Description of a Turbofan Engine Product Development Process

by

Douglas C. Hague

Ph.D. Materials Science & Engineering, The Pennsylvania State University, 1995
M.S. Metals Science & Engineering, The Pennsylvania State University, 1992
B.S. Engineering Physics, The University of Tulsa, 1989

Submitted to the System Design & Management Program
In Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering & Management

at the

Massachusetts Institute of Technology

February 2001

Signature of Author__

Douglas C. Hague
System Design and Management
December 15, 2000

Certified By___

Steven D. Eppinger
General Motors LFM Associate Professor of Management Science
Thesis Supervisor

Accepted By

Paul A. Lagace
LFM/SDM Co-Director
Professor of Aeronautics & Astronautics and Engineering Systems

Accepted By

Stephen C. Graves
LFM/SDM Co-Director
Abraham Siegel Professor of Management Science

Description of a Turbofan Engine Product Development Process

by

Douglas C. Hague

Submitted to the System Design & Management Program on December 15, 2000
in Partial Fulfillment of the Requirements for the Degree of Master of Science
in Engineering & Management

ABSTRACT

This research explores what requirements are necessary for the development of a turbofan engine and how they evolve through the product development cycle. This work utilizes a parameter-based design structure matrix (DSM) to define the interfaces and interdependencies present in a large commercial aircraft propulsion system. The DSM was developed from the system level to the module level allowing one to examine the assumptions made throughout the entire life cycle of the product.

The work utilizes the system-level DSM to show the similarities between the turbofan engine product development process (PDP) and the software spiral product development process. This work examines the parameter-based DSM in each of the design phases and attempts to understand the assumptions made in each phase and how the assumptions change as the product proceeds through the development cycle. By examination of the DSM, it was found that program goals and requirements lead to an initial set of design parameters. These design parameters are then iterated until a satisfactory product definition is developed. Each stage concludes with the integration and testing of that stages work. In all stages risk management occurs and with the necessary revision of the program plan for subsequent stages (not in the system-level DSM). The work shows that the PDP for a turbofan engine can be viewed as a spiral process. The thesis then suggests that, in general, the current industry practices for the development of complex physical systems have similarity to the spiral framework for development of software.

Thesis Advisor: Steven D. Eppinger
Title: General Motors LFM Associate Professor of Management Science

ACKNOWLEDGEMENTS

I would like to thank my advisor Steve Eppinger for his guidance and suggestions in this work. I would like to thank Glenn Bartkowski for sharing his in-depth knowledge of the gas turbine engines both during the development of the DSM utilized throughout this work and during our classes as we discussed the implications and situations within the Pratt & Whitney. I would also like to thank Ronnie Thebeau for keeping us balanced within UTC. The discussions and conversations between Glenn, Ronnie and I significantly enhanced the learning in our classes.

I would like to thank those whom sponsored me within P&W. Bill Yee was instrumental for nominating me for the opportunity to participate in the SDM program. His guidance and insight while I was in the Materials Lab opened many doors for me. I would also like to thank Tom Johnson and Jim Adams for supporting me during the program. Tom allowed me to work in his Design Integration group while participating in the program. This allowed me to both learn and attempt to apply the knowledge I was learning concurrently. This taught me about the many opportunities and difficulties in attempting to apply the new knowledge within a corporation. Thanks to Jim for being so flexible with my work arrangements.

Finally, I would like to thank my wife and my sons Collin and Galen. Thanks for being patient and for allowing me the opportunity to participate in this program. Without your unfaltering (and often unacknowledged) support I would not have made it to this point. May we move into the future on the path together without so many "opportunities."

Table of Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

Product development of complex products has been studied for many years. The field of system engineering has been developed to assist in many of the technical aspects of developing a product. While many of the tools of system engineering focus on the marketing (customer) needs and how to relate them to the functions and high level requirements of a product, this thesis uses one system engineering tool to focus on the timing of assumptions made during the product development process. Finally, this work suggests that the current product development process (PDP) for turbofan engines most closely resembles the spiral product development framework that has been outlined in the software industry.

The present work was originally undertaken to further understand what requirements are necessary and how they evolve through the product development cycle, conceptual design through validation. Production, field support, and disposal are not included in this work. This work utilizes a parameter-based design structure matrix (DSM) to define the interfaces and interdependencies present in a large commercial aircraft propulsion system. The DSM was developed from the top down instead of the bottom up (i.e., from a system level rather than the subsystem level). This unique perspective allows one to examine the assumptions made throughout the entire life cycle of the product. It concludes that the assumptions made in one stage of product development, once validated, often become the requirements for subsequent stages. The system perspective allows one to take (as we say in the industry) "the 40,000 feet" view of the propulsion system. The DSM work in this thesis expands upon previous work of

Craig Rowles and Greg Mascoli [Rowles, 1999; Mascoli, 1999]. The development of the parameters and their interactions for the DSM in this work was done in cooperation with Glenn Bartkowski [Bartkowski, 2001].

Within the aircraft propulsion industry, as well as many industries, product development can be viewed as proceeding along a generic program plan [Ulrich and Eppinger, 2000]. Figure 1 shows the generic product development cycles as it exists within many industries. While this process looks linear, many planned and unplanned iteration cycles occur within the process. Desire to reduce product development time and cost have put tremendous pressure upon the product development teams not only to eliminate unplanned iteration, but also to minimize planned iteration (if not eliminate it entirely).

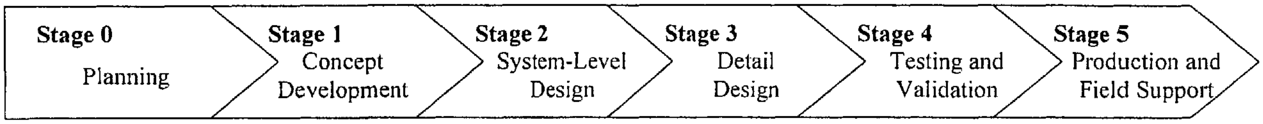


Figure 1: Generic product development process [after Ulrich and Eppinger, 2000].

This pressure has led to many techniques and methods that shorten the generic program plan. Among these are concurrent engineering, integrated product teams, integrated computer master models, and reordering tasks for more optimal information flow. Each of these allows for the shortening and overlap of the generic program plan given in Figure 1. When attempting to put the generic PDP into a framework, the process is generally described as a waterfall product development process (Figure 2) [Boehm, 1988]. However, this work examines the PDP for a turbofan engine and it concludes that a second framework for product development more closely resembles the turbofan PDP. This second framework has been termed a spiral development process due to its conceptual graphic (Figure 3) [Boehm, 1988; Rechtin and Maier, 1997]. In a spiral development process, the product is developed in every increasing complexity by progressing through a set of tasks very quickly and doing so several times. For each spiral around the process, the product becomes more complex and has more functionality. By validating the product in small increments, issues and errors within the product design are caught early and only small amounts of rework (iteration) are necessary to correct the issues. The reduced rework leads to significantly shorter development time and lower cost (as well as mitigating risk).

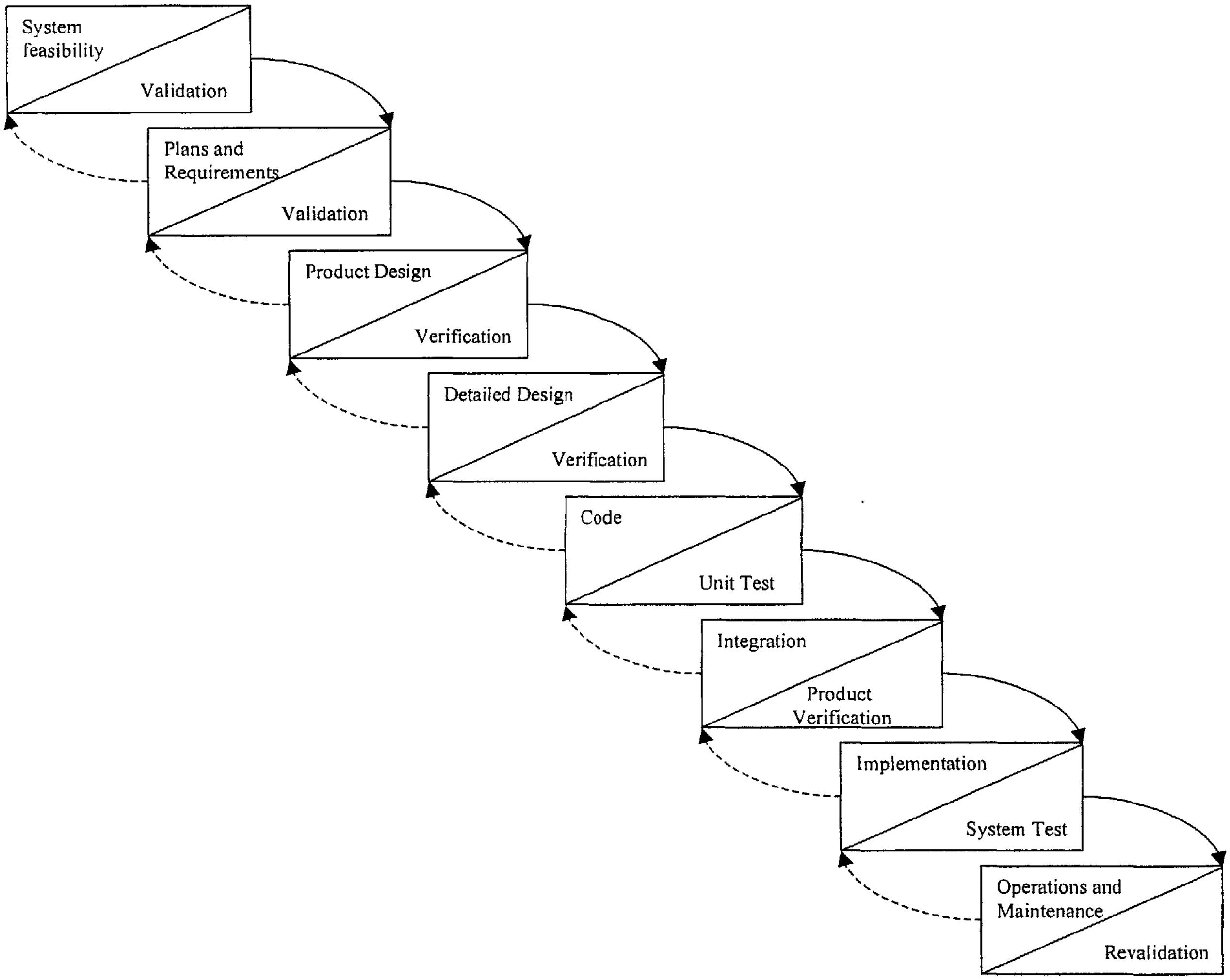


Figure 2: Schematic waterfall development process [Boehm, 1988].

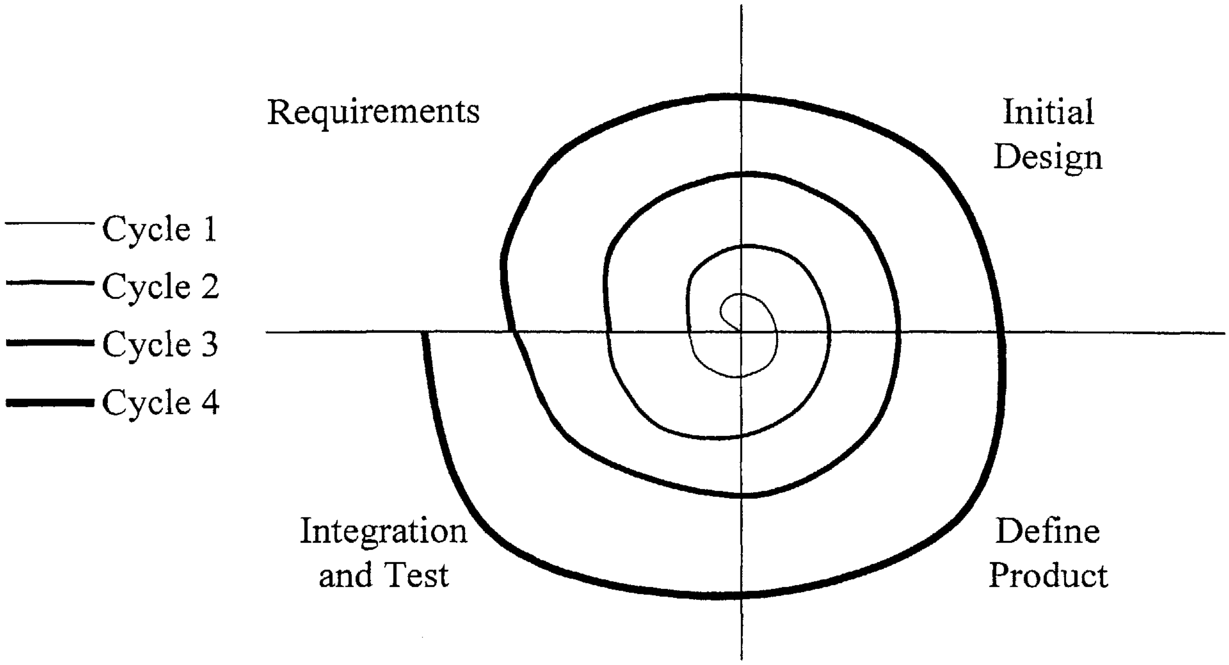


Figure 3: Schematic spiral development process with 4 cycles [Boehm, 1988; Rechtin and Maier, 1997].

Two primary differences exists between a waterfall PDP and a spiral PDP. First, the product requirements in a waterfall development process are set very early in the process and should not change throughout the process. In a spiral process, the requirements are more flexible. Exactly how flexible is not clear in the literature. Second, the spiral process emphasizes risk analysis and mitigation. There is no such emphasis in the waterfall process.

The current work utilizes the system-level DSM to show the similarities between the turbofan engine product development cycle and the spiral product development process. This work examines the single DSM in each of the design phases and attempts to understand the assumptions made in each phase and how the assumptions change as the product proceeds through the development cycle. A process called tearing is used to examine the assumptions in the DSM. Tearing a DSM is the process of making an assumption and then reordering the parameters based upon the assumption. By ordering the tears according to how information is developed in a particular stage of the product development process, each stage may be examined as to how and what is occurring during the phase.

By examination of the information flow between parameters in the DSM and the iteration that occurs in a stage, it was found that the information flows from the determination of program goals and requirements to an initial estimation of the design to an iterative cycle of product definition. Each stage concludes with the integration and testing of that stage's work. In all stages, risk management occurs and the necessary revision of the program plan is developed for subsequent stages. The work shows that the PDP for a turbofan engine can be viewed as a spiral process. The thesis then suggests that that, in general, the current industry practice for the development of a complex physical system is equivalent to the spiral framework for development of software. This work also develops specific recommendations for Pratt & Whitney as well as implications for general product development processes.

CHAPTER 2: BACKGROUND

Product Development (faster/better/cheaper)

As a market becomes more competitive and as the "clock speed" of an industry increases, the product development process is put under pressure to develop improved products in less time for less investment. This push towards faster/better/cheaper has led to the adoption of concepts such as concurrent engineering and product development process optimization. The design structure matrix technique is one of many tools that may be employed to improve the PDP through optimal ordering of the tasks and improved information flow. Given the large push to make drastic improvements, many changes are made without full recognition of their impacts and interactions with other portions of the product development cycle. Such blind reduction of the process without understanding of the interfaces and the interdependencies can ultimately lead to failure. One example of this was the Mars Polar Lander developed by Lockheed Martin for NASA. This failure was directly attributed to the push for development time and cost reductions [Dornheim, 2000]. Failures such as the Mars Polar Lander probably occur much more frequently than documented in public literature simply because companies are extremely reluctant to put any failure into the public arena. Under such an environment, product development management must continually push to improve their process, but must do so by developing a deep understanding of the interdependencies of the process and the risks they accept when pushing to achieve their goals.

Product Development at Pratt & Whitney

Pratt & Whitney (P&W) develops, manufactures and supports a wide range of gas turbines. The current work will focus on a gas turbine for large commercial aircraft. A schematic of a propulsion system for a large commercial aircraft is given in Figure 4. Traditionally, the overall system has been decomposed into nine major subsystems (fan, low pressure compressor, high pressure compressor, diffuser/combustor, high pressure turbine, low pressure turbine, mechanical components, externals/controls, and nacelle). The combination of the gas turbine (the first eight subsystems) with the nacelle is termed a propulsion system.

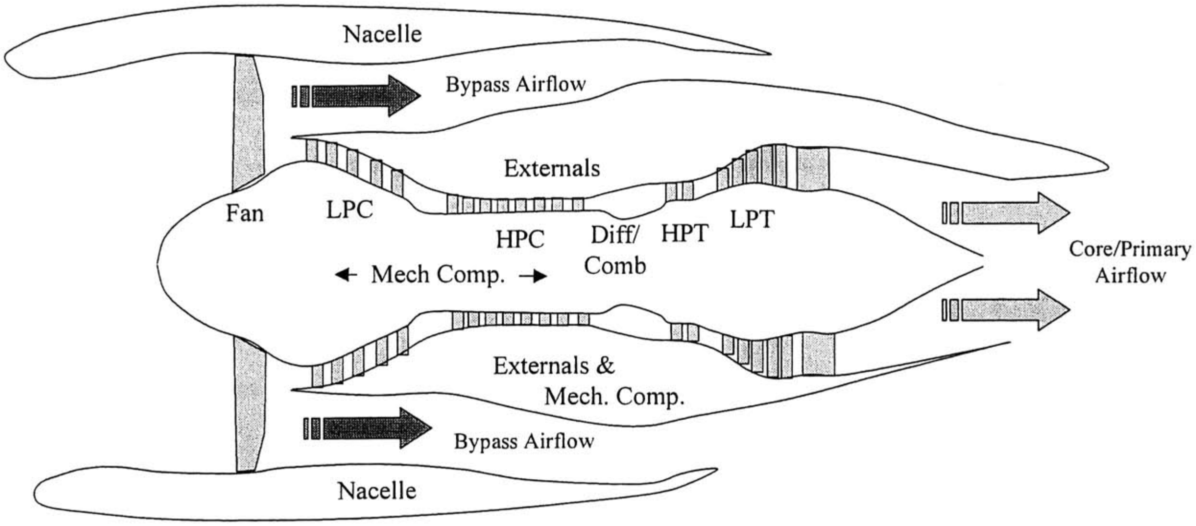


Figure 4: Schematic large commercial gas turbine. Subsystems are identified within the drawing.

A gas turbine provides propulsion to an aircraft by significantly increasing the momentum of air that goes through the engine. The propulsion systems accomplishes this by first compressing the ambient atmosphere, fuel is then added to the air and burned in a continuous manner to greatly accelerate the air. The gasses then pass through a turbine that turns the compressor stages. The hot gas then exits the core of the engine and provides thrust to the aircraft. This gas is termed the core or primary airflow. For large commercial aircraft, the thrust from the hot air exiting the core of the engine is only a minor portion of the overall thrust. The majority of the propulsion is provided by having the fan accelerate a large amount of air by a small amount. This is termed the by-pass air. (To increase the momentum of air, mass*velocity, one may significantly increase the velocity of a small mass of air or slightly increase the velocity of a large mass of air. For reasons beyond this thesis, large commercial engines choose to do the latter [Smith and Mindell, 2000]).

From the early development of gas turbines through the early 1980s, a product development cycle within the commercial gas turbine industry typically lasted about 5-6 years [Epstein, 2000]. This length of time was from "launch" of the product (start of detailed design), to certification of the engine by the United States Federal Aviation Administration (FAA) or the European Joint Airworthiness Authority (JAA). Due to the drive for better products in a more timely manner, P&W implemented process improvement initiatives to reduce the product development time to 2.5 years by the year 2000. P&W strove to accomplish this while reducing the development cost by 70%. These reductions were undertaken to increase the attractiveness of P&W engines to our customers. Integrated Product Development, integrated 3-D modeling, alignment (and eventual merging) of manufacturing and engineering organizations, and the development of a system engineering organization are among the initiatives implemented at P&W in support of these goals.

These initiatives eliminate wasted effort and significant amounts of rework within the product development cycle. The philosophy within the engineering and product development leadership was that the product development cycle should be replaced by a product validation cycle. P&W was to validate designs developed using the software tools implemented through the process initiatives listed above. As P&W transitioned from a development company to a validation company, there was recognition that a minimal set of development tests would be necessary. However, extreme pressure from management was put on any single test to either eliminate it or minimize the amount of work necessary. Of course the design teams would push to include as much testing as possible in order to validate their software tools and designs prior to FAA/JAA certification tests (i.e., validation) were failure is unacceptable. This tension between the management and the design teams is necessary and can in fact serve to balance the process. When one of the sides develops too much power, there is no balance and the process suffers. P&W recognized this tension and initiated the formation of a system engineering organization in 1997 to help maintain this balance. For detail discussion of the organizational transformation at P&W see the following references: Womak and Jones, 1996; Rowles, 1999; Mascoli, 1999; Glynn and Pelland, 2000.

System Engineering at P&W

System engineering at P&W was formally instated in 1997. At that time a "three-legged stool" was put in place with three system engineering organizations representing the three legs. The three organizations are Propulsion System Analysis (PSA), Product Development & Validation (PD&V), and System Design & Component Integration (SD&CI). The organizations were developed such that each had specific responsibilities and that together, they were responsible for the technical (engineering) side of product development and field service. The current organizational structure is shown in Figure 5. (Note that the true organizational structure cannot be drawn as neat boxes in a figure and that full understanding of the organization is only developed with experience, if ever). Prior work of Mascoli, 1999 and Rowles, 1999 has the perspective of the IPTs in the organization while the current work's perspective is from the system engineer (two very different parts of the organization).

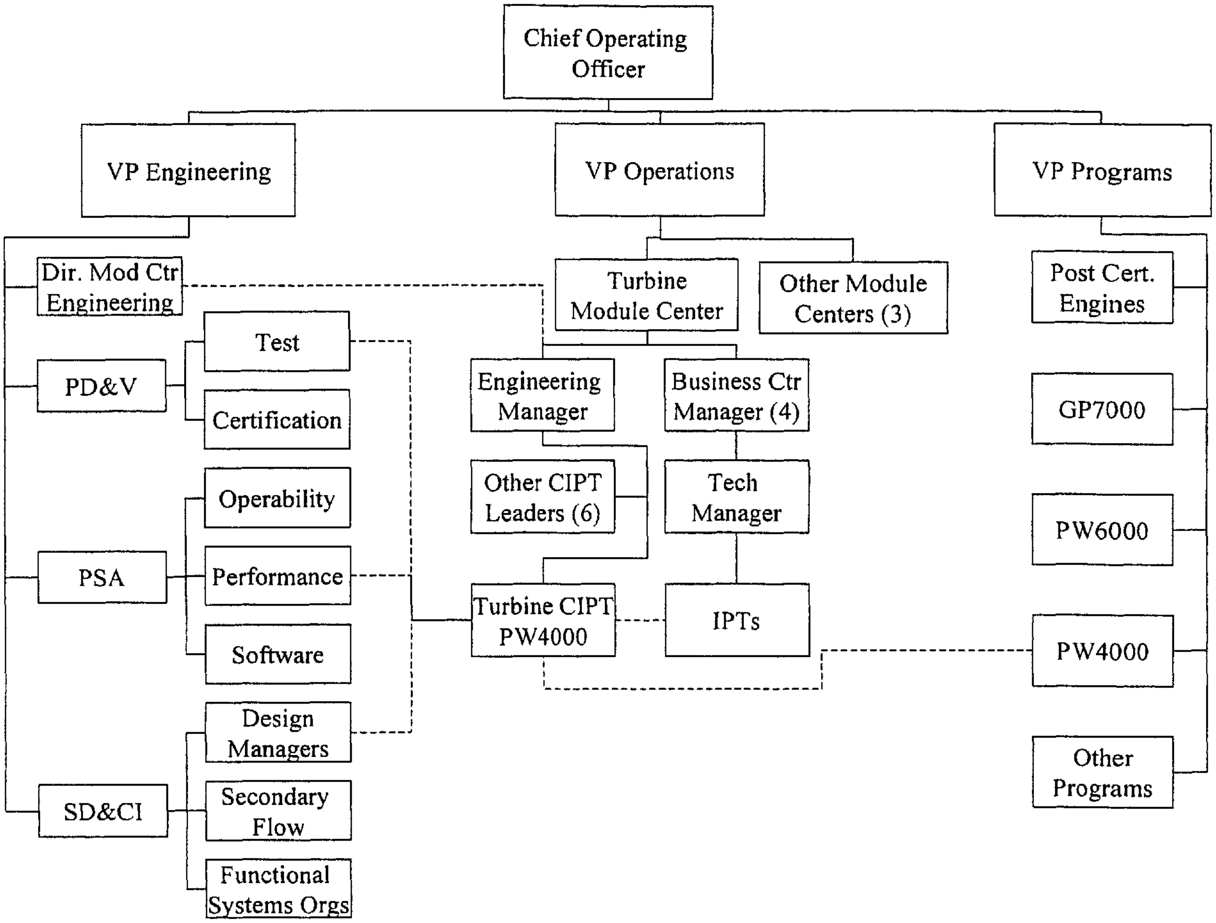


Figure 5: Product development organizational structure at P&W.

PSA has the responsibility for engine (system) level parameters. They own the simulations and models for the aerothermal cycle analysis of the engine as well as analysis for such integrative parameters as noise. In addition, PSA is responsible for the software that controls the engine. This system engineering organization is primarily responsible for the performance and operation of the engine. The PSA organization owns the "primary" gas flow in the engine.

The PD&V organization is primarily responsible for the assembly, testing, and certification of the engines. They are by far the largest of the system engineering organizations in that they include many assembly mechanics and test engineers. The PD&V organization develops and maintains the engine validation plan (EVP), formerly termed the engine development plan (EDP). This plan is developed within the overall schedule and resource constraints of the program (i.e., product development cycle). This organization is constantly attempting to balance the multi-project problem with limited physical resources (test cells).

The SD&CI organization is by far the smallest organization in terms of people, but they have the responsibility for leading and integrating the entire physical design of the engines. In effect they have the responsibility, but not the direct supervisory link, for the entire design team and all the IPTs that design the individual components. The SD&CI organization owns the requirements and the configuration of the engines. Also within SD&CI are groups responsible for the secondary (as opposed to primary) flow within the engine, static and dynamic structural considerations at the system level (e.g., fan blade out stress loads and engine vibration). These later groups are a very functional organization while the requirements, configuration, and system integration portion of SD&CI is very program focused.

Requirements Documentation within P&W

The documentation of requirements for a propulsion system is the responsibility of the SD&CI organization. The process for this has been evolving over the last 5 years to facilitate integrated product development (IPD) process (Figure 6). The internal IPD process and system level procedures developed for ISO9000 outline the expectations for a Propulsion System and Components Requirements Document (PS/CRD). The Propulsion System Requirements Document (Chapter 1 of the PS/CRD) is developed during the preliminary design phase and is approved upon completion of the preliminary design review, termed a Level II Review. Thus, system level parameters and their associated requirements are developed early in the product development cycle. Many of these requirements are developed in Advanced Engine System Engineering organizations using a detailed aerothermal simulation and other system level models and heuristics developed over the last 50 years within P&W.

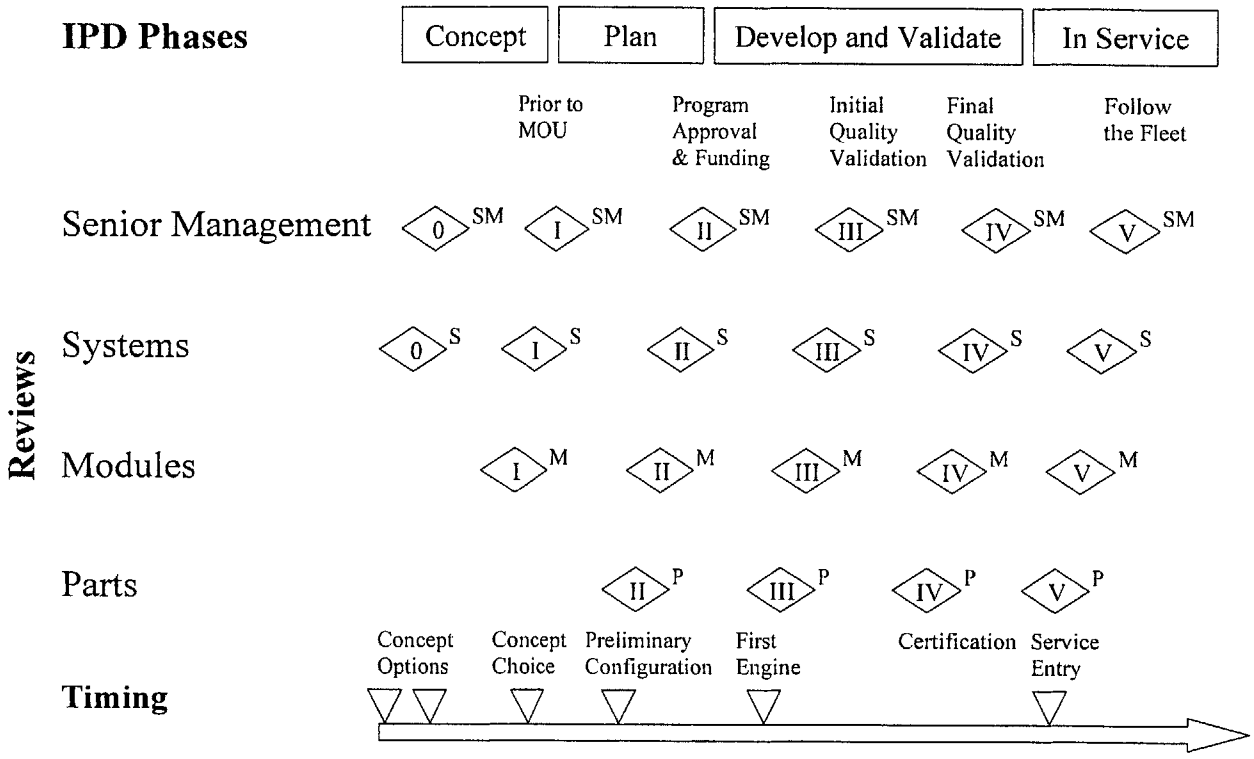


Figure 6: P&W integrated product development process [Anonymous, 2000].

Following a successful Level II Review, senior management launches product development cycle and detailed design begins. At this point the responsibility for the system engineering is handed from the Advanced Engine organizations to the program specific PSA, PD&V, and SD&CI organizations. This handoff is the general case for new engine designs (about one every 10 years). For derivative designs within a product family, the program specific system engineering organizations are responsible for the conceptual and preliminary design as well as the detailed design, validation and field support.

One of the first tasks after launch is the development of the Component Requirements Document (Chapter 2 of the PS/CRD). Each major subsystem (fan, LPC, HPC, diffuser/combustor, HPT, LPT, mechanical components, externals/controls, and nacelle) develops the detailed requirements that will enable the subsystem (modules as they are called P&W) to meet the requirements laid out in the Propulsion System Requirements. A review and approval of the Component Requirements occurs at a Level III Design Review.

Following the development of the detailed module requirements, the detailed engineering design begins. (This is sequential only in an idealized world. In actuality, the detailed design and development of requirements happen concurrently). During the detailed design, the requirements for the manufacturing of the parts are developed and when rolled together at the module level become sections of Chapter 3 in the PS/CRD. Review of the detailed design and the manufacturing requirements occurs at Level IV reviews in each integrated product team (IPT). During the detailed design of a component, it is common practice to have 3-5 designs for a specific component. These various designs define the window of design space the component will occupy. During the design and validation of the components and engine, the design is narrowed to a final design that is then released as the production design. There is a constant tension between the desire of manufacturing to have the final design to produce and the desire of the design engineers to maintain as wide a design window for as long as possible.

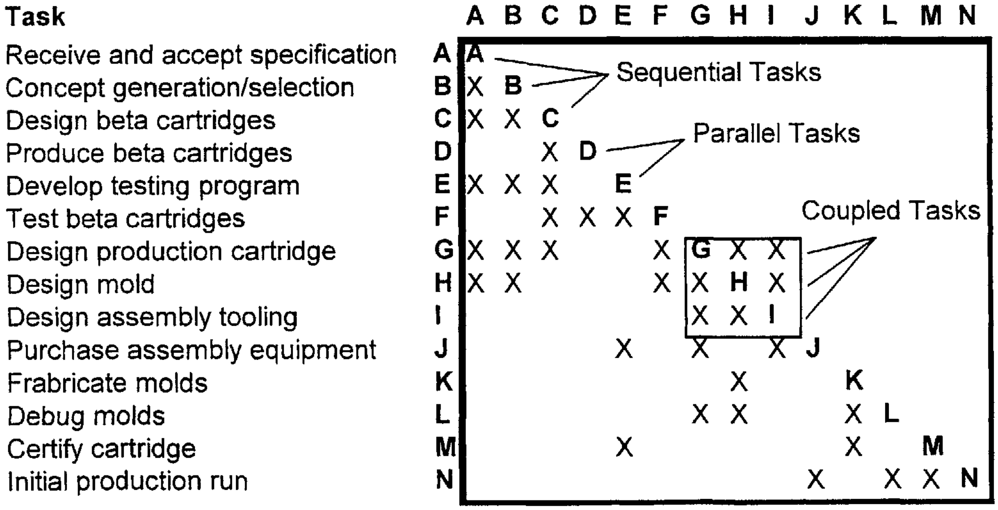
As the components and modules are produced and assembled into an integrated engine, the verification of each requirement should occur during the validation phase of the product development cycle. As gaps are identified between the actual engines and the requirements, individual recovery plans are put in place. If the gaps are in system level parameters, the system engineers usually lead a team of module level integration leaders (Deputy CIPTs) to develop and implement the recovery plan.

While the procedure for developing a PS/CRD and its relationship to IPD has been in place, it is written as a suggested process and its implementation has varied widely. Of particular note is the PS/CRD for derivative products. For derivative products, the PS/CRD has been implemented as a "what has changed" or a "what is at risk" instead of as a baseline for a complete design. Often PS/CRD documents do not exist for the baseline design (the original designs were completed prior to the development of the PS/CRD methodology and no effort has been made to retroactively develop comprehensive requirements). Thus, for several platforms within P&W only highly annotated sets of requirements exist. This methodology of only developing requirements for high risk areas is similar to the suggestions for a spiral development process [Boehm, 1988]. However, as will be discussed later in this thesis, this practice carries the risk of omission of a critical interface or requirement. For highly interdependent products, this may not be the most preferred technique.

Design Structure Matrix

The design structure matrix (DSM) is a system engineering tool that graphically represents the information flow and interdependencies among a set of items. The items can be tasks, parameters, teams or another set of related items. The DSM is a square matrix with the items listed both down the column and across the rows. An "X" or other mark at the intersection of the row and column indicates interdependency between items. The DSM analysis attempts to develop a preferred order to accomplish the items. The convention is that when completing items, a team starts at the top and works down. Note that in the convention used throughout this thesis, the marks below the diagonal are feedforwards and the marks above the diagonal are feedbacks. The following example DSM is taken directly from Ulrich and Eppinger, 1995.

In Figure 7, Task A is not dependent upon any other task and can be completed first. Task B is dependent upon Task A and therefore Task B must be completed following Task A (i.e., they are sequential tasks). Since Task E is not dependent upon Task D, they may be accomplished concurrently (i.e., they are parallel tasks).



Task		A	B	C	D	E	F	G	H	I	J	K	L	M	N
Receive and accept specification	A	A													
Concept generation/selection	B	X	B												
Design beta cartridges	C	X	X	C											
Produce beta cartridges	D			X	D										
Develop testing program	E	X	X	X		E									
Test beta cartridges	F			X	X	X	F								
Design production cartridge	G	X	X	X			X	G	X	X					
Design mold	H	X	X				X	X	H	X					
Design assembly tooling	I							X	X	I					
Purchase assembly equipment	J					X		X		X	J				
Frabricate molds	K								X			K			
Debug molds	L							X	X			X	L		
Certify cartridge	M					X						X		M	
Initial production run	N										X		X	X	N

Figure 7: Example Program DSM of Kodak Cheetah Project [Ulrich and Eppinger, 1995].

For Task H, information from items A, B, F, G, and I are necessary. In this case, Task G cannot be completed until Task H and I are completed; however, Tasks H and I cannot be completed until Task G is completed. Thus in all likelihood, iteration will be necessary when completing G, H, I. These are termed coupled tasks. Completing a set of coupled tasks always involves making an initial set of assumptions. In this example, Task G can be accomplished by first guessing the output of H and I. Next H could be completed with the information from G and an assumption about I. The designer would then have a choice on whether to go on to Task I or to check the assumption made in G about the output of H. If the assumption was too far off the output, the designer may want to iterate on G prior to completing I. In this methodology, how to complete coupled tasks is left up to the development teams.

A second methodology to completing a set of coupled tasks is to make the assumptions a requirement. In this way, items completed later are actually forced to fit the initial assumption. While this may not be ideal (or even possible), this practice is fairly prevalent in industry, especially for items that have a very long feedback loop.

Reordering of items within a DSM has been the subject of much research. Algorithms for clustering (reordering into groups of interdependent tasks) and partitioning (reordering in an attempt to have all the dependencies below the diagonal to eliminate coupling) have been developed [Fernandez, 1998; Thebeau, 2001; Steward, 1981]. The method that allows one to make assumptions on the outputs of items and then reorder the DSM is called tearing and partitioning [Steward, 1981]. Repeated tearing and reordering will eventually result in all unassumed dependencies lying below the diagonal. In this manner, by making the assumptions given, the DSM can be completed without iteration (if all assumptions are valid). In Figure 7, the three interdependencies above the diagonal would all have to be assumed. There is no reordering of this DSM that would reduce the feedback loops above the diagonal. The current work uses tearing extensively. An analysis of what parameters are assumed when and the resultant progression through parameters is used to develop the main conclusion in this thesis.

Assumptions can be made for two reasons. First as discussed above, assumptions can be made to break a set of coupled tasks. Second, an assumption may be made to reduce the product development time. Returning to the prior example in Figure 7, if the output of Task K could be assumed, then Tasks J, K, L and M could be accomplished in parallel (resulting in a reduction in the overall product development time). Thus, by making selective assumptions throughout an entire set of tasks, product development time can be reduced. Of course, this assumes that the development team can develop a good set of assumptions that do not force rework at a later time. Good program managers are aware of these relationships and will make the appropriate trades of time vs. rework probability. (For significantly more detail on risk management using the DSM, see the work of Browning, 1998).

DSM Research

The interfaces and information flow within a product development cycle have been studied by research groups lead by Eppinger and Whitney over the last 10 years [Pimmler and Eppinger, 1994; Eppinger *et al.*, 1994; Cronemyr *et al.*, 1999; McCord and Eppinger, 1993;

Smith and Eppinger, 1997; Whitney *et al.*, 1999]. Their work expanded the work of Steward on the DSM [Steward, 1981]. By defining the information flow between teams, parameters, or tasks, the DSM graphically represents the product development cycle. Partitioning and clustering of tasks within the DSM has lead to many suggestions for improvements concerning program planning (concurrent engineering and minimization of iterations and risk), product architecture, and organizational structure. The DSM has been utilized in many research groups to study a wide variety of products and is now being developed for use within various industries (see Second MIT DSM Workshop Program, Sept. 2000 [MIT DSM, 2000]).

The DSM is a graphical method for showing iteration and information flow between a series of items. Four main types of DSMs have evolved [MIT DSM, 2000]:

Team-based: where the items of concern are teams and information flows is passed between teams and through the organization.

Component-based: where the items are physical components of product architecture and the relationships are material, energy, spatial, structural, and/or information.

Activity-based: where the items are tasks to be completed within an overall program plan and the interactions are precedence relationships.

Parameter-based: where the items are variables or design points within a product and the interactions are often mathematical relationships between the parameters.

Each of these types of DSMs has specific benefits and may even be combined to show alignment between the various structures present within a companies product development process [Rowles, 1999; Sosa *et al.*, 2000].

The technique of using a DSM has been extensively studied in industry-university relationships. Although much of the initial work utilized automotive examples, various industries have now been involved in the academic research on DSMs. The latest conference on DSM provides some perspective on the applicability of this technique. Industries involved included automotive, aerospace, construction, and telecommunications. The primary link in all these industries is that they have a complex product (i.e., a product with many interfaces and interdependencies of the subsystems). Many large DSM systems are now being developed. For example, Boeing has a large, interlinked DSM that contains around 1000 parameters [Whitney *et al.*; Stowe, 2000]. While the process group has expended a large effort to develop this DSM, it has yet to gain general acceptance within the company and their program managers [Stowe, 2000]. Another interesting application comes from the construction industry for program planning [Huovila *et al.*, 2000]. Here a company has developed a generic DSM (program plan) for the construction of a building. For each individual project, they spend a small amount of time altering the generic DSM for the specific building on that project. In this manner they have been able to streamline the planning and construction of a building.

DSM within P&W

Pratt & Whitney students in the System Design and Management program at MIT have completed three Masters theses that utilized DSM. Craig Rowles examined component-based and team-based DSMs to determine the degree of alignment between the physical architecture and the organization. Greg Mascoli developed module (subsystem) parameter-based DSMs and combined them into a system level DSM to provide a view of the architecture of a gas turbine. Steve Glynn and Tom Pelland jointly examined the work of Mascoli and Rowles to determine how information flows through Pratt & Whitney. The DSM developed later in this thesis attempts to examine Pratt & Whitney's architecture from the perspective of the entire propulsion system. This present work differs in that it takes a top down methodology instead of a bottoms up methodology that was used in the prior Pratt & Whitney students work.

Rowles developed an IPT (organizational) DSM as well as a physical component (architectural) DSM [Rowles, 1999]. The overlay of these matrices lead to several conclusions and recommendations on the organizational structure and integration aspects of the product development cycle. Detailed analysis and refinement of the hypotheses were completed to further understand the data and the second order interactions [Sosa *et al.,* 2000]. Through Rowles' work, it was found that indirect interactions between components and teams are an important part of the propulsion system development and the system integration efforts were required to facilitate the interactions between large subsystems.

Mascoli approached the same development process from the parameter point of view [Mascoli, 1999]. He developed a large parameter-based DSM that showed that the system architecture and the organization were aligned. The level of refinement in Mascoli's work was below the system level, but above the large subsystem level. Thus, he showed that the gas turbine engine can be decomposed into major subsections, but that there were many "system" level parameters that did not fit well with any subsystem. In fact, many research projects have found "system" level parameters/teams/components that are usually put at the top or bottom of a DSM. Following the segregation of the system parameters, the remainder of the DSM is partitioned independently. Similar to Rowles, Mascoli recommended that the system integration efforts be strengthened between the subsystem organizations.

Pelland and Glynn showed how the recent organizational restructuring has traded system integration effectiveness for design-for-manufacturing effectiveness. While the communication and information transfer between modules has been made significantly less likely, the manufacturing and design engineers are communicating well. This trade presupposes that the system integration can still be effectively managed by the system engineers. They recommended that a "smart" DSM be developed by the subsystems technical discipline chiefs and managed by one of the system engineering organizations. In this manner, the flow of information between IPTs and across modules could be managed with minimal wasted effort.

CHAPTER 3: SYSTEM LEVEL PARAMETER DSM

Development of the DSM

As one of the goals of this work was to approach the product development and interface management from a different perspective than prior work, the DSM developed in this work took a "system" level perspective. In this statement it is meant that the parameters used in the development of the DSM are those most often used and thought of within the system engineering organizations. Although the system engineers often delve into subsystem parameters, the objective was to view the subsystems (modules) defined in prior work as black boxes and to focus on the integrative, highly dependent, variables that define a gas turbine engine and its cycle. While prior work encompassed the system, the perspective was from a component or module level. This work utilized PW4000 engines as the basis for the DSM. The development of the DSM was accomplished cooperatively with Glenn Bartkowski [Bartkowski, 2001].

To obtain the system level perspective, interviews were conducted primarily with the system engineers within three system engineering organizations in P&W. The lowest level person interviewed for the production of the DSM was a Deputy Component Integrated Product Team Leader (Deputy CIPT), who is responsible for an entire module and many integrated product teams (IPTs). This contrasts with prior work where the primary focus was at the IPT level [Rowles, 1999; Mascoli, 1999]. This is shown schematically in Figure 8 of the P&W organizational structure. Note that there is some overlap in the people used, but the people used in this work are primarily referred to as design experts and system engineers in the prior work.

Rowles used "design experts" to develop his architectural DSM. The specific employees that Rowles used were what are now referred to as discipline chiefs within the Module Centers. Rowles then utilized several systems engineers to validate the work of the discipline chiefs. While this may appear to be similar to the present approach, Rowles work was developed from the Module Center perspective and then checked by the system engineers. While subtle, it is different than the present work. Upon detailed comparison of Rowles work with the present work, the identical situation is present when Mascoli's work is compared to the present work. In both cases, ~10 of their "system" parameters have been expanded into ~100 parameters while the majority (60-100) of their parameters have been collapsed to 10 parameters. The work of Glynn and Pelland on the communication flow included the majority of the product development organization [Glynn and Pelland, 2000].

The parameters to be used in the DSM where obtained from written sources and from the system engineers experience. The types of documents that were used were all proprietary documents within P&W and are as follows: Design tables, internal training manuals, product reviews, FAA certification reports, Model Integrated Product Team reviews, airframe manufacturers requirements documents, and Chief Engineer's meeting presentations. Current PS/CRDs were specifically not used to define the list of parameters such that a later comparison could be made to determine what overlap and omissions were present between the documents.

The list of variables was developed and reviewed by system engineers within System Design & Component Integration and Propulsion System Analysis (two of the three system engineering organizations within P&W). The entire list was then grouped and narrowed to focus the list on what was considered the variables that would define the gas turbine engine to a knowledgeable outside expert. A list of 110 variables was developed.

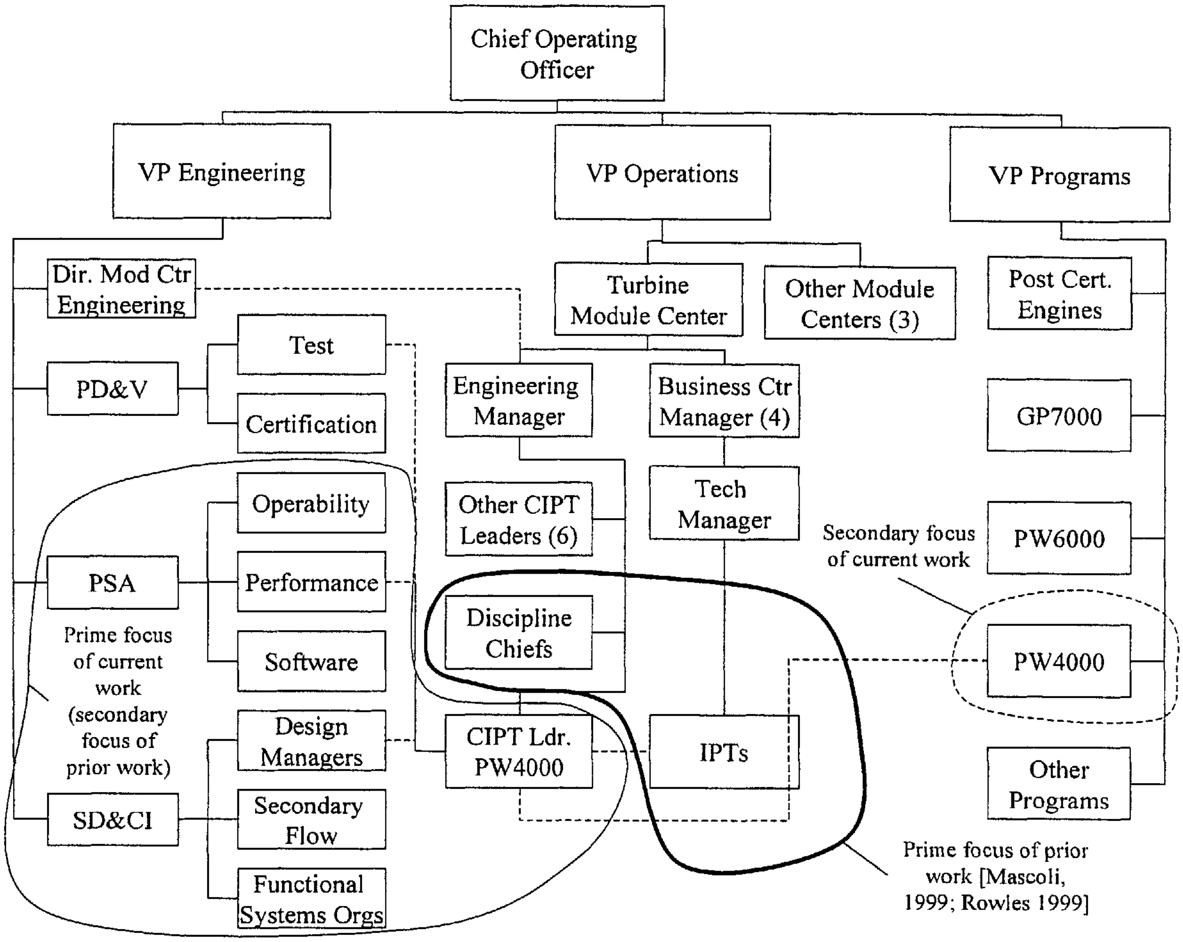


Figure 8: Sets of the P&W organization utilized in various work.

The method that was used to develop the interactions was to have the system engineers and other design experts in a room 2 or 3 at a time and to project the DSM onto a large screen from the computer. The design relationships were then discussed utilizing the particular expertise and knowledge of the system engineers in the room. Discussion between the engineers proved to be one of the most useful aspects of accomplishing the development of the interactions in this manner. The development of the list of parameters and their interactions was accomplished in a series of one-hour meetings. These meetings occurred 2-4 times per week over a three-month period. This practice is different from most DSM work where one-on-one interviews and surveys are sent to experts. By using a participatory methodology, a consistent set of definitions and model of the gas turbine engine emerged. Specific relationships that applied to one PW4000 model but not the others were not included in the matrix. In this way, the matrix may be thought of as a platform model. For any specific design, the DSM would need to be tailored to the specific architecture.

As has been discussed in prior work [Rowles, 1999], one trend of system engineers and design experts is to assume all parameters are related to all other parameters. This tendency to believe in full interaction was found to be common during the development of the current DSM. This belief was overcome by developing definitions and relationships within the parameters for direct interaction versus indirect interaction. In general, it was found that all interactions the system engineers believed were present could be found by tracing through direct relationships to get to the variables thought to be related. Thus, there are many indirect relationships that are considered important by the experts.

Direct and Indirect Parameter Interactions within the DSM

The distinction between direct and indirect interactions within the DSM was developed early in the work and was refined continuously as the situation merited. The working definition developed was that the parameters needed a mathematical relationship (or the possibility of such a relationship) to be considered a potential direct link. However, the presence of a mathematical relationship was not sufficient to deem an interaction a direct relationship. It was quickly realized that with a series of mathematical equations, the equations could be combined and reworked such that again all parameters can be related to most other parameters. Thus, a second constraint was added that the relationship should be as simple as possible. While this constraint is subject to interpretation, system engineers are used to ambiguity and this constraint was very powerful in practical use. Third, if an indirect relationship could be thought of that explained the perceived direct relationship of variables within the DSM then no direct relationship was entered into the matrix.

When applying these constraints on direct interactions, it was found that the aerodynamics of the engine could be used to separate the variables into groups and determine generic relationships. For example is was found that using core airflow as a general group and treating all the airflow parameters identically helped simplify the thought process and maintain consistency within the matrix. As an example, for core airflow the general mathematical relationship between the variables is conservation of mass. Thus, Fan Flow Capacity (airflow) has a direct link to LPC Flow Capacity has a direct link to HPC Flow Capacity. Stability Bleed flow has a direct link to HPC Flow Capacity, but it only indirectly links to the LPC Flow Capacity. To continue this example, HPC Flow Capacity and the HPC PR (pressure ratio) directly define a line called HPC Op Line. Similarly, the LPC Flow Capacity and the LPC PR define the LPC Op Line. As surge margin in a gas turbine engine is an important safety measure, it is defined as the Op Line subtracted from the stall line. The stall line is a parameter defined by the physical module design. Thus, Surge Margins are defined as directly related to the Op Line and the Design.

The system engineer may know that the Stability Bleed is used to change the HPC Bodie Surge Margin. In the DSM, this is accounted for in the following manner: the Stability Bleed is directly related to the HPC Flow that is directly related to the HPC Op Line that is directly related to the HPC Bodie Surge Margin. This one example is typical of the interactions and tracing of the "known" relationships through the DSM. The tracing of indirect relationships through the DSM was one method used to validate DSM.

Validation of the DSM

Validation of the DSM was accomplished in several ways. First, the use of many system engineers and their discussion of the interactions amongst themselves proved to be valuable in both developing the relationships as well as their mental model of the system. Second, the tracing of direct relationships to verify a known indirect relationship was used extensively while the matrix was being developed as well as after completion of the matrix (see Bartkowski, 2001 for several additional examples). Tracing of past areas of difficulty (many times called mistakes) through the DSM proved to be both valuable for validation as well as interesting and insightful to the system engineers. Third, the DSM was validated using lecture notes and textbooks from a course in gas turbines offered at MIT [Kerrebrock, 1992]. This material was used to validate both the direct interactions as well as the groupings of parameters and mathematical relationships between the variables. Finally, during subsequent analysis by both developers of the DSM (the author and Bartkowski), the interactions were validated. In the author's case, this was primarily accomplished by understanding the results of partitioning and tearing of the DSM. The parameters used and their identifying number are given in Table 1.

Table 1: Parameters and their unique identifiers used throughout this thesis.

1	Low Rotor Speed
2	High Rotor Speed
3	TSFC
4	LPC Stability (Op Line)
5	HPC Stability (Op Line)
6	Acceleration Time
7	Deceleration Time
8	Fan Op Line
9	Fan Design
10	LPC Design
11	HPC Design
12	Corrected Main Oil Pressure
13	Main Oil Temperature
14	Take Off Thrust
15	Reverse Thrust
16	Emissions
17	Noise
18	LPC Takeoff Surge Margin
19	HPCT/O Surge Margin
20	HPC Bodie Surge Margin (Sea Level)
21	HPC Bodie Surge Margin (Altitude)
22	HPC Accel Surge Margin
23	LPC Surge Margin (Max Climb)
24	LPC Surge Margin (Cruise)
25	LCF Mission (Flight Cycle)
26	Flight Envelopes (Alt, MN, Temp)
27	Typical Operating Mission
28	Burst Margin
29	Rotating Part Life
30	Vibration Low Rotor
31	Vibration High Rotor
32	Engine Weight
33	Bird Loads
34	Fan Blade Out Loads
35	Nacelle Thermals
36	External Component Thermals
37	Nacelle Drag
38	ECS Bleed
39	TCA
40	TCC Bleed
41	Starting Bleed
42	Stability Bleed
43	Thrust Balance
44	Nacelle Cooling
45	Air Oil Coolers
46	Buffer Cooler
47	HPT/LPT Leakages
48	HPT/LPT Clearances
49	Anti-Ice Bleed
50	Diff/Combustor Design
51	Life Cycle Cost
52	Manufacturing Cost
53	Airline Operating Cost
54	In Flight Shut Downs

55 Unscheduled Engine Removals
56 Shop Visit Rate
57 Engine Change Time
58 TMC
59 Delays and Cancellations
60 Fan Flow Capacity
61 LPC Flow Capacity
62 HPC Flow Capacity
63 Burner Flow Capacity
64 HPT Flow Parameter
65 LPT Flow Parameter
66 Fan Exit Area
67 Jet Exit Area
68 Fuel Flow
69 Total Inlet Temperature/Profile
70 HPC Inlet Temperature/Profile
71 Burner Inlet Temperature/Profile
72 HPT Inlet Temperature/Profile
73 LPT Inlet Temperature/Profile
74 Exhaust Gas Temperature/Profile
75 Fan PR
76 LPC PR
77 HPC PR
78 Burner dP
79 HPT Expansion Ratio
80 LPT Expansion Ratio
81 Fan Efficiency
82 LPC Efficiency
83 HPC Efficiency
84 Burner Efficiency
85 HPT Efficiency
86 LPT Efficiency
87 LPT Design
88 Fan Nozzle Performance
89 Primary Nozzle Performance
90 Duct Losses
91 HPC Clearances
92 Low Rotor Inertia
93 High Rotor Inertia
94 Control Stability
95 Time to Light
96 Time to Idle
97 Burner Blowout Margin
98 Stator Vane Schedule
99 Ambient Temperature
100 Warranty Cost
101 2.5 Bleed
102 Engine Pressure Ratio
103 Control Laws
104 Fan/LPC Tip Clearance
105 Fuel /Air Ratio
106 HPT Design
107 Nacelle Design
108 External/Control Design
109 Mechanical Components Design
110 Fuel-Oil Coolers

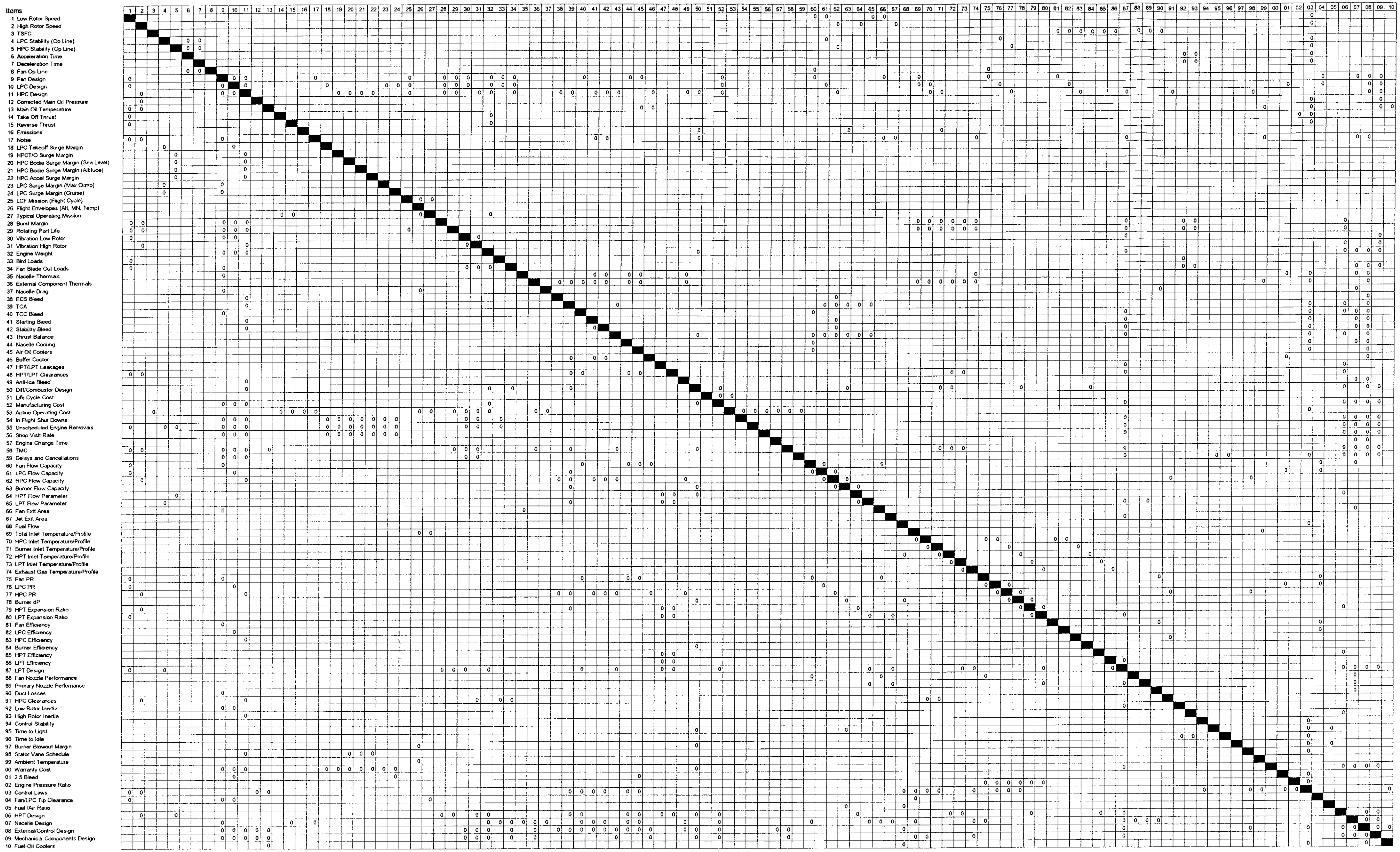
Baseline DSM

The baseline DSM developed for this work is shown in Figure 9. No immediate structure was obvious. By grouping of the parameters by areas of responsibility (i.e., imposing the current organizational structure), some structure begins to emerge in the DSM (Figure 10). However, there is no neat clustering of the interactions as found in previous work of Mascoli and Rowles. Bartkowski explores organizational and manual methods of organizing the DSM in more detail. This work focuses on mathematical partitioning and tearing of the DSM.

Partitioning and Tearing the Parameter DSM

Partitioning of a matrix involves reordering of the elements such that as many of the interactions as possible are all in the lower portion of the matrix (termed lower triangular). By ordering the elements in this way, information and interactions flow from one parameter to the next without any feedback. When a lower triangular matrix is not possible, partitioning may break the matrix into several sections with coupled blocks. Partitioning is not clustering in that no weighting is used to bring the upper triangular dependencies close to the diagonal. Partitioning makes no attempt to order items within a coupled block.

In a lower triangular matrix, a designer may determine the values of the parameters in the order of their listing and while doing so have complete information to determine the parameters. As a designer proceeds through the listing of parameters and a parameter has a feedback from a later parameter, the designer must make an assumption concerning the value of the parameter that has yet to be determined. By making this assumption, the designer is then able to determine



Items

1 Low Rotor Speed
2 High Rotor Speed
3 TSFC
4 LPC Stability (Op Line)
5 HPC Stability (Op Line)
6 Acceleration Time
7 Deceleration Time
8 Fan Op Line
9 Fan Design
10 LPC Design
11 HPC Design
12 Corrected Main Oil Pressure
13 Main Oil Temperature
14 Take Off Thrust
15 Reverse Thrust
16 Emissions
17 Noise
18 LPC Takeoff Surge Margin
19 HPCT/O Surge Margin
20 HPC Bodie Surge Margin (Sea Level)
21 HPC Bodie Surge Margin (Altitude)
22 HPC Accel Surge Margin
23 LPC Surge Margin (Max Climb)
24 LPC Surge Margin (Cruise)
25 LCF Mission (Flight Cycle)
26 Flight Envelopes (Alt, MN, Temp)
27 Typical Operating Mission
28 Burst Margin
29 Rotating Part Life
30 Vibration Low Rotor
31 Vibration High Rotor
32 Engine Weight
33 Bird Loads
34 Fan Blade Out Loads
35 Nacelle Thermals
36 External Component Thermals
37 Nacelle Drag
38 ECS Bleed
39 TCA
40 TCC Bleed
41 Starting Bleed
42 Stability Bleed
43 Thrust Balance
44 Nacelle Cooling
45 Air Oil Coolers
46 Buffer Cooler
47 HPT/LPT Leakages
48 HPT/LPT Clearances
49 Anti-Ice Bleed
50 Diff/Combustor Design
51 Life Cycle Cost
52 Manufacturing Cost
53 Airline Operating Cost
54 In Flight Shut Downs
55 Unscheduled Engine Removals
56 Shop Visit Rate
57 Engine Change Time
58 TMC
59 Delays and Cancellations
60 Fan Flow Capacity
61 LPC Flow Capacity
62 HPC Flow Capacity
63 Burner Flow Capacity
64 HPT Flow Parameter
65 LPT Flow Parameter
66 Fan Exit Area
67 Jet Exit Area
68 Fuel Flow
69 Total Inlet Temperature/Profile
70 HPC Inlet Temperature/Profile
71 Burner Inlet Temperature/Profile
72 HPT Inlet Temperature/Profile
73 LPT Inlet Temperature/Profile
74 Exhaust Gas Temperature/Profile
75 Fan PR
76 LPC PR
77 HPC PR
78 Burner dP
79 HPT Expansion Ratio
80 LPT Expansion Ratio
81 Fan Efficiency
82 LPC Efficiency
83 HPC Efficiency
84 Burner Efficiency
85 HPT Efficiency
86 LPT Efficiency
87 LPT Design
88 Fan Nozzle Performance
89 Primary Nozzle Perfomance
90 Duct Losses
91 HPC Clearances
92 Low Rotor Inertia
93 High Rotor Inertia
94 Control Stability
95 Time to Light
96 Time to Idle
97 Burner Blowout Margin
98 Stator Vane Schedule
99 Ambient Temperature
00 Warranty Cost
01 2.5 Bleed
02 Engine Pressure Ratio
03 Control Laws
04 Fan/LPC Tip Clearance
05 Fuel /Air Ratio
06 HPT Design
07 Nacelle Design
08 External/Control Design
09 Mechanical Components Design
10 Fuel-Oil Coolers

Figure 9: Baseline DSM developed in cooperation with Glenn Bartkowski [Bartkowski, 2001].

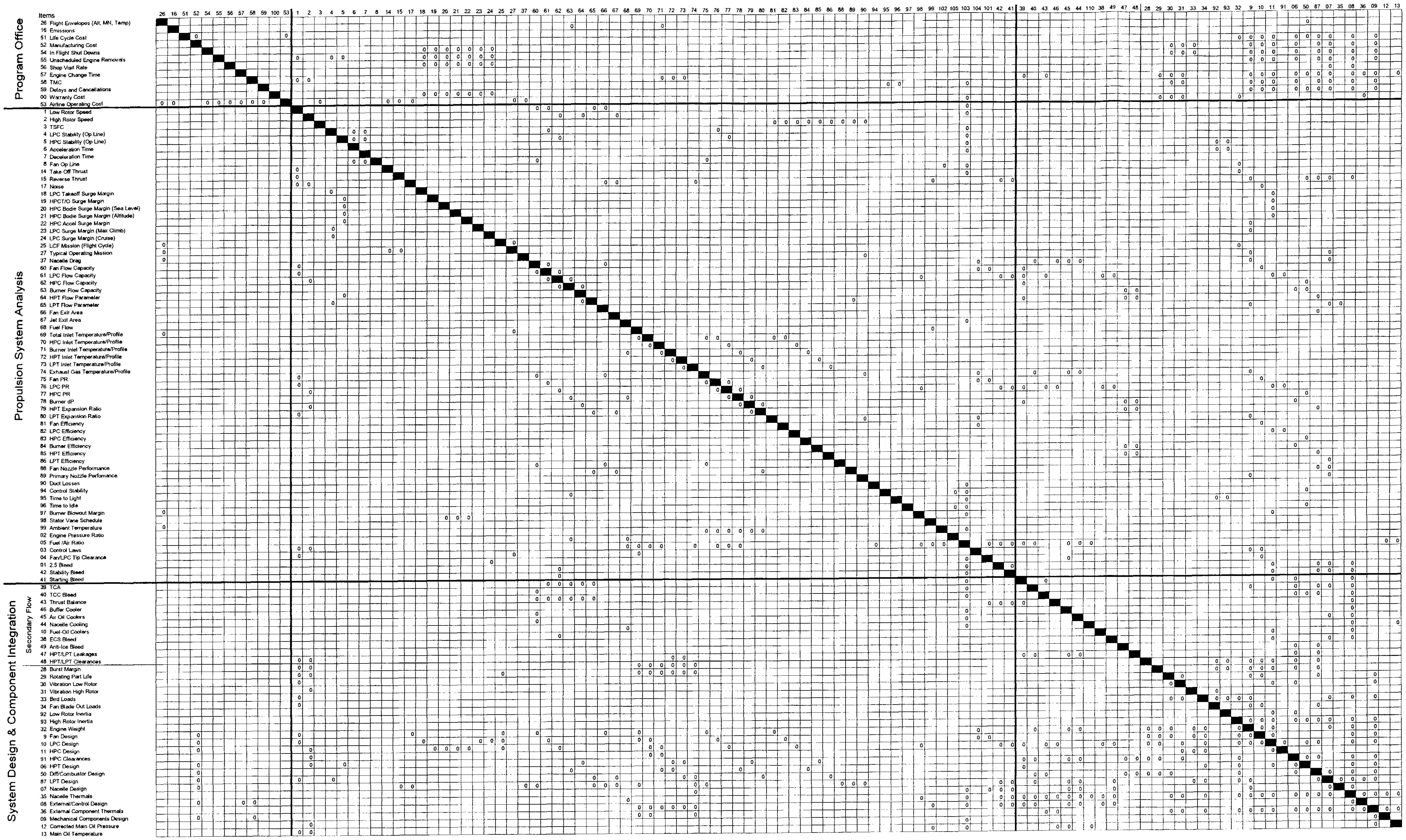


Figure 10: DSM organized by the organizational structure.

the parameter of interest. Later in the development, the assumed parameter will be determined and, depending upon the difference in the assumed and later determined value, an iteration cycle may be deemed necessary. Thus, the process of determining what is assumed and when it is assumed can significantly change the nature of the product development cycle and the preferred order for determining the parameters.

Requirements and their determination are intricately intertwined in the process of making assumptions that may later force iteration. One method of avoiding iteration (and thus improving the cycle time) is to make the initially assumed values the requirements. In this way, all designers use this same assumption and the final product must conform to the requirement. Enforcing requirements and not allowing for iteration will most often result in a sub-optimal design. In this case, the trade between cycle time and optimization of the product must be made. In most industry product development processes, the actual requirements are usually flexible (robust) to a (sometimes) known amount. Thus, in practice neither the requirements are unchanged nor are the iteration cycles allowed. The science (and some art) of setting robust design requirements is an entire field of study and will not be covered here.

Assumptions may be made in a DSM by first partitioning the matrix, and then ignoring individual interactions and repartitioning. This process is known as tearing a DSM. Overviews of tearing can be found in Steward's book [1981] and at several websites [PSM32, 2000; MIT DSM, 2000]. Repeated tearing of a DSM allows one to identify particular orders of parameters that may prove more resistant to iteration than others. The present work examines various sets of assumptions and the resultant torn DSM and discusses implications of these assumptions within the product development cycle of gas turbine engines.

Partitioning and tearing of the DSM in this work was accomplished with the software package PSM32. Don Steward and his collaborators developed this package for use within the DSM community [PSM32, 2000]. The package partitions a matrix and allows up to 9 additional levels of assumptions to be made.[1] Individual interactions between two parameters can be assumed or a systematic assumption may be made about a parameter.

The first systematic procedure is to assume a value for a single parameter. Following this assumption, this value can be used to determine all other parameters prior to the assumed parameter being determined (tearing by column in the matrix). Tearing by column will force the parameter to be one of the last parameters to be determined (i.e., the parameter is moved to the bottom of the coupled block). The second systematic procedure is to make assumptions about all the inputs to a single parameter that have yet to be determined and then determining the value of that parameter (tearing by row). Tearing by row forces the parameter to be one of the first to be determined (i.e., moved to the top of the coupled block). Tearing by row is similar to setting a requirement or constraint that must then be used in all subsequent parameter determinations. In general, tearing by column is often the preferred method as it only needs to assume a value for a single parameter. However, both methods are utilized throughout the current analysis.

---

[1] Additional levels of assumptions may be made within the program by a hierarchical method of using the tearing numbers. The current work limited itself to 9 levels of assumptions.

The PSM32 software package makes suggestions as to which parameters will break the longest linkages of parameters (circuits as Steward refers the them). In this manner, the matrix may be pushed to a lower triangular form. The suggestions made by the package are not unique, but could be changed within a small set of choices by having a different initial order in the matrix. For the current analysis, initial circuits found by PSM32 were between 55 and 60 parameters long. When tearing a matrix, one must proceed with caution when accepting the program suggested parameters. Only with detailed knowledge of the product and the product development process can one know whether making the suggested assumption is possible or would be acceptable to the product development organization. This work examined the acceptance of the program suggestions as well as other means of determining which parameters may be assumed. As is discussed in Chapter 4, the order and types of assumptions made was found to vary with the stage of the product development cycle.

The initial partitioning of the matrix resulted in an almost trivial answer (Figure 11). Two parameters, Flight Envelopes and Ambient Temperature, were moved to the top of the matrix. This is reasonable, as Flight Envelopes are normally give to the engine companies as requirements from the supersystem (the aircraft). Although there may be feedbacks between the engine and the aircraft that help determine the flight envelope, from the perspective of the engine as system, the flight envelope can be considered a requirement and thus, should be one of the first things determined. The ambient temperature is just the air temperature throughout the flight enveloped and is not altered by anything within the engine. Ambient temperature follows directly from the flight envelope requirement. Next there is a large block of 94 items that are interdependent. Finally, there is a small section of 14 parameters after the large interdependent block that is indeed lower triangular. These variables are known to be the emergent properties of the engine system. The variables are items such Airline Operating Costs, Delays and Cancellations and other highly dependent variables. Overall, not much information could be obtained from a single attempt to partition the matrix. Apparently, the view that almost everything is related to everything else is true, at least from a cursory examination.

The next step in the analysis was to tear the matrix by making various assumptions. In order to gain and experience with the tool and with tearing, the initial tears were made using the program-suggested variables. An interesting aspect of this was that the program consistently suggested many of the Design parameters (e.g., HPC Design) as the parameters which to assume. In addition, the Control Laws parameter was often suggested and appears to be similar to the module designs as far as the level of interdependency. From these first trials, it became apparent that a group of 10 related variables (Fan Design, LPC Design, HPC Design, Diff/Combustor Design, HPT Design, LPT Design, Mechanical Components Design, Externals/Control Design, Nacelle Design, and Control Laws) should be treated similarly. The fact that the Design variables should be related and that they are the first variables suggested is logical given the initial setup of the DSM. In essence, by adopting a system perspective, the designs of the modules become individual parameters that are highly interdependent with the system level parameters.

Each Design parameter in the current DSM is a block (cluster) in Mascoli's DSM (except for modules that Mascoli chose to ignore for the sake of time). To more fully integrate the information from Mascoli's work, each Design parameter could have been replaced by the

Figure 11: Partitioned DSM with large block of interdependent parameters.

approximately 10 variables of each module in Mascoli's work [Mascoli, 1999]. Although this may have resulted in different coupling of the parameters, it was not done in order to maintain a consistent perspective in the DSM. In the current work, the Design parameters become the parameters that most DSM practitioners refer to as the system parameters that are quickly moved to the top or bottom of their list and eliminated from further consideration. This work suggests that what is viewed as a system in an individual work depends entirely upon the perspective of the developers. System in one view may just be a subsystem or supersystem in another perspective. This supports the view that DSMs may be structured hierarchically and could easily be developed and interlinked with web-based technologies. Such a hierarchical system with approximately 1000 parameters exists at Boeing [Stowe, 2000].

Developing an order within the Design parameters required examination of the literature and industry practice as no discernable pattern was apparent within the PSM32 suggestions. In his examination of platforms within the gas turbine industry, Moy concluded that the HPC is the platform of a gas turbine engine [Moy, 2000]. Thus, the HPC Design should be the first design parameter that is assumed. Given its physical and parametric interactions with the HPC as well as the technology limitations, the HPT Design should be the second design parameter assumed. This completes the high spool of the engine that many long time gas turbine practitioners feel is the platform of an engine. Next, the low spool of the engine should be assumed. For the PW4000, the fan, LPC, and LPT are assumed in this order based upon the difficulty of the available technology and industry experience. The Diff/Combustor is next and is the only non-rotating module in the aerothermal design. These first six design parameters represent the aerothermal and structural design of the engine. The Mechanical Components

Design (the bearings and the associated oil and fuel systems including the gearbox) should be the first non-core design assumed followed by the External/Control Design and the Nacelle. This order works from the inside of the engine out with the outside of the engine (the nacelle) being the item that the public views on the aircraft. The relegation of the nacelle to be the last assumed may not be appropriate as this is a significant interface with the aircraft. (In fact, it is such a large interface that Boeing considers it part of their design responsibility while Airbus utilizes the engine companies or third parties to design nacelles for their aircraft). This ordering again is linked to the arguments on perspective of the DSM developer. Since the perspective of the present DSM is engine centric, the nacelle is the last design to be assumed. Table 2 summarizes the sequence and reasons for assuming the design parameters.

Table 2: Sequence and reasons for assuming module design parameters.

Order Assumed	Design Parameter	Explanation
1	HPC Design	Platform for engine [Moy, 2000]
2	HPT Design	Tightly coupled to HPC and technology limitations
3	Fan Design	Available technology and historical difficulty of low spool
4	LPC Design	Historical difficulty of low spool
5	LPT Design	Historical difficulty of low spool
6	Diff/Combustor Design	Completion of aerothermal design
7	Mech Comp Design	Importance of bearings and mechanical systems
8	Externals	Usually the last thing designed
9	Nacelle	Outside of engine (only sometimes part of engine)

Figure 12 and 13 represent the DSM partitioned after the Design parameters were torn in the order described above. Figure 12 represents the matrix when the parameters are torn by column (i.e., an assumed design is used for the calculation of most other parameters prior to the designs being completed). Figure 13 represents the matrix when the parameters are torn by row,

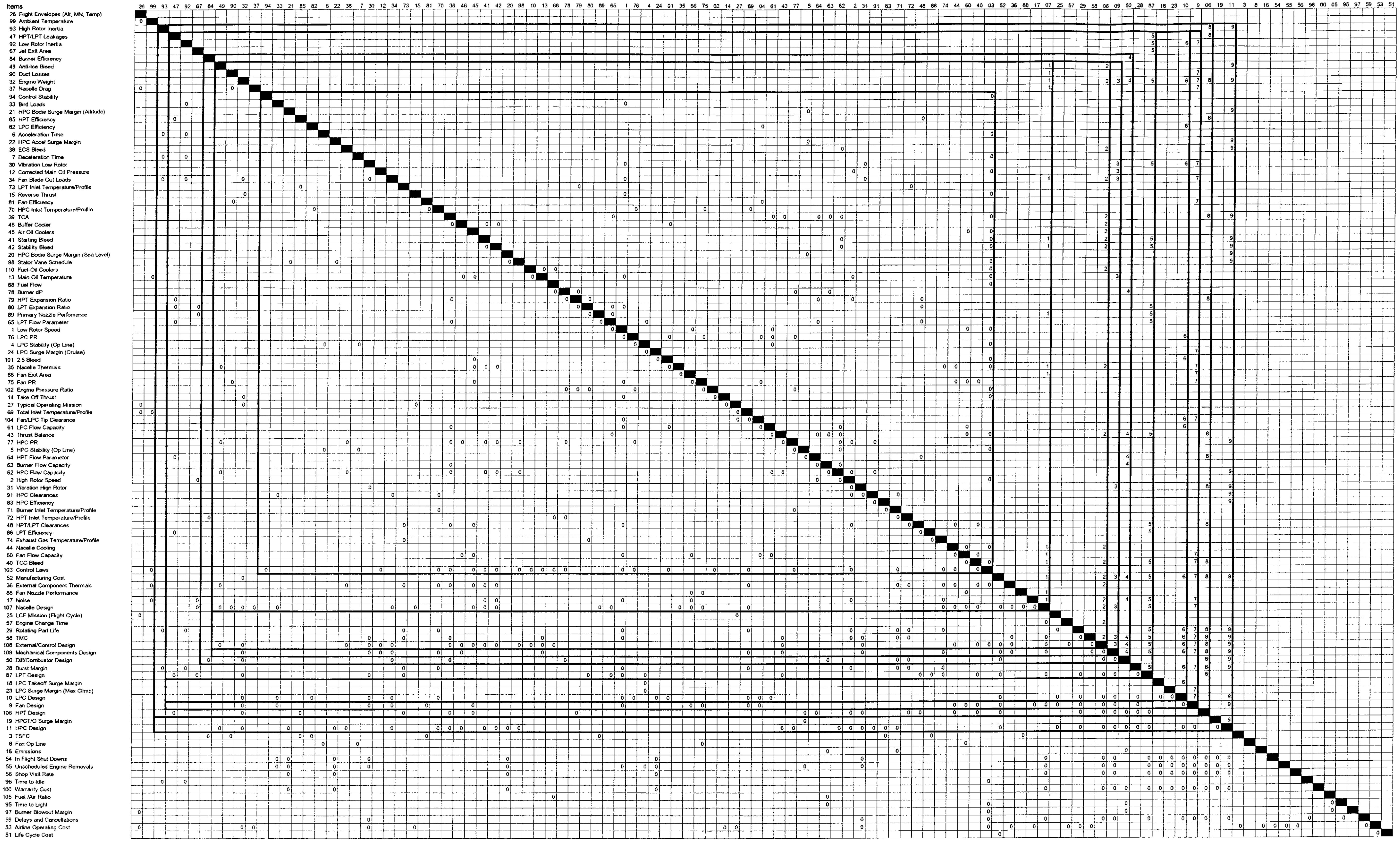


Figure 12: The DSM with the Design parameters torn by column.

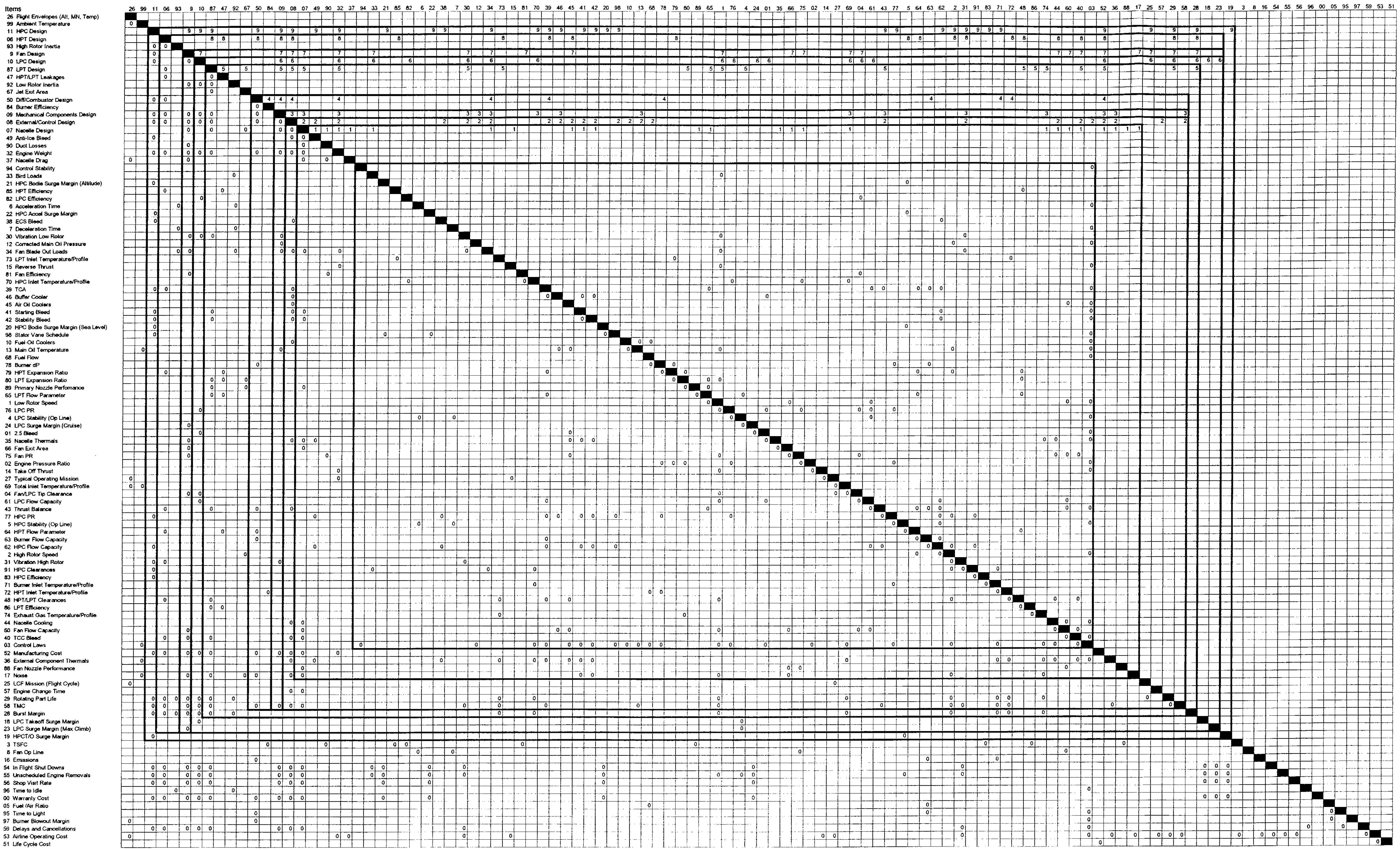


Figure 13: The DSM with the Design parameters torn by row.

(i.e., all necessary parameter information is assumed to be known and the designs are determined first). In both these (and all subsequent) matrices, a number above 0 in the matrix represents an assumption of the variable lower in the matrix. Higher numbers represent assumptions that are made first. Thus, any tick mark that is a 9 was assumed first in the partitioning, tick marks with an 8 were assumed next, etc. down to tick marks of 0 where there is no assumption.

In the above work with the Design parameters, the Control Laws were excluded from the analysis. This was done since it is not entirely clear to the author whether the Control Laws (basically the software that runs the engine) are similar enough to the physical design of hardware to be in the same set of parameters. The fact that the Control Laws parameter is part of one of the system engineering organizations shows that it is understood that this is an important system level function. For the remainder of this work, the Control Laws parameter is the first parameter assumed after the Design parameters.

For the remainder of this thesis, the nine Design parameters are assumed in the order discussed above and are assumed simultaneously (i.e., all the necessary tick marks are given a 9). Whether the parameters are torn by column or by row will be dependent upon the phase of product development as is discussed below. The DSMs with these assumptions are given in Figure 14 and 15. Note that in both cases, a core block of 64 parameters remains to be examined.

At this point several additional comments should be made on the specific ordering of the variables. In Figure 14, the Diffuser/Combustor Design and the Nacelle Design are not in the order stated above. This is due to PSM32 reordering the variables to earlier points just above were the base designs are torn by column. The author is not aware of the logic for this move, but includes the result for future analysis. In a similar manner, the Fan Op Line, Time to Light and other variables that should be part of the aerothermal block of parameters are removed and put below the main grouping of variables during the very first partitioning prior to any assumptions. The PSM32 program separated these variables due to the interdependencies included with the parameters that are present within the current DSM. While PSM32 acted accordingly, the limited scope of the variables chosen for the DSM is the result of their separation and not their independence. Thus, they should be included as part of the aerothermal design discussed later in this work. Another apparent logical error is the variables such as HPT/LPT Leakages, Deceleration Time and variables that are brought to the top of the matrices once the Designs are assumed. Again this is due to the limited nature of the selection of variables rather than true independence. Since the design variables are large conglomerations of many lower level parameters, once assumed in PSM32, all the lower level parameters would be assumed. In practice this is not the case, but only a subset of these lower level parameters are assumed. Many of the variables brought to the top should actually be included in the aerothermal design. Separation of the designs into lower level parameters to understand the full implications of making the design assumptions is left to future work.

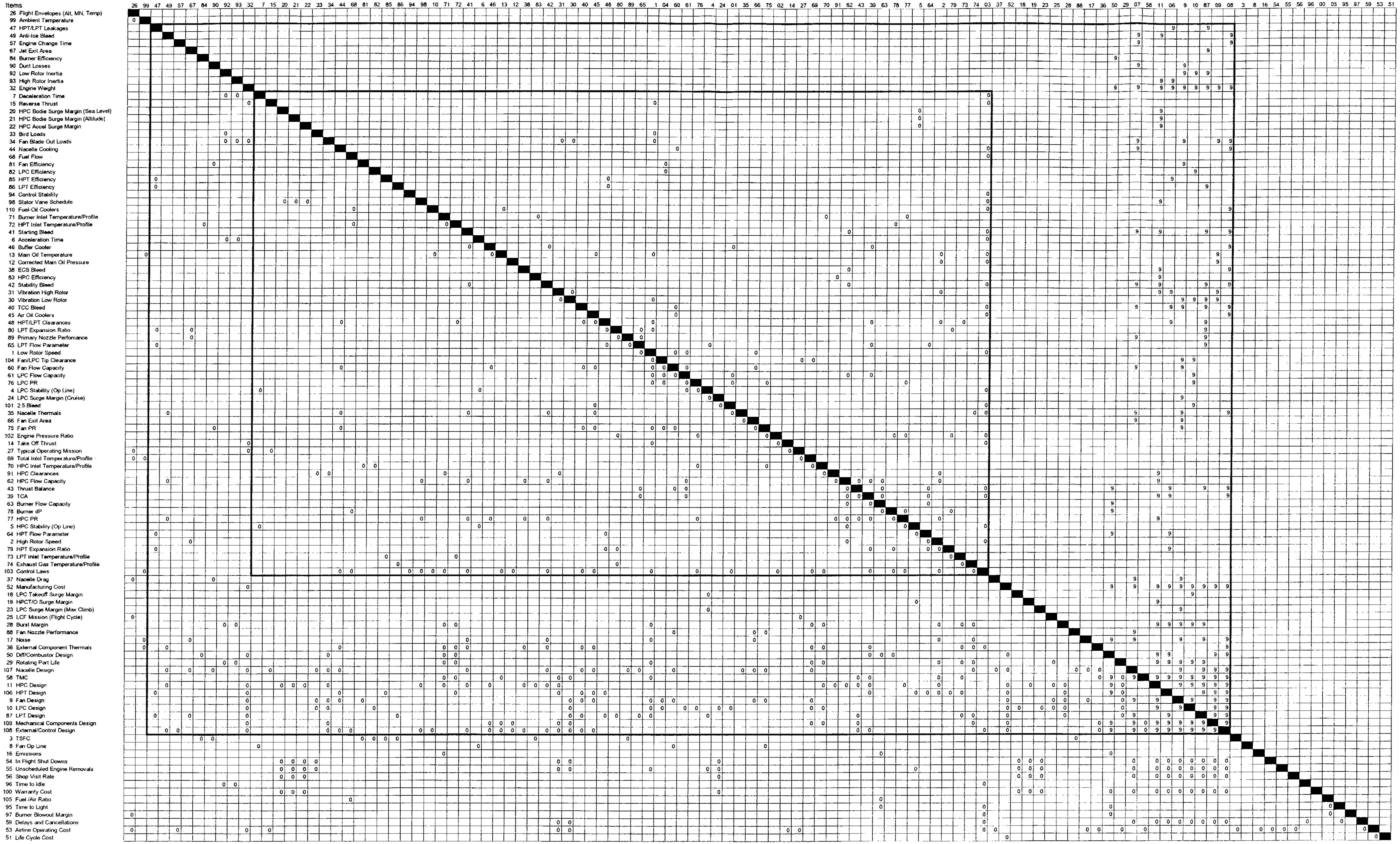


Figure 14: Base DSM for stages that have the Designs determined late.

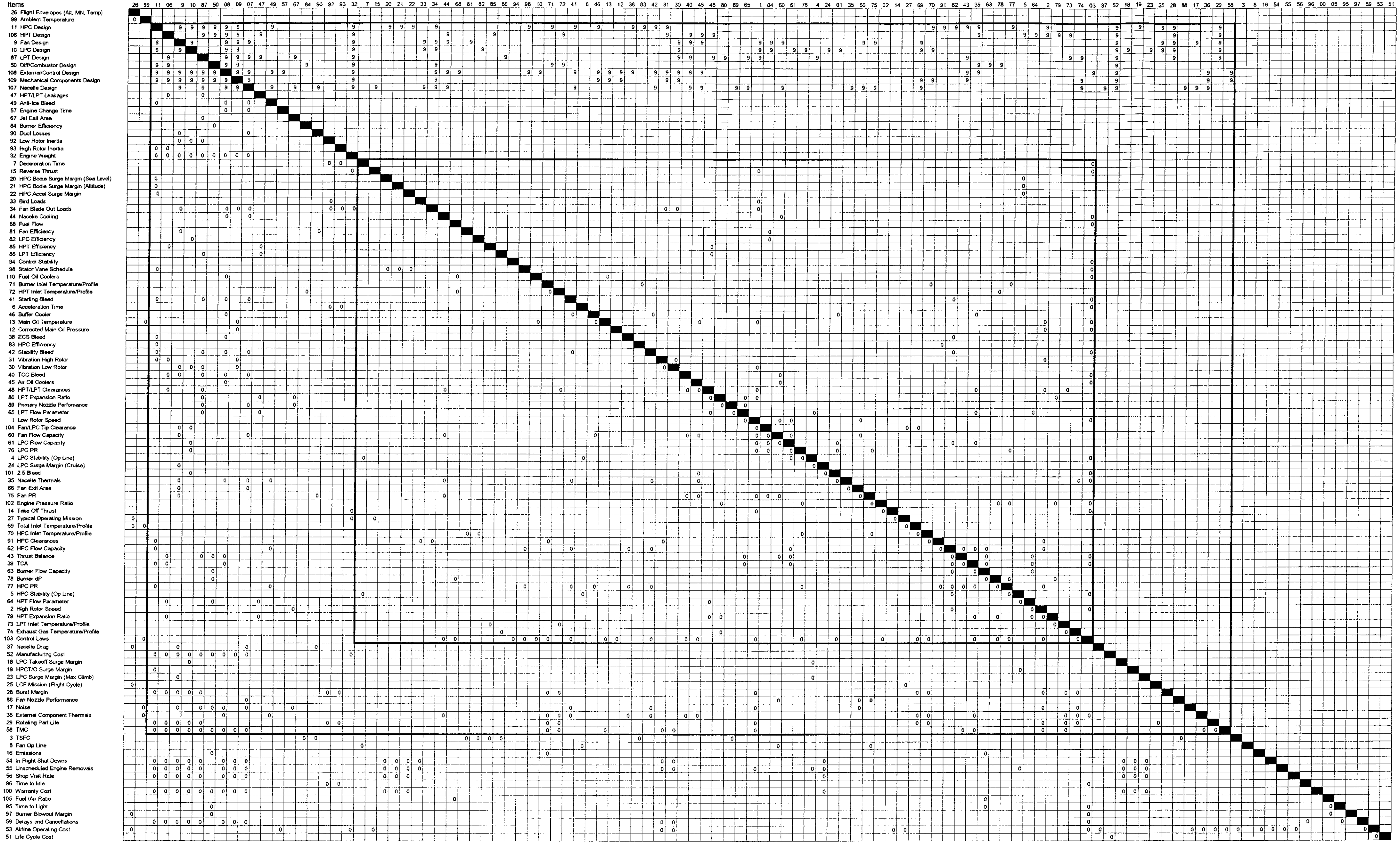


Figure 15: Base DSM for stages that have the Designs determined early.

CHAPTER 4: VARIATION OF ASSUMPTIONS BY STAGE OF PRODUCT DEVELOPMENT

The product development process involves developing and incrementally improving a design concept until it has evolved into a full product. The process by which the product is developed is a series of steps where the product is developed and validated in an ever increasing level of complexity. In general, as the product moves through the development process fewer assumptions are made and more has been validated. The process by which assumptions are made and how they are validated is the topic of this chapter. The analysis is broken down into four stages of product development (conceptual design, preliminary design, detailed design, and validation). Note that here and throughout the remainder of the thesis preliminary design and system-level design are meant to be synonymous. Production and field service are not included in the present analysis since all assumptions are (supposedly) validated prior to production and release to the field.

The analysis in this and subsequent chapters is confined to products that are past the stage where the architecture is undergoing radical innovation. In general, this work is believed to be valid for products that have a dominant design. By limiting the scope of the analysis, a single architectural DSM should be valid. For the incremental changes in architecture, the DSM may be updated. However for products that are undergoing radical architectural changes, little reuse is possible and a standard product development cycle is unlikely to be present.

In this framework, all analysis in the current chapter based upon either Figure 14 or Figure 15, both of which are derived from Figure 9. In other words, all analyses completed in this work are based upon a single parameter-based DSM. The analyses differ only in the assumptions made (i.e., how the parameters were torn).

Assumptions for Conceptual Design

The first question asked when developing a commercial engine is what is the required thrust. The thrust necessary sets the class of engine that the aircraft will require. With a simple analysis, the desired thrust of the engine dictates the optimal overall pressure ratio. Many thrust/pressure ratio combinations have been explored in the past and, for commercial engines where the primary need of the airlines is thrust with low fuel consumption, knowing the thrust and the application desired leads to a small set of engine designs (e.g., 90,000 lb. thrust for a commercial passenger aircraft).

Gas turbine manufactures emphasize the reuse of designs and typically develop concepts for engines around successful designs of the past. Once the class of engine is known the next questions that are asked are what has been done in the past on which to base this design and what technologies/design concepts are going to be inserted into the concept. The representation of this question in a DSM would be to assume design parameters that are the best the company has developed to date (not necessarily put into service). In a DSM, the thrust followed by the base design parameters would be torn by row (Figure 16). Modifications to the design parameters would then be accomplished based upon the new design concepts or technology. The concept designer then develops the aerothermal cycle based upon the constraints imposed by the designs. Note that most of the following discussion is based upon lectures at MIT and several texts [Epstein, 2000; Kerrebrock, 1992].

Figure 16: DSM with thrust and design parameters assumed.

For the last 50 years, one of the limiting technology aspects of a gas turbine is its turbine inlet temperature. For modern engines, the turbine inlet temperature is well above the melting point of the materials that make up the turbine. Coatings and cooling technology are used to keep the alloys from melting and maintaining the component integrity throughout its life. The compressor exit temperature is also limited by the materials at the rear of the compressor although design trades are such that no coatings or cooling are currently used in the compressor (and therefore the allowable temperatures are significantly below the melting point of the materials). In the DSM, the first two parameters assumed during the aerothermal design of an engine (the large block of Figure 16) are the HPT Inlet Temperature and the compressor exit temperature (termed the Burner Inlet Temperature in the current work). Note that not only are the overall temperatures important, but also their profiles. Making these assumptions leads to narrowing of the large interdependent block of the DSM (Figure 17).

After these two technology limited parameters are determined, the aerothermal design can be determined. The flow capacity of the fan and the core modules (and thus, the bypass ratio) are the first parameters to be determined during the aerothermal design. Interdependent with the determination of flow capacities are the pressure ratios across the modules. This combination then sets the operating lines and therefore the surge margins. Transient behavior and secondary flow requirements are also interlinked with the flow capacity and pressure ratio determination. Figure 18 and Figure 19 represent two possible set of assumptions with the difference being when and how the assumptions are made about the secondary flow parameters (bleeds and leakages). In practice, aerothermal design undergoes many iterations in computer simulation. Due to the number of iterations, the particular order the assumptions is not extremely important.

During conceptual design at P&W, many possible technology and cycle combinations are explored with the deliverable of two or three of the options completed in detail. These options are then considered in a upper level management review with a single concept being selected to proceed to preliminary design. The DSMs in Figure 18 and 19 would be typical of the conceptual design process. For P&W, the conceptual DSM can be represented by:

1) obtaining the supersystem requirements from the airframer as well as program goals
2) obtaining the details of historical designs and developing new design concepts
3) assuming technology improvements and evaluating risks
4) completing many iterations on the aerothermal design, and for select options
5) validating integrative parameters to ensure the airframer requirements and program goals are met.

Here the author has taken the liberty of changing the pure parameter DSM to a task-based DSM where each task is the determination of a single parameter from the parameter DSM. Following conceptual design, the full engine has been designed. It is a virtual, paper engine, but the complete engine with all modules is running within the computer simulation. The design is based upon many rough assumptions that are based upon historical capability and scaling as well as new design concepts and technology.

These five steps can be generalized and represented in a schematic DSM (Figure 20). One note of caution on this and the remaining schematic DSMs in this thesis. The schematic

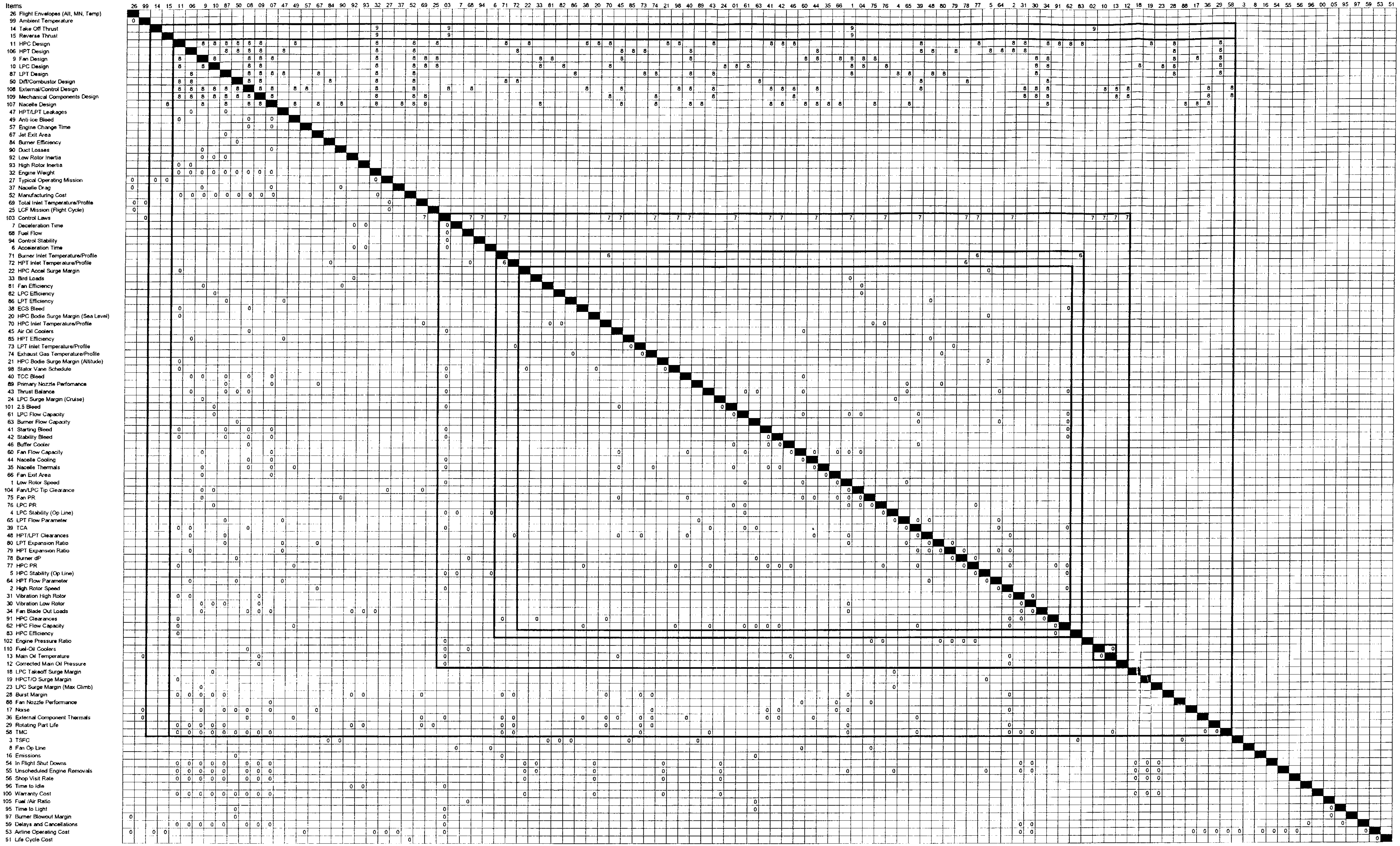


Figure 17: Partitioned DSM with assumptions for new temperature capabilities.

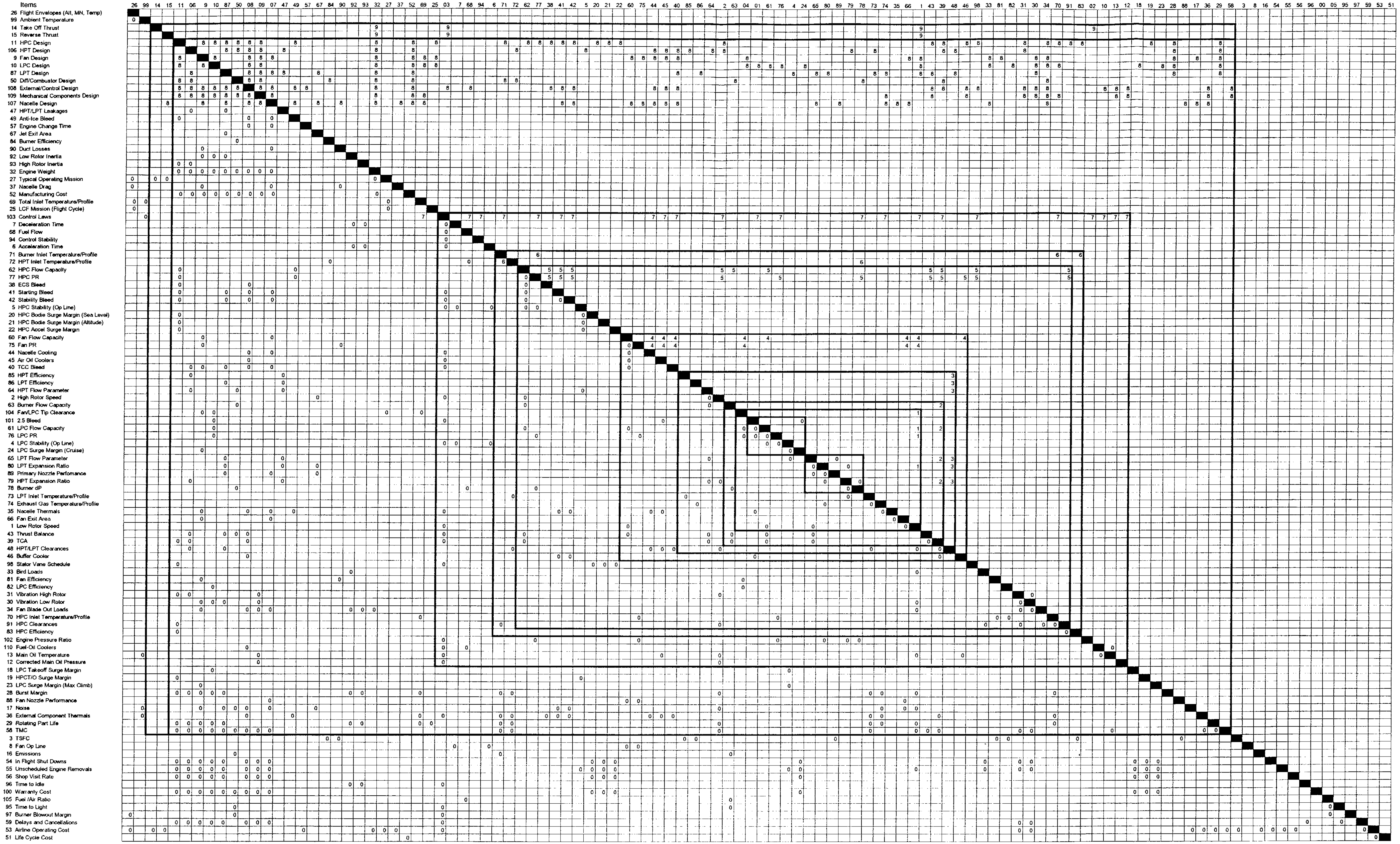


Figure 18: DSM with one set of assumption during the aerothermal design.

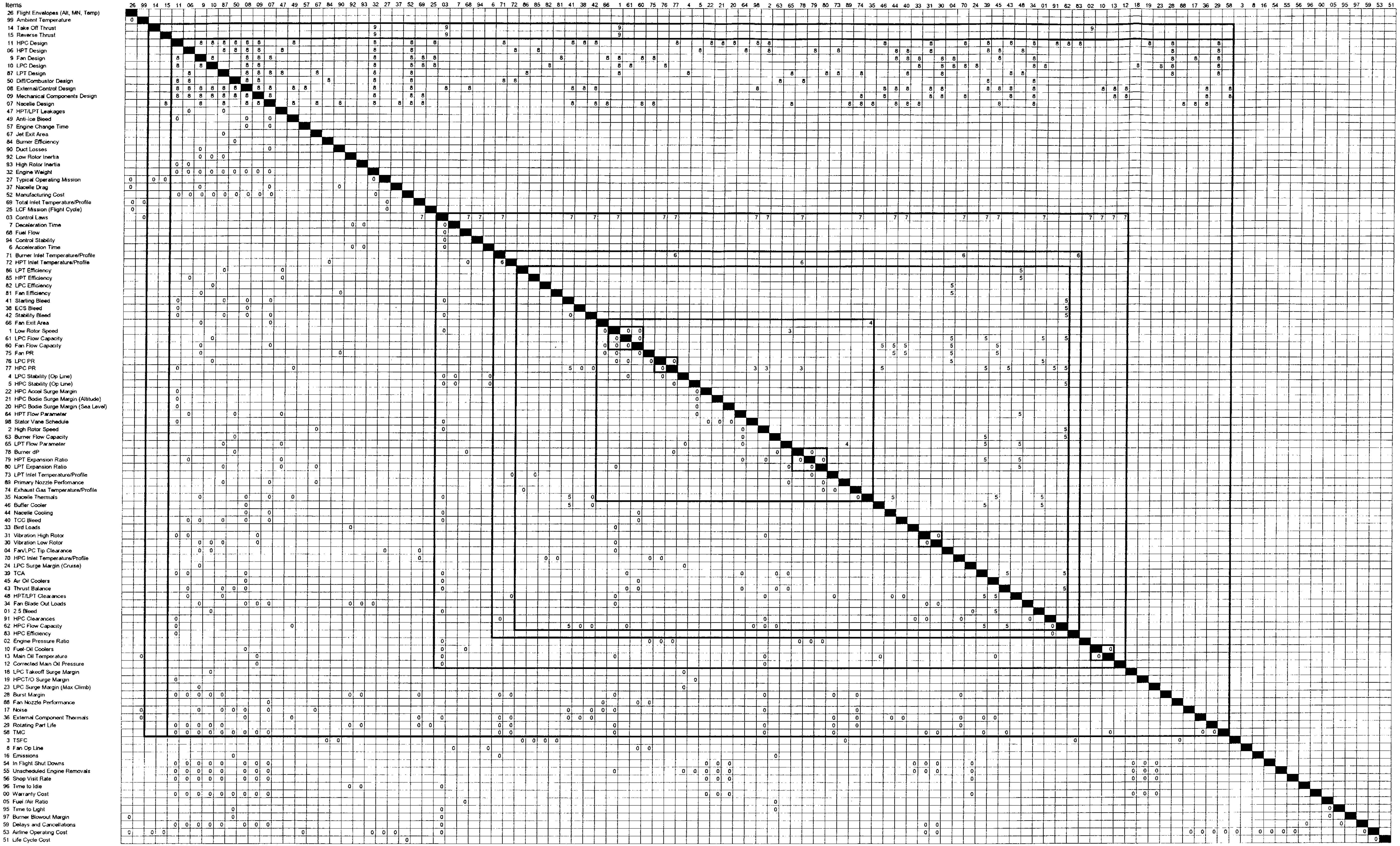


Figure 19: DSM with a second set of assumptons during the aerothermal design.

DSM are believed to be applicable to a wide range of PDPs across industries. The authors experience suggests that the gas turbine PDP is fairly typical and appears to be consistent both with the authors System Design & Management education at MIT and with informal discussions with employees in other industries. However, detailed examination of other companies' PDPs is left to further work.

Following the development of all the parameters in the DSM, the P&W process calls for the development team to summarize all the progress and risks that are present in the current state of the product and to plan the next stage of the PDP. The team then holds a senior management review where a decision is made as to whether to proceed to preliminary design.

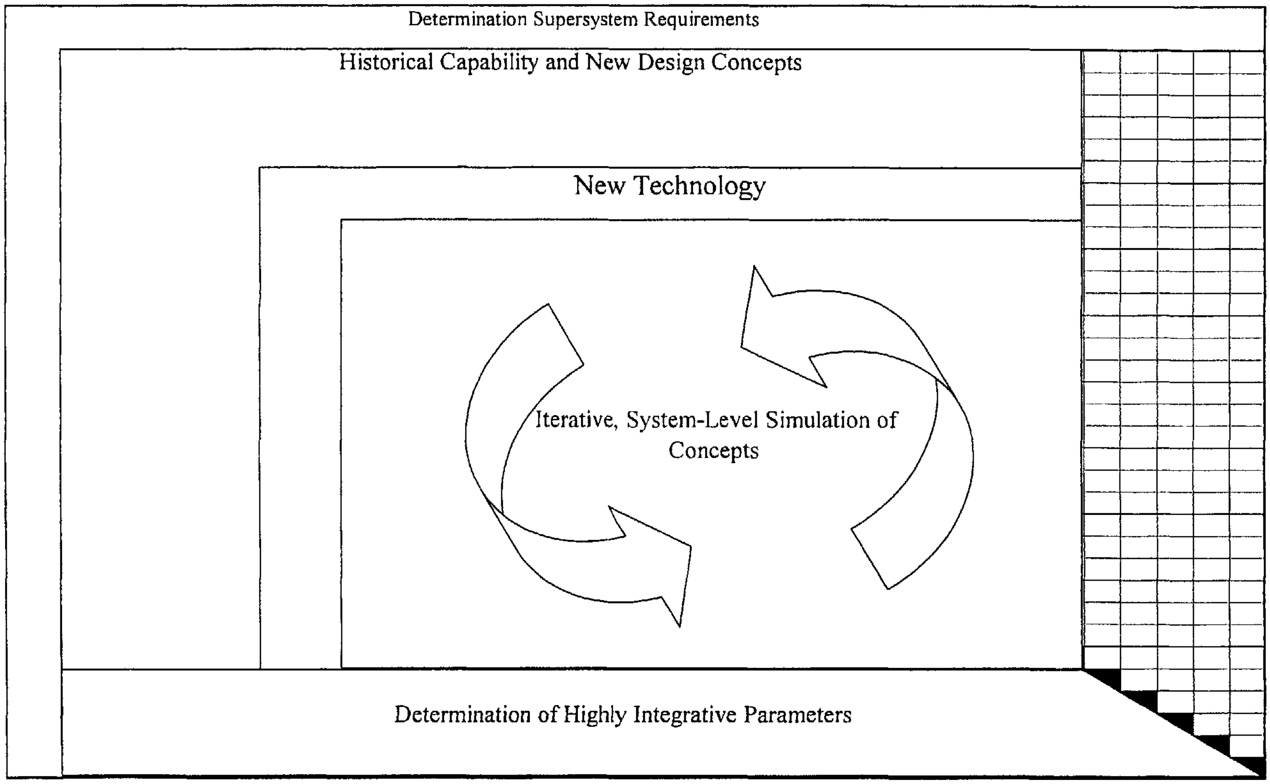


Figure 20: Schematic DSM for Conceptual Design.

Assumptions for Preliminary Design

Although, preliminary design may seem to have many features in common with the conceptual design, the order in which the assumptions are made is different. For preliminary design, the initial aerothermal design parameters are used as a starting point to build a detailed system level simulation. The detailed simulation output is used as test conditions to validate the new technology as well as the new design concepts assumed during conceptual design. Due to risks identified during preliminary design, the design then undergoes iteration until the preliminary design is found to meet the supersystem requirements and program goals at a minimal risk. At this point, the preliminary design is used to develop system requirements and the program is launched.

Figure 21 is the DSM ordered appropriately for the preliminary design phase. When extended to the general product development process, preliminary design consists of:

1) Obtain output of conceptual design, supersystem requirements, and program goals
2) Develop detailed system-level simulation to assist in risk identification
3) Validate the technology and design concepts within the module design
4) Document integrative parameters and system-level requirements

This schematic preliminary design process is shown in Figure 22. In general, the simulations and models used in preliminary design are of a higher level of fidelity than the simulations in conceptual design. Also following preliminary design, the new technology and design concepts have been verified at the modular level. Note that in preliminary design, the aerothermal design is completed prior to the validation of new technology and design concepts. Thus, these categories of parameters are determined in reverse order from conceptual design.

Similar to the end of Concept Design, a product review is held with senior management at the end of Preliminary Design. At the product review, the plan for the remainder of the program is presented to senior management. To develop the program plan a very detailed risk management process is followed to identify and mitigate the major risks to the program. The deliverable from the preliminary design is the validation of the assumptions made during conceptual design and a set of system requirements that will be used for the detailed design of the engine. At this point the system-level requirements are documented and become very difficult and costly to change. However, the subsystem and component requirements are still flexible within the overall system requirements. The system requirements are documented as Chapter 1 in the PS/CRD.

As Mascoli stated in his work, one of the enabling features designing a complex product with geographically distributed design teams is 1) a set of system-level requirements and 2) only small levels of interdependence of the subsystems. Preliminary design develops and documents system-level requirements with a high degree of fidelity for the detailed design and validation of the product. For large, distributed design teams, an architecture[2] that minimizes the inter-module

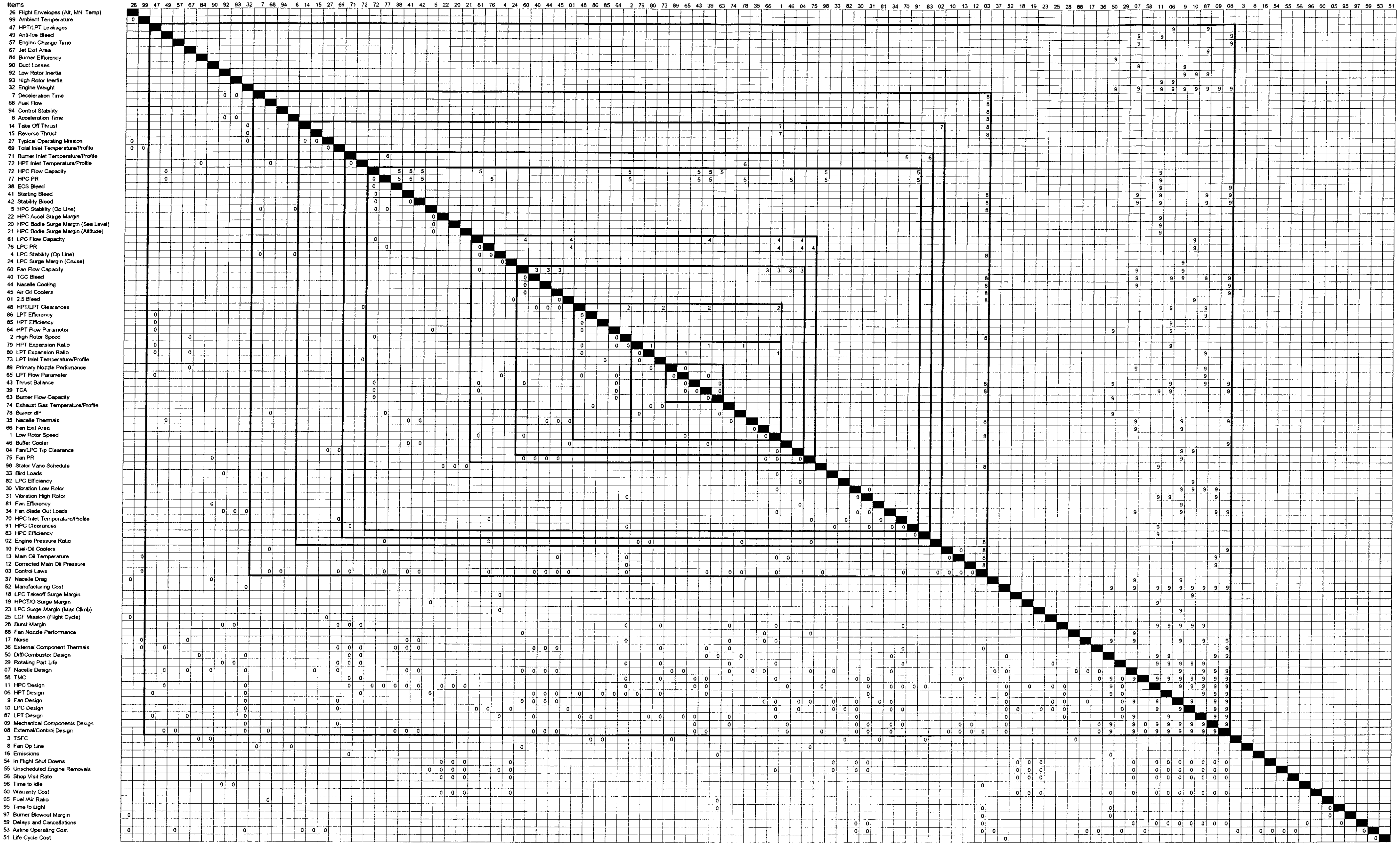
---

[2] When first attempting to develop an innovating complex product, the design teams are generally very small and are therefore do not need to worry as much about the independence of the modules. However, as an innovative product emerges into a market and competition based upon product architecture changes to competition based on process improvements, it may be that the architecture that allows the specialization and independence of the subsystems is able to improve faster than other architectures. This may lead the market to select the more modular architecture to become the dominant design.

interactions is one of the goals of concept design. Conceptual design thus sets one of the enabling features of the development of complex product architectures.

Assumptions for Detailed Design

The system-level requirements developed in preliminary design are used to start the detailed design of the product. Detailed design is normally accomplished on individual subsystems, at times termed parts, components, and modules. As such, the designers and design teams are developed and trained to be experts within their narrow functional area. The level of detail is much greater during this phase of design and as many people have stated, this is the deep dive of product development (Figure 23). During this phase, the system-level DSM developed in the current work is not appropriate. A DSM such as developed by Mascoli may be better suited for this phase of design, but even his DSM was considered a system level DSM. His work had approximately 10 parameters for each module. One view of this is that the 10 parameters defined the subsystems that are present within the module. A more appropriate level of detail may be to have a DSM of near 100 parameters for each module. This level of detail is what is most often developed during DSM research. For examples, see the work in the automotive industry [Pimmler and Eppinger, 1994]. This concept leans toward the hierarchical design of a set of DSMs. Figure 24 shows this schematically.



Items
26 Flight Envelopes (Alt, MN, Temp)
99 Ambient Temperature
47 HPT/LPT Leakages
49 Anti-Ice Bleed
57 Engine Change Time
67 Jet Exit Area
84 Burner Efficiency
90 Duct Losses
92 Low Rotor Inertia
93 High Rotor Inertia
32 Engine Weight
7 Deceleration Time
68 Fuel Flow
94 Control Stability
6 Acceleration Time
14 Take Off Thrust
15 Reverse Thrust
27 Typical Operating Mission
69 Total Inlet Temperature/Profile
71 Burner Inlet Temperature/Profile
72 HPT Inlet Temperature/Profile
72 HPC Flow Capacity
77 HPC PR
38 ECS Bleed
41 Starting Bleed
42 Stability Bleed
5 HPC Stability (Op Line)
22 HPC Accel Surge Margin
20 HPC Bodie Surge Margin (Sea Level)
21 HPC Bodie Surge Margin (Altitude)
61 LPC Flow Capacity
76 LPC PR
4 LPC Stability (Op Line)
24 LPC Surge Margin (Cruise)
60 Fan Flow Capacity
40 TCC Bleed
44 Nacelle Cooling
45 Air Oil Coolers
01 2.5 Bleed
48 HPT/LPT Clearances
86 LPT Efficiency
85 HPT Efficiency
64 HPT Flow Parameter
2 High Rotor Speed
79 HPT Expansion Ratio
80 LPT Expansion Ratio
73 LPT Inlet Temperature/Profile
89 Primary Nozzle Perfomance
65 LPT Flow Parameter
43 Thrust Balance
39 TCA
63 Burner Flow Capacity
74 Exhaust Gas Temperature/Profile
78 Burner dP
35 Nacelle Thermals
66 Fan Exit Area
1 Low Rotor Speed
46 Buffer Cooler
04 Fan/LPC Tip Clearance
75 Fan PR
98 Stator Vane Schedule
33 Bird Loads
82 LPC Efficiency
30 Vibration Low Rotor
31 Vibration High Rotor
81 Fan Efficiency
34 Fan Blade Out Loads
70 HPC Inlet Temperature/Profile
91 HPC Clearances
83 HPC Efficiency
02 Engine Pressure Ratio
10 Fuel-Oil Coolers
13 Main Oil Temperature
12 Corrected Main Oil Pressure
03 Control Laws
37 Nacelle Drag
52 Manufacturing Cost
18 LPC Takeoff Surge Margin
19 HPCT/O Surge Margin
23 LPC Surge Margin (Max Climb)
25 LCF Mission (Flight Cycle)
28 Burst Margin
88 Fan Nozzle Performance
17 Noise
36 External Component Thermals
50 Diff/Combustor Design
29 Rotating Part Life
07 Nacelle Design
58 TMC
11 HPC Design
06 HPT Design
9 Fan Design
10 LPC Design
87 LPT Design
09 Mechanical Components Design
08 External/Control Design
3 TSFC
8 Fan Op Line
16 Emissions
54 In Flight Shut Downs
55 Unscheduled Engine Removals
56 Shop Visit Rate
96 Time to Idle
00 Warranty Cost
05 Fuel /Air Ratio
95 Time to Light
97 Burner Blowout Margin
59 Delays and Cancellations
53 Airline Operating Cost
51 Life Cycle Cost

Figure 21: DSM with assumptions for Preliminary Design.

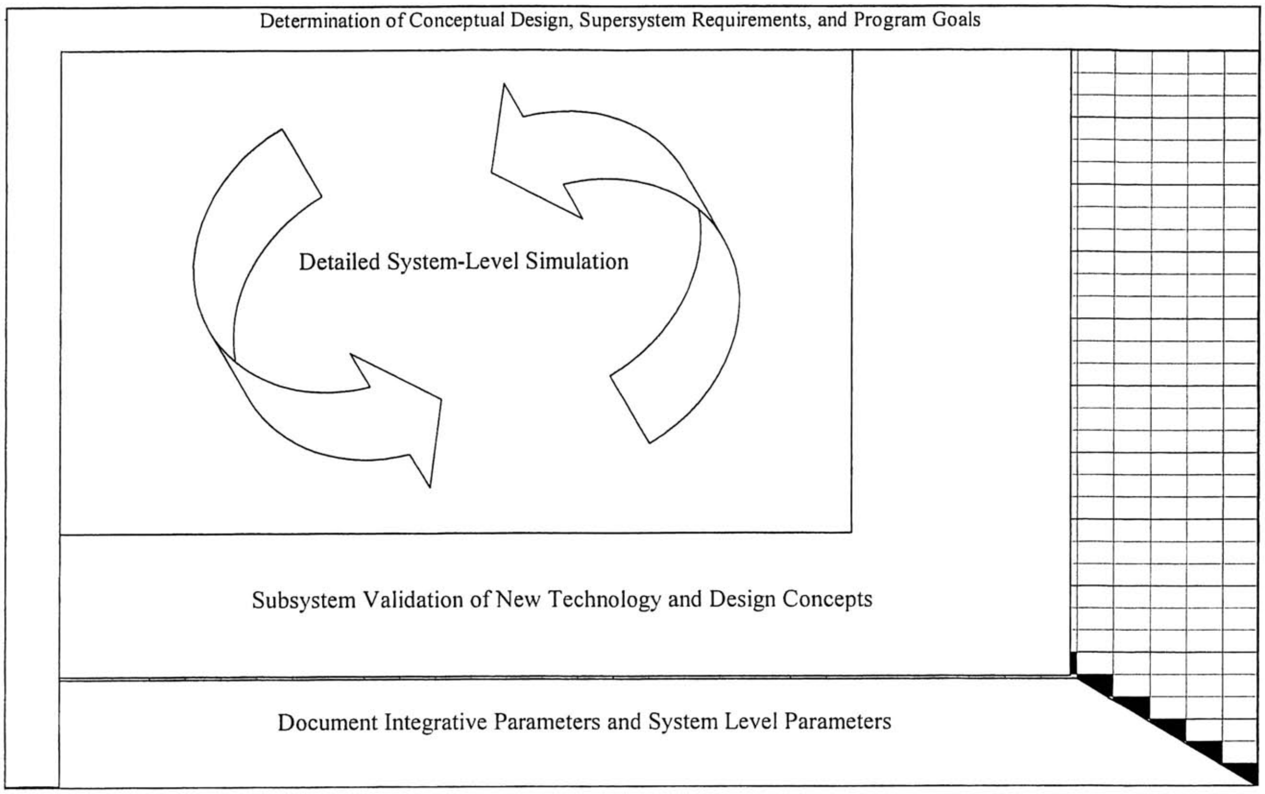


Figure 22: Schematic DSM for Preliminary Design.

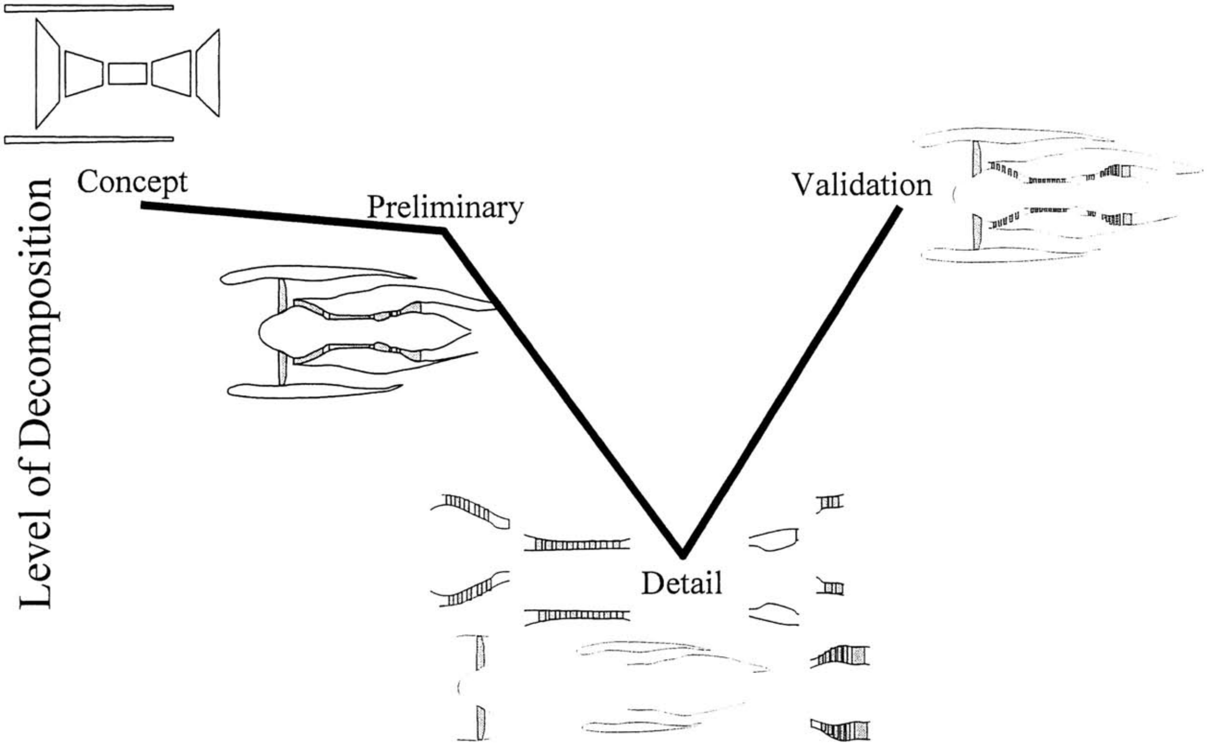


Figure 23: Levels of decomposition examined during product development stages.

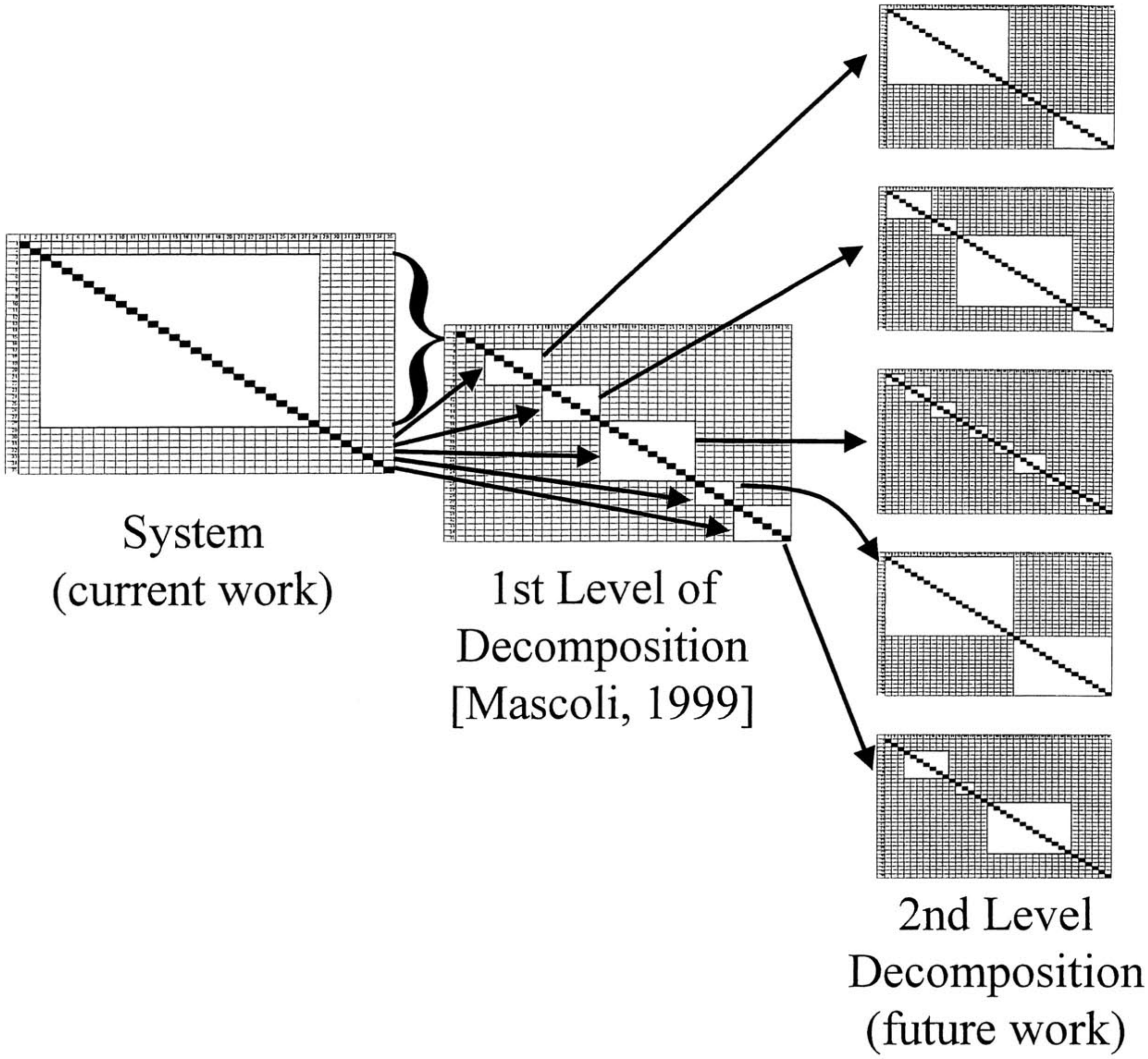


Figure 24: Hierarchical set of DSMs with ever increasing detail.

Iteration in a more integrated level of decomposition than is currently being worked, e.g., at the system level during detailed design, is both expensive and time consuming; however, many times it is both desirable and necessary to ensure larger problems do not occur even later in the product development process. Within P&W, the development of a program plan that includes iteration at the system level has historically been accomplished in an *ad hoc* manner if it occurred at all. As the business focused on reducing the product development time and costs, iteration during the detailed design phase was replaced by simulation and iteration in the preliminary design phase. While this does decrease costs and product development time, without the proper tools and risk mitigation planning, it could potentially result in design issues being uncovered during validation where all changes are both very expensive and time consuming (which result in customer satisfaction and public relations nightmares). While risk mitigation during program planning has been occurring within P&W for many years, up until recently, there has not been a formalized process. Within the last year, a formalized process for risk reduction during program planning has been established within P&W. The P&W process does not utilize DSM, but rather explicitly accounts for the interdependencies in another manner [see Zeidner and Wood, 2000 for a public discussion the process].

Utilization of a formalized risk reduction process has resulted in system level iteration cycles being inserted to program plans. While the specifics cannot be discussed in this work, the iteration cycles inserted have been of the inter-module type (the area of the "glue" engineer discussed by Mascoli) as well as the system-level type (the area of the aerothermal cycle). This risk mitigation planning now occurs during preliminary design and is inserted into the program plan prior to the start of detailed design. Risk identification and mitigation within modules occurs, but it is not currently part of a formalized process and still occurs on an *ad hoc* basis.

The detailed design phase at P&W is represented well by the work by Mascoli (although development of a DSM with increased level of detail for each module would enhance the representation, this is left for future work), Figure 25[3]. The steps that occur during the detailed design stage may be generalized as follows:

1) Obtain system-level requirements from the preliminary design phase
2) Develop requirements and identify risks for module to meet 1)[4]
3) Complete detailed design and develop integrative module parameters
4) Validate component design through production of prototype parts

Steps 1-3 occur within a parameter DSM. Step 4 occurs in preparation for system validation. A schematic of the detailed design DSM is shown in Figure 26. The end of detailed design and the start of validation is not as well defined within P&W as these stages always overlap. This is in part due to the desire for the detailed requirements to remain flexible as long as possible. The Level IV product review essentially ends Detailed Design for a component; however, changes to occur following the review. Throughout the Detailed Design and Validation stages, the program plan in continually updated to both reflect the current status of the program and to insert additional risk mitigation efforts that are identified during these stages.[5]

Assumptions for Validation

Validation of gas turbine engines consists of measuring parameters on complete engines in a series of tests that are defined by the FAA and JAA. The passing of these tests allows for certification that the engines are safe to fly on an airframe. The FAA and JAA testing does not

---

[3] Figure 22 is taken directly from the work of Greg Mascoli. The author would like to thank Greg for permission to use his work.

[4] Historically at P&W, the module requirements are decomposed into two sets of requirements: the component requirements (design), and the product requirements (manufacturing). These requirements are approved by module management in Level III and Level IV Design Reviews.

[5] The program plan is allowed to change during the detailed design and validation stages as long as the certification date and the budget are maintained. If either of these must change for the worse, a senior management review of the reasons is required.

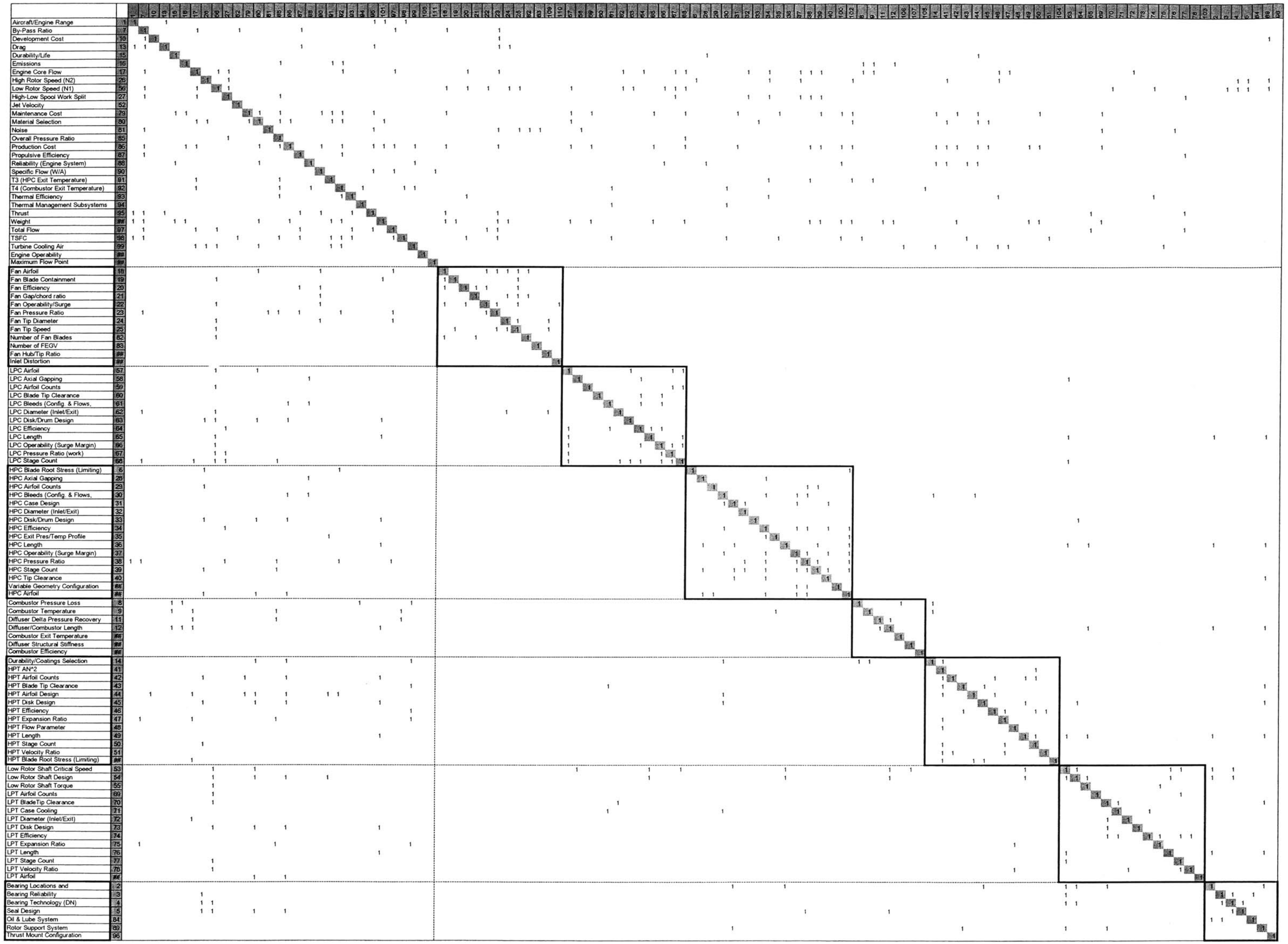


Figure 25: DSM of detailed design for gas turbine engine [Mascoli, 1999].

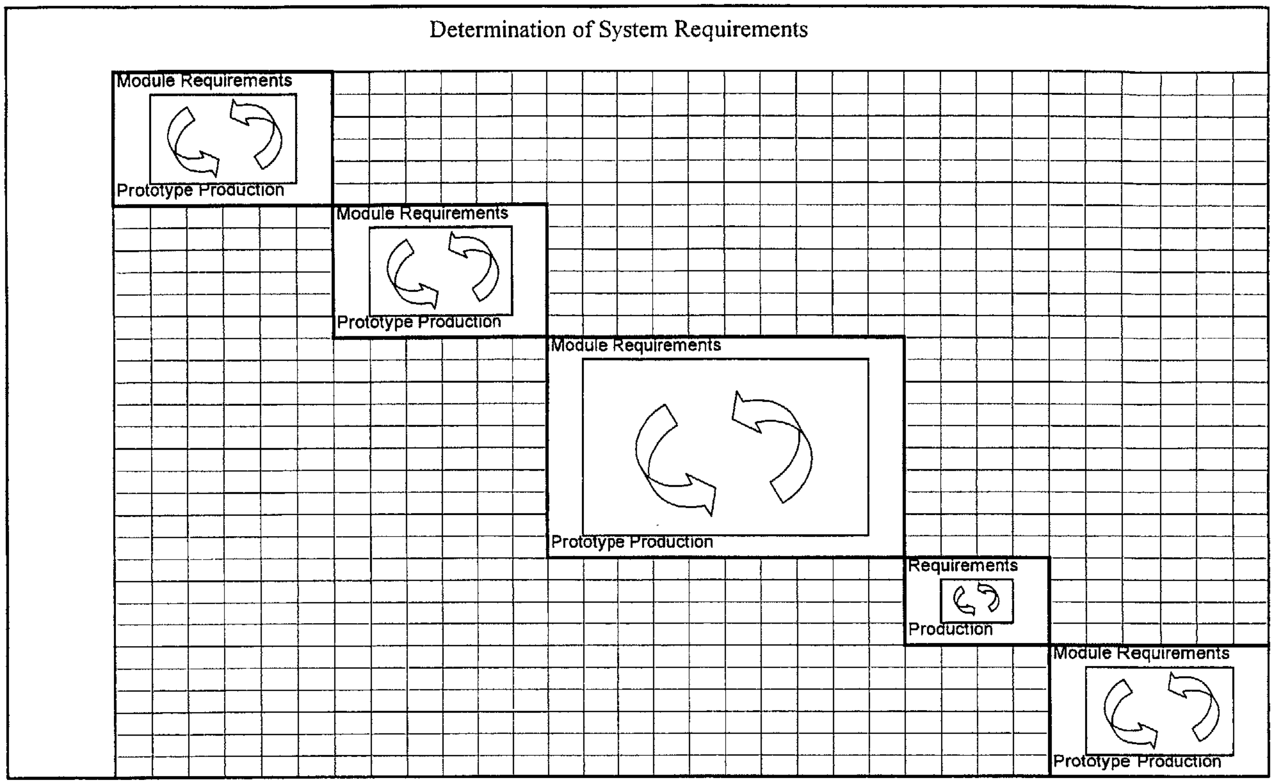
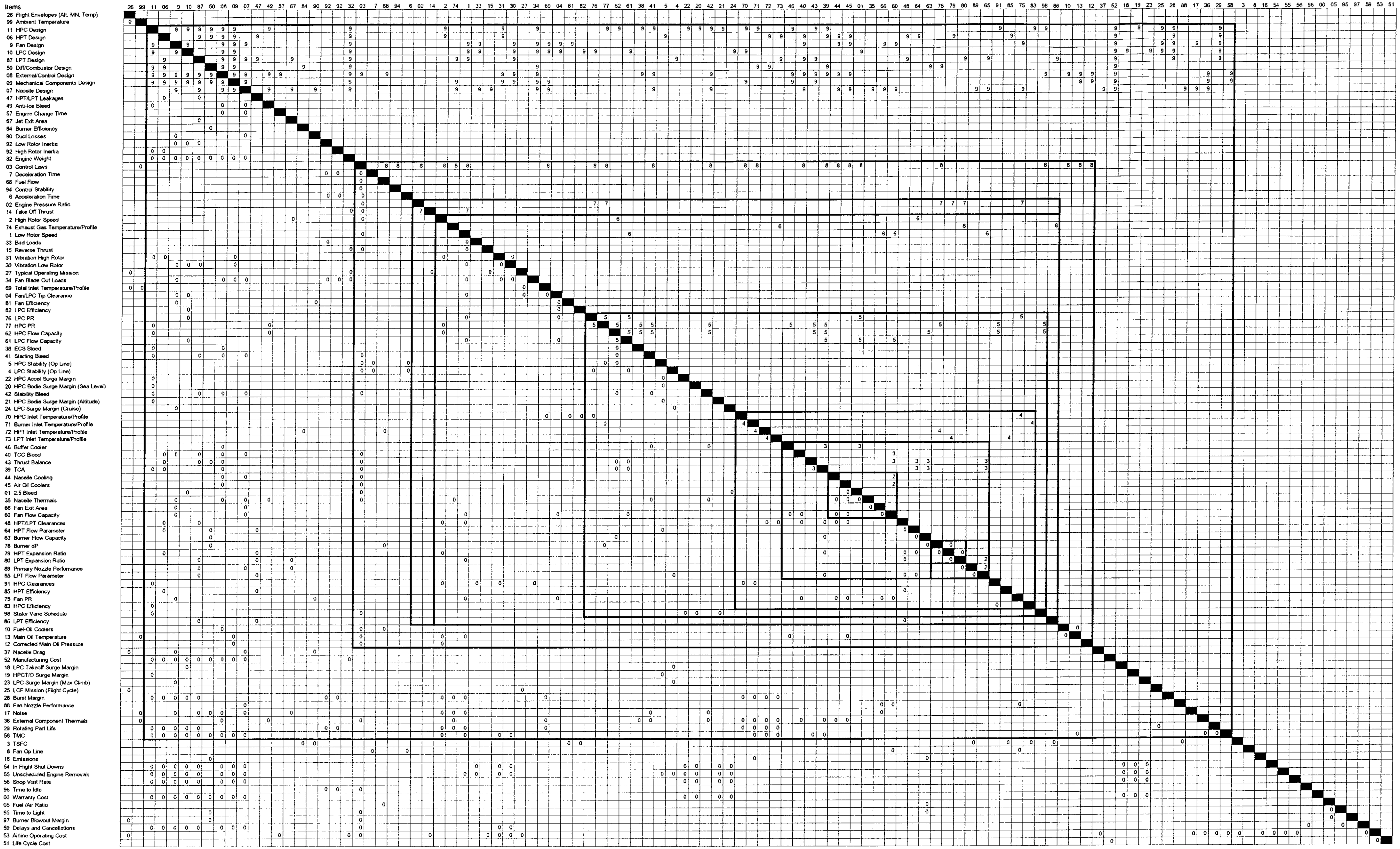


Figure 26: Schematic DSM for detail design.

require superior fuel consumption, low maintenance cost or other customer driven requirements. Thus, many of these parameters are also measured during the validation of the engine. For several of the highly integrative, customer requirements, no direct measurement is possible prior to years of field experience. For example, the rotating part life is not determined during validation testing do to the length of time and expense to directly measure it. In order to "validate" these parameters, P&W will measure many of the parameters that are inputs to and/or control the integrative parameter. P&W then states that given the input parameters were as expected and the quality of the estimation procedure developed over P&W's history is good, the integrative parameter has a particular value. In this way, P&W validates all parameters whether they are measured directly or indirectly. During validation P&W measures thousands of parameters, many of which are parameters utilized in the DSM of the present work.

Examination of the validation phase in terms of a DSM, the assumptions (or requirements) are the hardware output from the detailed design phase. To proceed through the validation phase, many aerothermal and mechanical parameters are measured and then compared to the system-level requirements developed during preliminary design and the component requirements developed during detail design. With two major and several minor exceptions, many of the parameters in the system-level DSM developed in this work are measured early on the first prototype engine that arrives in the test stand. The two exceptions are the detailed performance parameters (TSFC and module efficiencies) and the operability parameters (the surge margins). Although the performance parameters are roughly measured on the first engine, validation of these parameters is accomplished on a highly instrumented, dedicated performance engine. Similarly for the operability parameters, the operating line is measured on the first engine, but the surge line is not measured (hopefully) and the surge margins validated until a dedicated engine. Although several of the component requirements are validated on the first engine, most of the detailed module requirements are validated during a dedicated stress and telemetry engine for each core. The minor exceptions to the measurement and validation of the system-level requirements are due to the parameters that depend upon the detailed validation of modules (and therefore the dedicated engines) prior to their validation. Upon validation of the modules, these parameters are then integrated and validated as a whole. A validation DSM is shown in Figure 27.



Items
26 Flight Envelopes (Alt, MN, Temp)
99 Ambient Temperature
11 HPC Design
06 HPT Design
9 Fan Design
10 LPC Design
87 LPT Design
50 Diff/Combustor Design
08 External/Control Design
09 Mechanical Components Design
07 Nacelle Design
47 HPT/LPT Leakages
49 Anti-Ice Bleed
57 Engine Change Time
67 Jet Exit Area
84 Burner Efficiency
90 Duct Losses
92 Low Rotor Inertia
92 High Rotor Inertia
32 Engine Weight
03 Control Laws
7 Deceleration Time
68 Fuel Flow
94 Control Stability
6 Acceleration Time
02 Engine Pressure Ratio
14 Take Off Thrust
2 High Rotor Speed
74 Exhaust Gas Temperature/Profile
1 Low Rotor Speed
33 Bird Loads
15 Reverse Thrust
31 Vibration High Rotor
30 Vibration Low Rotor
27 Typical Operating Mission
34 Fan Blade Out Loads
69 Total Inlet Temperature/Profile
04 Fan/LPC Tip Clearance
81 Fan Efficiency
82 LPC Efficiency
76 LPC PR
77 HPC PR
62 HPC Flow Capacity
61 LPC Flow Capacity
38 ECS Bleed
41 Starting Bleed
5 HPC Stability (Op Line)
4 LPC Stability (Op Line)
22 HPC Accel Surge Margin
20 HPC Bodie Surge Margin (Sea Level)
42 Stability Bleed
21 HPC Bodie Surge Margin (Altitude)
24 LPC Surge Margin (Cruise)
70 HPC Inlet Temperature/Profile
71 Burner Inlet Temperature/Profile
72 HPT Inlet Temperature/Profile
73 LPT Inlet Temperature/Profile
46 Buffer Cooler
40 TCC Bleed
43 Thrust Balance
39 TCA
44 Nacelle Cooling
45 Air Oil Coolers
01 2.5 Bleed
35 Nacelle Thermals
66 Fan Exit Area
60 Fan Flow Capacity
48 HPT/LPT Clearances
64 HPT Flow Parameter
63 Burner Flow Capacity
78 Burner dP
79 HPT Expansion Ratio
80 LPT Expansion Ratio
89 Primary Nozzle Perfomance
65 LPT Flow Parameter
91 HPC Clearances
85 HPT Efficiency
75 Fan PR
83 HPC Efficiency
98 Stator Vane Schedule
86 LPT Efficiency
10 Fuel-Oil Coolers
13 Main Oil Temperature
12 Corrected Main Oil Pressure
37 Nacelle Drag
52 Manufacturing Cost
18 LPC Takeoff Surge Margin
19 HPCT/O Surge Margin
23 LPC Surge Margin (Max Climb)
25 LCF Mission (Flight Cycle)
28 Burst Margin
88 Fan Nozzle Performance
17 Noise
36 External Component Thermals
29 Rotating Part Life
58 TMC
3 TSFC
8 Fan Op Line
16 Emissions
54 In Flight Shut Downs
55 Unscheduled Engine Removals
56 Shop Visit Rate
96 Time to Idle
00 Warranty Cost
05 Fuel /Air Ratio
95 Time to Light
97 Burner Blowout Margin
59 Delays and Cancellations
53 Airline Operating Cost
51 Life Cycle Cost

Figure 27: System Level DSM with assumptions for validation.

Generalizing the P&W example results in steps for validation as follows:

1) Obtain module hardware and develop system prototypes
2) Measure parameters as possible through system testing
3) Determine integrative parameter values
4) Validate parameters meet all requirements, mitigate risks identified

A schematic validation DSM is shown in Figure 28. Risk identification and mitigation occur continuously during validation. Also during validation (or earlier) a plan of support of the engines in the field is developed. This includes any issues that arise during validation that can be resolved following certification (e.g., a cyclic life shortfall in a rotating part).

Deliverables from the validation phase are measurements (or calculation from measurements) of all the system level and component level requirements as well as certification by the FAA and JAA.

Summary

A system-level DSM has been developed in this work for a gas turbine engine. It differs from earlier work in that it was developed from a high level system engineer's (or chief engineer's) perspective. As such, the subsystem (module) designs of a gas turbine engine are represented by single parameters. It was found that most of the issues that the system engineers work with on a daily basis are not represented in the DSM directly. Instead it was found that chains of direct interactions (i.e., indirect links), represent most issues dealt with on a daily basis. From discussions and examples found when working with the system engineers, most issues are linked through three or more direct interactions (although no example was found to have more than six links).

Given the indirect nature of the DSM and the interdependencies of the parameters, it was of no surprise when initial attempts at partitioning the matrix produced few results. In order to develop a better understanding of the DSM, this work examined what parameters are assumed and when they are assumed. The nature of the assumptions made and its interrelationship with the requirements were examined as a function of time within the product development cycle. It was discovered that a single DSM used multiple times could represent the majority of the product development cycle (conceptual design, preliminary design, and validation). The exception is detailed design where the system-level DSM is viewed as requirements and more detailed DSMs are required for each subsystem being designed. Combining the DSMs for each stage into a single matrix is schematically represented in Figure 29. Note that in this context each parameter is actually a task to determine the parameter value and not the parameter itself.

While the schematic DSMs show rearrangement of the parameters in the DSM from a high live view, Figure 30 compares the sequencing of each of the DSMs to the order in Figure 18, the one order of assumptions that could occur during conceptual design.

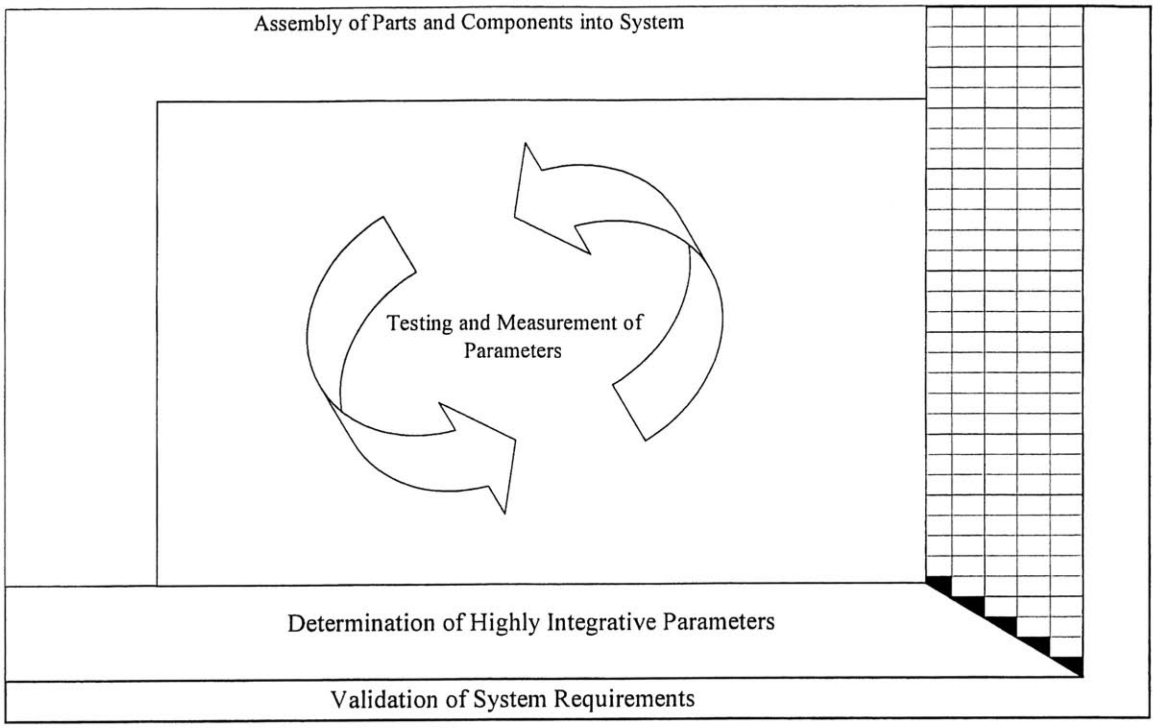


Figure 28: Schematic DSM for the validation phase.

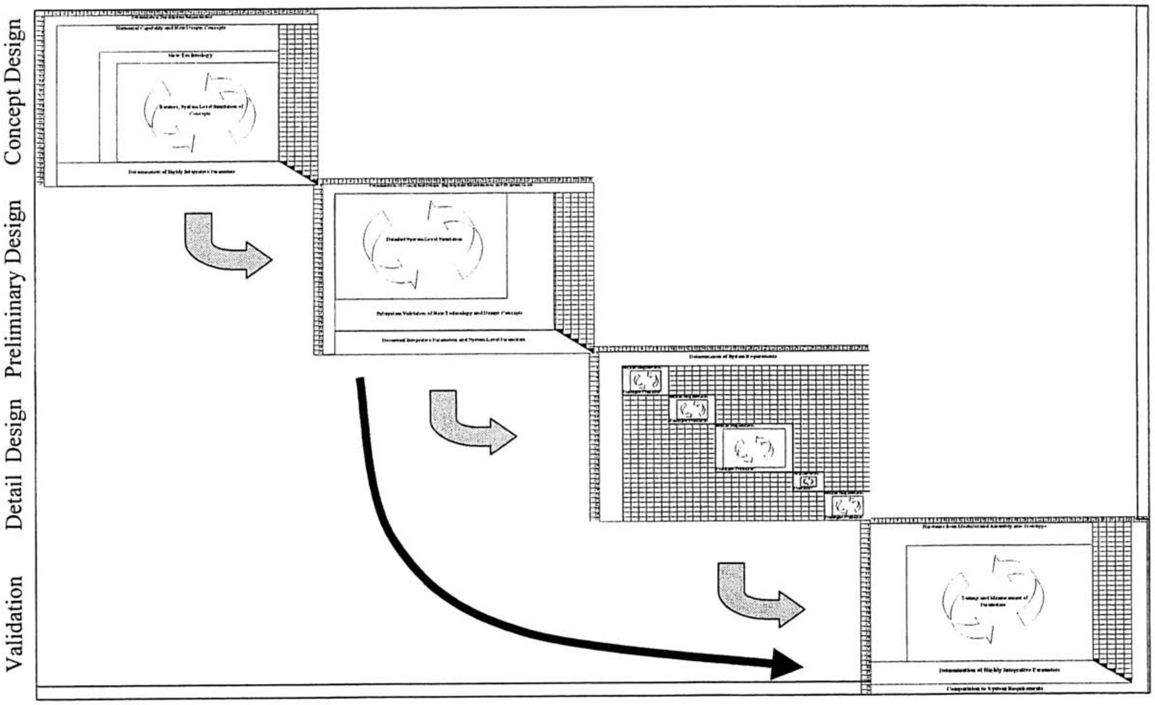


Figure 29: Schematic DSM for the entire product development cycle.

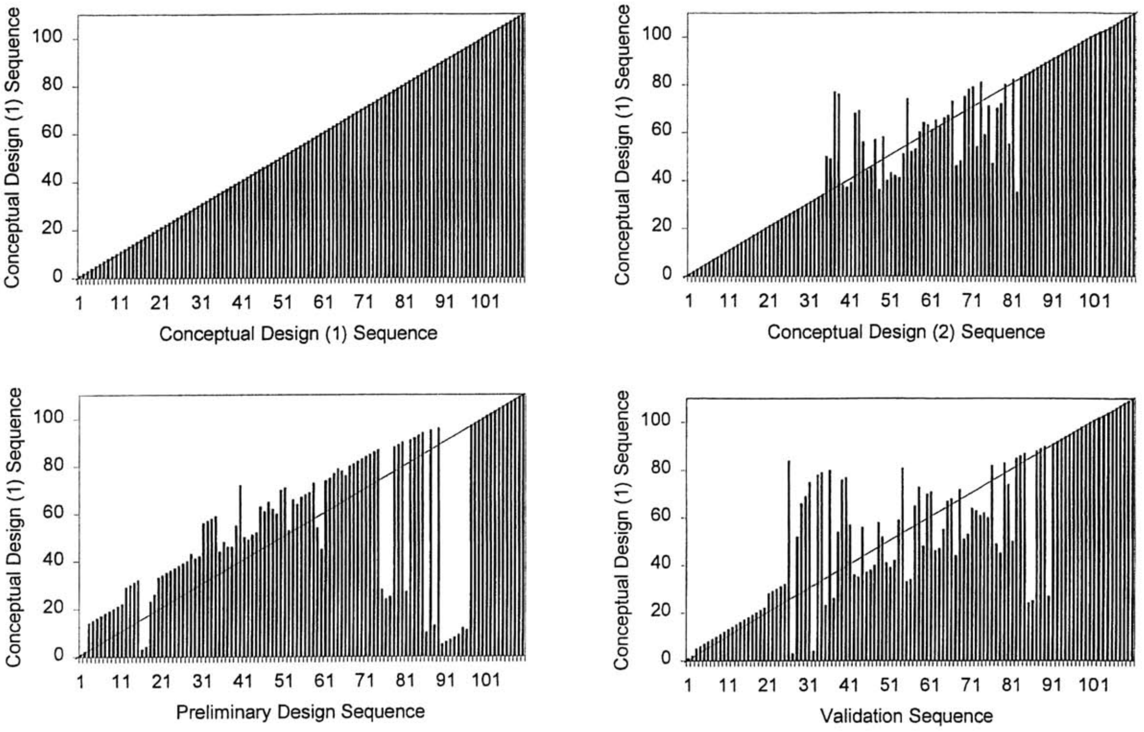


Figure 30: Comparison of sequences of Conceptual Design, Preliminary Design, and Validation.

CHAPTER 5: DISCUSSION

This chapter has been divided into two sections. First, the implications of the research for Pratt & Whitney are discussed. In this section are several discoveries that are not conclusions that could be drawn from the work discussed in the prior chapters. However, the path of research often leads one to discover items that should be discussed (thus, their appearance in this chapter). In the second section, the specific P&W recommendations are generalized and other implications discussed.

Implications for Pratt & Whitney

Requirements Documentation

The requirements documentation at P&W, the PS/CRD, has evolved into a quality set of documents. The suggestions given here will only serve to enhance the documents and streamline their implementation and flow-down. Five recommendations are suggested by the current research.[6]

First, include the Design Table as part of the PS/CRD. The design table formalizes the aerothermal requirements at the system level and is used by the IPTs to design their components. Having two separate sets of requirements only serves to confuse the IPTs. This is especially true when the aerothermal design is undergoing fast iteration for various scenarios that could develop during product development and field service. The inclusion of the Design Table in the PS/CRD

---

[6] The author had the opportunity to implement these recommendations in the PW4000-100 PS/CRD and will subsequently propose a revised SLP where these suggestions may become part of the formal P&W procedure for the development of a PS/CRD.

will help clarify the requirements for the design of an engine. Subsequent release of a new Design Table would force the PS/CRD to be updated and a holistic set of requirements released simultaneously. With the Design Table as a separate document, conflicts between the Design Table requirements and the PS/CRD are much more likely to exist. The integration of the Design Table with the PS/CRD will also encourage more interaction and collaboration between the PSA and SD&CI organizations.

Second, insert the risk reduction and mitigation planning into the PS/CRD. Communication of the methods and reasons behind the development of the program plan will serve to drive the understanding and appreciation of the risk through the product development organization. Tracking of the progress of the program against the planned risk waterfall will also serve to highlight the success and difficulties in a product development cycle. This helps to drive understanding from the product development team to both upper management and the detailed design engineers at a geographically dispersed locations.

Third, eliminate the Design and Validation Process sections of the PS/CRD. This section is redundant with System Level Procedures (SLPs) controlled through ISO9000 certification. The PS/CRD should only contain requirements and not procedures by which to accomplish the requirements. Only in cases where a decision is made to violate certain procedures should a procedural requirement be included.

Fourth, develop a minimal set of times for a PS/CRD to be released and updated. The PS/CRD is initially released at launch of the detailed design, but its control and maintenance has not been a focus of the SD&CI organization. At a minimum, the PS/CRD should be updated and rereleased following the testing of the first engine, upon certification, following flight test, and after three years of service. For these last two releases, the focus of the PS/CRD should be on support of the engine in the field (i.e., specifics about the product development cycle should be removed or restated).

Finally, the PS/CRD should be updated and released at the initiation of any large change to a module (i.e., a derivative design or a FAA major change). As has been the theme throughout this work, it is not possible to change a module in any significant way without altering the entire system. All PS/CRDs should be a complete set of requirements. This is especially true for derivative models or module changes as it reinforces the requirements to maintain commonality. Focusing a PS/CRD too much on only the desired hardware changes risks missing an interaction that may prove to be important. The issue here is resource constraints for derivative programs that do not have a baseline PS/CRD. Management will be required to trade resources required to develop of a holistic set of requirements with the potential for confusion and error caused by incomplete or conflicting requirements. Fortunately, as time progresses with all new engines having PS/CRDs, this situation will occur less and less.

Organizational Impacts

Since one of the abilities of the DSM is to align organizations with the associated architecture and program plan, the DSM in this work examined to see if any changes would be beneficial. This analysis does suggest some realignment within the system engineering organizations.

First, the DSM does reinforce that PSA (the aerothermal parameters) and SD&CI (the design and structural parameters) may be separate organizations at some level; however, the interconnections between the parameters suggests that a single chief engineer should be present over each product development cycle. The current lack of a single person responsible for the entire technical delivery of the engine relegates the authority for any conflicts to the Program Office and its Program Manager. While on some programs, the Program Office has informally designated one of the three systems engineers the role of Chief Engineer, it does not always occur and in these cases the program office makes many technical decisions. This upsets the balance between forces pulling towards faster/cheaper and the forces pulling towards better (when they are in conflict). When this balance is upset, it may result in a less than desirable product. A Chief Engineer role to balance the Program Office and Program Manager role should be put in place. The single chief engineer suggestion is in direct agreement with a conclusion of Mascoli and agrees with the spirit of Rowles suggestions. (Rowles did not specifically address the three system engineering organizations, but rather referred to them in combination as system engineers.)

One difficulty in implementing this role would be the development of single person with all the skills necessary to balance the needs of the organization. While these talents do not exist in enough people to implement this across P&W, with strong deputies in each systems engineering specialty, a person could grow into this role. One thing that is a given is that without the Chief Engineer role present in the organization, little incentive exists for people to obtain the broad expertise necessary for such a role.

One comment on the current work is that the PD&V organization is not represented in the DSM other than being responsible for the measurement of the parameters during validation. While this may lead one to suggest that PD&V is not of sufficient importance to the product development to warrant a system engineering organization, the author believes that this omission is due to the focus of this work on the parameter-based DSM. If a task-based DSM would have been completed, it may have suggested that PD&V is the only system engineering organization necessary. Thus, the author believes that each of three system engineering organizations serves a valuable role within P&W. The final glaring omission in the present study is the presence of a Manufacturing Systems Engineering organization. At the present time, it is unclear to the author as to the need and value created by this organization. This uncertainty is due to the author's lack of knowledge of this organization and its role within P&W. The study of this organization and its role would be a fruitful area for future thought and work.

Within the three system engineering organizations, the DSM suggests that the secondary flow organization may align better with PSA than with SD&CI. As can be seen in Figures 18, 19, 21, 25, and 27, the parameters that are the responsibility of the secondary flow organization (i.e., bleeds and leakages) are an integral part of the aerothermal design that is PSA's responsibility.[7] There are two main lines of communications within the secondary flow organizations: one based upon the aerothermal design already discussed and the other based upon the cooling and temperature impacts on the structural integrity of the components. The author assumes that in the development of the current organization it was thought that

---

[7] The large block in the middle of all the P&W specific DSMs is the aerothermal design of the engine. This region is the system-level simulation in the schematic DSMs.

communication along the structural impacts was more important and thus, the inclusion of the secondary flow team within SD&CI. The present work suggests that the alignment be changed not because of the aerothermal design being of more importance, but that closer alignment with PSA will occur without the loss of alignment with the structural organizations within each module. In the current organization secondary flow communicates structural issue to the module center teams. Little information flow is necessary to SD&CI and all SD&CI has been is secondary conduit for information flow (especially through the PS/CRD). Thus, little will be lost by removing secondary flow from SD&CI and significant alignment will occur with PSA.

Simulation vs. DSM

At P&W and in general within the gas turbine industry, system-level simulations of the entire gas turbine are fairly common and very detailed. These simulations have been developed over many years through experience at the companies as well as by the extensive research completed at universities across the world. A gas turbine physics and parameter interdependencies are well understood (at least for the aerothermal cycle). The companies use these simulations extensively during the design and development of engines and have even considered eliminating the development testing due to the capabilities of the simulations to design components "right" the first time. (Actually it is not the first time, it is the first time in producing physical components only. Prior to making components, many iteration cycles occur in the virtual world).

The implications of having these high quality system level simulations is that most of the interdependencies are already accounted for and the DSM is of less importance than in industries where these types of simulations and knowledge of the interdependencies do not exist. The most fruitful area for DSMs would then be in the areas where detailed, physics-based simulations do not exist. Another areas where a DSM may be helpful is at the interfaces between the simulations.

Applications for DSM within P&W

Two areas are perceived to exist for application of the DSM tool within P&W. The first is in training an engineer in the intricacies of the interfaces within the gas turbine. Currently, only years of experience within the industry at the system level (or graduate education at universities like MIT) give an engineer the mental model that includes many of the interdependencies present within the gas turbine. The DSM may be a valuable tool for newly hired engineers to begin to understand the interdependence of the parameters. The young engineer would then be aware enough to ask for assistance in determining the exact affects of a relationship.

Second, the DSM could act as a quick reference guide to system engineers when dealing with individual design issues. It may serve as a guide as to whom to ask about the impacts of an engineering change.

General Implications of Work

Lessons from Pratt & Whitney

Requirements

Several lessons can be learned from the Pratt & Whitney experience. First, the development and documentation of system requirements should be started early in the product development cycle and continuously monitored and updated throughout the cycle. Establishing system-level requirements (actually documented assumptions) early allows the entire product development team to work from a single set of assumptions. Establishing the time phasing of the requirements development to the product development cycle helps keep many of the detailed requirements flexible while maintaining the system-level requirements necessary to meet the customer desires. Requirements are only established when a change in the parameter becomes difficult or time intensive. For processes that have changing customer desires a hierarchical ordering of requirements establishment will allow the teams to understand what can be changed easily and what would require the entire PDP to be restarted.

Maintaining the requirements document with periodic updates is essential to keeping the entire product development team working to the same set of assumptions. One note of caution is that as the frequency of updates to the requirements increases, communication of the updates becomes more and more important and time consuming (depending of course on the tools used). Thus, a balance is needed between having the most recent requirements available and the manpower required to communicate these requirements across the product development teams.

The requirements document developed should include all requirements for a particular level of decomposition. For example, all system level requirements should be present in a single location. Pratt & Whitney had requirements developed and published by organizational boundaries. This is not a preferred method for the requirements. This work suggests combining all the system requirements into a single document. (Note a document is not necessarily a paper document, it could be a single electronic file or database). Subsystem requirements should either be in the same document as the higher level requirements or maintain strong links such that as requirements change on one level of decomposition, the changes flow to the other levels within the product.

The present work suggests that the risk mitigation plans should be inserted into the requirements documentation. While the author believes this is the best approach for P&W, it is not the only approach for disseminating the risk reduction plans and iterations cycles that have been inserted into a project management plan. The need is for the product development team to have the risk identification and mitigation plans communicated to them. On many large product development projects, most of the IPT members are far away from the program planning. Thus, the communication of the risk mitigation plan within the program plan allows the individual team members to understand the strategy, reasoning and layout of the tasks they are charged with completing. This helps to align the product development team around the program plan. The updating and tracking of the risk levels within the program helps the individuals understand the progress against plan and be more likely to accept changes to the plan when the need arises.

Product Development and System Engineering Organizations

As companies strive to improve their product development cycles by decreasing the cycle time (faster), decreasing the costs (cheaper), and increasing the quality of the product (better), conflict often occurs between the faster/cheaper and the better. While organizations strive to find initiatives and improvement that will allow all three to occur simultaneously, many times the goals are in conflict. For these situations, an organization should strive to balance the forces that are driving to each goal. Developing an organization or culture that has one or two of the goals stronger than the other is a situation that has a high risk of failure. Striving for too low a cost risks getting a product to market on time with poor quality. Trying to get the product to market in the quickest manner will sacrifice cost and quality. A perfect product will never get to market at any cost. Thus in most situations, a balance must be formed between these goals.

For very large, complex products where hundreds if not thousands of individuals are working to bring a product to market, the work must be decomposed into smaller pieces to then be integrated together. Balancing the forces that pull the product to market is not a trivial task. Many organizations accomplish this task by developing matrix organizations where individuals are responsible to both functional organizations (where quality of work is usually the dominant goal) and to product organizations (where cost and speed to market are often the dominant goals). The balance is set by maintaining the strength and power of each organization. For large product development teams, a system engineering or chief engineering organization often resides at the top of the functional organizations. For such organizations, the balance must be maintained between the product organization and the Chief Engineer. In the authors view, P&W had upset the balance by dividing the Chief Engineers role between three separate system engineering organizations. While P&W management correctly observed that the system engineering has at least three specialties in their product, separation of the system engineering into three organizations led to a situation where balance was not maintained and the speed and cost goals were driven at the expense of product quality. Inserting a Chief Engineer by product would likely reinstate the balance.

The main lesson in organizational development using a DSM is that the balance of goals in the product development cycle must be maintained while working to align the organizations to the architecture and program plan.

System Modeling and Simulation

The gas turbine industry has a very mature set of system models and simulations that include most of the interdependencies present in a gas turbine engine. For industries like the gas turbine industry, the simulation and modeling allows for many iteration cycles to occur during product development. Often these iteration cycles are not even recognized by the majority of the product development organization. These simulations and models also allow for quick response to issues and problems that arise during the development cycle. However, one of the dangers in using the simulations and models is that the only individuals that understand the entire set of interactions are the people that develop the model. Often not even the users of the model understand the interactions until they have a significant amount of experience using the tools. This leads to an over reliance on the simulations and can lead to the slowing of product development when this organization gets overburdened with requests of "what-if" questions. (This is the positive side as at least individuals are asking about the interactions. Often people may not even know enough to realize there are interactions that they should be concerned about). This is where the DSM may play an important role in product development. The DSM can raise awareness and understanding of the interdependencies present in the product and release some of the this burden from the simulation organization.

For industries where high quality system level simulations do not exists, the DSM may play a much stronger role in assisting the system engineer during product development. This situation is almost always the case for innovative products and new architectures. It is also the case in mature industries where the system level interactions are not simple physics (as in the gas turbine), but a mixture of complex interactions that are not yet well understood from first principal physics (e.g., the automotive industry).

Assumptions, the Parameter DSM, and the Product Development Cycle

Assumptions that occur during product development are often made to break a known feedback loop such that the first cycle through the development of parameters may be completed. As was suggested earlier, these assumptions often become requirements that are then passed through to later stages of product development. During the current work, 3 of the 4 stages of product development examined are represented by a single system-level, parameter-based DSM. As each stage had different assumptions, the order in which the parameters were developed changed for each stage. For the detailed design phase, the system-level DSM did not provide enough information. More detailed subsystem level DSMs were suggested as being more appropriate for detail design.

Overall, the repetition of the use of a single DSM (with an expanded DSM for detailed design) and the relationship between the DSMs is shown in Figure 29. While a single example the relationships between the DSM and the product development cycles has been shown, the author believes that this repetitive use of a system-level parameter DSM occurs in many product development cycles. The P&W product development cycle appears to be typical of a large, complex development cycle of a physical product. Many companies have represented their product development cycle with a V (Figure 23). In this representation, the development cycle starts with high level work, decomposes and designs at a low, part level and then integrates all the parts together to work the total product again at high level (thus, the high, low, high, V representation). Many system engineering and product development texts also support this generic process where conceptual and preliminary design develop requirements at progressively more detailed levels. These are followed by a detailed design phase that generates the individual components and parts, and an integration phase that assembles the entire product and validates that it meets the goals and requirements [Ulrich and Eppinger, 1995; Rechtin and Maier, 1997].

As the product moves through the development cycle the assumptions made change. For the conceptual design phase, the details of the performance of a subsystem are assumed from historical values and changed based upon new design concepts put forth for a particular concept. Technology that could be inserted into the new product is then assumed. Based upon the historical and new technology parameters assumed, the performance of the system is developed. After many iteration cycles, each which develops a different conceptual product, one may select several of the concepts to complete the development of the highly integrative parameters (or one may chose to determine them for each concept depending upon the costs to determine these highly integrative parameters). Again see Figure 20 for representation of this in a schematic DSM. For conceptual design, performance heuristics and parametric design validate the product. At this point the product is a design that represents the entire product (although at a very sparse way).

As the design moves into the preliminary design phase the system level performance analysis is completed in significantly more detail (e.g., from a parametric or heuristic based evaluation to a detailed simulation based evaluation). Following the detailed definition of necessary system level parameters, the technology and design concepts are validated. This validation often occurs through independent subsystem testing. There may be several (but not usually too many) iteration cycles between the detailed system level performance parameters and the subsystem validation. At the completion of the preliminary design phase the system level performance parameters are often then defined as the system level (and possibly subsystem level) requirements. During preliminary design, validation of the system occurs through the system level performance simulation. See Figure 22 for the schematic DSM for preliminary design. At this point the product is a complete (although virtual) product.

The requirements are then passed through to the subsystems for the detailed design phase. The detailed design phase assumes the system level requirements from the preliminary design phase and the subsystem development occurs relatively independently. While there may be some iteration between subsystems, it is rare that the iteration cycle extends up into the system level performance parameters. From this discussion, it is obvious that selection of system level parameters is very important. The desire to keep options and design space as open as possible suggests that the options that can be kept open the longest are interactions and interdependencies that are within the subsystem or between a limited number of subsystems and do not alter the overall system in a major way. For example, the size and external interface of a power generator may be somewhat flexible, however, the output voltage and maximum current may be constrained very tightly. The deep dive into the details of the design lead to the DSMs to be greatly expanded with each subsystem potentially having its own matrix. (Note that the matrices must be interlinked as there will be interactions between the subsystems, Figure 24). See Figure 26 for a schematic how DSM represents detailed design. In detailed design, the validation of the design is the actual production of the entire subsystem.

Once the detailed designs are complete and the subsystems produced, the validation phase of product development must integrate and test the complete system. For validation, the assumptions (although very real physically) are the individual subsystems. The subsystems are assembled into a complete system and each of the system level performance parameters are measured. These measurements are then compared to the requirements developed in the preliminary and detailed design phases to assess whether the product meets the necessary goals of the program. Following measurement of the system level parameters the highly integrative (usually customer driven) parameters that cannot be measured are calculated. The DSM shown in Figure 28 can represent this phase.

For each stage of product development, the output of the stage resulted in being the assumptions and/or requirements for the following stage. Thus, the aerothermal parameters of conceptual design are the assumptions in preliminary design. The validated system-level parameters of preliminary design become the requirements of detailed design. The physical hardware of detailed design becomes the required pieces for assembly of the prototypes in validation. Within P&W, this passage of bulk information from one stage of product development to another is always accompanies by a management Design Review. In this case single events conclude one stage and start another.

Spiral vs. Waterfall Frameworks for Product Development

Product development of complex systems has traditionally been represented by a "waterfall" type of process where a serial set of stages occur to bring a product to market (Figure 2) [Rechtin and Maier, 1997]. In general, the PDP can be broken down into conceptual design, preliminary design, detailed design, and validation (see Figure 1). Within the last several years, a new framework has been developed within the software industry that instead of a waterfall development process, the software industry uses a "spiral" (Figure 3). The spiral framework is one that the product development process involves moving through the a series of tasks. The original spiral moved through four phases (Determine objectives, alternatives, and constraints; Evaluate alternatives, identify, resolve risks; Develop, verify next-level product; Plan next phases) [Boehm, 1988; McConnell, 1996]. Each cycle through the process results in a product with more functionality. Since then several slightly differing sets of phases have been suggested (Function, Form, Build, Certify) and (Requirements, Design, Build, Integrate/Test) as well as several others [Rechtin and Maier, 1997]. The current work addresses both types of phases, but uses the following set of phases (Requirements, Initial Design, Define Product, Integrate/Test) as a hybrid. The risk management and planning phases in the original work are discussed as appropriate to the phases. The following discussion argues that the PDP of P&W (and as a typical PDP, many complex physical system PDPs) is better represented by the spiral framework than the waterfall framework. Since many practitioners understand the waterfall framework and consider most hardware processes to be a waterfall framework, the following discussion focuses on how the PDP of P&W fits into the spiral framework.

Examination of the Conceptual Design Stage in hardware development shows that first many *requirements* are developed from the supersystem, historical designs, and new technologies and design concepts. The team then determines a set of *initial* system level parameters and identifies the risks involved in a particular concept. If the concept is deemed acceptable, the designer refines the parameters and *defines the product* (although still at a very high level). The details are often the result of heuristics and parametric calculations. The designers then *integrate and test* their conceptual design through simple system-level simulations and comparison to previous designs. At this point the product has made its first complete loop around the spiral (Figure 31). The stages of the spiral can also be represented as part of the DSM (Figure 32). The program plan is then updated and a review with senior management held. At the end of Conceptual Design, the product is a high level virtual prototype.

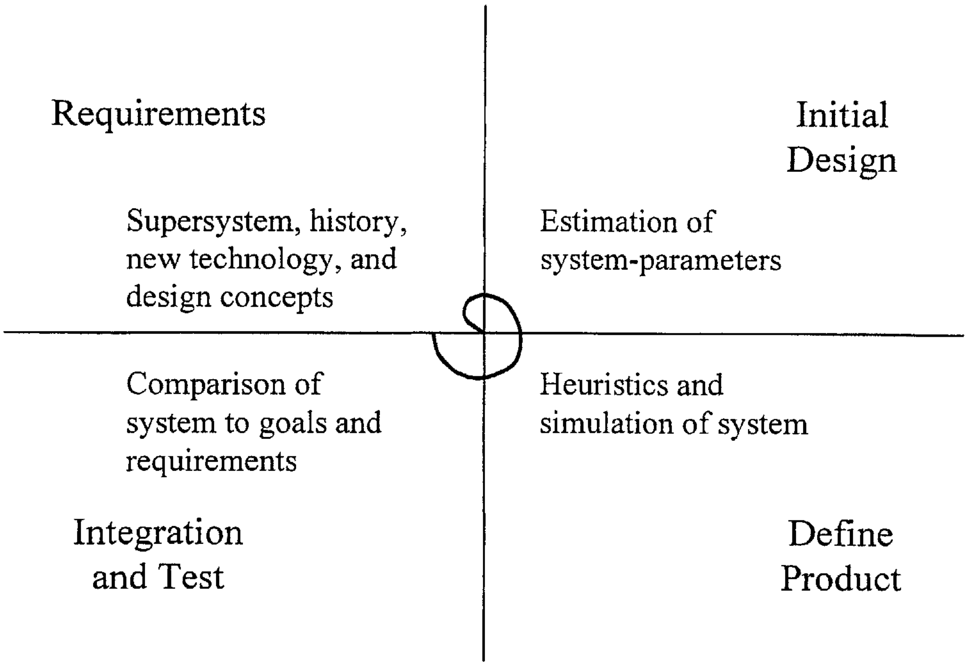


Figure 31: Spiral representation of the conceptual design stage.

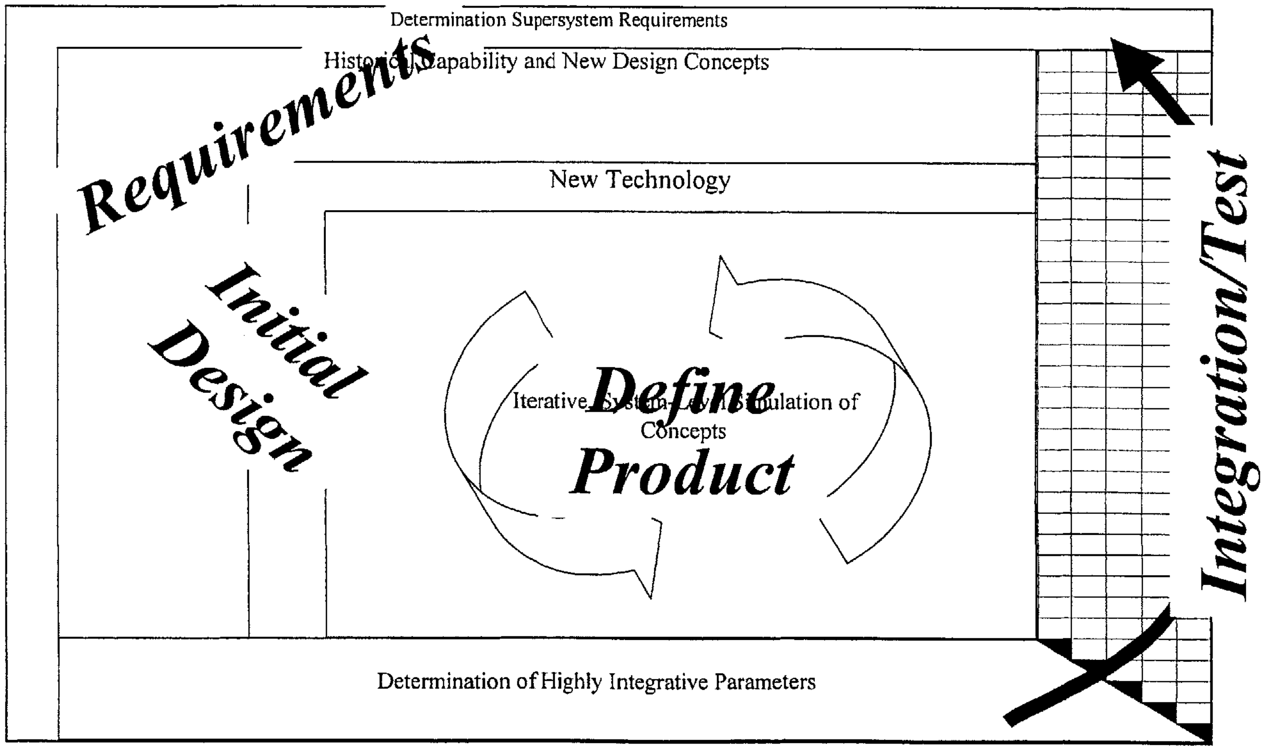


Figure 32: Schematic conceptual design DSM with labels for spiral development.

The Preliminary Design Stage accepts the *requirements* developed in the Conceptual Design Stage and proceeds the *initial design* of the auxiliary systems and estimation of the additional detail for current subsystems. A system-level simulation is often developed. This simulation is exercised to increase the level of *product definition.* This is followed by the *integration and testing* of the new technology and design concepts in individual subsystems. Several loops around the phases may occur during preliminary design, especially if one or more of the technologies or design concepts are not validated as meeting the assumptions. See Figure 33 for representation of the spiral Preliminary Design Stage. The spiral can also be represented in the schematic DSM (Figure 34). At the end of Preliminary Design, the product is a fairly complete virtual, system-level design and set of requirements. A detailed risk management activity is completed during Preliminary Design. Completion of the Preliminary Design Stage occurs by updating the program plan and holding a senior management review.

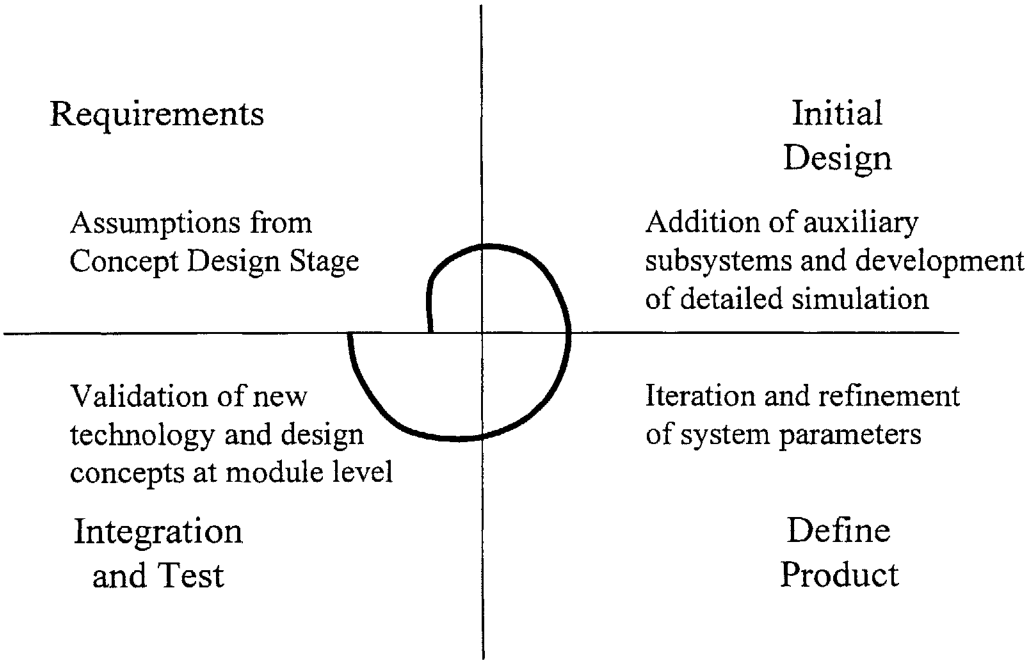


Figure 33: Spiral representation of the preliminary design stage.

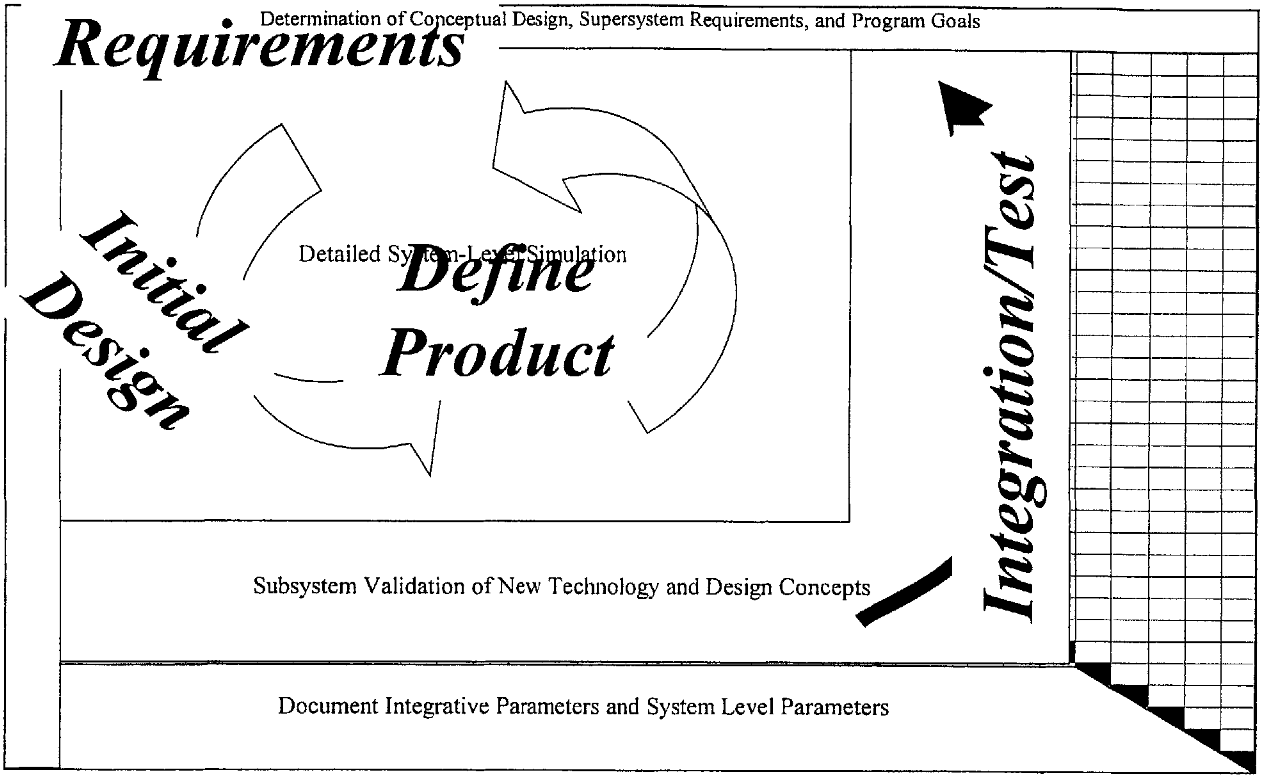


Figure 34: Schematic preliminary design DSM with labels for spiral development.

Separation of Detailed Design Stage into phases proceeds along similar arguments. Each module accepts the system-level *requirements* developed during Preliminary Design and then establishes module and component-level *requirements*. The team develops an *initial design* based upon the requirements and historical designs and then proceeds to *define the product* (components) to is highest level of fidelity. At this point a detailed 3-D model or print is completed. *Testing* of the design occurs as the part is manufactured into a physical entity. For detailed design each subsystem makes its own path through the spiral. At the end of detailed design the product exists in its entirety (often in both the physical world and as a fully developed 3-D solid model in the virtual world). Figure 35 represents the spiral of the Detail Design Stage. Representation in the form of a DSM of the spiral during detail design is given in Figure 36. Risk identification and mitigation occurs within each module on an ongoing basis. Detail Design

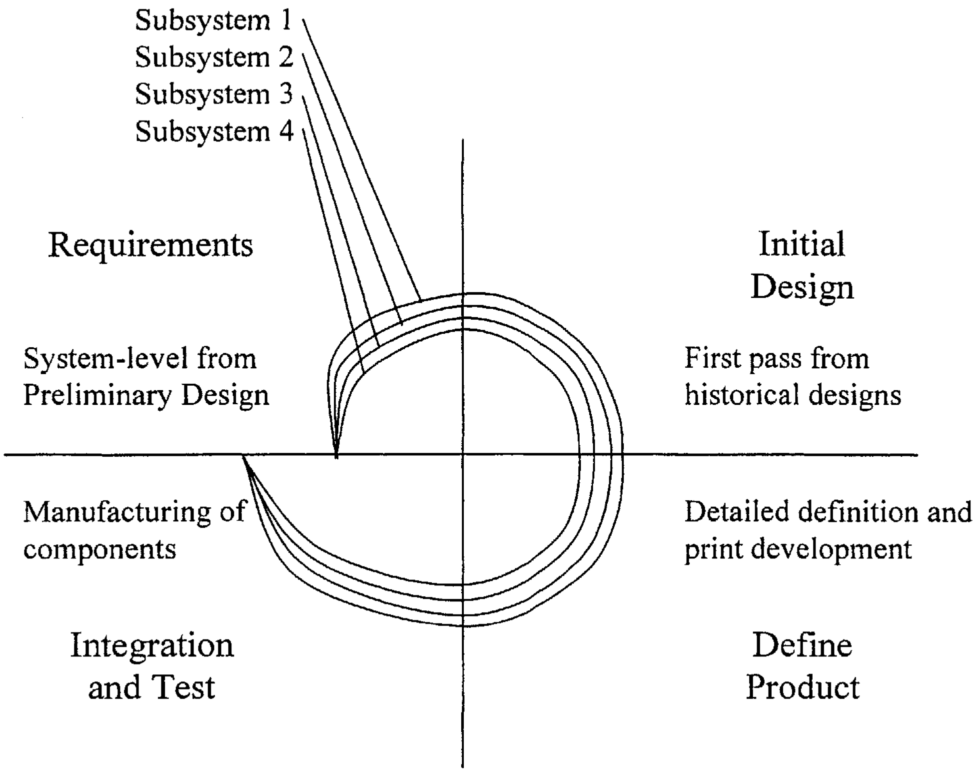


Figure 35: Spiral representation of the detail design stage.

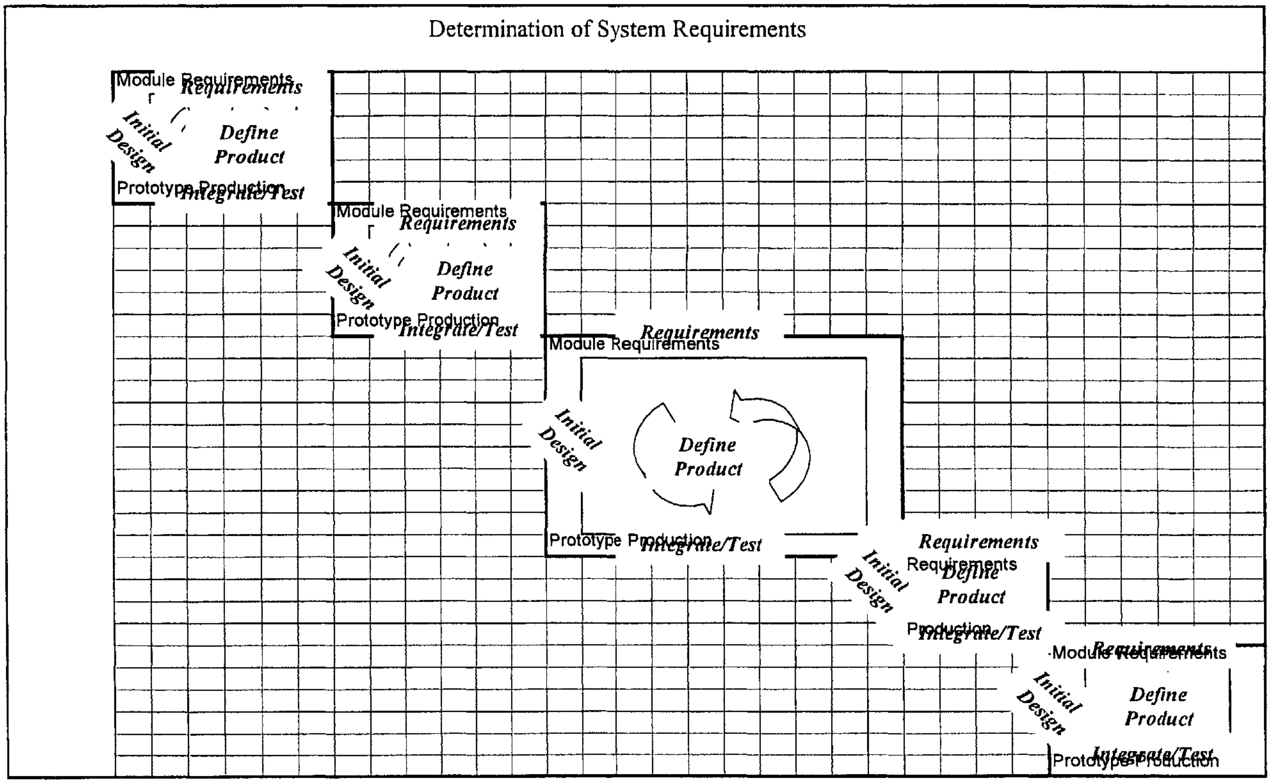


Figure 36: Schematic detail design DSM with labels for spiral development.

ends during the Validation Stage and is completed with final input for the module into the program plan and holding a management review.

Finally, the Validation Stage of product development receives is *requirements* from the FAA and JAA as well in the form of the physical hardware delivered from each module. The *initial design* (actually the assembly methods for the first physical prototype) is completed prior to the first prototype going to test. The product then proceeds through a set of detailed tests that fully define the performance and operation of the prototype. *Integration and test* occurs when the measured values are compared to the requirements and goals for the product to either validate the product or initiate and additional iteration cycle to mitigate the risk of releasing a substandard product. This completes the last cycle of the spiral development process. See Figure 37 for the representation of Validation in the spiral framework. Figure 38 is a schematic representation of spiral validation in a DSM. As the Validation Stage is ending, the program develops a plan for field support of the product. This spiral actually continues through field support and retirement, but that is beyond the scope of the current work.

The complete spiral representation of the waterfall product development cycle is shown in Figure 39. Note that in this schematic, 4 cycles around the spiral are complete. This need not necessarily be the case as any of the phases may proceed around the spiral more than once, or two or more stages may be combined. (The combination of stages occurs most frequently with the conceptual and preliminary design phases at P&W, especially for derivative designs).

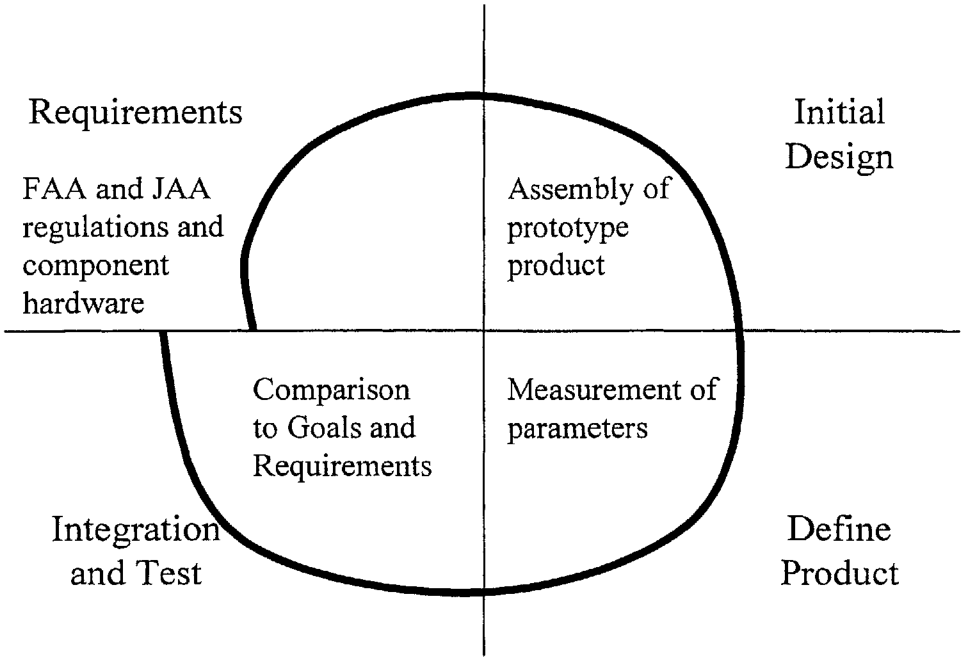


Figure 37: Spiral representation of the validation stage.

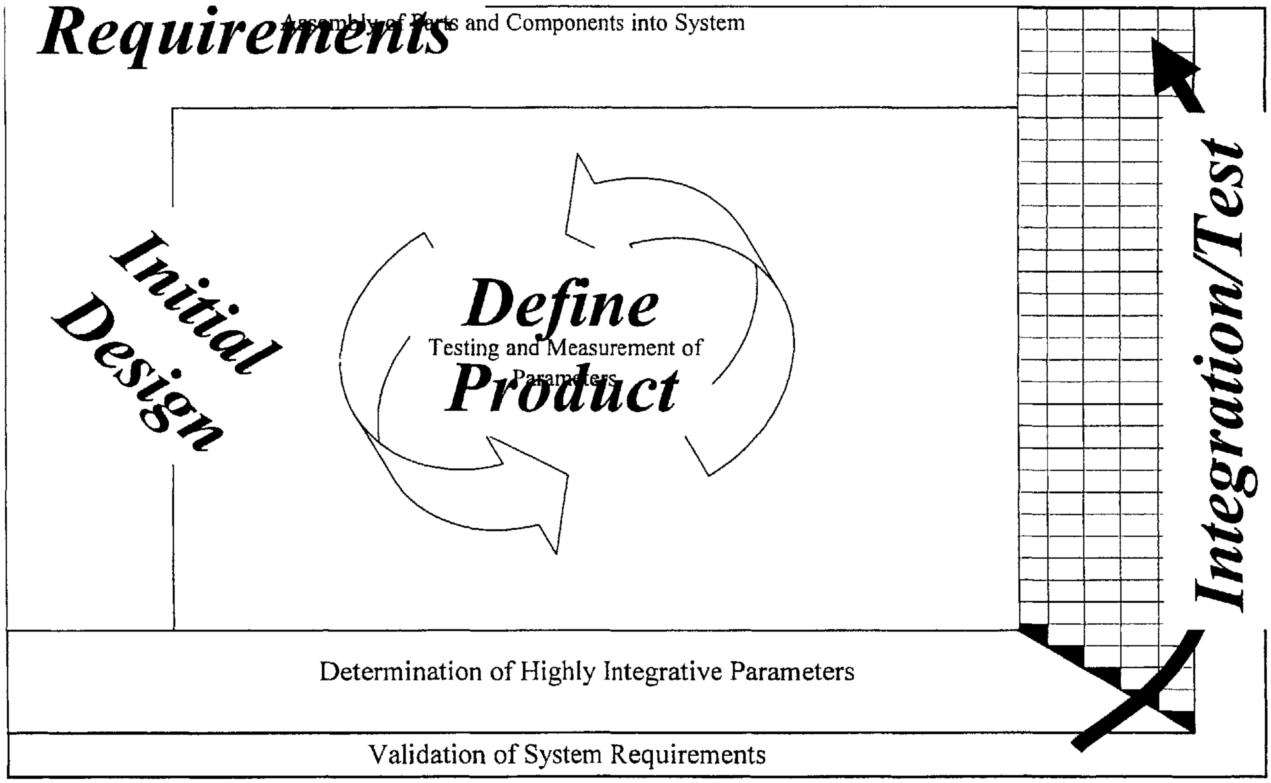


Figure 38: Schematic validation DSM with labels for spiral development.

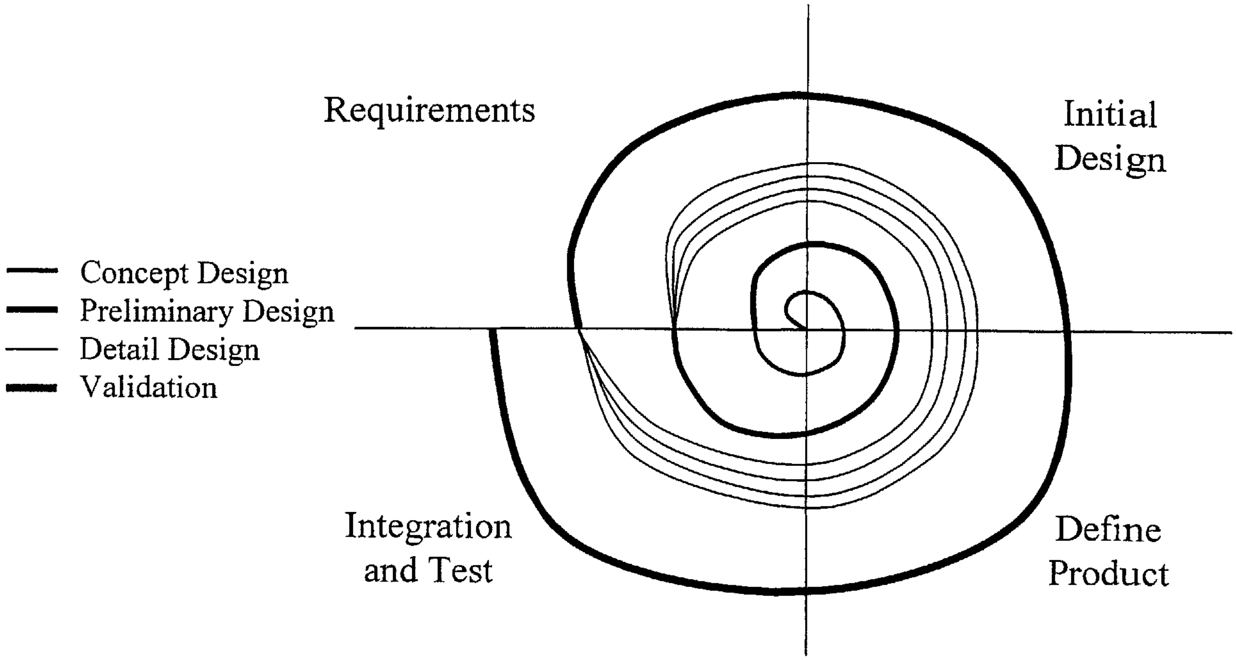


Figure 39: Schematic spiral for a waterfall product development process.

As the above discussion shows, the P&W PDP is very well characterized by the spiral framework of product development. Although it may also be viewed as a waterfall process, the current practices using continuous risk management, repeated program planning, and the repeated progression through increasingly detailed phases all support the P&W PDP being more of a spiral development process than a pure waterfall. In addition, the hierarchical method of setting requirements to maintain flexibility in the design for as long as possible (even into Validation) suggests more of the spiral framework. As this work has previously noted, the generic nature of the P&W PDP leads the author to generalize the conclusion and stage that in general complex physical products are developed using a spiral PDP rather than a waterfall PDP. While physical products were originally developed by the waterfall framework, practitioners have learned the weaknesses of the waterfall through mistakes and other problems. As such, they have adapted the waterfall process into the spiral process without changing the label.

Spiral Development and Derivative Products.

A separate discussion seems necessary concerning one author-perceived weakness in the above discussion. For many software products, the developer may actually put the product on the market after one of the first loops around the product development cycle. Subsequent loops around the cycle add functionality and are released as upgrades or new versions. It is difficult to imagine that a virtual engine could be sold to anyone; however, if one examines the platform strategies within the hardware companies, the hardware product development cycle again fits the spiral. In this case, the first spiral is the development of the platform and each subsequent loop is the development of a derivative product.

In general, the author believes that for the development of complex physical products the difference between the spiral PDP and the waterfall PDP is only one of perspective. *In practice*, the separation of waterfall and spiral product development processes is not meaningful. However as is usually the case, having multiple perspectives on a process is useful and can be insightful as it allows one to see improvement opportunities that could not be viewed from the other perspective. Thus, discussion and work on both frameworks is worthy of future work.

CHAPTER 6: CONCLUSIONS AND FUTURE WORK

Conclusions

This work focused on the assumptions and requirements that are made during product development. The work focused on the management of a complex product from the system level and used a higher level perspective than previous work. The DSM was the main tool used throughout this work. There are two main conclusions from this work:

1) A single, system-level parameter DSM may be used to represent three stages of product development (concept design, preliminary design, and validation). The only difference in the DSMs for each stage is the assumptions (i.e., what is assumed and in what order are the assumptions made).

2) PDPs for complex physical products are well represented by the spiral framework of a PDP. The authors viewpoint is that in practice there is little or no difference in the application of these frameworks to a product development cycle.

In addition, several recommendations for Pratt & Whitney were given:

1) Include all requirements (including risk management) for a product in a single document (do not include processes)
2) Update requirements at specified occasions
3) Reorganize the system engineering organizations and insert a single chief engineer for a product.
4) Move the secondary flow organization into PSA from SD&CI
5) DSM is not as important at P&W due to the detailed system simulation capability
6) DSM would be a good training tool for new hires to assist in the understanding of the interdependencies present in a gas turbine.

The issue of perspective was discussed many times in this thesis with the conclusion that many seeming conflicts are only differences in perspective. The experience and knowledge of individuals is limited and therefore, when approaching a problem or issue, each person brings a unique perspective. Open discussion of various perspectives can prove to be insightful as one attempts to change perspectives to view a problem or issue from a different angle. Often the most insightful conclusions come from these discussions. The author hopes that the discussions of perspective included in this thesis operate along these lines.

Future Work

Many directions may be taken from this work. Several specific opportunities are discussed within the text. The items outlined here are more general paths for potential future work.

1) Extend the work on spiral and waterfall development to determine if the framework outlined here is applicable to other industries. The current work generalizes and makes conclusions based upon the author's research and experience within a single industry. While it was hypothesized that the conclusions made are applicable across industries, work across several additional industries with physical products would prove useful either to reinforce the conclusions made here or to show their limited applicability.

2) Examine the detailed ordering of the parameters output by PSM32. When assumptions are made concerning very integrative parameters (e.g., one of the module designs), the program may recommend that a parameter be completed very early after assumption of the integrative parameter. In practice, these parameters are rarely completed that early. This may be due high level portions of the integrative parameter are what is assumed, but the parameters actually depend upon some portion of integrative parameter that is not necessarily assumed. This may lead to one saying that the integrative parameters should be divided into many smaller parameters. While this may be the most rigorous and scientific method, it quickly makes a DSM unwieldy. Thus, exploration of how to better represent the relationships may be useful.

3) Benchmarking of system engineering organizations and their relationship to the management of product development. The balance of power within product development organizations is becoming more and more important as companies are pushed to ever improve the product development process. System engineers should play a large role within P&W. Is this an appropriate approach and how do other companies utilize there system engineers?

4) Although not as general, develop detailed DSMs for each module in a gas turbine. With the addition of these DSMs, a hierarchical ordering of DSMs that would be a complete representation of a gas turbine would be available.

5) Finally, utilize the system-level DSM developed in this work to generate a generic risk management process. This would extend the specific process developed by Tyson Browning [Browning, 1998] to a general case. Care would need to be taken to make the process user friendly and quick to accomplish.

REFERENCES

Anonymous, 2000. "Operating Guidelines for Integrated Program Deployment at Pratt & Whitney." Pratt & Whitney. November 2000.

Bartkowski, G., 2001. "Accounting for System Level Interactions in Knowledge Management Initiatives." Massachusetts Institute of Technology Master's Thesis. 2001.

Boehm, B.W., 1988. "A Spiral Model of Software Development and Enhancement." *Computer*, May 1988, pp. 61-72.

Browning, T.R., 1998. "Modeling and Analyzing Cost, Schedule and Performance in Complex System Product Development." Massachusetts Institute of Technology Ph.D. Thesis. 1998.

Cronemyr, P., Öhrwall-Rönnbäck, A., and Eppinger, S.D., 1999. "A Decision Support Tool for Predicting Impact of Development Process Improvements." Journal of Engineering Design. Forthcoming 2001.

Dornheim, M.A., 2000. "NASA Says MPL Was Too Cheap, Too Fast." *Aviation Week & Space Technology*, pp. 40, April 3, 2000.

Eppinger, S.D., Whitney, D.E., Smith, R.P., and Gebala, D.A., 1994. "A Model-Based Method for Organizing Tasks in Product Development." *Research in Engineering Design*, Vol. 6, No. 1, pp. 1-13.

Epstein, A.H., 2000. MIT lecture notes in 16.511 Gas Turbine Engines. Fall 2000.

Fernandez, C.I.G., 1998. "Integration of Product Architecture to Support Effective Team Co-Location." Massachusetts Institute of Technology Master's Thesis. 1998.

Glynn, S.V. and Pelland, T., 2000. "Information Flow & Knowledge Capture Lessons for Distributed Integrated Product Teams." Massachusetts Institute of Technology Master's Thesis. 2000.

Huovila, P., Leinonen, J., and Truch, P., 2000. "Intimate Relations Between Theory and Practice Linking DSM with Construction Project Management Tools." Second MIT DSM International Workshop, Cambridge, MA, September, 2000.

Kerrebrock, J.L., 1992. "*Aircraft Engines and Gas Turbines, 2nd Ed.*" MIT Press, Cambridge, MA.

Mascoli, G.J., 1999. "A Systems Engineering Approach to Aero Engine Development in a Highly Distributed Engineering and Manufacturing Environment." Massachusetts Institute of Technology Master's Thesis. 1999.

McConnell, S., 1996. "*Rapid Development-Taming Wild Software Schedules*," pp. 133-161. Microsoft Press, Redmond, WA.

McCord, K.R. and Eppinger, S.D., 1993. "Managing the Integration Problem in Concurrent Engineering." *Sloan School of Management Working Paper Number 3594.* Massachusetts Institute of Technology.

MIT DSM, 2000. http://web.mit.edu/dsm

Moy, H.M., 2000. "Commercial Gas Turbine Engine Platform Strategy and Design." Massachusetts Institute of Technology Master's Thesis. 2000.

Pimmler, T.U. and Eppinger, S.D., 1994. "Integration Analysis of Product Decompositions." *ASME Conference on Design Theory and Methodology.* Minneapolis, MN. September 1994. DE-Vol. 68, pp. 343-351.

PSM32, 2000. http://www.problematics.com

Rechtin, E. and Maier, M.W., 1997. "*The Art of Systems Architecting.*" CRC Press. New York.

Rowles, C.M. 1999. "System Integration Analysis of a Large Commercial Aircraft Engine." Massachusetts Institute of Technology Master's Thesis. 1999.

Smith, G.E. and Mindell, 2000. "Emergence of the Turbofan Engine." Pp. 107-155 in Galison, P. and Roland, A. eds. *Atmospheric Flight in the Twentieth Century.* Kluwer Academic Publishers.

Smith, R.P. and Eppinger, S.D., 1997. "Identifying Controlling Features of Engineering Design Iteration." *Management Science*, Vol. 43, No. 3, pp. 276-293. March 1997.

Sosa, M., Eppinger, S., and Rowles, C., 2000. "The Effects of Product Architecture on Technical Communication in Product Development." *Sloan School of Management Working Paper*, July 2000 Massachusetts Institute of Technology.

Stowe, M., 2000. "Identifying & Integrating Project Information for Increased Throughput." Second MIT DSM International Workshop, Cambridge, MA, September, 2000. And personal communication, 2000.

Thebeau, R.E., 2001. "Knowledge Management of System Interfaces and Interactions for Product Development Process. Massachusetts Institute of Technology Master's Thesis, 2001.

Ulrich, K.T. and Eppinger, S.D., 2000. *Product Design and Development. 2nd Ed.* McGraw Hill, Inc.

Whitney, D.E. 2000. Private communication.

Whitney, D.E., Dong, Q., Judson, J., and Mascoli, G., 1999. "Introducing Knowledge-Based Engineering into an Interconnected Product Development Process." *Proceedings of the 1999 ASME Design Engineering Technical Conferences*, Las Vegas, NV. pp. 1-10. September, 1999.

Womak, J.P. and Jones, D.T., 1996. "*Lean Thinking: Banish Waste and Create Wealth in Your Corporation.*" Simon & Schuster, New York.

Zeidner, L. and Wood, R., 2000. "The Collaborative Innovation (CI) Process." *Transactions from the 12th Symposium on Quality Function Deployment*, pp. 375-385. QFD Institute.