Integrative Mechanisms for Multiteam Integration: Findings from Five Case Studies*

TYSON R. BROWNING

Massachusetts Institute of Technology, 77 *Massachusetts Avenue*, *Room* 33-407, *Cambridge*, MA 02139

Received January 16, 1998; accepted May 21, 1998

ABSTRACT

Many product development programs contain multiple *integrated product teams* (IPTs) and functional support groups. Interteam information dependencies greatly affect program success. Organization integration has thus become an issue of increasing interest. This paper focuses on the realm of team interdependence and categorizes and explores several *integrative mechanisms* (IMs) that facilitate interteam integration. IMs are strategies and tools for effectively coordinating actions across groups within a program. The IM categorization scheme should prove useful to those developing an integration "tool kit." This paper explores the use of IMs in real programs, summarizing findings from five case studies at Chrysler, General Electric Aircraft Engines, Boeing, Sundstrand, and Raytheon Systems. As the appropriateness of a given IM varies as a function of many parameters—such as program stage, size, complexity, risk, etc.—this research does not formulate a universal template for IM application. Rather, the hope is that the lessons learned by these five programs will help others determine the suitability of particular IMs in their situations. This paper centers on studies in the defense aerospace industry (with two commercial cases and one nonaerospace case), but the implications extend to any system development program. ©1998 John Wiley & Sons, Inc. Syst Eng 1: 95–112, 1998

1. INTRODUCTION

Many contemporary product development programs consist of multiple *integrated product teams* (IPTs) and functional support groups. Efforts towards concurrent engineering or *integrated product development* (IPD) and accompanying new forms of program organization have intensified issues related to interteam information dependencies. Program integration—a level above IPT

Direct and indirect funding for this work has been provided through the Massachusetts Institute of Technology's Lean Aerospace Initiative, sponsored jointly by the U.S. Air Force and a consortium of aerospace companies. Additional support came from MIT and the National Science Foundation. The author is grateful for guidance provided by Stanley Weiss and Janice Klein. This paper is based on material presented in two prior INCOSE conference papers [Browning, 1996a, 1997c] and an unpublished Master's thesis [Browning, 1996b].

Table I. Summary of Case Study Programs

Company	Sector	Program	Phase	Relative Size of Program
Texas Instruments (now Reytheon Systems)	Avionics	GEN-X	Preproduction	Small
McDonnell Douglas (now Boeing)	Airframes	F/A-18E/F	Detailed design (EMD)	Large
General Electric Aircraft Engines (GEAE)	Engines	TACOE and F110+	Support	Medium
Sundstrand	Commercial aircraft	737-700 EPGS	Development	Small
Chrysler	Commercial (nonaircraft)	Small car platform (neon)	Development/production	Large

integration—has thus become an issue of increasing interest. While most contemporary research has focused on the characteristics and effectiveness of teams in general [Katzenbach and Smith, 1993] and IPTs in particular [Cole, 1995, Klein, 1994; Klein and Susman, 1995; Klein and Maurer, 1995; Peters, 1995; Pomponi, 1997; Sheard and Margolis, 1995; Susman and Petrick, 1996; Susman and Ray, 1996], less work has addressed the realms of multiteam or program integration.

This paper presents a motivation for considering interteam issues explicitly and categorizes nine types of *integrative mechanisms* (IMs). Findings from five case studies and literature are then used to elaborate on the appropriateness of each IM. As IM aptness varies as a function of many parameters—such as program stage, size, complexity, risk, etc.—this research does not formulate a universal template for IM application. Rather, the hope is that the lessons learned by these five programs will help others determine the suitability of particular IMs in their situations.

To uncover a variety of interteam integration issues in ongoing programs and to investigate the use of IMs in varied contexts, I used an exploratory, case study research method. I researched and wrote five case studies, focusing on the IMs used, their level of success, and lessons learned. I sought instances where IPT integration was a concern and looked at actions taken in that regard. I made at least one site visit to each program and followed these with telephone conversations and facsimile, electronic mail, and postal correspondence. In each case, information regarding the programs and integration issues came from multiple sources. Full details of these efforts and the five programs themselves are available in Browning [1996b].

The case studies represent a breadth of defense aircraft industry sectors (avionics, airframes, and engines) and span a variety of program sizes[1] and stages. For the sake of comparison and for a wider collection of potential issues, a commercial aircraft program and a commercial nonaircraft program supplement the defense aircraft programs. Table I summarizes the five programs. The purpose of the research includes showing what types of issues might appear if this study was conducted at a more focused level by persons intimately familiar with a particular industry sector or product type. Also, the intent is that the reader will be able to relate at least one of these cases to programs with which she or he is familiar.

2. MOTIVATION: TEAM INTERDEPENDENCE

Product design can involve hundreds or thousands of individuals making millions of design decisions over several years. Few of these determinations can be made in isolation: Design choices involve trade-offs which affect many other product, process, organization, cost, and operational parameters. In support of this interdependence, product design managers have as an essential task the facilitation of the transfer of information among design groups [Allen, 1977]. "Their primary development challenge is to integrate the many subproblem solutions into a well-designed system. . . . The trouble is that such interactions are often poorly understood and are rarely known in advance" [Eppinger et al., 1994].

Developing even complex subsystems (let alone systems) generally involves multiple IPTs, and assigning

[1]In regard to the relative size of the program (Table I), "small" programs involved about 30–60 people, while "large" programs involved over 700 people.

and coordinating their work is challenging. Work allocation to IPTs should consider task overlap and "underlap." Without care, different teams' work may be inadvertently redundant, or tasks might "fall between the cracks" such that each IPT thinks that "someone else" will handle them. Furthermore, task sequencing becomes challenging as IPTs execute once serial tasks in parallel. Thus, IPT integration requires coordination within and between subsystem and system development organizations. One should have special concern about the information that will flow between IPTs as they work. Tausworthe concludes: "A team producing at the fastest rate humanly possible spends half its time coordinating and interfacing" [quoted in Rechtin, 1991: 284]. Work paradigms such as concurrent engineering change traditional work structures and information flows, further complicating IPT integration.

Effective organization integration becomes more challenging as product complexity increases. (Complexity here implies numerous, highly-coupled subsystems and components.) The IPTs working together to develop such a product face a daunting task. Team A needs to know what values team B has set for parameters *x* and *y*; team B needs to know what values team C is using for parameters *w* and *z*; but team C needs to know the result of team A's activities to determine *w* and *z*. Such coupled, "chicken and egg" problems may imply a slow, iterative development process. Without precaution, the number of required interfaces and thus the number of iterations required to converge to an acceptable design can increase exponentially with system complexity. Figure 1 shows how the number of inter-IPT communication channels increases with the

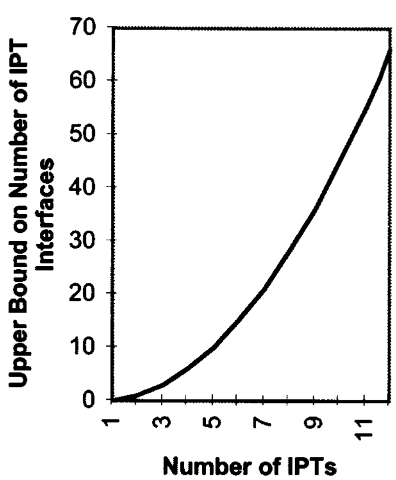


Figure 1 Number of Inter-IPT Interfaces Increases with Number of IPTs.

number of IPTs. (Of course, not every IPT will have to interface with every other IPT, so the curve represents an upper bound.) As the amount of task and IPT interdependence increases, often as a function of product complexity, the number of and traffic on inter-IPT communication channels increases.

If IPTs are held to their useful size—i.e., 10–15 persons—then their number will most certainly be large in complex programs. Some programs have questionably formed "IPTs" with 70, 100, or more members. As programs move towards effective implementation of the IPD paradigm, the number of IPTs generally grows and the issue of IPT integration becomes more acute.

Given these challenges, one perceives the potential for IPT integration issues. The automotive industry has given integration difficulties explicit attention: "One shortcoming that became apparent as [IPTs] worked on a specific product was lack of coordination within the Product Engineering group. This lack of coordination resulted in interface problems between [IPTs] that should have been solved functionally within Product Engineering" [Mattis, 1992, notice to IPTs added]. McCord and Eppinger [1993] also discern the potential for communication problems between interdependent teams:

> Relying solely on . . . an informal communication network for integration . . . means to depend on the engineers to comprehend and initiate all of the necessary interactions between PDTs [Product Development Teams]. Unfortunately, engineers are rarely sensitive to all inter-PDT relationships, especially concerning how their work affects the work of other PDTs. Furthermore, many unforeseen conflicts between PDTs arise throughout the course of the project which are too slowly resolved through informal integration. . . . More formal, planned integration mechanisms must be designed into the organization to ensure necessary information exchange between PDTs and to expose and resolve inter-PDT issues as early and as quickly as possible.

As Rechtin [1991] notes, "The greatest leverage in system architecting is at the interfaces." While true for a system architecture, it also applies to the organization architecture that develops it. A better understanding of the issues underlying IPT interdependence would seem to hold great potential for improving the product development organization and process.

3. INTEGRATIVE MECHANISMS AND FINDINGS

Aware of the issues, we now turn to the implementations that enable us to integrate multiple, interdependent IPTs. *Integrative mechanisms* (IMs) are strategies

Table II. Integrative Mechanisms (IMs)

Integration enablers
1. Systems engineering and interface optimization
2. Improved information and communication technologies
3. Co-location
4. Training
5. "Town meetings"

Integration maintainers
6. Manager mediation
   A. Management hierarchy ("up-over-down")
   B. Heavyweight product managers (HPMs) or integrators
7. Participant mediation
   A. Conflict resolution engineers (CREs)
   B. Liaisons
   C. Engineering liaisons (ELs)
8. Interface "management" groups and integration teams
   A. Predetermined
   B. Impromptu
9. Interface contracts and scorecards

and tools for effectively coordinating actions across teams and groups within a program. As catalysts, they facilitate information flow across communication barriers, such as a company or program's organization structure, incentive systems, location, leadership styles, cultural differences, and management traditions [Morelli, 1993]. IMs must also regulate information flow such that it does not overwhelm or underwhelm its recipients.

This paper categorizes and discusses nine types of IMs.[2] Table II divides these into two groups: (1) integration enablers—IMs which provide for the establishment of integration; and (2) integration maintainers—IMs which monitor and facilitate ongoing integration. Together, all the IMs provide an organization integration "tool kit."

Each IM's appropriateness may not hold for every program, interface, or situation: One must consider a firm's organization (formal and informal, including culture and traditions) and a program's technical information requirements when applying any of these approaches. Rechtin [1991] reminds us: "It is easier to match a system to the human one it supports than the reverse." Effective utilization of IMs requires an understanding of their capabilities, limits, and suitability to circumstances.

The use of IMs often invites trade-offs. For example, improved information and communication technologies can be traded with co-location to an extent. The membership of any arbitration or management group can also be varied to balance between pros and cons at each end of a spectrum. Further possible trade-offs will become evident from the discussions that follow.

The next nine sections describe the nine categories of IMs given in Table II in more detail and discuss case study findings and literature regarding them. The Appendix provides a series of tables that summarize these findings in an abbreviated format.

4. SYSTEMS ENGINEERING AND INTERFACE OPTIMIZATION

4.1. Description

Ideally, systems engineering of the organization is an *a priori* technique, using the product architecture and systems principles, to inform the organization design and IPT breakout. Hence, perhaps organization systems

[2]Other typologies of integration and coordination mechanisms have been developed. The list in Table II expands upon categories developed by McCord and Eppinger [1993]. Coordination mechanisms useful in *design for manufacturability* (DFM) contexts are discussed in Adler "Managing DFM: Learning to Coordinate Product and Process Design" in Susman [1992: 140–156].

engineering should not be classified as an IM in the strictest sense: It might be thought of as the work gloves one must put on before handling the other IM tools.

Gulati and Eppinger [1996] explore the coupling of product architecture to development program organization structure, noting that decisions in one realm affect and even constrain opportunities in the other. Thus, as one considers organization integration issues, one must first look at the system architecture that the organization is charged to develop. Figure 2 shows the proposed relationship: Architecture and organization are linked through the process of problem decomposition and system integration. Decomposition and integration are generalized inverse problems. Therefore, the first step in program integration is to understand as completely as possible (or practical) the nature of the proposed product architecture, especially its decomposition and internal interfaces, for these will directly affect the organization and the ease of integrating the teams working on the various subproblems. The architectural breakdown of the product leads to the work breakdown structure of the process and the team breakdown structure of the organization.

How, specifically, does a product architecture and its associated task set affect the organization and its communication patterns? Morelli et al. [1995] investigate the extent to which coordination-type communication between project groups is predictable given a known task set. With the proposed tasks before them, project participants were able to predict 81% of the communication that, in fact, took place. This result signifies the possibility of designing an organization on the basis of a proposed system architecture and its associated task set. Conceivably, interfaces could be made more optimal with respect to anticipated information flow needs. By paying attention to teams and their interfaces, an organization can be designed for integration [Browning, 1997a].

Given an unprecedented or revolutionary system, however, possessing a thorough understanding of the architecture and the tasks to develop it is understandably difficult. Knowledge of tasks and their duration is essential to the creation of the *statement of work* (SOW), *work breakdown structure* (WBS), and *integrated master schedule* (IMS). This knowledge is likewise crucial for interface planning and management. For upgrades and other largely precedented systems, it is much easier (although not necessarily easy) to outline tasks and their information requirements. Where the architecture and/or tasks are yet to be determined, organization designers must build in flexibility so the organization can adjust once the characteristics of the required IPT interfaces settle out. Baseline organization designs for programs developing unprecedented systems require the most flexibility of all. (Of course, this requires an incentive system that enables and encourages organization flexibility.)

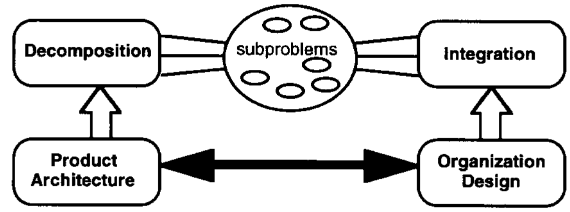


Figure 2 Architecture Tied to Organization Through Decomposition/Integration Problem [Gulati and Eppinger, 1996].

Systems engineering and interface optimization is actually more of an *a priori*, preventive measure—attempting to make IPTs as technically independent as possible, based on the premise that minimal interfaces imply fewer interface issues. Of course, this ultimately stems from the product's architecture—the more modular the architecture, the more independent the teams. As Rechtin [1991] notes, "Choosing the appropriate aggregation of functions is critical in the design of systems." Likewise, choosing the appropriate aggregation and integration of organization functions becomes the critical task of the organization designer. Doing this job well makes the job of the integrators and managers that much easier.

Choosing how to decompose a program organization into IPTs can benefit from the same heuristics Rechtin collected in reference to breaking down a system's architecture (Table III). Here, "communications" refers

Table III. Partitioning Heuristics [Rechtin, 1991]

- In partitioning, choose the elements so that they are as independent as possible—i.e., elements with low external complexity and high internal complexity [Alexander, 1964]
- In partitioning a distributed system, choose a configuration in which local activity is high speed and global activity is slow change[a]
- In partitioning a system into subsystems, choose a configuration in which minimal communications between the subsystems[b]
- Do not partition by slicing through regions where high rates of information exchange are required (e.g., computers)

[a]From Courtois, P. J., "On Time and Space Decomposition of Complex Structures," Communications of the ACM, Vol. 28, No. 6, 1985, pp. 590–603. Quoted in Rechtin [1991].

[b]Excepting massively parallel neural networks.

to "interrelationships, connections, interplay, information flow, etc." Thus, both subsystems and IPTs should be semiisolated to a reasonable extent so that a minimal number of external events have the potential to disturb their inner workings.

The issues of systems architecting are crucial as inputs to any design process. The importance of intelligently grouping functional requirements and decomposing architectural elements to meet those requirements cannot be overstated. It is here that multiteam integration begins, for it is here that much of the difficulty of the integration tasks is determined. However, this paper will not address the tenets of intelligent decomposition, which several authors have discussed [e.g., Alexander, 1964; Altus et al., 1995; Kusiak and Wang, 1993; Michelena and Papalambros, 1995; Pimmler and Eppinger, 1994; Rechtin, 1991].

4.2. Findings

Designing the organization to mirror the product architecture makes common sense and has been advocated in recent literature [e.g., Grady, 1994]. A well-partitioned system will have a minimal number of architectural interfaces, and the teams developing the subsystems will usually require the minimal amount of interaction with each other. Rechtin [1991] sums up this concept with a heuristic akin to that of minimum communications: "Design the elements to make their performance as insensitive to unknown or uncontrollable external influences as practical." This also applies to IPT relationships, where organization designers should maximize the ability to communicate while minimizing the need to do so.

However, most IPTs working as one of many designing a complex system will have some need to communicate with the rest of the program. One wants to avoid the problem experienced by the F-22 program, where ostensibly independent IPTs took their element and focused on it *too* closely:

> When IPTs received the people, funding and authority to develop individual products, each of the teams concentrated solely on its product and overoptimized it. Teams would produce components of outstanding design that were not easily integrated with other components. [Wagner and White, 1995: 38]

When cross-functional, upstream/downstream, customer, and supplier integration is the goal, making the organization mirror the architecture becomes especially challenging. At what level should the assimilation of these functions occur? Including them all in every IPT is not feasible. Teams must share some resources. Constraints will often dictate practical limits of cross-functional integration. However, an understanding of which constraints are the most limiting can help managers decide where to apply the resources that will gradually relax the constraints. Organization inertia is often the biggest constraint of all.

Complex system development requires the exchange of technical information among groups. Understanding the sources and sinks of this information flow can provide insights leading to improved organization and integration. It is important that decisions affecting entire programs and organizations proceed with a systematic foundation. For example, analyzing an organization using a *dependency structure matrix* [DSM—Browning, 1997b; Eppinger, 1997; McCord and Eppinger, 1993] enables a more systematic approach and informs trade-offs regarding the integration of IPTs and functional support groups at appropriate organizational levels and in the application of IMs. None of the case study programs utilized a systems approach to designing and integrating the organization, although some programs had groups in place to enable learning so future programs would have this opportunity.

Systemic approaches to organization have a good chance of leveraging improvements in cost, schedule, and performance. For example, *quality function deployment* (QFD) and approaches such as the *requirements allocation matrices* (RAMs) and *derived requirements allocation matrices* (DRAMs) used at Boeing [Kepchar, 1994] deserve consideration for their ability to systematically flow down requirements and responsibilities through a system architecture and an organization. Getting everyone to buy in to the use of system models can integrate decision makers and lead to more enlightened choices of organization structures and more appropriate applications of IMs. Thus, systems engineering principles, applied to the *organization* that develops the system, are a primary, *a priori* integration enabler.

5. IMPROVED INFORMATION AND COMMUNICATION TECHNOLOGIES

5.1. Description

Information and communication technologies serve to penetrate communication barriers and to increase the capacity, efficiency, and general efficacy of a program's information exchange.[3] Several representative technologies in this broad category include:

[3]For specific research on these intentions, consult Hauptman and Allen [1987] and Jakiela and Orlikowski [1990].

- linked CAD tools
- shared databases
- e-mail
- voice mail
- standardized hardware and software suites
- local and wide area networks (LANs and WANs)
- archival databases for mail and meeting minutes (with friendly search and retrieval mechanisms)
- computer and/or video conferencing
- teleconferencing

Much more has been said in other places about the roles these technologies play within IPTs, but here we focus on their role as an interteam IM.

5.2. Findings

Several technologies have the potential to enable and improve interteam integration. I will present several in turn, beginning with **electronic mail**. Today, employees on most programs have access to e-mail, at least internally. While many attribute the reduction of hard copy memos to the advent of e-mail, others see e-mail as a hindrance to integration. In some programs, people copy their messages to everyone else. It is just as easy to send the message to the whole program as it is to send it to a single person. Besides, one thinks it better to provide the information to everyone, lest some critical recipient be omitted and take offense. Hence, some find themselves inundated with information, much of it nominally relevant. While such "broadcast" messages are an excellent way to keep abreast of a program's (or multiple programs') activities, most people do not have the time to assimilate them all. Instead, people sometimes do not look at *any* of the messages, figuring that any really important notice will come to them through another channel. Soon, with others not able to count on these individuals to read their e-mail, senders do have to resort to other channels, and e-mail becomes little more than another potential time sink. Individuals and teams should realize who really needs to know something; i.e., they should be aware of their interfaces. Understandably, on a multidisciplined IPT, the amount of information to assimilate—because of the additional perspectives considered—will be greater. Eliminating superfluous information becomes even more important here. E-mail messages concerning the program and coming from outside a team could be filtered, perhaps by that team's liaison. Finally, some effort should be made to archive program-related messages, both centrally and individually. Sometimes e-mail documents important decisions. Thus, it would behoove programs to establish guidelines for the use of e-mail if they are going to rely on it as an IM.

Improving **common databases** involves embellishing the breadth and depth of the data stored, increasing accessibility while decreasing access time, establishing standardized formats, and training an ever-greater amount of the workforce in their use. Ideally, an IPT member could access all databases easily and routinely from a single terminal, such as a personal computer on his or her desk. Engineering, manufacturing, schedule, cost, test, and other data of many types should be archived and made readily available. Critical parameters should be tracked regularly for elements, subsystems, and the system as whole; and these data should be easily accessible. To be shared, data must be represented in a common language of mutually understood terms. Sometimes different teams and functional groups use terms differently—either using the same name for different data or giving the same data different names. In these cases, the information receiver often goes to great lengths to extract the desired information. Boeing found this to be a significant barrier in their commercial aircraft group. A truly common database overcomes these obstacles. Sometimes this means team-wide training in a new vocabulary. Sometimes this requires establishing a new data language altogether. Another option is to provide special translator tools in software that make the interface appear seamless (i.e., pre- and postprocessors). An excellent practice for accessing common databases in a standardized format is Boeing's utilization of web browsers and an intranet. CD-ROMs also provide a means of storing easily distributable archives.

Standardizing hardware and software presents a constant challenge—and a dilemma as well. While increased standardization facilitates IPT integration in the short term, its long-term effects are less definite. The global optimum consists of a software policy that provides good interoperability in the short term while maintaining flexibility—through alternatives and wide expertise and by fostering innovation—in the long term. This includes recognizing that the ideal software of today may not be the best choice for the future. Software companies change. So do the companies one works with: The new partner company of the future may have standardized on something else. Moreover, mature products can become slow and inefficient. New technologies leap ahead. Without consideration of the need to maintain flexibility (because one realizes that all of the variables that will influence the choice may not be accounted for), the tendency is to converge on and optimize the nearest suboptimal point. Optimizing at the incorrect point actually places one farther away from the global optimum (because of the reluctance to write off the sunk cost investment). Similarly, the choice of a single software suite that has limited interface and translation capabilities may be fine for a given program, but it may not translate easily to future programs or teams. The trade between standardization as an IM and the risks thereof must be considered. Decisions must consider when it is best to foster innovation through a variety of tools versus when it is best to channel innovation through one tool, even if towards a local optimum.

Electronic file transfer is often essential to interteam integration. Local area networks (LANs) and wide area networks (WANs) are good, fast options. Some form of network or intranet should exist to tie everyone's workstations and terminals together. E-mail can also be used to transfer files. Security concerns make some wary of using e-mail for intersite file transmission. Using encryption can overcome this barrier, although much widely used e-mail software does not currently offer automatic encryption, or else requires special configuration to utilize this capability.

CAD/CAM/CAE systems are a critical IM, facilitating file transfers and standardized formats, aiding in design conversations, and providing "a flexible and unambiguous design representation" [Robertson and Allen, 1991]. With the common point of reference these tools bring, fewer interdisciplinary misunderstandings occur and conversations are more effective. Research by Robertson and Allen [1991: 23] has shown that an increase in performance due to CAD use is most strongly realized when it is explicitly used to enable cross-functional communication.[4]

Israel [1992: 24] provides an example where CAD was used as an IM to enable concurrent engineering in the Convair Division of General Dynamics in the development of an advanced cruise missile:

> The participating engineering functions included structural analysis, human factors, maintenance, and flight dynamics. The primary communication mechanism between these functions was a Mechanical Engineering CAD system. Proposed designs were file transferred from one engineering group to another. Analysis was conducted and the results returned with commentary. The commentaries in this case identified structure over-designs. By using this information early on, a redesigned bulkhead was generated with a significant weight savings. Additional commentary identified a maintenance issue which required the removal of another bulkhead in order to service one of the electronics packages. This removal process would have required two men and a special support dolly. Use of the CAD system helped to incorporate a hinged supporting member, thus eliminating the need for the special dolly and one of the two support personnel. The General Dynamics example is illustrative of the use of a CAD system as a communications enabler which supported information flow and problem identification by overcoming distance and language barriers which typically arise between functional engineering disciplines.

Rosenbaum and Postula [1991: D.3.5] single out the three-dimensional capabilities of CAD tools as their chief integrative characteristic:

> We live in a three-dimensional world. Most people cannot quickly and easily visualize well from two-dimensional views. The result is that designs represented by drawings are frequently the private domain of designers and drafters. It is not surprising then that drawings often yield designs that cannot be manufactured, cannot be maintained, and do not meet customer expectations. . . .
>
> In fact, solid modeling is the key to successful team (concurrent) design. Through solid modeling of parts in extreme detail, very small clearances can be verified (including tolerances) electronically. In similar fashion, electronic mockup of tubing and harnesses can eliminate the need for physical mockups. Companies that have instituted such programs have shown savings in excess of 40 percent.
>
> Probably the most important attribute of the solid modeling approach is that all functions, from design to analysis to manufacturing to estimating to management, have simultaneous access to an unambiguous description of the product—in real time. [Rosenbaum and Postula, 1991: D.3.5]

Today, most development programs in the aerospace industry use CAD/CAM/CAE packages to some extent. As Sundstrand found, however, transitioning to new CAE tools can slow a project down. Such transitions should be avoided midstream whenever possible. Many companies have a software tools functional group that explores new CAE tool options and makes recommendations for future directions. These groups are hopefully aware of the integrative aspects of the tools. Many programs, notably the F/A-18E/F program and the Boeing 777 program [Sabbagh, 1996], have attributed vast improvements in the development cycle to the use of CAE tools such as CATIA™ and Unigraphics™ and CAM tools such as Variation Simulation Analysis™ (VSA). Two- and three-dimensional models have provided for the early recognition of problems (in some cases) and the ability to do rapid, virtual prototyping.

[4]They also recommend each CAD system have a text message template to standardize annotations.

Use of the same tools by the subcontractors has also facilitated integration with these groups.

Scheduling and process modeling software can also contribute to integration. A standardized schedule is a good place to highlight critical issues that could cause delays. Having a common modeling tool from which to analyze and ask questions about the program can provide an integrative effect. Some of the case study programs are experimenting with process modeling and simulation tools. However, the steep learning curves associated with many of these function-packed software packages may inhibit integrative overtures. Simple, highly visual methodologies incorporating *dependency structure matrices* (DSMs) hold promise in this regard [Browning, 1997b, 1998; Eppinger et al., 1994], although software available to work with DSMs is currently limited.

Many non-software tools and methodologies also proved successful as IMs in the case study companies. Boeing's IPD data sheets and GEAE's electronic worksheets serve to standardize the format of data characterization both within and between teams. Boeing's use of the Geometric Dimensioning and Tolerancing (GD&T) language establishes a common vocabulary for multidiscipline interactions. In addition to common reference terms, archives of lessons learned that can be saved and shared not only provide an IM but also foster a policy of learning within the organization. GEAE's Design Record Book and Chrysler's "Book of Knowledge" provide excellent examples of these types of efforts. Some programs plan to use knowledge-based software tools. On a broader scale, well-organized process guides—used *and* provided with training in their interpretation and use—can allow more of the product development process to proceed on an integrated basis. Excellent examples of good practices along these lines include GEAE's *Engine Development Cycle Process Guide* and Raytheon Systems' *RF/Microwave Business Unit Teaming Handbook*. While these particular guides could be expanded by explicitly outlining additional approaches and tools, they provide a common framework for an entire program to approach the design process. Such handbooks probably should include guidelines on the appropriate uses of IMs as well. Finally, one of the most effective, non-software IMs—bulletin boards—has been used for a long time. One should not underestimate their importance, even in an electronic age. Often an entire conference room will have walls filled with status reports and schedules. These provide opportunities for employees to discuss aspects of the program in a casual sense and get a better feel for its breadth and depth.

Note that this IM category includes several information tasks: transfer (dissemination), access, and assimilation. Technologies facilitating any one of these areas may not necessarily further them all. For example, some technologies, such as teleconferencing, make information exchange so expedient that the propensity to not document that exchange increases. One must consider such factors if record keeping is a priority. Taking CAD as another example, researchers have looked at the different roles CAD tools can play in an organization: as physical capital, as support for human capital, or as enablers for improvements in social capital (i.e., as an IM) [Robertson and Allen, 1991]. The existence of three (or more) ways of viewing these types of tools implies that not everyone recognizes them as an IM and that their mere presence does not guarantee superior integration. In fact, some research shows CAD can have a negative effect on integration [Jakiela and Orlikowski, 1990; Murotake and Allen, 1991]. Certainly, the ease of making changes in CAD files does not encourage documentation of the design history. While no one likes excessive documentation, and "improved communication" is seen as the way around it, some amount of design version history is necessary for future access. Hauptman and Allen [1987] highlight some of the major literature on information and communication technologies in their 1987 paper, which discusses the capabilities, drawbacks, and perceptions of these new approaches. Also, much more has been said in other places about the roles many of these technologies need to play *within* IPTs [e.g., see Hartley, 1992; Murotake and Allen, 1991; Robertson and Allen, 1991].

This section has focused on interteam integration aspects of improved information and communication technologies. While the IMs in this category possess great potential to enable integration, they do not adequately suffice as a bandage for an improperly organized program. As Wheelwright and Clark [1992: 242] point out, only programs that have broken down interteam barriers, integrated functional activities, developed structured design processes, and provided appropriate organization and leadership can expect to realize the full benefits of technological solutions.

6. CO-LOCATION

6.1. Description

Co-location involves positioning IPTs and functional support groups (already assuming co-location of the IPTs themselves) in close proximity (usually within sight and sound of each other, but sometimes within walking distance) for the purpose of facilitating communication, both formally and informally. Co-location offers at least two advantages: (1) It enables more expedient resolution of low level issues, and (2) it increases cross-functional and interteam awareness and appreciation.

6.2. Findings

Research by Allen [1997] shows the probability of communicating at least once per week decreases quickly with separation distance of employees. The probability of communication drops below 13% with a separation distance of just 10 meters or more. Since locating everyone in a large program within 10 meters of each other is impossible, co-location must be supplemented with other IMs. Co-location offers one of the most obvious examples of how IMs can be traded off: Greater co-location can reduce the need for other IMs.

To serve as an effective IM, co-location should be considered in terms of its influence on communication patterns rather than purely in terms of distance. If "co-located" IPTs or groups still use the phone as the primary means of communication, for example, they have not tapped the true advantage. If individuals still reside in mazes of cubicles, where seeing if another person is at their desk requires leaving one's own desk—again, perhaps a key benefit of co-location is yet to be realized. The extent to which co-location adds value (i.e., how far to take it) is still a subject for additional research. (Undoubtedly, like other IMs, it will vary by program circumstances.)

Chrysler's Neon program exhibits almost complete program co-location (although manufacturing is still some distance away from product design), with an entire floor of its new Technology Center dedicated to each product platform. On a smaller scale, Sundstrand's lack of cubicles contributed to the open atmosphere intended by implementing co-location. Raytheon Systems' GEN-X program faced several difficulties which co-location might have circumvented, although constraints seemingly limited the possibility. While most programs are co-located to some extent, special care should be taken to group the IPTs and functional groups with the tightest couplings in close proximity. Likewise, important cross-functional, upstream/downstream, and customer/supplier resources should be integrated, perhaps by co-location, within the IPTs, system teams,[5] or program as applicable.

Specialized test labs, manufacturing facilities, multicompany teams, *etc*. constrain co-location. The point of diminishing returns with regard to forcing these outliers to co-locate with the bulk of a program has not been determined and certainly remains a subject of discussion for organization designers. Besides, how far to take co-location (how hard to push for it) varies by program. For measuring the amount of co-location, some proposed metrics include: percentage of IPT co-located, percentage of system team co-located, percentage of program co-located, and ratio of co-located or integrated disciplines to total disciplines.

7. TRAINING

7.1. Description

IPT level integrators recognize the role of training in boosting team performance. In much the same ways as IPTs learn to operate as better teams, they can receive training on how to more adeptly arbitrate issues with other IPTs and utilize the IMs facilitating those interactions.

7.2. Findings

Training can take many forms, from team building to technical. After looking at the five programs and others as well, many interesting aspects of successful training enterprises become apparent. The most beneficial training programs were developed and administered before the program got underway. Teams were launched with team building training and not allowed to flounder without a clear sense of roles and direction. Integrative training provides awareness of the many roles, responsibilities, and contributions of the various disciplines on the program. IPT building training is best undertaken by the entire, multidisciplined team together. Training can help equip the teams with a common understanding of the program's interteam interfaces, in terms of data flow, goals, and priorities. It can also help establish and raise awareness of guidelines for documentation and protocol. Also, training ought to include systems thinking and organization learning skills [Senge, 1990].

Like information, training should be available to the right people, at the right place, and at the right time—with the realization that missing these marks quickly diminishes the value added. Training should not be seen as just an up front "project." Ideally, it should be part of a larger, ongoing enterprise improvement plan. Technical training is best provided during program slack time. In any case, except for team launches and certain interface awareness efforts, training probably does not belong on the critical path.

[5] System teams are groups of highly coupled (interactive and interdependent) teams and functional groups within a program. See McCord and Eppinger [1993] and Browning [1996b, 1997a].

8. "TOWN MEETINGS"

8.1. Description

Town meetings (and other similar rallies) gather everyone on a program together in one place to review the program's progress. Although relatively ineffective for transferring technical information, they serve to boost morale and camaraderie. On large programs, especially ones that span several facilities, such gatherings are logistically impossible. Smaller versions (e.g., all employees at a given site) and other types of meetings with these goals also fit into this category.

8.2. Findings

Most large, complex product development programs, such as those in the defense aerospace industry, find it impossible to bring everyone together at once for a meeting—although some of the case study programs bring sections or all employees at a given site to the occasional "all hands" meeting. A determination of how often town meetings should take place has not been made, although once a month, once a quarter, and after significant milestones are popular options. Management must decide the frequency and the agenda to boost program-wide awareness and camaraderie. The effectiveness of town meetings as an IM has not been established empirically, although most intuitively agree that they can have a positive effect if they are well planned and emphasize integrative concepts.

9. MANAGER MEDIATION

9.1. Description

This IM comes in (at least) two flavors, both of which have in common the facilitation of interface issues primarily by managers.

9.1.1. "Up-Over-Down" Some organizations use an "up-over-down" approach to interface management as shown in Figure 3. Managers above the IPTs mediate interteam issues rather than having the teams' members deal with them directly. This mechanism works better for relatively independent teams requiring little information transfer and the resolution of few issues. As the interdependence between teams increases, however, management quickly becomes overloaded and a barrier to information flow. Note that in this arrangement management is generally reactive as issues arise, stepping in only to resolve issues or review progress.

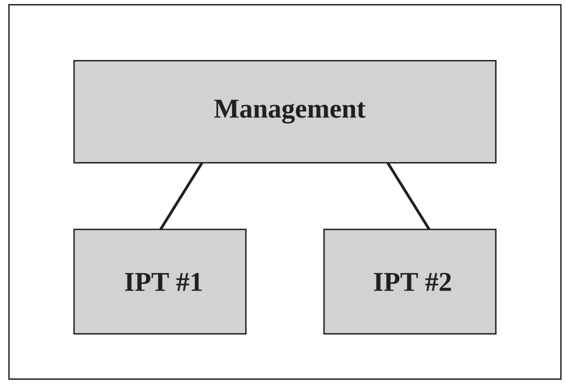


Figure 3 "Up-over-down" Management Arbitration.

9.1.2. Heavyweight Product Managers (Integrators) A *heavyweight product manager* (HPM), as defined by Clark and Fujimoto [1991], has "direct access to the working-level engineers" when necessary and exercises "strong direct and indirect influence across all functions and activities in the project." Lawrence and Lorsch [1967] describe this role as that of an "integrator." The HPM has more clout than the functional managers. Table IV lists the characteristics of successful HPMs in the automotive industry. After reviewing this list, one should discern the difficulty in finding such a superhuman individual, especially in the aerospace industry, with its great complexity and scope. Perhaps the key here is to contain these characteristics in a cadre of individuals who perform the HPM role together (with one individual as this group's leader). While this role/IM focuses on the individual's (or small group's) integrative influence, it does not exclude direct IPT interfaces, as does the up-over-down approach. On the contrary—it encourages them.

9.2. Findings

Companies reducing their numbers of managers found that those who are left often do not have enough time to micromanage their workers. In effect, this produced "empowerment by default." Hierarchical management structures, especially sparse ones, find it difficult to continue to arbitrate and mediate all interteam issues. In cases where teams have infrequent interactions, a hierarchical structure might suffice, yet probably not as the most efficient IM. Besides, these situations are less likely in complex development programs. Instead, managers will have to provide clear direction and certify that knowledge of that direction figures into decisions throughout the program.

Table IV. Characteristics of Effective Heavyweight Product Managers in the Automotive Industry [Clark and Fujimoto, 1991]

- Coordination responsibility in wide areas, including production and sales as well as engineering
- Coordination responsibility for the entire project period from concept to market
- Responsibility for concept creation and championing as well as cross-functional coordination
- Responsibility for specification, cost target, layout, and major component choices
- Responsibility for ensuring that the product concept is accurately translated into technical details
- Frequent and direct communication with designers and engineers at the working level as well as through liaisons
- Maintain direct contact with customers
- Possess multilingual and multidisciplined abilities in order to communicate effectively with marketers, designers, engineers, testers, plant managers, controllers, and so forth
- Role and talents in managing conflict surpass those of neutral referees or passive conflict managers; they may initiate conflicts to prevent product designs or plans from deviating from the original product concept
- Possess market imagination and the ability to forecast future customer expectations based on ambiguous and equivocal clues in the present market
- Circulate among project people and strongly advocate the product concept rather than do paperwork and conduct formal meetings
- Mostly engineers by training, they possess broad (if not deep) knowledge of total vehicle engineering and process engineering

Mohrman et al. [1995] recommend the following pointers for direction setting:

- Define a strategy, communicate it, and operationalize it at all systemic levels. Ideally, strategy informs the trade-offs that are made throughout the organization.
- Align goals vertically and laterally. Strategy gets translated into goals.
- Align support service goals.
- Choose goals that are measurable.
- Assign rewards in accordance with organization goals.
- Do not assume that direction precludes empowerment.
- Plan collectively.
- Facilitate flexibility and responsiveness.

The three aerospace case studies turned up no HPMs, although some managers possessed some of the characteristics of effective HPMs (Table 4). Indeed, the role of HPM as described in the automotive industry becomes difficult to implement in large, complex, unprecedented, and infrequent development programs that require the integration of thousands of individuals and tasks. A single person probably cannot get their arms around the breadth and depth of the program to the extent of effectively integrating it alone. In some of the cases, a program manager was able to assemble a group of people, either formally or informally, to be his “tentacles” throughout the organization. While plausible, it becomes more difficult through multiple layers of management. This “distributed integrator” role requires systems thinking and good communication and negotiation abilities. All else being equal, the HPM role becomes a little more feasible when the program is a precedented, lower risk one. Along these lines, Susman and Ray [1996] found—in general, and at least in the early stages—that low risk projects should have more power in the program organization, while high risk projects would do well to balance the effects of program and functional managerial influences.

The most practicable management mediation mechanism seems to be management teams composed entirely of managers from various functions and/or teams. This team “forges strategy and direction for the constellation of teams in the business unit, makes resource trade-offs between the different teams based on the strategy and the needs of the teams, and manages the performance of the various teams that report to them" [Mohrman et al., 1995: 121].

10. PARTICIPANT MEDIATION

10.1. Description

This IM category combines several types of non-management interface mediators. Most of the terminology for these subcategories comes from the automotive industry [McCord and Eppinger, 1993].

10.1.1. Conflict Resolution Engineers (CREs)
When technical conflicts are brought to their attention, CREs act as dedicated arbitrators between IPTs. "Conflict" refers to disagreements over technical issues or trade-offs that affect two or more teams. An example of a CRE is a "zone engineer," who arbitrates technical conflicts between teams within a given section of a program. The CRE handles "turf" issues of a technical nature, provided that both sets of "turf" fall under his or her jurisdiction. If the CRE comes from the ranks of management, the role would fit under Management Arbitration. More likely, however, the CRE would be a functional guru from the technical area within question or a systems engineer.

10.1.2. Liaisons Akin to a CRE, a liaison also works to resolve technical issues at team interfaces. However, liaisons play a more proactive role as they facilitate continuous and intensive information exchange. This role involves seeking out technical conflicts, discovering them earlier, and resolving them faster by participating in interteam interactions. A liaison is a member of one IPT and serves that team entirely (as far as the liaison role is concerned) by facilitating that IPT's communications with all other IPTs—a "from one to many" and a "from many to one" relationship.

10.1.3. Engineering Liaisons (ELs) Whereas liaisons reside in one IPT, ELs are formal members of two or more IPTs whose interface the EL coordinates. Hence, the EL will go to the team meetings and other team activities of two or more IPTs. Not only do ELs establish and maintain a firm communication link between IPTs, they also perform specific technical tasks on at least one of the teams. They are working, development engineers.

10.2. Findings

With a well-trained person placed at an appropriate interface, liaison roles—liaisons, engineering liaisons (ELs), and conflict resolution engineers (CREs)—have positive effects. Their advantages include: the rapid communication of information, the opportunity to experience and learn from the workings of other teams, and the removal of some of the interteam interaction burden from the team leader (where it usually falls) [Sheard and Margolis, 1995]. Boeing uses zone engineers to oversee interteam issues in the F/A-18E/F program. They have a liaison functional organization (although it is diminishing) that provides upstream/downstream integrators. Some IPTs have assigned designated persons to specific interfaces. On the other hand, some, like GEAE, do not desire roles where people "only talk to others." (This requires every team member to have some liaison functions.) Hence, some have found participant mediation and arbitration of interface issues expedient, while others see specialized roles as unnecessary.

Special consideration should precede the assignment of liaisons, ELs, or CREs. A good interteam representative will ideally possess a thorough understanding of and respect for the tasks and goals of both or all of the teams with which he or she works. Not only do knowledge and empathy lead to better decisions, they also lead to reciprocal respect and willingness to provide meaningful information. Often the EL can gain knowledge and respect by actually doing work on one or more of the teams. In addition, a participative mediator needs a systems view of the issues and a clear conception of the overall strategy that provides guidance for trade-off decisions.

Guidelines should also establish the amount of information filtering a liaison will provide. "The liaison concept requires a strong [IPT] leader, to enforce both the attendance of liaisons at other meetings and the brevity and relevance of the comments they bring back" [Sheard and Margolis, 1995]. Liaison-type roles fall along a spectrum, with authoritative roles at one end and simple information transmitters at the other. The amount of power and responsibility bestowed on a liaison, EL, or CRE will also vary depending on the character of the interface they oversee.

Although not investigated explicitly in the five case studies, overlapping membership is another (informal) form of participant mediation. Many programs form IPTs where the members move from team to team. Although this practice can result in synergy and integration if members participate on just two IPTs, the strategy can have negative effects if employees participate on more than two IPTs. Wheelwright and Clark [1992] found that an engineer's ability to add value to a task peaks when assigned to two development projects and falls below 50% when she or he must spend time with three or more projects.

11. INTERFACE "MANAGEMENT" GROUPS AND INTEGRATION TEAMS[6]

11.1. Description

This IM includes groups—perhaps a mixture of both management and participants—that "manage" IPT interfaces, largely from an "up-over-down" perspective. (Here "management" is enclosed in quotations as referring to management of the interface itself, which may or may not be done entirely—if at all—by managers. If such a group is composed entirely of management personnel, it should fit more properly under Management Arbitration.) Note the difference between *integrated* teams and *integration* teams, which may or may not be "integrated" (i.e., cross-functional) themselves. At least two genres of such "integration teams" exist: predetermined and impromptu (for lack of better terms).

11.1.1. Predetermined

Predetermined integration teams are formed from the outset of a program (or their date of formation is fixed from the outset), chartered to deal with foreseen issues of a complex, critical nature that involve multiple IPTs. Such preordination would most likely stem from a systems engineering analysis that reveals the crucial interfaces.

11.1.2. Impromptu

Impromptu interface management groups also deal with a single, complex, critical issue concerning multiple IPTs. However, these teams are formed when such an issue crops up in the middle of an ongoing program, unforeseen by the organization planners (at least at the program's outset). They dissolve when the issue is resolved. Often, such groups form to handle action items from product reviews. Impromptu integration teams consist of engineers and others drawn from across a project. Essentially, they correspond to task forces, ad hoc teams, splinter teams, tiger teams, and action teams.

11.2. Findings

Most of the programs studied use some type of team approach to interface oversight and mediation. A key systems team in the GEN-X program has a Management Team to handle issues between its subteams. The F/A-18E/F program uses a multilevel organization to provide a hierarchy of management and participative groups that oversee interactions between the teams at the level immediately below them. GEAE uses Leadership Teams to review individual teams and their interactions with their program. These teams often consist of both managers and participants.

Other programs have also used integration teams of various sorts to monitor and facilitate the successful development of subsystems or systems developed by multiple teams. The F-22 program, for example, uses Analysis and Integration Teams (AITs) to encourage IPT interaction [Wagner and White, 1995]. The reorganized Space Station program also uses AITs, which are responsible for systems engineering and are organized in tiers corresponding to the hierarchical IPT organization [Peters, 1995].

Integration teams can be proactive and anticipatory to varying degrees. The most successful ones have a clear idea of their roles, resources, and responsibilities. An integration team on the GEN-X program never had its roles defined and thus spent much of its meeting time questioning the relevancy of issues brought to its attention and postponing decisions. While integration teams exist at a higher hierarchical level than IPTs in a systems sense, they do not necessarily consist of individuals with a higher hierarchical rank in the organization.

12. INTERFACE CONTRACTS AND SCORECARDS

12.1. Description

Another way to enable and maintain integration is to formally define interactions with interface contracts, sometimes called *interface control documents* (ICDs). These documents explicitly note what data are expected to be transferred or shared and have the joint approval of all involved parties. Along with these, scorecards assist in monitoring the success of the interactions.

12.2. Findings

While formal interface contracts are an extremely helpful IM, the documentation effort required to establish and maintain them can be extraordinary. They are most crucial at the outset of a program, to raise awareness about key interactions. After groups know their interfaces, the contracts can be revisited less frequently. However, the dynamic nature of programs, especially when transitioning from one phase to another, requires appropriate maintenance of the contracts. Raytheon Systems' *RF/Microwave Business Unit Teaming Handbook* provides an excellent example of the beginnings of explicit interface identification and documentation.

Scorecards have the potential to be a valuable way of evaluating interteam interactions. However, a score-

[6]Mohrman et al. [1995: 120–128] have a good discussion on management and participative integration teams.

card is only as good as the metrics it notes and the ability of the user to evaluate them. Without a clear understanding of the information that needs to flow across an interface, its appropriate frequency, and the ease of transmission, proper scoring will be difficult. The process of developing helpful scorecards will in itself enlighten their users to the characteristics of effective interteam interfaces.

13. CONCLUSIONS

This paper frames the issue of IPT interdependence and categorizes nine IMs which are broadly representative of approaches taken within several industries to the mandatory task of IPT integration. Other categories and subcategories may exist as well. The paper also summarizes findings from five case studies of IM use in complex system development programs. While this approach does not permit the generalization of universal guidelines for IMs, it provides increased understanding of appropriate IM applications, both as integration enablers and as integration maintainers. The Appendix summarizes these findings.

In addition to the specific findings and lessons with respect to individual IMs, this work also reinforces general understandings concerning organization integration. IM suitability clearly depends on the characteristics of industries, programs, and organizations. Fundamental to the application of IMs is recognition and consideration of the need for such integration at the time IPTs are established. IMs are most effective when applied appropriately—with knowledge of their strengths and weaknesses in the environment under consideration and based on a systematic approach, stemming from the architectural structure of the product. Above all, the realization that interteam integration, when addressed explicitly and handled appropriately, has the potential to greatly decrease development cost and schedule and product performance risk should spur greater integrative efforts.

REFERENCES

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

Allen, T. J., *Managing the Flow of Technology: Technology Transfer and the Dissemination of Technological Information Within the R&D Organization*, 1984 paperback ed., MIT Press, Cambridge, MA, 1977.

Altus, S. S., Kroo, I. M., and Gage, P. J., "A Genetic Algorithm for Scheduling and Decomposition of Multidisciplinary Design Problems," *Proc. 21st ASME Des. Automat. Conf.*, Boston, September 1995.

Browning, T. R., "Multi-Team Integration: Interdependence and Integrative Mechanisms," *Proc. Sixth Annu. Int. Symp. INCOSE*, Boston, 7–11 July, 1996a, pp. 787–794.

Browning, T. R., "Systematic IPT Integration in Lean Development Programs," Master's Thesis, (Aero./TPP), Massachusetts Institute of Technology, Cambridge, MA, 1996b.

Browning, T. R., "Exploring Integrative Mechanisms with a View Towards Design for Integration," *Prod. Fourth ISPE Int. Conf. Concurrent Eng. Res. Appl.*, Rochester, MI, 20–22, August, 1997a, pp. 83–90.

Browning, T. R., "An Introduction to the Use of Design Structure Matrices for Systems Engineering, Project Management, and Organization Planning," Working Paper #WP97-001-18, Lean Aircraft Initiative, Cambridge, MA, 18 February 1997b.

Browning, T. R., "Mechanisms for Interteam Integration: Findings from Five Case Studies," *Proc, Seventh Annu. Int. Symp. INCOSE*, Los Angeles, 3–7 August 1997c, pp. 649–656.

Browning T. R., "Use of Dependency Structure Matrices for Product Development Cycle Time Reduction," *Prod. Fifth ISPE Int. Conf. Concurrent Eng. Res. Appl.*, Tokyo, Japan, 15–17 July 1998.

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

Cole, W. E., "Cooking Up a Batch of Team Synergy: Ingredients for Setting Up Successful Teams," *Program Manager*, 1995, pp. 28–33.

Eppinger, S. D., Whitney, D. E., Smith, R. P., and Gebala, D. A., "A Model-Based Method for Organizing Tasks in Product Development," *Research in Engineering Design*, Vol. 6, 1994, pp. 1–13.

Eppinger, S. D., "A Planning Method for Integration of Large-Scale Engineering Systems," *Proc. Int. Conf. Eng. Des. ICED—97*, Tampere, 19–21 August 1997.

Grady, J. O., *System Integration*, CRC Press, Boca Raton, FL, 1994.

Gulati, R. K., and Eppinger, S. D., "The Coupling of Product Architecture and Organizational Structure Decisions," Working Paper No. 151, MIT International Center for Research on the Management of Technology, Cambridge, MA, May 1996.

Hartley, J. R., "Concurrent Engineering Is Wasted Without CAD/CAM," in *Concurrent Engineering: Shortening Lead Times, Raising Quality, and Lowering Costs*, Productivity Press, Cambridge, MA, Chap. 11, 1992.

Hauptman, O., and Allen, T. J., "The Influence of Communication Technologies on Organizational Structure: A Conceptual Model for Future Research," Working Paper No. 1892-87, MIT Sloan School of Management, Cambridge, MA, 1987.

Israel, S., "An Integrated Performance Measurements System for Product Development," Master's Thesis (Mgt.), Massachusetts Institute of Technology, Cambridge, MA, 1992.

Jakiela, M. J., and Orlikowski, W. J., "Back to the Drawing Board? Computer-Mediated Communication Tools for

Engineers," Working Paper No 3145-90, MIT Sloan School of Management, Cambridge, MA, 1990.

Katzenbach, J. R., and Smith, D. K., *The Wisdom of Teams: Creating the High-Performance Organization*, Harvard Business School Press, Boston, 1993.

Kepchar, K. J., "Breathing Systems Engineering Life into Existing DoD Programs," *Proc. Fourth Annu. Symp. NCOSE*, San Jose, CA, 10–12 August 1994, pp. 121–127.

Klein, J. A., "A Case Study of Self-Directed Work Teams at Boeing Defense and Space Group—Corinth," Case Study LEAN 94-02, MIT Lean Aircraft Initiative, Cambridge, MA, 1994.

Klein, J. A., and Susman, G. I., "Lean Aircraft Initiative Organization & Human Resources (O&HR) Survey Feedback-Integrated Product Teams (IPTs)," White Paper LEAN 95-03, MIT Lean Aircraft Initiative, Cambridge, MA, 1995.

Klein, J. A., and Maurer, P. M., "Integrators, Not Generalists Needed: A Case Study of Integrated Product Development Teams," in *Advances in Interdisciplinary Studies of Work Teams*, JAI Press, Vol. 2, pp. 93–115, 1995.

Kusiak, A., and Wang, J., "Decomposition of the Design Process," *Journal of Mechanical Design*, Vol. 115, pp. 687–695, 1993.

Lawrence, P. R., and Lorsch, J. W., "New Management Job: The Integrator," *Harvard Business Review*, Vol. 45, 1967, pp. 142–151.

Mattis, D. P., "Case Study on the Strategies of Systems Engineering," Master's Thesis (Mgt.), Massachusetts Institute of Technology, Cambridge, MA, 1992.

McCord, K. R., and Eppinger, S. D., "Managing the Integration Problem in Concurrent Engineering," Working Paper No. 3594, MIT Sloan School of Management, Cambridge, MA, 1993.

Michelena, N. F., and Papalambros, P. Y., "Optimal Model-based Decomposition of Powertrain System Design," *Proc. ASME Adv. Des. Automat.*, 1995.

Mohrman, S. A., Cohen, S. G., and Mohrman, A. M., Jr., *Designing Team-Based Organizations*, Jossey-Bass, San Francisco, 1995.

Morelli, M. D., "Evaluating Information Transfer in Product Development," Master's Thesis (Mgt.), Massachusetts Institute of Technology, Cambridge, MA, 1993.

Morelli, M. D., Eppinger, S. D., and Gulati, R. K., "Predicting Technical Communication in Product Development Organizations," *IEEE Transactions on Engineering Management*, Vol. 42, 1995, pp. 215–222.

Murotake, D. K., and Allen, T. J., "Computer Aided Engineering and Project Performance: Managing a Double-Edged Sword," Working Paper No. 47, MIT Sloan School of Management, International Center for Research on the Management of Technology, Cambridge, MA, 1991.

Peters, J. F., "The Transition of Functional Organizations to Integrated Product Teams on the Space Station Program," *Proc. Fifth Annu. Int. Symp. INCOSE*, St. Louis, 22–26 July 1995.

Pimmler, T. U., and Eppinger, S. D., "Integration Analysis of Product Decompositions," *Proc. ASME Sixth Int. Conf. Des. Theory Methodol.*, Minneapolis, MN, September 1994.

Pomponi, R. A., "The Organization of Integrated Product Teams (Literature Review)," Working Paper No. WP97-002-34, MIT Lean Aircraft Initiative, Cambridge, MA, 6 October 1997.

Rechtin, E., *Systems Architecting: Creating & Building Complex Systems*, Prentice Hall, Englewood Cliffs, 1991.

Robertson, D., and Allen, T. J., "CAD System Use and Engineering Performance in Mechanical Design," Working Paper No. 32, MIT Sloan School of Management, International Center for Research on the Management of Technology, Cambridge, MA, 1991.

Rosenbaum, E. S., and Postula, F. D., "Computer-Aided Engineering Integrates Product Development," *AACE Transactions*, pp. D.3.1–D.3.7, 1991.

Sabbagh, K., *21st Century Jet: The Making and Marketing of the Boeing 777*, Scribner, New York, 1996.

Seng, P. M., *The Fifth Discipline*, Doubleday, New York, 1990.

Sheard, S. A., and Margolis, M. E., "Team Structures for Systems Engineering in an IPT Environment," *Proc. Fifth Annu. Int. Symp. INCOSE*, St. Louis, 22–26 July 1995.

Susman, G. I., Ed. *Integrating Design and Manufacturing for Competitive Advantage*, Oxford University Press, New York, 1992.

Susman, G., and Petrick, I., "Lean Aircraft Initiative Product Development Team Effectiveness Follow-up Study," Working Paper LEAN 96-01, MIT Lean Aircraft Initiative, Cambridge, MA, 1996.

Susman, G., and Ray, J., "Product Development Team Effectiveness," *IEEE Transactions in Engineering Management*, 1996.

Wagner, G. F., and White, R. L., "F-22 Program Integrated Product Development Teams," *Program Manager*, pp. 34–38, 1995.

Wheelwright, S. C., and Clark, K. B., *Revolutionizing Development: Quantum Leaps in Speed, Efficiency, and Quality*, Free Press, New York, 1992.

APPENDIX: SUMMARY OF FINDINGS REGARDING INTEGRATIVE MECHANISMS

A.1. Systems Engineering and Interface Optimization

1. As much as possible, design the organization to mirror the product architecture. A well-partitioned system will have a minimal number of intersubsystem interfaces. Therefore, the teams developing these subsystems will require a minimal amount of interaction.
2. Maximize the ability of teams to communicate while minimizing their need to do so.
3. Meanwhile, do not let teams get "tunnel vision" and disregard their external interfaces.

4. A systematic approach works best for grouping the IPTs and functional support groups into system teams and for determining how integration will occur within and between organization levels. People-based dependency structure matrices (DSMs) provide a useful tool in this regard. System models used for these purposes make useful IMs themselves.
5. None of the case study programs utilized a systems approach to designing and integrating the organization, although some programs had groups in place to enable learning so future programs would have this opportunity.

A.2. Improved Information and Communication Technologies

1. E-mail can lead to overcommunication: people broadcast messages to large distribution lists. Most people do not have the time to assimilate all the messages. It would behoove programs to establish some formal guidelines for the use of e-mail if they are going to rely on it as an IM.
2. Common databases should span high bandwidth interfaces. Common databases require accessibility, decreased access time, standardized data formats, and user training. Intranets and browser software show promise for providing such access. All data relevant to design objectives, including cost data, should be readily available.
3. Critical system, subsystem, and component parameters such as cost and weight should be tracked regularly versus design goals and made obvious to everyone.
4. Standardization of software facilitates integration but can become a trade-off with innovation and flexibility.
5. Seamless electronic file transfer is essential. LANs and intranets are good, fast options. E-mail is sometimes an option as well. Security issues are barriers; encryption needs to be available and simple to use.
6. CAD/CAM/CAE systems are critical, providing a common reference point for cross-functional designers. Yet, formal guidelines are often necessary to achieve optimal integration.
7. Integrated scheduling leads to improved multiteam integration.
8. Other successful communication IMs include: standardized data sheets and electronic worksheets, data language standards, archives of lessons learned, well-organized process guides (in conjunction with training and incentives), and bulletin boards.
9. Note that communication expediency IMs such as teleconferencing tend to circumvent adequate documentation of design decisions.

A.3. Training

1. Develop and administer appropriate training up front in the program, even as a carefully contained critical path item.
2. Integrative training provides interteam role, responsibility, and contribution awareness.
3. IPT-building training is best experienced by the entire, multidisciplined team together.
4. Equip the teams with a common understanding of the program's interteam interfaces—in terms of data flow, goals, and priorities.
5. Establish and raise awareness of documentation protocol.
6. Like information, training should be available to the right people, at the right place, and at the right time. Missing these marks quickly diminishes the value added.

A.4. Co-location

1. Co-location is an excellent IM, although many do not utilize it in its most effective form. Usefulness is greatly diminished when group members are more than 10 meters apart. Hence, for large programs, correctly choosing which groups to co-locate is crucial. Systematic methods, such as the people-based DSM, facilitate making appropriate co-location choices.
2. Specialized test labs, manufacturing facilities, multicompany teams, etc. constrain co-location.
3. Greater co-location can reduce the need for other IMs, while a lack of co-location requires special consideration of other IMs.

A.5. "Town Meetings"

1. Most intuitively agree that town meetings have a positive effect if they are well-planned and emphasize integrative concepts.
2. How often? Once a month, once a quarter, and after significant milestones are popular options.

A.6. Manager Mediation

1. Hierarchical management structures are *not* the most efficient way to arbitrate and mediate all interteam issues.
2. Management must provide a clear vision of program direction, ensuring that knowledge of that direction is held broadly within the program.

Strategy = Goals = Incentives
= Good low level decisions

3. The role of a *heavyweight project manager* (HPM) is difficult to implement on a program-wide basis in large, complex programs, but may apply at the system team level.
4. The most practicable management-related, interface mediation mechanism seems to be management teams (similar to integration teams—see IM #8) composed of managers from multiple functions and/or teams.

A.7. Participant Mediation

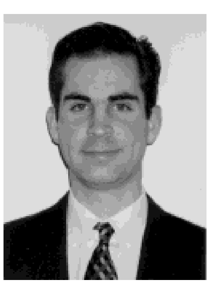
1. Liaison roles have positive effects: the rapid communication of information, the opportunity to experience and learn from the workings of other teams, and the removal of some of the interface management burden from the team leader.
2. Zone engineers oversee interteam issues in the F/A–18E/F program.
3. IPTs can designate members to handle specific interfaces.
4. A good interteam representative possesses a thorough understanding of and respect for the tasks and goals of both or all of the teams with which he or she works.
5. A participative mediator needs a systems view of the issues and a clear conception of the overall strategy that provides guidance for the trades that must be made.
6. Guidelines should establish the amount of information filtering the liaison will provide.
7. Liaison-type roles fall along a spectrum, with authoritative roles at one end and simple information transmitters at the other. The amount of power and responsibility bestowed on a liaison will vary depending on the character of the interface they oversee.
8. Overlapping team membership is another (informal) form of participant mediation, but employees should not be part of more than two teams at once.

A.8. Interface "Management" Groups and Integration Teams

1. Most of the five programs studied use a team approach to interface oversight and mediation.
2. These teams consist of both managers and participants.
3. While integration teams exist at a higher level than IPTs in a systems hierarchy sense, they need not necessarily consist of individuals with a higher rank in the organization.
4. Integration teams can be proactive and anticipatory to varying degrees. The most successful ones are very clear on their roles, resources, and responsibilities.

A.9. Interface Contracts and Scorecards

1. The documentation effort required to establish and maintain interface contracts can be extraordinary. They are most crucial at the outset of a program, to raise awareness about key interactions. After that, maintaining them should not be allowed to consume as much effort.
2. Scorecards provide a valuable way of evaluating interteam interactions. A scorecard is only as good as the metrics it notes and the ability of the user to evaluate them.
3. Without a clear understanding of the information that needs to flow across an interface, its appropriate frequency, and the ease of transmission, proper scoring will be difficult.
4. The process of developing helpful scorecards will in itself enlighten their users to the characteristics of effective interteam interfaces.



Tyson R. Browning is currently a doctoral student in the Technology, Management, and Policy Program (interdisciplinary systems engineering and technology management) at the Massachusetts Institute of Technology and a research assistant with the Lean Aerospace Initiative, a consortium funded by government and industry participants. Mr. Browning's current research investigates management strategies and tools for complex system product development, including methods for project cost, schedule, and performance risk trade-offs. Mr. Browning has previous work experience at Honeywell, Inc. and at Los Alamos National Laboratory and is a member of INCOSE, AIAA, and other organizations.