Improving Gas Turbine Engine Control System Component Optimization by Delaying Decisions

by

Craig T. Stambaugh Sr.

B.S. Mechanical Engineering, University of Missouri-Rolla, 1983
B.S. Engineering, Illinois College, 1983

Submitted to the System Design and Management Program
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management

at the

Massachusetts Institute of Technology

June 2003

© 2003 Craig T. Stambaugh Sr.
All rights reserved

The author hereby grants to MIT permission to reproduce and to
distribute publicly paper and electronic copies of this thesis document in whole or in part.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
JUL 10 2003
LIBRARIES

Signature of Author_____

Craig T. Stambaugh Sr.
System Design and Management Program
June 2003

Certified by_____

Daniel Whitney
Senior Research Scientist
Center for Technology, Policy & Development
Thesis Supervisor

Accepted by_____

Steven D. Eppinger
Co-Director, LFM/SDM
GM LFM Professor of Management Science and Engineering Systems

Accepted by_____

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

BARKER

[This page intentionally left blank]

Improving Gas Turbine Engine Control System Component Optimization by Delaying Decisions

by

Craig T. Stambaugh Sr.

Submitted to the System Design and Management Program
on May 03, 2003 in Partial Fulfillment of the
Requirements for the Degree of
Master of Science in
Engineering and Management

ABSTRACT

This work provides a comparative analysis of the current gas turbine engine control system development process and a proposed framework. The current process prescribes a development process in which control system component design is performed as the culmination of multiple layer system decompositions. Due to the complexity of gas turbine engines (particularly military) and associated uncertainty of several key attributes, control system design requirements must include a significant degree of conservatism to prevent any operational limitations for initial development engines.

The proposed framework investigates the risks and benefits of providing "boilerplate" test apparatus for certain control system components on initial development engines to allow the acquisition of key engine characteristics such as actuation system loads. The architecture of these test components was defined by identifying the design requirements containing significant uncertainty and providing a range of hardware options that could be combined in modular fashion to maximize flexibility. With design requirement uncertainty significantly reduced, final "flight configuration" designs could then be released with high confidence of a truly optimized design. Another important part of this framework was an approach aimed at identifying when to apply the proposed process since design requirement uncertainty varies significantly from component to component.

To reduce the engineering lead-time associated with finding an optimum control system solution once engine data is available, a linear optimization modeling approach was defined, which allowed key design features such as actuator piston size and pump performance to be traded against important component attributes such as product cost, weight, and heat generation.

Thesis Supervisor: Dr. Daniel Whitney
Senior Research Scientist
Center for Technology, Policy, and Industrial Development

ACKNOWLEDGEMENTS

I would like to thank everyone from UTC who sponsored and supported me through the course of the SDM program. Specifically, I would like to thank Zara Larsen for her sponsorship and Bob Slack for his encouragement to apply for the program. I remember having a conversation with Bob just after learning of my acceptance. I was worried about keeping up with the academics since I had not taken any college credit courses since undergrad nearly 20 years prior. Bob, being an SDM alumnus, told me that I would do fine academically and that time management would be the biggest challenge. As usual, he was right on target.

I would also like to thank the SDM staff for making the program not only bearable, but also enjoyable. Your efforts have made SDM/LFM a truly world-class enterprise.

Thanks to my SDM01 classmates for sharing their experiences and knowledge. Our class enjoyed the good fortune of having a broad mix of industry backgrounds that made for an extremely rich and rewarding learning experience. I will miss the camaraderie, but will look forward to continued connection to SDM as an alumnus.

My sincere thanks go to my thesis advisor, Dan Whitney whose probing questions and thought-provoking conversation encouraged me to think about things from different perspectives. Dan helped me understand my own business and engineering process better as we worked through this thesis. The depth and breadth of his knowledge in many different areas of the automotive and aerospace industries helped me to explore areas that otherwise would have gone unnoticed.

Most of all, I thank my wife, Mary for her understanding and patience throughout the program. I'll never begin to understand how you managed to keep up with the house, our three teenagers, your job, and pursuit of your own masters degree. You are truly a special person. Finally, thanks to my children Jessica, Angela, and CJ for enduring the sacrifices our family had to make over the past two years. With your mom and I both graduating this spring, we can now get back to the business of being a family. It comes none too soon.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS 4

LIST OF FIGURES 7

LIST OF TABLES 8

1 INTRODUCTION 9

1.1 Problem Statement and Motivation 9
1.2 The Heart of the Problem 10
1.2.1 Gas Turbine Engine Development Program 10
1.2.2 Design Dependencies – DSM Approach 15
1.3 Context within General System Design Process 17
1.3.1 Need Assessment 24
1.3.2 Concept Generation & Evaluation 26
1.3.3 Requirements Definition 30
1.3.4 Design 33
1.3.5 Develop 34
1.3.6 Test 34
1.3.7 Implement 35
1.4 Thesis Overview 35

2 BACKGROUND 37

2.1 General Gas Turbine Engine Control System Architecture 37
2.2 Functional Decomposition 40
2.2.1 Engine Control Needs 40
2.2.2 Control System Concept 44
2.2.3 Identification of Functions and Form 50
2.3 Chapter Summary 59

3 RELATED WORK 60

3.1 Understanding the System Design Process 60
3.2 Set-Based Design 61
3.3 The Benefits of Experimentation 61
3.4 Chapter Summary 62

4 ANALYSIS OF CURRENT DEVELOPMENT PROCESS 64

4.1 Integrated Product Development Process 64
4.2 Requirement Flowdown 67
4.2.1 Control System Design 67
4.2.2 Control Component Design 72
4.2.3 Control Component Development Test 74
4.3 Quantification of Risk 77
4.4 Reasons for Non-Optimum Designs 80

4.5 Chapter Summary 82
5 FRAMEWORK FOR IMPROVED DEVELOPMENT PROCESS 83
5.1 Determining Applicability of Framework to Components 83
5.1.1 Component Risk Assessment Matrix 84
5.1.2 Single Requirement Affecting Multiple Components 89
5.1.3 Determination of Requirements to be Assessed 91
5.2 Generic Test Apparatus 93
5.2.1 Architectural Considerations 93
5.2.2 Interface Considerations 98
5.2.3 Functional Considerations 100
5.2.4 Technology Maturity Considerations 101
5.2.5 Implications for Product Validation 102
5.2.6 Business Case of Implementation 106
5.3 Optimization Modeling 108
5.4 Risks and Benefits Compared to Current Process 115
5.5 Chapter Summary 116
6 ORGANIZATIONAL ANALYSIS 119
6.1 Current Organization 119
6.2 Organizational Improvements 122
6.3 Chapter Summary 124
7 CONCLUSIONS AND FUTURE WORK 126
7.1 Viability of Revised Development Framework 126
7.2 Future Work 126
GLOSSARY 128
BIBLIOGRAPHY 129

LIST OF FIGURES

Figure 1-1. Typical Gas Turbine Engine Development Program 14
Figure 1-2. Simplified Turbofan Engine Design Structure Matrix 16
Figure 1-3. Generic Product Development Process 20
Figure 1-4. Modified PDP with Proposed Framework 21
Figure 1-5. Generic PDP System Vee Model 22
Figure 1-6. Modified General PDP Gantt Chart 23
Figure 1-7. Concept Development Process [Ulrich & Eppinger, 2000] 29
Figure 2-1. Pratt & Whitney F100-PW-229 Military Engine Cross Section 37
Figure 2-2. Gas Turbine Engine Control System Functional Decomposition Block Diagram 41
Figure 2-3. Examples of Minor Loop Control Concepts 47
Figure 2-4. Example of an Actuation Subsystem Functional Schematic 52
Figure 2-5. Master-Master Actuation Architecture 54
Figure 2-6. Slave-Slave Actuation Architecture 55
Figure 2-7. Example of a System Schematic 56
Figure 2-8. Typical Control System Schematic for an Afterburning Turbofan 58
Figure 2-9. Fuel Control System with Electronic Engine Control 59
Figure 4-1. P&W IPD Activity Linkage 65
Figure 4-2. Hamilton Sundstrand Passport Review Process 66
Figure 4-3. Typical Control System Development Program 68
Figure 5-1. Component Risk Assessment Matrix 86
Figure 5-2. Potential GTA Actuator Architecture 95
Figure 5-3. GTA Concept Development Schedule 105
Figure 5-4. Example of Actuation Capability Design Space 110
Figure 5-5. Example of a Fuel System Optimization Model 114
Figure 5-6. Example of Engine Simulation Deck Output 115
Figure 6-1. Current PW Organizational Structure 121

LIST OF TABLES

Table 2-1. Control System Functions with Examples of Subfunctions 43

Table 5-1. Results of Business Case Analysis for GTA Implementation 118

1 INTRODUCTION

1.1 Problem Statement and Motivation

Recent experience on a jet engine development program has identified the opportunity for significant improvement in terms of cost and schedule performance during the early stages of the program. For obvious reasons, weight is a critical attribute of aircraft systems, particularly high performance military fighters. Today's business environment also demands affordability, which often conflicts with attaining low weight. Development of complex gas turbine engine control systems typically requires multiple design iterations for weight/product cost optimization and to correct technical shortfalls, resulting in significant development program cost growth and schedule risk, therefore reducing customer satisfaction.

Toyota has been very successful in achieving consistently high levels of engineering quality at surprisingly low cost through the use of concurrent set-based design in which design decisions are delayed, pushing delivery of hard specifications to their suppliers until late in the process[1]. This thesis will attempt to show the benefits of delaying design launch for turbine engine control system components whose design can be significantly affected by the uncertainty of requirement and interface definition. Put simply, "Design the engine, and then design the control system". Alternate non-flight components must be provided for initial engine test to allow development of other engine components such as software and turbomachinery and to gather test data to reduce the uncertainty of control system requirements. Causes for early design iteration are theorized and research from past development programs is presented to support these theories. Analysis is presented to support the alternate development approach.

---

[1] Ward, Allen, Jeffrey K. Liker, John J. Cristiano, and Durward K. Sobek II. "The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster." Sloan Management Review, Vol. 36, Issue 3 (Spring 1995), 43.

1.2 The Heart of the Problem

1.2.1 Gas Turbine Engine Development Program

Figure 1-1 shows two potential gas turbine engine development processes. Rather than indulge in excessive detail describing the engine development process, this section will be limited to a discussion on the basic schedule conflict at the heart of the issue about which this work is concerned.

The top of the figure represents a development schedule driven by requirement flowdown in which design tasks begin when adequate requirements have been defined by task predecessors. The development program begins with the engine preliminary or conceptual design phase in which high level propulsion system concepts including the basic engine cycle design are defined. Major engine module designs (compressor, fan, turbine, combustor, etc) are launched as the functional requirements are defined. Since many of the control system functional requirements result from the design implementation of the other engine modules, the control system design would not begin until the turbomachinery and exhaust system designs were reasonably well defined. In an ideal world, the design of the control system would not begin until the design of the turbomachinery and exhaust system are finalized since many key control system requirements such as actuation loads and slew rates and sensor placement are defined by the outcome of those tasks. This issue of design dependencies is discussed in more detail in section 1.2.2.

The desired control system development schedule, shown as the fourth task on the upper chart in the figure, includes Design, Manufacture, and Closed-Loop Bench (CLB) subtasks that all lead up to FETT. The CLB is a subsystem-level functional integration test that verifies proper operation of the control system hardware and software and is an activity that is somewhat

unique among the major engine modules. Its primary purpose is to develop the functionality of the control system in advance of the rest of the engine in order to minimize the risk to the turbomachinery and enable the immediate acquisition of aeromechanical and performance data at FETT. In order to perform this testing and obtain an accurate representation of the true control system performance to be expected at FETT, hardware with the same functional characteristics as the "production" configuration is needed. Accomplishing the overall "desired" control system development program from the upper chart in Figure 1-1, however, would result in the control system hardware and software being several months late to FETT, thus creating a significant program risk.

In order to meet program schedule constraints, a "right-to-left" plan culminating in delivery of functionally verified hardware and software to FETT is actually used and is represented by the lower chart on the figure. At an appropriate point during or near the end of the Engine Preliminary Design phase, the design of most, if not all, major engine modules is launched, including the control system. This is done with the basic (quite optimistic) assumption that all hardware delivered to the First Engine to Test (FETT) will be of a production configuration. Note that the CLB, manufacture, and design subtasks drive the start of the "actual" control system development task to be nearly aligned with the start of the turbomachinery and exhaust system design tasks. At this point in the program, the control system requirement uncertainty is at a high level as indicated by the dotted curve on the figure. This uncertainty is reduced from a High to a Moderate level with the completion of the turbomachinery and exhaust system design tasks since most of the key control system requirements are defined by the design of the rest of the engine modules. The uncertainty

continues a gradual decline through the hardware manufacturing phase through the development and refinement of analysis and simulations.

The next significant reduction in uncertainty occurs after FETT with the acquisition of key engine data. Unfortunately, all decisions affecting control system component definition had to be made during the control system design phase when requirement uncertainty was at a moderate to high level. To reduce technical program risk, conservative assumptions are made regarding control system component requirements resulting in sometimes significant product cost and weight impacts.

This basic schedule conflict can be summarized by the two shaded boxes on the figure. The one shown on the upper chart represents the point of requirements availability for "normal" system decomposition (i.e. the control system is the last to receive requirements). The shaded box on the lower chart shows that, for a compressed development schedule, the control system is the first engine module to require hardware.

Impacts of non-optimum product cost and weight result in one of two courses of action.

1. Components are eventually redesigned to optimize critical attributes. In the case of actuators, pistons are downsized reducing weight. In the case of pumps, flow capacity and/or pressure levels are reduced resulting in weight and/or heat load savings. Due to the high non-recurring cost (engineering and flight certification) of initial development, the cost of redesign is usually quite substantial, ranging in the low $100k's for relatively simple parts such as temperature sensors to the millions for complex components such as electronic controls.

2. Components continue into production carrying the burden of unnecessarily high product cost and weight. The impacts of excessive cost and weight are shouldered by the operator in terms of life cycle cost (LCC). LCC or total cost of ownership is comprised of development, production (acquisition), operations, and support costs. Unnecessarily

high product cost would induce a minor impact on the development program due to the normally small amount of hardware associated with development. However, due to the potentially large number of units associated with a significant acquisition program, the impact of one dollar of product cost could exceed $400 in LCC[2]. Due to the huge impact on fuel consumption, however, the trade factor for weight tends to be much higher. One pound of propulsion system weight could equate to over $10 million in LCC over the life of the program!

[2] Trade factor defined for a modern military development program. Exact source not disclosed due to sensitivity.

TYPICAL GAS TURBINE ENGINE DEVELOPMENT PROGRAM

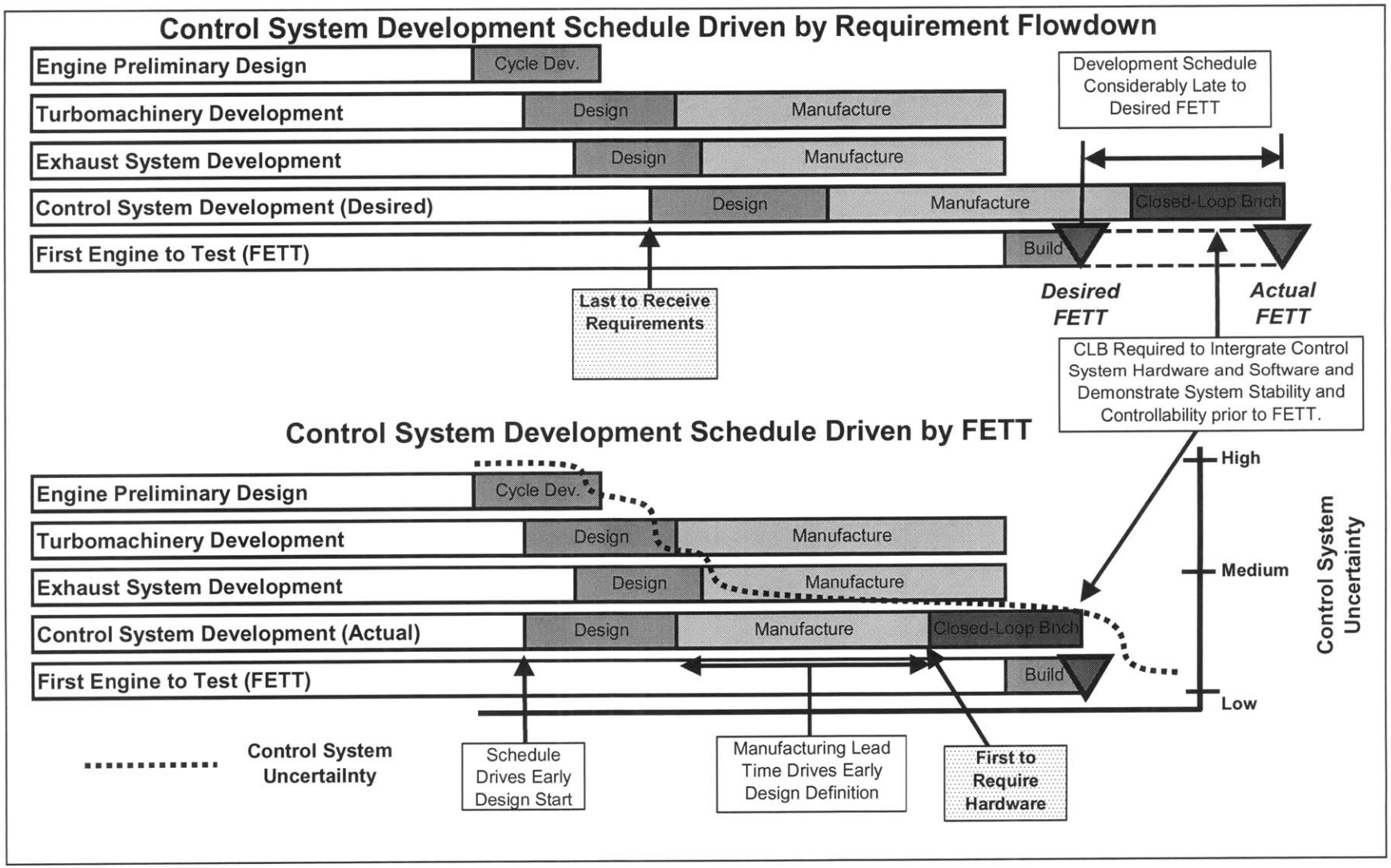


Figure 1-1. Typical Gas Turbine Engine Development Program

1.2.2 Design Dependencies – DSM Approach

In general, a gas turbine engine is designed in much the same way that it is built – from the inside-out. As Hague discussed in his work on parameter-based design of turbofan gas turbine engines, the order of design of the major engine modules begins with the high pressure compressor (HPC) and proceeds "outward" with the high pressure turbine (HPT), low spool (fan and low turbine), diffuser/combustor, mechanical components (shafts and bearings), and finally the controls and externals[3]. The connectivity of these major modules is described in more detail in section 2.1. Hague's work on mapping the turbofan engine development process using the Design Structure Matrix (DSM) shows graphically the interdependence of the various design and development tasks. Of interest are the initial turbomachinery design tasks on which the control system is dependent for requirements. Figure 1-2 represents a greatly simplified DSM for a typical commercial turbofan engine[4]. The rows represent tasks required in the product development process, in this case high-level design tasks for the major engine modules. The tasks are duplicated for each column across the top of the matrix. As indicated in the annotations, an "X" in a particular row means that in order for the task in that row to be performed, information is required from the task in column containing that "X". Take, for example, the row for HPT (High Pressure Turbine) Design that contains an "X" in the columns labeled "Diffuser/Combustor Design", "LPT Design", and "Control System Design". This indicates that, in order to complete the HPT Design task, information is required from the Diffuser/Combustor Design, LPT (Low Pressure Turbine) Design, and Control System Design

---

[3] Hague, Douglas C. "Description of a Turbofan Engine Product Development Process." SM Thesis, Massachusetts Institute of Technology, 2000, 52.

[4] Adapted from Hague, Douglas C. "Description of a Turbofan Engine Product Development Process." SM Thesis, Massachusetts Institute of Technology, 2000, 49.

tasks. In any DSM, the X's that exist above the diagonal represent tasks that must begin without all of the required information.

It is significant to note here that the Control System Design task requires input from every other major engine module design task. Hence, the entire engine must be nearly designed in order to provide input to the control system design process, i.e. to define the control system requirements. Historically, the approach taken in a schedule-driven program (nearly all programs are schedule-driven) is to make assumptions for requirements based on the best available data. In some cases, requirements are based on crude models where in other cases, other similar product design aspects are used.

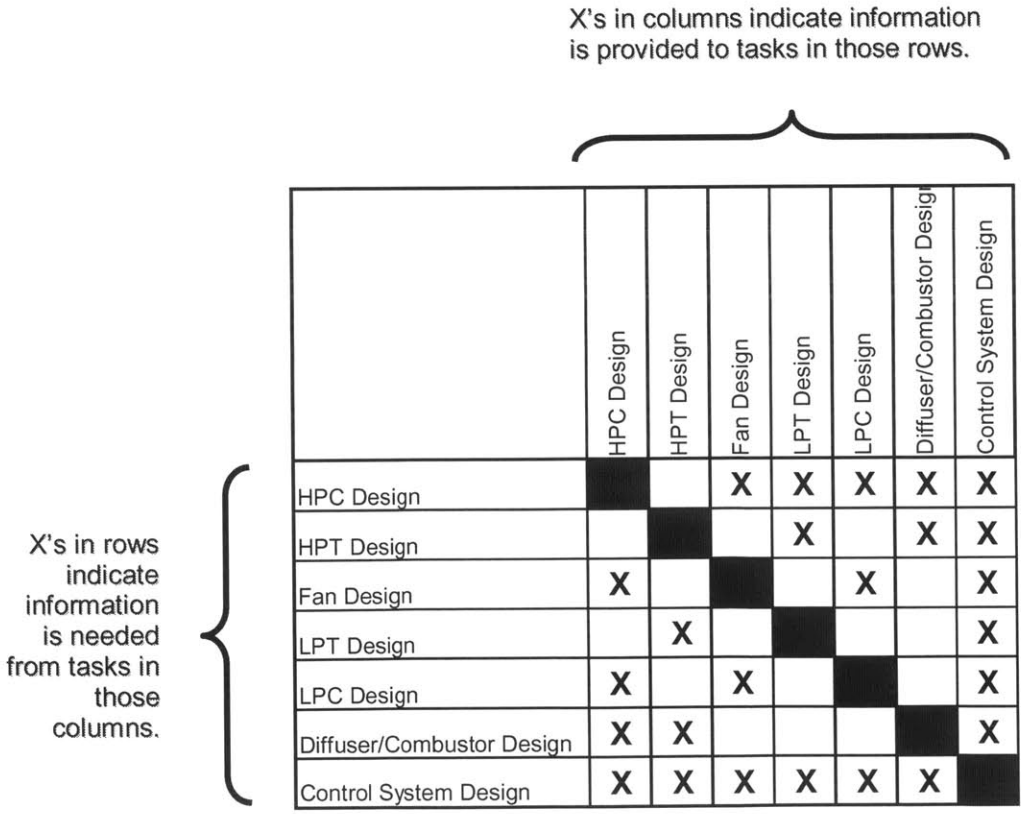


Figure 1-2. Simplified Turbofan Engine Design Structure Matrix

1.3 Context within General System Design Process

Figure 1-3 shows the framework of a generic Product Development Process (PDP)[5]. This process, being presented at a very high level of abstraction, does not attempt to represent the magnitude of relative effort or iteration that occurs within each major step. These will vary with the particular project being undertaken. Suffice it to say, however, that in the gas turbine engine business, significant iteration usually occurs during the Requirement Definition, Design, and Development phases due to system complexity and resulting uncertainty in the performance and integration aspects of the modules, subsystems and components.

Figure 1-4 represents a revised PDP showing the additional steps proposed in this work. During the Requirement Definition phase, the risk level for each control system component is assessed. The methodology for assessing component risk is discussed further in section 5.1. Components having a sufficiently high risk level (which must be determined locally based on the program manager's threshold) will, if possible, delay detailed definition of the design feature in question, or potentially the entire component if definition of this feature is critical to defining the component's architectural layout. Generic Test Apparatus (GTA) hardware must be procured to provide subsystem and engine test with the functionality of the component being delayed. This is described further in section 5.2. Data from initial subsystem and/or engine test are then acquired and fed back into the development process to refine requirements and reduce uncertainty, hopefully resulting in a component detailed design that is very close to optimum.

---

[5] Crawley, E. F., Adapted from System Architecture Class Notes, 1/23/01.

Figure 1-5 represents the "System Vee" model for the General Product Development Process[6]. The left side of the model shows the design phase starting from the upper left with Identification of Product Need, proceeding down the Vee with Product Planning, Concept Development, System Design, and Detailed Design. This process conforms to the classical system design methodology of decomposition in which the product design emerges through progressive steps of definition and refinement from the system level down to the piece-part level. Of note is the iterative Simulation process that exists between the Preliminary Design and Detailed Design steps. Particularly for complex systems with highly interactive elements, simulation is often a necessary activity to efficiently explore the design space and make trade-offs between competing requirements and attributes. Thomke describes the importance of simulation with the example of BMW[7]. Only a few years ago, experimenting with novel ideas to improve vehicle crash survivability was an expensive and time consuming proposition since this required the construction of physical prototypes. The feedback loop stretched over months of time creating a barrier to innovation for designers. Data was acquired too late to influence early decision-making in the product development cycle, which sometimes resulted in expensive changes just prior to full production. Today, crash tests are performed in a virtual computer environment allowing the design team to rapidly evaluate many different configurations before key design decisions need to be made. The company is able to better understand not only what works, but why.

The step at the bottom of the system Vee represents the creation of a prototype system comprised of initial configuration hardware meeting the design intent. This provides an

---

[6] Crawley, E. F., System Architecture Class Notes, 1/23/01.

[7] Thomke, Stephan, Enlightened Experimentation – The New Imperative for Innovation, Harvard Business Review, 2001, 69.

opportunity to develop assembly processes and perform initial physical integration activities. Once this initial prototype system has been assembled, the process begins traversing up the right side of the system Vee during which system validation occurs. Although not shown in this model, many companies such as Ford and Pratt & Whitney execute the validation process in a manner that is a mirror image to the system design process. Validation begins at the "bottom" (i.e. the detail piece-part level) and progresses to higher levels of aggregation in build-up fashion. This model contains Refinement activities that encompass the bottom three steps of the process (Detailed Design, Prototype Build, System Validation). As design shortfalls or optimization opportunities are encountered during the validation process, design iterations and subsequent retesting occur until the gap is closed between the system's requirements and its demonstrated capabilities. This iteration and the issues surrounding it is the subject of this work.

Figure 1-6 represents the activities shown in Figure 1-4, but in Gantt chart format. This format is useful to illustrate the time phasing of the various tasks that will more clearly show the underlying issue being addressed by this work.

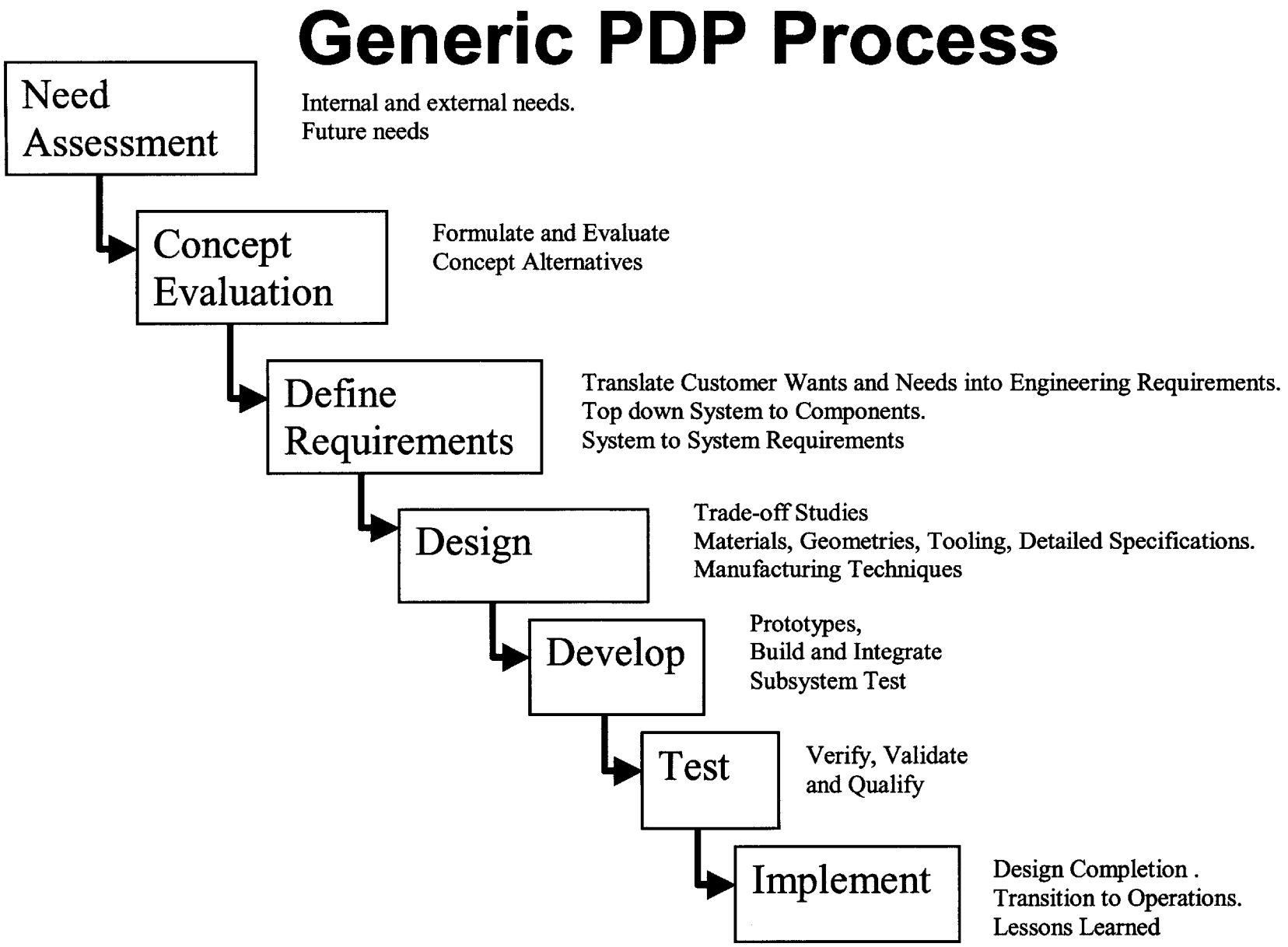


Figure 1-3. Generic Product Development Process

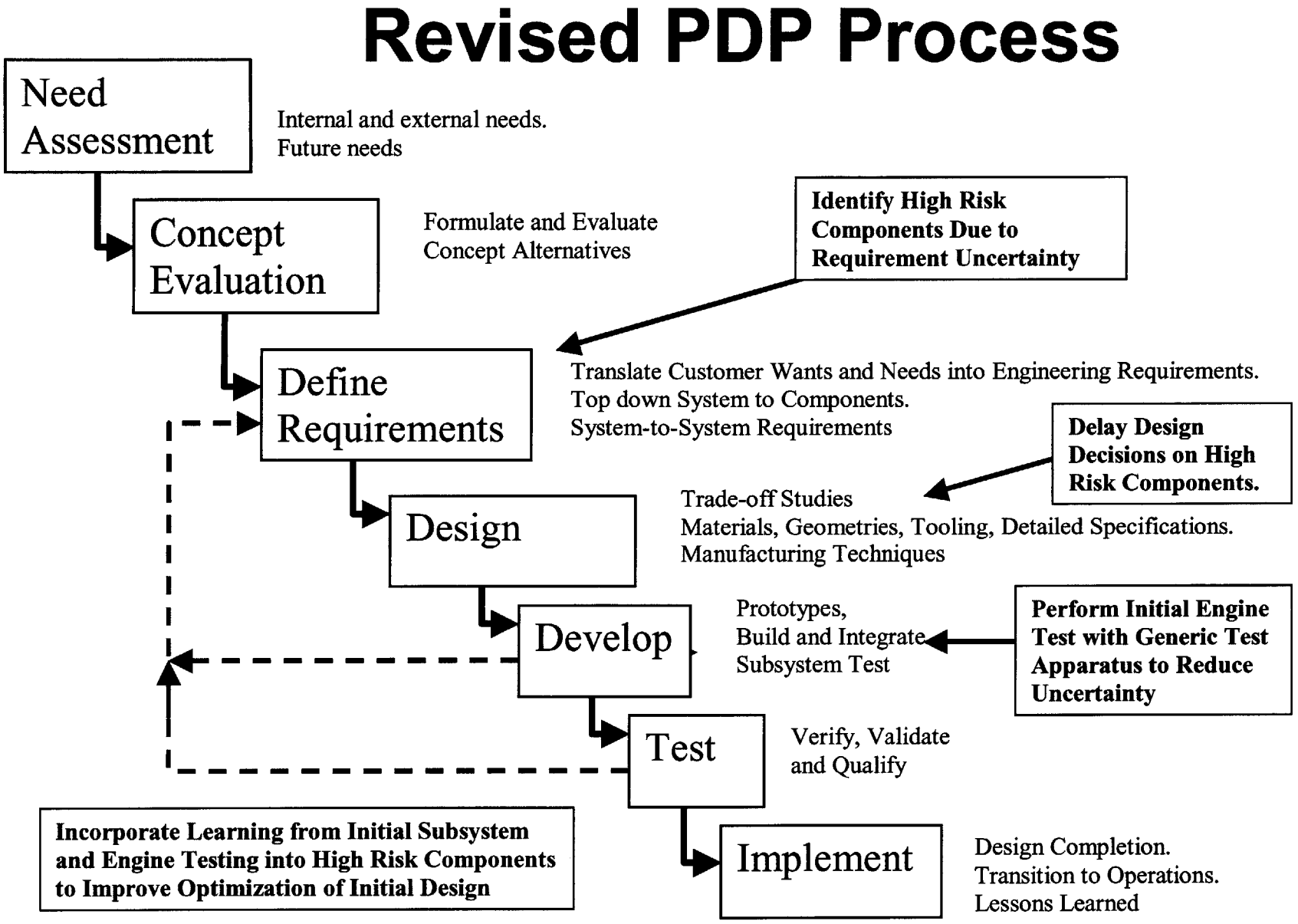


Figure 1-4. Modified PDP with Proposed Framework

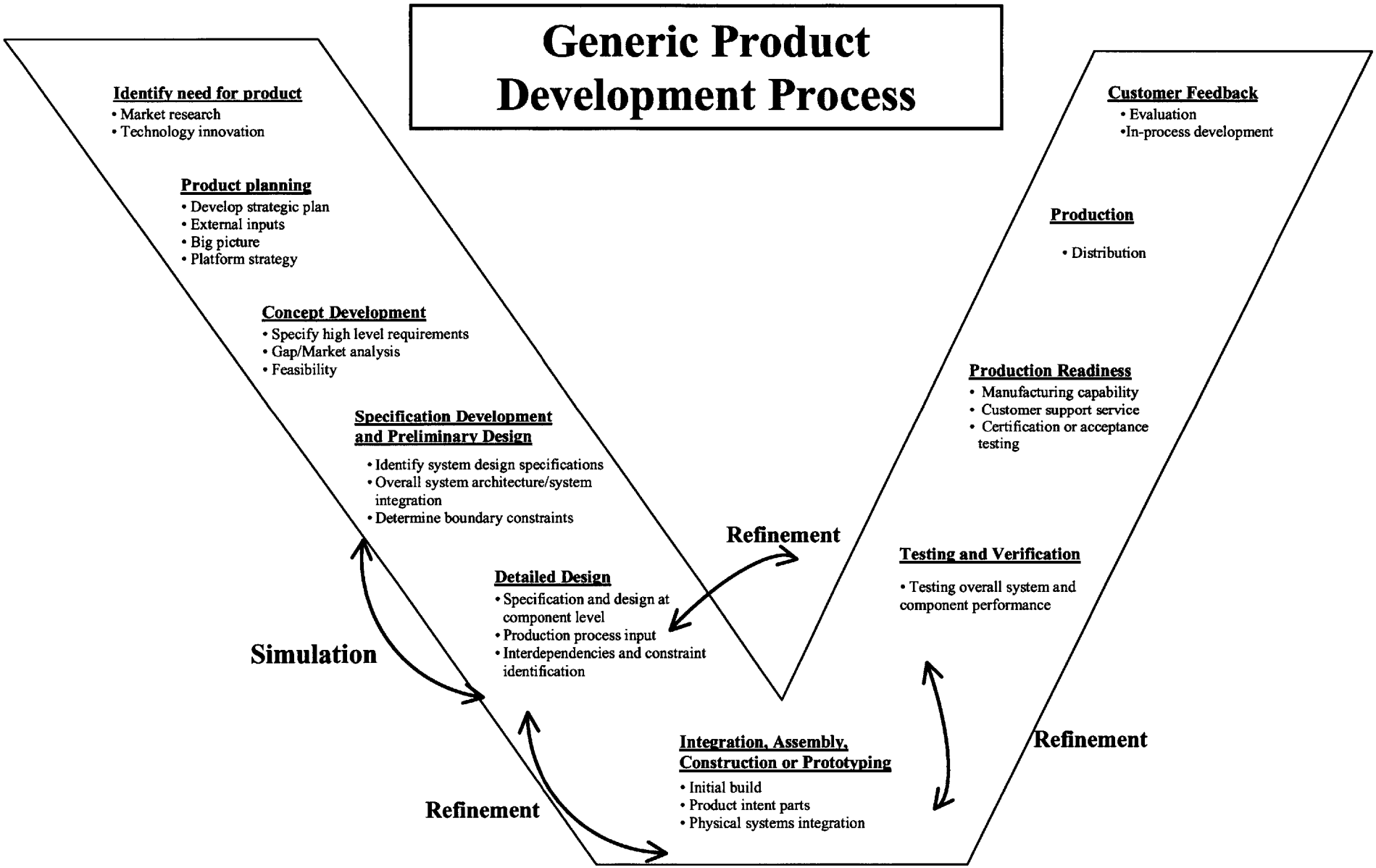


Figure 1-5. Generic PDP System Vee Model

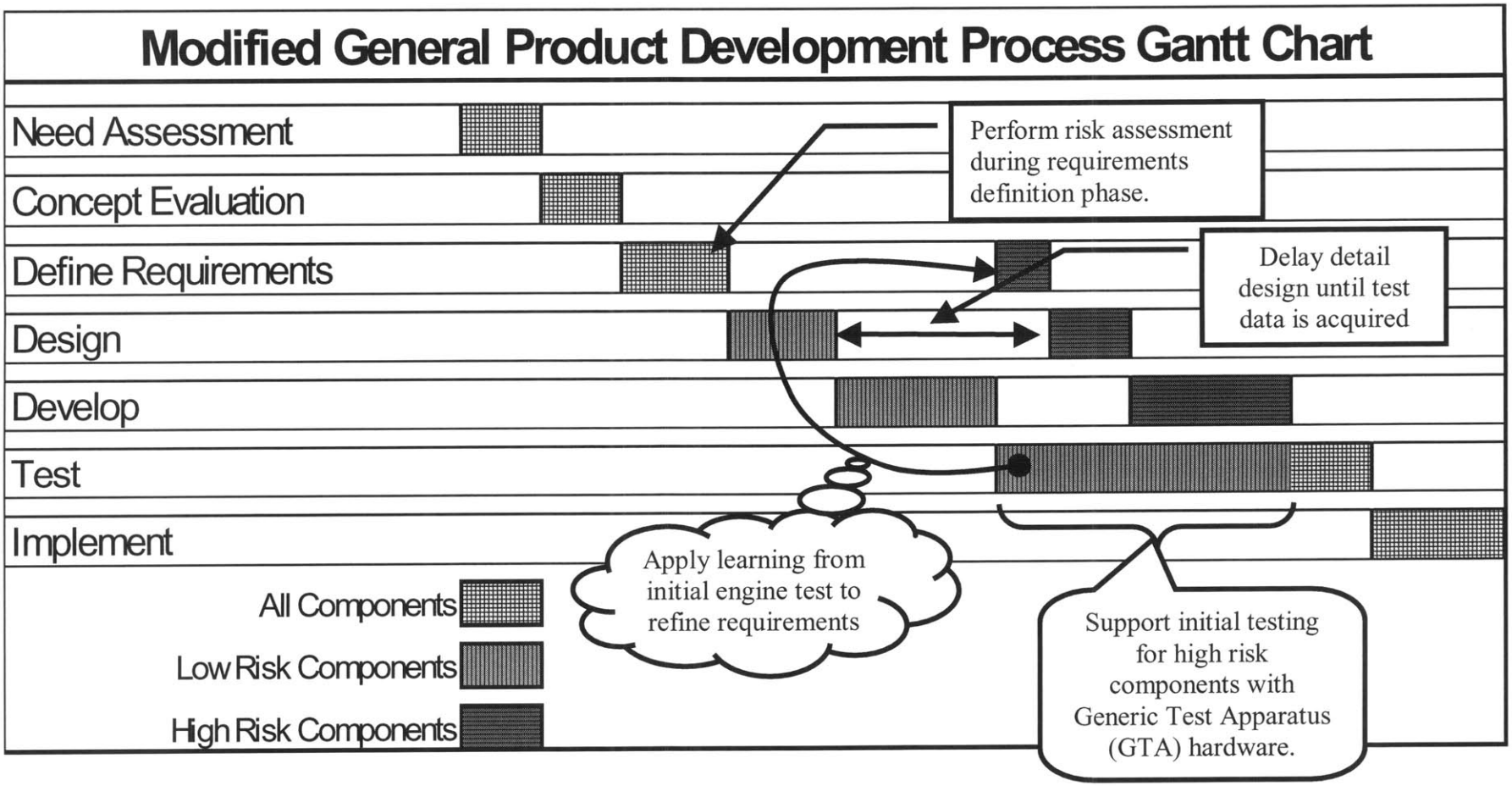


Figure 1-6. Modified General PDP Gantt Chart

It can be seen graphically from Figure 1-6 that requirement definition and design for high-risk components can be delayed until the test phase begins during which key data is acquired to resolve uncertainty. This key data is then fed back into the requirements definition process to allow the development of high-risk components to proceed. To allow engine test to proceed unencumbered, functionally equivalent hardware (GTA) must be provided during the initial test phase. The production configuration of the high-risk components is inserted into the development process late into the test phase resulting in less engine development time than their low-risk counterparts. This induces an additional potential technical risk of reduced product maturity at the beginning of the implementation phase. This issue deserves careful consideration in the risk assessment process (part of the requirements definition phase of the revised PDP) and will be discussed further in chapter 5.

The preceding figures represent a general Product Development Process framework that can be applied widely to many different types of products. Since the gas turbine engine business involves some rather specialized activities that are somewhat unique when compared to the automotive or other commercial industries, a more detailed description of the PDPs as they apply to the gas turbine engine control system business is in order.

1.3.1 Need Assessment

At the highest level of abstraction, Needs Assessment usually represents an initial Request for Proposal (RFP) from the procuring activity (government or airline) for an aircraft propulsion system. The document contains performance requirements at the propulsion system level such as thrust, thrust specific fuel consumption (TSFC), thrust transient time limits, and airstart envelope. Targets for other attributes such product cost, weight, maintenance costs and turn time, support costs, reliability, and safety are also provided.

Occasionally, engine manufacturers will recognize a customer need that is not satisfied by their existing product line. If the business opportunity holds adequate promise for future revenue streams, a formal business case will be developed. As with most business case studies of complex products with relatively long life cycles (at least 30 years for most jet engine models), the more effort that is applied to the study, the more accurate the prediction of the potential revenue stream. Since these studies usually are internally funded, managers must make difficult decisions regarding the amount of scarce engineering resources to apply to the studies. If the business case meets internally defined criteria for investment, revenue, and risk, the airline or airframer will be approached with an unsolicited proposal to re-engine an existing aircraft with the new product. The investment required to develop a new centerline powerplant with even modest technology incorporation can be quite substantial (in the hundreds of millions of dollars), therefore, some form of customer commitment is usually negotiated prior to launching the engine development program.

Since the control system itself does not produce thrust, but *facilitates* the thrust-producing process of the turbomachinery by providing the proper inputs such as fuel metering and variable geometry positioning, there are only a few customer needs that are directly satisfied by the control system. Probably the most significant needs addressed directly by the control system are communicating operator commands and airframe data to the propulsion system and monitoring and communicating propulsion system health to the operators and maintainers (i.e. pilots and ground maintenance crew). Performing the latter task with a high degree of accuracy and comprehensiveness can substantially reduce the customer's cost of ownership of the propulsion system through reduced unnecessary maintenance (i.e. fewer "false calls"), more efficient resolution of on-condition maintenance (accurate troubleshooting), and the accurate

prognostication of future maintenance needs *before* they become safety issues. The latter need is only recently becoming feasible with the development of on-board electronics, sensor suites, and software algorithms that can sense and record this data. With this capability, operators can better manage maintenance operations to minimize or prevent disruption of flight operations due to unplanned maintenance activities.

The remainder and vast majority of needs satisfied by the control system are one or more levels of decomposition deep and as such are *derived.* The basic architectural concept of the engine must be defined in order to completely define the needs for engine control. Approximately half of the high-level needs for engine control and diagnostics are "generic" in nature (i.e. they are present on nearly every gas turbine engine, regardless of engine architecture) and half are specific to the thrust-producing and airframe installation concepts chosen for the propulsion system. This is discussed in more detail in section 2.2.1.

1.3.2 Concept Generation & Evaluation

The current state of technology, corporate strategy, regulations, and customer expectations also play a significant role in concept selection. Propulsion system concepts are selected to meet customer needs based on many criteria, but the significant ones are:

- Engine/Aircraft Performance
- Current State of Technology Maturity
- Historical Success of Similar Products

At this step, these criteria are usually applied not only at the highest level of abstraction (i.e. the propulsion system) to define the basic engine parameters such as airflow rating and engine cycle, but also at the first level of decomposition (i.e. at the major engine module level). For example, in the case of the engine control system, an electrical actuation system might be

selected over a hydraulic system if the technical trades (i.e. product cost, weight, reliability, etc.) show a benefit and the current state of technology maturity indicates an acceptable level of risk. Due to the safety criticality of aerospace propulsion systems, customers are typically risk-averse (some more than others). Accordingly, the state of component/system technology maturity and the historical success of similar product architectures play a significant role in the control system concepts selected for a particular application. Elements of complexity and uncertainty are again involved in a customer's risk-aversion due to the complex processes and interactions associated with the gas turbine engine operating environment (complicated flow regimes, difficult-to-predict vibration signatures, and wide ranges of thermal extremes). These levels of complexity and uncertainty coupled with the long design/manufacture cycle time for most aerospace components (normally 12-18 months) will normally result in selection of somewhat conservative concepts that are backed up by hard data or field experience.

Corporate strategy can influence concept selection, particularly if use of the concept might result in increased marketability, product differentiation from the competition, or a streamlining of the company's product line to reduce costs through economies of scale. In today's challenging economic environment, technical innovation, even for high-tech aerospace systems, is taking a back seat to product cost as the key product differentiator. This has driven companies to increasingly utilize common platform architectures to improve cost performance.

In some instances, customer expectations can also drive concept selection, primarily in two ways:

1. Customers using legacy products have usually made a significant investment in support equipment and training of their support personnel. These customers will invariably attempt to influence concept selections to minimize the impact in these areas unless significant benefits are projected. For example, adopting a digital communication

protocol between an electronic control and ground support equipment might require a significant investment in new equipment, but could offer the benefit of reduced maintenance time and reduced cost of product ownership through the use of improved diagnostic and engine life usage data.

2. Returning to the previous discussion on risk aversion, certain conservative customers rely heavily on their field experience to drive concept selections, rather than a bottom-up need-based approach. In most of these situations, certain architectures might be *ruled out* based on customer experience, even though the situation may not be directly comparable. Particularly in military applications, it is becoming extremely rare for a customer to specifically request a particular concept, mainly due to the Federal Acquisition Reform Act of 1995, which prescribed the use of performance-based specifications (what the system must do) rather than specifying the widespread use of hardware conforming to Military Standards or Military Specifications (how the system functions)[8].

Crawley informally defines architectural concept as a product or system vision, idea, notion, or mental image that maps form to function and embodies "Working Principles".[9] Let us consider the example of the function of variable geometry actuation. The term "variable geometry" refers generically to any mechanism within the engine that dynamically alters the direction of gas flow in order to enhance the performance, operability, or functionality of the engine (e.g. variable compressor stator vanes or variable exhaust nozzle flaps). The working principles that could be employed to position these mechanisms include:

- A fluidic actuator that develops force by the application of a differential fluid pressure across a power piston.

---

[8] Federal Acquisition Regulation (FAR) Subpart 37.6 (http://www.acqnet.gov/far/current/pdf/FAR.book.pdf)
[9] Crawley, E. F., System Architecture Class Notes, 1/23/01.

- An electro-mechanical actuator that develops force by rotating a ball-screw mechanism with an electric motor.

Both concepts map their forms (which are substantially different) to the function of developing a force to position a bell crank or similar kinematic linkage which, in turn, positions the target mechanism such as variable compressor stator vanes or variable exhaust nozzle flaps.

Ulrich & Eppinger define the additional step of Defining Target Specifications between Need Assessment and Concept Evaluation. This step is highlighted in Figure 1-7[10].

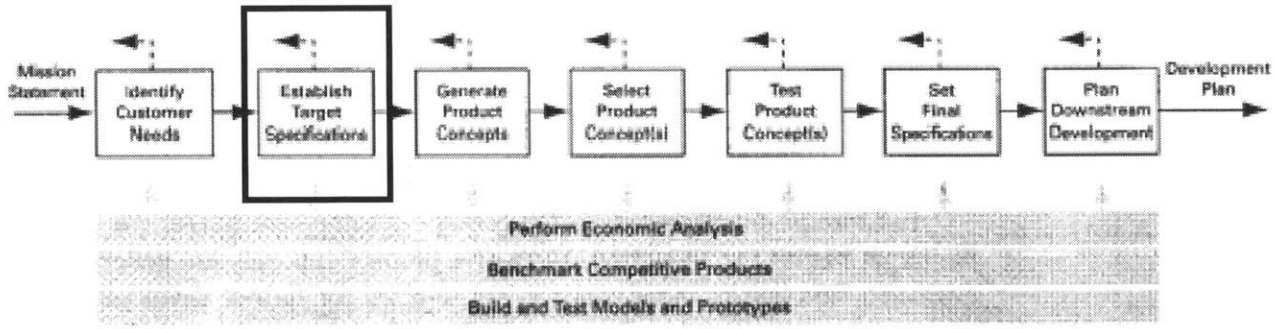


Figure 1-7. Concept Development Process [Ulrich & Eppinger, 2000]

Quantified target specs for key control system attributes at this point are very difficult to define since most of the key spec attributes (e.g. product cost, weight, reliability, safety, vulnerability, etc.) are allocations derived from the propulsion system requirement. Pratt & Whitney has used several methods in the past to define these allocations, but they are primarily based on similar products from the company's product line, usually with some incremental "challenge" for improvement. Some of these allocated requirements, such as product cost and weight, are heavily influenced by the concepts selected since they are closely coupled to the

---

[10] Ulrich, Karl T. and Steven D. Eppinger, Product Design and Development, McGraw-Hill, Inc., 2000, pg 18.

control system hardware implementation. Other attributes that are not closely coupled to hardware implementation such as reliability, safety, and vulnerability are defined mainly as goals to be superior to previously fielded systems. For example, reliability and safety goals for the Air Force F-22 fighter propulsion control system are set higher than the system they intend to replace, namely the Air Force F-15 fighter.

1.3.3 Requirements Definition

1.3.3.1 Identification of Control System Component Functionality

Based on the propulsion system and control system concepts selected in the earlier step, an architecture is identified which will satisfy the functional requirements. For the control system, a functional schematic is developed at this point, which serves to allocate functionality to components. Continuing with the previous example of the actuation system, actuation effectors, power source, and connectivity are schematically defined. For an electromechanical actuation system, this might include a generator, power distribution and control harnesses, electronic control, and various actuators. For a hydraulic actuation system, the generator is replaced by a pump and power distribution harnesses are replaced by plumbing. It is at this point in the process that the majority of product cost and weight are locked in. For this reason, the previous step of Concept Selection is extremely critical in meeting low cost and weight objectives.

1.3.3.2 System Design

This thesis is focused on this step of the general design process. It is at this point that the engine design must be known well enough to define control system functional requirements.

Ulrich and Eppinger refer to this step as "Set Final Specifications" [11], drawing a distinction from Establishing Target Specifications, as discussed earlier. This refinement represents a revisiting of the target specifications set prior to concept selection, taking into account the technological constraints that are now better understood. The design team must make some difficult trade-offs between performance, cost, weight, safety and many other requirements. It is extremely rare that aerospace components defined at this stage meet all target specs initially proposed for them since, in actuality, the real goal in achieving customer satisfaction is to meet all technical requirements while *minimizing* product cost and weight, not merely achieving a target value. It has been said that if the initial design of a component meets all target specs, the component was not adequately challenged.

Examples of some of the high-level functional attributes for control system components are listed below:

- Actuator size (load and stroke)
- Pump type, pressure level, and flow capacity
- Fuel metering accuracy and dynamic response
- Anti-ice/De-Ice system airflow and temperature (pneumatic concept)
- Heat exchanger sizing
- Ignition system energy and spark rate
- Sensor accuracy and response
- Electrical system sizing (power generation)
- Electronic Control Processor Throughput

---

[11] Ulrich, Karl T. and Steven D. Eppinger, Product Design and Development, McGraw-Hill, Inc., 2000, pg 94.

Unfortunately, during the initial engine design phase, significant uncertainty in one or more of the above requirement areas usually exists. There are two prevalent reasons for this uncertainty:

1. High System Complexity – Modern propulsion systems are becoming increasingly integrated with airframe systems. This tends to increase the complexity of an already incredibly complex machine. Physical processes inherent in gas turbine engines such as compressible flow through a complex flowpath and combustion are still difficult to accurately model, although significant progress has been made over recent years. Increased airframe integration also increases the number and complexity of propulsion system interfaces. In the case where a new aircraft and engine are both developed, the aircraft development program normally lags behind the propulsion system development by 1-2 years. This is primarily due to the lengthy ground test development and qualification program required by the engine manufactures and customer/regulators such as the US Air Force or Federal Aviation Administration (FAA). This lag results in significant uncertainty in propulsion system interface definition early in the engine design cycle (engine detail design typically runs concurrent with aircraft preliminary design).

2. Desire for Increased System Capabilities – increased functional performance and safety levels while reducing weight and cost result in more iteration at higher levels of abstraction (i.e. at the major engine module level). It is the major engine module designs along with the design of the top-level engine control laws that set the requirements for control system hardware. As the major engine modules (e.g. fan, compressor, and exhaust nozzle) and control modes (e.g. closing loop on fan speed, calculated airflow, or engine pressure ratio) continue to iterate, so do the significant design requirements for the control system components. Hague's work in describing a gas turbine engine development process by using the Design Structure Matrix (DSM) clarifies this statement by showing the dependencies and

interactions between the various design tasks[12]. This was discussed in greater detail in section 1.2.2.

In general, final specifications can be quantified with greater accuracy for functions that satisfy needs that are generic to all gas turbine engines, as opposed to those that are application-unique (reference section 1.3.1). The reason for this stems from the fact that the science behind these needs is generally well understood and has a large historical database from which to draw. For example, burn flow delivery requirements can be predicted with a relatively high degree of accuracy based on a thermodynamic analysis of the engine cycle. Thermodynamic cycle analysis tools and techniques have been improved and refined over many years and have attained a relatively high degree of accuracy and fidelity.

1.3.4 Design

This step represents definition of form or the physical representation of the control system components. Design features such as actuator piston size, pump impeller or gear size, alternator winding design, and fuel metering valve window configuration are defined to allow initial development hardware manufacture to begin. Typically, the conceptual, preliminary, and detail design phases (in total) for a particular control system component will represent 20% to 50% of the total non-recurring engineering (NRE) in the development program of that component. The magnitude of the design effort varies widely from component to component and depends on factors such as the component's lineage (whether the component is a "clean sheet" design or a derivative of a previous design), its complexity, and the number of interfaces with other components.

---

[12] Hague, Douglas C. "Description of a Turbofan Engine Product Development Process." SM Thesis, Massachusetts Institute of Technology, 2000.

1.3.5 Develop

This step represents manufacture of initial hardware to support subsystem integration testing leading up to first engine to test (FETT). For components that enjoy a low level of risk (i.e. design requirements are fairly firm), the initial hardware configuration in some cases will be the final production version. For those with high risk, functionally representative GTA hardware is chosen and integrated into initial subsystem testing. At this stage, some uncertainty regarding interactions between components can be resolved, but typically, a significant amount of uncertainty will remain until the initial engines are built and run.

1.3.6 Test

As the name implies, this step is focused on verification (i.e. ensuring that the engine, subsystems, and components meet requirements) and validation (ensuring the engine, subsystems, and components function as intended in the target environment, meeting customer needs). The data generated during initial engine testing at sea level and simulated altitude conditions will resolve the vast majority of uncertainty remaining in the control system requirements. A small subset of requirements such as those associated with the integrated thermal management system, installed engine bay environment, and external nozzle flap loads cannot be finalized without flight testing. Those types of risks must be managed and mitigated by simulation and test with simulated inputs and environments.

It should be noted that, in the current product development process, a significant portion of verification (i.e. flight certification) testing for control system components is performed in a bench environment (i.e. off the engine) at a component and/or subsystem level. Compared to other engine hardware, control system components are allowed to perform a greater percentage of verification testing in this manner primarily because they are, for the most part, installed

external to the engine ducts. This allows the engine environment to be simulated much more easily and accurately than a gas path component such as a turbine blade. An important feature of the proposed development framework is the relatively minor increase in technical risk due to the delay in gaining early engine experience. The timing of FETT data gathering and flight certification testing is discussed in greater detail in section 4.2.3.

1.3.7 Implement

At the completion of the development program, the propulsion system enters into revenue service for commercial programs or field service for military programs. All technical requirements have been addressed and the logistics and support systems are in place to support the operator.

1.4 Thesis Overview

Chapter 2 provides background information on the operation of gas turbine engines and discussion on the functional decomposition of a generic control system using the initial steps of the Generic PDP discussed in section 1.3.

Chapter 3 provides background on the system design process, the concept of set-based design, and the benefits of rapid and early experimentation.

Chapter 4 provides an analysis of the current control system component development process including the exploration of possible reasons for design iterations.

Chapter 5 presents an alternate approach to the control system component development process in which Generic Test Apparatus (GTA) hardware is utilized for initial subsystem and engine testing until uncertainty is sufficiently resolved. The framework provides a methodology

for its application, a discussion on optimization analysis to accelerate determination of final design parameters, and a business case analysis.

Chapter 6 presents an organizational analysis of the current requirement definition process and some recommendation on how it might be improved to resolve some of the tension between performance requirements and product attributes such as cost and weight.

Chapter 7 draws conclusions about the validity of the thesis hypothesis and proposes some areas for additional research to advance learning on the subject.

2 BACKGROUND

2.1 General Gas Turbine Engine Control System Architecture

Since gas turbine engines vary in application and functionality, so does the architecture of the control system. Of all subsystems that comprise the general form of a gas turbine engine, the control system probably enjoys the highest diversity of form of any of the subsystems. Particularly for aerospace applications, where cost and weight must be driven to absolute minimums, the architectural form of the control system will vary with the application (high thrust commercial jetliner, low thrust business jet, high performance military fighter, etc.) as well as customer-specific installation requirements (Airbus A-330, Boeing B-777, Lockheed-Martin F-22 fighter, etc.). Figure 2-1 shows a typical military gas turbine engine cross-section.

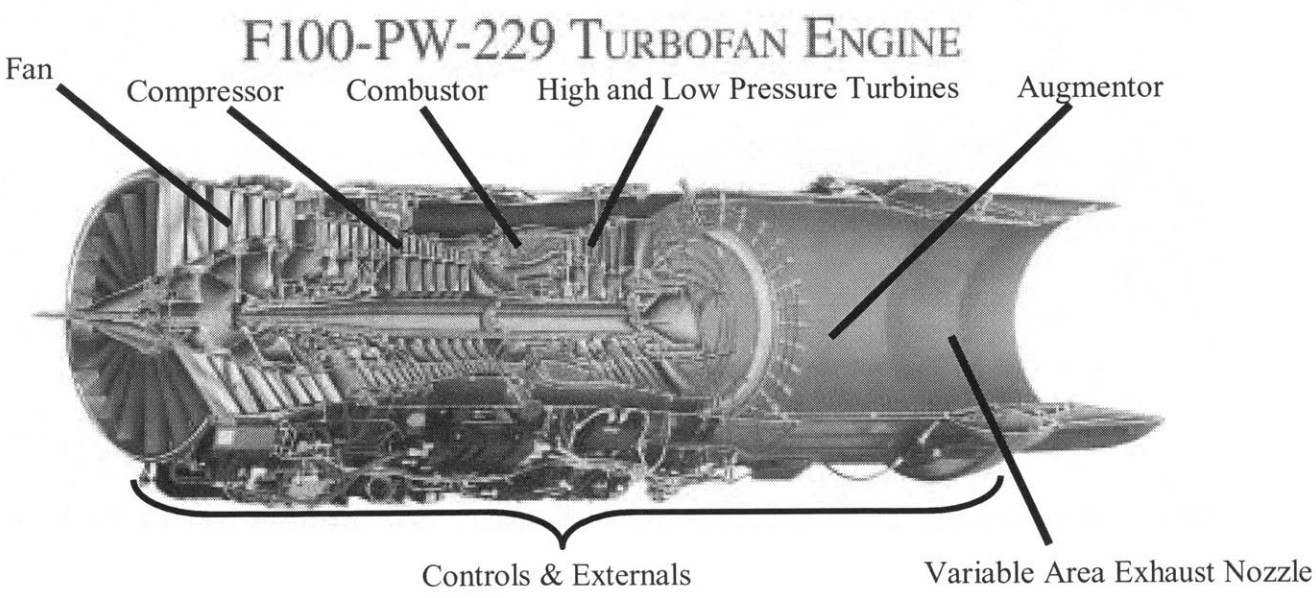


Figure 2-1. Pratt & Whitney F100-PW-229 Military Engine Cross Section

As its name implies, the main purpose of the control system is to control the engine. Since the engine's main purpose is to produce thrust, the primary and often most critical function of the control system is to control the necessary functions of the engine which allow thrust to be modulated. Jet engines produce thrust by employing Newton's 2nd Law,

$$F = M \bullet a$$

where F = Thrust

M = Mass Flow

a = acceleration of the mass flow

In this case, the majority of the mass flow is air that can range from a few pounds per second (pps) in the case of small engines used for auxiliary power units in commercial aircraft to hundreds of pps in the case of the large commercial turbofan engines that push the aircraft aloft. These massive amounts of airflow are produced by the engine fan that is turned by the fan-drive or low-pressure turbine (LPT). The LPT is turned by airfoils that extract energy from high temperature, high velocity gas which is produced by combusting a mixture of air and fuel in the engine combustor or burner. A portion of energy of this accelerated gas is extracted by the high-pressure turbine (HPT) to drive the compressor and the LPT to drive the fan and the remainder exits the engine through the exhaust nozzle. The fraction of airflow, by weight, passing only through the fan to the airflow passing through the core (compressor/combustor/high turbine) is known as the bypass ratio. Relatively low bypass ratio turbofans (such as military engines) produce the majority of propulsive thrust from the highly accelerated hot gas exiting the engine through the exhaust nozzle. High bypass ratio commercial engines produce the majority of propulsive thrust by a relatively low acceleration of an extremely high volume of air through the fan. This process of accelerating gas in a controlled fashion is initiated through the precise

metering of fuel into the engine combustor, atomizing the fuel by fuel nozzles, and igniting the mixture, and is sustained by the modulation of fuel flow. Gas generator fuel flow modulation is at the heart of all gas turbine engine control systems. Other control system functions, which will be discussed later, have been devised to optimize compression system efficiency and improve thrust modulation flexibility.

In general, control system architecture is an outcome of the following:

- Aircraft/propulsion system technical requirements
- Current state of technology maturity
- Historical success of similar products

The propulsion system technical requirements are usually flowed down from the customer and/or airframer in the form of a top-level engine specification. Particularly in the aerospace industry, technical risk level for new products weighs heavily in the selection process. Neimeyer concluded that engine programs that are launched with higher technology readiness levels (TRLs) were more likely to achieve TSFC commitments.[13] Although the research has not been conducted that would extend this conclusion *downward* to lower levels of decomposition, it is reasonable to assume that TRLs could also be correlated to control system performance commitments. As a direct consequence, when components or concepts with high technology maturity are selected for the architecture, historical success of similar products (either products of similar design or those provided by successful suppliers) plays an important role in guiding the design process down a particular path.

---

[13] Neimeyer, Jonathan K. "Framework for Risk Reduction in Gas Turbine Product Development." SM Thesis, Massachusetts Institute of Technology, 2002.

2.2 Functional Decomposition

2.2.1 Engine Control Needs

In a general sense, the needs of the engine that must be satisfied by the control system can be functionally decomposed as shown in Figure 2-2. The boxes with solid lines represent functions present on all gas turbine engines while the boxes with dotted lines exist on only those models requiring that functionality. For example, not all engines require variable geometry (e.g. variable compressor vanes or exhaust nozzle throat area), but all require metered fuel, an ignition source, and some method of communicating thrust request from the pilot or flight control system (mechanical, or digital) to the engine. The airframe is also shown at the same level as the engine with input to the "Manage Air Vehicle Heat Load" function. This function is becoming more prevalent on engines in modern aircraft that have integrated thermal management systems. The purpose of a thermal management system is, as the name would imply, to manage the heat generated by the subsystems on the aircraft by limiting heat generation as much as possible and by efficiently utilizing available heat sinks such as the ambient environment and fuel burned by the engine. As more electric/electronic systems and components are incorporated into aircraft designs, the need to reject heat in a cost and weight-efficient manner has increased in importance.

Gas Turbine Engine Control System Functional Decomposition

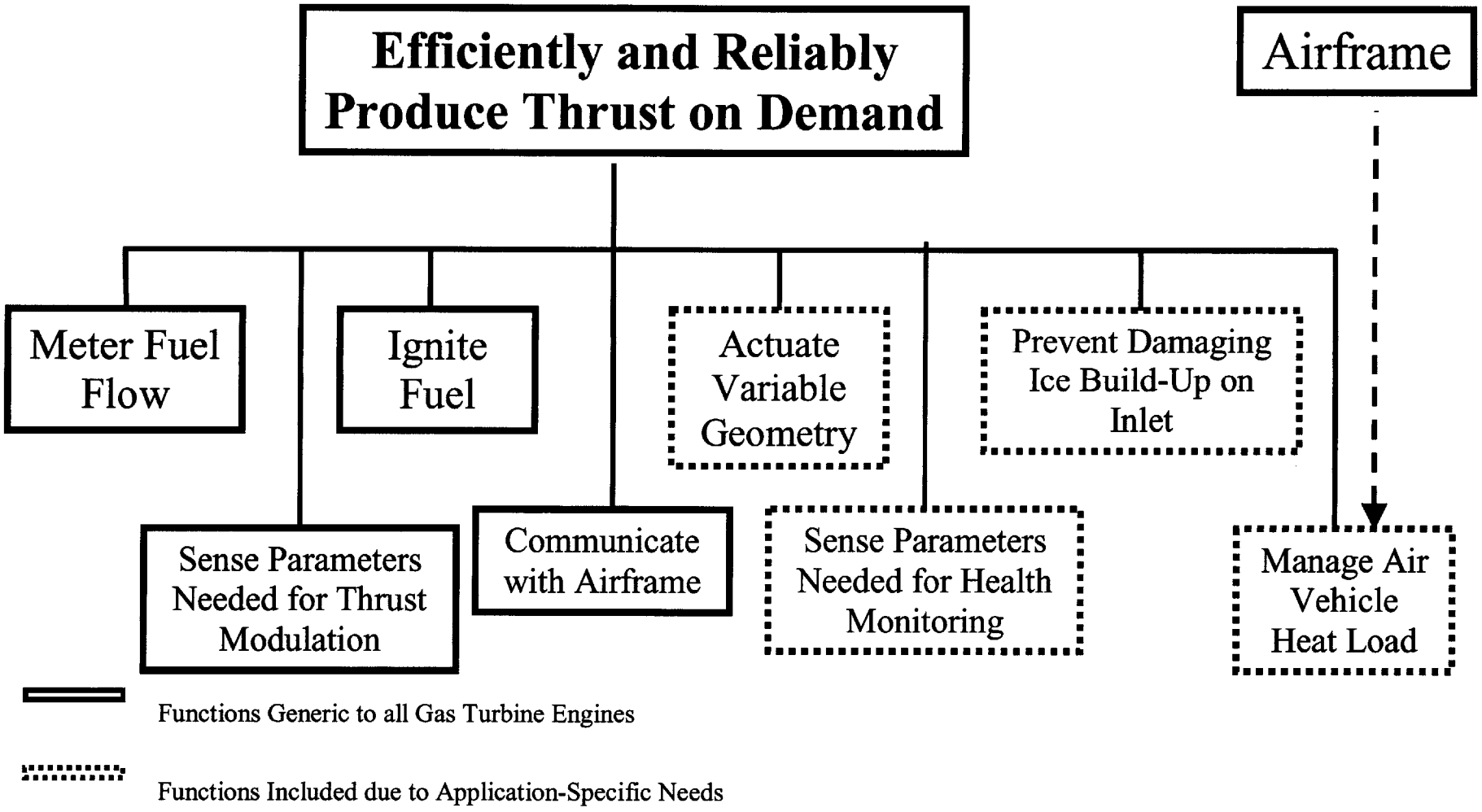


Figure 2-2. Gas Turbine Engine Control System Functional Decomposition Block Diagram

Each of the first-level functions shown in Figure 2-2 has multiple layers of functional decomposition below them. Lower levels of decomposition become increasingly dependant on the application of the propulsion system as mentioned in section 2.1. Table 2-1 provides a list of high-level control system functions with some examples of lower level functions at the next level of decomposition.

Table 2-1. Control System Functions with Examples of Subfunctions

Control System Functions	Examples of Subfunctions
Modulate Burned Fuel Flow	• Modulate Gas Generator Fuel Flow • Modulate Augmentor Fuel Flow (military-unique)
Ignite Fuel	• Ignite Gas Generator Fuel • Ignite Augmentor Fuel (military-unique)
Sense parameters needed for thrust modulation	• Sense Inlet Air Temperature • Sense Inlet Air Pressure • Sense Fan speed • Sense Compressor Speed • Sense Gas Generator Combustor Pressure • Sense Turbine Temperature
Actuate Variable Geometry	• Actuate Fan Variable Vanes • Actuate Compressor Variable Vanes • Actuate Exhaust Nozzle (military-unique)
Anti-Ice/De-Ice Engine Inlet	• Modulate Compressor Bleed Air • Control Electrical Heating Elements
Manage Air Vehicle Heat Load	• Limit Engine Lube Oil Temperature • Limit Engine Burned Fuel Temperature • Limit Engine Fuel Inlet Temperature • Limit Aircraft Subsystem Temperatures • Burn all Heat Possible • Recirculate Unburned Heat to Appropriate Heat Sink
Sense parameters needed for prognostic/health monitoring functions	• Sense Engine Vibration • Sense Lube Oil Debris • Sense Lube Oil and Fuel Filter Pressure Drop
Communicate with Air Vehicle	Airframe to Engine • Communicate Thrust Request • Communicate Air Data (e.g. altitude, airspeed, attitude) • Communicate Thrust Vector/Reverser Request • Communicate Inlet Distortion Parameters Engine to Airframe • Communicate Engine Performance Data (e.g. thrust, engine pressure ratio) • Communicate Engine Health Data (e.g. turbine temperature, rotor speeds, vibration levels)

All of the above functions can be put into four basic categories from the standpoint of the central controlling mechanism or "brain" of the control system, the electronic engine control (EEC)[14]:

- Effecting – functions that are imposed on the engine such as fuel modulation or variable vane positioning. Valves or actuators that are continuously positioned and normally have some feedback device with which to provide closed-loop control are termed "minor loops". "Major loops" refer to the overall engine control loops that are controlled by the minor loops (e.g. low rotor speed, airflow, high rotor speed, exhaust nozzle area, engine pressure ratio (EPR), etc.).
- Sensing – functions that acquire data about the condition or state of a part of the engine in order to perform some controlling or monitoring function.
- Communicating – transferring data within engine subsystems or to and from the airframe subsystems.
- Computing – Execution of software algorithms that take sensed or inferred information about the engine and/or aircraft, make appropriate calculations or correlations, and produce decisions regarding the control of engine effectors, assessment of the health of various aspects of the propulsion system, and determination of content and timing of communications with aircraft systems.

2.2.2 Control System Concept

At the highest level of abstraction within the control system, different concepts are evaluated to satisfy the functional needs identified in the previous step. In general, concepts for the four basic categories of functions described at the end of the previous section are evaluated first. The reason for this is that by identifying a single concept for a category of functions (e.g.

---

[14] The terms Electronic Engine Control (EEC) and Full Authority Digital Electronic Control (FADEC) are not necessarily equivalent in functional capability; they are used interchangeably in this work.

effecting), the cost of the EEC (a significant percentage of the cost of the control system) can be minimized using common circuitry. Utilizing common concepts for categories of functions also helps to reduce software costs and improve integrity through the internal reuse of common modules that interface with the hardware. For example, three common concepts for minor loop effectors are:

1. Single string in which the EEC controls a single-channel effector with a single-channel feedback.
2. Redundant active-standby in which a function has redundant effectors with one in control at a time, the EEC switching control to the other in the event of a failure.
3. Redundant active-active in which a function has redundant effectors with both in control simultaneously, each getting effectively half of the authority signal from their respective EEC channel. The three concepts are shown schematically in Figure 2-3.

For effectors, there are actually two levels of decomposition required to completely map function to form to the extent that the functional characteristics are completely defined. The schematics shown in Figure 2-3 show the first level of decomposition, defining the basic organization of subfunctions within an effector architecture. Architectural configuration at this level of decomposition will usually be driven by mission capability, safety, and failure effect requirements. For non-safety-critical effectors that have a minimal impact on mission capability (i.e. the ability to satisfy the intended mission or usage), a single-string effector architecture might be chosen to minimize cost and weight. For effectors with more stringent safety requirements, a redundant effector architecture might be needed. The primary differentiator between the active-standby and active-active concepts is the amount of perturbation or off-schedule operation that occurs during a failure situation. In the event of a subcomponent failure (either in the command or feedback side of the loop), the active-standby concept will typically subject the engine to a larger amplitude transient due to the amount of time required for the EEC to detect the failure and transfer control to the redundant channel. The active-active concept will

provide more of a "seamless" transfer from dual-channel control (in which each EEC provide ½ of the command signal simultaneously to position the effector and each receives a continuous feedback signal) to single-channel control since the amount of time for a channel to increase its command signal from ½ output to full output is nearly instantaneous. Also, in the event of a "runaway" command signal from one channel (e.g. a current driver failure), the other channel will tend to counteract the runaway by commanding the effector in the opposite direction of the failure, resulting in a smaller off-schedule transient of the effector. The disadvantage of an active-active system is that software complexity is increased substantially, driving up development and software maintenance costs.

The second level of decomposition required for effector loops is the selection of the types of effectors and feedback devices. Requirements such as frequency response, contamination resistance, and hysteresis will affect the concept selection of the effector device, such as a single-stage electro-hydraulic servo valve (EHSV), two-stage EHSV, or direct-drive valve (DDV). Requirements such as accuracy, linearity, and reliability will drive the selection of the feedback device for the effector loop. Typical devices used for aerospace control systems are linear variable displacement transducers (LVDTs), rotary variable displacement transducers (RVDTs), and resolvers. As mentioned previously, it is desirable to adopt a common effector loop architecture for a particular propulsion system in order to have a single electronic interface design for the multiple loops required. The number of effector loops can range from around a half dozen in the case of the somewhat simple commercial engine control systems to 2-3 times that many in the case of state-of-the art military systems which employ thrust vectoring, thrust augmentation, and/or vertical lift functions.

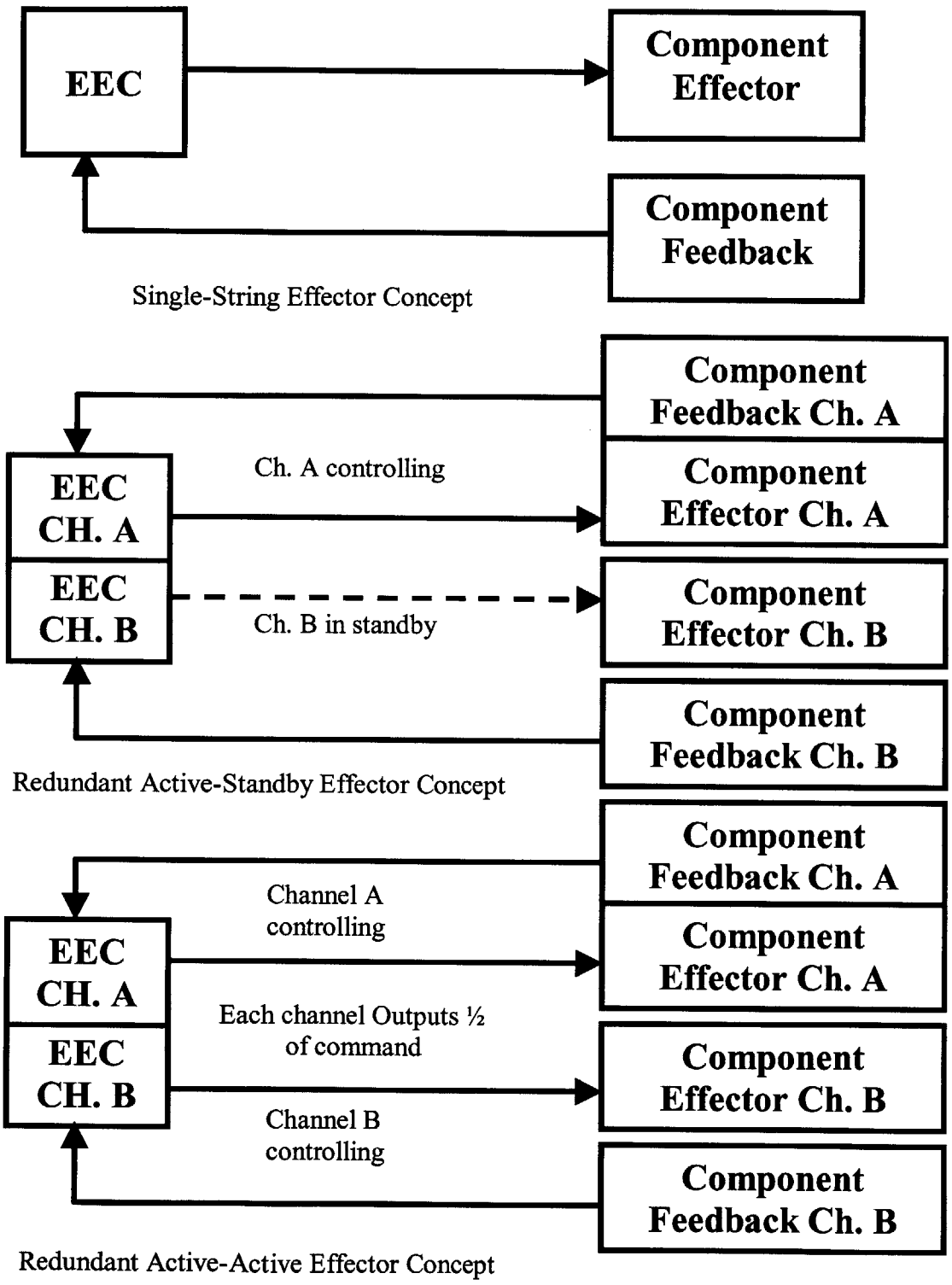


Figure 2-3. Examples of Minor Loop Control Concepts

Likewise, for the function category of Sensing as discussed at the end of section 2.2.1, control system concepts are selected to meet the engine needs as flowed down through functional decomposition. Let us consider the example of an inlet air temperature sensor (T2 sensor) that is needed as a basic input to schedule most of the engine operating parameters. In the days prior to the advent of electronic controls, a hydromechanical sensor concept (such as a helium-filled capillary tube) was utilized to sense inlet temperature. With the development of electronic controls, temperature sensing concepts switched almost exclusively to the use of dissimilar material junction devices such as a type K thermocouple. They were lighter in weight, less expensive, and more reliable than their hydromechanical counterparts were. As the use of on-board and off-board electronic systems such as RADAR electronic warfare systems became more widespread, the threat of Electromagnetic Interference (EMI) drove yet another architectural change to T2 sensors. One problem with the electrical signal generated by dissimilar material junctions was that it was an extremely low voltage signal (in the millivolt range) and was therefore susceptible to EMI-induced electrical noise. The issue of inadequate signal-to-noise ratio could be solved by either reducing noise or boosting the signal level. Reducing electrical noise in an ever-worsening EMI environment by adding shielding became a very costly and heavy option. This precipitated the incorporation of a resistive thermal device (RTD) that produces a signal in the range of volts rather than millivolts. This type of device uses the concept of correlating the change in resistivity of a metal (e.g. a platinum winding) to its temperature. They have demonstrated adequate accuracy and excellent signal-to-noise ratios for moderate temperature ranges (below about 1000 F). For temperatures above this level, a thermocouple is still the sensor of choice.

Similar to the progression of technology of the T2 sensor, the function category of communicating has progressed significantly over the past few decades. In general, communication concepts utilized between engine control system components and between the engine and airframe have progressed from mechanical to electrical to electronic (i.e. digital).

Let us examine the function of engine-to-airframe communication. This communication function can be thought of as a two-way exchange of information with different data associated with the different paths. The data normally communicated from engine to airframe generally serves the purpose of allowing the monitoring of critical engine functions such as turbine temperature, rotor speeds, and oil pressure. Concepts that provide this function in general have progressed in the following manner:

- Mechanical interfaces such as connecting a tube from the engine oil system to a gauge in the cockpit.
- Electrical analog interfaces such as providing an analog voltage signal from a transmitter in the engine oil system to a gauge or display in the cockpit.
- Electronic digital interfaces such as an ARINC or MIL-STD-1553B in which a sensor in the oil system is read by the electronic engine control which transmits oil pressure in digital format to an airframe computer, which in tern transmits the data to cockpit display computer for use by the pilot.

The data transmitted from airframe to engine normally consists of pilot or flight control computer commands for thrust request and in certain cases thrust vectoring (thrust reversing for commercial aircraft or thrust vectoring for some military fighters). The following represents a general progression of concepts over the past few decades:

- Mechanical interfaces such as a push-pull cable from the throttle in the cockpit to the fuel control on the engine.
- Electrical analog interfaces such an analog voltage signal from the engine anti-ice switch in the cockpit to the electronic engine control, which is used to enable the engine anti-icing system.
- Electronic digital interfaces such as an ARINC or MIL-STD-1553B in which a resolver or rotary variable displacement transducer (RVDT) on the cockpit throttle quadrant is read by an airframe computer which transmits thrust request in digital format to the electronic engine control. The control in tern schedules fuel flow and other engine effectors to modulate engine thrust, satisfying the request.

2.2.3 Identification of Functions and Form

2.2.3.1 Functional Schematic Creation

Following control system functional decomposition and concept selection, the next step toward identification of form (i.e. assignment of functionality to components) is the generation of a functional schematic. All of the lowest-level functions required to meet the needs of engine control and diagnostics are included on the schematic in the context of the concept selected to implement the function. Let us assume for example that the high level function "Actuate Variable Geometry" was decomposed to

- Actuate Fan Variable Vanes (FVV)
- Actuate Compressor Variable Vanes (CVV)
- Actuate Exhaust Nozzle Throat Area (ENA)

Let us also assume that the concept selected for all of these sub-functions is a fuel-powered (termed "fueldraulic") actuation system consisting of a fuel pumping function, an actuator (i.e. effector) function, fuel connectivity between the pump and actuator, and a closed-loop control connectivity between the actuator and FADEC. This is represented by Figure 2-4. Note that the pumping and actuator functions may give the impression that a particular form is inferred, but the functions are only represented pictorially in this way for clarity.

Other subsystems and functions are added to the functional schematic to gain a holistic view of potential packaging options and component architectures. Component architectural options can still be traded at this point based on parametric targets (product cost, weight, reliability, safety, mean time to replace, vulnerability, etc.) that are allocated at the module level. Component requirements that are allocated at the module level (fan, compressor, turbine, control system, etc.) are usually delayed until component architectural decisions are made. Examples of these decisions are whether to use a centrifugal or gear pump for burn and/or actuation flow, whether to use a master/master, master/slave, or slave/slave actuator configuration for a multiple-actuator function (such as an exhaust nozzle), or whether to use metering valve/head sensor or in-line pressure regulating valve for fuel metering.

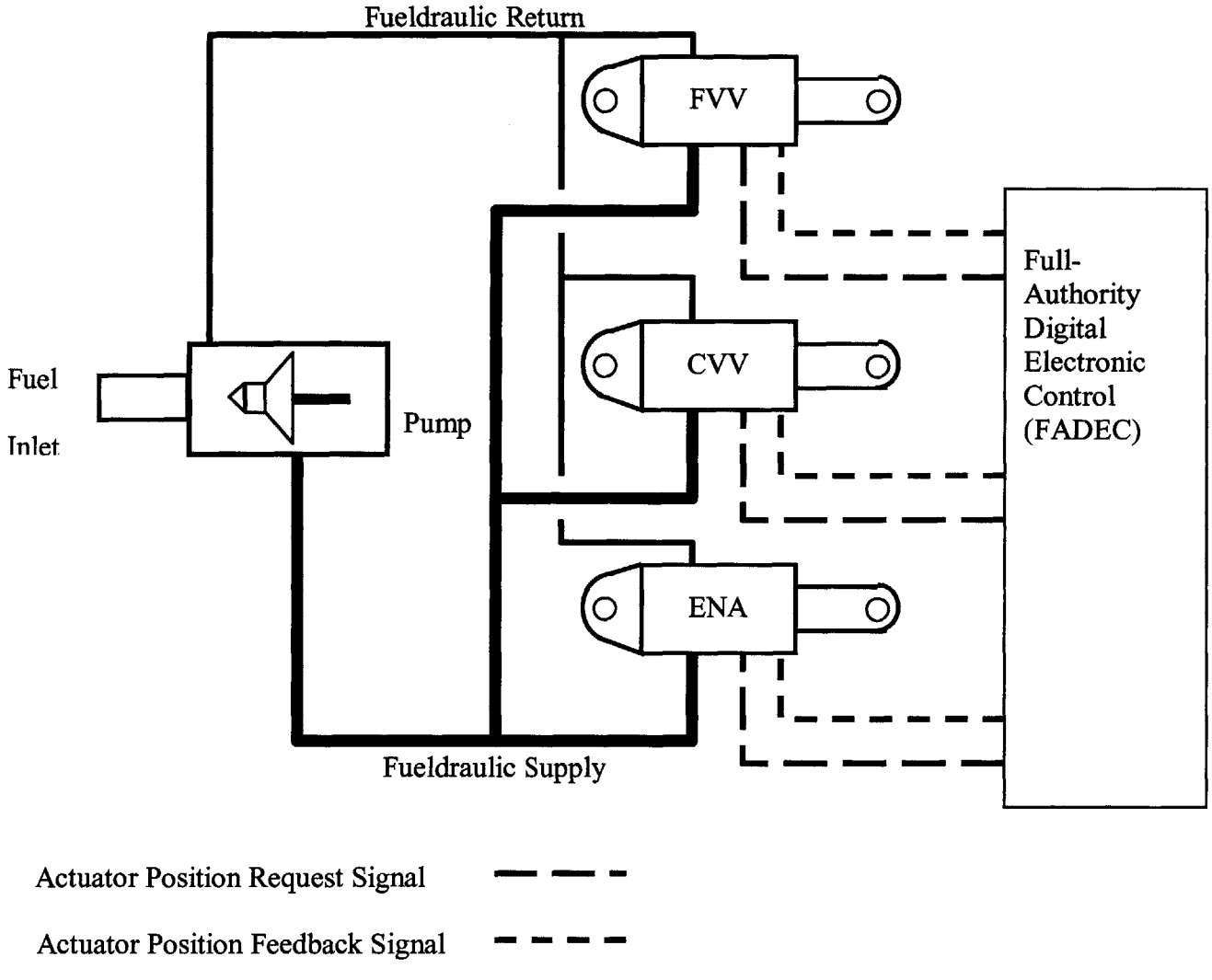


Figure 2-4. Example of an Actuation Subsystem Functional Schematic

As an example, if the propulsion system concept includes a variable area axi-symmetric (i.e. round) exhaust nozzle, a flap train configuration that has demonstrated historical success is a multiple overlapping flap and seal design that requires actuation via the translation of a relatively stiff synchronization (sync) ring. In order to prevent binding of the sync ring during translation, multiple actuators are connected at equally spaced locations about the circumference of the ring. The load reacted by each of the actuators must be closely balanced to minimize the required sync ring stiffness, thereby reducing weight. The load between the actuators can be balanced by actively controlling the position of the individual actuators (i.e. a "master-master" actuator architecture as shown in Figure 2-5). Truly balancing the load requires matching actuator positions to a very tight coplanar tolerance during transient as well as steady-state operation. This might require extremely accurate feedback devices and very high bandwidth servos, both of which might push technology limits let alone increase cost.

An alternate method of balancing the load is to convert electrical commands to hydraulic signals in a central controller or control manifold, then distribute the extend/retract hydraulic signals to simple actuators (i.e. a slave-slave actuator architecture as shown in Figure 2-6). This architecture significantly reduces the dynamic load balancing risk by adding a central control function. This also means adding an additional component and interconnecting plumbing, although the actuators are significantly simplified. A trade study must be performed on both configurations to quantify attributes of merit for each configuration such as product cost, weight, reliability, safety, maintainability, vulnerability, technical risk, and performance.

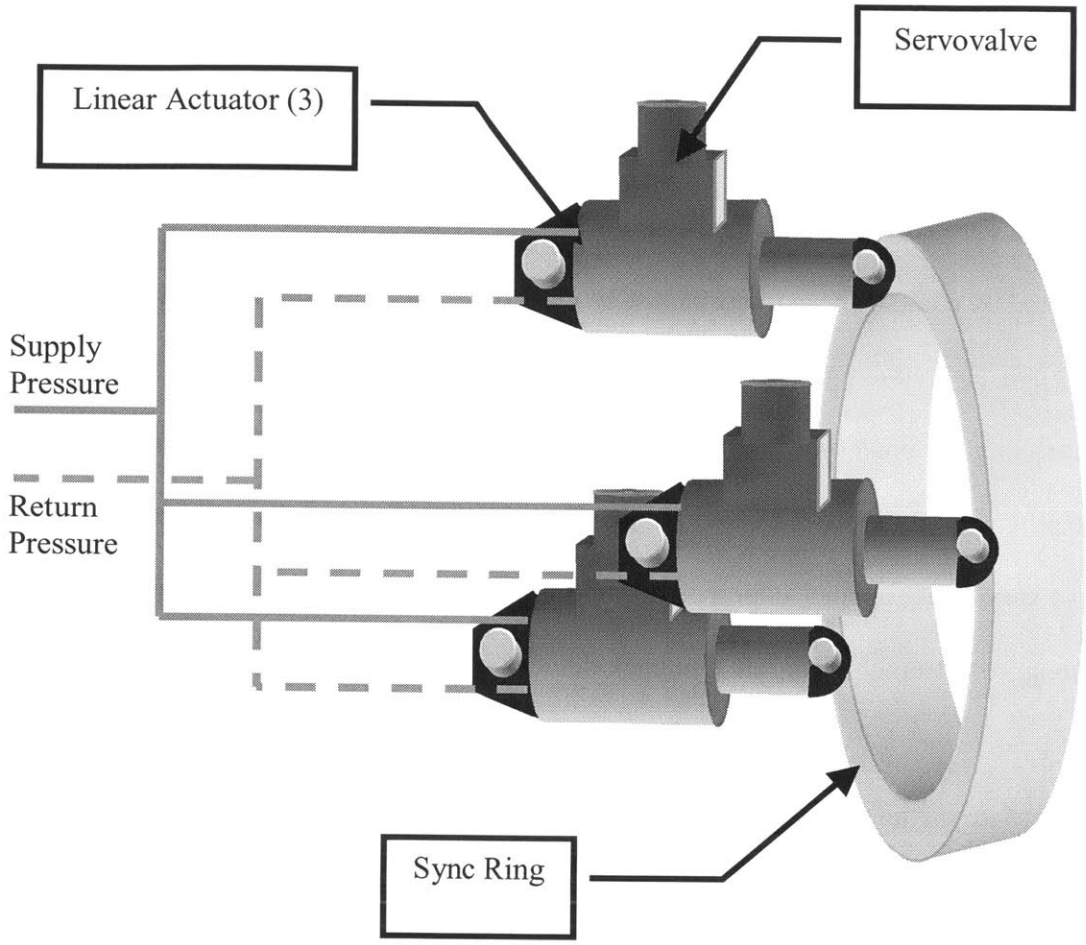


Figure 2-5. Master-Master Actuation Architecture

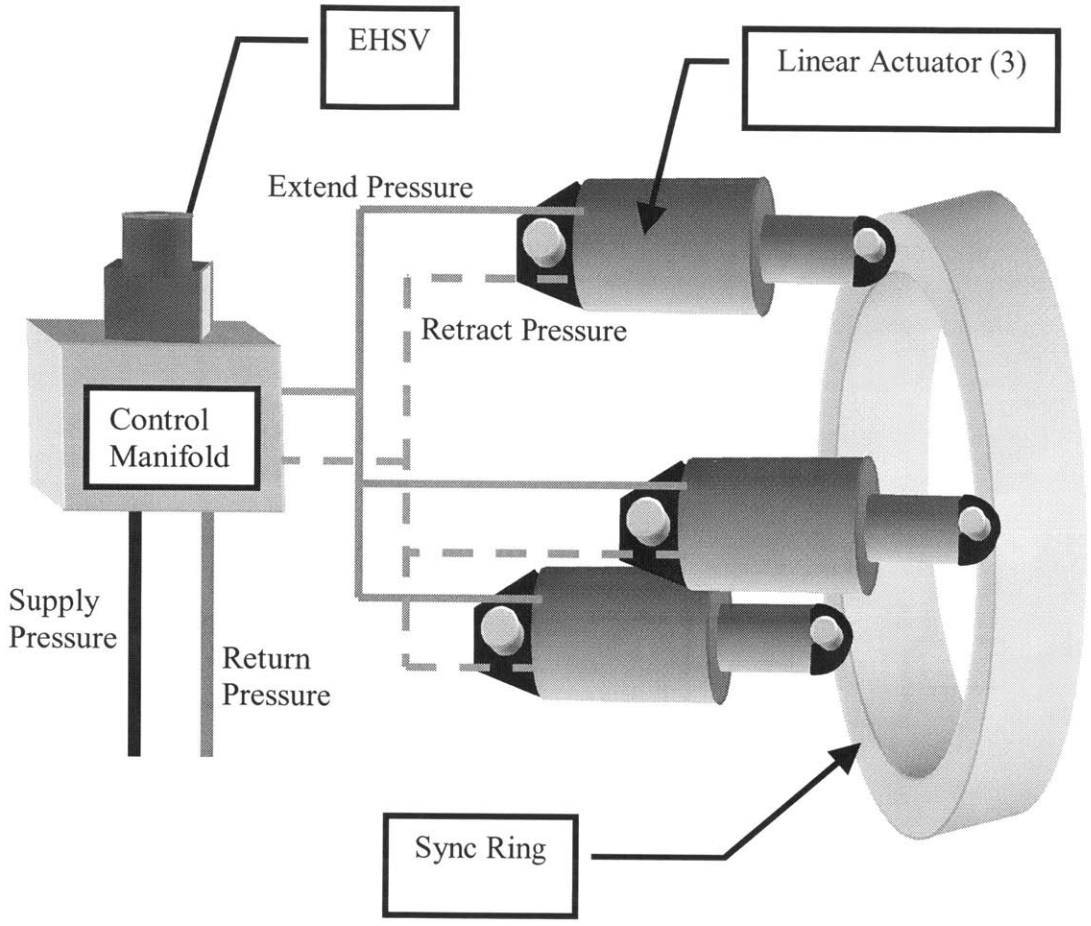


Figure 2-6. Slave-Slave Actuation Architecture

2.2.3.2 System Schematic Creation

The functional schematic is the predecessor of the system schematic, which allocates function to form and includes all control system components and interfaces. A simplified example of a portion of a system schematic is shown in Figure 2-7. The actuator function has now morphed into a master actuator with a single servo valve and a linear variable displacement transducer (LVDT) for feedback position to the FADEC. The pump supply pressure has been

shared between the actuator and the fuel metering unit (FMU), which performs the fuel metering function. The metering valve in the FMU is positioned by a servo valve (presumably of similar architecture, but not necessarily of the same specs) and has an LVDT for feedback similar to the actuator. Transition from the functional schematic to the system schematic requires the knowledge of component form (i.e. the type of pump, feedback devices, etc) but does not normally capture any detailed requirements such as piston size, flow capacity, or pressure values.

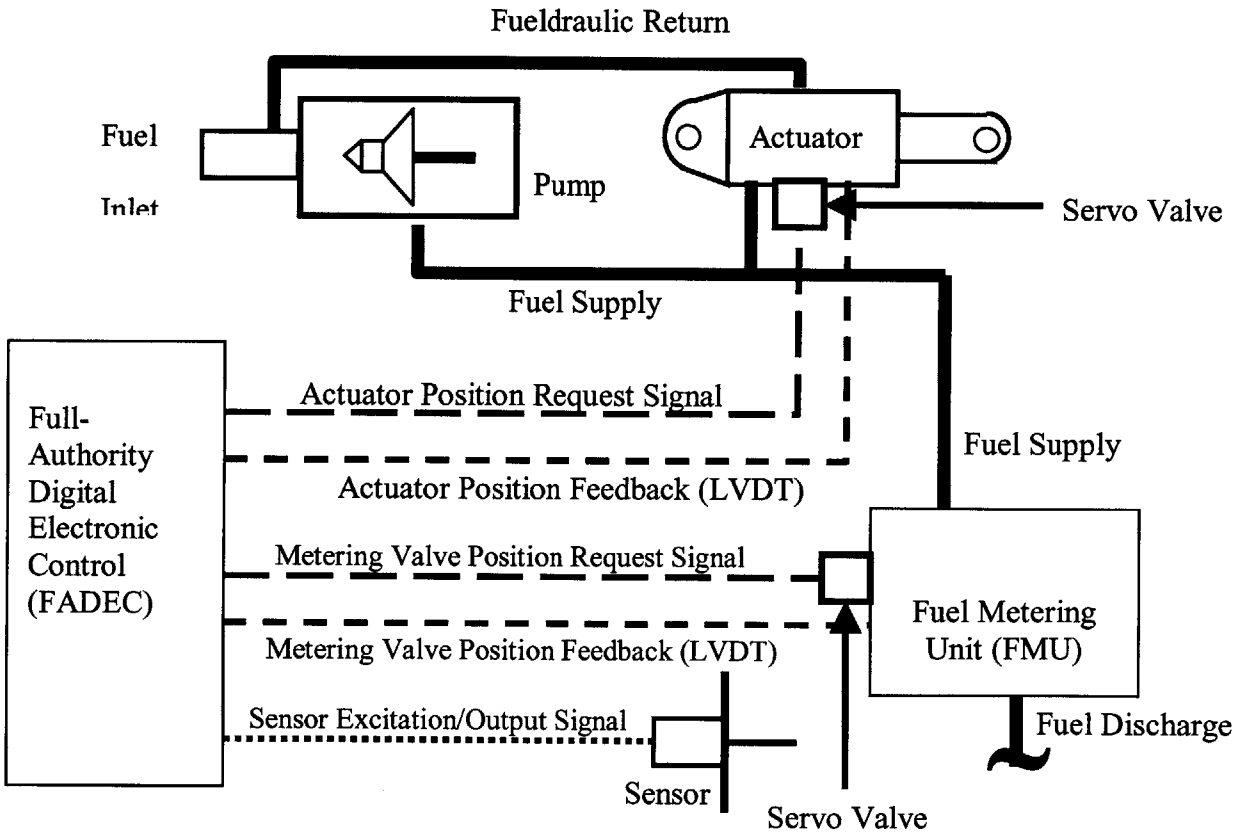


Figure 2-7. Example of a System Schematic

System schematics for control systems that provide a high degree of functionality can grow quite large and be very complex. Figure 2-8 shows an example of a control system schematic for a mature military afterburning turbofan engine[15]. The major components and their interconnectivity is represented. Since the fuel system contains several different pressure levels to perform different functions, they are represented with different colors and/or patterns. Since the primary intent of the system schematic is to show form and function at the control system level, form at the component level is abstracted emphasizing the connectivity between the components.

Figure 2-9 shows actual hardware and connectivity for a Pratt & Whitney PW4000 commercial engine components. This level of detail shows the actual form and connectivity of components and is usually utilized for field support and training materials.

---

[15] Support Services, Pratt & Whitney, "The Aircraft Gas Turbine Engine and Its Operation", United Technologies Corp, 1982., 113.

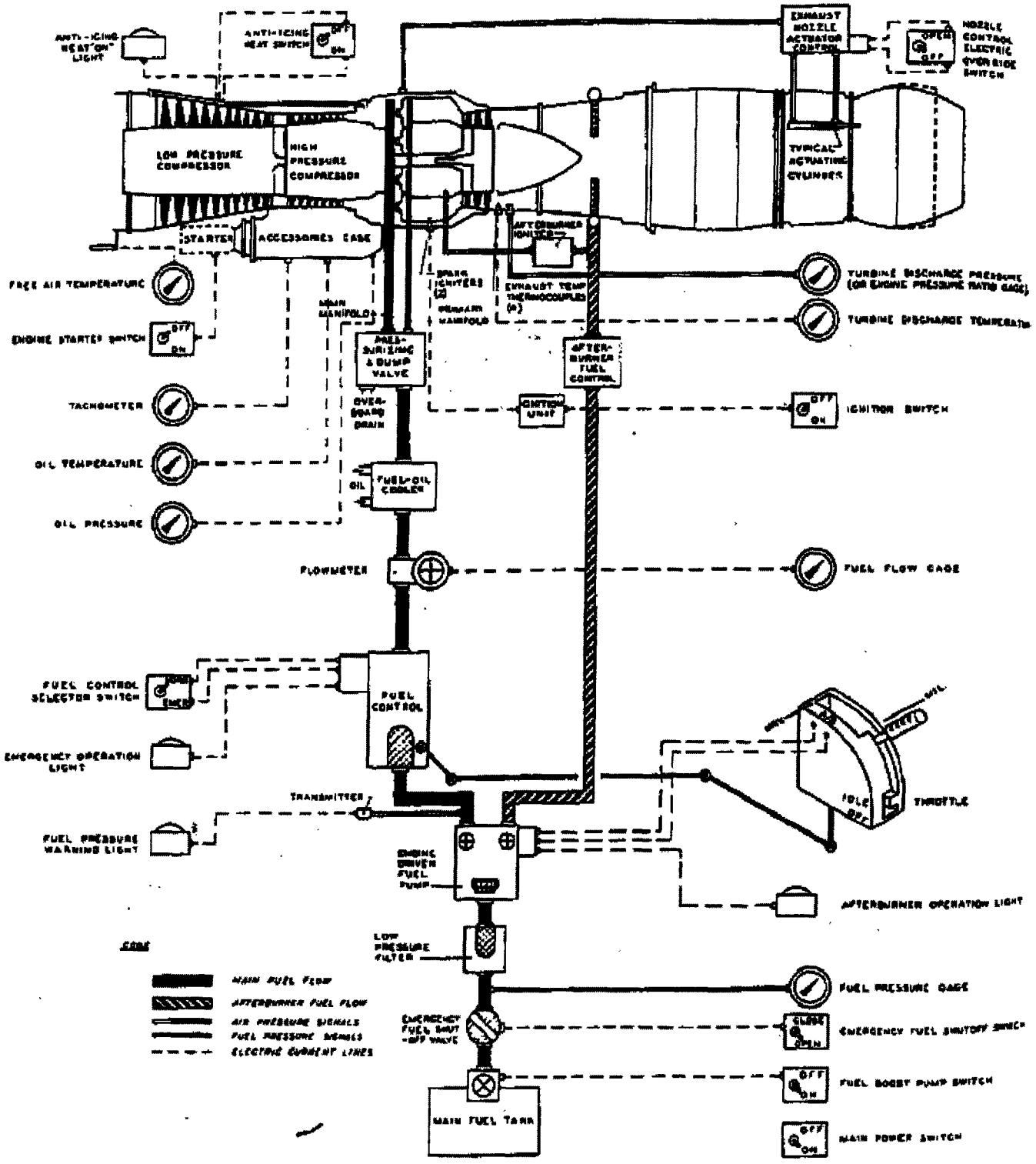


Figure 2-8. Typical Control System Schematic for an Afterburning Turbofan

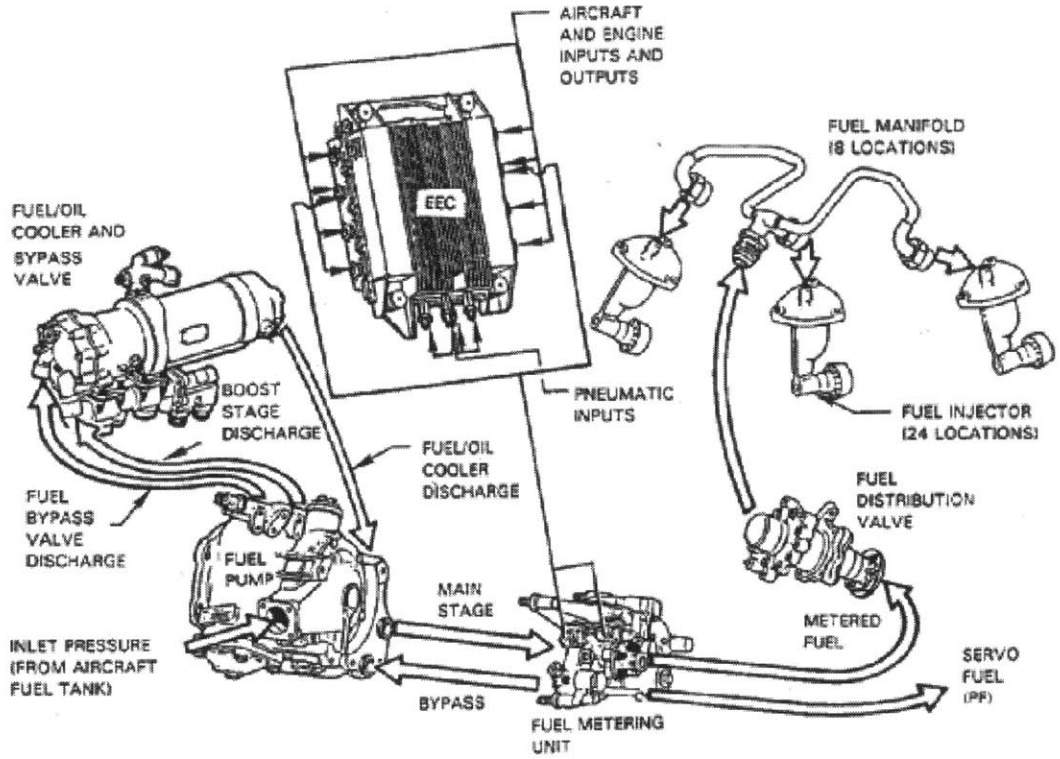


Figure 2-9. Fuel Control System with Electronic Engine Control.

2.3 Chapter Summary

This chapter served to provide the reader with a basis of understanding of gas turbine engine control system general architecture. Specific features and hardware implementations were discussed in the context of system decompositions that are geared toward satisfying engine functional needs. Various artifacts of the design process, such as system schematics, were also described and examples were provided. The chapter concluded with some specific pictorial examples of mature fielded systems.

3 RELATED WORK

3.1 Understanding the System Design Process

In order to effect change (hopefully for the better) in the aerospace component development process, it is first necessary to have a clear understanding of how, at least theoretically, the current process is structured. Since this work focuses on understanding and developing product requirements and specifications in the face of uncertainty, it is prudent to review the process from a general point of view. The issues under investigation are captured in the Concept Development phase as described by Ulrich & Eppinger,[16] which is shown in Figure 1-7. Drawing a parallel to the PW jet engine development process, the step of establishing target specs, which immediately follows identification of customer needs, is indeed a high level analysis that establishes basic engine-level requirements such as thrust-specific fuel consumption (TSFC), weight, low observability, etc. At this point, the design space for the control system is extremely large since control system requirements are determined primarily by the engine design and the engine design is in its infancy. The next three steps involve concept selection, following which is setting final specifications. At this point, the engine requirements are fairly well known to the point that the major module development plans (including the control system) can be launched. For the next lower level of decomposition (i.e. the major module level), this process can be *re-entered* and executed in the same manner that it was at the engine level. The current jet engine development process proceeds through the setting of final specifications for the control system based on analysis and simulation. This hits at the heart of the issue of designing with requirement uncertainty that Ulrich & Eppinger do not address.

---

[16] Ulrich, Karl T. and Steven D. Eppinger, Product Design and Development, McGraw-Hill, Inc., 2000, pg 18.

3.2 Set-Based Design

Toyota has been a pioneer in the field of delaying design decisions in order to maintain large design spaces well into a development program, a practice known as set-based design. Even though Toyota is credited as one of the originators of concurrent engineering (CE), in which products and their manufacturing systems are simultaneously engineered, the company does not seem to follow western practices of highly structured design processes with often collocated multidisciplinary teams. Moreover, while typical CE practices tend to drive design decisions early to freeze specifications, Toyota maintains a larger design space early on, delaying these design decisions until very late in the process[17]. The early convergence to specific design specifications based on highly uncertain requirements results in non-optimum components. While Toyota seeks to evaluate many potential solutions at early phases of development programs, (primarily in the area of styling), this approach really does not apply to the aerospace propulsion business, since product success is not primarily judged by aesthetics, but by technical performance, reliability, and cost. Although Toyota practices set-based design for different reasons than PW/HS might, doing so has apparently worked for Toyota and certainly appears to be beneficial to PW/HS.

3.3 The Benefits of Experimentation

Thomke explains the benefits of experimentation in the realm of innovative product development[18]. In the design of complex components and systems, organization plays a very basic and important role in the efficient design, development, and production of the components

---

[17] Ward, Allen, Jeffrey K. Liker, John J. Cristiano, and Durward K. Sobek II. "The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster." Sloan Management Review, Vol. 36, Issue 3 (Spring 1995), 43.
[18] Thomke, Stephan, Enlightened Experimentation – The New Imperative for Innovation, Harvard Business Review, 2001, 69.

comprising the system. As discussed in the previous section, implementation of concurrent engineering has substantially improved both the development cycle time and quality of the results through corroboration of a multi-disciplinary team. In a like manner, rapid iteration can be achieved more easily if specific attention is paid to organize for it with the correct key skills. In the early development phase of control system requirements, rapid iteration usually means building analytical models and simulations to explore the design space. This requires a cross-section of systems analysts, modelers, and project engineers, none of which are necessarily bent on nailing down detailed component design specifications, but are open to exploring the design space and devising hardware to gather this data during initial engine testing. This leads to another point Thomke makes regarding front-loading; identifying problems upstream, where they are easier and cheaper to solve. Development teams should consider lower-fidelity, relatively cheap tests early in the program and leave the higher fidelity, more expensive tests for the production version. This strategy can be applied directly to the creation of low-fidelity prototype control system hardware to be used early in an engine development program to gather key design requirement data with which to optimize the production design. Higher fidelity tests such as those required for flight certification are performed on the final version of hardware after significant learning has occurred. This learning about the engine environment will also serve to confirm the conditions under which the certification tests are performed.

3.4 Chapter Summary

This chapter provided the background to assist in framing gas turbine engine control system design within the general system design process. Discussion was also included on the importance of maintaining system design flexibility and of early experimentation. Toyota's results seem to indicate the success of set-based concurrent engineering. While Toyota may

follow this process to optimize different parameters than PW/HS, the ultimate goal of improved customer satisfaction and increased market share is common between them. Finally, early testing has been a longstanding risk reduction tool, but Thomke's discussion on making use of lower-fidelity early prototypes to enable efficient learning is a central theme of this work.

4 ANALYSIS OF CURRENT DEVELOPMENT PROCESS

4.1 Integrated Product Development Process

Gas turbine engine product development at Pratt & Whitney (PW) is accomplished in accordance with the Integrated Product Development (IPD) process, which drives developers, from the outset, to consider all aspects of product life cycle from early concept through development to retirement and disposal. Figure 4-1 shows the relationship between IPD activities at the program, system, module, and part levels at PW[19]. In a similar manner, Hamilton Sundstrand (HS) has adopted the IPD process for aerospace component development. Figure 4-2 shows the HS Passport Review Process[20] that supports the PW IPD process shown in Figure 4-1. HS passport component reviews (P0-P4) are held at key decision gates throughout the program, and are aligned with similar decision gates in the PW process. The HS P0-P4 reviews have been superimposed onto the PW process to show the connectivity between the processes. It should be noted that the HS Passport Review process is generic to all HS customers (Pratt & Whitney, Rolls Royce, General Electric, Boeing, Lockheed-Martin, etc.) and may require minor tailoring to align with different customer's processes. HS and PW hold a special relationship resulting from a reorganization orchestrated by the companies' parent, United Technologies Corporation (UTC) in January of 2001. This reorganization involved transferring the entire controls and externals organization (i.e. the "Module Center") from PW to HS, which transformed HS from a component supplier to a system provider. Subsequent discussion will focus on the PW/HS development process due to the visibility and insight available.

---

[19] Internal P&W Documentation
[20] Internal HS Documentation

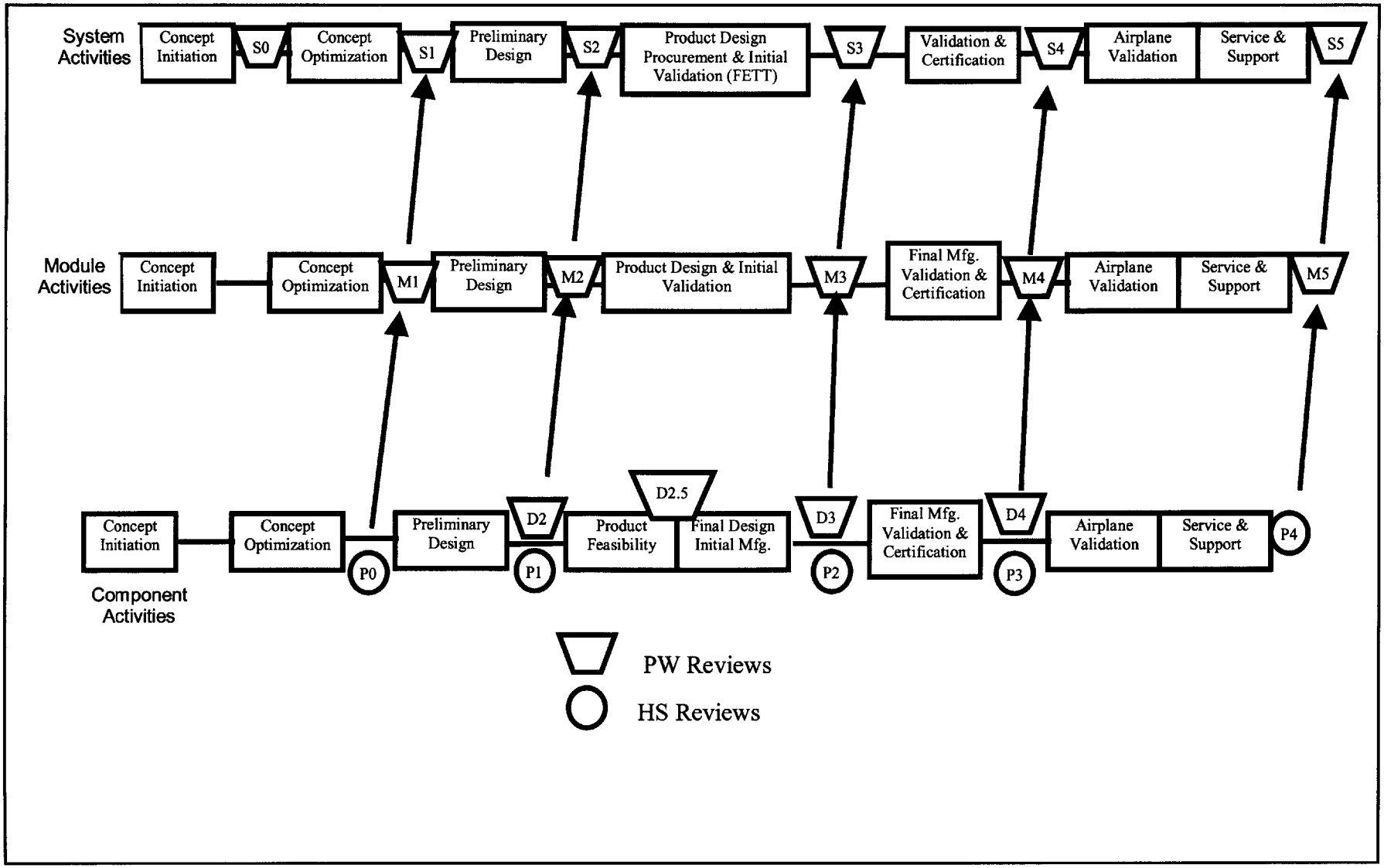


Figure 4-1. P&W IPD Activity Linkage

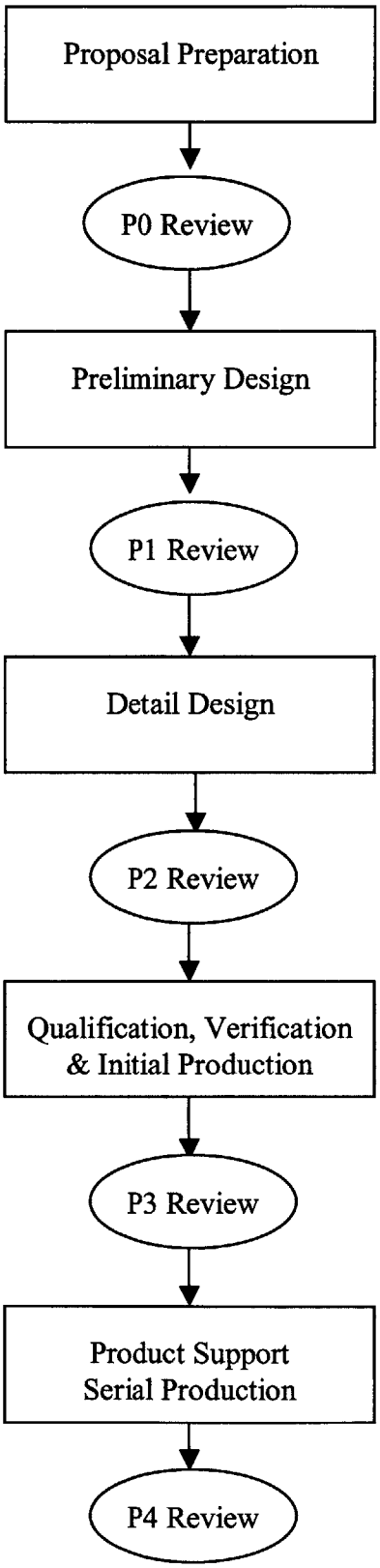


Figure 4-2. Hamilton Sundstrand Passport Review Process

4.2 Requirement Flowdown

4.2.1 Control System Design

Particularly when propulsion system requirements are decomposed to the control system level (as opposed to the control system component level), the requirement decomposition and flowdown tasks are cornerstones of the system design process. Figure 4-3 shows a typical development program for a gas turbine engine control system from program launch to flight qualification. This figure is similar to Figure 1-1 but expands the control system development process into greater detail.

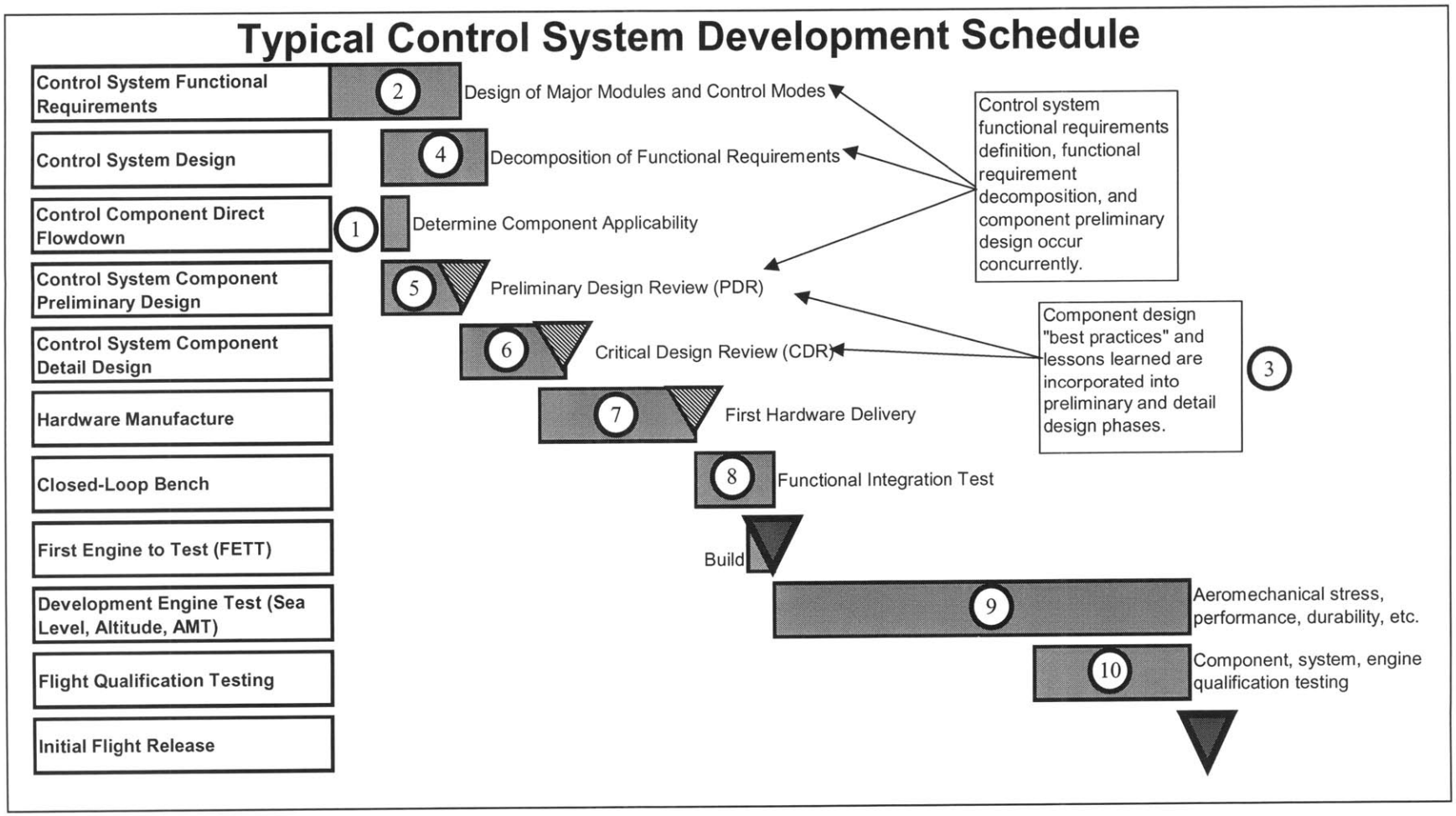


Figure 4-3. Typical Control System Development Program

Control system requirements are flowed from three basic sources:

1. Requirements from the engine specification that require no decomposition and flow down directly to components. These consist mainly of "generic" requirements directed at the construction of the components such as structural margins, operational envelope (altitude/Mach number envelope of operation), and maintenance/support guidelines. Requirements of this type are applied to the various components by merely assessing their applicability based on the nature of the component. For example, certain EMI requirements imposed on the engine would be applied to components to which an electrical harness connects, but would not apply to non-electrical components such as plumbing tubes and brackets. This flowdown process is represented by the symbol ①
2. Requirements that must be derived from the engine specification. These consist mainly of functional requirements originating in the engine specification that are decomposed to the major "module" level. A good example of this is thrust range, in which all major engine modules play a part. At the propulsion system level, thrust range is normally specified as a maximum allowable thrust at the idle throttle setting (in order to provide controllable aircraft operation during taxi and maximize aircraft landing gear brake life) and a minimum thrust at intermediate/maximum throttle setting to provide the required aircraft performance specified by the aircraft manufacturer. They also include "generic" requirements that apply to all components but vary in requirement level or value depending on specific component attributes. For example, the thermal environment for each component will depend primarily on the component's axial location within the nacelle and on the presence of heat sources other than the engine case such as bleed air ducts. This flowdown process is represented by the symbol ②

3. Requirements associated with lessons learned and best practices. These are intended to provide best commercial practices for requirements that are not otherwise specified by the customer. This category of requirements is normally maintained at the module center level within PW since the module centers provide "one-stop-shopping" for that particular module type. For example, the turbine module center provides the design, manufacturing, delivery, and support services for all turbines for all programs (military and commercial). Similarly, HS, being the controls and externals "module center" for PW programs, provides "cradle-to-grave" support for all hardware associated with controls and externals. This flowdown process is represented by the symbol ③

The first and third categories above can be flowed down in a very straightforward manner since no complex calculations or simulation is involved. A certain amount of "engineering" must be applied even to requirements that are directly flowed down. Structural integrity requirements must sometimes be interpreted to determine the applicability to various components. Since the control system contains a wide diversity of different component types (small electrical sensors, electrical harnesses, ignition components, fuel controls, actuators, pumps, electronic controls, etc.) all non-decomposed requirements do not necessarily apply to all components. For example, rotating parts might have a 25% overspeed burst margin imposed on them for safety reasons. This requirement obviously applies to the turbomachinery, but also applies to fuel and oil pumps, accessory gearbox, and electrical generator. It would not apply to other control system components since they do not contain rotating parts. On the surface, this would seem to be a mundane task. However, writing requirements in a clear, unambiguous manner without over-specifying or over-constraining is a difficult task that is often taken for granted and performed poorly. In many cases, coordination with the customer is required to

understand the context and intent of the requirement in order to interpret and apply it properly. The process of determining applicability of general requirements to the various control system components can be initiated concurrently with other engine modules, allowing some design activity to occur in the early stages of the development program.

The second category above embodies the classic systems engineering approach of requirement decomposition. Engine-level functional requirements are evaluated and decomposed into module level requirements in a (hopefully) solution-neutral manner. Specifying control system requirements in this way maximizes the design space going into the subsystem and component design phases. In some cases, however, the system-level analysis to define minimum acceptable performance levels using a "bottom-up" methodology would have to be handled in an iterative manner (assume a performance characteristic, perform simulation analysis, determine resulting margin, iterate), which would be prohibitively lengthy and tedious. In these situations, certain performance parameters are fixed based on historical hardware implementation in order to converge on a system solution more quickly. Unfortunately, this serves to narrow the design space by locking in the performance of a particular hardware implementation. This is an issue left for future research.

Once the control system derived requirements are determined, analysis is performed to further allocate functional requirements to components. On Figure 4-3, this flowdown process is represented by the symbol ④

For example, if gas generator fuel flow accuracy (request to delivered) is specified as ± 5% of point at the control system level, trade studies are performed to identify candidate architectures which will meet this performance requirement as well as other requirements such as reliability, safety, durability, and EMI with minimum cost and weight. The trade study considers

all potential system pieces including the fuel metering effector, feedback device, FADEC interfaces, pumping system, harnesses, plumbing, and fuel nozzles. When the trade study identifies the optimum concept, performance allocations to each of the pieces are flowed to the appropriate components. In many situations, the product cost and weight associated with the design solution that meets all other performance requirements will cause the components that must implement the solution to exceed established targets.

4.2.2 Control Component Design

The preliminary design phase for control system components begins following the allocation of function to form via the system schematic (reference section 2.2.3.2) and the establishment of some preliminary component level requirements. On Figure 4-3, this task is represented by the symbol (5)

Recalling from the previous section, some component level requirements are directly flowed down from the engine specification with no decomposition. Thus, it is possible to begin component preliminary design with these requirements and a system functional schematic. Progression through the product definition phase is influenced by the type of component and by the timing of the definition of the component's functional requirements. Simple components with specific functions such as a spark igniter can be nearly completely defined fairly early in the engine design process since the requirements are simple and straightforward, the environment is generally well defined (gas path pressures and temperature predictions at this point are reasonably accurate) and the number of interfaces are, by definition, few. All of these factors tend to drive out ambiguity and uncertainty. Complex components, however, usually require a great deal more decomposition and analysis in order to allocate requirements in an optimized manner. To compound matters, there is also inherent risk in this phase of the program due to the

uncertainty existing in the control system requirements analysis that is being performed concurrently at the next higher level of system hierarchy. Continued iteration in the control system functional requirements will trickle down causing re-do's in the control system decomposition and component preliminary design tasks. The preliminary component design phase culminates in a Preliminary Design Review (PDR), which corresponds to a P1 Review in the HS Passport Process. At this point, the component physical envelope, architecture, arrangement, and preliminary interfaces and requirements are established. Many important design decisions such as material, component architecture, and maintainability features are made at this stage that can heavily influence product cost and weight.

The next phase of component development is detail design during which the remaining design decisions are made in order to completely define component configuration. This phase is represented in Figure 4-3 by the symbol (6) . For some components, the remaining requirement uncertainty is low allowing detail design to proceed with reasonably low risk to downstream iteration. For other components (generally those directly affected by aspects of the engine design that remain uncertain), assumptions must be made on requirements that are still being iterated at higher levels of the design hierarchy. In order to ensure that engine performance and operability requirements are not limited by the capabilities of the control system, these assumptions err on the side of conservatism to account for the existing uncertainty. For example, if exhaust nozzle actuation load at the design point is estimated nominally at 10,000 lbf with an uncertainty of $\pm$ 25%, the actuation system is designed **worst case** to meet 12,500 lbf (10,000 + 25% uncertainty). This means also adding the tolerances of the actuation system (pressure scheduling error, minimum spec pumps, minimum piston areas, etc.), thus further adding to the overdesign situation and contributing to unnecessary cost and weight

penalties. The detail design phase culminates in Critical Design Review (CDR), which corresponds to a P2 review in the HS Passport Process. At this point, all analysis and detail design aspects should be complete such that manufacturing can begin.

Over the past decade, increasing emphasis has been placed on early development of manufacturing processes in order to learn out the production process during the course of the development program. This strategy has proven successful in avoiding "production transition" issues associated with manufacturing development parts from one source and using another source for full-scale production. This up-front development activity, while proving to be a wise investment, puts additional pressure on the component development process by forcing early design decisions. Manufacturing lead times vary widely depending on the component's complexity, manufacturing process, selected material, and design tolerances. Lead times can range from a few weeks for simple valves to nearly a year for components with complex castings or forgings. The manufacturing phase is represented on Figure 4-3 by the symbol ⑦ . This phase culminates in first hardware delivery, a major milestone in the component development program.

4.2.3 Control Component Development Test

Referencing back to Figure 1-5 (Generic PDP System Vee Model), we are now ready to begin the journey up the right-hand side of the Vee (system verification, validation, production, and field service). The Ford System Vee Product Development Process model emphasizes system hierarchy in the product design (left-hand side) and product validation (right-hand side) processes. The design process utilizes sequential decomposition from system to subsystem to detail parts while system validation proceeds in a mirrored fashion beginning at the detail part level, progressing upward through subsystems, ending with validation at the vehicle level. The

PW/HS validation process works in a very similar manner beginning with design verification testing at the component level, followed by subsystem (control system hardware and software) integration testing on the Closed-Loop Bench, leading up to full-up engine test. Depending on the complexity of the component, design verification testing can take anywhere from a few days for very simple parts with only a few functional requirements, to months for complex components such as fuel controls or electronic controls. Once certain key hardware and software components have completed design verification testing at the component level, they are assembled onto a system bench where they are tested in an integrated environment with a "closed-loop" computerized engine model. This is signified by symbol ⑧ on Figure 4-3. This is a key activity in the functional verification process of the control system and serves to significantly reduce the risk to FETT by verifying proper hardware/software integration and adequate software maturity in a low-risk environment (i.e. software errors can be uncovered, evaluated, and corrected without putting millions of dollars worth of experimental engine hardware at risk). Since the engine is simulated, however, there is little opportunity to resolve many areas of uncertainty associated with engine-driven requirements such as actuation loads and surge margin. These types of uncertainty can only be resolved by running an actual engine.

Finally, 18-24 months and millions of dollars after program launch, the first heavily instrumented engine is assembled, installed into a development test stand, and started for the first time. Initial testing focuses on vibration surveys, aeromechanical stress of the turbomachinery, and component (i.e. major engine module) performance verification. Once confidence is built with the steady-state operation of the machine, throttle transients are gradually made faster until snap transients (throttle movements in less than one second) are achieved. This development process normally begins at or near sea-level conditions in an open-air test stand and is often

quickly followed by a second test engine at simulated altitude conditions in an altitude test facility. One of the most prominent is the Arnold Engineering Development Center (AEDC) located near Tullahoma, Tennessee, which is owned by the US Government and operated by a private contractor. All major jet engine manufacturers send engines to this facility to undergo development testing prior to becoming flight qualified. This activity is represented on Figure 4-3 as symbol ⑨ .

A typical engine development program will allow at least one major hardware iteration between FETT and the configuration freeze for Initial Flight Release (IFR). IFR is considered a major engine milestone and is the anchor from which the majority of the engine program up to that point is planned. Assuming a reasonably successful ground development program occurs, flight qualification hardware is manufactured to the final configuration resulting from the ground development program. Control system hardware, being mostly mounted on the outside of the engine, normally enjoys a much more benign environment in an open-air test stand than enclosed in a minimally ventilated engine bay in an aircraft. Therefore, the majority of control system components must undergo a series of flight qualification tests in a simulated installed environment. These are represented on Figure 4-3 as the symbol ⑩ and include simulations of the thermal environment (hot and cold ambient air and fuel), vibratory environment, EMI/lightning environment, and contaminated fuel environment. Due to these environmental extremes, this battery of testing is much more stringent than the environment encountered on ground development engines. However, due to the complex nature of engine functionality and the complex interactions between components, functional qualification must be performed at the engine level, rather than by attempting to simulate the functional requirements in a component or

subsystem environment. These tests take the form of an engine endurance test and an altitude qualification test.

4.3 Quantification of Risk

In order to attempt to quantify risk, we must first understand its definition as used in this thesis[21]:

$$Risk = (\text{Probability of Failure}) \bullet (\text{Consequence of Failure})$$

Risk has many different forms that sometimes call for different methods of assessment and mitigation. Some of the different areas of risk associated with aerospace development programs are financial risk due to tight development budgets, technical risk due to design uncertainty and the desire to insert technological innovation, and schedule risk due to aggressive development program schedules. They all have two things in common; 1) managers must constantly assess their programs for the presence of risk and, 2) once risky areas are identified, it is likely that they will adversely affect the development program if left unmitigated. Unfortunately, aerospace development programs are fraught with all three of these types of risks due to the high cost of product development, the need to constantly improve capability and reduce cost of an extremely complex product, and pressure to be first to market for new and improved products.

This work focuses on risk induced by design and requirement uncertainty. As with nearly any complex system, the design uncertainty in a jet engine program can be significant in the early stages. As discussed previously, the outcome of the design of the major engine modules (turbomachinery, exhaust system, control laws, etc.) provides the design requirements

---

[21] Browning, Tyson R. "Modeling and Analyzing Cost, Schedule, and Performance in Complex System Product Development." PhD Thesis, Massachusetts Institute of Technology, 1999.

for the control system. Thus, uncertainty in the basic engine design translates directly into uncertainty in control system requirements. In the current product development process, early design uncertainty is generally assessed and mitigated "locally" within each module center. For example, the combustor group might devise an innovative method to improve combustor stability and lean blowout margin based on complex three-dimensional computational fluid dynamics (CFD) modeling. Until the model is validated, a substantial level of risk might exist due to uncertainty in the analysis. The uncertainty could be resolved by running a full engine test, but the consequence of failure would be severe from a schedule standpoint since the lead-time of a redesigned combustor could be several months. To mitigate this risk, a combustor rig that simulates the combustor interfaces and gas path conditions is fabricated and assembled. It provides the opportunity to heavily instrument different areas of the combustor that might be very difficult or impossible to instrument in a full engine in order to gain a better understanding of the aerodynamic phenomenon within the combustor. This data will be used to "calibrate" the model to reduce the design uncertainty. Parametrics with different hardware configurations can be assessed to better optimize the design prior to full engine test.

This methodology is followed in some form by every module center to mitigate risks on a "local" level. When interface uncertainty arises, however, the more common approach used to mitigate risk is through design conservatism. For example, when the compressor group designs the variable inlet stators and actuation linkage, it must define the associated aerodynamic and friction loads. Since uncertainty in the load has little impact on hardware they design, the risk is mitigated by flowing down a conservative load requirement (usually by adding the uncertainty to the nominal prediction, then adding an additional margin of safety). As previously discussed in section 4.2.2, the actuation system is designed to meet this load capability with a worst case

system, meaning that actuation design uncertainty is also taken into account. This stack-up of conservatism results in an extremely robust actuation system at the expense of cost, weight, and potentially fuel system heat load if higher fuel pressure is utilized to provide the actuation capability. Resolution of interface uncertainties does not normally occur until a full engine test is performed.

By the planning of success-oriented engine development programs that assume minimal design iterations, the PW (and therefore HS) culture has fostered this conservatism by punishing technical shortfalls. Risk taking is only rewarded when successes are achieved.

Within the past decade or so, increased focus has been placed on making risk mitigation a normal part of development programs rather than as the result of the occurrence of an unforeseen program issue. Tools have been developed and deployed across airframers, propulsion system contractors, and throughout the supply base. A risk management tool used by Pratt & Whitney aids in the assessment of risk by assessing the two factors which comprise risk, probability of occurrence and severity of consequence. The probability of occurrence is estimated by comparing the particular issue to a list of criteria based on the type of risk such as supplier maturity, requirement uncertainty, design maturity, etc. This is assessed as a high, moderate, or low probability. The consequence is then assessed in three separate areas – performance, development cost, and program schedule with choices of significant, moderate, or minor. A mathematical formula combines the four scores together to produce a single value between 0.1 and 0.9. The weighting factors in the formula can be adjusted to put more emphasis on a particular attribute (such as schedule) if deemed necessary by the program. Any particular item that exceeds a predetermined risk threshold (0.6 for example) is tracked by the program risk management team and requires regular statusing. Those below the threshold are tracked and

managed internally within the particular module center. Once the risk level is quantified, mitigation steps with specific exit criteria are defined which serve to mitigate the risk in stepwise fashion. Examples of these steps are development of an analytical model and/or performing a rig test. Along with each exit criteria, a workaround plan is identified in the event the exit criteria cannot be met. For example, if a risk mitigation step was to evaluate the results of an analytical model and the exit criteria was that the results met requirements, a workaround plan (in the event the results did not meet requirements) might be to launch a component redesign.

Considering the previous example of the compressor variable vane actuation load, the risk item in this case would be excessive cost and weight of actuation system due to high uncertainty in the actuation requirements. The probability would be a function of the uncertainty in the prediction (probably either high or moderate) and the criticality would represent the cost and schedule associated with a redesign of the actuation system. Given that a significant number of control system requirements would fall into the category of needing risk mitigation plans, the engineering labor cost associated with creating and maintaining these plans would be significant. In most situations like this, the only defined mitigation is to obtain engine data to resolve requirement uncertainty and launch a component redesign if the development cost and schedule impact is justified by the weight and product cost savings.

4.4 Reasons for Non-Optimum Designs

There are many reasons for non-optimum control hardware designs, but two prevalent ones will be discussed.

1. Late Definition of Requirements. Recalling the discussion from section 1.2, control system hardware has nearly the same lead time as other engine hardware, yet must be delivered earlier for subsystem integration testing (Closed-Loop Bench) prior to FETT. This results in the need for requirements **while** the rest of the engine is still being designed, with the

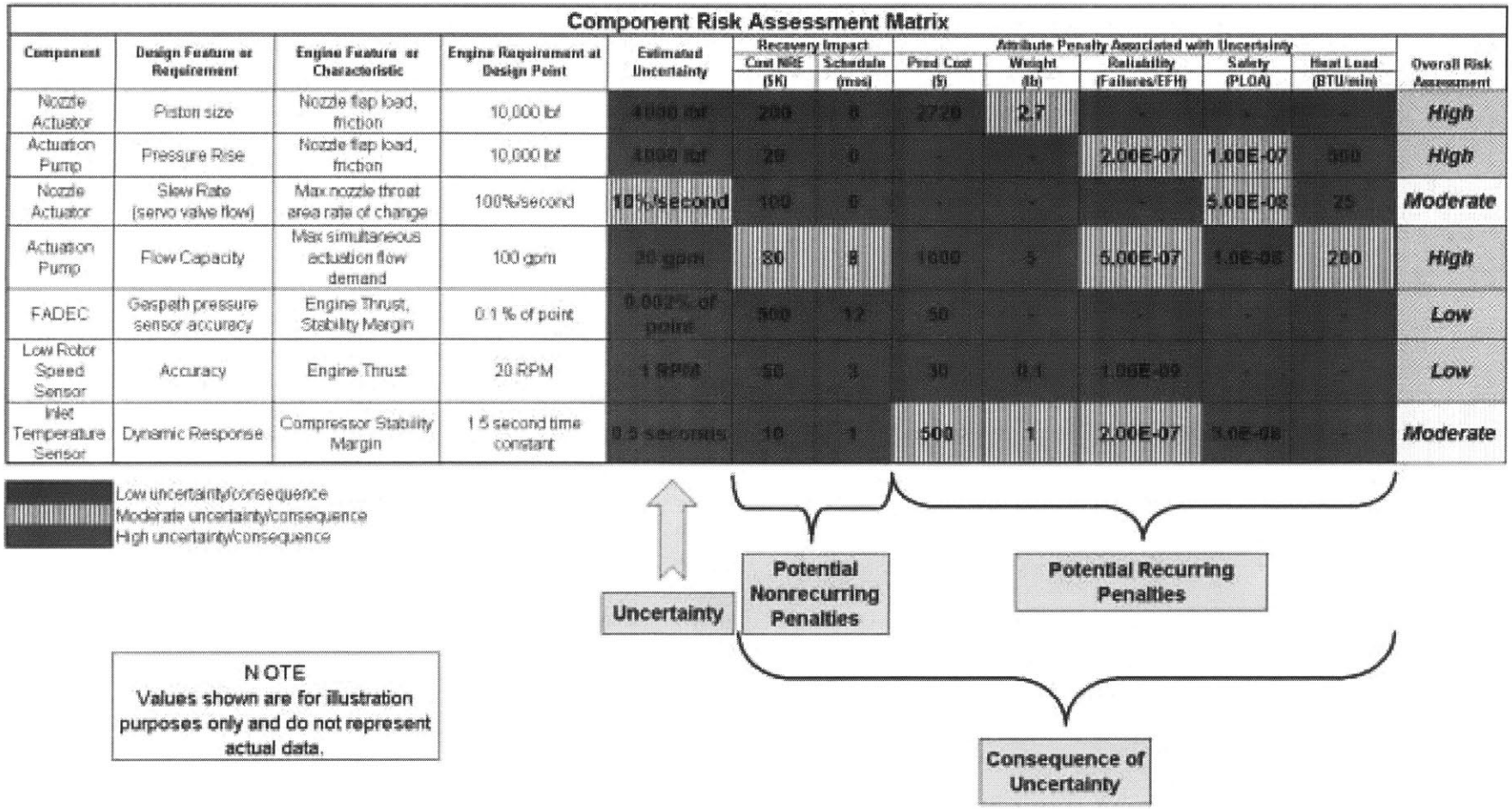


Figure 5-1. Component Risk Assessment Matrix

thermal environment for control system components is heavily influenced by the design of the engine bay ventilation system, which lags significantly behind. Again, somewhat conservative design assumptions must be made to allow control system components to meet the propulsion system development schedule.

4.5 Chapter Summary

This chapter discussed the current gas turbine engine control system development process as applied to Pratt & Whitney programs and provides a baseline against which to compare the revised framework. The current control system development process tends to be schedule-constrained based on manufacturing lead-times and hardware delivery requirements rather than on the availability of solid design requirements and design lead-time. The definition of risk as used in this work was provided and the current Pratt & Whitney risk mitigation process was described. The tools used for risk mitigation vary from company to company, but the basic approach to mitigate risk is consistent across the aerospace industry. The reasons for non-optimum control system component designs boil down to two basic issues: late definition of requirements and late definition of interfaces. Late control system requirements stems from the way the engine is designed (from the inside out). The turbomachinery, exhaust system and engine control laws must first have a reasonably mature design in order to generate control system requirements, however design and manufacturing lead times and the need to perform hardware/software integration tests on the closed-loop bench prior to engine test drive the control system component designs to be launched with significant requirement uncertainty. Additional schedule pressure results from the fact that many control system requirements are driven by interface and environmental requirements defined by the aircraft, whose development program typically lags behind the propulsion system by at least one year.

5 FRAMEWORK FOR IMPROVED DEVELOPMENT PROCESS

To produce more optimum designs, decisions on certain key requirements must be delayed until uncertainty is resolved. To allow the engine development program to proceed, thus acquiring key data to adequately resolve this uncertainty, functionally capable hardware (Generic Test Apparatus or GTA) must be provided. This chapter discusses GTA development considerations, associated business aspects, a system optimization methodology, and finally the risks and benefits of implementing this framework.

5.1 Determining Applicability of Framework to Components

The decision to delay design decisions for various components should not be taken lightly. Delaying detailed design and initial hardware delivery to early development engines carries some inherent risk since valuable development engine experience is forgone in the process. It is therefore just as important to understand *when* to apply the alternate development process as it is to utilize the process itself. For example, if the consequence of producing a non-optimum component is high, (i.e. tighter specs than necessary driving up product cost or weight or looser specs than required driving technical risk into meeting performance requirements) but the probability that this will occur is low (possibly because the component requirements are known with high confidence), the resulting *risk* level may not warrant delaying decisions. Likewise, if the consequence is low but the probability is high, this still might not warrant decision delays.

Referencing the definition of risk from section 4.3, the probability of failure can be thought of in terms of the degree of uncertainty of the requirements that directly affect component design. The greater the uncertainty band on the requirements that drive major

component design features (e.g. actuation loads drive piston size and/or pump size, heat load generation drives heat exchanger sizing, etc.), the higher the probability that a significant change in requirements will occur once actual engine data has been obtained. Hence, the probability of failure can be thought of as requirement *uncertainty.*

The consequence of failure can be thought of as the cost and/or schedule risk associated with recovering from such a change in requirements *or* the penalty incurred (in terms of product cost, weight, reliability, safety, etc.) by accepting an overly conservative requirement. Requirements can change in either direction, but due to the conservatism baked into the initial requirements to cover the uncertainty, these derived requirements will often be relaxed as uncertainty is resolved. In this respect, consequence of failure can be thought of as *consequence of uncertainty.*

Substituting our revised terms into the previous definition of Risk yields

$$Risk = (\text{Uncertainty}) \bullet (\text{Consequence of Uncertainty})$$

5.1.1 Component Risk Assessment Matrix

Figure 5-1 presents a Component Risk Assessment Matrix that can be used to aid in the quantification of risk as previously discussed. The data used to populate the matrix are fictitious, but is intended to be representative of real-world hardware. A "high-moderate-low" coding of certain attributes has been done based on somewhat arbitrary risk thresholds. During the initial engine design phase, the uncertainty associated with engine-level or module-level design parameters that become control system requirements (actuation loads, burned fuel flow rates, etc.) must be estimated. The ease and accuracy of this estimate will vary widely depending on the particular parameter being assessed. Thomke suggests that organizations play an important

role in the early stages of a product development process[22]. This suggestion can be directly applied to the process of estimating the impacts associated with requirement uncertainty. For requirements with uncertainty that carry significant potential component penalties (such as nozzle actuation load), establishment of multi-discipline teams is crucial to facilitating rapid iteration within the design space, allowing the assessment of various architectures. Once an architecture is selected, trade factors and sensitivities based on the uncertainty can be quickly generated. It is important that the team is allowed to function somewhat autonomously, even though its members may belong to several different module centers. The organizational incentive to produce an optimized product when module center boundaries are crossed (such as in the case of a nozzle or fan module owned by the respective module center, that contains actuation owned by HS) has been shown to be problematic based on PW/HS experience. This will be discussed further in section 6.

---

[22] Thomke, Stephan, Enlightened Experimentation – The New Imperative for Innovation, Harvard Business Review, 2001, 69.

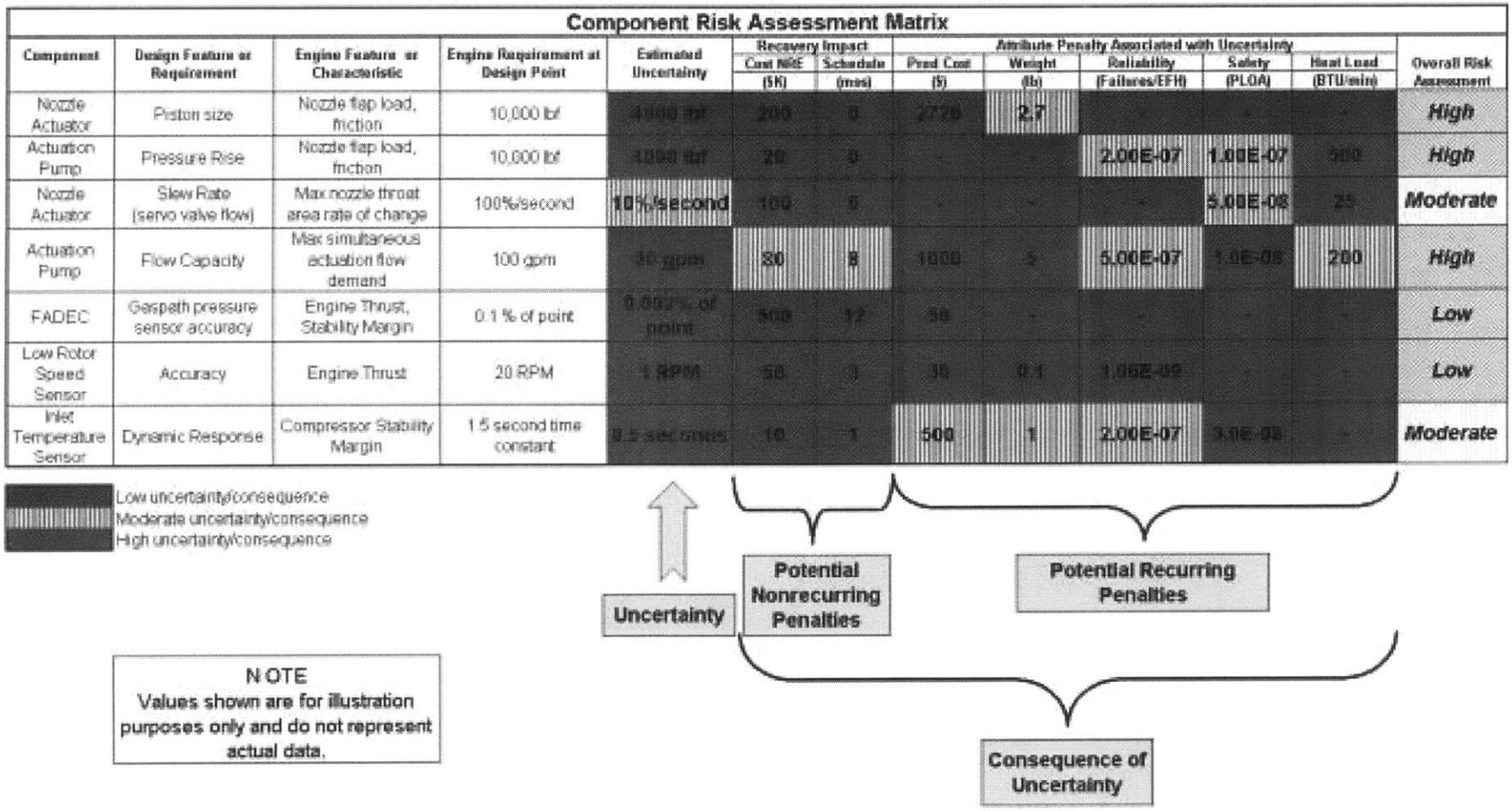


Figure 5-1. Component Risk Assessment Matrix

The consequence of uncertainty is shown as two types of attributes – potential non-recurring penalties and potential recurring penalties. The non-recurring penalties of development cost and schedule represent the "one-time" investment required (component redesign) to recover from the particular requirement uncertainty, assuming the component continued on a "normal" product development path (no decision delays). The recurring penalties represent some of the actual part attributes that would be impacted by the requirement uncertainty. These penalties (cost, weight, reliability, safety, etc.) are associated with the design of the component and as such affect the production hardware. This represents only a partial list of attributes, but an attempt should be made to assess all pertinent attributes. In some instances, quantified impacts are nearly impossible to generate without having a complete design definition, in which case qualitative impacts should be estimated (e.g. small increase, moderate decrease, etc.).

In the rightmost column, an Overall Risk Assessment is shown for each requirement being assessed. Following is a brief discussion on the rationale behind the determination of the Overall Risk Assessment.

- The first two rows assess the impact of nozzle actuation load on the nozzle actuators and fuel pump. The nozzle actuators are assessed as an overall "High Risk" due to the combination of high uncertainty (4,000 lbf uncertainty out of a 10,000 lbf design requirement) and high consequence of uncertainty (high nonrecurring cost and schedule, moderate weight impact, and high recurring cost impact).
- Similarly, the fuel pump was assessed as an overall "High Risk" due to the same high uncertainty and high consequence driven primarily by the high heat load resulting from increased pressure rise.

- The third row in the matrix covers a different design feature of the nozzle actuators – the velocity or slew rate requirement, which translates to nozzle throat area rate of change. While the uncertainty is assessed as moderate and the non-recurring impact as high (due to servo valve redesign costs), the remainder of the recurring impacts are minor. In this case, the overall assessment is given a "Moderate" rating.
- The fourth row represents the fuel pump flow capacity which might be sized either by an engine starting condition at minimum cranking speed or by the maximum simultaneous actuation flow demand (multiple actuators slewing at the same time). The requirement uncertainty was considered high (20 gallons per minute (gpm) out of a nominal 100 gpm requirement). Increasing the flow capacity is done by growing the pumping element (gear length for a gear-type pump, vane length for a vane pump, impeller depth for a centrifugal pump, etc.) which translates to a 5 lb weight increase and a $1000 product cost increase, both considered a high impact. This combination results in an overall "High" rating.
- The fifth row in the matrix includes a FADEC gas path pressure sensor (inlet, burner, or augmentor duct pressure) that is used for basic engine thrust and stability scheduling. The uncertainty of this requirement is shown as a very small value since pressure sensor error can be translated into stall margin reduction and thrust scheduling error with reasonably good accuracy. Even though the non-recurring impact of redesigning the sensors is high, the overall risk level is given a "Low" rating since the uncertainty is low and all other recurring impacts are low.
- The sixth row represents the low rotor, or N1 speed sensor. The low rotor speed sensing accuracy requirement is primarily driven by the accuracy of scheduling thrust. This requirement can be analyzed quite accurately using a steady-state engine simulation,

knowing the fan speed-flow characteristics. Therefore, the uncertainty associated with the low rotor speed sensor accuracy requirement might be very low at an early phase of engine design. Given this situation, the low rotor speed sensor would be considered a "low risk" component and would not likely benefit from delaying detail design.

- Finally, the last row includes the dynamic response requirement for the inlet air temperature sensor, or T2 sensor. The time constant uncertainty is considered high with a moderate impact on product cost, weight, and reliability because of this uncertainty. This results in an overall "Moderate" rating. Other factors such as the criticality of the function might be considered to assist in making a decision to apply the revised framework.

5.1.2 Single Requirement Affecting Multiple Components

In certain circumstances, a single requirement can potentially affect more than component. An example of this is nozzle actuation load, which consists of the two basic elements of flap aerodynamic load and friction, is historically difficult to accurately predict. The uncertainty of the friction piece is primarily a function of historical data, relying on the performance of previous designs. Friction of the linkage and flap train can be analytically estimated based on established engineering techniques, but will change over time with engine usage as working clearances change, anti-gallants wear away, and surface corrosion occurs. Historical factors are generally applied to new part predictions to estimate worst-case friction values, but this prediction can contain substantial error. Prior to the availability of reasonably priced, high-powered computing, flap aerodynamic load was estimated in a similar manner – by empirical comparisons to historical data. Within the past decade, complex Computational Fluid Dynamics (CFD) codes have been developed which can run on economical but extremely powerful workstations producing results in a fraction of the time previously required on

expensive mainframe computers. This has led to substantial reductions in uncertainty of aerodynamically driven actuation load. However, significant uncertainty remains due to the immaturity of the CFD code and due to engine scheduling uncertainty. Recalling the discussion from section 2.1 on Newton's 2nd Law, thrust is the product of mass and acceleration. Gas acceleration is heavily influenced by engine pressure ratio (EPR), which is controlled by exhaust nozzle area. Engine airflow and EPR are scheduled to satisfy the engine thrust specification while minimizing the impact on several other engine parameters such as rotor strain range and compression system surge margin. Engine scheduling is part of the engine's basic control laws that are implemented in the engine control software. As such, the software will continue to iterate throughout the engine design and ground test phases and will only be frozen during flight certification. Exhaust nozzle load uncertainty associated with engine scheduling might be substantial and difficult to estimate with any degree of accuracy.

A complication factor associated with actuation loads in general is the fact that the actuation system design consists of two basic elements: actuator piston size and pump pressure. Both attributes represent "knobs" that can be turned during the system design process to optimize the system based on known constraints. The optimization process is discussed in greater detail in section 5.3. For the example of nozzle actuation load uncertainty, each component affecting actuator load capability should be assessed individually while holding the other components' attributes constant. This will allow determination of risk for all components affected by the uncertainty of this requirement. It is possible that some components will qualify as "high risk" while others do not. For example, observing the first two rows in Figure 5-1, actuator weight and product cost associated with nozzle actuation load uncertainty could be substantially more than the weight and product cost of the fuel pump, however the heat load associated with the

pump (due to the higher pressure required) could also be significant. The impacts of these attributes must be assessed in relation to the status of the control system holistically. In other words, if system heat generation margin exists, the additional heat load associated with the fuel pump might result in a "low risk" assessment. Conversely, if the control system weight estimate exceeds the allocated target, the nozzle actuator might be assessed as "high risk". It is possible that both will be assessed as "high risk". In any case, the decision to apply the revised framework and delay design decisions will be based on risk thresholds determined by program management.

The process described here attempts to define a quantitative approach to assessing risk. The process, in fact, utilizes quantified trade factors (i.e. sensitivities) for various product attributes to aid in the overall risk assessment. However, the actual process of combining the attributes to generate an overall risk assessment is qualitative in nature. Improvement of this overall process is left for future research.

5.1.3 Determination of Requirements to be Assessed

Since each control system component might have hundreds of requirements levied on them, only the "significant" requirements responsible for driving the important design attributes (product cost, weight, reliability, envelope, etc.) need undergo this uncertainty assessment. In general, these will be the primary functional component requirements such as stroke and slew rate for actuators, flow and pressure rise for pumps, and accuracy for sensors. These requirements should be capable of being adequately resolved within a timeframe that meets program expectations. For example, if program management expects the Initial Flight Release (IFR) hardware configuration to be representative of the production configuration, all requirement uncertainty resolution and design iteration must occur prior to flight certification.

Therefore, requirements with uncertainty that cannot be resolved until later in the program (such as those associated with the aircraft environment) would not be assessed for application of the revised development process. Similarly, requirement uncertainty must be *capable* of being adequately resolved through the acquisition of engine data. In other words, if parameter margins cannot be ascertained through experimental measurements during the engine test process, there would be no way to use this data to optimize a component's design. Thomke makes this point when discussing the basics of experimentation[23]. For example, actuation loads are typically measured by recording the differential pressure across the actuator piston. Knowing the piston area allows simple calculation of the load. Load data at various points throughout the engine operating envelope (at sea level and at simulated altitude conditions) can be used to adjust models and resolve prediction uncertainty. Even system friction can be measured reasonably well using this method by slowly moving the actuator throughout its range of motion at a "no load" condition. For a variable exhaust nozzle, this can generally be done at ground idle conditions since gas path pressures, which drive aero loads, are sufficiently low. For variable compressor and fan vanes, this is usually performed with the engine off using a hydraulic cart for actuator muscle due primarily to aeromechanical stress and/or operability concerns resulting from running with the vanes so far off schedule. However, exact determination of a requirement such as ignition system spark rate or energy can be much more difficult since the result tends to be more discrete than continuous. In other words, the result of having an adequate spark rate and energy level is to have a successful combustor light. Determination of the *minimum* spark rate and energy requirement would be an extremely costly and tedious process since the amount of

---

[23] Thomke, Stephan, Enlightened Experimentation – The New Imperative for Innovation, Harvard Business Review, 2001, 69.

lightoff *margin* at a given condition is extremely difficult to quantify without performing a multitude of tests.

5.2 Generic Test Apparatus

Generic Test Apparatus (GTA) refers to surrogate hardware that is developed to substitute for "high risk" components (as defined in section 5.1.1) during initial ground engine test. This section discusses the design and implementation considerations that should be made as the GTA hardware is developed.

5.2.1 Architectural Considerations

In order to make a strong business case, minimize risk to initial ground test engine development, and enable acquisition of key control system design requirement data, some deep thought should be given to the architectural aspects of Generic Test Apparatus (GTA) hardware. As a minimum, the component architecture should meet the following needs:

- Minimize setup time. Since hard design decisions are being delayed, components to which the framework is being applied should have a relatively short lead-time and should be reasonably simple to modify.
- Maximize durability. In order to make a strong business case, hardware should be capable of usage on multiple programs with varying duty cycles. The design should consider extreme robustness for major structural members such as pressure vessels and replaceable details for wear-susceptible items such as actuator uni-balls.
- Maximize functional robustness. To ensure an unencumbered ground test engine development program, system functional capabilities should cover nominal requirements *plus* requirement uncertainty.
- Minimize physical and functional interface differences with "final" design. The goal would be for the final optimized design of a component to be a "drop-in" replacement for its GTA predecessor.

- Maximize flexibility. The business case is substantially strengthened by allowing detail subassemblies to be "mixed and matched" using common interfaces. This provides the flexibility to fully explore the design space to ascertain the optimum solution.

In order to maximize utilization across multiple programs, a Risk Assessment Matrix for the most recent military and commercial jet engine development programs should be compiled based on design requirement uncertainties that were present early in the programs. Functional features assessed as high or moderate risk should be compiled for both programs and compared for common aspects. GTA architecture should be defined to *match flexibility to design feature uncertainty.* For example, in the case of actuators, two design features than may carry significant requirement uncertainty (reference Figure 5-1) are piston size and slew rate. An architecture that would provide a family of solutions might look like the hardware shown in Figure 5-2. The figure shows a series of housing/piston sizes that can address a range of load uncertainty and a series of Electrohydraulic Servo Valve (EHSV) flow ratings (the L/M/H labels denote low, medium, or high flow rates) that can address a range of slew rates. Common EHSV interfaces or "footprints" on the actuators allows flexibility to mix and match the two design features to satisfy a reasonably large design space.

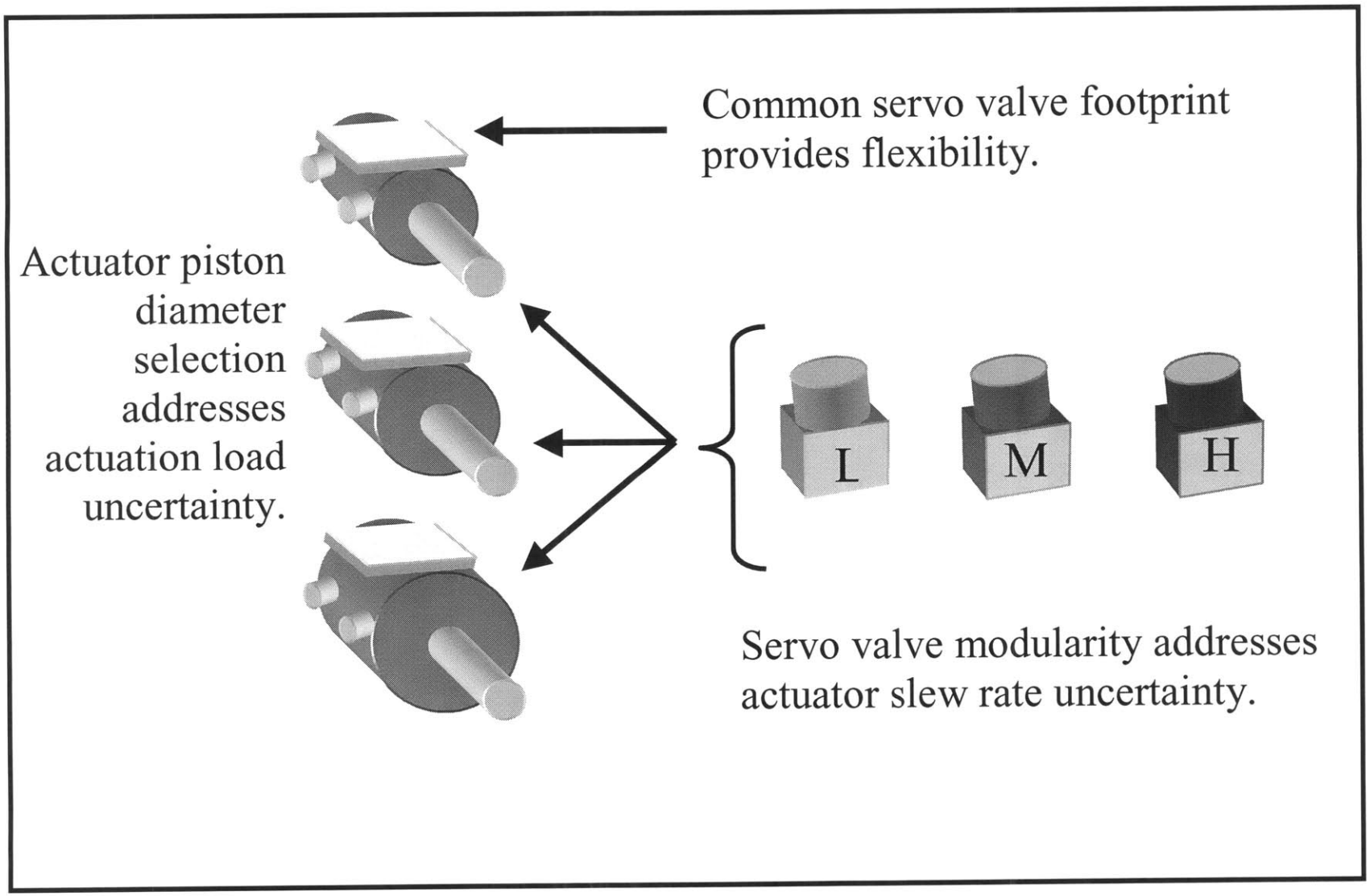


Figure 5-2. Potential GTA Actuator Architecture

With this in mind, some common control system hardware with potential GTA features is listed for consideration:

- Actuators:
  - Actuator piston/housing diameter (reference Figure 5-2). Since the intent is to provide flexibility for resolving load-carrying uncertainty, piston diameters and hydraulic system pressure should be evaluated concurrently to determine the best method. Growing piston diameters imposes additional mass/HCF stress into the mounts as well as requires more space on the engine while increasing pressure carries pumping system durability limitations.

  - Piston stroke. Various piston strokes might be required to meet tight installation situations. Typical vane actuator strokes are on the order of 2 inches with typical nozzle actuators on the order of 6 inches. Replaceable piston stops and feedback devices such as Linear Variable Displacement Transducers (LVDTs) will allow actuator stroke to be adjusted between selected values. In general, actuators are never scheduled against the hard stops to avoid the weight impact associated with the structural requirements. Therefore, usage of the proper stroke LVDT to achieve the correct positioning tolerances and capabilities is the prime consideration.

  - EHSV Flow Gain (reference Figure 5-2). Provide a range of EHSV flow ratings with standard footprints to allow interchangeability. To modify maximum actuator slew rate for a particular flow rating, the maximum EHSV flow can be reduced by modifying spool valve end plugs. Slew rate can also be limited in software by limiting EHSV electrical current. Failure conditions can best be evaluated by limiting spool valve travel (allowing hard-over failures to be simulated), whereas normal operation with reduced flow can be evaluated quite well by limiting EHSV current with software.

- Dynamic Response. Typically, minor loop dynamic response requirements can be defined with reasonably high confidence[24]. However, if benefits could be realized by reducing frequency response capability, EHSV first stage pressure could be reduced using a simple restrictor or regulator. This could also be achieved by reducing software gain.

- Pumps

  - Burn flow. In general, burn flow requirements can be accurately predicted using modern engine simulation techniques[25]. However, if burn flow pumping requirements are uncertain, the following options can be evaluated.

    - Provide an electric pump (not driven by the engine gearbox) that can be controlled in such a way that the output pressure would reasonably match the BOM pump. This would provide a highly flexible system since pump pressure could be changed dynamically external to the engine. The setup and up front investment might be significant.

    - Provide a range of mechanically driven flow/pressure combinations of impellers, possibly all installable in the same pump housing.

  - Actuation flow and pressure. The most straightforward method would be to provide an off-engine hydraulic cart with adjustable pressure level. This has been used in previous programs making it a low risk option.

- Fuel Metering. Provide as accurate a system as possible using software to simulate errors in fuel metering and perform performance/operability margin testing, allowing the effects of less costly systems to be evaluated.

- Anti-Ice Air Management (pneumatic systems). Provide a range of sizes of fully modulating, hydraulically controlled butterfly valves. Different valve inserts could possibly utilize the

---

[24] Johnson, Rodney, personal conversation, September 2002.
[25] Johnson, Rodney, personal conversation, September 2002.

same housing. Use modular EHSVs with standard footprint and adjustable/selectable flow gains as discussed in actuator section.

- Thermal Management System – Provide a fuel metering valve with adequate flow capability and low leakage capability. Flow capacity can be limited with software or by a hardware adjustable stop for failure mode testing.

5.2.2 Interface Considerations

Substituting a non-production hardware configuration into the early development program carries some inherent risk. One would have to question the benefit of the revised development framework if an unwanted surprise reared its ugly head while during the throes of flight clearance testing. Once the risk assessment described in the previous section identifies candidate components, careful consideration must be given to the potential risk incurred by non-Bill of Material (BOM) interfaces. This interface risk assessment must include ALL interfaces for the proposed GTA hardware – physical and functional. The goal of GTA hardware is to be a "drop-in" replacement for the production hardware for which it is a surrogate.

GTA hardware can be divided into two major categories – engine-mounted and non-engine-mounted. For the engine-mounted category, a component that does not provide physical interchangeability not only drives risk, but also results in the additional program cost of modifying the interfacing hardware to allow its use. Therefore, physical interfaces for engine-mounted GTA hardware should be defined on the same schedule as non-GTA hardware (as if the production version was to be used on FETT) if possible. This may serve to limit the design space, but this must be traded with the program cost and risk of iterating the interfacing parts once final design requirements are defined. For example, if a GTA nozzle actuator is needed due to nozzle load uncertainty, the design team should strive to define the design of the physical attachments for the production configuration and implement them on the GTA hardware. Design

flexibility for the GTA actuator rod end can be achieved by providing a threaded, replaceable rod end. This would allow the use of the "production configuration" rod end on the GTA hardware to gain confidence during initial engine development. For the non-engine-mounted category, such as fuel pumps, decisions regarding definition of the physical interfaces should be delayed along with the decisions for the component design features in question unless this delay results in taking on excessive risk to the interfacing hardware. For example, if the decision on actuation pumping capacity needs to be delayed due to uncertainty in simultaneous actuation slew rates, the GTA equivalent for an actuation pumping function might be an off-engine hydraulic cart. This would mean that design decisions regarding the gearbox interface might also need to be delayed, driving risk into the gearbox. The risk of delaying decisions on this interface must be weighed against the detriment of defining the interface and limiting the pump design space (such as selecting a nominal pad speed to allow definition of the gear train ratios). Due to the complexity of the gearbox, it is likely that delaying decisions and finalizing the pump interface just before flight clearance will be deemed too risky, and as such, interface decisions will be made early. In the case of the interfacing plumbing, the relative complexity is low allowing decisions to be delayed with much lower risk.

Functional interfaces such as electrical and electronic should also be scrutinized to the same extent as physical interfaces. For effectors such as actuators and fuel valves, the electrical characteristics of the Electro-Mechanical Interface Devices (EMIDS) should be as close as possible to the intended production configuration. This is becoming progressively easier to do in practice since, over the past decade, a concerted effort has been made to standardize electrical input/output (I/O) architecture in the interest of reducing the cost of electronic controls. This has resulted in the definition of design norms that result in functional I/O consistency across all

PW/HS programs with the intent of expanding across all HS product lines. Rather than selecting a particular EMID supplier and designing all EMID physical interfaces around that supplier's parts (hence reducing cost benefits through competition), design flexibility can be maintained by providing adequate room in the GTA hardware to accommodate different EMID suppliers' hardware. The EMID supplier could provide initial development units for GTA hardware that are electrically equivalent to the "production" configuration, but are mechanically packaged to interface with the GTA hardware. This would result in a small development program NRE cost, but would pay off in the long run through reduced interface risk.

5.2.3 Functional Considerations

At this stage, GTA components and their physical and functional interface configurations have been defined. One more decision that is significant needs to be made before the implementation phase and this is the functional performance of the GTA hardware. Considering that the primary reason for delaying design decisions in the first place was to deal with functional requirement uncertainty, deciding on the functional capabilities for the GTA hardware is a very important step. The easy answer would be to provide functional performance at levels that cover the nominal prediction plus uncertainty. In the case of a nozzle actuator with a nominal maximum load prediction of 10,000 pounds and an uncertainty of 4,000 pounds, a GTA actuator with a 14,000-pound capability and the features needed to actually measure the load reacted (such as extend and retract pressure taps) would be provided. This additional load-carrying capability adds no additional risk to the development program since the closed-loop functionality of the effector loop will result in scheduling only enough pressure across the power piston in the actuator to exactly react the imparted load from the flap train. In the case of actuator slew rate, the EHSV flow rating needs to cover the requirement uncertainty in the same manner as actuator

piston size (or pump pressure) covers load uncertainty. However, having an EHSV that is potentially larger than the production version induces some additional risk to the development program since failure mode testing (simulating loss of effector control) would be affected by a faster actuator rate during the simulated failure. This could result in more severe consequences to the engine, such as a higher fan overspeed in the event of the nozzle failing in the opening direction due to reduced backpressure in the gas path. This situation could be mitigated by providing the capability to easily change the maximum slew rate through the limiting of EHSV electrical signal current by software or through replaceable hardware stops for the EHSV secondary spool valve. Implementing software limiting would result in a small NRE cost to provide the capability, but would be much easier and quicker to modify during the course of the development program (it could potentially be changed during an engine run without even shutting the engine down). In any case, the delivered GTA capability for each function requires careful consideration to determine if excess capability does or does not induce risk into the engine development program. For those situations where excess capability drives additional risk, the capability to "throttle back" to lower levels of capability should be provided by a method that is easily and quickly changed on the engine test stand.

5.2.4 Technology Maturity Considerations

Another important consideration for components that would appear to benefit from the GTA development concept is the assessment of technology maturity. Delaying the detailed design of a component with immature technology would be extremely risky without a substantial laboratory/bench development program to mature the technology (reference section 2.1). The interactions between the component and the jet engine environment can be extremely complex, therefore the TRL as defined by the NASA/PW definition should be a 7 or higher, indicating that

the technology being used in the component has been successfully utilized previously in the target environment[26]. This gate would almost seem to be common sense, but is important enough to deserve a specific task in the process.

5.2.5 Implications for Product Validation

The proposal of utilizing non-BOM hardware during the initial engine ground test program would not be complete without an assessment of the risk to the engine and control system validation programs. The risk to engine-level validation is covered by assessing the interface risks and functional risks as described in the previous sections. This leaves the risk to control system validation.

Since the basic premise of the GTA development concept is to delay the detailed design phase of affected components, this results in a reduction in valuable development engine experience, inserting the "production" configuration just prior to flight certification testing. This leaves little if any time for learning about this hardware configuration. Given a mature technology, the inherent risk of reduced ground test engine experience associated with the GTA development concept is substantially mitigated by the normal development process that is followed by control system components. This process involves a substantial amount of bench testing at the component and system level as discussed in section 1.3.6. As discussed earlier, the two primary reasons for substantial bench development is to find and correct problems with the control system in a "safer" and less costly environment than on the engine, and because the extremes of environmental operation (ambient temperatures, worst case vibration at resonant conditions, etc.) will not be experienced until flight test or operational service. Issues associated

---

[26] Neimeyer, Jonathan K. "Framework for Risk Reduction in Gas Turbine Product Development." SM Thesis, Massachusetts Institute of Technology, 2002, 66.

with these extremes of operation must be surfaced and corrected early in the engine development program to ensure a successful flight clearance program. In general, nearly all control system component structural and environmental testing occurs in a bench environment rather than on the engine. The only engine-level validation activities where no attempt is made to qualify on the bench are high-energy surge events, engine ingestion tests (water, ice, bird) and blade-out tests (turbomachinery blades are intentionally liberated to ensure case containment and engine mount structural capabilities). In these cases, the test environments carry a high degree of uncertainty due to the complex nature of the events themselves. Since a good deal of the component validation activities can be accomplished in the bench environment, the GTA development concept lends itself very well to control system components since they can be tested thoroughly, economically, and in a concentrated manner once the final configuration is defined and produced. As usually occurs during flight clearance testing, several units undergo various tests such as burst for pressure vessels, vibration, fire protection, Low Cycle Fatigue (LCF), bench endurance, etc. simultaneously by utilizing different test facilities. Since components that participate in the GTA development concept will be delivered late in the development program relative to those that follow the "normal" process, significant attention needs to be paid to the bench validation process. Components developed to the new process must undergo intense and highly concentrated bench testing to quickly bring the component up to an appropriate maturity level.

For GTA-process components, a "right-to-left" development program anchored at the flight certification date required by the program should be planned. The development program should include tasks for engine data gathering, data analysis, component design, and manufacturing lead-time. A high-level program plan including these tasks is shown in Figure

5-3. The task bars shown in light blue represent tasks common to both the "normal" and GTA concept development processes. The task bars in orange represent activities unique to the GTA development concept. The yellow box at the bottom center of the figure represents a key constraint in the process. This time interval begins when engine data analysis is complete and final component requirements are defined. This information is rolled into the final design of the component that culminates in the component Critical Design Review (CDR). Manufacturing then begins in earnest resulting in first delivery of the production configuration (and optimized) component. Note that the schedule shows the detail design phase beginning before the final engine test data is available. This is done to define the component design features that are not dependant on acquisition of the engine data. This phasing depends on the design and manufacturing lead-time associated with the component under consideration, but due to the compressed schedule between availability of engine data and when flight clearance testing must begin, this task scheduling is recommended.

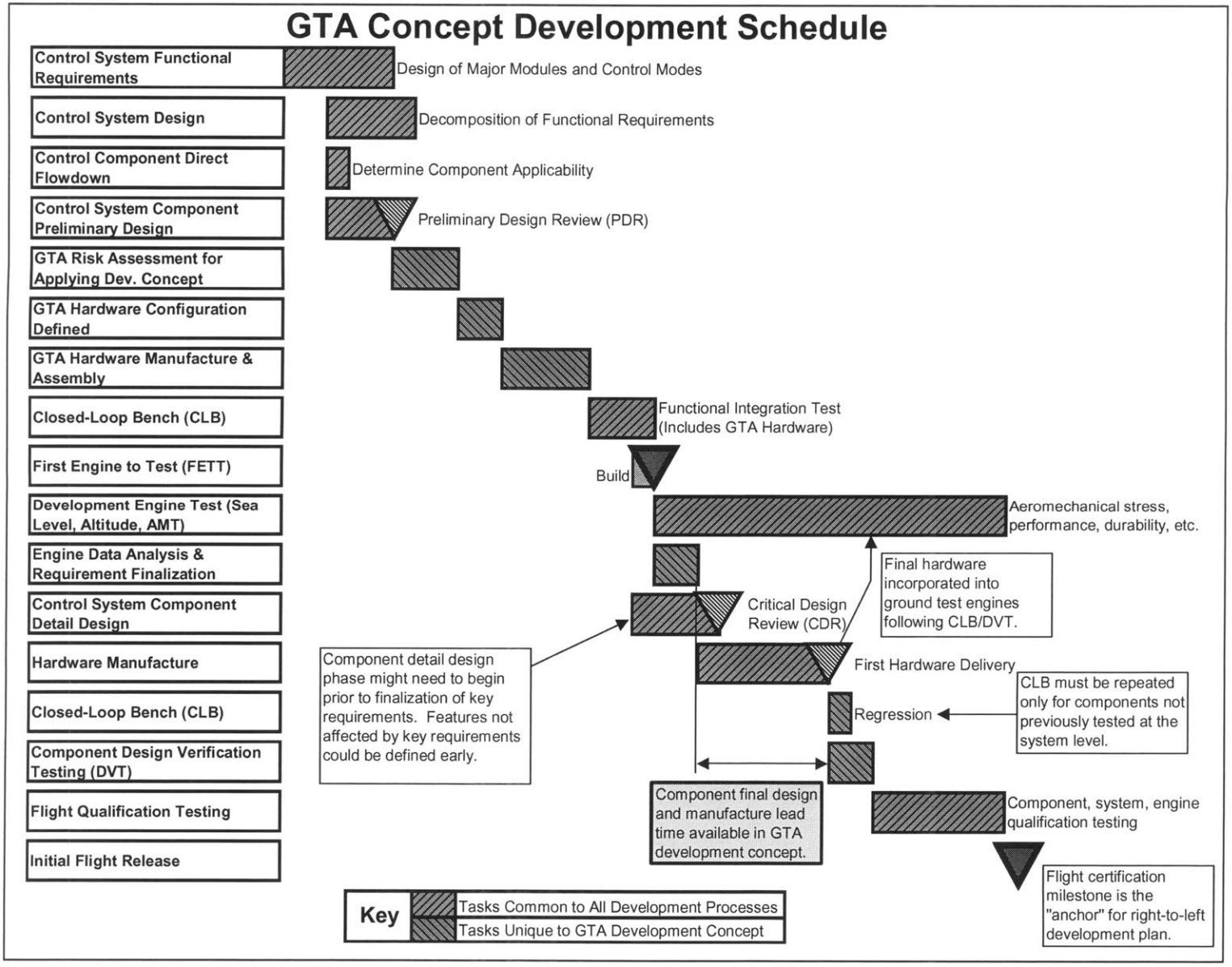


Figure 5-3. GTA Concept Development Schedule

5.2.6 Business Case of Implementation

The fact that this new development framework would result in more optimum component designs sounds good on the surface, but in order to obtain management buy-in to implement the process, a convincing business case is necessary. Although the cost to implement the process will vary from program to program based on the number of components involved, an estimate of the initial investment required to design and manufacture a representative set of GTA components was completed. This cost was then compared to the benefit realized based on two different scenarios:

1. Avoidance of the design iteration and subsequent requalification of the assumed components to produce an optimized design based on data learned from the initial development program.
2. Avoidance of a life cycle cost (LCC) impact on the customer due to the impact of living with a non-optimum design throughout the product life cycle.

The investment cost of developing a GTA hardware suite for an entire engine actuation system was estimated. An actuation system was used for the business case model due to recent development program experience (making actuation a likely candidate), and due to the fact that the actuation system lends itself well to this process since most of the examples sited in this work involved actuation functions. The assumed engine actuation system architecture included one "small" and one "medium" sized actuator, each with redundant EHSVs and LVDTs and a transfer solenoid in an active-standby configuration (reference section 2.2.2). It also assumed three "large" actuators in a slave-slave configuration, each with simplex LVDTs that are controlled by a single servo manifold having redundant EHSVs, a transfer solenoid, and transfer valve (reference Figure 2-6). "Small, medium, and large" EHSV and LVDT designs for the respective actuators were also assumed. Specification of a COTS off-engine hydraulic cart along

with the design of associated plumbing and electrical harnessing was assumed for the actuation pumping system. Engineering effort to integrate the system together was also included. Five sets of hardware (one for closed-loop bench, three for ground test engines, and one spare) were assumed.

The benefit for scenario number 1 listed above (design iteration avoidance) was estimated by calculating the cost of the redesign and requalification of three actuator part numbers, three EHSV part numbers, and one pump part number. Costs included were NRE (design), additional development units (two supplier units, three engine, one CLB, and one spare) and requalification test costs for flight certification.

The benefit for scenario number 2 listed above (LCC impact avoidance) was estimated by calculating the LCC impact due to excessive weight only. This is a conservative assumption realizing that reduced weight will likely also result in reduced cost and potentially reduced heat load. A one-pound reduction for the small actuator, two pounds for the medium actuator, three pounds for the large actuator, and five pounds for the pump were assumed. Assuming a life cycle cost weight trade factor of $10M/lb (a typical value for a modern jet fighter with a 30-year life cycle), benefits for an average year of operation and for the engine's life cycle were calculated.

The results of the business case analysis **confirmed** the hypothesis of the revised development framework. For scenario 1, the cost/benefit ratio for iterating only the actuators and EHSVs was calculated at 0.77, meaning that the cost of implementing GTA hardware for the actuators and EHSVs was only 77% of the benefit gained for avoiding a redesign/requalification of this hardware. Assuming the avoidance of a design turn on ALL GTA hardware (actuators, EHSVs, and pump), the cost/benefit ratio was 0.69. This indicates that, assuming a hardware

redesign would result, the cost of implementing the GTA development approach would more than pay for itself in a *single development program.* Assuming that the hardware can be re-used in future programs (which is the design intent) would further improve the business case. An important consideration for the funding of GTA hardware development is that for use on subsequent development programs, right-of-use of government-owned assets must be secured. Although this does not represent an insurmountable hurdle, the situation can be avoided by company funding of GTA hardware developed on government contracts.

For scenario 2, the cost/benefit ratio for the LCC impact of only the actuators and EHSVs *for a single average year of operational service* was calculated at 0.80, meaning that the cost of implementing GTA hardware for the actuators, EHSVs and pump was only 80% of the benefit gained for avoiding the LCC impact for one year of operational service due to the increased weight of the non-optimized actuators and EHSVs. Assuming the avoidance of a weight-driven LCC impact on ALL GTA hardware (actuators, EHSVs, and pump), the cost/benefit ratio was 0.66 for a single year of service. Over the 30-year lifespan of the fleet, the cost/benefit ratio is estimated at 0.02! From a life cycle cost standpoint, the benefit is quite substantial!

5.3 Optimization Modeling

Referring again to Figure 5-3, a critical task that is unique to the GTA development concept is labeled "Engine Data Analysis and Requirement Finalization" in which key engine data is acquired and analyzed in order to produce more optimum component designs. Some key attributes for aerospace fuel system components are functional performance, product cost, weight, envelope (outer contour), and heat generation. Each of these attributes has a desired direction in order to optimize the system (maximize performance, minimize cost, weight, envelope, and heat generation), but they are usually in conflict with each other. Furthermore, in

many situations, more than one parameter or design feature will affect a given requirement. An excellent example of this is elements of the actuation system such as the various actuator piston sizes and pump pressure. This particular example was discussed in section 5.1.2. In order to quickly trade piston size with pump pressure, an optimization model is an excellent tool that allows rapid, automated iteration of these parameters while observing system constraints such as a maximum pressure level due to pumping technology or a maximum piston size due to aircraft engine bay space (i.e. installation envelope) limitations. Another significant benefit of an optimization model is its ability to quickly optimize the system based on different goals (such as product cost or weight) or based on a combination of goals.

In a specific example, suppose a particular actuation effector must react a load of 10,000 pounds at a particular operational condition. This is done by providing a particular pressure differential across a piston. There are an infinite number of solutions to this problem, but the solution space shrinks as constraints are applied. For instance, Figure 5-4 presents a family of solutions with the constraints of 4000 psid maximum pressure and 3.2 inches maximum piston outer diameter. The valid solution space is shown by the green area. However, as key attributes are associated with the parameters being traded, such as cost, weight, and heat load with the pump differential pressure, and cost and weight with actuator piston size, the two can be traded against each other to find the optimum system design.

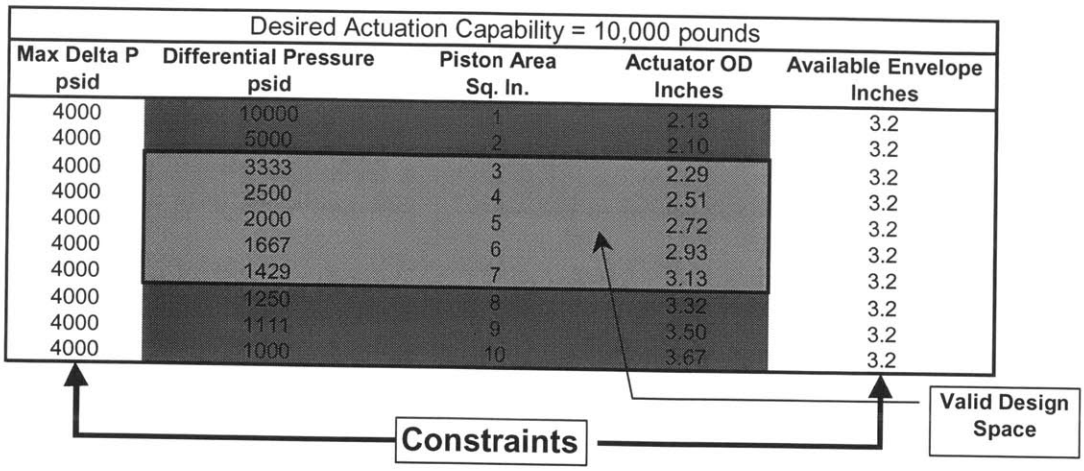


Figure 5-4. Example of Actuation Capability Design Space

Figure 5-5 shows an example of the "front end" of a simplified fuel system optimization model utilizing the Excel Solver tool that attempts to illustrate the concepts of modifying certain system parameters, applying constraints, and optimizing a set of output parameters based on established targets. The variables are the yellow-colored cells in the upper left of the figure, and are the parameters that Solver sequentially modifies as it attempts to reach the goals while satisfying the constraints. The variables selected are the piston size of three different actuator effectors, the impeller diameter and depth of a fuel pump, and the displacement of a positive displacement fuel pump. These typical parameters are changed during the fuel system design process as attempts are made to satisfy design constraints at minimum cost, weight, and heat load. To the right of the actuator variables are cells for the number of actuators. These are manual inputs to allow quick assessments of different numbers of actuators in the event that design constraints with the desired number of actuators cannot be met. These values could potentially be incremented automatically by linking the output of solver to these cells. The next four columns to the right are the outputs of the model (product cost, weight, and heat generation

at two different flight conditions), which are totaled for all components. The totals represent the goals that the model seeks to minimize (the blue cells). The goals can be individually analyzed or can be analyzed in an aggregate manner by calculating the variance from a given target (as shown in the figure), then summing the squares of the variances and minimizing the sum (shown in the red cell at the extreme lower right). Weighting factors have been applied to bias the optimization by priority of product cost (30% factor), then by weight (25% factor), then by heat load (22.5% for each condition). These can obviously be easily modified based on prevailing program conditions. The green cells just below the variables show the constraints that are applied to the calculated parameters. For each set of values of the variables, an entire model of the fuel system is evaluated at a large number of flight conditions that are defined by *design tables*. The fuel system model is not shown, but was included to validate the concept of optimization modeling. One issue associated with optimizing pumps and actuators concurrently stems from the basic relationship between them.

$$ActuationLoad = (Pump\Delta P - Losses) \bullet PistonArea$$

Since Pump ΔP and piston area are both variables that require optimization, the fact that a constraint (actuation load) that must be satisfied is the *product* of the variables leads to a non-linear situation. Although a bit more difficult, techniques for non-linear optimization exist which can still lead to successful application of this concept. One potential method would be to hold one of the variables such as pump impeller size fixed for a given run, store the results, generate another set of data with a different impeller size, and so on. Using the compiled results from the initial runs (actuator sizes are now optimized for each impeller size), this set of data can now be optimized for impeller size, in essence a two-step process.

The design tables represent performance requirements from the end-item customer (airframer and/or government). The customer specifies a specific maximum and minimum thrust levels to be achieved at various flight conditions, which is termed the flight envelope. The engine manufacturer (PW in this case) must design an engine concept and cycle that satisfies these requirements. An engine simulation of the powerplant cycle is created that results in predictions of burn flow requirements, exhaust nozzle area, variable vane position, rotor speeds, gas path pressures and temperatures, etc. A partial listing of this engine simulation deck output is shown in Figure 5-6. The number of flight conditions and throttle settings has been greatly reduced to fit within Solver's capabilities. This listing represents only a single temperature "day" condition known as "standard day", which is defined as 59 degrees F at sea level. "Cold day", "tropical day", and "hot day" conditions have also been defined based on worldwide environmental extremes. Ambient temperatures and pressures for each type of day at various altitude conditions have also been defined. In totality, the entire fuel system must be evaluated against literally thousands of individual conditions observing both internally and externally imposed constraints. This is far beyond the capability of Excel Solver; therefore, an "industrial strength" optimization program such as AMPL would be required.

A significant benefit of optimization modeling is the ability to have the model identify the design points (i.e. the conditions that determine the sizing of variables), then very quickly performing a sensitivity analysis at those conditions to determine the benefit of incrementally reducing requirements. For example, if the pump sizing condition is determined to be a low altitude, high Mach number condition, but at an idle (i.e. low thrust request) throttle condition, this would represent a condition where the thrust produced by the engine is much less than the drag induced by the aircraft, therefore the Mach number would decrease very quickly causing the

flight condition to be transient in nature. This would result in a more detailed analysis of the exact flight scenarios at the engine or even aircraft level surrounding the pump sizing point, which would hopefully lead to a more realistic and less severe sizing condition.

Component	Variable		Weight	Product Cost	Performance Parameters		
Actuators	Piston Extend Area	# of Actuators			0.9/40k Heat Load	0.5/10k Heat Load	Design Slew Rate
Actuator #1	4.105	1	16.71	$ 38,419			4.0
Actuator #2	2.794	1	14.09	$ 33,175			2.0
Actuator #3	4.799	6	61.59	$ 133,185			4.2
Pumps							
Fuel Pump Impeller Dia. (in.)	1.485		22.4	$ 26,911	1400	1350	
Fuel Pump Impellar Depth (in.)	1.500						
Fuel Pump Displ. (cu in/rev)	3.1		32.1	$ 25,680	1750	1900	

Constraints		
Max Pump Delta P	1600	psid
Min Pump Delta P	200	psid
Max Actuator 1 Size	6	Sq Inches
Max Actuator 2 Size	4	Sq Inches
Max Actuator 3 Size	5.8	Sq Inches

GOAL			*Minimize*	*Minimize*	*Minimize*	*Minimize*	
Totals			146.92	$ 257,371	3150	3250	
Targets			100	$ 160,000	2700	3500	
Percent Target Miss			46.9%	60.9%	16.7%	-7.1%	
Weighting Factor			25.0%	30.0%	22.5%	22.5%	100.0%
Weighted Target Miss			0.117	0.183	0.038	-0.016	0.049

Figure 5-5. Example of a Fuel System Optimization Model

Inputs			Engine Parameters						
Mn	Altitude	FnRq	High Rotor Speed	Burn Flow	A/B Flow	Total Flow	Act #1 Load	Act #2 Load	Act #3 Load
	Feet	Percent	Percent	Percent	Percent	Percent	Percent	Percent	Percent
0	0	10.0	74.2	7.5	0.0	2.6	6.5	6.5	-26.6
Max	0	50.0	94.8	55.4	0.0	19.4	70.3	36.6	-94.7
Max	0	100.0	99.1	81.4	0.0	28.5	69.8	48.5	-119.3
Max	0	150.0	99.1	82.7	92.9	89.3	69.8	48.8	-118.7
Min	5000	10.0	72.8	6.2	0.0	2.2	5.4	5.3	-22.1
Max	5000	50.0	97.0	63.6	0.0	22.3	69.7	39.2	-101.5
Min	5000	50.0	85.5	21.4	0.0	7.5	27.9	15.5	-39.9
Max	5000	100.0	99.5	78.9	0.0	27.6	69.3	47.2	-116.0
Min	5000	100.0	93.7	43.4	0.0	15.2	14.8	18.7	-58.4
Max	5000	150.0	99.5	80.1	92.9	88.4	69.3	47.6	-115.5
Min	5000	150.0	93.7	44.2	59.8	54.4	14.9	18.0	-57.6
Min	10000	10.0	72.1	5.3	0.0	1.9	4.5	4.7	-18.6
Max	10000	50.0	97.5	63.0	0.0	22.0	65.4	37.9	-98.0
Min	10000	50.0	84.0	17.4	0.0	6.1	23.1	12.8	-33.0
Max	10000	100.0	100.3	79.7	0.0	27.9	64.9	46.0	-113.2
Min	10000	100.0	92.1	35.1	0.0	12.3	12.1	15.5	-48.2
Max	10000	150.0	100.3	81.0	92.9	89.7	64.9	46.3	-112.7
Min	10000	150.0	92.1	35.8	51.0	45.7	12.2	14.9	-47.6
Min	20000	10.0	74.4	6.0	0.0	2.1	4.8	5.0	-15.5
Max	20000	50.0	96.9	62.1	0.0	21.7	59.9	35.9	-92.8
Min	20000	50.0	80.9	11.4	0.0	4.0	15.4	8.6	-22.0
Max	20000	100.0	101.1	74.7	0.0	26.1	60.0	42.5	-104.4
Min	20000	100.0	88.7	22.6	0.0	7.9	7.9	10.6	-32.1
Max	20000	150.0	101.7	79.1	92.9	88.1	59.9	43.6	-106.4
Min	20000	150.0	90.3	25.5	36.7	32.8	5.4	9.7	-33.2
Min	30000	10.0	74.6	5.5	0.0	1.9	4.5	4.5	-12.7
Max	30000	50.0	100.2	52.8	0.0	18.5	52.2	30.3	-78.3
Min	30000	50.0	79.3	9.0	0.0	3.1	11.2	6.7	-17.1
Max	30000	100.0	101.5	58.9	0.0	20.6	52.3	34.5	-84.7
Min	30000	100.0	86.1	16.8	0.0	5.9	6.0	8.3	-24.7
Max	30000	150.0	101.7	60.7	83.4	75.5	52.3	35.0	-85.0
Min	30000	150.0	87.7	19.5	29.7	26.1	4.0	7.6	-25.9
Min	40000	10.0	77.0	5.7	0.0	2.0	7.6	4.4	-11.0
Min	40000	50.0	78.2	6.3	0.0	2.2	7.6	4.7	-11.7
Max	40000	100.0	100.5	41.5	0.0	14.5	32.6	23.3	-57.4
Min	40000	100.0	84.8	11.4	0.0	4.0	4.0	5.9	-16.7
Max	40000	150.0	101.7	46.3	60.6	55.6	31.6	24.6	-60.2
Min	40000	150.0	86.5	13.2	13.0	13.1	2.6	5.4	-17.7
Min	50000	10.0	84.8	7.7	0.0	2.7	4.4	4.5	-11.2
Min	50000	50.0	84.2	7.2	0.0	2.5	5.0	4.4	-10.8
Min	50000	100.0	86.8	9.0	0.0	3.1	2.9	4.7	-12.3
Min	50000	150.0	86.6	10.5	9.0	9.5	1.8	4.3	-13.1

Figure 5-6. Example of Engine Simulation Deck Output

5.4 Risks and Benefits Compared to Current Process

As stated in the business case analysis (see section 5.2.6), a good potential exists for reduced program cost by invoking the GTA development concept. At the very least, the first

program implementing the framework would have a reasonable chance to break even with subsequent programs reaping the benefits of GTA hardware re-use.

Program schedule has the potential for negative impact unless careful planning of key activities and decision milestones and managing to the plan is accomplished. Although the development schedule becomes must less arduous leading up to first hardware delivery since hard design decisions are being delayed, schedule compression is still likely to occur between FETT and final hardware delivery just prior to flight certification. Management support of acquisition and analysis of key engine data at the earliest possible opportunity is of the utmost importance to avoid a program delay.

Clearly, the primary benefit of the GTA development concept is more optimum component designs, resulting in lower cost and weight and ultimately a more satisfied customer. A secondary benefit is the ability to design components right the first time to "real" requirements, rather than designing to educated guesses, knowing that downstream iteration is inevitable. This approach is psychologically more palatable to the average engineer and will result in a happier, more motivated work force.

5.5 Chapter Summary

This chapter described an improved methodology for control component development in which design requirement decisions are delayed until uncertainty can be resolved. This is not a "one-size-fits-all" framework since different levels of risk are associated with the various control system components. The initial step of the revised process is to assess the risk of design requirement uncertainty for all components and make a judgment on application of the revised process. The risk threshold should be defined on the "local" level by program management based on program goals and performance against those goals.

At the heart of the revised process is the hardware concept that allows the engine development program to proceed in the absence of certain "production configuration" components whose designs are being delayed. The Generic Test Apparatus (GTA) represents the "boilerplate" version of these components that will act as surrogates during initial ground engine test in order to gather key data to resolve requirement uncertainty. Several things must be considered when architecting, designing, and implementing GTA hardware. A key architectural consideration is to match flexibility to uncertainty (i.e. ensure the GTA architecture provides the ability to react to various levels of performance for parameters that typically are uncertain). The designs must ensure functional robustness by covering requirement uncertainty and must avoid inducing more risk (due to their non-production design) than they mitigate. This is done by paying careful attention to interface and technology maturity considerations. The product development program for affected components must be carefully planned to ensure key data is acquired, analyzed, and rolled into the system design in time to support component design and manufacturing cycles. Timely system design and optimization can be accomplished through the utilization of linear optimization tools. This approach lends itself well to the complexity and diversity of control system components that must observe many different constraints such as physical space and operating pressures while attempting to balance conflicting requirements such as cost, weight, and heat generation.

Business cases were compiled for two scenarios – (1) Avoidance of downstream design iteration to correct non-optimum designs and (2) Avoidance of the life cycle cost (LCC) impact to the customer/operator of fielded non-optimum designs. For both scenarios, cost/benefit ratios were calculated for two GTA hardware implementation cases – (a) assuming three actuator part

numbers and three EHSV part numbers are affected and (b) assuming the components listed in (a) plus one fuel pump part number. The results are summarized in Table 5-1 below.

Table 5-1. Results of Business Case Analysis for GTA Implementation

Scenario	Description	Components Affected	Cost/Benefit Ratio
1a	Downstream development program iteration avoidance (one development program)	3 actuators, 3 EHSVs	0.77
1b	Downstream development program iteration avoidance (one development program)	3 actuators, 3 EHSVs, one fuel pump	0.69
2a	LCC avoidance (first year of field operation)	3 actuators, 3 EHSVs	0.80
2b	LCC avoidance (first year of field operation)	3 actuators, 3 EHSVs, one fuel pump	0.66

A cost/benefit ratio (CBR) of less than 1.0 indicates that the cost to implement is less than the benefits received; therefore, each of the business case scenarios analyzed showed a positive result. It is important to note that the CBRs were calculated assuming the accrual of benefits for a single development program for Scenario (1) and for only the first year of field operations for Scenario (2), which are extremely conservative. As the GTA hardware is re-used on subsequent development programs, the business case becomes even more favorable. Likewise, as an engine continues field operations, the customer's LCC savings increases, further improving the business case.

6 ORGANIZATIONAL ANALYSIS

6.1 Current Organization

The current organizational structure at PW is shown in Figure 6-1[27]. For the purposes of this discussion, the figure has been simplified to include only engineering entities. For each program, the Executive Committee selects a program manager and approves the selection of the Integrated Product Management Team (IPMT), which has responsibility for program execution. Within the IPMT, the chief engineer is responsible for allocating and flowing all technical, cost, and delivery requirements to subsystems and parts via the Propulsion System/Component Requirements Document (PS/CRD). This is a dual-role filled by one of the three engineering managers and is usually filled by the design integration manager who is part of the System Design & Component Integration (SDCI) organization. Under the current process, SDCI provides direct flowdown requirements from higher-level specifications, such as the engine model specification and airframe interface document, as well as module-level allocations of propulsion system requirements such as product cost, weight, reliability, safety, maintainability, vulnerability, etc. Functional requirements, however, are defined and flowed by the analytical arm of Systems Engineering known as Propulsion Systems Analysis (PSA). For the control system, the majority of cost and weight is driven strictly by functional requirements. This stands to reason since the control system, by its nature, is a module satisfying the purely functional need of controlling the engine. There does not appear to be any clear connection between the propulsion system requirements as listed above (particularly cost and weight) that are allocated by SDCI and functional requirements levied by PSA. Since PSA has the responsibility to ensure

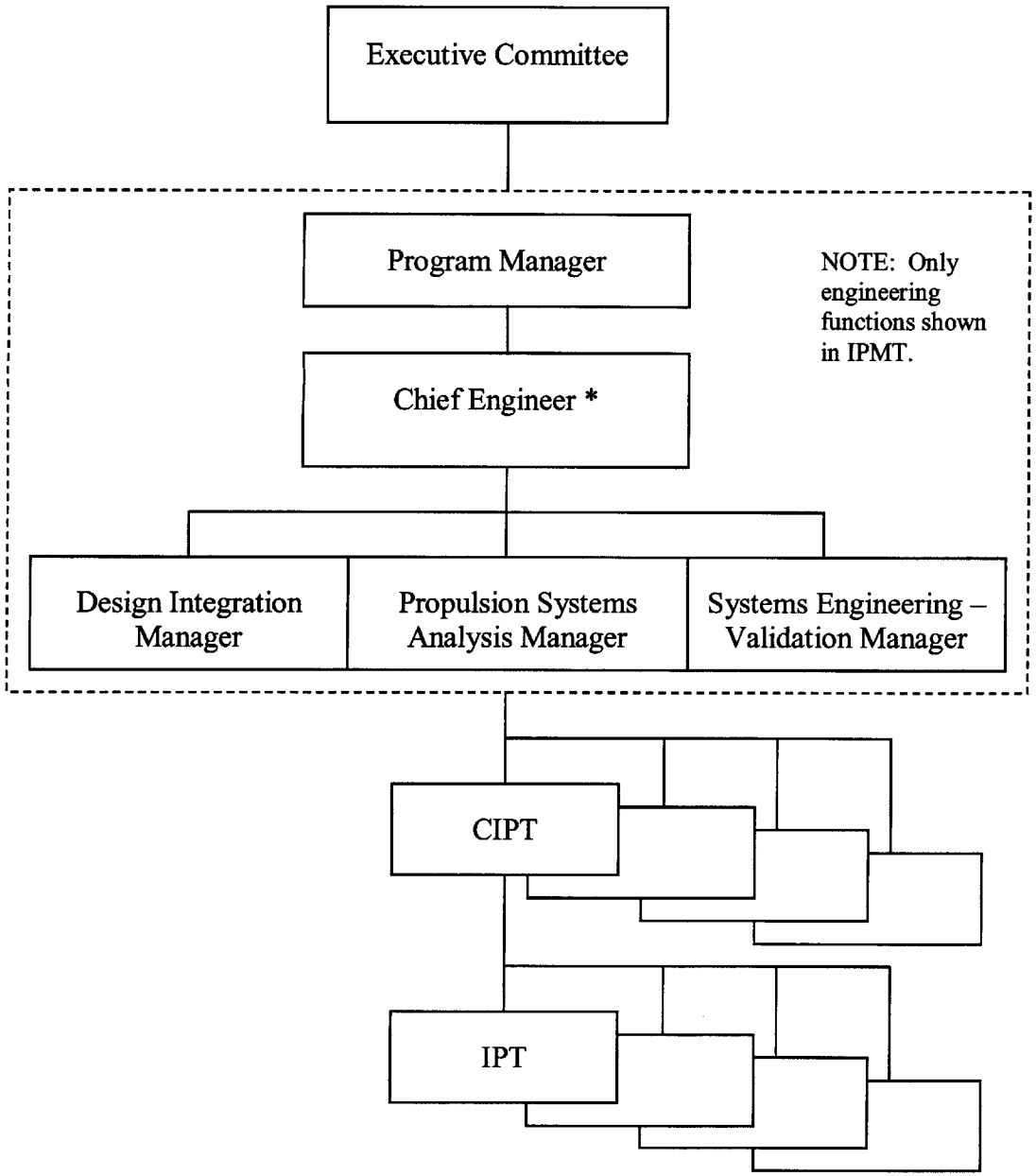
---

[27] Internal Pratt & Whitney documentation.

that the engine meets functional requirements but is not involved in the cost/weight allocation process, modules where cost and weight are heavily influenced by functional requirements (such as the control system) are destined for significant conflict in the attempt to meet all requirements. Although one of the roles of the Chief Engineer is to resolve technical conflicts and issues, the basic conflict of cost and weight requirements versus functional requirements in essence originates within the SDCI/PSA organizations. To be fair, equitable allocation of cost and weight is an extremely difficult task and is essentially a no-win situation for the Chief Engineer. All modules must be challenged to achieve aggressive cost and weight targets at the propulsion system level. However, if a process exists that ties functional requirements to cost and weight allocations, it is not evident.

Another organizational process issue that may be a bit more manageable concerns reallocation or transfer of cost/weight target for subsystem Integrated Product Teams (IPTs) that cross module center boundaries. When an IPT is formed for a module or subsystem that crosses module center boundaries (such as variable geometry actuation in which the fan, compressor, or nozzle owns the variable geometry, but controls owns the actuation system), there is no formal process to equitably reallocate cost/weight based on trade studies that result in an overall propulsion system cost/weight reduction. This is particularly detrimental when additional cost/weight is taken on by one module center to allow the other to gain a cost/weight reduction resulting in an overall reduction at the system level. This behavior tends to stifle creativity and innovative thinking if team members believe they will not receive some reward (in the form of additional component weight/cost allocation) for ideas that result in a system improvement.

.



Executive Committee
- Strategy

Integrated Program Management Team (IPMT)
- Flawless execution of program

* Chief Engineer is a dual role designated from one of the three engineering managers.

Component Integrated Product Team (CIPT)
- Manages all aspects of Module
- One set of CIPTs per IPMT
- HS functions as one of the CIPTs

Integrated Product Team (IPT)
- Manages all aspects of assigned parts

Figure 6-1. Current PW Organizational Structure

6.2 Organizational Improvements

It is not readily clear if the issue of disjointed allocated and functional requirements can be solved through process or if an organizational change is needed. It might be possible to resolve this issue through an improvement in the integration process of allocated and functional requirements, but it might take more than mere complaining from the receivers of requirements to make this happen. For module centers whose product's attributes are more heavily driven by functional requirements than by structural/durability requirements, it may make more sense to integrate organizational aspects of *all* requirements flowdown processes. In other words, move responsibility for flowdown of all requirements to a single organization within the Systems Engineering organization with the expertise to understand the trades between cost/weight and functionality, namely the Chief Engineer. Currently, the Chief Engineer plays more of a "hands off' role when it comes to functional requirements flowdown, leaving this responsibility to the analytical arm of Systems Engineering, Propulsion Systems Analysis (PSA). To some extent, the incentive structure to link allocated and functional requirements is already in place since the Chief Engineer is responsible for meeting technical and affordability requirements at the integrated propulsion system level. It is in his best interest to equitably spread the "challenge" of cost and weight allocations across modules since under-challenged modules will stop iterating too soon and over-challenged modules may not be able to achieve the allocations due to other technical requirements. In practice, however, the Chief Engineer plays a more active role in resolving disputes and requirement challenges than in the initial requirement flowdown process, considering requirements in a holistic sense.

Comparing organizational responsibilities of the PW Chief Engineer to a Toyota *shusa* (program manager or chief engineer of a new automobile development program), many

similarities are evident. Both positions appear to hold significant power and influence to affect the ultimate design of the product, but in the PW organization, the Chief Engineer is primarily focused on the technical aspects of the product whereas the Toyota *shusa* appears to have total program responsibility[28]. The PW Program Manager (to whom the Chief Engineer reports on a matrix basis) is ultimately responsible for ALL aspects of the product, from marketing to technical to logistical support to warranty budgeting. This would seem logical given the increased technical complexity and criticality of a jet engine over an automobile, and due to the substantially different cost of ownership and life cycle. These aspects require much more planning and strategy in the gas turbine engine world, therefore require a larger and more diverse organization to manage them.

Regarding the issue of equitable reallocation of cost and weight across module center boundaries, this could easily be handled by process without an organizational change. Allocation of cost and weight as well as issue resolution clearly is the job of the Chief Engineer. This could be a simple two-step process beginning with 1) reallocating cost/weight between module centers to "level the playing field" and then 2) reallocate remaining savings equally among participating module centers. The net result would be for each participating module to make an equivalent weight reduction against its allocation. As an example, let us assume that two module centers A and B are each partial owners in a particular engine subsystem and the weight status of each currently equals their respective weight targets. If module A makes a change that results in a weight increase of 4 pounds, allowing module B to reduce weight by 14 pounds (a net savings of 10 pounds),

---

[28] Womack, James P, Daniel T. Jones, and Daniel Roos. The Machine that Changed the World. New York: Rawson Associates, 1990.

1. 4 pounds of target weight allocation to be transferred from module B to module A, resulting in a net zero against A's target. Module B would be at a net 10 pounds under its target.
2. This remaining 10 pounds under the target could be equally divided between the two participating module centers, resulting in a total of 9 pounds of target being transferred from module B to module A (4 to cover the original investment and 5 of the remaining system savings) for a net savings to module A of 5 pounds.

The weight status of Module B would be reduced by 14 pounds by incorporating the change, but by transferring 9 pounds of target, it would make a net gain against target of 5 pounds (14 – 9), the same as module A. This simple process change could encourage more innovative behaviors resulting in various module centers proposing solutions that are more integrated.

6.3 Chapter Summary

This chapter discusses issues of an organizational nature that are associated with the requirement flowdown and allocation process as it currently exists between PW and HS. One issue that was explored involves the linkage, or lack thereof, between derived functional requirements and allocated requirements such as cost and weight. The issue appears to be rooted in the Systems Engineering (SE) organizational structure since functional decomposition occurs within the analytical entity (PSA) and requirement allocation is performed by the Chief Engineer. The lack of linkage between these two types of requirements can result in requirement conflict leading to excessive engineering iteration during the development program. Either an organizational change to merge the two activities or a process change to better integrate the activities would improve the situation.

The issue of equitable reallocation of cost and weight across module center boundaries resulting from system trade studies can be handled by a simple process changes. Resolution of

this issue will serve to encourage innovative thinking at the system level with the assurance of receiving fair and equitable distribution of the trade study benefits.

7 CONCLUSIONS AND FUTURE WORK

7.1 Viability of Revised Development Framework

The analysis discussed in Chapter 5 supported the hypothesis that a net benefit can be realized by delaying design decisions for certain "high risk" control system components. A business case was presented that showed potential payback of the investment required to implement the development framework within a single development program. Proper attention to the architecture and design of the GTA hardware will allow the assets to be utilized on future programs, further bolstering the business case. If a future program contemplates implementing this approach, a specific business case should be formulated for that application to ensure viability.

In addition to the positive business aspects, the framework utilizes a much more intuitive approach to dealing with uncertainty, allowing decisions to be delayed until uncertainty is resolved. This approach serves only to move, not eliminate, the schedule compression so common to today's aggressive development programs. However, development engineers should be able to more easily deal with this schedule compression psychologically, since they will be more confident in the optimality of their initial design since they are designing to measured environments rather than estimates and predictions. Therefore, the revised framework could also result in an improved work environment.

7.2 Future Work

This work brought to light several areas for follow-on activity. One area that should be pursued is an improvement to the requirements definition process. Significant advances have been made in the past few years with regard to requirement decomposition, flowdown, and

documentation. However, further work needs to be done to improve the linkage between technical requirements and attribute allocations such as product cost and weight. Significant amounts of engineering resources are being spent attempting to achieve unattainable goals. Further research should be done to try to understand the specific relationships between technical requirements and design space in an attempt to write more realistic specifications at early development program stages.

An area within the proposed development framework that deserves additional work is quantification of the component risk assessment process. The current process specifies that component attribute impacts (recurring and non-recurring) be identified based on the level of estimated requirement uncertainty. This is no easy task since relatively detailed analytical models of components and subsystems are required to obtain a reasonable understanding of the impacts. Even with quantified attribute impacts, the assimilation of these impacts to make a final decision as to whether or not to apply the revised framework requires a high degree of subjectivity. Additional work needs to be done to provide a common "yardstick" to relate various attributes such as development cost, product cost, weight, safety, and heat generation. In the same vein, the risks associated with non-BOM interfaces and functional differences should be better quantified to allow management a clearer picture of the risks and benefits of the revised framework.

GLOSSARY

AEDC	Arnold Engineering Development Center
AMT	Accelerated Mission Test
BOM	Bill of Material
BPR	Bypass Ratio
CBR	Cost/Benefit Ratio
CE	Concurrent Engineering
CFD	Computational Fluid Dynamics
CLB	Closed-Loop Bench
COTS	Commercial Off-the-Shelf
DVT	Design Verification Testing
EC	Engineering Change
EEC	Electronic Engine Control
EHSV	Electrohydraulic Servo Valve
EMI	Electromagnetic Interference
EMID	Electro-Mechanical Interface Device
EPR	Engine Pressure Ratio
FADEC	Full Authority Digital Electronic Control
FETT	First Engine to Test
GTA	Generic Test Apparatus
HCF	High Cycle Fatigue
HPC	High Pressure Compressor
HPT	High Pressure Turbine
HS	Hamilton Sundstrand
I/O	Input/Output
IPD	Integrated Product Development
IPT	Integrated Product Team
LCC	Life Cycle Cost
LCF	Low Cycle Fatigue
LPC	Low Pressure Compressor
LVDT	Linear Variable Displacement Transducer
NRE	Non-Recurring Engineering
PS/CRD	Propulsion System/Component Requirements Document
PW	Pratt & Whitney
SE	Systems Engineering
TSFC	Thrust-Specific Fuel Consumption

BIBLIOGRAPHY

Ward, Allen, Jeffrey K. Liker, John J. Cristiano, and Durward K. Sobek II. "The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster." Sloan Management Review, Vol. 36, Issue 3 (Spring 1995), 43.

Hague, Douglas C. "Description of a Turbofan Engine Product Development Process." SM Thesis, Massachusetts Institute of Technology, 2000

Crawley, E. F., Adapted from System Architecture Class Notes, 1/23/01.

Thomke, Stephan, Enlightened Experimentation – The New Imperative for Innovation, Harvard Business Review, 2001

Federal Acquisition Regulation (FAR) Subpart 37.6 (http://www.acqnet.gov/far/current/pdf/FAR.book.pdf)

Ulrich, Karl T. and Steven D. Eppinger, Product Design and Development, McGraw-Hill, Inc., 2000

Neimeyer, Jonathan K. "Framework for Risk Reduction in Gas Turbine Product Development." SM Thesis, Massachusetts Institute of Technology, 2002.

Support Services, Pratt & Whitney, The Aircraft Gas Turbine Engine and Its Operation, United Technologies Corp, 1982.

Browning, Tyson R. "Modeling and Analyzing Cost, Schedule, and Performance in Complex System Product Development." PhD Thesis, Massachusetts Institute of Technology, 1999.

Johnson, Rodney, personal conversation, September 2002.

Internal Pratt & Whitney documentation.

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

Moy, Habs M. "Commercial Gas Turbine Engine Platform Strategy and Design." SM Thesis, Massachusetts Institute of Technology, 2000.

Chang, Tzyy-Shuh, Allen C. Ward, Jinkoo Lee, and Edwin H. Jacox. "Distributed Design with Conceptual Robustness: A Procedure Based on Taguchi's Parameter Design", Concurrent Product Design, Vol 74, November, 1994.

Womack, James P, Daniel T. Jones, and Daniel Roos. The Machine that Changed the World. New York: Rawson Associates, 1990.

3853-113