
by

Matthew J. Aquaro

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 2008

May 2008

© 2008 Matthew J. Aquaro
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 in any medium not known or hereafter created

Matthew J. Aquaro
System Design and Management Program
May 2008

Daniel Whitney
Thesis Supervisor
MIT – Engineering Systems Division

Maria Yang
Thesis Supervisor
MIT – Engineering Systems Division

Patrick Hale
Director
System Design and Management Program
Abstract

Product development in the automotive industry has evolved around the design of components. The organization is established around components and people have a very component centric perspective on problem solving. This has led to local optimization of individual components, while the larger system spirals out of control. The penalty is often measured in terms of development time and cost. New programs are given autonomy to make independent choices without regard for what other programs are doing, which leads to a wide variety of architectures put into place. Program managers and functional managers have different prioritizations. Furthermore, new designs are provided by a separate organization from the group responsible for implementation. They have a very different value system and are unaware of the difficulties experienced in the implementation phase. This type of practice leads to programs nearing production deadlines with poorly optimized systems. Engineers must relearn due to the lack of standardization across program. The team absorbs additional resources from within to fix issues prior to launch. The robbing of resources leads to delays in subsequent programs and the cycle repeats itself. These issues are partly cultural, part organizational, part due to lack of understanding of systems engineering.

A new organization is designed, which strengthen the systems perspective and give power to a new role in the organization, the Systems Engineer. The Systems Engineer is chartered with global optimization of the entire system, which includes both functional aspects as well as business aspects like resource availability, development cost and time. They are responsible for developing the complete system, from concept to final implementation. The Design Structure Matrix (DSM) shows the boundaries of the system and reveals new areas where the Systems Engineer can influence the design at lower cost to the organization. The Robustness Checklist, standardization and Systems Architecture provide Systems Engineers tools to change from a component mindset to a systems mindset and to optimize the system as a whole.
Acknowledgements

The SDM program at MIT proved to be more difficult than I had ever expected. I owe tremendous amounts of gratitude to my wife, Maggie, and my children, Olivia, Annie, and Joey for their love and understanding throughout this long journey. There were many nights were I was too tired to play, but still the children love me. There were also many nights that Maggie spent alone, but still she loves me. Tonight is no exception. Despite this, we all learned a lot, some about systems engineering and management but more about ourselves, about priorities and about what is important and what is not. I’ve been often reminded of what a slow learner I really am. But the greatest challenge was yet to come.

Near the end of this journey our family began a new journey when Joey was diagnosed with Neuroblastoma, a rare children’s cancer. This event more than any other, has reinforced the idea that we cannot do anything alone. Whether it is to get a degree from MIT, or to fight cancer, we are reminded that God is there with us, holding us up all the time. The structures that we build up around us that we feel keep us strong are actually quite frail. We have intellect, education, money, family, and friends but still, we are knocked down. God helped our family to get through MIT, when all those things were not enough, and now He is helping us to fight the next great fight that our family has been asked to face. God has been very good to us indeed, providing us with the patience, courage, perseverance and most of all LOVE to stand through these storms, and so to Him go the greatest thanks.
# Table of Contents

## CHAPTER 1 INTRODUCTION
- Systems Engineering ................................................. 1

## CHAPTER 2 SYSTEMS THINKING
- DSM / Attribute DSM ................................................. 9
- Global Optimization .................................................. 11
- Chapter Summary ...................................................... 15

## CHAPTER 3 ORGANIZATION
- Program Managers vs. Functional Managers .................. 16
- Corporate Culture ..................................................... 19
- The Balance of Power ............................................... 21
- Chapter Summary ...................................................... 24

## CHAPTER 4 THE PRODUCT DEVELOPMENT PROCESS
- The PDP High Level Overview .................................... 26
- Advanced Engineering ............................................... 28
- Software Engineering ............................................... 29
- Chapter Summary ...................................................... 31

## CHAPTER 5 CREATING A SYSTEM THAT MEETS THE NEEDS OF ITS BENEFICIARIES THROUGH SYSTEM ARCHITECTURE
- System Architecture: Understanding the Beneficiaries of the System .................................................. 33
- Combinatorial Concept Selection Method .................... 39
- Needs of the System .................................................. 41
- Meeting the Needs of the Beneficiaries ....................... 42
  - Critical Characteristics ........................................... 43
  - Important Characteristics ....................................... 45
- Chapter Summary ...................................................... 46

## CHAPTER 6 MOVING FROM A COMPONENT TO A SYSTEMS LEVEL FOCUS THROUGH ROBUSTNESS CHECKLISTS AND DESIGN STRUCTURE MATRICES (DSMS)
- A purely functional view of the world leads to local optimization .................................................. 47
- An expanded view of the boundaries of the systems engineer reveal more opportunity for optimization ........................................ 49
- A new approach – the DSM ........................................ 50
- The Design Space ..................................................... 52
- Chapter Summary ...................................................... 54

## CHAPTER 7 STANDARDIZATION
- Analytical Tools and Processes ................................... 56
  - Traditional story..................................................... 56
  - A new approach through standardization .................. 60
- Chapter Summary ...................................................... 63

## CHAPTER 8 CONCLUSIONS
- Organizational Structure Changes .............................. 65
- Tools Steps and Processes that Facilitate Institutionalization of Knowledge ....................................... 69
- Future Work ............................................................ 70
- Bibliography ............................................................ 71
List of Figures

FIGURE 2-1: EXAMPLE DSM - THE X'S REPRESENT INTER-RELATIONSHIPS. COLUMNS REPRESENT DEPENDANT RELATIONSHIPS. ROWS REPRESENT PRECEDENT RELATIONSHIPS.................................................................10

FIGURE 2-2: ATTRIBUTE DSM FOR AUTOMATIC TRANSMISSION HYDRAULIC CONTROL SYSTEM. MANY X'S REPRESENT A HIGH DEGREE OF COUPLING. THERE ARE FEW RELATIONSHIPS THAT ARE NOT DEPENDANT ON EACH OTHER.........................................................................................................................11

FIGURE 2-3: IRON TRIANGLE. SCHEDULE, FUNCTION AND COST ARE BOUND TOGETHER. THERE ARE NO INDEPENDENT RELATIONSHIPS. THE DESIGN MUST BE OPTIMIZED WITHIN THE BOUNDS OF THE IRON TRIANGLE ................12

FIGURE 2-4: SYSTEM DYNAMICS - THE DESIGN PROCESS IS DYNAMIC, AND THERE ARE MANY LOOPS THE MUST BE CLOSED AND REPEATED TO EXECUTE A DESIGN. THIS FIGURE SHOWS THAT THE AMOUNT OF WORK BUDGETED WAS MUCH LOWER THAN THE AMOUNT OF WORK ACTUALLY DONE ........................................14

FIGURE 3-1: ORGANIZATIONS MUST RESPOND TO THE MARKET AND ALSO MUST MAINTAIN TECHNICAL EXPERTISE. THE MATRIX ORGANIZATION WAS CREATED TO MEET BOTH NEEDS. ORGANIZATIONS MUST FIGURE OUT HOW TO MANAGE THE MATRIX .................................................................18

FIGURE 3-2: AUTOMATIC TRANSMISSION ORG CHART FOR NA OEM. TWO PROGRAM CHIEFS WITH PROGRAM MANAGERS REPORTING. FUNCTIONAL ORGANIZATION FALLS UNDER TWO COMPONENT MANAGERS. COMPONENT ENGINEERS ARE MATRIXED TO EACH PROGRAM MANAGER BASED ON NEED..................................................20

FIGURE 3-3: A MATRIX VIEW OF THE NA OEM ORGANIZATIONAL STRUCTURE FOR ONE TRANSMISSION DEPARTMENT. ALSO INTRODUCED ARE ADVANCED TRANSMISSION AND SOFTWARE ENGINEERING .................................22

FIGURE 4-1: IMPLEMENTATION READINESS IS A MEASUREMENT OF THE PROGRESS OF A DESIGN TOWARDS PRODUCTION READINESS. MANY TIMES THE HANDOFF PROCESS RESULTS IN LOST INFORMATION OR A MISALIGNMENT OF THE DEFINITION OF IR AND REWORK MUST BE DONE ..................................................29

FIGURE 4-2: ADVANCED ENGINEERING AND SOFTWARE ENGINEERING PROVIDE SIGNIFICANT CONTENT AND INFLUENCE THE DESIGN SPACE OF NEW TRANSMISSION AND SUBSYSTEM TECHNOLOGIES, BUT ARE REMOVED ORGANIZATIONALLY. THIS SEPARATION CREATES CONFLICT AND DISCONTINUITY OF THE DESIGN PROCESS ........................................30

FIGURE 5-1: EXAMPLE OPM DIAGRAM FOR A HYDRAULIC CONTROL SYSTEM. THE POWER IN THE OPM IS ITS ABILITY TO CAPTURE INTENT OF THE ARCHITECT, NOT JUST THE FORM THAT THE SYSTEM TAKES ON ........................................37

FIGURE 5-2: INTENT IS MAPPED TO THE PROCESS SELECTED FOR ACCOMPLISHING THE INTENT. THEN THE SPECIFICS OF FORM ARE FREE TO FLOW OUT OF THE HIGHER LEVEL NEEDS ........................................................................38

FIGURE 5-3: COMBINATORIAL MATRICES CAN BE USED TO DISCOVER NEW COMBINATIONS AND TO PIECE TOGETHER NEW ARCHITECTURES ...........................................................................39

FIGURE 6-1: ROBUSTNESS CHECKLIST - THIS CHECKLIST HELPS TO ESTABLISH LINKAGES BETWEEN COMPONENTS AND SYSTEM PERFORMANCE ATTRIBUTES ........................................................................49

FIGURE 6-2: ROBUSTNESS CHECKLIST - THE CHECKLIST IS EXPANDED TO INCLUDE OTHER ATTRIBUTES NOT CONSIDERED TO BE WITHIN THE SYSTEMS ENGINEER'S ROLE OF RESPONSIBILITY. THESE ATTRIBUTES ARE HEAVILY INFLUENCED BY CHOICES MADE REGARDING THE DESIGN AS SHOWN IN THIS NEW MATRIX ..................................................50

FIGURE 6-3: DSM THAT SHOWS THE STEPS IN THE DESIGN PROCESS OF THE HYDRAULIC CONTROL SYSTEM AND THE INTERDEPENDENCIES OF THOSE TASKS ...........................................................................51

FIGURE 6-4: AN EXAMPLE OF A SIMULATED SHIFT EVENT WITH OSCILLATIONS PRESENT. ALSO SHOWN ARE CHANGES TO THE SOFTWARE TO MITIGATE THE OSCILLATION WHILE STILL DESTROKING THE CLUTCH ON TIME. THIS WAS PREVIOUSLY THOUGHT TO BE A SOLENOID ISSUE, NOT A SYSTEM ISSUE ..........53

FIGURE 7-1: ACTUAL SHIFT TRACE FROM VEHICLE WHERE THERE IS AN OVERSHOOT (SHOWN IN BLUE) THAT DOES NOT FOLLOW INPUT (COMMAND IN BLACK AND ACTUAL INPUT SIGNAL IN RED) ........................................58

FIGURE 7-2: MODEL RESULTS - VERY STRONG CORRELATION BUT ONLY AFTER THE OVERSHOOT WAS DISCOVERED AND MEASURED EXPERIMENTALLY .............................................................59

FIGURE 8-1: SYSTEMS ENGINEERING OVERSEES COMPONENTS, SOFTWARE, CORE CALIBRATION AND ADVANCED AS THEY ALL RELATE TO THE SYSTEM. IMPORTANTLY, THE TEAM DELIVERS FULL SYSTEMS TO THE TRANSMISSION PROGRAM MANAGERS ........................................66
Chapter 1 Introduction

This work has two different readers. The first are the management of the sponsor company. The second are the systems engineers who follow in the author's footsteps. A key reason for this work is to highlight the benefits that can be achieved in terms of shortened development cycles, lower resource expenditures (in terms of people, prototypes and development costs) and improved function. The sponsor company is a major North American OEM of automobiles, hereto referred to as NA OEM. The division that is sponsoring this work is the automatic transmission product development group.

NA OEM has been a component centric company for decades. It has evolved into an organization with clearly defined boundaries in component space that interface with each other but the interfaces are not managed in a systematic way. Specifications are cascaded down from the vehicle level to the Powertrain level and then down to the transmission and finally to the component level. The specifications lose clarity at each interface and by the time a specification arrives at the component level, meeting the specification often has little bearing on the performance of the larger system.

In the late 1990s, the heaviest duty truck transmission in the fleet was nearing the end of its life cycle. It was a four speed transmission, and had evolved over the course of 20 years to become a very durable and sound performer, but not capable of meeting fuel economy and shift quality performance targets for the future, so a major redesign was commissioned. Included in this redesign was a new control system. The concept had been implemented by a competitor before, but was not in production for very long, indicating trouble with the execution of the concept, but the concept was so desirable that this new program team adopted it. It was a version of Direct Electronic Shift Control (DESC), where rather than having a complex main control body with dozens of valves and layers, one high flow, high pressure solenoid was assigned to and sent pressurized fluid directly to a each clutch. The concept was so good, it had to be cheaper and easier to implement than any other, or so the program manager thought.
The trouble was, there were no solenoids available in the automotive market that would meet these performance targets, so an engineering specification was drafted and the supply base was market tested. One candidate was selected. Recall that the culture in this organization is component centric, so the systems engineers thought that if they cascaded some performance expectations to the component activity, they could spec a solenoid and have it delivered and everything would work great.

The solenoid was produced by a single supplier and was working well through the development phase. The more that were produced, however, revealed some weaknesses in the robustness of these solenoids in mass production. As more and more transmission prototypes were made, robustness weaknesses began to arise in the sensitivity of the solenoids to the transmission environment, specifically to the contamination mix within the transmission. In a test stand, at the supplier, the solenoids were failing at unprecedented rates, but those that passed the final test, were failing at equally high rates in the transmissions due to contamination sensitivity from the friction material selected for the clutches. A high efficiency filter was spec'ed to clean up the transmission oil. That worked well. However it was a bypass filter, so there was a cleanup delay. The oil took several hours of operation to filter down to clean levels. Next, the engagement strategy showed some interactions with the solenoids. One of the solenoids had exceptionally high range (0-210psi) compared to the others (0-150psi), making it even more sensitive to contamination and it was critical for engagements (Park – Reverse or Park – Drive). There was no other way to do an engagement, the typical control valves had been removed in favor of the high precision solenoids.

At production, the yield at the solenoid supplier was as low as 30%. 70% of all the solenoids made were thrown away! This drove higher investment of equipment to make more solenoids in order to meet the demand with the yield that was being generated. This also drove the price to over double the anticipated piece price. This is in addition to the capital required. The story was equally bad at the transmission plant. The manufacturing process for the transmission components left residue on the parts. Once the transmission was fully assembled and the oil was added, the oil washed the residue off of the parts and the contamination level of the fluid in the transmission rose. The solenoids clogged up with dirt from this fluid and performance degraded.
The First Time Throughput (FTT) of the complete transmissions went down to 80%. Repair bays backed up and the line stopped because there was no place to put the transmissions that were failing the screening test at the end of the line. Some that did pass arrived at the vehicle assembly plant and were built into trucks. The trucks failed at a rate of approximately 10%. Parking lots began filling up with trucks needing replacement solenoid bodies. The transmission plant then tightened the final test stand limits to capture the parts within the transmission plant and to ensure that the trucks that received transmissions would meet the customer's needs. This drove yield at the transmission plant even lower. There were days when the workers were sent home and the assembly line placed on hold so that repairmen could clean out the rework loops and bays by replacing solenoid bodies. There was no room left to store the overflow, so they had to shut down the line. The plant ran overtime to make up for low yield. The solenoid supplier again ran into constraints and had to once again add capacity to make up for low yield.

The team assigned to get this situation under control found ways to revise the engagement strategy and used calibration modifications to make the engagement less sensitive to the solenoid performance, but without a main control body, this process was heavily constrained. Few customers received trucks with poor performance, but it took months to reduce the stockpile at the assembly plant and to get the vehicle yields up to industry par. The team next attacked the transmission plant yield. Work was done on the algorithms that were used to determine good and bad performance and false rejects were eliminated. Next, the engineers worked on the solenoids to increase their robustness to contamination. Soon, yield at the transmission plant rose to 90% and after two years was at 98%. The solenoid supplier was the last to receive help. The price remained at over 2x the pre-launch price for years due to low yield and to recover the investment dollars added to increase capacity. Today, the control system of this transmission is the most expensive in the fleet.

In the end, systems engineering is about global optimization. The team that was assigned to fix this issue was the first such systems engineering team in the control system community of NA OEM. The team learned valuable lessons about component specifications and the need to integrate designs together to form a system. Optimization of the system is the goal, not optimization of the individual components. In the end, the filtration strategy, the main control
body and the solenoids all work together to meet the functional needs of the customer, but also to meet the business needs of the program manager. The program must be delivered on time; this example was launched on time but took two full years to optimize. The program must be delivered on cost. This example will never meet its original cost target. The program must be delivered within resource budgets. This example took dozens of extra people and investment to resolve.

It is not to say that there were not systems engineers. There were. The trouble is that the culture revolved around components and even the systems engineers had a component level view of their responsibilities. What separated this team was a strong individual passion for systems engineering and the autonomy to attack the problem at a systems level. The team consisted of technical specialists, engineers and calibrators (whose roles will be described later in the work) who were all trained in six sigma and who were either raised in or educated in other industries with a strong systems engineering focus, namely MIT and the aerospace industry.

The same team was then assigned to deliver the next transmission control system, but this time from the beginning. The key team members were assigned to a new six speed, front wheel drive transmission program and were chartered with delivering the control system. The team was not told to do things differently from their predecessors, but they had learned lessons through the launch and resolution of the old control system. At the same time, one of the team leaders was going through the SDM program at MIT. What follows is an example of what this new team did differently in delivering the control system of this new transmission, not because the lessons learned were institutionalized in the Product Development Process (PDP), but because the team was made of people who had a passion for systems thinking and who carried the scars of experience on programs that were not developed as systems.

The goal of this work is to convince the management team of NA OEM's transmission division that it is critical to institutionalize the lessons learned by this team. If the team is disbanded or reassigned, then the lessons are lost. Component design guides and component checklists are not enough to institutionalize the systems knowledge. Throughout this work, comparisons will be made of the old way of doing business (component centric) and the new way of doing business
(systems engineered) but the reader should note that the new way is not prescribed within the PDP at NA OEM. That step has not yet occurred, but it is one of the major goals of this work.

The next goal is to leave the systems engineers who follow at NA OEM with a document that they can refer to when struggling with how to deal with the systems engineering aspects of hydraulic controls. Lessons learned about interactions and global optima are captured in this work. These lessons are only applicable, however, as long as the control system architecture is not changed dramatically; standardization is pivotal. If a new way of doing things evolves, then the details of the interactions must be relearned, but the process and systems mindset will make this step natural and easy.

**Systems Engineering**

What is it that makes systems engineering so much more difficult than traditional design or component engineering? When components interact, they form something that neither component could have been separately, the system. This interaction or interface of multiple components to create something new is at the core of systems engineering.

Peter Senge discusses the idea of systems thinking in his book *The Fifth Discipline* and describes a rainstorm. You cannot think about a rainstorm by breaking it into small parts, the wind, the clouds, and dark sky. You have to consider them all together to understand what is happening. Senge says, "[Systems] are bound by invisible fabrics of interrelated actions, which often take years to fully play out their effects on each other." These interrelated actions, or interfaces are the thing that makes systems engineering critical to designing complex products.

The problem that many product development organizations face today, including the transmission community at NA OEM, however, is that they are focused around component engineering, rather than around the system. This lack of recognition of the System is at the heart of the difficulty that many product development organizations face. There is a dedicated unit that does each individual component, but not an equally staffed unit that takes care of the system, and as such, the game continues. "It's not my part, it is your part." "It's not my fault." "I didn’t do it. Prove that my component is the bad one." All of these are common arguments, but all are
fatally flawed, without recognition that someone must be responsible for the interactions between the individual components and more, for the new entity that is created by joining components.

Some facets that contribute to the problem are technical. In this work, the technical pieces of the problem are highlighted and laid out, as well as suggestions and tools that can be used to overcome these technical limitations. The SDM program at MIT is all about Systems Engineering principles applied to product development. The concept of designing systems is so fundamental to the program that it is included in the name. So what is it about SDM that apply to the goal of developing systems engineering methodologies in the product development transmission and driveline engineering community of the NA OEM? Other facets of the problem are organizational and behavioral. This area is also analyzed and studied. Solutions must be dealt with in both spaces, for one without the other cannot be successful.

The journey begins in Chapter Two with descriptions of the elements of systems thinking in depth. Three new concepts are revealed. First is the Design Structure Matrix (DSM) which is a tool that enables engineers to see the world from a system perspective. Second is the Iron Triangle, which states that Cost, Performance and Time are interconnected and mutually constrain each other. One cannot be manipulated without having and affect on the other two. Third is the idea that a system needs to be optimized globally, rather than locally. Each component engineer, software engineer, or researcher may see optimization of their piece, but until the system comes together, it cannot be optimized. Importantly, development time, total cost (piece price + development cost of people, machines, equipment, etc.) and resource constraints are included in global optimization and all are well outside the boundary of a systems engineer as it stands today.

Chapter Three is centered around the organization and culture of NA OEM. It specifically talks in detail about the two different types of managers present in the organization, program and component managers and their different paradigms. Component centric design is the focus of the current organization and there is a distinct lack of focus on optimization of the system as a whole.
In Chapter Four, the PDD is introduced and discussed, but it is really a building block that introduces two additional divisions, namely, Advanced Transmission and Software Engineering, that play integral roles in the development of the hydraulic control system but who are separated organizationally. Being separated organizationally, they behave even more in isolation than do component and program people. Advanced Transmission is responsible for delivering new concepts in the form of whole transmissions. Software Engineering writes the code that drives the hydraulic control system. They are focused on and measured by very different criteria from the hydraulic control systems engineer and hence, constrain the system from ever reaching global optima.

Chapters Five through Seven are more focused on tools, techniques, processes and skills that the systems engineer can use in moving from a component to a systems perspective. Here a new approach begins to emerge. Chapter Five is about cascading customer requirements to systems level attributes and concept selection. The process used is called System Architecture, and it follows a very different process from what is in place today. Introduced is a new way of framing needs of the system and laying out the beneficiaries. This then enables the system engineer to develop a product or family of products that meet all the needs and stand the test of time.

Chapter Six is focused on tools that allow engineers to see the world as systems. The Robustness Checklist connects attributes and targets to functional features. The DSM, goes one step further and defines how those features interrelate to each other.

In Chapter Seven standardization is the main focus. This important concept allows multiple generations of system engineers to build on each others’ work. It allows multiple programs to benefit from each other’s experiences. It also facilitates analytical experimentation by constraining enough of the design to allow models to be predictive, rather than explicative.

Finally, in Chapter Eight, two main questions are attempted to be answered. The first is regarding the organizational structure of NA OEM and other component centric companies.
"What effect does the organizational structure and corporate culture of a company have on the ability of that company (or division) to achieve global optima with respect to the design and development of a complex product?"

The second question is regarding the institutionalization of the skills and knowledge that go into developing a product at a systems level.

"What processes, tools and skills enable engineers to institutionalize systems level engineering in order to be able to execute globally optimal designs on a regular basis."

These two questions will be referred to continually throughout this work. Each chapter touches on a different element of the corporation, process or culture of the subject company and each chapter addresses one of these two core questions in different ways.
Chapter 2 Systems Thinking

This chapter will introduce three major revolutions in converting from a component centric culture to a system centric culture. The first is that of the Design Structure Matrix or DSM. A DSM is a tool that can be used to explain the iterative nature of design and to reveal the interactions between different aspects of the design. In this chapter, a DSM is constructed that is built around the attributes of the hydraulic control system. In later chapters, DSMs will be shown that show the interactions between different components themselves.

The next major concept discussed in this chapter is the concept of the Iron Triangle, or the tradeoff between cost, function and time in the development of technical systems. Finally the concept of global optimization is introduced. This is perhaps one of the most important concepts presented in this work.

**DSM / Attribute DSM**

A Design Structure Matrix is tool that can be used to illustrate a complex design process in a simple form. The metrics that are represented can be functional, attributes, constraints, requirements or many other types of interactions. The different elements are fundamentally not independent of each other, hence the reason for the DSM illustration. It is used to analyze and to understand their interactions. A sample matrix is shown in Figure 2-1 using tasks in a design process as the metrics.
The marks in the DSM represent relationships to other tasks. The diagonal is left blank because tasks cannot be dependant on themselves. Columns in the DSM represent dependant relationships. Rows represent precedent relationships. In the beginning of this example process, the tasks are sequential. Task 2 is dependant on Task 1. Task 3 is dependant on Task 2, and so on. These are simple linear progression. Finish one task and use that information when moving on to the next. The DSM becomes more complicated and useful, however, when tasks are interdependent on each other. Task 7, for example, depends on information from Tasks 4 and 5. But the marks in the upper right hand quadrant indicate that information is required from a downstream task. Task 5 requires information from Task 8, which hasn't been completed yet. Similarly, Task 2 requires input from Task 10, which will not be complete until the end of the project. At this point, information that flows back to Task 2, may change its output, thus driving changes to Task 3, then to Task 4 and Task 8. Thus the iteration begins.

The DSM shown in Figure 2-2 represents attribute interactions in the example described in Chapter 1. At first glance, it is interesting to note how interconnected the attributes are for the hydraulic control system. This is due to the density of the X's in the DSM. Three attributes, cost, schedule and resource availability drive all other attributes and are driven by all other attributes. If the team doesn't have the time or the staff to accomplish a globally optimal design, then all other attributes will be affected. There is no independence, and hence, there is no ability for a team to work on individual attributes of the system and not to work on other attributes. Later in Chapter 3, program managers and functional managers with differing objectives will be reviewed.
and compared against each other. Each will be working towards optimizing some of the attributes in this DSM but not all. One person must be able to manage the tradeoffs of attributes as well as the functional or engineering interfaces between the components. Through this work, it will re-emerge that that person is the systems engineer, and all of these tradeoffs must be in his/her domain.

![DSM Image](image)

**Figure 2-2: Attribute DSM for Automatic Transmission Hydraulic Control System.** Many X's represent a high degree of coupling. There are few relationships that are not dependant on each other.

### Global Optimization

Before discussing global optimization in great detail, it is important to consider the design space and define it. Generally, the term design space refers to the area in which an engineer is given authority to work. If an engineer is responsible for designing pumps, then the design space for that engineer is the pump. A software engineer should not be designing changes in the hardware, since it is out of his/her design space. It is a very important point that the system will define the design space. The DSM shown in Chapter 6 will serve to define the design space and it will show an example of where by expanding the design space, a new level of optimization can be achieved. Next, global optimization will be defined.

Global optimization by definition is a maximum across the entire design space. Local optimization, on the other hand is also achieving a maxima, but of a single attribute, rather than

<table>
<thead>
<tr>
<th>Cost</th>
<th>Weight</th>
<th>Package Space</th>
<th>Shift Feel</th>
<th>Cooler Flow</th>
<th>NVH</th>
<th>Development Time</th>
<th>Resources Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
the entire design space. The key is to understand and define the design space, so that a true global maxima can be reached. The list of attributes of the system that must be satisfied in order to implement the design is quite long. It will come out in subsequent chapters that the design space is actually quite large and extends beyond the organizational space that engineers have typically worked. This is both a cultural (managers giving up their fiefdom) and organizational issue.

A core SDM concept is that of the iron triangle, as shown in Figure 2-3. The iron triangle represents the natural tension between cost, schedule and function that exists in all project management activities. Function is extremely complex to define, because it has so many sub-attributes to it. The functions of the hydraulic control system are shift quality, shift time, NVH, fuel efficiency and cooling/lube. In a later chapter, the beneficiaries’ needs are described in greater detail, and the needs of the system are systematically broken down. For now, it is sufficient to define function as the physical performance of the system.

![Iron Triangle Diagram](image)

Figure 2-3: Iron Triangle. Schedule, Function and Cost are bound together. There are no independent relationships. The design must be optimized within the bounds of the Iron Triangle.

Schedule is the amount of time it takes to develop a product and to put it into production. The Product Development Process (PDP) has a certain cadence built into it, highlighted by important decision points called milestones or tollgates. It is always the goal of a program team to shrink the PDP and to develop products in less time. Global optimization accounts for the competitive advantage of shorter development cycles and being able to respond to the market demands and competition faster.
Of course, no list of outputs is complete without mentioning cost. Global optimization includes the reduction of the summation of the cost of the system. No individual components should be cost reduced at the expense of the system. In order for this to occur, however, individual performance targets must be managed well. If each component engineer has the objectives of delivering cost reduction on their parts or to implement their technology at some cost target, then the system's engineer's job is over constrained. The systems engineer will have competing objectives that steer them away from global optima into local optima. It is important to recognize that cost includes many aspects of the design. Piece cost, labor, overhead, investment are all part of the system cost, but also are costs such as the engineering headcount required to execute a design and the prototype budget for development parts.

Schedule and cost are interrelated in several ways. Manpower, perhaps the most scarce of all resources, both costs money and determines the amount of time that it will take to implement a solution. A well architected system will be easier to implement than one that was designed with inherent limitations. All compromises and substandard design attributes will flush their way into a need for additional resources and attention at the end of the program.

As illustrated in MIT's System Dynamics for Project Management, a poorly managed and or architected system presents an exponentially increasing resource demand curve as it nears production. It is critical for a program's success to be able to implement the technology with as few resources as possible.
Each industry typically has a different means of balancing the iron triangle. In many sectors of the aerospace industry, schedules are typically allowed to slip. NASA in the 1960s spent significant money and added thousands of people to hold to the deadline that President Kennedy set of sending a man to the moon by the end of the decade. There are also cases where missing a launch window will result in decades of time lost waiting for the next window. In other cases, when budget scrutiny is high and schedule pressure is low, dates are allowed to slip while spending remains on track. In the auto industry, schedules are held very firm. Production delays are most costly and resources are typically added to avoid missing the production deadline, regardless of the cost that they contribute to other programs. Rather than letting the schedule slip, resources are added. Figure 2-4 shows an example from Lyneis, Cooper and Els (2001) in which rather than allow the schedule to slip, an automaker increased the resources on the program. In this example, the resources nearly doubled. This is consistent with program launches at NA OEM. Resources are robbed from other programs, as the example in the introduction highlighted. The entire team that was brought in to work-out the problems with the heavy duty truck transmission were scavenged from other teams. After two years, they were reassigned to other teams, but the entire organization had to reconfigure itself during the transitions. This behavior is not seen as increasing the cost of the product, however, since the...
engineers are considered to be a fixed cost and allocations to various programs is poorly tracked. It never shows up as increased cost, but it is. A key goal of this work is to highlight the cost of resources to the development of a program. If the same work can be done in the same time with less resources, then the total cost to deliver is going to be reduced. Companies can take advantage of these savings by reducing their staff levels without a decrease in performance.

**Chapter Summary**

It was important to introduce the DSM tool early in this work, since this tool is core to defining a system and understanding the interactions that are untraceable in a traditional component centric view. The DSM will reemerge often throughout this work in various forms and a clear foundation of it is now established. The DSM is perhaps the most useful tool that a company can institutionalize which will take peoples view to a higher level, allowing them to move on to objective number two, global optimization.

The overarching assumption of this entire work is that systems engineering is at the foundation of properly balancing the iron triangle and achieving global optimization. Without looking at the design space from a systems level, global optimization is inherently impossible and global optimization is again, the major goal: use less resources, meet customer needs and wants, deliver on time and on budget. If these conflicting objectives can all be met, then the design will make shareholders, managers, executives, consumers, activists, regulators and the working team all feel very satisfied that the best balance was achieved.
Chapter 3 Organization

This chapter extends the discussion to the organizational structure and personality types at work inside of NA OEM. As mentioned earlier, the old way of doing business was component centric. The organization, the core values and the corporate culture all work together to support this component minded approach. All three must change in order to make the shift towards system level way of developing products.

One might ask, “systems engineers, oh yea, we’ve got those, so what do YOU have to say that’s new.” At NA OEM, there are indeed systems engineers, but their role is limited. For all practical purposes, the role is used exclusively as a tracker, who then assigns work to component engineers. Early in the product development cycle, they cascade targets and requirements, and as the project moves forward, they begin to cascade blame and gaps in system level performance that are diagnosed to one component or the other. The trouble lies in that the gaps are often not caused by or solvable by changing the components. The new entity exists, called a system, but the Systems Engineer has not taken responsibility for it or become accountable to it.

In a new organization, which will be laid out in Chapter 8, the systems engineer leads component engineers and coordinates with program managers. System engineering is responsible for delivery of attributes of the complete system to the program. These attributes are not all functional, they are both functional and managerial in nature and include cost and schedule. Changes to the organizational chart, the roles and responsibilities and to the culture are all required and each is elaborated on throughout this chapter.

**Program managers vs. Functional Managers**

Organizations must be setup in such a way as to address three separate timelines. First, the rate of change of information in the particular industry must be considered. Second, the time that it takes to develop and launch the product into marketplace and third is the rate that the market changes and adapts. These variables are all key ingredients in the type of organizational
structure chosen to best deliver the product to market. Program lead teams are on the left hand side of Figure 3-1.

The rate of information growth in the industry needs to be carefully considered when organizing a PD environment. If the information changes quickly, then the ties back to the functional department heads need to be stronger. This implies then that the organization will be more technology or product focused, rather than market focused. The rise and fall of the technology is what enables the company to prosper and an intimate and up-to-date understanding of the product is more important for success than a close relationship with the program (market). Technology lead teams are on the top of Figure 3-1.

This is not to downplay the role of a PD engineer being connected to the market demands, but rather to say that the technology in this type of situation is more dominant. All PD must happen with an intimate understanding of the market's use for and acceptance of the technology. An organization that is functionally staffed has silos, representing the different expertise in the organization that report up through various technology managers. For example, a team of pump engineers report to a pump manager, and a team of solenoid engineers report up through a solenoid manager, etc. Again, the component departments are represented on the top of Figure 3-1. They would then feed technologies to a small program management function who would pick “off the shelf” technologies to incorporate into the product. Programs are shown on the left hand side of Figure 3-1.
Figure 3-1: Organizations must respond to the market and also must maintain technical expertise. The Matrix Organization was created to meet both needs. Organizations must figure out how to manage the matrix.

If, on the other hand, the technology is slowly evolving, but the market conditions change quickly, then the organization must address this by making more strong ties with the program team to deliver the project to market quickly. This is done in this case with a less up-to-date technology where the knowledge changes slowly and is not likely to be obsolete by the time the product launches. Therefore the functional relationships can be weaker. This type of an organization has program or project teams, each team consisting of a staff of dedicated engineers who are experts in the various components that go into the product. For example, a transmission program team will have dedicated solenoid, pump, controls and calibration engineers. They will be assigned to that team until the program is completed. NA OEM was setup this way for many years and finally engaged in the matrix form of management in 1998 as described below.

Like most large corporations, and to balance the program and technical demands, the teams at NA OEM are matrixed. A general example of a fully matrixed organization chart is shown in Figure 3-1. There are department heads who are responsible for technological (especially
component) design. There are program managers who are responsible for delivering a complete transmission program.

There is natural tension between these two branches of the corporation. If the technical side is too strong, then the benefits are efficient use of technology, architectures, reuse of design and deep technical expertise. The drawback is that the time to market may suffer and the connection that the team has to the needs of the customer are weak. On the other hand, if the program team is stronger, then the advantages are strong connection to the needs of the market and typically faster delivery, but at the expense of a weaker connection to the technology. This weaker technological connection results in inefficient use of platforms, shortcuts in the technology evolution cycle, which may produce immature products without thorough testing. The tension is necessary though, to maintain equilibrium in the organization.

**Corporate Culture**

The organizational chart shown in Figure 3-1 shows the general nature of a matrix organization and the benefits of program management and functional management have been discussed. Now it is time to outline the organization specifically for NA OEM. A hierarchical org chart is shown in Figure 3-2. This shows the organization in a way to highlight the rank and power structure in NA OEMs Automatic Transmission Division. The Director is at the top, with two Chief Engineers reporting. Managers fall under the Chiefs, but there is no chief for component engineering, only for program management. Due to this type of organization, most of the budget and emphasis falls within the program managers’ area. The component managers do report to the Director directly, but there is a gap in power between the component managers (or generically, functional managers) and the program managers.
Combining this aspect of the organization with the iron triangle from Chapter 2, shows how the different branches have different priorities. The differences are contrasted in Table 3-1 and are based on interviews with managers in both areas. The corporate culture at NA OEM is clear: production dates do not slip! This sends the message to the program manager who is responsible for delivery of the project to the vehicle program. Followed by making the production schedule, the next largest objective of a program manager is cost (but not total cost, only certain cost aspects show up on a program manager’s budget due to the accounting policy and allocations of resources) and finally functional tradeoffs must be made. Cost is important, but not to the total detriment of function, but when the decision is tough, cost is usually chosen until the functional performance degrades below a certain threshold.

Conversely, the component manager is primarily concerned with function, as is the entire component design team. The terms component are specific to NA OEM, the functional managers are called component managers, but the terms throughout this paper are synonymous. The grumblings heard through the group are about how program managers don't want to wait or pay for the best technology. Next most important is schedule, because of the clarity of this priority from the corporate culture and lastly is cost.

It is also interesting to note that the priorities of the component engineers change after production. Once a program is delivered and enters the maintenance phase, cost reduction plans are much more specific and cost becomes the primary driver. Functional requirements are
already established and the team is tasked with reducing cost while maintaining function. Schedule is still a concern in that during each fiscal year, cost objectives are laid out and the component manager must weigh which cost reduction ideas can be implemented in each year to meet their annual targets.

<table>
<thead>
<tr>
<th>Program Manager</th>
<th>Functional (Component) Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Schedule</td>
<td>Function</td>
</tr>
<tr>
<td>2. Cost</td>
<td>Schedule</td>
</tr>
<tr>
<td>3. Function</td>
<td>Cost</td>
</tr>
</tbody>
</table>

Table 3-1: Different branches of the organization have different priorities.

**The Balance of Power**

There is another way of looking at the organizational structure in Figure 3-2 that shows the matrix in a more clear way. Figure 3-3 brings the matrix for and shows one department and the interfaces that that department has. In green is a new department that performs the function of developing new technologies and preparing them for implementation. This group is called Advanced Transmission. Another new group, Software Engineering is shown in pink. They are both discussed in greater detail in Chapter 4. In yellow is the functional or component side of NA OEM. In white is the program side and finally, in blue is the current systems engineering function.

The status at NA OEM, however, has long been very strong program teams with weaker technological teams. Each transmission, for the better part of 30 years at NA OEM, made fairly independent decisions about technology from the other ones. Some transmissions used simple planetary gearsets, others use a Ravenaux gearset, and still others use a Simpson gearset. The story is the same on the controls side and in pump architectures. The only thing that could be said for the transmission architectures over from the 1980s until today is that each one was going to be different from the last one and that the decisions were made in most cases by program managers, with or without input from functional managers.
Allen argues that program driven cultures are responsive to market shifts, while technology driven cultures either drive towards lower cost by commonizing architectures and driving up reuse or attempt to implement the latest and greatest technology available, depending on the rate of change of technology in that industry. NA OEM practices the first case. While it is true that technologists also like to incorporate the latest and greatest technology, this is only the case when the technology is quickly evolving. In summary, a program focused culture should lead the company when the market is pulling technology and forcing change. A functionally driven culture should lead when the technology changes rapidly and it is important to stay at the tip of the technology spear or when the technology changes very slowly and it is important to drive commonality through architectures or platforms across multiple programs.

A careful look at the automotive transmission business reveals very little and very slow shift on the demand side (market pull). Transmissions are very much unchanged in basic design since the 1960s. The addition of more gears and electronic controls has occurred over a period of decades and is very much evolutionary, rather than revolutionary. Additionally, the transmission is not a strong differentiator in the marketplace. Consumer's buying decisions about vehicles are rarely based on the transmission features and attributes. Lastly, a heavy truck and small car
transmission are arranged, scaled and packaged very differently, but the underlying functions are nearly identical. They use the same types of gears and power flow, the same controls, the same types of torque converts, etc. There is little about a heavy truck customer that a transmission engineer for that customer segment needs to know that is different from the needs of a small car transmission engineer.

Since the market is slow to change, the program managers should tend to adopt proven and existing technologies and therefore, the functional managers, who are more focused on the reuse and commonality of existing technology, should be firmly in control. It turns out to be the opposite in this case. Program managers at NA OEM have assumed a much more powerful and influential place in the organization than their component manager peers. They rank higher through their reporting structure and control more of the budget. Functional managers at NA OEM are focused on efficiency and reuse, platforms and architectures, rather than on implanting the latest and greatest technologies available as is natural given the slow change in technology. They have filled the role of managing costs, perhaps due to the pressures that are placed on them to reduce and manage costs after production. Program managers have tended to push for new and revolutionary designs. A possible explanation is that it is the nature of engineers to drive towards change; they want inherently to put their stamp on a new design. While program managers are behaving like engineers, driving new technology, functional managers are driving for reuse and commonality; in essence, they are not behaving like engineers. Patents, publications and peer recognition are more important to engineers than other types of incentives, including financial and promotional incentives. Perhaps this fact keeps the program managers from listening to the market, where the customer is saying, “We are happy with the technology in the transmission, just refine it and don’t fix what is not broken.” Instead they force un-necessary adoption of new technologies, despite the risk that such changes make towards meeting the cost and timing goals. And why do the functional managers choose to leverage existing architectures, maybe due to the fact that they are being stretched very thin due to resource constraints and feel like reuse and platforms will enable them to stretch their resources as far as possible. Recall from the example in the beginning about the team being scavenged from other programs. Since the functional arm is matrixed and is required to support the entire organization, when a program is in turmoil, it is easier to dedicate a resource who is already matrixed than one who is dedicated
to a separate program. The functional manager can easily assign more of his people to program A while program B is quiet. The process to reassign a dedicated resource from program B to program A requires manager B to give consent. So the stretch within the organization to accommodate launch pressures is felt more by the functional side.

So in addition to the primary goal of delivering transmissions to the vehicle programs, the next main goal of a transmission design organization is to reduce the cost to the organization. In an industry with such razor thin profit margins and long product life cycles, cost decisions are critical to business success and continue to apply pressure for years (in some cases decades) into the future. Customers are inherently happy with the technology and are not asking for innovation, but instead want incremental improvement driving towards perfection. Since they are inherently satisfied with the performance of their products, the goal of the organization then should be to maintain (or slowly improve) the level of functional, while reducing cost. As for schedule, the market does not require frequent changes in transmission technologies and so, schedule pressure should be lesser. Based on this recognition, it is important to organize the PDP to be driven by platform architectures designed as systems, and thus to be primarily in the hands of a new functional manager responsible for systems optimization. The organization is almost arranged, it is simply rebalancing the power away from program managers and towards functional managers where global costs can be managed better. A new, systems level functional manager position should be created to be responsible for the new entity. The power of this position needs to be emphasized within the organization through reporting structure and budget control.

**Chapter Summary**

In this chapter, culture of the NA OEM team has been explored. The culture shows inherent conflict between the program managers and the functional managers, and has traditionally leaned in favor of the product side, despite a lack of pull from the marketplace. It has also been shown that no one is managing cost of the system. Functional managers may be optimizing the cost of their own technologies, but the system costs are neglected. Decisions are made regarding technology choices to satisfy a perceived need to incorporate new technologies without a legitimate pull from the marketplace, and these changes drive significant cost to the product in
terms of people required to implement a design and the investment required to produce that design. These costs are carried for years due to the long product life cycles. In the next chapter, the Product Development Process (PDP) is explained. More detail will emerge regarding the organizational structure and where information comes from, when it is delivered and who is responsible for the system. At the end of chapter four, a need for a new organizational structure will be clear that must better align with the first goal of the work, addressing the affect that organizational structure and corporate culture have on global optimization. The need for a revised PDP builds on goal number two, to institutionalize the process and make it repeatable. Both of these will emerge as suggestions for change later in this work.
Chapter 4 The Product Development Process

The product development process at NA OEM and many PD companies is considered proprietary information and is kept very confidential, so the explanation of the product development process (PDP) at NA OEM will be explained only generally. After reviewing the PDP from a high level, several aspects of it are considered more specifically, including the handoff process and how it compares with competing companies. Also considered are the organizational aspects of program managers vs. functional managers and their differing objectives. This chapter builds on what was discussed regarding the organization in Chapter 3. Here, a new aspect of the organizational structure emerges, and is coupled with the PDP itself. Discussed here in Chapter 4 is the question of where do ideas come from and what is the design space of the problem. The PDP and the organization both limit the design space and hence, constrain what the Systems Engineer can do.

The PDP high level overview

The PDP begins with a new vehicle decision, whether it is brand new or based off of an existing platform. Once that decision has been made, the information is cascaded from vehicle to powertrain in a Program Direction Letter (PDL) which is a high level functional cascade. From the PDL, a Powertrain Program Planner would make selections about which transmission, engine and driveline would meet the program needs. These attributes are things like fuel economy, acceleration, gradeability (in what gear and at what weight can the vehicle climb a given incline) and towing capacity. This occurs prior to the program actually being studied, during a pre-kickoff exercise. Once the program is kicked-off, at a checkpoint named Program Start <PS>, it proceeds through several milestones and checkpoints, where assumptions are reviewed and the management team can review the financial assessment that corresponds to the assumptions about content.

While the vehicle and powertrain planning teams are analyzing the vehicle attributes and assessing potential powertrain packs to use, another team is working on developing new technologies to integrate into the powertrain selection set. These advanced engineering teams are in between true research and development (R&D) and the teams responsible for delivery of
the production ready design (Forward Model). Advanced Transmission is the group responsible for this at NA OEM. This team is working on developing new transmissions to replace existing models. They analyze the market requirements and the competitive landscape and attempt to mix ideas that are coming out of research with known ideas to present a technology that is Implementation Ready (IR). During this process, existing vehicles are used as surrogates to assess the performance of a handful of transmission prototypes.

At the <PS> milestone, there is a handoff of technology from the Advanced Transmission group to the Forward Model team. This team then readies the design for production. This is a critical handoff, but very little discipline has occurred during this transition in the past. These were handled so poorly in fact, that several new designs were purchased through partnerships outside of NA OEM, rather than through the Advanced Transmission channel. The future of this division was on everyone’s mind for quite some time, but recently, new designs have again emerged from the Advanced team.

Once a program has moved through a series of checkpoints within the Forward Model team, it reaches the final decision point, named Program Approval <PA>. At the <PA> milestone, a firm decision is made about whether to proceed and to fund the new program. Following <PA>, the powertrain Forward Model team is responsible for development of functional prototypes and to support the vehicle design team with engines and transmissions that meet some functional criteria so that the remaining teams can proceed with their work using functional powertrains. At the same time, the powertrain team refines the engine and transmission along the remaining stages.

Finally, the vehicle approaches Job #1 <J1> or the production milestone. At this point there is a launch team responsible for final tweaks to the design to make it manufacturable and to fix any details that have not been properly addressed earlier. For many areas of the company this stage represents another handoff, but the transmission development team does not have separate launch teams like many other groups. The Forward Model development team carries the design through production.
**Advanced engineering**

The Advanced transmission group is charged with delivering Implementation Ready (IR) technology to the Forward Model engineering team. There are four major weaknesses inherent with this process.

- Advanced Engineering's definition of IR is insufficient compared to Forward Model's definition of IR.
- Architectural decisions made by Advanced define the lifetime program cost and schedule without regard for global optimization as defined earlier.
- Advanced team is not aware of the implementation difficulties and the lessons learned by the Forward Model team and they are unable to incorporate these lessons into future designs and architectures.
- Advanced is chartered with delivering whole transmission designs, not subsystems and therefore, there is little opportunity to incorporate new technology separate from a major transmission program. Technology here could be improvements in components that lead to better system level performance or it could mean a whole new system, when such breakthrough technologies present themselves.

A technology is said to be ready to go by the Advanced team, but the Forward model team then inherits the design and has to do further development to prepare it for implementation. As shown in Figure 4-1, the readiness for IR is some % of production readiness. The trouble with this process is that the advanced group is often not aware of the needs of the beneficiaries. They are entirely an engineering organization, and as such, without project management responsibility, they are focused mainly on function. With that said, however, they are still detached from the functional requirements and lessons learned by the mainstream PD teams. Recently, an Advanced engineer commented that he had not looked at a calibration data trace for years. This points out the relative isolation that Advanced has from PD and the lack of understanding. In the next section, a comparison will be made to how to integrate ideas and concepts better with the PD team, the program manager's needs and the needs of the customer and business. This process is called System Architecture and is based on concepts learned in SDM.
When a design comes from Advanced, there is often a handoff that requires rework of requirements, functional specifications, packaging, cost, or many other needs. This rework takes the PD team months to complete, and in some cases years. Based on recent experiences, the process usually completes near the PA milestone. At PA, the actual readiness is where it was expected to be at the PS milestone. This leaves very little time to solidify the design and integrate it into the larger systems before production. The staffing peaks, shown in Figure 2-4, are often due to this lag. The program is usually unaware of the delays and the impact that they cause in terms of system dynamics (cost overruns, program staffing and prototype budgets, late changes to manufacturing processes, inefficiency in design, etc). The process of moving a design from functional to eloquent takes time, but requires the design to keep moving. The early phases are typically the most important, because these establish the architecture that the design is based off of.

![Graph showing implementation readiness](image)

**Figure 4-1**: Implementation Readiness is a measurement of the progress of a design towards production readiness. Many times the handoff process results in lost information or a misalignment of the definition of IR and rework must be done.

**Software Engineering**

Another group exists which provides significant design content but which is also separated organizationally. The same types of hindrances exist between the Software Engineering community and the Forward Model Engineering (hardware) community as exist between the Advanced Engineering and Forward Model Engineering. In this case, the issues arise because the software team has its own objectives that it is trying to optimize and sometimes these
objectives are in conflict with the goals of the larger system. Software engineers are measured by things like lines of code, reuse of software modules and number of bugs detected and fixed. In an effort to reduce the code, drive as much common modularity and content as possible and minimize rework and bug fixing, they freeze content very early in the design process. Because they are an independent group in the organization, they also have the ability to push back on the hardware design and force the hardware to change in lieu of the software changing to solve a given issue. This will be highlighted in an example in Chapter 6 regarding the design space.

Figure 4-2: Advanced Engineering and Software Engineering provide significant content and influence the design space of new transmission and subsystem technologies, but are removed organizationally. This separation creates conflict and discontinuity of the design process.

The systems engineer needs to have responsibility for managing the interfaces between these two critical but separated divisions of the company. The organizational structure is creating rifts that will have long standing affects in terms of the speed of delivery and the level of global optimization and integration of the design that will ultimately be achieved. Still, that organizational structure has some purpose. Software engineering would suffer from the same fate of technology development, if left to be managed by program managers; they would feel little need to adopt software designs and codes that were shared across multiple platforms. The difference in this case, however, is that the functional arm of Software Engineering has a more dominant role than do technology managers. Rather than promoting individuality, like the program managers enjoy in Forward Model engineering, the structure promotes conformity, even if the software and hardware are a poor fit. The result is far from global optimization.
Chapter Summary

This chapter introduces two separate divisions within NA OEM that play another significant role in the development of the transmission. These two separate divisions influence the hydraulic control system in addition to the program and component managers discussed in Chapter 3. The four different players are all highlighted in Table 4-1. This table outlines who the players are and how they all have conflicting objectives that work against global optimization.

Advanced Transmission and Software Engineering are neither well integrated into the PDP and the result is a continuation of local optima. Advanced engineering optimizes whole transmission concepts, most of whom do not make it to production, and those who do, are significantly redesigned by the Forward Model team during time in the PDP that does not exist. This forces schedule pressure on the Forward Model team, most of which is driven by the Program Managers. Software engineering had developed a very powerful lobby and forces conformity. This approach optimizes the reuse of code and efficient use of people within that division, but at the expense of global optimization of the system. The next chapter begins to lay out a new approach, derived by practices learned at MIT and developed on-the-fly by a new team of Systems Engineers. It will be shown throughout this work how the role of a Systems Engineer must be broad enough to encompass all these functions so that optimization of the system can take precedence over optimization of components, technology or code.
<table>
<thead>
<tr>
<th>Who</th>
<th>What is their focus</th>
<th>What is the outcome of this focus</th>
</tr>
</thead>
</table>
| Program Managers    | • Design Whole Transmissions.  
                    • Incorporate new technology.  
                    • Respond to the needs of the vehicle program, especially regarding schedule. | • They begin focused on a new and exciting transmission that will feature new technology.  
                                                                                                  • Programs end consuming many additional resources to make up for the poor handoff from Advanced (or other source of design) and everyone scrambles to meet schedule. |
| Component Managers  | • Design Components.  
                    • Deliver on functional targets.  
                    • Meet production dates.  
                    • Reduce cost.  
                    • Maximize utilization of their people. | • Push platforms and architectures that enable them to stretch their resources as far as possible but with a component mindset.  
                                                                                                  • Optimize the performance and cost of the components under their influence.  
                                                                                                  • Near production, many additional resources are scavenged from component groups.  
                                                                                                  • After production, the focus shifts towards reducing component cost. |
| Advanced Transmission | • Innovation and the adoption of new technology.  
                          • Providing an implementation ready (IR) design. | • Technology is pushed regardless of market pull.  
                                                                                                  • Time is spent on new designs but only on whole transmissions. No subsystem designs are handed off. This forces a new subsystem design to be incorporated into a new transmission design and forces program timing on subsystem development. |
| Software Engineering | • Developing software solutions for powertrains.  
                          • Maximizing the utilization of code across multiple programs.  
                          • Maximizing utilization of their people. | • Software dictates hardware requirements  
                                                                                                  • No opportunity to modify code to suit a program, since the same code is used in other programs. There is push to constrain the code and force the hardware to adapt.  
                                                                                                  • Optimization of the software, minimize people, maximize reuse. |

Table 4-1: There are four major parties involved in the development of a hydraulic control system and each as different motivations which lead to local optimization.
Chapter 5 Creating a system that meets the needs of its beneficiaries through System Architecture

A new approach exists, however, to connect the architecture to the beneficiaries and to properly map out and characterize their needs. This process allows the Advanced engineers to understand and to identify with the system level performance attributes discussed earlier. It will be shown where those attributes needs arise from and how the system can be architected to meet those needs. In the final chapter, which ties all the suggested changes together, a new modified PDP and organizational chart will be shown where a Systems Engineer leads this process in place of Advanced Transmission. It is important to have the same people who solve problems and put designs into production participate in the process of defining the architecture of the system. This knowledge was key to the team who was put in place as described in Chapter 1, during their time fighting fires in the heavy truck transmission program, lessons were learned that were then carried forward to the development of the new transmission. Although the approach defined here is more for illustrative purposes regarding how to think and what to think about, the framework of the architecture defined is relatively comprehensive. For readers from the transmission design community, it provides a great starting point from which to build a complete picture of the needs of the system.

System Architecture: Understanding the Beneficiaries of the System

One of the most important steps in system architecture is to identify the beneficiaries of the system and to understand their underlying needs and motivations. Without spending time focused on who the customers are, and customers are plural, an architect cannot layout a plan to meet their needs. The plural is very important, because at first glance, only one customer is usually addressed, the end consumer of the product. The consumer is a very important element in the design of the product, but so many others must be addressed. It is critically important to spend time on this in a disciplined way. Good engineers may intuitively see this map, but most do not, and even those who have intuition will leave off key beneficiaries and therefore will not design the most effective solution. This is one of the most important steps in moving design from being merely effective to being truly eloquent. An eloquent system uses few resources to
implement, to design, to manufacture, is sustainable, is accepted in different regions, pleases its users, delivers outstanding function and is affordable. It is so important to recognize that this does not happen by accident. It must be recognized in advance, and be part of the design process or critical attributes will not be met.

The high level beneficiary map presents fundamental goals for the control system as described below. The first beneficiary is the consumer, or in this case, the driver of the vehicle. The driver demands a smooth, fast, responsive and quiet shift event out of the hydraulic control system. They also demand reliability and robustness to wear, time, temperature, environment and some robustness to driving habits and usage. This is more detailed in the section on robust design principles in Chapter 6. The driver is the most obvious beneficiary.

Next, the systems engineer's needs must be examined. The system engineer is primarily concerned with the technical function of the system. Shift times, measurements of smoothness and quietness are key output variables, and a majority of the technical component to the job is spent understanding interactions and noise factors of the components and environment. It becomes important for the systems engineer to have good measurement capability and to be able to quickly tune or manipulate the design. This becomes a key attribute of an elegant design, to be tunable. Designing in features which allow the design to be easily measured without disturbing the basic system is very important. It is also important to have a ready supply of parts that can be used to turn knobs which may influence the design. These tunable features include pressure and flow rates of solenoids, pump capacity, orifice sizes, valve metering notch sizes and so forth. One of the most important understandings that come through experience with a given design is knowing which characteristics matter most and what direction each of those characteristics will take the system.

This development process also influences the design. There have been examples which are very good and bad with respect to this feature. Some designs actually have features embedded in them to allow the development engineer to quickly access pressure signals for measurement, and others require special adapter plates or elaborate routing to access that information. This makes a significant difference in the ability to meet schedule while negatively affecting prototype cost.
Leaving out measurement capability is an example of local optima, the managers ask if these features are really necessary, and an inexperienced engineer usually answers that it's nice, but not required. The program manager saves this cost, but 99% of the time, this decision will come back to haunt the systems engineer later. In traditional cases, however, the advanced engineer or manager has long since handed the design off without this capability and the Forward Model engineer is left trying to patch together a fix. In the end, it is nearly always fixed, through expensive measurement tools, but inevitably this costs development time and money.

Cost and timing are perhaps where the most tradeoff occurs. It has already been mentioned that programs run over budget and drain resources from the organization to come in on time. Addressing this tradition will be the system engineer's greatest contribution to the process, in addition to releasing a better integrated functional product.

The manufacturing engineer is responsible for laying out the production facility. He/she is concerned with cycle time, machine serviceability and maintenance schedules, throughput, quality control, assembly ergonomics. The fields of design-for-assembly and design-for-manufacturability revolve around the needs of this beneficiary. A good design must be manufacturable, and an eloquent design must be easy, cheap, flexible and fast to manufacture. It will fit within the manufacturing environment and will pass through the process smoothly, with low rework percentages.

Another obvious beneficiary is the program manager who is responsible for the execution of the transmission itself, and needs to find a control system that is deliverable on time, on cost and within staffing levels. Recall that that the iron triangle cannot be broken.

Above the program manager, are several layers of management ending at the powertrain vice president. Here the needs expand again. United Auto Workers (UAW) contracts and workforce distribution are issues that need to be addressed. The VP may not be able to allow a control system that is manufactured in another country due to trade agreements and may have to balance plant capacity across multiple locations that could drive the architecture in a certain way.
<table>
<thead>
<tr>
<th>Beneficiary</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmission</td>
<td>Shift from Gear A to Gear B</td>
</tr>
<tr>
<td>Driver</td>
<td>Smooth shift event</td>
</tr>
<tr>
<td></td>
<td>Fast shift event (targeted shift duration of 500 msec.)</td>
</tr>
<tr>
<td></td>
<td>Responsive (min. delay from throttle tip-in)</td>
</tr>
<tr>
<td></td>
<td>Quiet shift event</td>
</tr>
<tr>
<td>Control systems engineer</td>
<td>Smooth implementation</td>
</tr>
<tr>
<td></td>
<td>Tunable</td>
</tr>
<tr>
<td></td>
<td>Controllability</td>
</tr>
<tr>
<td>Manufacturing engineer</td>
<td>Easy to install</td>
</tr>
<tr>
<td></td>
<td>Reliability</td>
</tr>
<tr>
<td></td>
<td>Repeatability</td>
</tr>
<tr>
<td>Service technician</td>
<td>Serviceability</td>
</tr>
<tr>
<td>Product Manager</td>
<td>On-time delivery</td>
</tr>
<tr>
<td></td>
<td>Cost target</td>
</tr>
<tr>
<td></td>
<td>Within design specification (scope)</td>
</tr>
<tr>
<td></td>
<td>Manage headcount</td>
</tr>
<tr>
<td>Powertrain Vice President</td>
<td>Plant capacity considerations</td>
</tr>
<tr>
<td></td>
<td>Capital equipment reuse</td>
</tr>
<tr>
<td></td>
<td>Workforce distribution</td>
</tr>
<tr>
<td></td>
<td>NAFTA / other trade agreements</td>
</tr>
</tbody>
</table>

Table 5-1: Beneficiaries and their needs. Many beneficiaries exist beyond the obvious and it is important for a system architect to spend time considering all the layers who will be affected by their decisions.

The analysis of the beneficiaries allows the architecture to be framed using the Object Process Methodology (OPM). An OPM diagram of the hydraulic controls system is shown in Figure 5-1. This diagram allows a quick glance of the system in terms of its primary function. The intent of the system is captured in this diagram. The flaw that most engineers make is to think about the function of the system in terms of its form, or physical layout. When the view of form (the physical arrangement of the system) is intertwined with function (the underlying purpose), the architect is severely limited in creativity. For example, consider a pump. If you are consumed with the gears in the pump and the casting, the shape of the inlet and outlet, those are all elements of form. It is easy to be so absorbed in these that the purpose, to create fluid flow, is lost. It becomes hard to see a pump in any other way than the way it is, but there are many different ways to create fluid flow. This perspective forces the architect to consider options for solving the problem that already exist. The OPM process allows the function to be explicitly
defined, based on the high level system intent. This allows the architect to choose many solutions to this open ended problem statement without the constraints implicit in the current form of the solution.

Separating form and function cannot be stated strongly enough. It is at the core of many engineering difficulties. A wise old golf pro once said that the match is won or lost on the first tee. This is perplexing at first, did he mean that after the first shot, you could tell who was going to win or lose? While it seems ridiculous to be able to conclude how a match is going to be played out based on one shot, the pro made the story even more unbelievable by telling the young golfer that the match was won or lost before the first shot was even made. You see, on the first tee, the match is negotiated, strokes are given or taken and once the match has been setup properly, only a very great or a very poor execution could change the fate of the match. Similarly, by separating form and function, the engineer is able to see the problem separated from the solution, and then can setup a solution that will succeed. Proper setup of the architecture accounts for a very large percentage of the success of the end product. Improper setup of the architecture embeds costs, misses targets, fails to address needs, the list goes on.

Figure 5-1: Example OPM diagram for a hydraulic control system. The power in the OPM is its ability to capture intent of the architect, not just the form that the system takes on.

In Figure 5-1, the needs of the beneficiary are mapped to the intent of the system, to shift smoothly, quietly and quickly. The diagram then maps the intent to the highest level process that accomplishes the intent and then dives deeper into different forms that can accomplish this
function. This process is shown in Figure 5-2. The specific form that is chosen in this example is transitioning a clutch. It can be accomplished by pushing or pulling, which can then be done by a multitude of devices. The green boxes represent the architecture of the hydraulic control system that is dominant in the automatic transmission community, including NA OEM.

Figure 5-2: Intent is mapped to the process selected for accomplishing the intent. Then the specifics of form are free to flow out of the higher level needs.

The form of the solution now represents a conscious choice that was arrived upon by careful selection based on a clear understanding of the needs of all the beneficiaries of the system. Other choices exist, but were ruled out. Up to this point, the example has been about what the needs look like and what the intent of the system is. Now let’s take a few pages to leave the reader a tool that can be used to explore additional design possibilities. There are several different models for innovation. The Pugh Method\textsuperscript{vi} is a very powerful one that can be used. Another popular tool is to use TRIZ, a technique developed by Russian engineer Enrich Altshuller. Much has been written about the Pugh Method and TRIZ. Another simple tool was used in exploring possible choices for the development of this control system. It is called the combinatorial concept selection method and is explained in detail below. A huge advantage of the combinatorial model is that it is very simple to execute and provides designers a new way of framing an old problem without the time and involvement of more sophisticated tools.
Combinatorial Concept Selection Method

Three different concepts are identified using the combinatorial matrix in Figure 5-3. The process allows the architect, in this case an advanced transmission engineer, to proceed through available options that meet the high level goal statement. Further analysis of each path compared with the other beneficiary needs is also provided.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Dedicated Transmission Control Unit with controller</th>
<th>Dedicated TCU with feedback controlled driver</th>
<th>Modular Powertrain Control Module with open loop driver</th>
<th>Modular PCM with feedback controlled driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signal Device</td>
<td>Direct Acting Solenoid</td>
<td>Two Stage Bleed Type Solenoid</td>
<td>Low Leak Variable Bleed Solenoid</td>
<td>Magnet</td>
</tr>
<tr>
<td>Amplifier</td>
<td>None</td>
<td>Regulator Valve</td>
<td>Regulator Valve with Latch Valve</td>
<td>Regulator Valve with Latch Ball</td>
</tr>
<tr>
<td>Source</td>
<td>Gerotor Pump</td>
<td>Variable Displacement Pump</td>
<td>Crescent Pump</td>
<td>Piston Pump</td>
</tr>
</tbody>
</table>

Figure 5-3: Combinatorial matrices can be used to discover new combinations and to piece together new architectures.

The first example of this type of architecture is the Direct Electronic Shift Control architecture that was used as the example in the introduction. Although it was a very rocky implementation at NA OEM, it is re-emerging in transmissions being brought to market by leading transmission suppliers. The idea here is that you flow oil at one clutch from one control element (no multiplexing). The concept narrows from here and used high force, high flow direct acting solenoids to control the oil at the clutch.

The next concept is fast becoming the standard in the industry. There is still significant debate about what type of controller and what power source are most robust and cost effective, but the low leak variable bleed type solenoid coupled with a regulator valve and a latch valve have become nearly universal among designs from GM, Ford, ZF, Aisin Warner, Jatco, Mercedes
Benz and others. There are still variants that use direct acting solenoids for precision and oil management. This design standard meets many of the intents of the beneficiaries, including robust implementation (low contamination sensitivity, high controllability, robust manufacturing tolerances) with a strong business case (low cost commodity sourcing, standardized designs for to avoid sole sourcing, flexible manufacturing, etc.). The benefits of the design stem from collaboration amongst multiple manufacturers. This collaboration resulted in a standard design that the OEM's could then source to several suppliers. This strategy led to lower piece cost, less investment, shared learning and shorter development times (for the followers). But did the concept select itself? No, the concept was chosen for other reasons. This concept is very frugal with oil and allows the use of smaller pumps with lower mechanical losses, boosting AT efficiencies into the mid 90%'s. This was unheard of until this concept presented itself and was one of the selling points of manual transmissions. This also makes engineering the system more important. In this case, the oil management functions (lube and cooling) and the shift quality functions are coupled. This coupling will be investigated further in Chapter 6.

The final concept is a choice that has been revealed through the concept selection process. When the shifting of the transmission was generalized and not thought about in terms of form, a new process emerged. Still using a clutch and the same base mechanical architecture (to maintain commonality with other designs) a new design was imagined using a high powered magnet to engage the clutch and thus to change the ratio. Some aspects of the goal statement that this design would benefit are:

- Very low mechanical losses (could potentially approach 99% and rival manual transmissions), especially if end clearance of the clutches were large, thus reducing drag losses.
- Quiet (no fluid flow disturbances – no pump noise concerns)
- Commonality – same clutch and mechanical architecture of transmission. Same power flow schematic for torque transfer.

Some cons of this concept are:

- Service – magnets may need to be integrated into the barrel of the transmission and would not be field serviceable.
• Engineering Support Staff – Current staff is not capable of designing magnets. New expertise would have to be developed. Given enough development time that is decoupled from program timing, it is possible that this technology could be implemented with a support staff of three engineers and could still meet all program milestones.
• Quality – Until tested, it is not possible to determine the applicability of the concept and it's robustness to manufacturing and external noise factors.

Although it is not proposed as a solution, this example illustrates the process that the Systems Engineer acting in the capacity as a System Architect can go through in examining architectures. Later in this work, more discussion will unfold about revolutionary vs. evolutionary design, and although this concept selection method seems more applicable in revolutionary design, the principles are applicable in evolutionary designs also. Even in evolutionary designs, the important step of connecting the architecture to the needs of the beneficiaries must still be completed for each design change proposal. This makes these steps critical for all design changes, not just within the advanced community, but within the mainstream PD community too. If these tools were utilized, then the handoff would be managed so much better than it has been in the past. The new team would understand the architecture selection process, the intent, who the beneficiaries considered were and what their needs were as they were examined by the advanced team. Then WHEN rework needed to be performed, the PD team wouldn’t have to begin anew, they would have a much stronger foundation from which to build the design because they would know why, not just what the design is.

**Needs of the System**

This analysis of the beneficiaries and the OPM diagram approach of defining the system's intent yield the system goal statement. The goal statement contains the intent of the system in specific and measurable terms. Then the goal statement categorizes needs according to their relative importance. These attributes are similar to the ones mentioned in Chapter 2. The more attributes that are accounted for in the system goal statement, and the more beneficiaries whose needs are addressed, the more robust and desirable the architecture is. A statement of needs of the system is shown below, with proprietary details omitted.
Provide users a system
to shift (change ratio) the transmission smoothly (TDV Metric), quietly (dB metric) and with durations < X msec and delays < X msec.
While also:
Critically:

- providing low mechanical losses (< X hp)
- high resolution of shift event profiles (< X psi resolution / X mAmp of current command change)
- quiet (less than XX dB measured at driver's ear using Achen head)
- manufactured at site XYZ and using a labor pool of X people

Importantly:

- maintaining commonality with other control systems in the company portfolio (X% reuse of components – X% common schematic)
- costing less than $X per unit with $X in capital equipment invested
- installable using X operator stations in less than X second tac time
- launching with less than X defects per thousand on end of line testing
- with less than X repair per thousand and X Things Gone Wrong (TGW) per thousand
- meeting all milestone delivery targets with a staff of X engineers

Desirably:

- being serviceable without full transmission removal from the vehicle.

Meeting the needs of the beneficiaries

Basic function emerges as the mainstay of a system, which in this case is to make smooth, quiet and quick shifts. Beyond this, however, a plethora of other needs appear. This is why it is so important to think carefully and methodically about a design. At first glance, which it appears, is all that many designers spend, you can easily make a functional product. But beyond this is where a design emerges as eloquent and beautiful. The Apple Ipod is a wonderful example of a product that met customer’s basic need and then went beyond, to the next level and really created itself as a cultural icon. It plays music, sure, but it is stylish, intuitive and benefits all the beneficiaries on many levels.
In the same way, a transmission or any other product can emerge from basic function to elegant design. In the case of an automobile, there is obvious artistic and cultural charm. Also, though are recycleability, energy footprint, package space, amenities, ergonomics, and political impact. Hybrid vehicles are functional in that they get better fuel economy than most competing cars. This is not absolute, however. It is very possible to get the same fuel economy from a diesel engine, due to its higher torque output, it can be made smaller and with the addition of particulate traps, it can be made clean. Still, a diesel engine is not as elegant an architecture because it lacks something in the eyes of the consumer. It may meet the end user's functional needs, but it does not address all the other beneficiaries in the same way. Hybrids have become political and fashion statements, in addition to their ecological benefits. This is how elegance in architectures is achieved. If the designer properly maps the needs of the beneficiaries, and is able to account for more beneficiaries, to see beyond the obvious ones, then the design has a better chance of transforming itself from survival in the marketplace to significance in the lives of consumers.

In the transmission hydraulic control system illustrated throughout this work, however, some needs are consistently being met while others are for the most part ignored. As the needs are discussed, it is important for the second type of readers, other control system engineers, to note which needs are being addressed in future control system designs and which ones are not and require more attention. The more needs that are met, the more eloquent the design and the longer it will last.

**Critical Characteristics**

The critical characteristics of the system are obvious, even without the detailed beneficiary needs work being done. Shift times are one of the most important characteristics of the system. Time can be broken down into two attributes, first is the shift delay, which is measured as the time from electronic initiation of the event to the time when the torque begins to transfer from the off-going gear to the oncoming gear. The next time is the shift duration, which is measured as the time for the ratio to complete the transition. These have been established using jury analysis and by spending a lot of time interviewing and studying customers response to different shift characteristics. There is such a thing as a shift that it is too short, and it should not be the goal to
make the shift occur as fast as possible. This typically results in a rough and jerky shift but even when it is smooth, the customer perceives it as something wrong and is dissatisfied. Delay is important only for certain maneuvers. In many cases, the delay is imperceptible, because the computer initiates it without any input from the driver, so the driver does not know when the shift was told to begin, only when it does begin. In other shifts, the delay is critical, and faster is often better, as in the case of a throttle-on downshift – the driver needs to accelerate, so they tip into the throttle. In this case, the driver has initiated a shift by tipping in, so the need for a fast response is high.

In addition to shift times, shift feel is also a critical characteristic of the system. Feel is measured in two ways, first using a torque meter attached to the driveshaft of the transmission and second by an accelerometer mounted on the driver’s side seat track. The torque disturbance is measured and provides a metric for shift feel. Customers have been surveyed and studied in controlled situations, and it is not only important what the torque disturbance is, but at what frequencies it occurs. To account for this, there are sophisticated methods of breaking torque readings into a measurement called TDV – torque dose value, which filters the raw torque signal for certain frequency ranges that are more discernable to the driver. The TDV then transfers through the powertrain mounts and into the vehicle structure to the seat of the driver. The seat track vibration is measured and processed as VDV – vibration dose value (which includes the vehicle transfer function). The VDV is again processed and filtered to certain frequency ranges. TDV and VDV are the metrics that define the feel of a shift. When delay and duration are added to the TDV and VDV measurements a proper analysis of the basic function of the hydraulic control system can be made. These needs are basic, but critical. The calibration community and program managers will scrutinize these fundamental aspects of the system. For the most part, it is very difficult to ignore these needs, as this is what is most often measured on performance reviews and during team meetings. The culture at NA OEM places heavy emphasis on the Calibration activity, and calibrators, as was already mentioned, are the main beneficiaries of these attributes.

The final basic function of the hydraulic control system is to provide some cooling and lubrication to the transmission. This function is a simple bypass flow from main line pressure
that passes out to the cooler and back into the transmission where it passes through the lube circuit and finally ends up in the sump of the transmission, waiting to be recirculated into the hydraulic control system.

**Important Characteristics**

Moving forward from basic function come the plethora of other needs that the system must meet. This is where an architecture can evolve past adequate to become something more. This is also where designs rise and fall. The remainder of the attributes from Chapter 2 will fall into this category. As has been shown throughout this work, this is where the design really matures. It also represents the major emphasis of a systems engineer, both within Advanced and Forward Model engineering. So what are some of the attributes that a good systems engineer, designing an eloquent system will address?

These secondary characteristics include management of cost and resources. Delivery on time, on budget, within resource allotments, are all important characteristics and unfortunately, most system engineers do not feel responsible for delivery of these attributes and are not managing the design to meet these needs. In this way, more than any other, the role of a Systems Engineer in Advanced and Forward Model must change because it is here that most designs fail. Customer satisfaction ratings show that nearly all the major auto makers are in a statistical dead heat with each other in terms of the quality perception of the end consumer. Customer satisfaction ratings appear around the 80% range for nearly all major manufactures. Where then is a design to separate itself. Since transmissions are not distinguishable in the marketplace, as was already mentioned, the focus of an elegant design should be on internal metrics that influence global optimization. If the systems engineer looks down the list of beneficiaries, and further into the system needs statement, then things like development cost, resources and delivery time need to be highly considered. Also are labor relationships and contracts with the UAW. Manufacturing processes and capabilities are high on the list of features that can make a design eloquent. Environmental impact, not only of the complete vehicle, but of the processes and parts that went into the vehicle should be considered. Political influences can be important. These things are not taught in engineering school and are unfortunately not considered regularly by engineers. In many cases, some of these considerations will emerge as part of the management routine around
the <PA> milestone. A thorough business case is developed leading up to <PA> and many designs die at that phase. Several transmission ideas have been shelved in favor of carryover content. It’s amazing to see a design with poor performance continue in the marketplace while new, potentially better designs are shelved. The answer to why this happens is often expressed in the new design not meeting these important considerations. When the focus is on technology or functional performance the business is not inherently satisfied. Good designs, with proper attention to these factors early in the architecture stage will pass <PA> and go well beyond; meeting many beneficiaries needs for years to come.

**Chapter Summary**

In this chapter, a new process for laying out the architecture of a control system has been established. The process starts with mapping the beneficiaries, who extend within, but also go well beyond the organization. This chapter outlines a new process and offers several new tools to be used by the systems engineer which will allow designs to thrive. The more needs that are met and the better the system is at meeting those needs, especially the latent ones, the more successful the system will be. Tools introduced include the combinatorial selection model for innovation. The Pugh method was mentioned but much documentation is already available regarding that technique. The OPM diagram leads to the list of beneficiaries. Finally, the system goal statement captures all the above factors in a concise list. The list presented here, as mentioned earlier is a good starting point, but future systems engineers could build on it and refine it even more. The list can never be too comprehensive, for the clearer the picture of the end is, the more likely the team is to stay on track.

Also mentioned were considerations of how the organizational structure affects the design. Because of the gap between Forward Model, where most system engineers reside, and Advanced Transmission, where most architectures originate, the most important beneficiary of the system the system engineer, is not core to the team that defines the architecture. This is a recipe for disaster and must be addressed within the organizational structure before any of the other suggestions can take hold.
Chapter 6 Moving from a component to a systems level focus through Robustness Checklists and Design Structure Matrices (DSMs)

In Chapter 5, a new tool was introduced to define the needs of the system and to extract a higher level of value to the organization. The system has beneficiaries that include managers, executives, cross functional team members and many more. The environment is even a beneficiary of the system chosen to execute a shift event. This chapter continues the discussion towards the execution of the ideas discussed previously. In this chapter, another new tool and a new concept will be discussed. The robustness checklist is a tool that is used similarly to a DSM to show interconnections between functional goals and design parameters. The robustness checklist will provide the system engineer with a closer look at cause and affect relationships between components and targets. Its use further institutionalizes basic understanding of the system. The concept is around the management of the design space that extends the boundaries of the system, borrowing from discussions earlier about the PDP, where ideas come from and the organizational aspects of NA OEM. As always, the organization is intertwined with the process. It remains important to see how the component centric culture and limited scope of the role of systems engineers contribute to a poor overall optimization of the system.

A purely functional view of the world leads to local optimization

The Automatic Transmission division at NA OEM is rare amongst divisions at the company in that a position exists, titled "Transmission System Engineer". This is significant in that there must be awareness that the role of systems engineers is important enough to devote a title and job function to it. The problem is that the role is defined in a way that does not require true systems engineering. Historically at NA OEM, the role of a systems engineer is three-fold. First, the system engineer is responsible for high level functions that cannot be assigned to a component individually. Oil management and cooling are amongst those. Since the oil management and cooler are not able to be measured or worked independent of the entire transmission, the job must fall to a systems level engineer. Second, the systems engineer was responsible for making sure that component engineers were delivering their designs on-time and
that there wouldn't be any program delays. They effectively track the work of component engineers. Lastly, the systems engineer was responsible for development testing. Unfortunately, the common method of doing development testing is by swapping components when issues arise. Once a component is swapped that resulted in an improvement of the issue (even without statistical significance to the test methodology), the component engineer was provided the guilty part and was told to fix their broken design. In many cases noise in the system masks the true root cause but the guilty component, i.e. the one that was swapped when the test was run is still to blame. A process called Global 8D is used to document component defects and there are other tracking documents used to make sure that a bad component is traced to root cause (at the component level) and resolved. Nowhere in this process is acknowledgement that the parts are put together to form a system that is a separate entity. One + One = Three; you, me and us.

Component engineers at NA OEM focus predominantly on function. Reliability and robustness engineering process at NA OEM is centered around functional requirements, but lack any awareness of the other needs of the system. Figure 6-1 is an example of the robustness checklist that was established at a systems level by component engineers who saw only functional targets as being within the scope of the system. At this level, it may be possible to optimize the system, but only for the attributes shown, all of which are functional. This leads to local optima.

The checklist, similarly to a DSM, captures several important considerations. First, it links the critical system level attributes to the components within the system that affect it and it further reduces the list to the degree of interaction, 3's are highly dependant and 1's have some dependency. Blanks have no effect. Two troubles exist while looking at this list, however. The first is to properly map the interaction, and the second is that this list is too highly focused on functional attributes; shift quality attributes, to be more precise. There are lube and cooling functions that are not captured and the host of other considerations, cost, development time, etc. that are not listed either. In an expanded world, where more attributes are considered, a true global optima is possible. Recall from Chapter 2, Figure 2-2, an attribute DSM that shows high degree of coupling between the various attributes of the system. The attributes are so highly coupled, that changing one affects the others. The robustness checklist helps to show what features of the system are linked to each attribute.
Project Description:
Hydraulic Controls Development and Characterization

Functional Req'ts: Target (for the subject component / system)

<table>
<thead>
<tr>
<th>CTS #</th>
<th>Description</th>
<th>Design Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TDV Target</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Torque response frequency analysis</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Shift Duration</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Shift Delay (for engagements only)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 6-1: Robustness Checklist - this checklist helps to establish linkages between components and system performance attributes.

An expanded view of the boundaries of the systems engineer reveal more opportunity for optimization

A new robustness checklist with other attributes paints a much broader picture of the system and how it meets the needs of the beneficiaries. Once the concepts of system architecture are blended into the functional list, the matrix becomes much more useful. Figure 6-2, shows the new list of critical X's or CTQ's (Critical to Quality).
As part of the expansion in scope from a pseudo systems engineer before to a true systems engineer after the introduction of these new concepts, the boundaries of a systems engineer have also expanded. Reminding the reader, the goal is for global optimization. This implies a shift in the role of the systems engineer, from specification author and tracker to chief problem solver, with influence over a broad design space. Prior to this conceptual shift, the design space was governed by each component engineer, and rarely did they have the ability to collaborate on solutions.

**A new approach – the DSM**

Still there is a shortcoming. The list of design parameters across the top of the document fail to capture interactions between the components, and thus, recognition of the system. This is where the DSM becomes very powerful, it highlights interactions, and thus, places for a systems engineer to dwell. The DSM in Figure 6-3 shows that no component is independent of any other components and that the system is highly coupled. There is a large block of the development
process that must be done as one large system, iteratively and in parallel. For example, the pump design cannot be completed until the clutch, solenoid, cooler and lube designs are complete.

Another interesting observation is apparent, in addition to the high degree of coupling. The hydraulic control system targets will determine inputs for 9 subsequent tasks. It touches nearly every piece of the control system. If there is a change to the targets, based on the difficulties observed during the design process, then the whole system will be affected. If the targets are cascaded down, without feedback allowed back up, then constraints are added to the design that may negatively influence the ability of the design to achieve global optimization.

![Figure 6-3: DSM that shows the steps in the design process of the Hydraulic Control System and the interdependencies of those tasks.](image)

The most important observation, however, is that the scope of the systems engineer must be very broad, much broader than it is currently in practice. Earlier, it was shown how the needs of the beneficiaries are within the scope of influence of the systems engineer, including utilization and management of resources. Expansion of the influence of the systems engineer to include the
advanced engineering team and the software engineering team surfaced next. The yellow highlights show the point in the process where Software Engineering establishes requirements for the software. Looking down the yellow column, everything looks fine. Strategy is developed after the software targets are specified. But if the reader looks across the yellow row, you can see that the calibration and the shift quality verification steps both impact the software targets. In green, is shown the development of the software itself. The software design receives input from very few sources (look across the row) but drives changes to nearly the entire system (look down the column. Both of these functions are outside of the current boundaries of the role. Now, all the pieces fall into place and the true scope and boundaries of the systems engineer are captured in the DSM. The high degree of coupling forces a system that cannot be broken up. The systems engineer has to be responsible for and accountable to all the attributes, globally, of this system, and all the resources necessary to deliver it.

**The design space**

For example, an issue arose during the development of a control system with destroke function on a clutch. The destroke phase is when one clutch element is let go, simultaneously with another clutch element stroking (coming on). The purple traces in Figure 6-4 show the behavior of the system when a clutch is destroked quickly. In this control system, there is a latching function, which must be overcome prior to the regulator valve moving into a regulating position. This, coupled with the gain in the circuit, the stroke of the regulator and latch valves and the discharge flow rate of the solenoid, produce a large undershoot in the baseline cases. The dashed purple line represents the theoretical behavior if the event is commanded as a single step. The solid purple line represents a step to delatch followed by another step to destroke the clutch. The undershoot caused the clutch to lose its ability to hold torque and resulted in a shift flare (the engine speed rises due to no resistance through the drivetrain).

Several months of work went into modifying the components, starting with the solenoid and then the regulator valves to avoid this issue, but to no avail. All the design parameters that had an affect on this issue also had negative affects on other shift events. The design space was reconsidered, using the DSM in Figure 6-3 to explore other tradeoffs, and reprogramming the shift event was considered to be a viable alternative. This was experimented with using an
analytical modeling tool and was confirmed in vehicles. The new event profile is shown in light blue. The resulting pressure profile is in dark blue. It can be seen that the reshaping the event results in a 14% reduction in the undershoot and was able to manage the event in such a way that the delatch event was able to be separated from the destroke of the clutch, even further reducing the chance that the engine would flare due to a loss of torque holding capacity. The design was then further explored for robustness. The hold time of the delatch portion of the event was modified to understand the affect of variation on the event, in case there is part-to-part variability. The dashed dark blue line shows that the event is not very sensitive to changes in command profile, which is later confirmed by implementing the revised strategy in multiple vehicles with different performance and showing that they both proceed through a delatch event without flare.

Figure 6-4: An example of a simulated shift event with oscillations present. Also shown are changes to the software to mitigate the oscillation while still destroking the clutch on time. This was previously thought to be a solenoid issue, not a system issue.
By looking at an expanded design space, revealed through the DSM (Figure 6-3), the systems engineer was able to manage cost, development time and resources while solving this issue, something that was previously not done. The expansion of the design space is shown in yellow and green in the DSM. These two steps were outside of the old design space but as this example illustrates, they are highly coupled in the system and belong within the influence of the systems engineer. It has traditionally been an environment where this type of issue would be traced back to a part or several parts, through trial and error testing. The component engineers would then have an issue logged against them and would be responsible for solving the issue by modifying the components. Often this occurred in isolation. Each component engineer who was suspected (in this case, a solenoid engineer and a valve engineer) would work separately on what they thought was the cause, and then the modified components were assembled, the trouble event was studied and if improved, the team moved on. If there were further interactions, the process would repeat. Finally, the modified components would be tested in other conditions and events, and new problems would be discovered. The team would then attack the new issues, which may have been created while solving the old issue. This component swapping process had too much rework, and took too much time to be considered effective. Plus, it usually failed to solve problems without creating new ones. This example shows how component engineering is ineffective at solving systems issues. It also shows how understanding the design space of the system, which now includes software and calibration, can open up new solutions that would never have been tried before.

**Chapter Summary**

This chapter is significant for defining what processes, tools and techniques are available to transform and organization from thinking about components to thinking about systems. The Robustness checklist helps to link attributes, which now include things like resources and development time, to the specific characteristics of the system. Shown are both component centric and systems centric views for contrast. Taking this thought a step further, however, the DSM is able to identify the iteration loops and the points of interconnection between the different steps in the design process. A wonderful outcome of this perspective was in being able to see a new path for the solution of a system level problem that was not available in a component centric world.
While teams at NA OEM are cross functional, they are focused still on modules and components, rather than on the larger system. The relationships between product and manufacturing engineering are very strong, despite the natural tension between the two organizations. The connection between product and purchasing is growing deeper and stronger each year. At the working level, the trouble that NA OEM faces is not that there is disagreement internally between the different chimneys of the organization, but that the teams are simply looking at the product in the wrong way. The cross functional teams are focused on modules. The modules look like systems, but the boundaries for the modules are drawn according to packaging and sourcing patterns, not according to functional definitions, as defined by the DSM. A new modular pattern and a push for standardization follow. It uses analytical tools to drive the change and an example to highlight the needs for and benefits of standardization.
Chapter 7 Standardization

The previous chapter defined the role that the systems engineer plays and showed valuable tools in defining that role, the robustness checklist and the DSM. It also discussed the design space and how by managing the design space, more optimal solutions can be achieved. This chapter takes that concept one step further by talking about standardization in designs. Also discussed are analytical tools which can shave critical development time but only through standardization. By properly standardizing, efficiencies can be realized and global optimization can be reached.

Analytical tools and processes

Traditional story

Analytical tools exist to help the systems engineer to understand the main affects and the interaction affects of the components on the critical attributes highlighted in the robustness checklist shown in Figure 6-2. These analytical tools are supplemental to the development process. Traditional forms of development occur with hardware testing. These tests use Designed Experiments (DOE), One Factor at A Time (OFAT) and trial and error testing. In a typical program, once advanced has produced a design that they believe to be IR, the design is passed along to the PD team. The PD team then begins to install the control system and/or transmission into a surrogate vehicle for testing. Calibrators drive the car around, subjecting it to a battery of different conditions and they begin to make a list of things that they don't like about the performance of the control system and/or transmission as a whole. This find-and-fix method is haphazard and unpredictable and reveals very little about the nature of the control system.

OFAT testing is more sophisticated than find-and-fix, but leaves gaps in understanding interactions. DOE, a much more intricate experimentation method is useful when the interactions and main affects are relatively well understood in advance of the experiments. If the team knows what to look for and how to look for it, then a DOE can be useful to optimize, but more often, the DOE is a shotgun approach to understanding systems level problems. A systems engineer makes a list like the RRCL and designs a very expensive experiment to manipulate the parameters and to measure their affect on one or more of the critical Y’s. Out of the 16 or so runs,
most are filled with experimental noise. Usually, only a few of the factors have significant
affects on the desired outcome. Most of the runs of the experiment produce very little in the way 
of knowledge, except to tell the engineer what is not important. Don’t misunderstand, knowing 
what is not important is often as important as knowing what is important, but there is a better 
way.

Analytical tools are the most sophisticated and elaborate method available to understand the 
behavior of a system. The main benefits are that the many runs, configurations and parameters 
that lead to knowing what is not important can be done virtually, for a fraction of the cost of 
doing the same experiments with hardware. The difficulty with analytical models, however, is 
that they are very difficult to create and to correlate. In most cases, at NA OEM and other large 
PD organizations, the process follows something like this: Architectures are selected and the 
design is frozen. Then parts are fabricated and testing commences while the model is created. 
The model is then correlated to a handful of maneuvers that are benchmarked in one set of 
hardware. The model is then manipulated and used to determine why things acted the way they 
did in the hardware. The design is then optimized using the analytical model and the parts are 
modified to match the hardware. The new design is then benchmarked again, and the model is 
re-correlated. This cycle iterates a few times.

The major problem with this process is that the model performs very little prediction. It is a 
useful process for understanding the physics behind the system, for understanding the 
interactions and the system as a whole. In essence, the model can only look backwards and then 
reacts to the things that have already been seen in physical tests. Dynamic modeling is far less 
advanced than is static modeling. In Finite Element Analysis (FEA), the methodologies of 
modeling, solving and analyzing are so good that hardware can largely be avoided. The engineer 
has to run a validation test at the end to confirm that the model prediction is correct, but the 
iteration is done analytically. This is not the case, or at least has not been the case, for hydraulic 
control system models. If this can change, then the team will benefit from this process by 
developing faster and with less cost than if they had to rely solely on hardware experimentation.
The following example is one of reactive modeling. Figure 7-1 was measured experimentally in a vehicle during the development phase of a hydraulic control system. The equation for the regulator valve results in a predicted pressure of 60 psi, given the 35 psi of signal to the valve. The red in the trace shows the actual signal with the commanded signal in black. There was an obvious issue that the blue line (regulator valve) overshot its intended pressure by over 40 psi. Once the issue was discovered, the analytical team was engaged to try to understand what happened and how it could be fixed.

![Figure 7-1: Actual shift trace from vehicle where there is an overshoot (shown in blue) that does not follow input (command in black and actual input signal in red).](image)

The analytical team produced the results as shown in Figure 7-2. The model had already existed and was already correlated to the hardware through a model correlation process. The team had learned through earlier projects that the analytical model could be extremely useful and so the model as built and validated ahead of time. This process, however, of building and validating a model to the first prototypes available, still has not been institutionalized at NA OEM. It only occurs when the systems engineer makes the requests and provides the experimental results to the modeling activity.
Using the validated model, the same inputs were modeled and the outputs from the model follow. It turned out that there was entrained air in the circuit, causing a loss of damping in the regulator valve. This loss of damping caused the valve to move too far out of position, and resulted in the overshoot. Further inspection of the hardware revealed that a port was exhausting to air that was through to be exhausting into oil. This change was a result of lowering oil level in the transmission to improve efficiency. Lowering the oil level reduced viscous drag in the barrel of the transmission, but also inadvertently revealed this exhaust path that was intended to drain into the sump of the transmission.

Figure 7-2: Model results - very strong correlation but only after the overshoot was discovered and measured experimentally.

This lesson and many like it would have been much more difficult, if not impossible, to learn without the use of the analytical CAE models. The models provide detail that is immeasurable in hardware, like valve positions, port openings, acceleration and velocity of valves during different phases of the shift event. They reveal clues about damping, natural frequencies and other dynamics of the system that are also difficult or impossible to measure experimentally.
Analytical processes today, are reactive, as the previous example shows. The model must be
designed, the hardware built and tested and the model then correlated. This results in the
following process map

Build – Experiment – Model – Correlate – React

Although this has some benefits, as highlighted in the previous example, the process has a
plethora of drawbacks. Most of the learning is about why things happen, not about how to
prevent them from happening. This is obviously the first necessary step, but the potential exists
to take the process on step further and to make the model predictive in nature, but the design
must be made with the model in mind. Two major processes must be executed in order to use
the analytical tools predicatively, the first is standardization and the second is modularity.

A new approach through standardization

Standardization implies evolutionary development, rather than revolutionary. The history of NA
OEM shows a long line of revolutionary design changes, but not all with positive outcomes, as
the example of the heavy duty truck transmission in the introduction showed. Revolutions are
often costly, while evolutions occur so slowly that no one hardly notices, until they go back to
see where they have come from. Especially in terms of a hydraulic control system, the reasons
for evolutionary change are strong, and for revolutionary change are weak. Technologies don’t
evolve that fast, customers are inherently happy and program managers are not encouraged to
take risks. Despite this, however, a brief discussion of recent history reveals the opposite trend.

In the 1960s, 70s and 80s, the transmission division at NA OEM was very focused on
evolutionary changes. The transmission families grew slowly and were industry renown for
durability, quality and toughness. Shift quality, mind you, was not very much on the radar of the
consumer of that day, however. In the 1990s, technology did change, and it drove significant
revolution in the industry on two fronts. Fuel economy became more of a transmission problem,
where until then it was seen as a vehicle mass and engine problem. Also, shift quality was
suddenly on the radar of consumers, mainly due to the recent implementation of electronic
controls.
Four separate transmission families evolved, three for rear wheel drives, and one for front wheel drive transmissions, and older models were retrofitted with add-on electronic modules. Every one of the modules was sourced from a different supplier. Every one had a different architecture and every one had a unique implementation. Technological S curves tell us that this is normal. In the early stages, many players appear on the scene and technology evolves very quickly. At the later stages of adoption, however, a consolidation occurs. This consolidation is what has taken too long for NA OEM to adopt. It has become ingrained in the culture that each program makes its own decision, in the name of time-to-market, that end up in the end driving in cost, complexity, inefficiency and inevitably, delay time to market.

What is needed now is a standardization of technological platforms and architectures, the last phase of the technology S curve. If the basic architecture of the control system can be standardized, then not only will the analytical team have an opportunity to do predictive modeling, where the changes are very subtle and can be done in advance of hardware experimentation, but the program manager, the functional manager and most of all, the systems engineer will benefit too.

The reason that the analytical team can keep up with evolutionary change is captured in the following example. The find and fix team of the heavy duty truck transmission illustrated in the introduction, was assigned to update the architecture of the main line pressure solenoid after the major efforts were completed and the production schedules were under control. The team had already developed a system level analytical model and now the project was to change a piece of the model (the solenoid) and to integrate it into the rest of the model.

The project was a huge success, the model revealed features that were incorporated into the new solenoid design. There was little correlation activity driven by changing only a small segment of the model, and once hardware, which was driven by the model, was prototyped, the experimental results were then used for validation. The validation process went smooth and the resulting solenoid design worked brilliantly. Unfortunately, the story does not have a happy ending. The hardware prototypes were tested in extreme environmental conditions, outside of the
mathematical boundaries of the analytical tool, and failed. The development process became one of OFAT, DOE and other hardware based tools, but eventually the design was not implemented. The existing design continued to improve, yields went up incrementally, and as more time passed, the urgency for changing the architecture diminished. The next generation transmission become the focus, including a complete redesign of the control system, rather than improving the existing transmission performance and the project was canceled. This last part is important, because there are some things that cannot be modeled, and the need to develop prototypes will never vanish. Standardized designs, however, will always be in good taste, because they transcend the analytical tool and reduce the effort required experimentally also.

For another look at standardization, it is worth considering what leaders in the industry are doing and how they are applying platforms and architectures into their designs. One of the leaders in the industry chooses to use a wide variety of direct acting solenoids, similar to the ones selected for the heavy duty truck transmission discussed at NA OEM. The issues that were faced by NA OEM in contamination sensitivity are the major handicap of this architecture, but this drawback was cured by the competitor through systems engineering. There is a state-of-the-art filtration system that accompanies the solenoid choice. Direct acting solenoids require a state of the art filtration strategy, so they are implemented as a system. The result is outstanding shift quality and consistency. The reason that this filtration system and solenoid architecture has not been chosen at NA OEM has been the apparent cost of the system. It is unknown, however, what the cost of the competitor’s design is. The competition uses up to 25 valves, 8 solenoids, 8 accumulators and five spring-dampers, compared with NA OEM’s best application using 15 valves, 7 solenoids, no accumulators and no spring-dampers. Anecdotally, however, the leading competitor is known to produce low cost and efficient designs. Perhaps cost can be managed in other ways than content, and perhaps, as has been alluded to throughout this work, the true cost of a system is not measured in piece-price, but in development, investment, people and in time to market.

Lastly, the major competitor has recently released a new control system architecture. This new architecture is a DESC type of control system, like the heavy duty truck transmission attempted at NA OEM. The question is, “will it work for them?” or will they too face similar difficulties.
Here is where the story of the major competitor changes from the one at NA OEM. Rather than use a new solenoid supplier and changing their entire design, they are weaving DESC into their existing architectures. Maintained are accumulators, dampers and the filtration / solenoid interconnection. They have combined the solenoid (signal) and regulator (amplifier) functions into one and have developed this new technology with their exiting supplier partner.

The final thing that was found to be particularly telling, however, is that the competitive benchmarking data provided by NA OEM was nearly 100% functional. There was no mention of the resources used, the manufacturability, the weight, expected cost (other than counting parts and multiplying by NA OEM costs per part) or development time. This final clue as to the component and functional bias of the engineering team at NA OEM reiterates the need for organizational and process change.

If an architecture is standardized and then evolved slowly over time, then implied is the fact that most transmissions within NA OEM will share parts and share basic design features. Reuse is fundamental to standardization. In Chapter 8, discussion will go into the ability of a systems engineer to then modify or adapt smaller sections of the design, like the competitor changed the solenoid and regulator valve but maintained the pump, clutch, accumulator and filtration components, the system changes, but only in containable amounts. Then, the systems engineer can work on perfecting the interconnections that have changed, without having to reengineer the whole system.

Chapter Summary

Standardization opens the door to global optimization. No one engineer on one program could possibly get so many things right at once as to achieve global optimization. A standard design platform, on the other hand opens up the possibility of refining and iterating a design to become optimal. Standardization is the ultimate tool that can be applied to allow the development process to be perfected, and works beautifully when it is a system level platform. It would of course be easier to standardize components, but the interactions, as have been shown previously in this work would lead to sub-par system level performance. Platforms and reuse of sub-systems is perhaps the greatest advantage that overseas competitors have over NA OEM. In the
final chapter a new organizational chart, with the system engineer at the center will tie together many of the thoughts presented so far. This new systems engineer has many tools at his/her disposal to now optimize the system, balance the iron triangle and win in the marketplace.
Chapter 8 Conclusions

Two main questions were asked at the beginning of this work:

- "What effect does the organizational structure and corporate culture of a company have on the ability of that company (or division) to achieve global optima with respect to the design and development of a complex product?"
- "What processes, tools and skills enable engineers to institutionalize systems level engineering in order to be able to execute globally optimal designs on a regular basis."

In this chapter, the answers presented throughout this work will be summarized. Most of the changes are organizational and cultural. Until those battles are won, the second question of how do we do this, is mostly moot.

Getting back to the example in Chapter 1 of a team of systems engineers designing a new control system, the team did see many improvements. Despite the fact that many of the suggestions in this work were made too late to see benefit in the example, the team did have a systems level focus. The architecture was defined by the team, but prior to learning about and seeing how to apply system architecture in a disciplined way as was shown in this work. All of the discussion regarding beneficiary needs and defining the system goal statement was done after-the-fact in the design of this control system. Despite that, having a system focus and being aware of the difference between local and global optimization did help the team to make some good decisions. The organizational changes, obviously, have not been realized, and so true optimization is still a long way off. Although not aligned organizationally, good working relationships enabled the strategy changes which improved the system performance to be implemented.

This highlights two major initiatives that NA OEM specifically, needs to implement. First is to change the way that component engineers think about design in NA OEM. They must shift from being component focused and step into a role of being systems focused. This implies organizational change as well as cultural change. It also requires a new pool of talent to develop around how to design systems. Secondly, and very importantly, NA OEM will have to
restructure the role that Advanced Transmission and Software Engineering play in the development process.

**Organizational Structure Changes**

The organizational structure at NA OEM was adopted to match the architecture of very early automatic transmissions. They could be broken up into pumps, clutches, control bodies, and so forth, so as people developed expertise in these areas, they grouped themselves. The trouble was that this structure was established based on components that can be discretized. The system cannot be seen, so there was never recognition to organize around it. Throughout this work, points were made about how the objectives of various parties are contrary to global optimization and systems engineering. Figure 8-1 shows a new organizational diagram that highlights the importance of systems engineering and how it fits within the transmission world at NA OEM. First of all, systems engineering must oversee the component activities. Additionally, software and core calibration are part of the complete package.

![Organizational Structure Diagram](image)

*Figure 8-1: Systems engineering oversees components, software, core calibration and advanced as they all relate to the system. Importantly, the team delivers full systems to the transmission program managers.*
Several important points need to be mentioned about the rest of the organization before discussing the details of the new group. First is that Software Engineering and Advanced Transmission still have roles to play outside of hydraulic controls subsystem development. They remain in the organization, separated as they were before. A piece of what they do is integral to the system discussed and those functions move inside of the Forward Model organization. Second, there is still a Transmission Systems Engineering group within the program group. These engineers will do vehicle specific systems engineering. Trailer tow, packaging, and ratio selection are examples of what this group will continue to do. Third, there will still be a need to calibrate control systems in each vehicle application. No two vehicle programs are alike, but a cutoff point exists at the hydraulic control system level where much calibration can be done internally. These calibrators will work between hardware and software engineers, with the oversight of the system engineer. The tuning of the software and calibration are now knobs that the system engineer can manipulate to optimize the system performance. Further refinement to suit each vehicle program can be done by the program specific calibrators.

A future goal of systems engineers at NA OEM should be to understand which calibration points are core to the system and which ones should be used to tune the transmission into a vehicle. It was beyond the scope of this work. At this point, systems engineers recognize that much calibration is done to address the hydraulic control system performance but not all of it. A clear example of a vehicle tuning parameter is shift points. These represent engine/vehicle speed combinations that initiate a shift. A calibration point that is not clearly internal to the system is the shift profile. Once a shift has been initiated, the pressure signals that transition one clutch to another may need to be tuned based on vehicle stiffness or engine flywheel mass. This development is an evolution to the process which will be discussed in the next section.

Finally, new architectures must be developed from within the systems engineering group. There is too much influence exerted early in the architecture definition stage to allow it to occur outside of the influence of system engineers and still hold them responsible for global optimization. The systems engineer is at the center of this universe and must be aware of the intent of the system and the needs of the beneficiaries so that proper tradeoffs can be made. This information can be documented and conveyed from individuals to other individuals, but they must all be from the
same team. The handoff from one organization to the other presents too much of a gap to have
the intimacy required to make the right decisions. There are still be two teams of people, one
who do architecture development and focus on the needs of the beneficiaries. They will study
the technological landscape, read papers and benchmark competition. Then they will consider
how to adapt the architecture in place (standardization is discussed later in this chapter) to
integrate with newly available technology and only when a mature subsystem has been
developed will it be integrated into a production plan. This team must maintain some separation
from the program team. The second team is responsible for detailed development of the
subsystem. They are kaizen (continuous improvement) experts and their job is to refine and
deepen the understanding of the system. This job is not finished at program launch.

This leads to modifications to the PDP. The PDP is fine in how it develops new vehicles, but
powertrains must maintain some independence from vehicle programs. Vehicles must react
faster to the market than powertrains. Powertrains must be amortized over longer time periods
due to the higher investment costs. This separation already exists. Vehicles and powertrains are
on different refresh schedules. But a new layer of separation is necessary. Hydraulic controls
cannot be developed from the ground up in the same amount of time that vehicles can. Even
whole transmissions are easier to develop, or perhaps integrate is a better word, than hydraulic
control systems. When the three, vehicle, transmission and hydraulic control system are all new
or when the definition of IR is not satisfied, then the program is in big trouble. By allowing
subsystems to evolve independent of programs, there is a much higher likelihood for long term
success. Hydraulic control systems are separable in that repackaging is insignificant compared
with redeveloping the interconnections. The interconnections revealed through the DSM must
be maintained but the parts can be repackaged into new shapes and sizes, as long as those
constraints are maintained.
Tools steps and processes that facilitate institutionalization of knowledge

Standardization is the most important step in institutionalizing knowledge. Standardization has so many benefits and the concept is not by any means new. Some benefits from standardization are:

- Ability to use models predicatively
- Evolutionary design
- Leverage knowledge from multiple engineers
- Leverage lessons learned across multiple programs
- Increase utilization of engineering, prototyping and testing resources by doing things once and replicating them for multiple programs.
- Increasing reuse across multiple programs, including components and suppliers.

The culture, as was detailed in Chapter 3 explains much of why there is less standardization. Program managers are not motivated to talk to each other, they are too focused on managing their resources and delivering their programs to help drive commonality. The benefits of commonality take too long to realize and they be penalized in the short run. Here, the technological arm is able to maintain some long term vision, while the program arm adapts to the specific requests that each vehicle program makes. They maintain the program management, vehicle calibration and vehicle systems engineering functions. The technological arm is now strengthened in the organization through equal reporting structure and signoff ability.

New tools are available, like system architecture for defining the platform, the Robustness Checklist for establishing the link between attributes and functions and the DSM for defining the interconnectivity of the system that will allow a system level view, global optimization and sustainable, long-term results to come to fruition. These tools, and a new paradigm about what is included and what is outside of the boundaries of the system enabled one of the most successful control system launches in recent NA OEM history. The staffing levels did go up, but not as much as previous programs. Prototype budgets were not blown, although there were many changes and continuous churn. This is unavoidable until the lessons on standardization are
institutionalized. The bottom line is that by just changing your way of thinking, much can be accomplished, but as was stated in the very beginning, if the team is disbanded, the lessons are lost. The shift from component to systems culture must be ingrained through the suggestions made in this work to see true optimization.

**Future Work**

There is still much more work to be done on this subject. Systems engineers, once established in their role, strengthened through the organizational structure and winning the cultural battles will have to spend time perfecting the established architectures and integrating them into programs. At NA OEM, this process is underway. The spectrum of differences between control systems in the fleet is getting smaller, but it is still not sufficiently institutionalized and the cultural battles have barely begun. There is not enough acknowledgement of the problem to implement a new reporting structure. Ninety five percent of the workforce sees the world through component eyes and 50% of the engineers report through component managers. The program managers still have much more influence over new designs than the functional managers. Functional managers are still trying to optimize components. Advanced Transmission and Software Engineering still behave independently. The biggest future work is in transforming and changing the culture and getting executives to support the new approach enough to change the organizational structure. After those battles have been won, teams can dive deeper into the tools and processes that will refine the technical work. The greatest battle is winning over the long entrenched culture. The fight shouldn’t be too hard, however, since times are so tough. Everyone in the industry knows that they need to change; the hard part is making commitments with the few resources that are left which may take time to return on the investment. This takes courage, vision and leadership. Is anyone listening?
Bibliography

i Senge, Peter, The Fifth Discipline, 2006