, it hinders separation of the effects of organizational boundaries and system boundaries. Future studies of organizations that do not mirror the architecture of the product may help address this limitation.

From an analysis standpoint, we first test the hypothesized effects by assuming independence between cells on each of the matrices to complete a categorical data analysis. Subsequently, we relaxed the independent-cells assumption by developing a dyadic interaction log-linear model. This model specifies the probability distribution of a network by assuming independent dyads. We then multiply the dyad probability distributions to obtain their joint distribution. The independent-dyad assumption is merely an approximation to reality since the model cannot control for effects other than those already implied by tendencies toward reciprocation, differential attraction, and differential expansiveness. Future research might take advantage of more advanced models that better handle dyadic dependence issues (Wasserman and Pattison 1996).

Although we measure both design interface strength and team interaction intensity as multi-point discrete variables, we dichotomized our data to simplify both categorical data analysis and log-linear analysis, and to filter the arbitrariness associated with the scale used to collect the data.
Such loss of data richness might be avoided in the future by using log-multiplicative models (Anderson and Wasserman 1995).

6.3. Research Implications

This paper opens a new stream of research on the interface of product architecture and development organization. A challenge for future research work is to extend this method to explore the evolution over time of both design interfaces and team interactions for several generations in a product family. We expect the massive use of electronic-based communication media will improve the efficiency and effectiveness of the process of documenting team interactions over time.

This study is based on the assumption of a direct mapping of product architecture and development organization. What if this were not the case? Which types of barriers are more severe (organizational or system barriers)? Is an organizational design that mirrors the architecture of the product a good one? Extending this method to study various mappings of product architectures and development organizations would be a challenge (and opportunity) for future research efforts.

7. Acknowledgements

Funding of this research has been provided by the MIT Center for Innovation in Product Development. We appreciate the assistance of the engineers from Pratt & Whitney Aircraft during the development of the example presented in this paper. We are grateful to Christophe Van den Bulte for insights related to the log-linear analysis.

8. References


