Babson College
Digital Knowledge at Babson

Technology, Operations & Information Management Working Papers

Technology, Operations and Information Management Division

6-28-2012

Orchestrating coordination: Reducing rework in complex product development

Benjamin Dawson
Massachusetts Institute of Technology

Sebastian K. Fixson
Babson College, sfixson@babson.edu

Daniel Whitney
Massachusetts Institute of Technology

Follow this and additional works at: http://digitalknowledge.babson.edu/toimwp

Part of the Operations Research, Systems Engineering and Industrial Engineering Commons

Recommended Citation

Dawson, Benjamin; Fixson, Sebastian K.; and Whitney, Daniel, "Orchestrating coordination: Reducing rework in complex product development" (2012). *Technology, Operations & Information Management Working Papers.* Paper 4.
http://digitalknowledge.babson.edu/toimwp/4

This Article is brought to you for free and open access by the Technology, Operations and Information Management Division at Digital Knowledge at Babson. It has been accepted for inclusion in Technology, Operations & Information Management Working Papers by an authorized administrator of Digital Knowledge at Babson. For more information, please contact digitalknowledge@babson.edu.

Orchestrating coordination: Reducing rework in complex product development

Benjamin Dawson
Engineering Systems Division
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
bdawson@mit.edu

Sebastian K. Fixson[1)]
Technology, Operations, & Information Management
Babson College, Tomasso Hall 226
Babson Park, MA 02457, USA
sfixson@babson.edu

Daniel Whitney
Engineering Systems Division
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
dwhitney@mit.edu

This version: June 28, 2012

[1)] Corresponding author

Orchestrating coordination:
Reducing rework in complex product development

Abstract

Rework is a persistent problem in many complex product development projects. Major strategies that have been proposed to reduce rework in complex product development projects are reducing the task interdependencies, i.e., modularizing the design, and improving the communication between participating engineers, i.e., co-locating them and encouraging collaboration. Yet, some recent industrial projects could not follow either strategy due to external constraints, with dramatic negative consequences for product development performance. Building on these observations, we develop a simulation game to test various strategies to mitigate the detrimental misalignment effects when a complex product with an integral product architecture is developed by multiple firms forming a modular supply chain. Our simulation experiments replicate system behavior observed in real systems with much higher levels of complexity, and revised strategies using faster information sharing, and temporary acceptance of suboptimal partial solutions show a significant product development performance improvement. We discuss implications for research and practice in the management of complex product development projects.

Keywords: Iterations, Modularity, Product Architecture, Incentives

1 Introduction

Rework is a persistent problem in many complex product development projects. Revisiting tasks and activities that have been assumed to be completed adds time to the project schedule and expenses to the project costs. Recognizing that complexity itself can cause and amplify these oscillations in project progress, guidelines typically recommend reducing the task interdependencies, i.e., to modularize the design, and improving the communication between participating engineers, i.e., co-locating them and encouraging collaboration. But what if a firm finds itself in a misalignment situation, in which it can neither modularize the product nor integrate the organization? What exactly causes the low performance when a complex integral product is developed by separate entities forming a rather modular supply chain? And most importantly, what levers do participating firms have to mitigate the problems that result from such a misalignment configuration?

Motivated by these questions, and by industrial example cases from recent years in which the misalignment contributed to losses of hundreds of millions of dollars, in this paper we develop a simulation game with which we can mimic the coordination challenges of developing large, complex, and integral products through modular organizational structures. The resulting oscillations in interacting design parameters prevent the participants from executing the design work efficiently. We then test alternative strategies to explore how they could improve product development performance, and find that faster information sharing, and tentative acceptance of suboptimal partial solutions drastically improve performance. These results suggest that the design of incentives and trust building mechanisms should receive broader attention in the misalignment debate.

2 The Effect of Rework on Product Development Performance

On the project-level, new product development performance is typically measured in product performance, development time, and development cost (1995; Clark & Fujimoto, 1991; Reinertsen, 1997). Product performance can comprise several dimensions such as quality, innovativeness, manufacturability, etc. Development time describes the time that elapses between project start and project completion, and development cost measures the resources in labor (e.g., engineers' time), materials, and equipment consumed to execute the product development project.

2.1 Causes and consequences of rework

Many complex product development projects fail to achieve their set targets on one, or more, of the three dimensions product performance, time, and cost (Cooper, 2005). Of those projects that achieve satisfactory product performance, many miss their scheduled launch date, consume more resources than anticipated, or both. The slip in schedule is often caused by unanticipated needs to revisit tasks that were already considered complete, a phenomenon known as rework (Browning & Ramasesh, 2007). Rework encompasses all documented changes that are made to a design during the development project. This phenomenon has also been studied under the name of engineering change (Loch & Terwiesch, 1999; Terwiesch & Loch, 1999) and various remedies have been suggested such as prevention, front-loading, effectiveness, efficiency, and learning (Fricke, Gebhard, Negele, & Igenbergs, 2000; Jarratt, Eckert, Caldwell, & Clarkson, 2011). Of these five types of remedies the latter three (effectiveness, efficiency, and learning) argue for a more productive way of dealing with the changes that do occur (Loch & Terwiesch, 1999). Front-loading follows the idea that if changes could be made earlier they tended to be less costly and could be implemented faster (Thomke & Fujimoto, 2000). Finally, the idea of simply needing fewer changes in the process of developing a new product obviously points to immediate improvement of project performance.

However, reducing the number of changes during a development project has been made more challenging through the acceleration of competition in recent years. The increasing focus on the project performance dimension 'development time' has led to the growing use of concurrent engineering, i.e., increasing overlap of formerly sequential engineering activities, and the greater degree of uncertainty with which many engineering activities now have to start has created its own wave of rework. Countermeasures that have been suggested are the reduction of the interdependence between activities, i.e., a modularization of the design (Terwiesch, Loch, & De Meyer, 2002; Yassine, Joglekar, Braha, Eppinger, & Whitney, 2003) and an increase of communication between relevant engineers (Clark & Fujimoto, 1991; Gokpinar, Hopp, & Iravani, 2010).

Some circumstances, however, prevent the direct implementation of these countermeasures. This is particularly the case when the product to be developed exhibits an integral product architecture, i.e., it cannot be modularized, at least not in the short term, and yet the product development is conducted in an extended enterprise with multiple firms participating. This type of misalignment is not uncommon, and it can have real and dramatic consequences.

2.2 Occurrence of misalignment and its consequences

That the occurrences of misalignments between product architecture and organizational structure are not rare has been documented in a recent meta-analysis. In their study on what they call the 'mirroring hypothesis' (i.e., the match between product structure and organizational structure), Colfer and Baldwin (2010) find that in 27% of the cases rather integral products and systems were developed by multiple organizations connected by non-hierarchical relationships (of the remaining 73%, 69% confirmed the mirroring hypothesis, i.e., represent cases of alignment, and 4% exhibited a second form of misalignment in which modular products are developed by integral organizations).[1]

To understand the consequences of such a misalignment between integral product architecture and rather modular organizational structure, consider one of the more famous cases of recent years: the Boeing 787 aircraft. The project exhibited an unprecedented combination of an, at least in part, integral product architecture, and a development organization that stretched across various companies all over the world (Figure 1). Admittedly, describing an entire aircraft as integral is somewhat simplistic. On the other hand, some of its key requirements clearly are. For example, (low) weight is not only one of the most critical features of an aircraft, weight is also what Ulrich and Ellison (1999) call a holistic customer requirement, meaning the entire product contributes to its fulfillment, which is clearly non-modular. In addition, while on some levels the various suppliers concentrated on certain subsystems, the line that separates components from one supplier from components of another supplier often went right through the subsystems (consider components such as the fuselage or wings in Figure 1).

Insert Figure 1 about here

Hindsight is always 20/20, but the performance consequences of misalignment have been disastrous. While the new airplane sold exceptionally well–starting in 2004 the airline industry had ordered over 900 of these airplanes within a few years–the company repeatedly missed delivery targets. The first delivery was originally scheduled for the summer of 2007, but the

[1] The set of articles analyzed by Colfer and Baldwin can be considered representative of the management literature, and the innovation management literature in particular. We recognize that representation in scholarly journals does not necessarily translate into representation of real cases, but we see no reason to assume that there is not at least a substantial correlation.

actual delivery of the first airplane to the launch customer ANA occurred in the fall of 2011. The exact costs of this four-year delay are unknown outside Boeing, but one can speculate that the combination of delaying revenue by several years, paying contractual penalties for late deliveries, paying for engineering resources for much longer than anticipated, and the cost of lost customers (notice the decline of cumulative orders in Figure 2 after its peak), adds up to hundreds of millions, if not billions, of dollars.

Insert Figure 2 about here

2.3 Causes of misalignment

Given the detrimental effects that a misalignment between product architecture and organizational architecture can entail, why would firms put themselves into such a position in the first place? To answer this question, as an example we will further unpack the situation faced by Boeing when it developed the 787.

At first, product architecture appears to be a firm's choice. While that is true in some situations, this choice can be severely constrained in others. Consider modern aircraft at the beginning of the 21st century. The market for airplanes is increasingly driven by airlines' concern for the fuel efficiency of their fleets. Besides environmental concerns, rising oil prices make fuel consumption an increasing fraction of an aircraft's operation costs. Oil prices are at a historic high (Figure 3), and globalization and rising incomes in many developing countries will require a substantial resource revolution if this trend is to be changed in any meaningful way (Dobbs, Oppenheim, & Thompson, 2012).

Insert Figure 3 about here

Because an airplane's weight directly impacts its fuel consumption, rising oil prices have increased pressure on aircraft manufacturers to reduce the weight of their products. Fundamentally, there are two avenues to achieve this goal: choosing materials with a lower specific weight, and optimizing the overall design with respect to weight. Boeing chose to do both: to build an aircraft using an unprecedented fraction of carbon-fiber composites instead of aluminum, and pushing the design optimization to new extremes. Although some parts of an airplane could be considered modular, some components clearly are not. Consider the wing of a modern aircraft. It must provide the lift for the plane, transmit the force from the engines to the fuselage, and in many cases serves as a fuel tank. The optimal design for one of these functions is rarely the optimal design for another; often the best design for one role is not even feasible for another. Engineers for such structures must therefore generate designs that perform acceptably across all functions' performance parameters. This search for an acceptable design across multiple requirements is made even more difficult when combined with new materials and manufacturing processes. In summary, the fact that substantial portions of a modern airplane exhibit an integral product architecture is driven in part by performance expectations and in part by the nature of the product, neither of which can be influenced by the firm.

What about the organizational architecture? While theoretically this is a variable under the firm's control, several forces pushed Boeing towards outsourcing over 70% of the development work to suppliers under a model known as risk-sharing partnerships (Holmes, 2006). First, and most apt to its name, the risk sharing partner model allowed Boeing to spread the enormous cost of developing the aircraft across several companies. Instead of paying for development costs itself, Boeing offered its risk sharing partner suppliers an equity stake in the aircraft in exchange for paying development costs themselves. Second, Boeing needed to locate some manufacturing work in countries such as China and India as part of offset agreements allowing it to sell airplanes in those quickly growing markets. In addition, outsourcing the design gave Boeing access to engineering labor beyond its own walls and helped improve design for manufacture. Finally, choosing suppliers from around the world also helped hedge Boeing against the risks of currency fluctuations.

The risk-sharing partnerships are governed by contracts that simultaneously incorporate cost, schedule, and performance targets. In its day-to-day operations the relationships are structured as an extended enterprise in which the Original Equipment Manufacturer (OEM), acts as an integrator and hub for interfacing partners to share information and coordinate their development efforts. Often in these situations, an OEM discourages suppliers from collaborating directly with one another to maintain protection of its own proprietary information, encouraging them instead to use the OEM as an intermediary. It has also been speculated that this was an approach to maintain schedule pressure on all suppliers by preventing "fast" suppliers from learning of delays at others.

In summary, requirements for capital and expertise, as well as concessions to customers for a certain geographical distribution of work pushed Boeing towards outsourcing more than 2/3 of the project, despite the fact that the aircraft exhibits to large degree an integral product architecture.

Recent research has begun to explore the underlying causes of poor performance in these misalignment situations. Studying a large organization developing aircraft engines, Sosa *et al.* (2004) create an alignment matrix by overlaying a design interface matrix (which resembles the product architecture) with a team interaction matrix (which resembles the organizational architecture). They find that when strong design interfaces between separate systems did not receive the appropriate cross-team attention, significant cost and delays resulted. Similarly, studying two cases in the aerospace industry, Kratzer *et al.* (2011) find that informal networks, while less efficient than formal networks, are nevertheless critical for project success. Using a fine-grained data set from automotive development, Gokpinar *et al.* (2010) identify the degree to which engineers 'undercommunicate' relative to an expected level of communication as suggested by the technical product architecture indicative for quality problems as measured by later warranty claims. Finally, Ülkü and Schmidt (2011) identify existing capability differences between suppliers and buyers as one potential factor that makes a deviation of the perfect the optimal configuration choice.

2.4 Research question and method

Given that for firms like an aircraft OEM it can be unavoidable to find itself in a misalignment situation, our research question became the following: *Provided that neither the product architecture nor the high-level organizational architecture can be changed, what mechanisms cause low project performance in complex product development, in particular with respect to development time, and what can the partners do to improve the project performance?*

Given the massive scale and complexity of the real-life projects we were interested in, we decided to study this question with a simulation approach. The simulation was designed to mimic the challenges of developing composite aerostructures in a misalignment situation and was conducted with the participation of aerospace engineers who were working in such a scenario. Next we will describe the product development process in this context, followed by a discussion of the simulation set-up and results.

3 Aerostructure Product Development Process

3.1 Contractual arrangement

When airlines commit to purchase an aircraft they incorporate contract clauses regarding target mass to offset the costs of excess mass. If the delivered aircraft exceeds this target mass, the airline is entitled to assess penalties against the OEM in proportion to the overage, and these mass targets cascade into supplier contracts. In its request for quote (RFQ) documents sent to potential suppliers, the OEM proposes a mass target for the component in question, as well as targets for price and delivery dates. In general, proposed mass targets are estimates based on mass distributions in similar metallic aircraft. However, with the new opportunities and constraints of composite materials, there is no guarantee the ideal distribution will remain the same for all components. Contract targets are negotiable, but constitute primary performance measures by which the OEM evaluates bids, so suppliers are reluctant to revise them upward.

Aerostructure supply contracts include penalty and incentive clauses tied to agreed-upon target component mass. If the delivered component is significantly overweight, the penalties could be large enough to eliminate the supplier's profit for that part. Similarly, when the supplier delivers a component weighing less than the target mass, it receives an incentive payment proportional to the mass savings. Penalties per unit of mass are typically much higher than incentive payments per unit of mass.

Excess mass also adds to the supply chain costs of purchasing, transporting, and processing composite material and to additional airport landing fees for heavier aircraft. These factors add to the pressure for suppliers and the OEM to reduce mass.

3.2 Product development metrics

In order to fly safely, and to receive flight certification from regulatory agencies such as the US Federal Aviation Administration or the European Aviation Safety Administration, airframes must be engineered to ensure the structure is strong enough to withstand all the loads they must bear. The loads on a component, combined with its geometric and material strength properties, determine the stress in each part of the component. If stress in any component is too high, the component will fail.

Safety factors, sometimes called design factors or reserve factors, are an important set of metrics in structural engineering, including for aerostructures. These values give the ratio of allowable stress divided by actual or calculated stress. For example, if the heaviest loads on a component are expected to create 100 units of stress, requirements might call for a safety factor of 1.5, meaning the design must withstand 150 units of stress. If the calculated stress exceeds the allowable stress, the safety factor will fall below the required safety factor. Safety factors below the target value are unacceptable in a final design—they indicate the design cannot withstand all the loads it is intended to bear. On the other hand, safety factors above the target value indicate excess material that decreases fuel efficiency without adding structural value. In practice, engineers often treat the safety factors of areas within each component as a proxy metric to identify and quantify excess mass.

Target safety factors can vary such that some loads and components might require safety factors of 2.0 while others might require 1.5. For simplicity and clarity, we normalize safety factors to 1.0 so that a safety factor of 1.0 represents the condition where the maximum allowable stress per the requirement is equal to the calculated stress at the point in question. This means that in all further discussions, the ideal safety factor for any component is 1.0.

3.3 The sizing loop

How do design engineers arrive at, or come close to the ideal safety factor? They try to determine the proper geometries and thicknesses to balance stress and mass in their component. For a composite airframe, the design thickness determines the number of composite material layers to be laid on one another and cured to form the part, which in turn determines the mass. The load per unit thickness applied to the design then determines the stress in the component. Thus engineers can trade mass and stress for each other, within limits. The activities of defining thicknesses, applying loads, and calculating stresses alternate in what is called the sizing loop. The term sizing loop is used because these tasks form a goal-seeking feedback loop whereby engineers attempt to meet stress requirements with minimum mass, generally using safety factors as a metric of optimality. One might expect this process to converge quickly on a final design. But it typically does not because the load on any given part comes in large measure from the other parts attached to it, and some of those parts are made by other suppliers. Inter-part load transfers provide the product integrality in this situation while multiple suppliers provide the supply chain modularity. This misalignment, together with the communication lags induced by working across organizational boundaries, requires the suppliers and the OEM to engage in the sizing loop.

The sizing loop process begins when the OEM incorporates design updates from suppliers into its global finite element model for stress analysis of the entire aircraft or for a large section such as the wing. By evaluating the finite element model against the structural loads applied to the overall aircraft, the OEM determines the loads applied to each component. While the external environment applies some of the loads, many of them are called interface loads because they transfer from one airframe part to another at the interface between the two. Ribs in the aircraft's wing, for example, only experience loads as exerted on them by neighboring components and not from external air pressures.

Figure 4 shows the tasks in the sizing loop. The overall sizing loop is actually two sizing loops, one inside the other. The inner loop, labeled the supplier sizing loop, occurs at the supplier stress team level and consists of adjusting thicknesses to balance stress and mass given the most recently delivered interface loads; the long-dashed (green) line shows the path of tasks and decisions in this loop. The outer loop, labeled the enterprise loop, treats design updates from suppliers the same way suppliers treat thickness changes for their components; the short-dashed (blue) lines show the tasks and decisions for this loop. Design changes feed into the global finite element model for the entire airframe to evaluate how well the current designs collectively meet overall stress and mass requirements.

Insert Figure 4 about here

After running its global finite element stress model, the OEM notifies suppliers of updated load sets for each component. Each supplier then downloads the updated loads data and converts it into a format appropriate for its local finite element stress analysis tools. Then the supplier team incorporates the new loads in a local stress model, which it uses to evaluate and adjust its design. Running the local stress model typically requires a few hours and engineers receive reports of safety factors after the full analysis is complete. Supplier engineers use safety factors as a direct measure of whether the design meets stress requirements and as a proxy measure for whether there are areas with excess or insufficient mass. Accordingly, they add material to areas with safety factors below 1.0 and may remove it from areas with safety factors above 1.0 in an effort to reduce mass, then run the local stress model again. The engineers follow this process until all safety factors fall within upper and lower safety factor thresholds. When this design update is complete the supplier submits its updated design to the OEM for the next enterprise sizing loop iteration.

The lowest allowable safety factor in a submitted design at any phase of the detailed design process is 1.0, meaning all parts of the design meet minimum stress requirements. The upper threshold is between about 1.05 and 1.10, reflecting an aggressive goal to minimize excess material in any part of the component design. If designs meet these requirements before milestone deadlines, engineers typically continue optimizing the design to remove excess mass. This approach reflects the program OEM's expectations, and also matches aerospace engineering practice in general.

Because mass and stress create conflicting goals they form a dynamic system, and their pursuit can contribute to oscillation in design parameters in the solution search. This oscillation delays the convergence of the process to a target value. The problem of oscillation is well-known in control theory. In product development, and particularly in the development of complex products, it is often experienced as rework, iterations, or engineering change (Clarkson, Simons, & Eckert, 2004; Loch, Mihm, & Huchzermeier, 2003; Terwiesch & Loch, 1999; Yassine, et al., 2003).

3.4 Loop gain

Control theory describes how dynamic systems with feedback are managed to achieve desired outcomes. Under control theory, goal-seeking negative feedback loops like the sizing loop either periodically or continuously compare a system's current output against its desired output, sending a control signal to the system to adjust system parameters to decrease the gap between the current and target output, often in proportion to the size of the gap. The enterprise sizing loop periodically samples its output by running the global finite element model, and the suppliers' sizing loops sample their outputs by running local stress models.

The parameter that determines the relative amplitude of the control signal is called the loop gain. A system with high loop gain adjusts toward its target state more quickly or forcefully than a system with low loop gain. Control theory shows that high loop gain can cause the system to oscillate, and even higher loop gain can cause the system to go unstable, with each successive oscillation being larger than the previous one.

In the sizing loop the intuitively attractive loop gain is 1.0, meaning that engineers want to remove all discrepancy between current designs and the ideal design in one design iteration and without "overcorrecting" for discrepancies. In many real product development settings, loop gain is indeed high in both the enterprise and supplier sizing loops. Engineers make changes intended to bring safety factors as close to 1.0 as possible in each iteration of the supplier sizing loop, indicating high loop gain. Similarly, each supplier tends to iterate through the supplier sizing loop until the component design both meets stress requirements and is relatively optimized for mass as measured by the safety factor parameters discussed earlier. As a consequence, suppliers often aim to submit designs within a narrow target band, for example with all safety factors above 1.0 and below 1.05 or 1.10.

It is important to understand that in classical feedback systems, time delay has the effect of increasing the loop gain. Thus local loop gain (the tendency of the engineers to seek the ideal safety factor quickly) can combine invisibly with time delay to create an oscillatory sizing loop.

OEM expectations reinforce this aggressive standard. In the program studied, general policy often is to run the global finite element model quarterly, with the expectation that after receiving loads updates, suppliers optimize designs to this standard before submitting them for the next cycle. The primary reason for quarterly, rather than more frequent, sizing loop iterations is a fear of injecting chaos into the process by releasing new load updates before suppliers have finished updating designs in response to the most recent set. This philosophy is consistent with Braha and Bar-Yam's (2007) observation that designs fail to converge when the rate at which tasks are affected by neighboring changes exceeds the task's internal completion rate.

In summary, the sizing loop is at risk of oscillating due to two practices that seem sensible in isolation: the supplier engineers seek to submit a capable design with satisfactory safety factors every time while the OEM delays sending the newly calculated loads to the suppliers to reduce chaos. In fact, according to the experienced aerostructures engineers and managers who played the simulation game described below, the sizing loop usually oscillates and converges slowly. Control theory, however, tells us that different behaviors (submitting incapable designs and providing feedback more often) could reduce the oscillations and make the sizing loop converge faster. But these different behaviors are counter-intuitive and might be rejected by both the OEM and the supplier engineers, especially if they conflict with their experience and intuition. Instead they might attribute the team's challenges to the simple fact that aircraft are complex products and composite materials are a challenging new technology. A credible demonstration of an alternative strategy's effectiveness is critical to motivating its implementation

4 Simulation Game Design

Directly experimenting with changes to the product development strategy in such a large and complex aircraft program is infeasible. Instead, we developed a simulation exercise game with a design challenge patterned after this industry's product development environment and structure. We use the game to test the effects of strategies to mitigate the negative effects of misalignment configurations in complex product development. Our hypothesis is that the sizing loop will converge faster if the participants take deliberate steps to reduce the loop gain.

The game design incorporates the process and information structure of the sizing loop described earlier and the pressures that the OEM and supplier firms face in projects like the one described above. It mimics the structure of actual processes, incentives, constraints, data flow paths, and feedback data. Table 1 summarizes how the simulation game is built to mirror the actual ecosystem of aerostructure development, and notes a few of the important differences.

Insert Table 1 about here

4.1 Simulation game task and objective

The simulation game's task is for a four-player team to design an optimized, but relatively simple cantilever truss (Figure 5) with a predetermined set of loads applied at its joints (the external loads are not revealed to the game players, only the net loads at the joints).[2] The truss is described as the structure for the wings of the fictional Wright Flyer III airplane. It should be as light and thin as possible without exceeding a maximum stress threshold in any beam; as long as it meets stress requirements, the lighter the truss the more trusses the team sells. The team also sells more trusses if it completes development quickly than if it does not. These objectives mirror the incentives typical in the industry to meet stress requirements, minimize mass, and finish design definition quickly.

Insert Figure 5 about here

----

[2] The truss is intentionally more tightly coupled than the actual aerostructures in question to compensate for the simplicity of its structure and to ensure the effects of the interdependencies are quickly and powerfully felt during game play.

Three participants play the role of suppliers designing their respective sections of the truss, an inboard section, outboard section, and a covers section; here abbreviated as IS, OS, and CS, respectively. The fourth participant plays the role of OEM, described below. Some parts of each section overlap members of another section—they connect the same pair of joints of the truss—and thus directly share load-bearing functions with those of another section. This mimics the actual aircraft program where overlapping components similarly share load-bearing functions. In the game, the effective thickness of the truss between joints connected by overlapping parts is the sum of their thicknesses. Each of the overlapping parts bears the load in proportion to its thickness relative to the combined thicknesses of both parts. Thus, if the inboard section's CD member has a thickness of 200 and the outboard section's CD members has a thickness of 100, the combined thickness is 300 and the inboard section's CD part bears two thirds of the load between joints C and D while the outboard section's CD part bears one third of the load.

The distribution of mass in the final truss design is equitable if overlapping truss parts bear the load between their shared joints equally. For example, if the true optimum thickness for the two overlapping parts connecting joints C and D is 300, this could be achieved in any configuration where inboard and outboard CD parts' thicknesses sum to 300. However, the mass is distributed most equitably if the load is shared equally, which would happen if each part's thickness is 150. Comparisons of thickness values against optimal thickness values assume that if the part in question overlaps another part, that in the true optimum configuration the overlapping part have equal thickness and thus bear their shared load equally.

The three section design players' (SDP) work is to adjust part thicknesses, checking safety factors via a local stress model, and to submit design updates to a fourth player, the integrator. The integrator mimics the OEM's role by running a global stress model on the whole truss as section designs are updated, and then disseminates updated loads for the sections' local stress models.[3]

4.2 Performance measurement

The game's primary performance metric for human players is profit. In the game scenario, several customers have placed several contingent orders with requirements that the final design falls below a customer-specified mass threshold and final product definition occurs before a specified time. These requirements create parallel pressures to minimize final product mass and time to market typical in the industry. The values for these requirements were generated as uncorrelated normal distributions. An additional requirement, ostensibly from a regulatory agency, specifies the maximum allowable stress in any truss member in order for the product to be sold.

4.3 Player actions

At any point in the game, each SDP may take any of several actions:

- Adjust the thicknesses of one or more of the truss members in his or her section.
- Submit the current design to the integrator.
- Run the local stress model to calculate safety factors based on section thicknesses and current loads.
- Write a message to the integrator.
- After the integrator runs the global model, SDPs receive notification that updated loads on their section joints are available and may retrieve them as soon thereafter as they wish.

The integrator may also take any one of several actions:

[3] In real airplane design, stress analysis is a computationally intensive activity. Companies often use supercomputers to analyze finite element models of the design. Fortunately much simpler approaches are readily available for analyzing simple trusses like the one in the game. The game analyzes stress using the direct stiffness method. This method treats each member of the structure as a spring governed by Hooke's law and uses linear algebra to closely approximate stresses in the truss. This method is the computational basis for most finite element analysis software, though most packages add significant sophistication.

- Run the global stress model, which incorporates the all submitted section designs. Upon completion of the global model, all SDPs are immediately notified that new loads are available.
- Set upper or lower bounds to one or more thickness parameters. This is equivalent to instituting a design freeze on certain parameters or constraining them arbitrarily.
- Adjust the maximum or minimum safety factors expected of designs before submission.
- Adjust the target completion time. This has no effect on customer orders, but communicates an expected completion time to SDPs.
- Write a message to a SDP.

The integrator player can always see the total profits and number of sales it would achieve if the design were finalized at the present time and mass and at the target time and mass. The integrator player can also find out at any time what sales and profits would be if final design occurred an integrator-specified number of virtual weeks after the present time or at the current time with an integrator-specified percent change from the current mass. This provides the integrator a method to evaluate, after the design meets stress requirements, the tradeoff between ending the game to lock in sales with approaching deadlines or to delay completion in hopes of reducing mass to capture additional sales.

Finally, data logging occurs in the background during game play. Each time a player executes an action, JavaScript code records the action taken and all key system parameters, such as section masses, truss member thicknesses, and safety factors, at the time the action was taken. This data is used to examine the patterns of behavior in game play and to support post-mortem discussion of the game among participants.

4.4 Set-up preparations

Before running the actual simulations we determined the true optimum design, the lowest mass configuration that still meets stress requirements, and were therefore able to compare final designs that occurred during the game to the true optimum.

We also developed an agent-based version of the game to simulate product development policies typical in the industry, and used the results to calibrate the game for play with human players.

5 Simulation game results

We designed the experiment of humans playing the game as a two-round exercise. In the first round, participants were free to pursue their intuitive strategies. Design oscillations which delayed convergence occurred in every case we played the game. Results from a pre-game survey of participants and participants' comments during and after the first round of the game provided clarifying insights about the intuitive strategies participants pursued. These generally matched the policies described by engineers in the industry. We ran two instances of the game.

In both cases, the first-round results were disappointing. After 55 minutes (82 virtual weeks) one team had never achieved an acceptable design, and the other finished with only a small number of orders with unexpired deadlines. As in reality, team participants found themselves struggling against oscillating loads which delayed final design definition well beyond the initial target of 40 virtual weeks. As an example, Figure 6 shows the design changes made by the outboard SDP; the top portion of Figure 7 shows the loads as observed by the same player, as a function of all the changes made by all SDPs; and the top portion of Figure 8 shows the safety factors for the outboard section truss members during round one.

Insert Figure 6 about here

Insert Figure 7 about here

Insert Figure 8 about here

Game participants’ behavior patterns indicated certain mental models by which they reasoned about the structure they were designing and the process by which they designed it. In short, all members of the team acted as though the truss sections were significantly more independent than they actually were. After the first round, one participant explained “we didn’t understand what affected what in the game [truss].” This seems to have expressed itself in players discounting the effect of other components’ designs on the loads experienced by their component. As a result, each player treated the current load set as if it were the final load set or close to it. Similarly, participants did not seem to recognize the effect of their design changes on other sections.

There are at least two possible reasons for this mindset, both of which may have contributed to game participants’ approach. The first is that participants may assume that when future and final loads are unknown, it is best to treat the current loads as a reasonable approximation of them. The second is that because participants spent little time studying how their sections interacted, they were unaware of the interactions or discounted their impacts.

When game participants treated current loads as if they were final loads, they effectively treated the current optimum design as the true optimum. The fundamental source of oscillation in a product development system like this is the discounting of loads uncertainty that leads participants to spend longer optimizing a design than is desirable given the uncertainty in the parameters on which it is based. And the participants’ own pursuit of ‘perfect’ designs made the oscillations only worse.

Feedback, in the form of changing loads from other participants was so delayed that the design change process likely appeared to be an open loop. Each SDP was very aware that others’ changes affected his loads, but much less cognizant of his impacts on theirs. This should not be surprising; there were no direct incentives to manage impacts to other sections, and there was sufficient time pressure that participants began making adjustments without studying section interactions.

After the first round of the game, teams conducted a facilitated post-mortem examination of their disappointing results and how they might be improved. We first reviewed the charts shown in the previous section that describe how all the key parameters evolved over the course of the game. Next we discussed plausible sources of the oscillation in light of the revelation that external loads had never changed.

Analytical modeling work has identified a number of potential strategies that can help reduce the oscillation that prevents fast project completion. Mihm *et al.* (2003) suggest the following six strategies: (i) Limit System Size, (ii) Modularize System, and Robust Design Methods, (iii) Cut Interdependencies: Thinning, (iv) Immediate Broadcast of Design Updates, (v) Release Preliminary Information, and (vi) Cooperate: Optimize Overarching Chunks As a System. With strategies (i), (ii) and (iii) essentially representing a change in product architecture, which we ruled out, and strategy (vi) imagining a degree of cooperation unrealistic in our setting, we focused the discussion with the game players on the remaining two strategies.

In the second round of the game, teams tried being less aggressive in their response to change and quickly cycling local and global models to keep every team member's data as "fresh" as possible (strategy (iv)). Integrators would run their global models and disseminate results as soon as they received any update from any SDPs. SDPs would reduce their gain in response to changed inputs and submit updated designs when they had achieved only a "70% solution" adjustment to updated loads. In essence, they lowered their loop gain. This required SDPs and integrators to both significantly widen the band of safety factors acceptable for submission (strategy (v)).

Teams were dramatically more successful during the second round of the game. Where one team had failed to even achieve an acceptable design in the 82 virtual weeks of the first round, this time they achieved an acceptable design in 22 virtual weeks, then optimized it until they ended the game at 34 virtual weeks. The mass of the least optimal final section design was only 4.49% greater than the true optimum and the overall mass was within 1.28% of the true optimum. While oscillation still occurred, the periods of oscillations were much shorter and the amplitude much smaller.

This new approach kept the interface loads on sections much more stable (bottom portion of Figure 7; note that the vertical scale from the first round is an order of magnitude larger than the second round scale). Similarly, the absence of wild swings in safety factors helped players avoid large swings in their thicknesses as well (bottom portion of Figure 8). Overall, smaller incremental changes to thicknesses with each sizing loop helped keep the loads stable as shown above and kept the mass relatively stable as well.

6 Discussion

Our simulation results suggest several insights. First, even relatively simple systems very quickly become too complicated to be intuitively understood. As a consequence, people working on these systems cannot be expected to easily understand overall system behavior, and thus need support from the systems integrator. Second, incentives on the local level in complex systems can lead to oscillations of project convergence, and thus to delays of project completion. While they attempt to align the various parties, in effect they achieve almost the opposite. Third, by manipulating the process rules for projects in which multiple parties collaborate on creating innovative products and solutions, the overall project performance can be significantly improved. Below we discuss theoretical and managerial implications of these observations.

6.1 Role of integrator's competence in system integration

As the rules for collaboration in complex projects are often set by the system integrator, it needs to develop a high level of system integration competence. Others have noted that the successful system integrators know more than they make (Brusoni, Prencipe, & Pavitt, 2001) and our results support this notion. The system integrator is in the best position to develop a deep understanding of the systemic behavior of the complex product or system to be developed. Developing this understanding may be achieved by continuously framing and re-framing the problem at hand, thus involving more abstract levels of innovation such as knowledge areas of understanding, practices, and capabilities (Brusoni & Prencipe, 2011). This includes careful consideration of the degree of outsourcing as too much outsourcing might endanger the capability to reintegrate (Weigelt, 2009). A practical example of this might be the development of superior IT systems and the careful development of rules and standards (Argyres, 1999).

6.2 Role of incentives

The structure of incentives throughout the game and throughout the real extended enterprise significantly affects the behavior of the actors therein. In our simulation, after the first round one participant observed that "one of the situations we have in some of these [aircraft] programs is that various parts have penalties associated with weight. So what you end up with is each partner [firm] trying to optimize its own area and sub-optimizing everything else in the meantime." The focus of incentives on local performance, rather than system performance, in both the game and the extended enterprise, make it easy for both sets of teams to get "stuck" at local optima that may be quite far from the system's global optimum. For the same reasons, they reinforce a mindset that each component is relatively independent of the others, which in fact it often is not in complex systems. These local incentives contribute to oscillations.

In particular, because each supplier and section designer faces significant penalties for exceeding mass targets for its own part or section, and incentive payments for designing below the mass target, they will resist any changes which require adding mass to the design of their own parts. In systems with independent components this is a desirable behavior (Mihm, 2010), but it becomes problematic in tightly coupled systems with nonlinear interdependencies. In such systems, there may be many situations where increasing the mass for one part would allow design changes in other parts that provide a net decrease in mass for the system. More broadly, in coupled systems the apparently suboptimal design of one component might allow for a more optimal global system design. In these cases, incentives oriented around local performance will generate resistance to system improvements that come at the expense of individual components. Instead of seeking the optimal interface for the wider system, each supplier organization will attempt to minimize its contribution to load sharing and persuade or force the team designing the interfacing part to bear as much of the shared loads as possible. In our game, though participants could not negotiate directly with one another, some indirect negotiation-like behavior occurred. For example, several participants commented they had avoided increasing the thickness of truss members in their sections, somewhat akin to refusing to add material in a component, in hopes that they could either find another way to reduce the stress in that part or that their counterpart would make changes to reduce it.

6.3 Role of trust and contracts

The reality of modern large-scale engineering work is moving from in-house development by large integrated companies toward development by extended enterprises—teams of cooperating firms–but we do not see a corresponding shift toward modular product design. The Boeing 787 example illustrates that in some cases the design must be even more integrated than before. In those misalignment configurations, as our simulation exercise demonstrates, a project easily oscillates and sometimes fails to converge, despite the best intentions of reasonably knowledgeable people. As the discussion above indicates, the initial penalties create a disincentive to trust that one would be rewarded if he 'took a hit for the team.'

On the other hand, the observed performance improvement as a consequence of changes in participants' actions and behavior, which themselves were triggered by changes in the process rules, raise questions about how such an approach could be formalized. For dyadic situations, Sommer and Loch (2009) suggest for activities under unforeseeable uncertainty to choose effort-based remuneration when the action of the agent can be monitored. If that is impossible, or too costly, they recommend switching to outcome-based incentives with downside protection, and, in case of the principal not observing the unforeseen event, an upward incentive.

In the original set-up in our game, the contracts provided outcome based incentives, certainly without downside protection. However, the introduction of the 70% rule drastically lowered this penalty (or, viewed inversely, increased the downside protection). In our setting, two aspects make focusing more on process-based contracts interesting. First, due to the fact that more than two players are involved, the outcome is no longer dependent only on one's own action but also on the action of other players. Second, because there are multiple rounds to be played, there is an opportunity for trust to be built via repeated interactions with the other parties. Given that all innovation work is characterized by the parties' inability to specify the outcome ex ante, the contract design perhaps should be a series of iterative steps that achieves efficient project progress through building trust iteratively between participants. In other words, the contract for our misalignment setting would be both a series of consecutive incentives. Recent research on contracting for innovation articulates a similar insight: "The key … is creating a regime in which regular and reciprocal provision of information about each party's capacity and willingness to cooperate teaches the parties how to collaborate more effectively, binding them more tightly to imprecisely defined common projects through increased switching costs *resulting from that process* – or alerting them to possible breakdowns before the costs of failure in the relation become ruinous." (Gilson, Sabel, & Scott, 2009:499, *emphasis added*) A second feature of these contracts specifying consecutive steps for both parties is that they combine formal and informal elements. Gilson et al. (2010) suggest that, rather than crowding out one another, formal and informal elements can in fact complement each other to create a 'braid' that helps building trust over time.

Note that the problem in our setting goes beyond a dyadic problem in that it exhibits the added challenge of being a multi-party problem; in addition to well-defined OEM-supplier relationships, it requires well-defined supplier-supplier relationships, and the ability for all parties to protect proprietary information as appropriate.

We should also note that the game participants, all experienced engineers, were unable to design a relatively simple system whose interactions are relatively visible. The real situation is much more complex. The interactions are almost completely invisible and require weeks of intense computation to pursue and resolve. Thus rules of action and incentives to certain productive behaviors will be necessary since the participants cannot by themselves see directly how to steer the design in a top-down systematic way.

Finally, note that the focus of this paper is on settings within a single project and neglects competition. Consequently, there are two other, longer-term issues that remain unconsidered. First, it does not look at the possible consequences of a misalignment configuration being a temporary occurrence which might migrate over multiple project generations towards a configuration that is aligned, be it in either integral/integral or modular/modular form. Second, this paper does not look into long-term competitive considerations of the value of either maintaining broader organizational architectures to better fend off attacks employing architectural innovations (Henderson & Clark, 1990), nor of purposefully changing the product architecture to actively force competitors into different modes of competition (Fixson & Park, 2008).

7 Conclusion

In this paper we study, with help of simulation experiments, the performance impact of different product development coordination set-ups in a misalignment situation of an integral product being developed by actors of a modular supply chain, a problem that is a case of a broader class of challenges when organizing for distributed innovation (Baldwin, 2012). We find that the incentives, both formal and informal, in which the system is embedded have significant implications for the project's performance. We also find that ideas from control theory were able to provide us with testable hypotheses about what alternate behaviors could improve the situation without actually removing the misalignment.

Like all research, our study has some limitations. The small sample size of the simulation runs is obvious. Another potential limitation is the focus on a single industry. It is conceivable that another industry, with different explicit and implicit standards – for example the software industry with its long tradition of releasing 'beta-versions' of its products – exhibits different project progress behavior and performance. More broadly, we see an interesting area for future research in the study of incentives in innovation operations.

References:

Argyres, N. S. (1999). The Impact of Information Technology on Coordination: Evidence from the B-2 "Stealth" Bomber. *Organization Science, 10*(2), 162-180.

Baldwin, C. Y. (2012). *Organization Design for Distributed Innovation*. Harvard Business School Working Paper 12-100. Boston, MA.

Braha, D., & Bar-Yam, Y. (2007). The Statistical Mechanics of Complex Product Development: Empirical and Analytical Results. *Management Science, 53*(7), 1127-1145.

Brown, S. L., & Eisenhardt, K. M. (1995). Product Development: Past Research, Present Findings, and Future Directions. *Academy of Management Review, 20*(2), 343-378.

Browning, T. R., & Ramasesh, R. V. (2007). A Survey of Activity Network-based Process Models for Managing Product Development Projects. *Production and Operations Management, 16*(2), 217-240.

Brusoni, S., & Prencipe, A. (2011). Patterns of Modularization: The Dynamics of Product Architecture in Complex Systems. *European Management Review, 8*, 67-80.

Brusoni, S., Prencipe, A., & Pavitt, K. (2001). Knowledge specialization, organizational coupling, and the boundaries of the firm: Why do firms know more than they make? *Administrative Science Quarterly, 46*(4), 597-621.

Clark, K. B., & Fujimoto, T. (1991). *Product Development Performance*. Boston, Massachusetts: Harvard Business School Press.

Clarkson, P. J., Simons, C., & Eckert, C. (2004). Predicting Change Propagation in Complex Design. *Journal of Mechanical Design, 126*, 788-797.

Colfer, L., & Baldwin, C. Y. (2010). *The Mirroring Hypothesis: Theory, Evidence and Exceptions*. Harvard Business School Working Paper 10-058, (10-058). Boston, MA.

Cooper, R. G. (2005). *Leadership: Pathways to Profitable Innovation* (2nd ed.). New York: Basic Books.

Dobbs, R., Oppenheim, J., & Thompson, F. (2012). Mobilizing for a resource revolution. *McKinsey Quarterly*(January).

Fixson, S. K., & Park, J.-K. (2008). The Power of Integrality: Linkages between Product Architecture, Innovation, and Industry Structure. *Research Policy, 37*, 1296-1316.

Fricke, E., Gebhard, B., Negele, H., & Igenbergs, E. (2000). Coping with Changes: Causes, Findings, and Strategies. *Systems Engineering, 3*(4), 169-179.

Gilson, R. J., Sabel, C. F., & Scott, R. E. (2009). Contracting for Innovation: Vertical Disintegration and Interfirm Collaboration. *Columbia Law Review, 109*(3), 431-502.

Gilson, R. J., Sabel, C. F., & Scott, R. E. (2010). Braiding: The Interaction of Formal and Informal Contracting in Theory, Practice, and Doctrine. *Columbia Law Review, 110*(6), 1377-1447.

Gokpinar, B., Hopp, W. J., & Iravani, S. M. R. (2010). The Impact of Misalignment of Organization Structure and Product Architecture on Quality in Complex Product Development. *Management Science, 56*(3), 468-484.

Henderson, R. M., & Clark, K. B. (1990). Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. *Administrative Science Quarterly, 35*, 9-30.

Holmes, S. (2006). Boeing's Global Strategy Takes Off *Business Week,* accessed January 9, 2012.

Jarratt, T. A. W., Eckert, C. M., Caldwell, N. H. M., & Clarkson, P. J. (2011). Engineering change: an overview and perspective on the literature. *Research in Engineering Design, 22*, 103-124.

Kratzer, J., Gemuenden, H. G., & Lettl, C. (2011). The Organizational Design of Large R&D Collaborations and Its Effect on Time and Budget Efficiency: The Contrast between Blueprint and Reality. *IEEE Transactions on Engineering Management, 58*(2), 295-306.

Loch, C. H., Mihm, J., & Huchzermeier, A. (2003). Concurrent Engineering and Design Oscillations in Complex Engineering Projects. *Concurrent Engineering: Research and Applications, 11*(3), 187-199.

Loch, C. H., & Terwiesch, C. (1999). Accelerating the Process of Engineering Change Orders: Capacity and Congestion Effects. *Journal of Product Innovation Management, 16*, 145-159.

Mihm, J. (2010). Incentives in New Product Development Projects and the Role of Target Costing. *Management Science, 56*(8), 1324-1433.

Mihm, J., Loch, C. H., & Huchzermeier, A. (2003). Problem-Solving Oscillations in Complex Engineering Projects. *Management Science, 49*(6), 733-750.

Reinertsen, D. G. (1997). *Managing the Design Factory*. New York: The Free Press.

Sommer, S. C., & Loch, C. H. (2009). Incentive Contracts in Projects with Unforeseeable Uncertainty. *Production and Operations Management, 18*(2), 185-196.

Sosa, M. E., Eppinger, S. D., & Rowles, C. M. (2004). The Misalignment of Product Architecture and Organizational Structure in Complex Product Development. *Management Science, 50*(12), 1674-1689.

Terwiesch, C., & Loch, C. H. (1999). Managing the Process of Engineering Change Orders: The Case of the Climate Control System in Automobile Devlopment. *Journal of Product Innovation Management, 16*, 160-172.

Terwiesch, C., Loch, C. H., & De Meyer, A. (2002). Exchanging Preliminary Information in Concurrent Engineering: Alternative Coordination Strategies. *Organization Science, 13*(4), 402-419.

Thomke, S. H., & Fujimoto, T. (2000). The Effect of "Front-Loading" Problem-Solving on Product Development Performance. *Journal of Product Innovation Management, 17*, 128-142.

Ülkü, S., & Schmidt, G. M. (2011). Matching Product Architecture and Supply Chain Configuration. *Production and Operations Management, 20*(1), 16-31.

Ulrich, K. T., & Ellison, D. J. (1999). Holistic Customer Requirements and the Design-Select Decision. *Management Science, 45*(5), 641-658.

Weigelt, C. (2009). The Impact of Outsourcing New Technologies on Integrative Capabilities and Performance. *Strategic Management Journal, 30*, 595-616.

Yassine, A. A., Joglekar, N. R., Braha, D., Eppinger, S. D., & Whitney, D. E. (2003). Information hiding in product development: the design churn effect. *Research in Engineering Design, 14*, 145-161.

8 Table and Figures

Table 1: Game Features Mirroring Actual Complex Product Development

Program Characteristic	**Simulation Feature(s)**
Coupled structural system: airframe	Coupled structural system: Cantilever truss (simpler, more tightly coupled than aircraft)
Several companies, each responsible for designing components	Several players, each responsible for “designing” its own section
Aircraft must meet stress requirements for flight certification	Truss must meet stress requirements to be acceptable
Pressure to deliver aircraft to market quickly	Truss orders specify latest acceptable delivery date
Pressure to minimize aircraft mass, aircraft has overall mass targets	Truss orders specify maximum acceptable mass
Global finite element stress model of aircraft, visible to OEM only	Global direct stiffness method stress model of truss, visible to integrator only
Local finite element stress models and software visible to component design teams	Local direct stiffness method models of truss sections, visible to section design players (SDPs)
Component engineers adjust component part thicknesses and geometries	Truss SDPs adjust truss member thicknesses but not geometries
OEM may institute design freezes for components or interfaces	Integrator may set upper and lower bounds for member thicknesses
OEM makes updated loads available to suppliers after each time the global model runs	Integrator makes updated loads available to SDPs after each time the global model runs
Suppliers receive incentive payments for components under mass target and incur penalties for components over mass target	Truss SDPs receive incentives for sections under mass target and incur penalties for sections over mass target
Suppliers are encouraged to collaborate with the OEM and discouraged from directly collaborating with one another	Truss SDPs may send and receive messages to and from the integrator but not to or from one another.

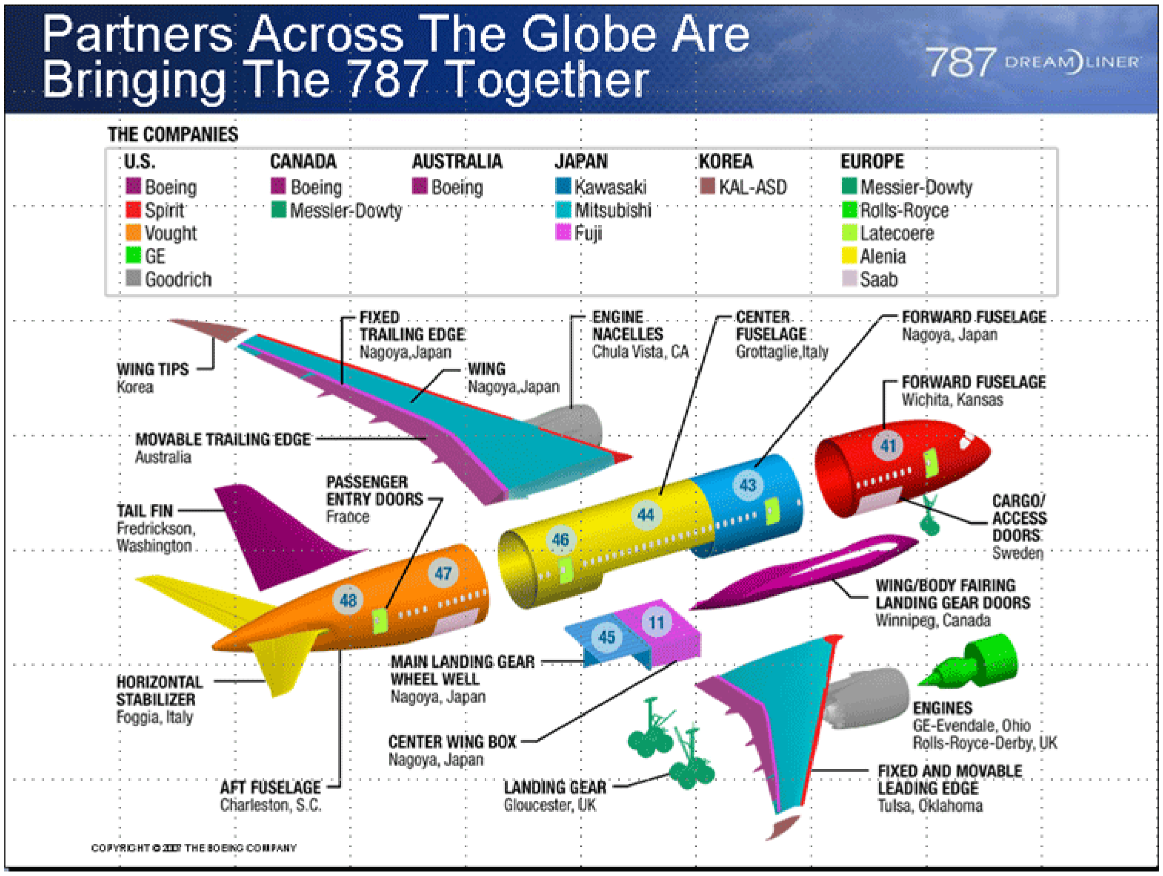

Partners Across The Globe Are Bringing The 787 Together
787 DREAMLINER
THE COMPANIES
U.S.
Boeing
Spirit
Vought
GE
Goodrich
CANADA
Boeing
Messier-Dowty
AUSTRALIA
Boeing
JAPAN
Kawasaki
Mitsubishi
Fuji
KOREA
KAL-ASD
EUROPE
Messier-Dowty
Rolls-Royce
Latecoere
Alenia
Saab
WING TIPS Korea
FIXED TRAILING EDGE Nagoya,Japan
WING Nagoya,Japan
ENGINE NACELLES Chula Vista, CA
CENTER FUSELAGE Grottaglie,Italy
FORWARD FUSELAGE Nagoya, Japan
FORWARD FUSELAGE Wichita, Kansas
MOVABLE TRAILING EDGE Australia
PASSENGER ENTRY DOORS France
TAIL FIN Fredrickson, Washington
CARGO/ ACCESS DOORS Sweden
WING/BODY FAIRING LANDING GEAR DOORS Winnipeg, Canada
HORIZONTAL STABILIZER Foggia, Italy
MAIN LANDING GEAR WHEEL WELL Nagoya, Japan
ENGINES GE-Evendale, Ohio Rolls-Royce-Derby, UK
CENTER WING BOX Nagoya, Japan
FIXED AND MOVABLE LEADING EDGE Tulsa, Oklahoma
AFT FUSELAGE Charleston, S.C.
LANDING GEAR Gloucester, UK
41
43
44
46
47
48
11
45
COPYRIGHT © 2007 THE BOEING COMPANY

Figure 1: Major supply chain partners for Boeing 787 aircraft

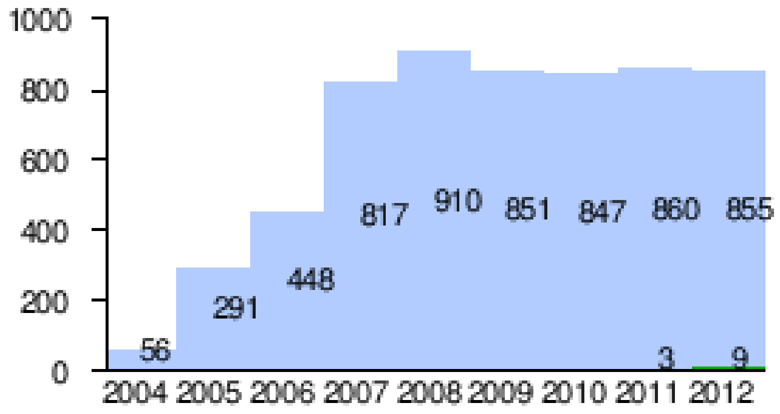


Figure 2: Boeing 787 net orders and deliveries (cumulative by year), Source: Wikipedia.com, accessed March 26, 2012

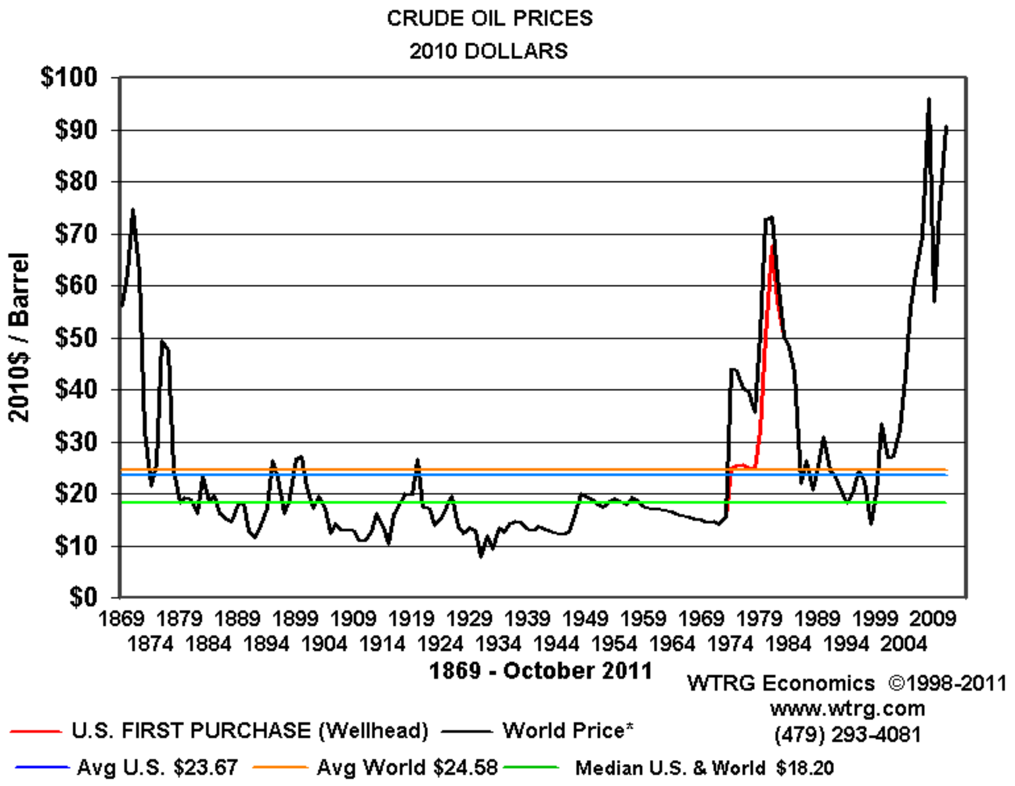


Figure 3: Oil price overthe last 140 years (in inflation-adjusted dollars)

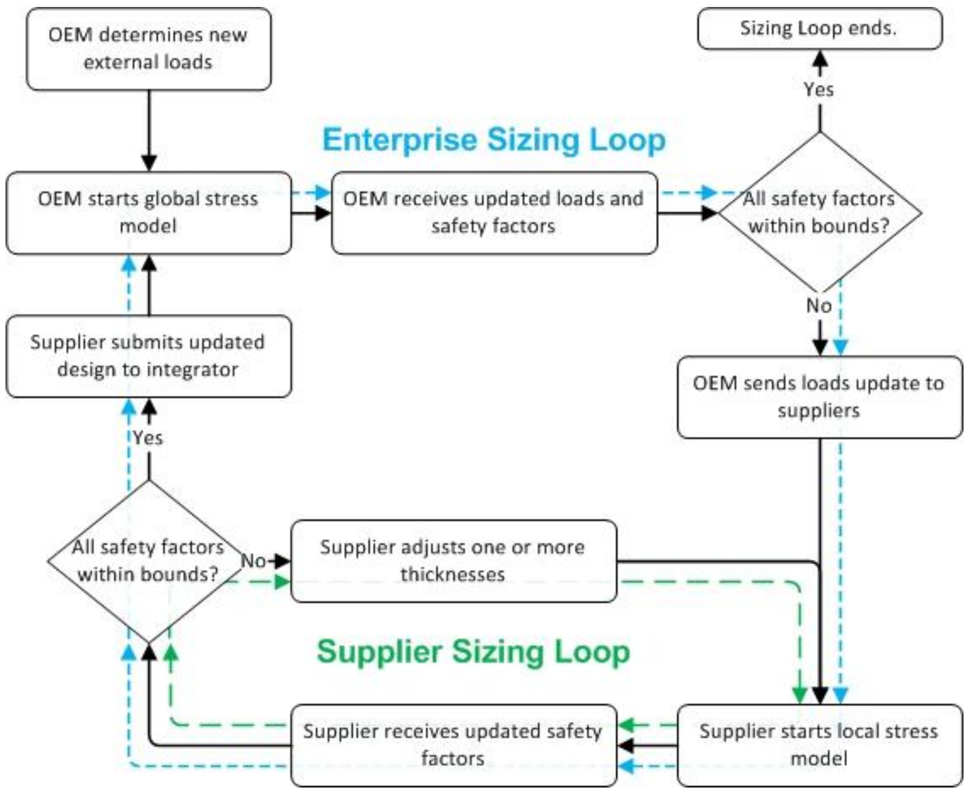

OEM determines new external loads
Sizing Loop ends.
Yes
Enterprise Sizing Loop
OEM starts global stress model
OEM receives updated loads and safety factors
All safety factors within bounds?
No
Supplier submits updated design to integrator
OEM sends loads update to suppliers
Yes
All safety factors within bounds?
No
Supplier adjusts one or more thicknesses
Supplier Sizing Loop
Supplier receives updated safety factors
Supplier starts local stress model

Figure 4: The Sizing Loop

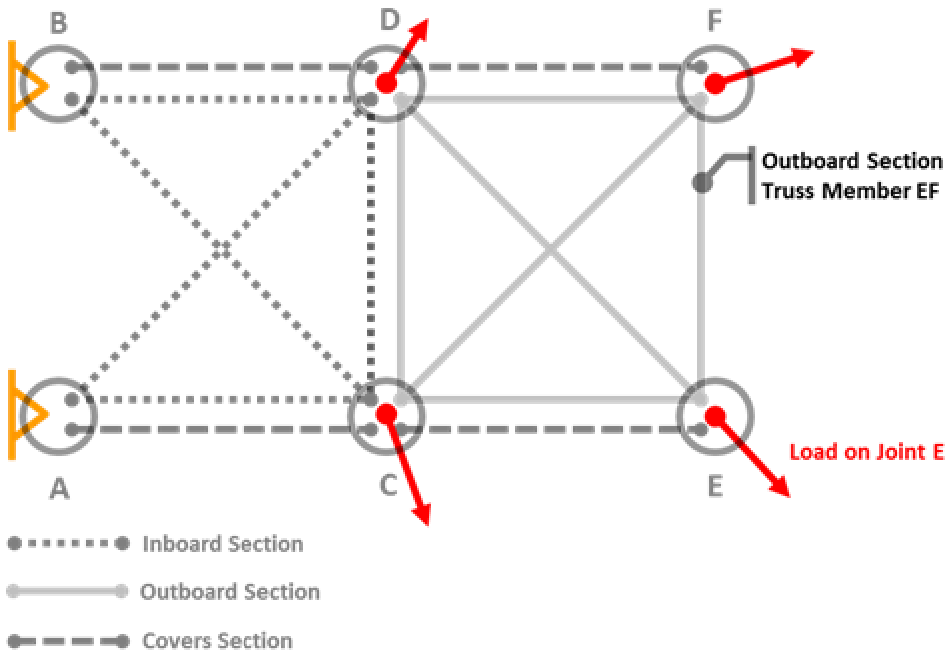


Figure 5: Design game truss

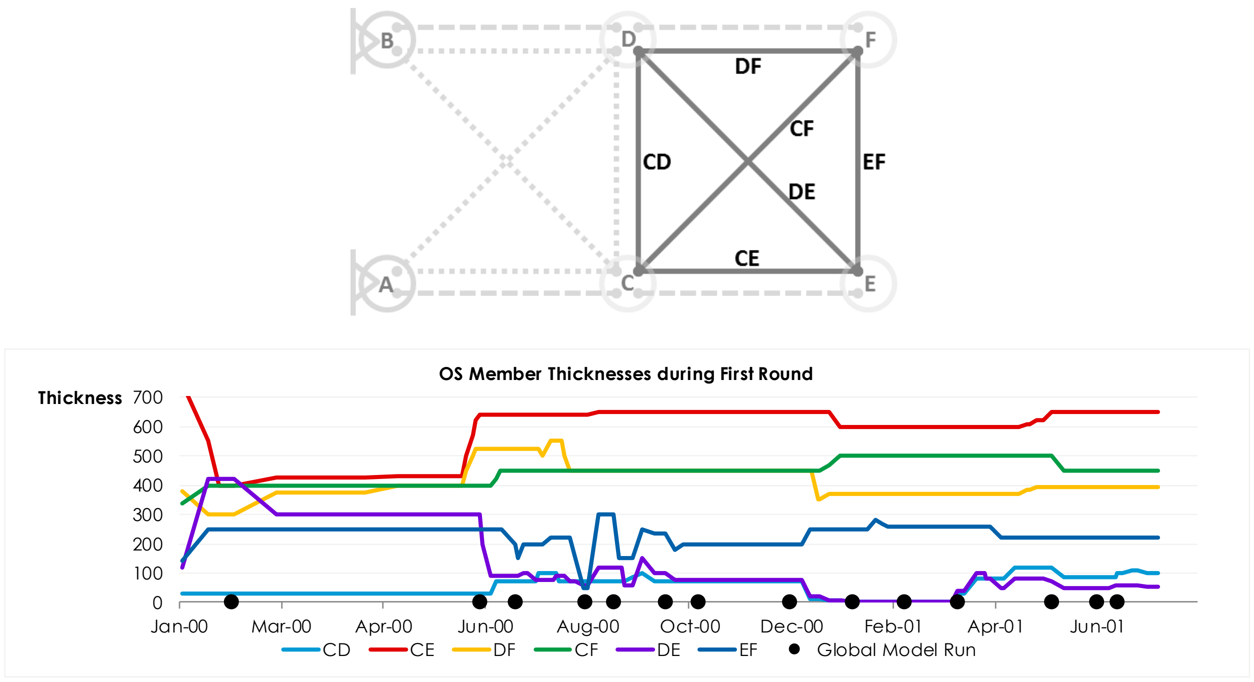
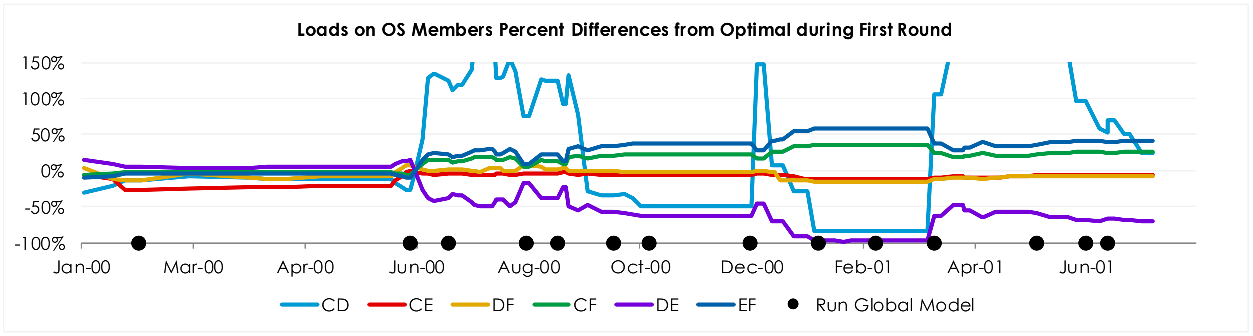


Figure 6: Real-time thickness values of outboard section truss member during first round



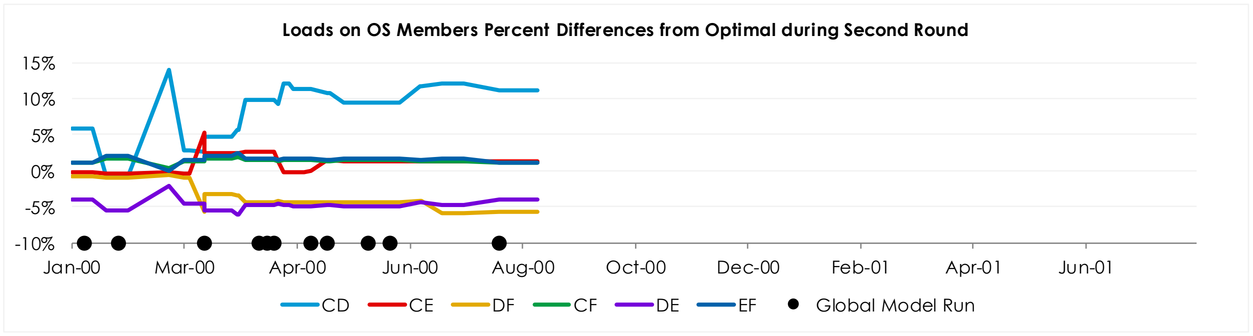
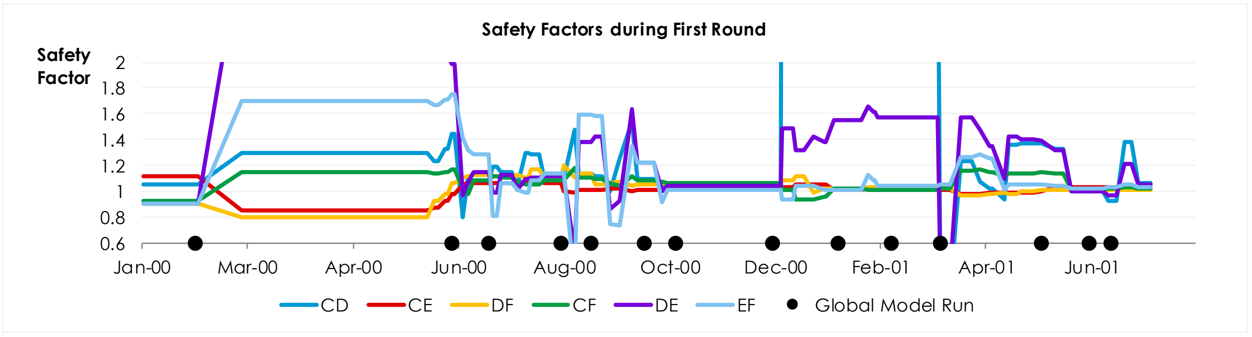


Figure 7: Loads on outboard section truss members, first round (top) and second round (bottom)



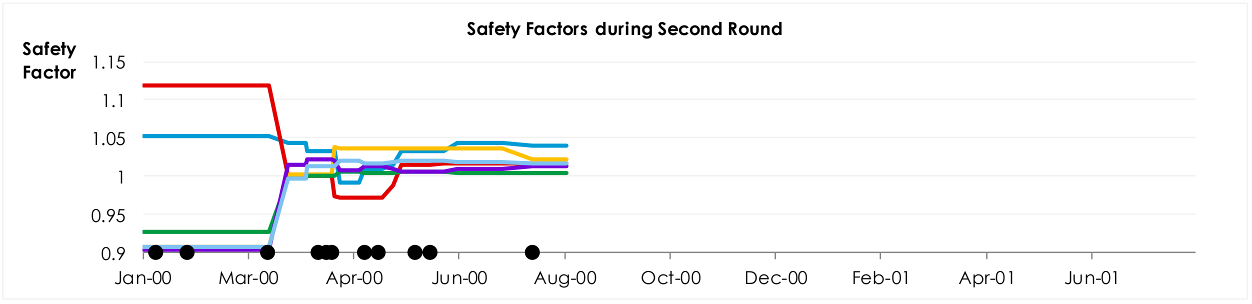


Figure 8: Safety factors on outboard section truss members, first round (top) and second round (bottom)