Using the Design Structure Matrix and Systems Thinking to Develop a Requirements Driven Automotive Closures Design Process.

by

Robert J. Corby, Jr.

Bachelors Mechanical Engineering
University of Detroit Mercy, 2000

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

February 2008

Signature of Author ________

Robert J Corby, Jr
Department of System Design and Management
February 2008

Certified by ____________________

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

Accepted by ________________________

Patrick Hale
Department of System Design and Management - Director

Acknowledgments

A very special and sincere thank you is owed to Denise Mitchell for her suggestion and support for my involvement in MIT's SDM Program.

Another significant thank you is owed to Tony Zambito for his countless hours in mentoring and guiding me through the SDM Program. His commitment in my development, professionally and personally, extended well beyond the office walls.

Additionally, I would like to say thank you to Dan Whitney. His experience and knowledge in systems thinking, DSM, and the auto industry are superior. His commitment to this graduate student extended well beyond normal office hours and involved many hours of phone conversations outside of normal business hours.

Finally, James Allison and Joseph Mravec, two of the finest combat leaders of men I have ever met, have to be recognized for their support and flexibility in helping me manage my "part-time" career in addition to my full-time job and academic interests. In Orbe Terrum Non Visi.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction and Motivation

Chapter Introduction

In this chapter we first review the predominant external factors currently pressuring automotive product development teams. Links between major market factors including Time-to-Market, Show Room Age, and Market Share are established and their implications to automotive product development teams are introduced. We then review some of the internal challenges facing PD organizations. The motivation for this work is then discussed. Next, we study applicable literature and develop an understanding of the strengths and weaknesses of these prior works. This leads to the development of our problem statement and finally, the outline of this thesis.

Background

North American Automotive companies have long been criticized for their slow time to market and poor quality. Although this is the general perception of customers and many popular media outlets, recent trends show significant improvements in both these areas and the North Americans are steadily catching up to their Asian rivals (JD Power and Associates, 2007), even to the point of beating the Asian car makers in isolated "islands of success". However, North American car companies still have much improvement ahead of them as the industry is experiencing extreme pressure on many fronts. For the NA OEMs, their long standing dominance of the US market has steadily eroded. Figure 1 shows the dramatic decrease in market share that the North American car companies have experienced. Now, very large companies that have traditionally supported much larger market share and larger demand are being forced to restructure and eliminate both hourly and salary jobs by the thousands to "fit" their companies into the market size they now exist in.

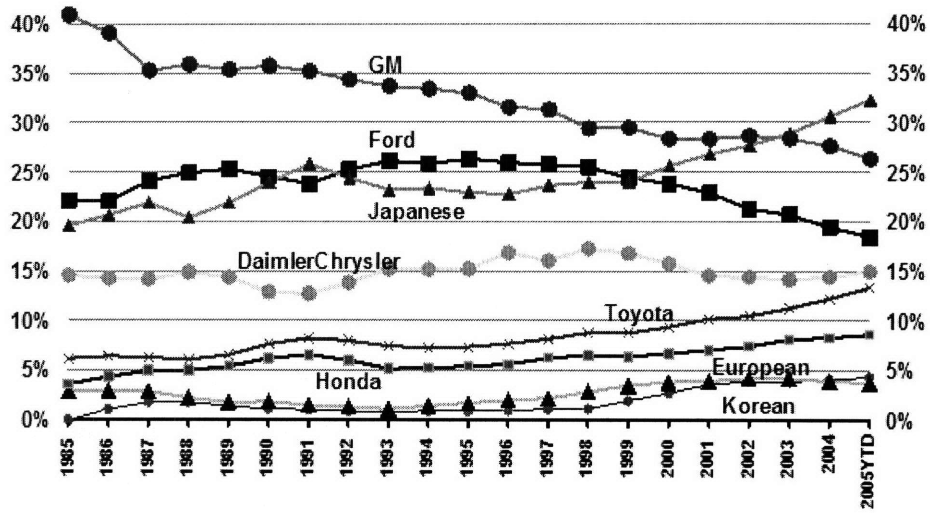


Figure 1: Automotive Company US Market Share. Source: Wards Auto Info Bank

Coupled with the challenge to do more with less, all OEMs are attempting to turn product over at a much faster rate. Product development cycles have shrunk in units of years in the recent decade and this has affected the average show-room time of products. Leading companies today have minimized this metric and most Asian car companies lead the way here. Two very common metrics used to express automakers abilities to get new product to market are design freeze time-to-market and replacement rate.

Design freeze time-to-market represents the time it takes an auto company to go from design freeze to production. Some explanation is required here as this has at times been a slippery concept. Design freeze is the point at which major aesthetic design and development activities are completed and the skin of the vehicle is "frozen" and limited to only the smallest of change there after. This also represents the point at which most auto companies begin to invest in earnest in hard tooling and manufacturing resources. As such, a significant amount of capital is potentially spent post design freeze and the less time a company spends in this stage the better off they are likely to be relative to cost expenditures. Through out the 90's most American automotive companies had times around 35-40 months from design freeze to product launch; whereas, Asian competitors were much faster. Today, American auto companies have reduced this time to around 18-25 months.

The other major metric, replacement rate, represents an auto company's ability to replace its current product line up with new product and is the percentage of current product that will be replaced each year with new product. Again, the terminology demands a bit of explanation. By *new* it is meant a *major* product offering. The auto industry typically classifies minor in-cycle product revisions as *minor refreshenings* and all major developments as *new* or *major products.* The industry average replacement rate since the early 1990's has been approximately 13%; however, recent trends indicate that this number will very quickly rise to approximately 18% in the very near future (Merrill Lynch: 2007).

Replacement rate manifests itself into showroom age – or the number of years on the market a car model typically stays on the showroom floor, with a lesser time being better. The industry average is approximately 2.9 years and this metric has steadily decreased over time (Detroit Free Press, 2006). See Figure 2 for the trend in average showroom age. Some of the best Asian auto companies have showroom rates around 2.4 years and some of the worst North American auto companies have showroom rates around 3.5 years.

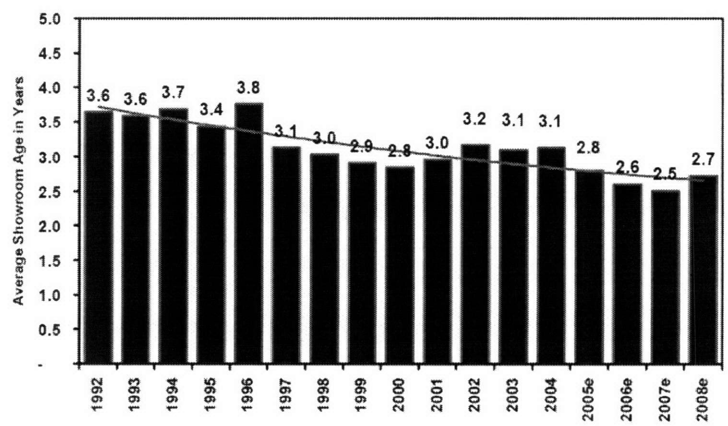


Figure 2: Trend in average showroom age; Source: Merrill Lynch, 2006

Collectively, the above metrics represent a tremendous amount of pressure for auto companies to be able to create and manufacture products quickly and at a high frequency in order to keep their product lineups looking fresh. These metrics continue to have meaning when viewed sequentially. The basis for Merrill Lynch's annual report, Car Wars, is that replacement rates affect average showroom ages which in turn affect market share. Market share then affects capacity utilization and that directly affects company's profits.

The 1990's witnessed a global consolidation across the automotive industry where many large OEMs acquired smaller companies that often offered luxury vehicles. A prime example was Ford Motor Company's purchase of Jaguar (1990), Volvo (1999), and Land Rover (2000). These consolidations present their own advantages and disadvantages, including the challenges of how to integrate best practices and product development schedules across a global enterprise that was formed essentially piecemeal.

Like most complex products, automotive product development represents the creation of products within the context of a number of requirements – from the system level, all the way down to the component level. The list of requirements in automotive development is long and varied. Requirements come from the Federal government (in the form of Federal Motor Vehicle Safety requirements), internal designs specifications, and many many more. Given the vast array and magnitude of requirements, it is possible to overlook or even not fully meet some requirements. Most unfortunately, when requirements are not addressed in product development, they manifest themselves later in the form of costly rework or later yet when customers find them. Even when product failures relative to requirements are caught in the development stage, unexpected failures have long range effects throughout the PD community. Therefore, whether during product development or field service, failing to meet product requirements can be very costly in both time, money, and a company's reputation.

Internally, large automotive companies are struggling with a number of recent realities. These include such challenges as executing product development with fewer prototypes, performing a greater amount of work at a faster rate and doing it with fewer resources, establishing higher quality products, and reducing churn through out the product development cycle.

How significant are these challenges? The author has seen some companies report that up to forty percent of their human resources are working on re-engineering efforts. This represents nearly one half of their personnel being tied up fixing problems on products already in the field. Of course, coupled with these re-engineering demands are the recent reductions in the work force, where thousands of personnel have been cut. General Motors headlines declared 30,000 job losses and plant closing in 2005; likewise, in 2006 Ford Motor Company's headlines declared 30,000 job losses and 14 plant closings over a six year period. Clearly, automakers no longer have the resources to manage huge re-engineering efforts without having a significant negative impact on developing new products.

One automotive company in particular (identified as Auto Original Equipment Manufacturer – or Auto OEM - throughout this thesis) is revisiting how product development is executed. With the aforementioned pressures in mind, this company has initiated a project to map the current state product development process for the closures sub-system (doors, hoods, trunk lids, etc) and integrate into this process the requirements and major interfaces of the sub-system. By utilizing the design structure matrix to map the process, requirements, and interactions, an optimized requirement driven product development process will be developed. The intent in integrating requirements into this process is to ensure that all requirements are met for the sub-system. This philosophy will be discussed more later. Furthermore, this project represents a staged-deployment in that if the project is successful, the same or a similar methodology will be deployed across the other major sub-systems (Underbody, Front End, etc.). Ultimately, the goal is to determine the most efficient product development process while also accounting for requirements and complex system interfaces.

Motivation

The motivation for this work comes from a strong desire to contribute to the North American auto industry in a positive and helpful manner. Market and competitive pressures have forced North American car companies to revisit not only the time it takes to get product to market, but also the fundamental ways in which product development is viewed; and despite great improvements, Asian competition still remains the leader in both these areas. The author's motivation is to contribute to the North American car company's return to success by providing a design structure matrix inspired methodology that will offer itself as a knowledge repository as well as generate a requirement-centric process, ensuring product development is done efficiently while being of high quality with respect to requirements. Most interestingly, this methodology may be applied well outside of the automotive industry, thus having far-reaching potential in application.

Literature Review

Given the magnitude of the automotive market, one can readily find prior work that is rich in content and application with regards to the DSM. Eppinger, Whitney, Smith and Gebala discussed matrix representation to capture relationships in complex projects and to define the "technical structure" of a project. Their work looked at a number of different ways to apply the DSM, including task based, parameter based, and hybrid matrices. Furthermore, they extended the usefulness of the matrix by incorporating numerical values representing the relative importance of a tasks as well as completion time for tasks. Browning applied the DSM by looking at team communication by mapping team interaction to a DSM applied to an automobile engine development team. Browning also discussed barriers to industrial use of the DSM. These barriers include the difficulty in acquiring data from people, the potential amount of data needed, and finally the practical limitations of large DSMs in excess of 500 rows and columns. Zambito applied many of the aforementioned practices to the development of an automotive hood system. In this work, a matrix of 43 x 43 was developed to capture the hood design process. He also accounted for task dependency, volatility, and sensitivity. He further developed the strength of the DSM by incorporating resource requirements and was thus capable of studying product development resource and timing issues and even cost implications. His predictions were later verified in industrial practice. In 2007 Noor and Whitney produced a paper applying the DSM to automotive closures development. Their work applied numerous systems oriented tools (datum flow chains, DSM, and systems dynamics modeling) in an effort to look at closures design from a number of perspectives, both technically and organizationally. They focused heavily on attributes and the interplay that exists between attributes and the complete closures system, including organizational implications. Noor and Whitney also applied simulation to determine probabilistically how long the door design process should take at a NA OEM. McGill also applied the DSM to the automotive industry where he constructed a moderately large sized matrix (773 X 773) to capture the product development efforts surrounding closures. As Noor and Whitney focused on attributes, McGill looked very closely at the manufacturing side of things. After maturing common DSM MatLab algorithms, McGill performed simulation on the PD process that predicted a reduction in the average completion time of ~80% and a potential savings of ~$5B. Cho also produced documentation for the partitioning and simulation of DSMs using Excel. In this work, full partitioning and simulation facilitated studying the process as well as rework.

Outside of the auto industry, in the housing construction business, Planweaver (and other DSM oriented software companies) has utilized DSM matrices in the magnitude of 3000 x 3000, proving that large matrices can be created and applied usefully.

There are many common and familiar tools for mapping out processes. One such tool is Critical Path Method (CPM) which represents project tasks as a network using graph theory. A thorough introduction and review is covered by Levy, Thompson, and Wiest. Whereas CPM treats time estimates as deterministic, the Project Evaluation and Review Technique (PERT) applies probabilities in project planning. Another common project tool is the Gantt chart, which typically shows work stream tasks horizontally and over time. Although Gantt charts do not explicitly show relationships between tasks, their graphical layout it very user friendly. An example of a Gantt chart is shown in Figure 3.

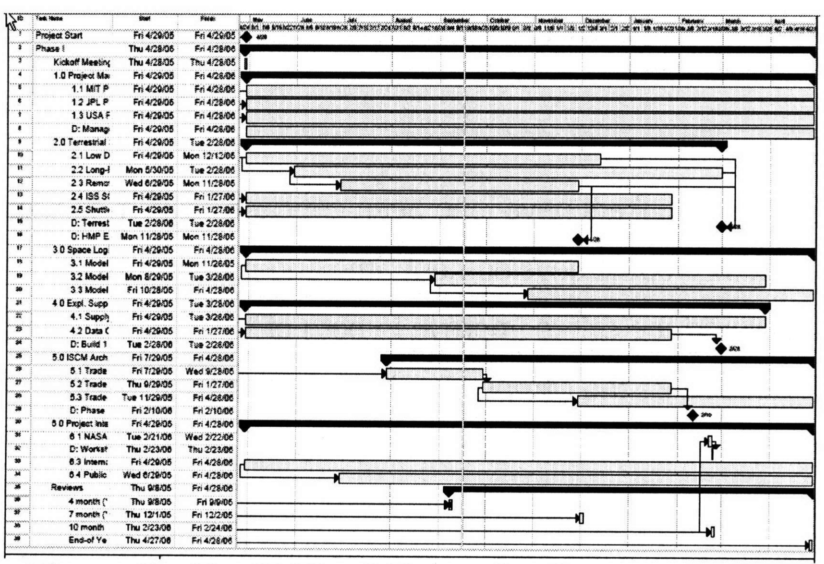


Figure 3: Gantt Chart Example; Source: Author

These tools, and others, enjoy widespread use amongst program managers. However, one significant detail is that neither of them capture the nature of iterative work. Task relationships in a process can be broken down three ways: serial, parallel, or coupled. In a serial process, one task is completed that feeds another in a defined and sequential manner. Parallel tasks, unlike serial tasks, can occur at the same time and are independent of each other. Finally, coupled tasks are those that rely on information from other tasks in a non-serial or parallel manner. The nature of this task relationship is best described as "iterative" and is often called a "chicken and egg" scenario. Unfortunately the fundamental rules in both CPM, PERT, and Gantt do not allow for this type of chicken-and-egg scenario. In other words, information is always fed forward. Large, complex projects are iterative by nature and the best tools to apply are ones that can account for such iteration.

As summarized above, there have been many applications of both DSM and project management tools through out the auto industry and in other areas. In this work, we look to integrate both product development - the creation of information via a prescribed process - and requirements – the assessment of information created via product development.

Problem Statement

North American auto makers have long been criticized relative to their Asian competitors in both product development time and product quality. Although NA OEMs have answered the call to improve in both of these key areas, room for improvement remains as the competition has not simply stood idle as NA OEMs worked to improve themselves. Today, vehicle programs are executed quicker than ever before, and they are done with fewer resources than ever. A discerning media and knowledgeable investors continually increase their expectations of auto companies. It is critical that product development plans are executed as planned and within budget in order to meet both internal and external demands.

This work presents a product development process that accounts not only for what tasks have to be done, but also for their deliverables and the product requirements that must be met. In this context, we integrate product development tasks and requirements to ensure products are delivered to specification. With this effort, we expect to address such critical issues as reducing product development time, reducing product development churn, and improving the quality and fidelity of product designs. We measure the effectiveness of our effort by measuring the mean process duration of the current process and comparing that to the same metric for the same process, but including proposals developed in this thesis. We can use the design structure matrix created in this thesis as a knowledge base which can identify the inputs required to make product assessments and at which point in time we should make these assessments.

Thesis Outline and Flow

Chapter 1

We begin by first reviewing the predominant industry and company dynamics driving the automotive industry today. We establish the need for shorter product development cycles as well as the need to create products to specifications. A review of relevant prior work is also presented to show where others have worked in this space. The problem statement and motivation for this work evolves subsequently.

Chapter 2

In chapter two we discuss product development activities as they relate to automotive development. We present automotive product development in the context of the system's engineering system-V. We also review the significant role of the virtual development process that is key to modern product development and why we must consider more than just geometric compatibility when evaluating designs for completeness.

Chapter 3

Chapter three introduces the reader to automotive closures, specifically to side doors. A broad review is completed to facilitate the reader's understanding of subsequent chapters. Also, a view of how requirements are created and a description of the predominant requirements in automotive closures is included.

Chapter 4

In chapter four we get to the more technical and analytical section of this thesis. We begin with an overview of the design structure matrix and then explain the process used to create the case study DSM. We also discuss the art of architecting a DSM so it may be used as a very effective corporate knowledge base.

Chapter 5

Chapter five represents the closures created DSM and the real world application of the requirements integrated design process. We present the DSM created for this thesis and discuss the nature of its content and then its structure.

Chapter 6

Chapter 6 runs a Monte Carlo simulation on the As-Is DSM to create a benchmark. The average process completion time and standard deviation resulting from this simulation are used to measure the effectiveness of process improvement proposals. We discuss a number of the process improvements suggested by Auto OEM's subject matter experts (SMEs). These improvements are then incorporated into the simulation. Conclusions from our research are summarized.

Chapter 7

Reflecting on this work completed in this research, chapter 7 shares practical insights gained during this research. Chapter 7 ends with suggestions on what future work may completed.

Chapter Summary

In this first chapter we discussed the prevailing dynamics in the automotive industry today. A relationship was developed to make the case for decreased product development cycles and increased quality and early detection of design failures. Having this background information, we developed and presented the motivation for this work. Next, we reviewed some of the relevant prior work that has been done in this space. Subsequently, our problem statement evolved and was presented. Finally, the overall progression of this thesis was presented on a chapter by chapter basis. Collectively, this chapter warms us up and sets the stage for the following chapter where we discuss in more detail some of the activities of automotive product development.

Chapter 2: Automotive Product Development

Chapter Introduction

In this chapter we look at automotive product development from a high level perspective. We see how the systems engineering systems-V model is followed then we discuss the significant role of CAD and CAE in modern product development processes. We then construct a framework for one view of product development that considers product specifications as part of the overall product development process.

Automotive Product Development

At Auto OEM, the process of developing an automobile follows the systems engineering systems-V. This is a top-bottom and bottom-up process where key inputs are identified at the vehicle level first. The overarching process is explained in four key phases: define, design, verify, and launch. These four phases, and the system-V model, are captured in Figure 4 below.

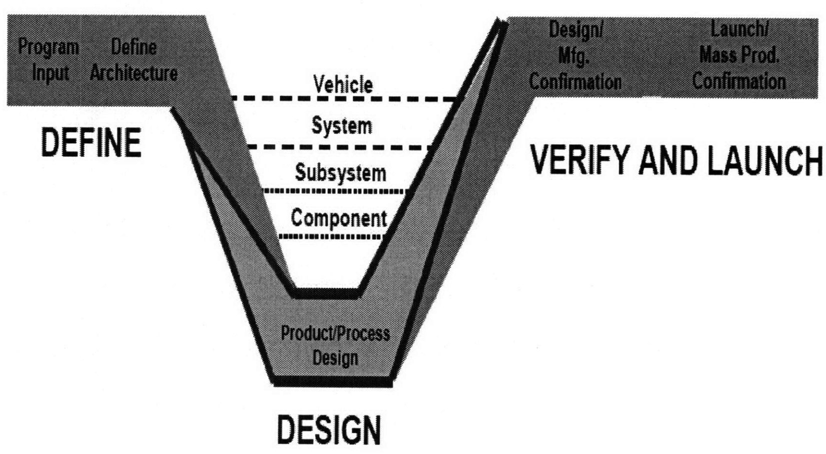


Figure 4: The generic systems engineering systems-V and the four phases of product development. Source: Author

During the define phase, key inputs such as customer, regulatory, and corporate wants and needs are identified. They are then formalized into specifications and requirements that are used through-out the design process. These specifications are then used again during the verification and validation portions to validate designs and ensure designs have been created to meet customer and corporate wants and needs. Finally, the launch phase takes the work accomplished in product development and integrates it into manufacturing facilities for full-on production.

Complex systems such as automobiles and airplanes cascade specifications down through systems and subsystems and components with the intent that the system level requirements get captured at the system's most detailed level - components. This effort occurs in the top-bottom, or left hand side of the systems-V. The bottom-up process is represented on the right hand side of the systems-V where verification and validation starts at the component level and progresses through sub-systems and systems to end at the vehicle level.

Unique to the notion of product development, relative to other process-centric activities, is the notion of iteration. By this we mean to say that in product development processes the product is evolved in an iterative manner. Designs are created and evolved, often times changing many times as they mature. Through out this evolution more information about a given design is generated and this knowledge contributes to the further development and maturation of the design. An example of such activity would be the analysis of an automobile relative to Federal regulations such as side impact (known as Federal Motor Vehicle Safety Standard 214 (FMVSS 214). This requirement is set in the very early development of a vehicle; however, because there is a large amount of data required to assess the design's performance, a great amount of design work occurs prior to completing the first assessment. Once the study is completed, more knowledge of the design's performance ability to meet the requirement is known and it is very likely that changes will be made. This example is illustrative of the left hand side of the systems-V where design and study work is common. Thus, in Figure 4 above, information exchange on the left hand side of the V not only progresses down and to the right but also back up the left hand side.

It is well known that about 80% of a product's lifecycle costs are determined by the end of detailed design (McManus, 2005), or the bottom of the system-V model. Prior to the end of component design, the primary cost expenditure is represented in the salaries of the product development staff, or labor. However, after detailed design, funding is released for the purchase of tools and fixtures. Obviously, change becomes exponentially more expensive when "real" things have to be modified rather than CAD lines and surfaces. A simple example, such as moving a hole, illustrates this very clearly. To move the location of a hole on a sheet metal component in CAD may only take a few minutes, or just a few dollars worth of effort. However, to execute that same hole move in tooling will require a much larger investment in cost and time and typically will cost between $3000 and $5000. Another example, such as having to soften corners radii on a die may cost $10,000-$20,000 if done after the virtual phase of product development. Cost and time implications are further amplified when there are cascading effects through out a system. Additionally, it is not just the cost of downstream rework that is important to point out. Equally important is recognizing that the downstream costs are basically established in the initial design. Initial design decisions such as where a hole will be and how it will be manufactured have costs associated with them. These decisions will effect how cost is impacted later in the process.

Computer Aided Design (CAD) and Computer Aided Engineering (CAE) in Automotive Product Development

In modern automotive product development there is a heavy reliance on both CAD and CAE. CAD technologies today allow the construction of parametric models that can be shared virtually all over the world. Points, lines, and surfaces are created in the virtual world using tools such as Catia or Ideas, and these CAD files become the single source that the many sets of product development eyes view. The visual presentation capabilities of CAD models help to communicate complicated designs with more effectiveness to both technical and non-technical personnel. Furthermore, given the advances in Product Information Management (PIM) tools, designs can be available to team members regardless of their geographical location.

Just as CAD revolutionized the drafting world, CAE has revolutionized the testing world. In CAE, models created from CAD files and consisting of a mesh of squares and triangles are used to simulate real world effects virtually. Advances in CAE have led to a significant reduction in prototypes of all types; in some cases, physical prototypes have been completely eliminated.

Design Completion

At some point in product development the design is completed, but *how* does one know the design is complete and how does one know *when* the design is complete? We view the product development activity as a very large problem solving organization - questions are asked and the actions of product development seek to answer these questions.

We propose the notion that a design is complete when all questions regarding it are answered to satisfaction. This notion is reinforced by Browning (Browning, 2002), "The goal of PD is to produce a product recipe to requirements or acceptance criteria." Questions come in a variety of forms and some examples are listed here:

- How much does the design weigh?
- What size is the design?
- Does the design meet all safety requirements?
- Is the design manufacturable?

- Does the design meet the functional requirements?
- Does the design meet all geometric requirements?
- How much does the design cost to manufacture?

These questions are formalized and manifest themselves in variety of formal specifications. These more formal and complete specifications will be addressed in the following chapter.

With the advent of CAD, some product organizations have taken a very close look at attempting to meet requirements very early in the development process. In fact, Auto OEM has a very extensive process for monitoring CAD progression in the early development stage up until the point CAD drawings are released. This process tends to be system to system focused, looking at geometric compatibility between two or more systems. CAD progression is monitored by tracking a list of these system to system interfaces and how the designs are performing relative to these requirements. For clarity, a simple example follows. To understand the scope and definition of what a system or subsystem is in automobiles, we must first understand how vehicles are typically partitioned. Figure 5 below depicts a typical vehicle partitioning; we must go to level one to get to a reasonable level of decomposition for subsystem to subsystem interfaces.

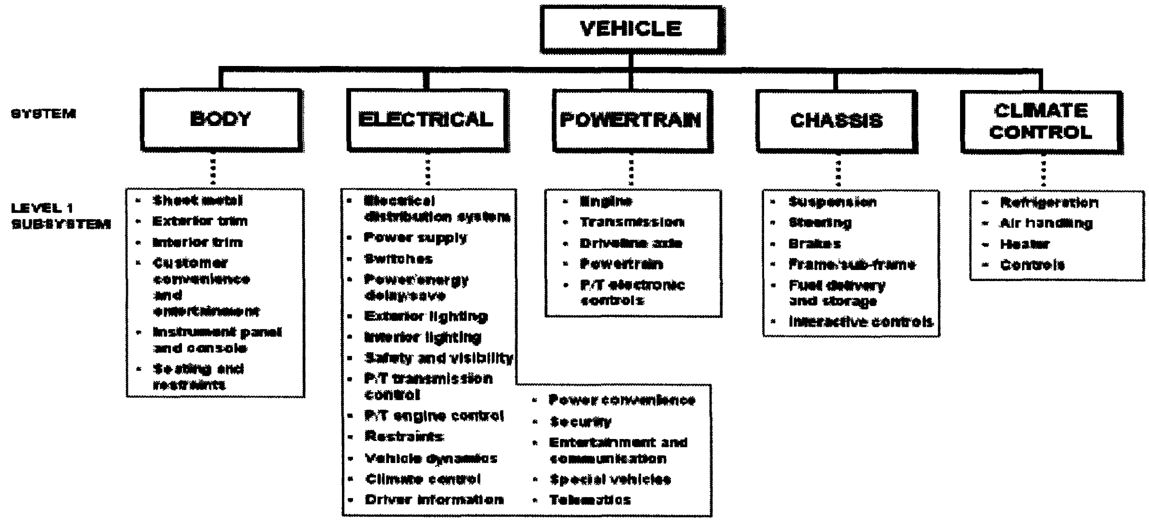


Figure 5: First level partitioning of an automobile; Source: Author

From the partitioning above we can easily see that subsystem to subsystem interfaces present themselves at Level 1. This is consistent with how geometric compatibility is assessed; checks are made between sheet metal and trim, for example. Specifically, there will be a geometric requirement that all trim attachments are aligned to their respective holes in sheet metal. Of course, there are also interfaces between the boxes shown in the Level 1 partitioning. For example, there are sheet metal requirements that facilitate electrical connectors and wiring.

Although this process is a good step in the right direction of improving CAD quality and giving greater consideration to the importance of interfaces (rather than just component focus), it still misses a very large portion of what makes a design complete. In other words, one could successfully pass all the geometric requirements and still end up with a poor performing design. Why is this? Quite simply, geometric compatibility is only one slice of a much larger pie that constitutes maturity. A mature design will not only be geometrically compatible with its surrounding environment, but it will also meet all of its requirements.

These other requirements come in many forms, including durability, safety, NVH, cost, and weight. Additionally, each requirement may use a different verification method. For example, a durability test requires CAE or Testing. Similarly, Safety and NVH requirements can only be assessed through CAE and physical testing. Furthermore, CAD can be used to check geometric compatibility and component weight. In the context of these two views, a design is not only complete, but the *right* design is complete. The author's first hand knowledge at Auto OEM confirms that achieving near-perfect geometric compatibility does not translate into designs that meet other (functional, manufacturing, etc.) requirements.

To illustrate the difference between geometric compatibility and other types of requirements, we will discuss the grab handle that is mounted on the inside of a vehicle, usually along the roof rail, so customers can assist themselves in entering and exiting the vehicle. As the grab handle is developed, it is created within the context of its surrounding environment. Geometric compatibility ensures the grab handle is within defined packaging zones and that it is not interfering with its surrounding environment. In this case, the environment typically consists of the headliner, structural sheet metal, and any wiring or airbag components that may be in the area. Geometric compatibility will measure the distances to these surrounding sub-systems and ensure no interferences are present and that intended mating surfaces (typically at attachment areas) are in fact mating. However, this evaluation does not tell us how the grab handle will perform. Of course, the grab handle has functional requirements such as being capable of withstanding a certain load while not exceeding a defined deflection (i.e. strength) and being able to do this a specified amount of times (i.e. durability). Tests and/or simulation are needed to answer these additional questions; because geometric compatibility will not answer these additional functional requirements, it is in itself not enough. These tests and simulations are typically done by a separate group than the one that initially created the design. Dedicated testing and CAE organizations will perform these functional assessments and provide their results back to engineers and designers to incorporate into the design. CAE studies can be completed fairly early in the development process to help in guiding the design. However, if physical testing is required, it will come much later in the development process, most likely up the right hand side of the systems-V. As a result, companies employ CAE capabilities as early as possible in the development cycle to minimize costs associated with physical tests and the increased potential for costly rework generated from a far down stream test.

Chapter Summary

In this chapter we introduced and reviewed automotive product development in the context of the systems engineering system's-V. We also discussed the very significant role that CAD and CAE have in modern automotive product development. Next we introduced the notion of design completeness and knowing when a design is complete by understanding that all specifications are met. This idea was further enhanced in the context of CAD compatibility requirements and how there is a need to address more than just environmental compatibility. In the following chapter, automotive closures will be introduced so the reader can have a better understanding of the later and more technical work in this thesis. Also, automotive requirements are discussed in greater detail to facilitate better understanding the more technical work later in this thesis.

Chapter 3: Automotive Closures and Requirements

Chapter Introduction

Although not essential, it will be helpful to understand the closures environment to better comprehend the framework for integrated requirements in product development that is further discussed in the following chapters. Therefore, in this chapter we introduce the basics of the closures environment and then discuss some of the many and varied requirements that effect closures development.

Automotive Closures as a Complex System

Automotive closures embody what a complex system is. Edwards Demming describes a system as, "A system is a network of interdependent components that work together to try and accomplish the aim of the system." The closures system - or as most customers refer to them, doors – represent a major subsystem of the overall vehicle. This system alone has hundreds of components of different material types, many of them moving, which complete the system. Closures have a number of aims. Some of them are occupant safety, vehicle stiffness, attributes (such as wind noise and closing efforts), and styling.

From the customer's viewpoint, not only do they see the door and its contribution to the overall aesthetics of the vehicle, but they also interact with the door. They grab it to open it and push the buttons on it to activate door locks and window operation, they rest their elbows on the interior trim of the door and they even draw judgments as to the quality of a vehicle based on how the door sounds and feels when they open and close it.

From an OEM's view, closures also represent a complex system. From the very beginning of vehicle development assumptions must be made as to the architecture of the closures. In the case of side doors, there are three predominant architectures in industry today. They are header reinforcement, separate header, and integrated header. Figure 6 below shows the major

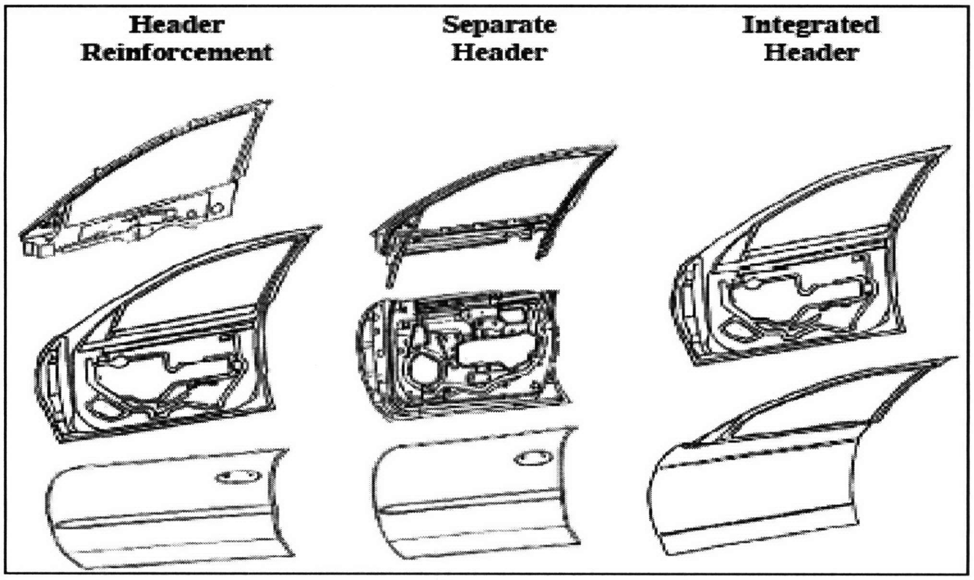


Figure 6: Three industry standard side door architectures (Note: only sheet metal components shown). Source: CAR, 2001

differences in an illustration and Appendix A includes further illustrations. This initial architectural decision has far reaching impacts as each of these different architectures has different assembly processes and related reliability. Likewise, the different architectures have different requirements that must be managed and design to. Furthermore, the different architectures have different capabilities in meeting requirements. The assembly of hundreds of components must be carefully managed to ensure quality standards are met. During the design phase of the system's engineering-V closures represent the design embodiment of both styling and manufacturing. This adds an element of complexity as a greater number of organizations within a company must interact to develop the system. The styling organization creates the look of the customer viewed surfaces (referred to as Class A surface) and other organizations, such as assembly, manufacturing, engineering, and testing all have their input. To add further complexity, many components in the doors system are designed and manufactured by full service suppliers.

Automotive Specifications

As we have previously discussed, automotive products have to meet a variety of requirements. In this section we will discuss one approach to creating and sorting requirements.

At Auto OEM, customer wants and needs are very important and as such a method is used to translate these wants and needs into specifications that can be used by the product development staff. This process begins by reviewing a list of standard vehicle level attributes (some examples of attributes include cost, safety, and handling) and first assigning priority to each one. Assigning a priority assists in performing tradeoff studies. Also, a competitive assessment is done and determination of the degree of competitiveness that is desired. Auto OEM makes conscious effort to consider competitive information, brand identity, and customer wants in this process. The outputs from this process are the targets for each of the attributes.

System Specifications

System specifications are just that, system or subsystem level specifications that describe the "what" a product must meet. An example SS would be the specified capability the vehicle must meet in such things as acceleration or stopping distance. A generic list of SSs is stored in an online database. This generic list is filtered to a shorter list for each program depending on the program assumptions. Program specific SSs are managed and monitored through their own online database.

Geometric Requirements

Geometric requirements are requirements that are employed primarily during the virtual design process. These types of requirements typically look at subsystem to subsystem relationships such as hole and attachment alignment. They are monitored and tracked in an online database on a per program basis. There is some confusion surrounding geometric requirements and their use at Auto OEM. Geometric requirements are generated based on other requirements; for example, if manufacturing says it needs a 15mm diameter hole to access a nut, there will be a geometric requirement to check this. In this light, the geometric requirement is really the verification method employed, it is not the requirement itself.

Regulatory - Safety

These requirements represent the minimum acceptable criteria as set forth by the federal government. It is common for Auto OEM to set targets much more stringent than those set by the federal government. These requirements include Federal Motor Vehicle Safety Standards (FMVSS). FMVSS 214 and 206 are most relevant to closures development; these are side impact and hinge separation during impact, respectively. Appendix B includes a thorough description of each of these.

Corporate Requirements

Corporate requirements are those that are set internally. They are often used to manage requirements across different brands and in this vein affect cross-company alignment. Other significant corporate requirements are those that govern architectural decisions. Typical door architectures were already shown above; however, there are many other architectural decisions that can be made. Other examples include body-on-frame vs. unibody and major assembly sequence assumptions. Although there was a significant push in the late 1990's to drive global architectures, consensus from all business units was difficult to achieve and the effort fizzled out. Corporate requirements can be found in different online databases and some are stored in the same location as SSs.

Manufacturing Requirements

Manufacturing requirements come in a variety of forms and often ensure designs are created that facilitate assembly in production. An example would be a minimum requirement for a tooling clearance hole. A specific example of such a requirement is illustrated in the required tooling clearance on the inside of the body shell at the B-pillar. In vehicles with four door architectures, the rear door hinges may be attached to the B-pillar and may require a nut runner to pass the through the inside of the pillar to perform assembly at the manufacturing facility. In such a case, a minimum clearance hole must be designed in the inner side of the B-pillar to ensure the nut runner can be used properly. Many of these requirements can be studied during the virtual development phase.

Manufacturing requirements are not always assembly related. There are a number of requirements that help to ensure components are manufacturable. Examples include how close punched holes may be to each other and targets for depth of draw for sheet metal components.

Styling

The styling organization has many requirements as well. These include minimum crowns and sweeps of surfaces as well as gaps between panels. Furthermore, the styling organization also has a number of package related requirements. Commonly referred to as hard points, these requirements represent common points shared between engineering and artists in the design studio. Examples of hard points are the front windshield surface, binocular vision, and engine plug points. These hard points all represent areas of the vehicle that are typically difficult to coordinate between engineering and design studio. The hard point requirement specifies the progression design must follow. This progression has stated tolerances that are checked at different gateways. For example, a 3-D point can be designated as an engine plug point representing the upper most package of the engine. The studio is not allowed to develop a surface that intrudes on this point and the engine team is not allowed to develop engine components that would intrude beyond the plug point and into Class 1. Most interestingly, at Auto OEM, the styling organization's studio engineering team will study requirements that are often studied by other organizations as well, creating a duplication of effort. An example is the binocular vision band study. This particular study is often completed by studio engineering, engineering/CAD, and packaging. This subject is covered in greater detail later in this paper.

Packaging Requirements

Packaging requirements typically involve the evaluation of occupants and their interaction with a number of vehicle facets. Examples include occupant ingress/egress, vision bands, and reachability to switches. The packaging group work to set these requirements and show them in the CAD environment. Although these requirements are set by the package group, design engineers - in their design execution - are the individuals responsible for meeting the requirements.

Although there are additional requirements, this subset covers the most predominant ones used by Auto OEM. As one can see, many of these requirements are stored in different locations, making for a management nightmare and often a hindrance to those who must use them.

When organizations have such a vast and large variety of requirements, there are a number of things the organization should look out for. An effective means of communicating cross-organizational requirements must be employed. In chimney structured organizations, communication between chimneys (or functional groups in the auto industry) is not always as great as it should be. Thus, there is a danger that some stakeholders will not be aware of requirements that must be met. Some times requirements are revised; this can be due to a change in technology or simply to increase the standard. Again, a good system must be used to cascade the modified requirement to the PD community.

Requirements by themselves are not significant enough. Requirements must be viewed in light of their verification method as well; therefore, each requirement has an associated verification method. In this context we can see the relationship between requirements, design, and verification. Requirements are determined at the outset of a program, the product development staff works to create designs to meet these requirements and verifies their design using the prescribed method. It is worth noting again that the early detection of failure to meet requirements is critical to reduced cost in product development. Again, if design failures can be captured during the virtual focused phase of design, they are less costly to fix.

The author recognizes that a comprehensive discussion of *all* requirements is beyond the scope of this document. Instead, we have captured a variety of requirements to illustrate the diverse set of requirements that must be made. These different requirements will later be seen in the development of the Design Structure Matrix for this paper. It is important to recognize that different types of products will have different types of requirements. What is key is that the answer to product completion is the satisfaction of all these requirements. It is an "all or nothing" game – Meeting hundreds of manufacturing requirements to build a door and body assembly means little if the customer has difficulty with ingress/egress. A holistic view is required.

Chapter Summary

In this chapter we discussed how automotive closures are a complex system. We also provided more detail on some of the predominant requirements that closures systems must meet and identified some of the troubles which accompany the current requirements storage and distribution activity. We acknowledge that a exhaustive list of closures requirements is beyond this work, instead trying to capture the predominant requirements needed to facilitate construction of the Design Structure Matrix later introduced in this paper. In the following chapter we will review the Design Structure Matrix as an effective tool for modeling a complex process.

Chapter 4: Design Structure Matrix

Chapter Introduction

Chapter four represents the more technical and analytical side of this thesis. In this chapter we begin with a brief introduction and review of the design structure matrix, or DSM. Attention is given not only to the analytical application of the DSM, but also to the art and process of creating a DSM. In this effort we show that when constructed correctly, the design structure matrix can serve as a very effective corporate knowledge base, in addition to its strengths in representing complex processes. This chapter serves as preparation for subsequent chapters where the DSM will be applied to this research project.

Basics of the Design Structure Matrix

The design structure matrix (DSM) is a process modeling tool that was created in academia but has seen wide spread use in multiple industries. One of the great benefits of this tool is that it displays information in a compact, visual, and analytically advantageous format (Browning 2001). DSMs take the basic form of a square matrix with identical rows and columns. Marks in the matrix represent interactions between tasks. There are two prevailing ways to interpret these marks. In the first method, a mark indicates a column feeds a row. The second method is just the opposite, marks represent rows feeding columns. <u>It is important to point out that in this work, the first method is practiced; thus, columns feed rows.</u> A simple illustration, shown in Figure 7, shows a typical DSM.

TASK	A	B	C	D	E	F
A						
B	X					
C		X		X		X
D			X		X	
E						
F						

Figure 7: A simple DSM; Source: Author

The basic interpretation of this model is to view task relationships as serial, parallel, or coupled. Serial tasks are those that occur subsequently. Recalling that rows feed columns, we see that Task A feeds Task B, this is a serial relationship. A parallel relationship is shown between Task E and Task F – they can be completed at the same time. Finally, couple task relationships are those that feed each other in what many refer to as "chicken and egg". This relationship is shown between Task D and Task C. These basic relationships are often easier to grasp when illustrated in a network graph. The three basic relationships are shown in graph format in Figure 8 below.

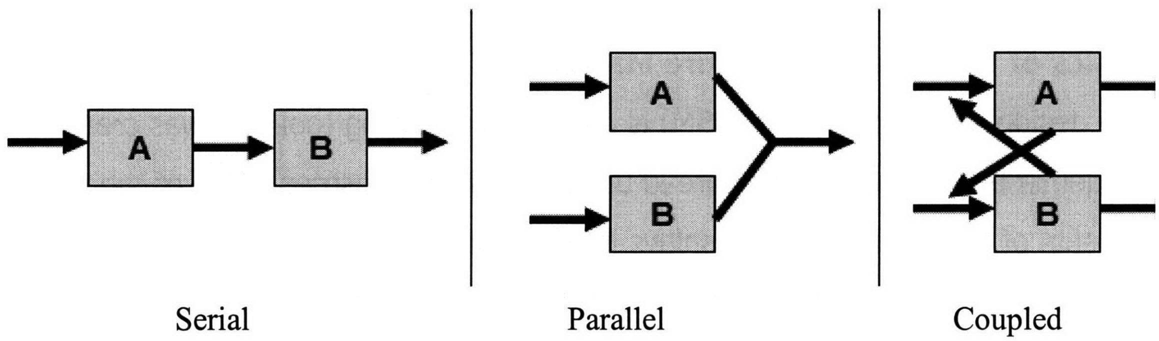


Figure 8: Three types of task relationships; Source: Author

One of the strongest attributes of a DSM is its ability to replicate feedback in a process. Most complex projects exhibit feed forward and feed rearward dynamics. This task interaction is represented in the DSM when marks are shown to the right of the center diagonal. In Figure 7, we see that three tasks have feed rearward relationships: Task D feeds Task C, Task E feeds Task D, and Task F feeds Task C. We can qualitatively consider the degree of impact feed rearward task have by seeing how far away from the center diagonal the mark lies.

DSM has been applied to a number of different situations and they are classified by the nature of what fills the columns and rows. As described by Tyson

Browning (2001), DSM is classified as either static or time-based. Static classified DSMs include Component Based DSM and People-Based DSM; where as Time-Based include Activity-based and Parameter-based DSM. These four types and their common application are shown below in Table 1.

Type of DSM	Typical Application
Component or Architecture	Used for modeling system architectures based on components and/or subsystems and their relationships.
Team-based or Organization	Used for modeling organization structures based on people and/or groups and their interactions.
Activity-based	Used for modeling processes and activity networks based on activities and their information flow and other dependencies.
Parameter-based	Used for modeling low-level relationships between design decisions and parameters, systems of equations, subroutine parameter exchanges, etc.

Table 1: Types of DSMs and their application; Source: Browning (2001)

Besides visual interpretation of the DSM, algorithms exist for the reordering of rows and columns to create more an efficient process. The most basic approach to reordering, called partitioning is described at the DSM website (www.DSMweb.org):

> 1. Identify system elements (or tasks) that can be determined (or executed) without input from the rest of the elements in the matrix. Those elements can easily be identified by observing an empty row in the DSM. Place those elements in the top of the DSM. Once an element is rearranged, it is removed from the DSM (with all its corresponding marks) and step 1 is repeated on the remaining elements.
>
> 2. Identify system elements (or tasks) that deliver no information to other elements in the matrix. Those elements can easily be identified by observing an empty column in the DSM. Place those elements in the bottom of the DSM. Once an element is rearranged, it is removed from the DSM (with all its corresponding marks) and step 2 is repeated on the remaining elements.

3. If after steps 1 and 2 there are no remaining elements in the DSM, then the matrix is completely partitioned; otherwise, the remaining elements contain information circuits (at least one).

4. Determine the circuits by one of the following methods:

- Path Searching
- Powers of the Adjacency Matrix Method

5. Collapse the elements involved in a single circuit into one representative element and go to step 1.

Excel based macros and public domain matlab scripts are available for partitioning. Further work can be done with DSM via simulation to determine average process duration and its standard deviation. There are already very well written papers which discuss the simulation side of the DSM, some of these include:

- Assessment of Rework Probabilities for Simulation Products Development Processes Using the Design Structure Matrix (DSM) by Yassine, Whitney, and Zambito
- Modeling Impacts of Process Architecture on Cost and Schedule Risk in Product Development by Browning and Eppinger
- Using the Design Structure Matrix to Streamline Automotive Hood System Development by Zambito
- Using the Design Structure Matrix to Estimate Product Development Time by Carrascosa, Eppinger, and Whitney

Design Structure Matrix Construction Process

Since DSM has had wide spread use, a mature process has evolved and is considered the standard process to be followed. The process, as described by Qi Dong (Dong, 1999), follows:

1. Define the system and its scope.
2. List all the system elements.
3. Study the Information Flow between System Elements.
4. Build a matrix to represent the information flow and decide on a suitable measure to be used in the DSM.

5. Give the matrix to the engineers and managers to comment on and use.

We propose one additional step, which should be completed in conjunction with or immediately after step 1 from the process above: Assemble the team. In this step key management leadership should be identified. This is critical so participants understand that management supports the project. Additionally, a basic review of DSM practices may be needed to educate team members on the benefits and limitations of a DSM oriented project. While are stakeholders are present, this is an appropriate time to create the scope and timeline of the project This step cannot be overlooked because today's PD staffs tend to be over worked. In addition to being overworked, many employees have seen different "process initiatives" come and go, thus engineers may view the effort of a DSM project as the flavor of the month. Assembling a team and acquiring proper management support assists in declaring the seriousness of the DSM project effort and in prioritizing engineers work. It should be noted, that smaller DSMs may not require such support as the time required of engineers will be much less. As a rough gage to assist in guiding the determination of what constitutes a large DSM project, we can consider three projects from the automotive industry that were presented in the Systems Project Management class at MIT in the Fall of 2006 by guest speaker Tony Zambito. See Table 2 below.

	Project 1	**Project 2**	**Project 3**
Scope	Sub-system	System	Vehicle
Project Duration	3-4 months	2 months	1 year
Team Size	6	12	53
DSM Size	50x50	125x125	1600x1600
Whitney Factor	5	5.6	6.0 (estimated)

Table 2: DSM project scope, duration, and team size considerations
Source: Zambito (2007)

Additionally, practitioners may want to consider the granularity of their models. The commonly referred to Whitney Factor metric will help in this effort. Dr. Whitney has extensive and deep experience in the construction and evaluation of DSMs and has determined through experience that most DSMs average about 6 marks per row; this is referred to as the Whitney Factor (Whitney, 2007). Although not explicit, one may want to gage the granularity of their model based on the Whitney Factor. DSMs with a Whitney Factor well above 6 may indicate too much granularity. Another consideration driving the Whitney Factor can the method of data collection used. Dong found that collecting data via interviews resulted in an average Whitney Factor of 6, whereas reviewing documents for data collection resulted in an average Whitney Factor of 3.

Design Structure Matrix as a Knowledge Base

The DSM represents the embodiment of a great amount of information and can very effectively serve as a knowledge base. However, this does not come free as it requires a conscious effort to architect the DSM for these purposes.

Traditional DSMs typically have one column and one row which represent a given task. This method of capturing the process is effective for partitioning and simulation exercises. It is concise and serves those intimate with the DSMs creation well. However, we can add additional columns to facilitate a greater understanding of the tasks and deliverables involved in the process. By adding these additional columns we are able to capture more information in the DSM and its usefulness is extended well beyond the understanding of only those associated with its creation. We propose an additional spreadsheet, duplicate to the one used for partitioning/simulation, but also containing the additional information.

In this additional spreadsheet, there are a number of different categories one can create to enhance understanding and usefulness of the DSM and these are only limited to the imagination of those creating the DSM. To illustrate this idea, we first look at a typical DSM task. The following task was taken from Noor's

	A	B	C	D	E	F
1		1	2	3	4	5
25	Sec. seal force (CLD, bulb dimensions, operating energy range of contact to over-travel)		1			
26	Cheat in header and pillar (CAE based deflection from material rigidity, shape and seal force)		1			
27	KBE validation of seal interference and seal gap (including jumps and smoothness)					
28	Mass and centre of gravity (sheet metal, trim, packaging)	1	1			
29	Centre of gravity relative to hinge centre line	1	1			
30	Door system stiffness (hinge pillar and door sheet metal)	1	1			
31	Latch type & characteristics (spring, height, size, energy, over travel, pawl vault)					
32	Latch-striker interface (sliding vs. non-sliding or no wedge)					
33	Latch bumper durometer (for chucking protection)					
34	Latch bumper interference (latch-striker load deflection curve)					
35	Latch angle relative to hinge centre line in design (particularly rear door)					
36	Latch Component Test (travel and efforts, misalignment)					
37	Nominal position of striker to Z centre line (vertical placement of tapping plate) and y-axis		1			
38	Hinge walk (movements during hinge mounting actions such as driving bolts)					1
39	CAE validation of door blowout (front A/B pillar at 100 mph and 20 degrees yaw)					
40	Rear door fixed glass division bar design		1			
41	Air trapped upon compression (between primary and secondary seals)		1			
42	Body/Door sheet metal stamping quality and assembly tolerancing (GDT from rule 1294)	1		1	1	
43	Door inner/outer pannel assembly variation (hemming, inner to outer master locators, weldi	1		1		
44	Hem adhesive before and after paint					
45	Seal gap DVA		1	1		
46	Striker position variations (in Margin and Flushness DVA)		1			
47	Hinge setting process variations (in Margin and Flushness DVA)		1			1

Figure 9: Task descriptions typical of traditional DSMs; Source: Noor, 2007

Closures DSM (Noor, 2007) which is shown below in Figure 9: Task #44: *Hem adhesive before and after paint*. In this instance, the DSM was a parameter based DSM that was created with an eye on the engineering parameters required to develop a door as well as heavy consideration given to attributes. This DSM is very effective in capturing the work of the engineering team that participated in it and in sharing this information with other closures engineers; however, its usefulness begins to expire beyond this scope. We can add further usefulness to this task by including a column which identifies who is responsible for the task. Additionally, a column which describes the task deliverable is of great value. Simply stating, "Hem adhesive before and after paint" doesn't have much value in communicating what is being done. A closures engineer may understand the task based on this brief description; however, this interpretation will vary based on the engineer's experience. We look to eliminate this variation by offering the suggested columns.

For much larger and complex DSMs that cross numerous organizations and commodities, we can add further columns to help in the communication of the DSM. For example, if a DSM is constructed that has tasks from a number of different organizations; we can add a column in the spreadsheet that will identify which organizations are responsible for which tasks. Furthermore, if we have a hybrid DSM, we can add a column that easily differentiates the types of tasks. By adding these identifiers, the usefulness of the DSM is heightened and personnel with common Excel skills can easily maneuver through the matrix to find their pertinent information. A perfect real-world example is that most organizational managers will often ask, "Out of all this work to do, what am I accountable for?" A simple sorting in a column for Organizational Responsibility will deliver the answer.

Furthermore, tasks themselves should contain enough detail to be useful. Giving careful consideration to how tasks are recorded not only helps in down stream usefulness, but also in the stages of data collection. The matrix created for this work has both tasks (i.e. things to do) and explicit bits of data (i.e. things). The difference is subtle but important and while interviewing to gather data for this matrix, having clear definitions was helpful in communicating what was being asked for. Additionally, the task deliverable should be recorded in a column adjacent to the task column. To illustrate these ideas, we apply them to the DSM created for this work in the next section.

Architecting the Integrated Requirements Closures DSM as a Knowledge Base

To architect the DSM used for this work, we first revisit the notion of product development. The purpose of product development is to produce information that is then studied and verified for completeness against requirements. Given this model for product development, we categorize PD efforts as either Inputs or Completion Criteria (C^2). Inputs are *things* that are generated in PD, such as CAD models, CAE models, theme surfaces, engineering parameters, or even decisions. C^2 represent the requirements that must be met, these manifest themselves in studies being executed and verified to meet requirements as well as in the maturity of CAD models.

The company under study has a mature PD process that breaks the virtual development process into four major phases, referred to here as Design 0 (D0), Design 1 (D1), Design 2 (D2), and Release Date (RD). Prior process engineering efforts identified the completeness of a design through definitions that were applicable during each design phase. For example, during the D0 phase designs were considered complete by declaring them to be Defined. The definition of Defined is: *An item that is Defined is one that is preliminarily specified, with or without a tolerance/zone, and available to the customers of the data. Defined items may change through the feasibility process.* A more specific example of Defined is applied to the task of creating the Door Latch during D0:

- Task: Closures – Front Door – Latch Location Defined
- Deliverable: Latch location *defined* in 3D CAD
- Interpretation: The position of the front door latch is shown in CAD, however its location may change due to future feasibility work

In our model of PD we do not explain design completeness through definitions. Instead, we explain design completion by the design's ability to meet its requirements for a given design phase. Allowing this definition gives us the ability to design an integrated requirements-driven process. It allows us the ability to know what inputs are needed to perform verification to requirements and it allows us to design the process accordingly.

To create the DSM used for this paper, the team first identified a number of descriptive columns and how information would be reported in each column. Figure 10 shows the column headings. The first column identifies whether the line item is an Input or a Requirement; again, this allows for quick filtering of information for down stream users. The next column, Input/Req Type, explains what type of Input or Requirement the line item represents; examples include models (CAD or CAE), decisions, and different types of requirements. Because some line items were pulled from a previous database, the next column tracks the current line item to its original document. Next, we have the task itself. Because the text from this column will be copy/pasted into the Excel file used for partitioning and simulation, we have a very explicit way to define the text. Figure 11 shows the "formula" for the text, differentiating between Inputs and C^{2}, and Figure 12 shows the pro forma in practice with a snapshot from the DSM created for this work. Using this pro forma ensures all pertinent information is also captured in the Excel file used for partitioning and simulation. After the Task column is the Deliverable column which includes a brief description of what is generated from its respective task. A column is included to identify the Verification Method associated with any C2. C2 that do not have defined verification methods are easily discovered if there is an empty box.

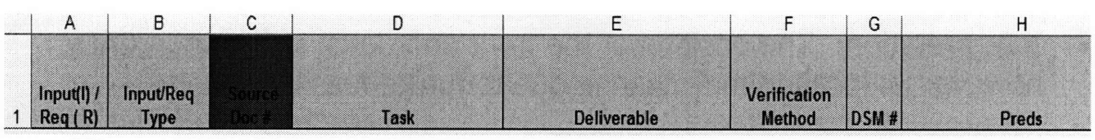


Figure 10: Column headers in the integrated requirements closures DSM; Source: Author

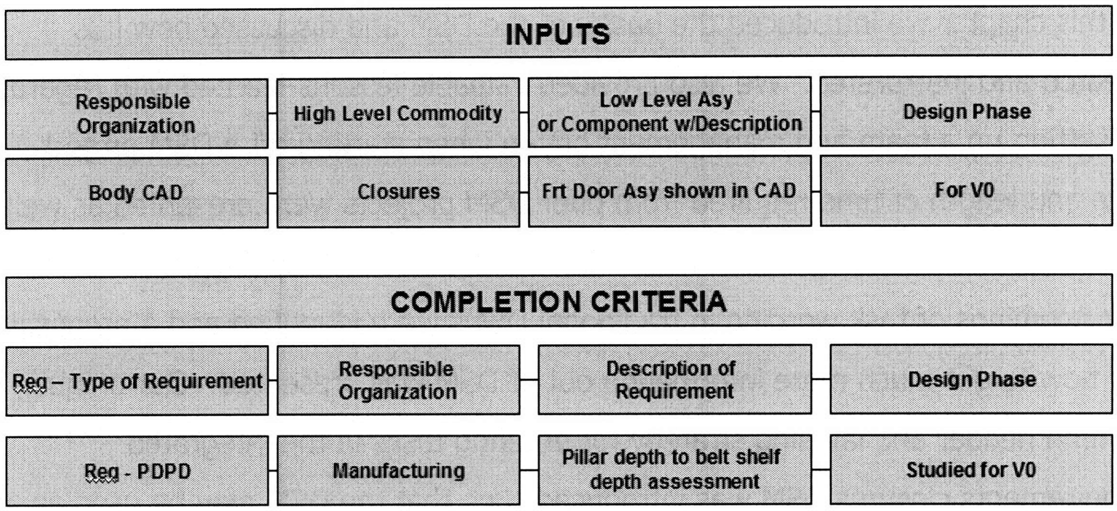
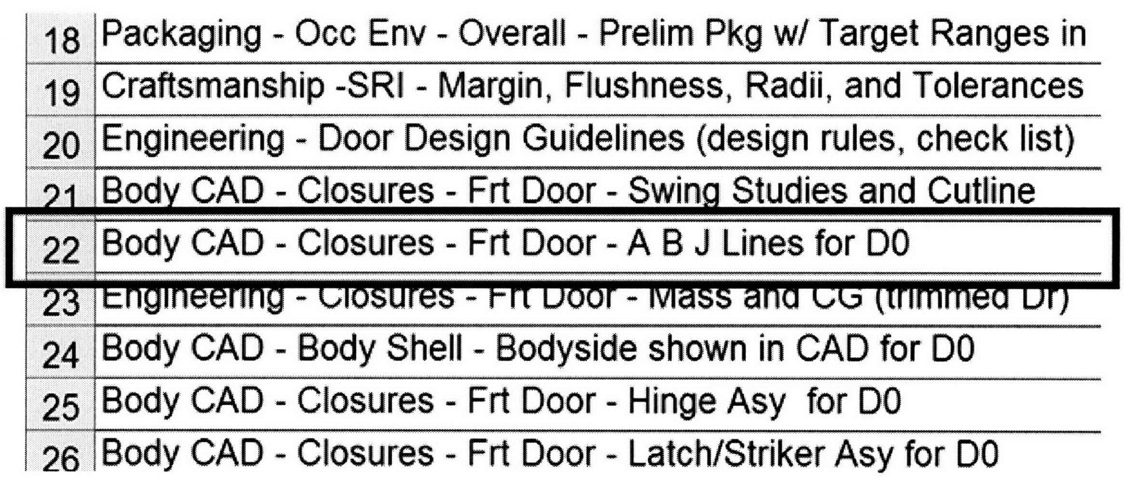


Figure 11: Pro forma for DSM task column with one example for Inputs and one for Completion Criteria; Source: Author



18	Packaging - Occ Env - Overall - Prelim Pkg w/ Target Ranges in
19	Craftsmanship -SRI - Margin, Flushness, Radii, and Tolerances
20	Engineering - Door Design Guidelines (design rules, check list)
21	Body CAD - Closures - Frt Door - Swing Studies and Cutline
22	Body CAD - Closures - Frt Door - A B J Lines for D0
23	Engineering - Closures - Frt Door - Mass and CG (trimmed Dr)
24	Body CAD - Body Shell - Bodyside shown in CAD for D0
25	Body CAD - Closures - Frt Door - Hinge Asy for D0
26	Body CAD - Closures - Frt Door - Latch/Striker Asy for D0

Figure 12: A sample Input task from the DSM created for this work Source: Author

Chapter Summary

In this chapter we introduced the basics of the DSM and discussed how it is created and interpreted. We also provided valuable lessons-learned with regards to setting up a team and management buy-in when starting off a DSM project. Size and length of time required from prior DSM projects were presented as well as a way to check the granularity of a DSM using the Whitney Factor. Shortcomings of task wording in traditional DSMs were identified and a proposal for how to get much more information out of DSM was presented. Our unique column header and labeling strategy for entering tasks in the integrated requirements closures DSM was introduced such that the DSM may be used as a more effective knowledge base. In the next chapter we dive deeper into the details of the closures integrated requirements DSM and get into the analytical side of this DSM.

Chapter 5: DSM Application

Chapter Introduction

Prior chapters in this thesis have laid the foundation for understanding this chapter. We have discussed automotive product development, closures as a complex system, the role of requirements in product development and design completion, the basics of how Design Structure Matrices are created and used, and how the DSM for this work was created not only for process improvements but also as a useful knowledge base. We now apply this foundational knowledge to the creation of an integrated requirements closures DSM.

Integrated Requirements Closures DSM Scope and Purpose

Model Scope

The scope of this model can be defined along a number of dimensions. These include time, Inputs, and Completion Criteria. First, we consider the dimension of time. The product development cycle for Auto OEM can be said to start in what is commonly called the Pre-Program phase, which gives way to the Component Design phase. The Component Design phase ends with the release of detailed drawings for designs and from this point forward physical builds begin. Finally, the vehicle is launched at the assembly plant. Although computer based tools are used through out the process, the Pre-Program and Component Design phases exploit virtual and computer based tools to the greatest. The scope of our model is focused on the Component Design phase.

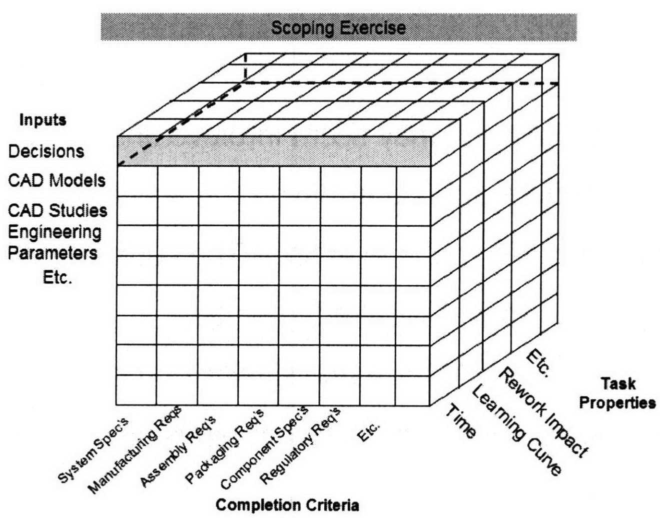
As mentioned before, Auto OEM breaks down virtual development into four sub-phases: D0, D1, D2, and DR. D0 represents heavy interaction between the program team and the development of the Class 1 surfaces. At this point data in the engineering and design community is primarily shared electronically, usually in the form of CAD type math models, and the focus is to develop the Class 1 surface aesthetically and from a point of feasibility. The primary feasibility studies done at this point are manufacturability and assembly of the Class 1 surfaces, packaging, and ensuring the Class 1 surface is representative of the program assumptions. As D0 ends and D1 begins, with the Class 1 surfaces fairly well defined, focus changes to further refinement of the vehicle structure and the comprehensive use of CAE to assist in evolving designs to meet program targets. D2 is very similar to the D1 phase. Class1 changes are expected to be minimal and the structure is further defined. CAE work done during the end of D1 is also communicated to the PD team and this knowledge is incorporated into designs during D2. The DR phase represents the completion of designs and what is most commonly referred to as "pencils down" for the engineers and designers. The DSM created for this work captures both D0 and D1 as well as the relationships between the two. It is observed that including D2 and DR increases the size of the DSM substantially, and because D2 and DR are very similar to D1, their inclusion to this work does not add substantial insight to the intended purpose.

From a content perspective, this DSM is focused on the development of the Front Door Closures system. Therefore, we include the Front Door Assembly, Hinge Assembly, Check Assembly, Striker Assembly, Latch Assembly, Side Glass, Windshield, and the surrounding geometry in the form of the Body Side. The term Assembly is not used to describe the process of putting together components; it refers to the already assembled components as one entity.

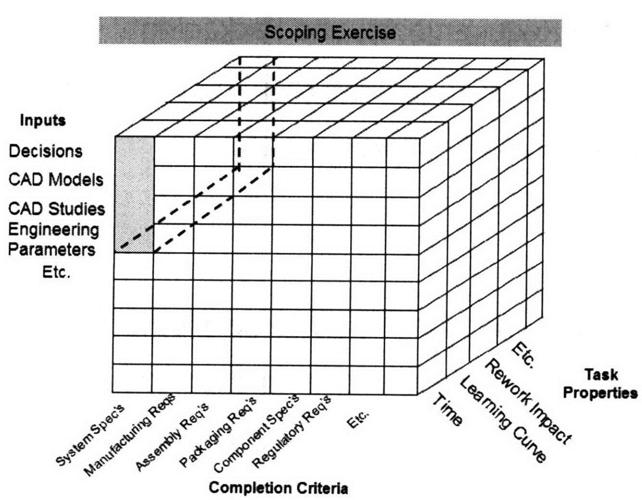
To incorporate completion criteria into the DSM, we have included a variety of different requirements. These requirements include, System Specifications, Assembly Requirements, Packaging Requirements, and Manufacturing Requirements. Again, we recognize this to be in incomplete list, as an exhaustive list of requirements is out side the scope of this paper. None the less, we feel that by including a variety of requirements we will capture the intended purpose.

3D Scoping Cube Exercise

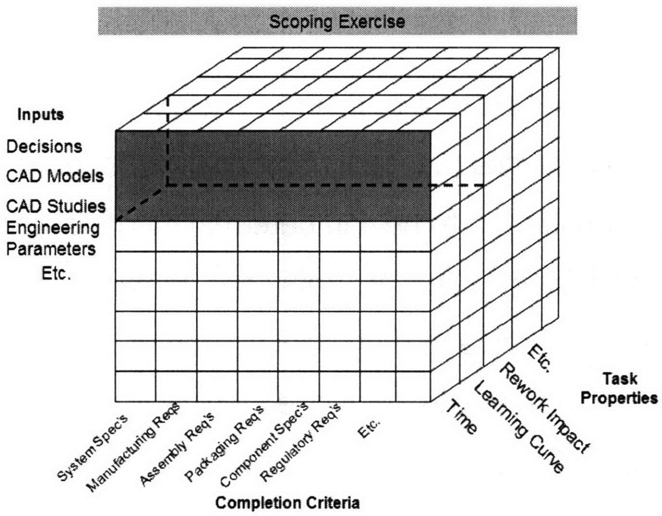
Because the scope for this project is defined along a number of axes, including Inputs, Completion Criteria, and Task Properties (i.e. task duration), we found it useful to develop a multi-dimensional illustration to help in determining our scope. We refer to this illustration as the 3D Scoping Cube. We apply our primary dimensions of interest along the axes of a cube and then select different volumes from the cube to determine the scope. An example, as applied to this work, of the 3D Scoping Cube exercise follows in Figure 13.



Interpretation:
Here, the yellow box represents the scope, including a variety of C2 and Task Properties. However, by limiting the variety of Inputs, this scope falls short of capturing the many types of information generated in product development.



Interpretation:
Selecting the volume shown to the left does capture the variety of information generated in PD but falls short of capturing the many completion criteria that may have to be met by this information.



Interpretation:
We can easily see that this volume easily captures a variety of PD information and a variety of completion criteria. Additionally, the volume captures a number of different task properties. This best captures all three dimensions in PD: Inputs, Completion Criteria, and Task Properties.

Figure 13: The 3D Scoping Cube; Source: Author

Although the 3D scoping Cube shown in Figure 13 may seem intuitive and unnecessary, we discovered it to be a useful tool in determining the scope, particularly since we had a few people involved in the exercise. The cube helped to visualize what each person was speaking to and resulted in better communication. Ultimately, the exercise resulted in a scope similar to the final cube shown in Figure 13. In an effort to represent the different types of information generated within PD, we selected a number of Inputs, including component/assembly designs (CAD files), decisions, and engineering parameters. To capture the variety of Completion Criteria we selected System Specifications, Manufacturing Requirements, Assembly Requirements, and Packaging Requirements. We capture task properties, such as time required to perform a task, learning curve, task sensitivity, and rework impact, as well.

Purpose

The intended purpose of our effort is to demonstrate how to develop a process that delivers product definition (Inputs) to satisfaction (Completion Criteria). We do this by capturing the information created in the PD factory as Inputs, and we measure their completeness with respect to Completion Criteria (C^2), in a Design Structure Matrix. We recognize that work done in PD is done within the context a product development plan (PDP) and we capture this context in our timing scope of D0 and D1. Simulation methods allow us to evaluate the process relative to average process completion time.

Integrated Requirements Closures DSM Construction Process

The DSM was constructed from interviews with subject matter experts, reviewing legacy documents, and first hand personal knowledge of the process. A number of closures Technical Specialists, Engineers, Designers, Integrators, Process SMEs, and Closures Supervisors and Mangers were spoken to. Collectively, the total time spent speaking to people to gather information was in the neighborhood of 120 hours. Considering the resulting DSM has 85 tasks and 385 marks, approximately 20 minutes of interview time was required per mark. DSM practitioners may want to use this as a gage when planning for a DSM project. Although the time/mark metric can change significantly based on collection method (i.e. survey vs. interview) and experience (i.e. first hand knowledge of subject under study vs. no knowledge), existing data indicates practitioners should expect an investment of approximately 20-25 minutes / mark (Whitney, 2008).

Considering the number of marks per task and total number of tasks we can calculate the Whitney Factor. The Whitney Factor for this DSM is 5 marks per row. This number may be considered a bit low, as we would expect to see at least 6 or more given recent closures DSMs. Noor reports his closures DSM as being highly complex and coupled, resulting in an average of 9 marks per row. We believe the fundamental difference in DSM content is what leads to this disparity and that by adding additional completion criteria to our DSM, the average marks per row would quickly converge to a number between eight and ten. For the purpose of demonstrating how to model an integrated requirement driven process, our model granularity will be fine.

In addition to interviews, a number of legacy documents were reviewed. These documents include Auto OEM's PDP, a prior closures DSM project, a prior PDP DSM, existing Commodity Work Plans, and a number of requirements databases. After interviewing and document review, duplicate and out of scope tasks were de-conflicted and removed. This collection effort is represented in Figure 14.

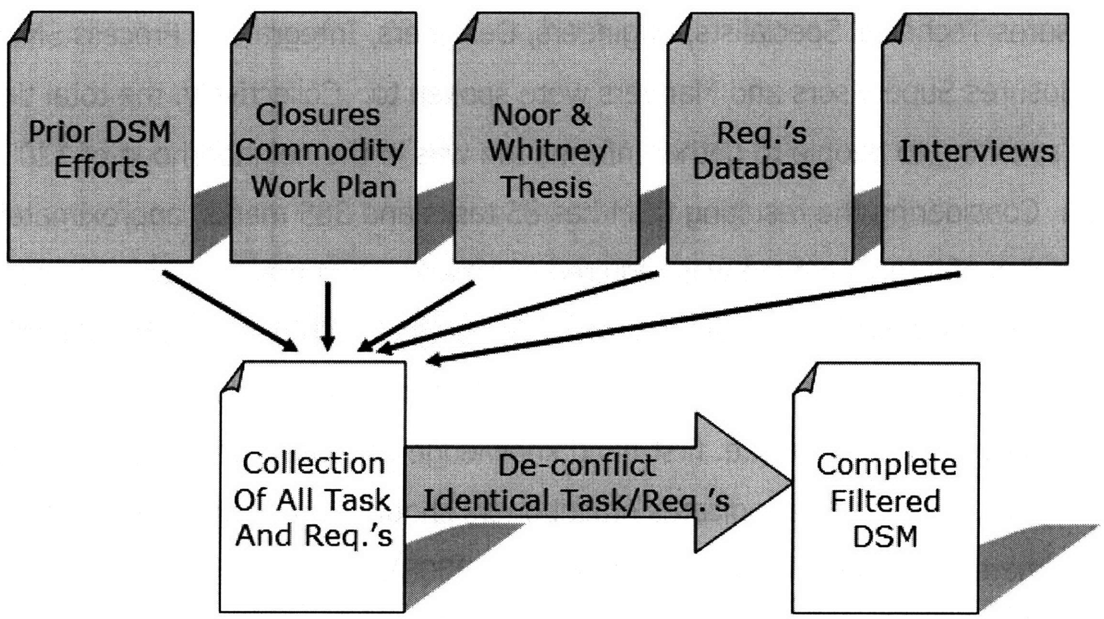


Figure 14: Information Sources; Source: Author

Integrated Requirements Closures DSM Task Content

The format used for entering Tasks was covered in Chapter 4. To better communicate the content of the model, we present a short representative list. Line items in the DSM are categorized as Inputs or Completion Criteria and both can be further classified. It is important to note that in the knowledge base Excel file, the column adjacent to tasks is the Deliverable column, which offers a brief description of the task deliverable.

- Inputs:
  - In the form of a CAD model: Front Door Assembly Cad for D0
  - In the form of a Decision: Full Service Supplier Selected – Door Latch and Striker for D0
  - In the form of a Surface File: Bodyside and Front Door Surface Design for FC 0
  - In the form of Engineering Parameters: Front Door - Mass and CG (trimmed Door) determined for D0
- Completion Criteria:
  - System Specification: SS - Hinge Span for Front Side Doors - Studied for V0
  - Manufacturing Requirement: Pillar depth to belt shelf depth - Studied at FC3 and 4
  - Package Requirement - Studio Engineering - Vision - Binocular Vision - Studied for V0 (reference Appendix B for detailed explanation)
  - Assembly Requirement - Loading - IP (Path, Sequence) studied at V0

Integrated Requirements Closures DSM Discussion

The DSM created for this project contains 85 tasks and 433 marks and is shown in Figure 15. We refer to this DSM as the "As-Is" DSM. We can readily see the process as it has been described at Auto OEM: We can see, identified as Area #1, that it begins with product assumptions, carry over content assumptions, and full service supplier sourcing. There are many additional tasks which occur in the development cycle; however, many of them are out side of the scope of this matrix. Therefore, we go directly from product planning into a Class 1 surface release and following the surface release we begin D0 (shown as Area #2 in

Figure 15). Area #3 represents the feeding forward of information from Completion Criteria studied in D0. The information is being fed forward into the D1 design phase. In Area #4 we see the D1 design phase, which happens to look much like the D0 design phase. This is an expected outcome, as the activities and their relationships are very similar; however, an identical task in D1 has shorter task duration.

After creating and confirming process accuracy, we partition the DSM. Partitioning, covered in Chapter 4, was accomplished using publicly available DSM macros from www.dsmweb.org. We compare, from a high level view, the As-Is DSM to the As-Is-P DSM by comparing Figure 15 to Figure 16, which has the Partitioned DSM. There is little difference between the two and we readily observe they both follow the same macro process patterns. Because Auto OEM has matured their process over many decades and gone through many evolutions of their PDP, this is no surprise. We would expect there to be a fairly large amount of order consistent with the described process. Partitioning has a much greater impact on immature processes (Noor, 2007).

The next logical step is to get into the details of each of the major areas of activity in the process by understanding the interfaces between them. With that purpose in mind, we will now look at the D0 block in greater detail; it is shown below in Figure 17.

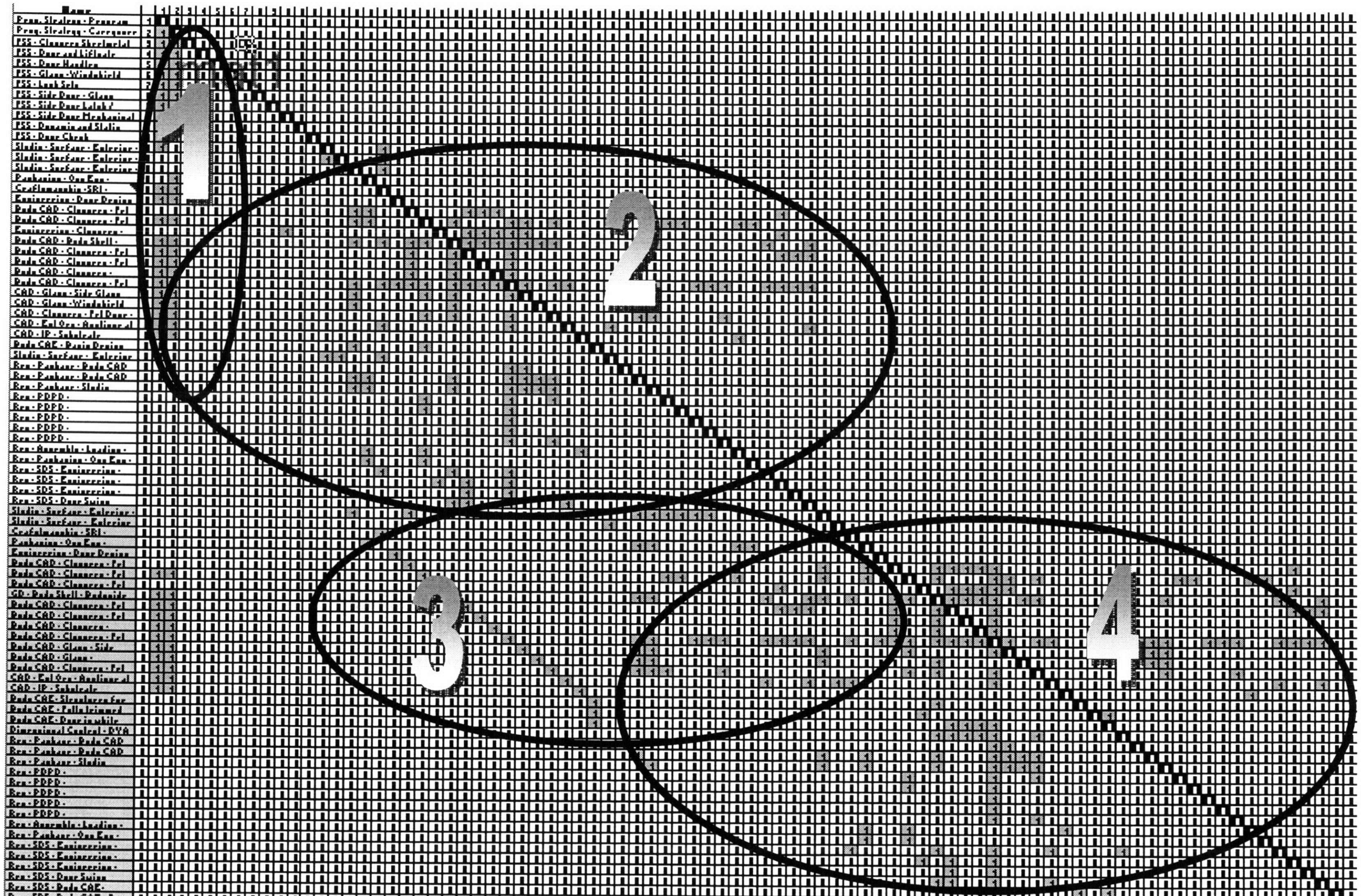


Figure 15: As-Is Integrated Requirements Closures DSM with both D0 and D1. Source: Author

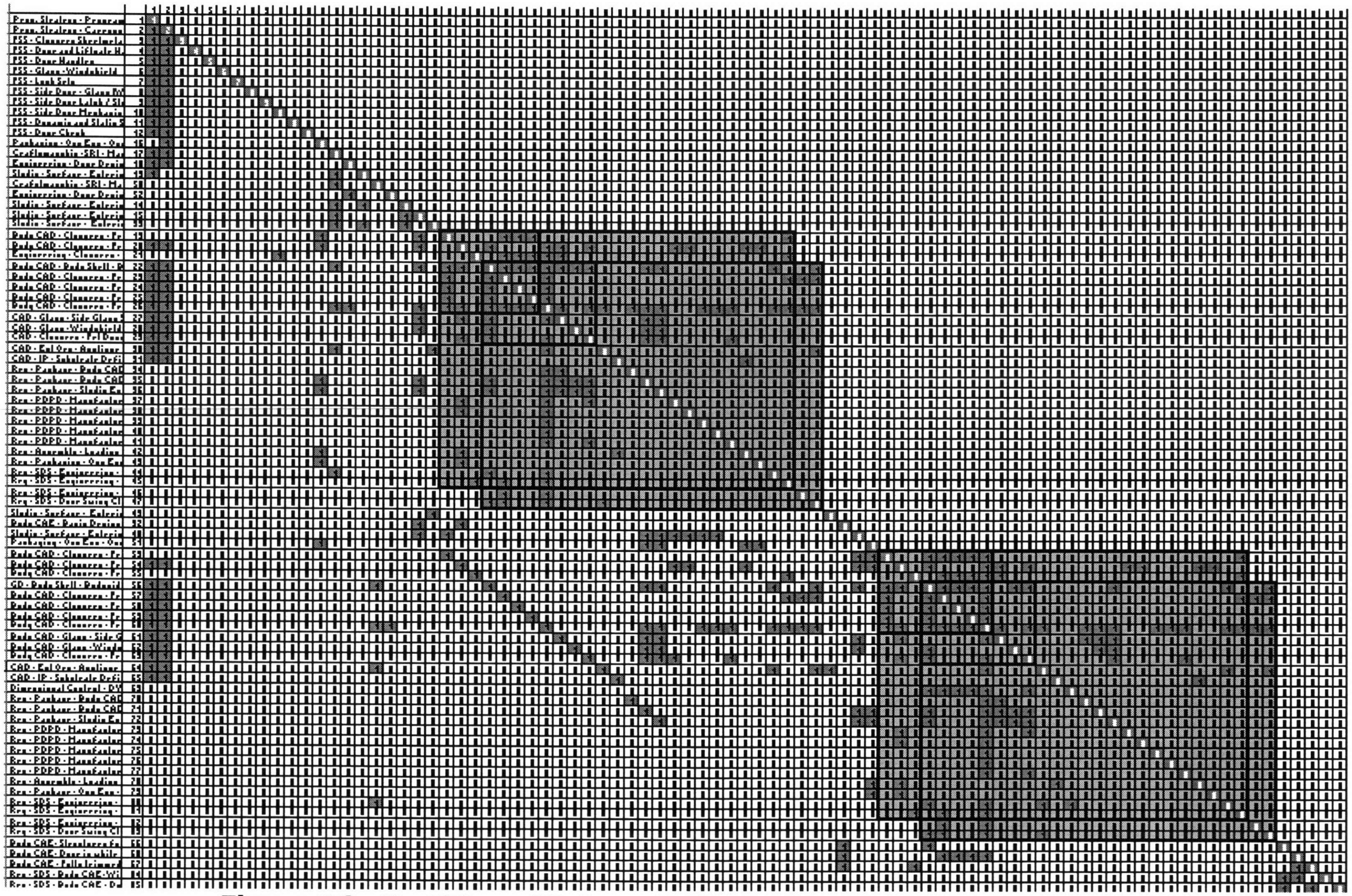


Figure 16: Partitioned Integrated Requirements Closures DSM.
Source: Author

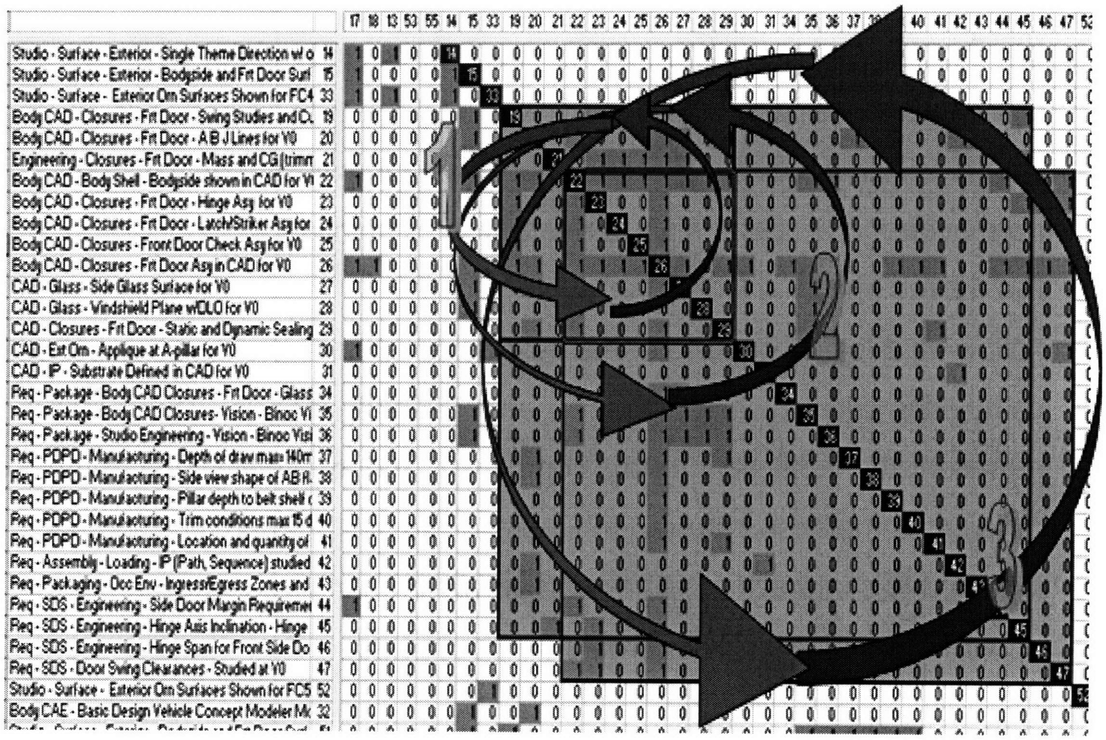


Figure 17: The D0 Design Phase in DSM format. Source: Author

A number of observations can be made about the process at D0:

- After the initial Class 1 surface is released, we can observe a flurry of activity in the form of feasibility studies and CAD component and assembly model development. These studies and the inputs into them are highly coupled. For example the Door Swing Study and Hinge Band Study are heavily coupled with the Hinges, Door, and Latch and Stryker Assemblies. This is an expected outcome and is not unusual – feasibility studies are being performed and the CAD models are being developed nearly in synch, so there is a lot of back and forth information transfer. This is typical and inherent in the design process at this early stage. This observation is highlighted in Figure 17 with the #1 red and blue looping arrows.
- Other inputs into the D0 design phase include the collection of engineering parameters such as design guidelines and program assumptions.

- CAD component and assembly models are heavily coupled. In other words, the Front Door Assembly is heavily coupled with assemblies such as the Latch and Striker Assemblies – this is in addition to their heavy coupling with feasibility studies. These relationships are expected as part of the normal design process. This iterative work is highlighted in Figure 17 with the #2 red and blue looping arrows.
- Following a logical sequence, the next observation is the coupling between completion criteria with component and assembly models. These marks represent the verification of the component and assemblies to a number of different completion criteria. In Figure 17 the #3 red and blue arrows highlight this interaction. These marks are primarily concentrated in the upper right and lower left corners of the process.

Having looked at the nature of activities and relationships within the D0 design phase, we now consider the output of D0 design phase to the next design phase, D1. Observations of feed forward marks from D0 to D1 are:

- Component and Assembly CAD files feed forward to their respective files in D1. D1 component and assembly files do not feed back to D0. This is expected because the continued development of these models is captured in their current file and subsequent files, not in legacy files.
- Some Completion Criteria provide information to in-phase (i.e. D0 Completion Criteria provide information within D0 design phase) components, while others only feed forward, still others provide in-phase and feed forward information flow. This is predominantly by design, and some examples will help to clarify. Some of the Completion Criteria are designed to provide feasibility directly to the Class 1 Surface. Although information gained from these assessments is shared nearly instantly with those responsible for the Class 1 surface, their impact is not shared with the PD community at large until the next Class 1 surface is released. As a

result, these D0 Completion Criteria receive inputs during D0 but their output marks are directed to the next surface release. Other Completion Criteria such as the System Specifications provide immediate feed back the D0 development effort as well as the follow on efforts in D1.

There are some unique content differences between D0 and D1 that are worth noting.

- D1 represents the creation of detailed FEA models, amongst other things, which facilitate the evaluation of data to additional Completion Criteria. For example, the System Specification for Door Torsional Stiffness was not evaluated in D0, but is assessed in D1 due to the availability of FEA models. Thus, we see "new" requirements in D1 that were not present in D0.
- The Class 1 surface is firmly set in this design phase; therefore, many completion criteria that were being evaluated will be discontinued at the end of D1.

The information flows observed above are illustrated in Figures 18 - 20 below.

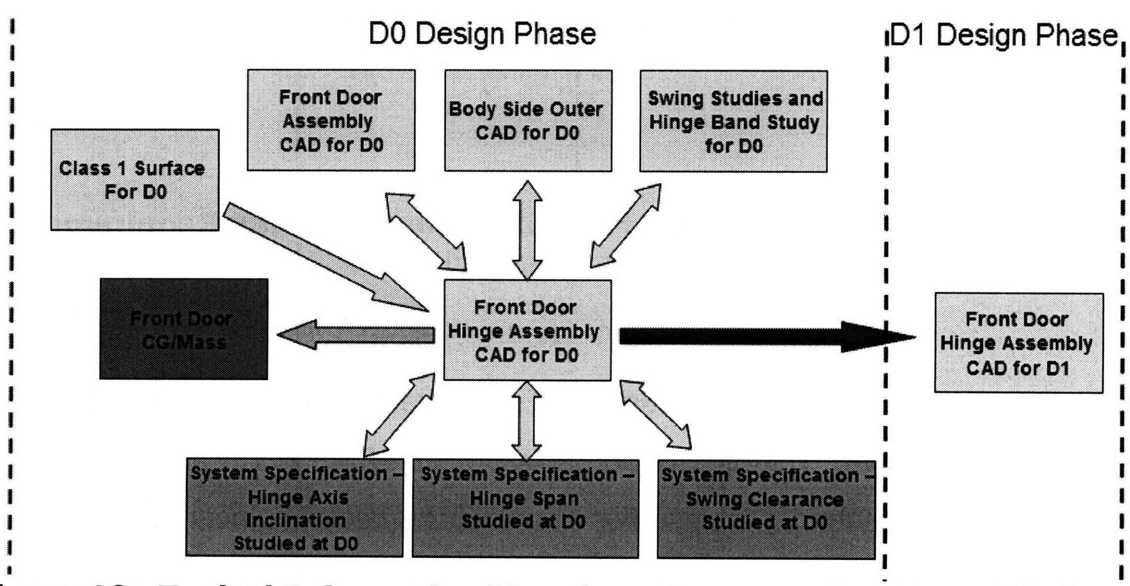


Figure 18: Typical Information Flow for a Component or Assembly Input

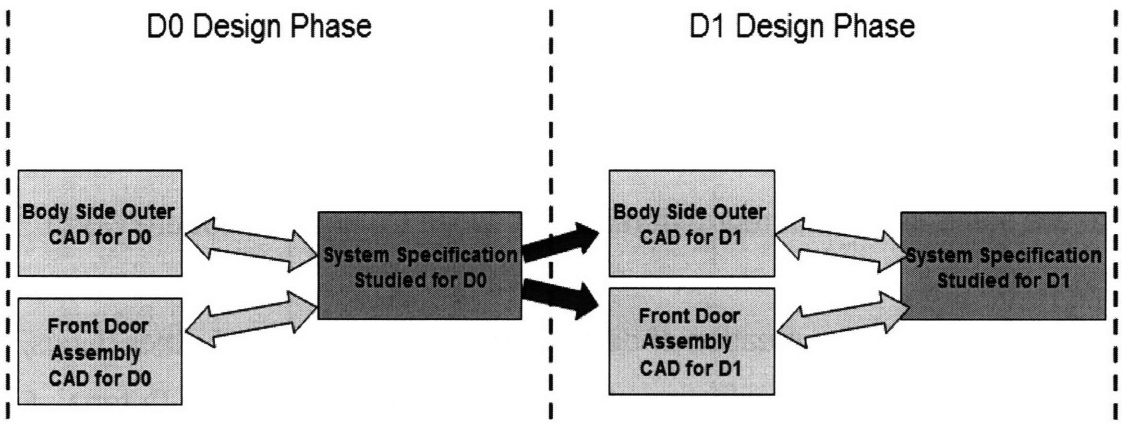


Figure 19: Typical Information Flow for a System Specification Requirement

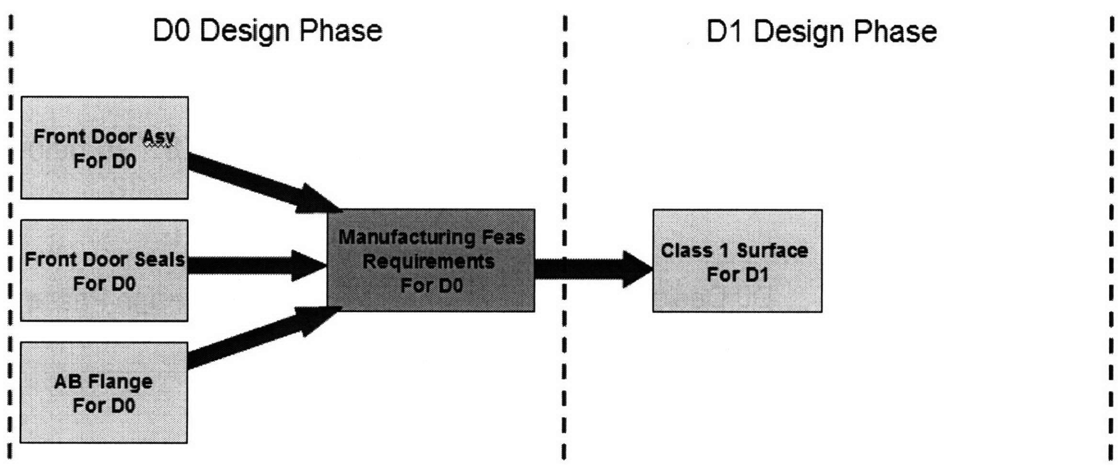


Figure 20: Typical Information Flow for a Manufacturing Requirement for Class 1 Feasibility

Additionally, we discovered that a number of organizations are performing similar, if not identical, studies. The different organizations perform identical studies out of their own development interests. For example, the Body Engineering group may perform an assessment so they may develop structural package space. At the same time, the Studio Engineering group may perform the exact same study so they may assist themselves in packaging Class 1 surfaces. As some completion criteria may be evaluated differently based on an individual's experience or from gathering different levels of Inputs, inevitably different results are obtained. In these cases, the different organizations must meet to understand the differences in their assessments and determine which study is valid. This essentially results in a hidden mini-factory within PD and tasks (such as "de-conflict study results") must be accomplished that are not captured in regular process planning.

In situations where Completion Criteria are evaluated using CAD tools and duplicate efforts are observed, we propose choosing one organization to be the lead. With today's information sharing technologies there should be no excuse for such redundant efforts. One organization could be assigned lead responsibility for the study and it could be shared digitally for others to use. Of course, in practice, this can be difficult to execute because organizations are rarely willing to give up the "authority" they feel they acquire from completing such studies. Also, they may feel that results of a study completed from another organization may not be shared in a timely manner. To address such concerns, the DSM can be used to illustrate the system view and how duplicate efforts can in fact result in mini-factories within PD.

The As-Is and As-Is-P DSMs reveal few surprises. In fact, the heavy coupling of component, assembly, and feasibility work is expected. An important observation to make is that D0 and D1 are effectively de-coupled. The process is designed such that once it has moved into a subsequent design phase, there is no returning to the prior design phase. In practice, we have found that decisions made in a latter design phase can and do ripple their way back to an early design phase. Some of these situations are addressed in the following chapter where we add marks to the matrix to capture the rework and then partition the matrix again.

Chapter Summary

In this chapter we explained the scope, purpose, and construction process used to develop the Integrated Requirements Closures DSM. We also introduced the model itself, explaining what each major area of the model represented and explaining the interactions within major design phases and from one phase to another. With this understanding, we can now begin to further dissect the process and perform analytical work.

Chapter 6: Process Analysis and Simulation

Chapter Introduction

In Chapter 6 we further dissect our DSM. First, we establish a base line to measure the effectiveness of different strategies. We then apply strategies provided by SMEs to improve process efficiency along the line of average process completion time. Additionally, we show how we can observe the process with regards to meeting completion criteria and a technique for measuring Completion Criteria complexity.

Analysis Approach

Our analysis approach will be to first establish a base line model. This model was explained in the previous chapter and is referred to as the As-Is model. Next, we partition this model to find the ideal sequence based on interactions. This is referred to as the As-Is-P model. We also run a Monte Carlo simulation algorithm (Browning, 1998) to determine process time mean and standard deviation.

With the base line model established, we begin to look for areas of opportunity in an effort to reduce the process time and decrease process standard deviation. We will pay particular attention to areas far away from the center diagonal that are involved in an iterative loop. We speak to subject matter experts for creation and guidance in applying strategies. After we develop strategies for restructuring or redefining tasks, we partition and simulate again to measure the effectiveness of our decisions.

As-Is and As-Is-P Model

The As-Is and As-Is-P model were both introduced and reviewed in the previous chapter. We concluded that partitioning had little effect on the As-Is DSM, attributing this to the fact that Auto OEMs PD processes are quite mature. Following simulation methods covered by Zambito (2000) using Information

Variability, Task Sensitivity, Task Volatility, Learning Curve, and Nominal/Best/Worst Task Durations (these properties are explained in detailed on Page 76; we are capable of running a 1000 sample run Monte Carlo simulation algorithm to calculate average process time.

The time available according to the schedule to complete the D0 and D1 time phases is approximately 240 working days, as reported in Auto OEM's process timing sheets. However, the author's own experience is that programs typically run 20%-25% longer than process sheets indicate, which results in an actual time of 300 days (six recent programs were reviewed to arrive at 20%-25%). The As-Is-P DSM simulation average process time and standard deviation are reported below in Table 3. The process time distribution is shown in Figure 21.

	Average Process Completion Time	Standard Deviation
Process Time (based on process sheets)	240 Working Days	-
Process Time (based on actual program experience)	300 Working Days	44 Working Days
As-Is-P DSM	289 Working Days	80 Working Days

Table 3: Process Completion Time for Process Sheets, Six Recent Actual Processes, and Simulation of As-Is-P DSM and Standard Deviation

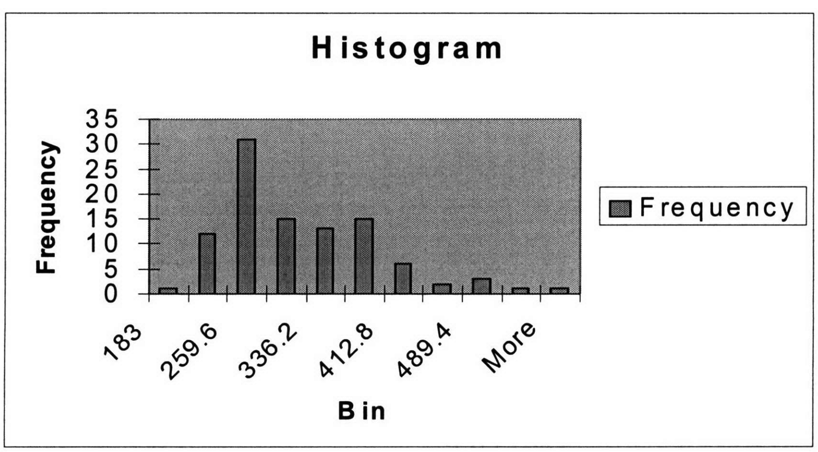


Figure 21: As-Is-P DSM Completion Time in Working Days Distribution. Source: Author

The As-Is-P DSM values in Table 2 were calculated based on a 30% probability of rework assumption. Based on the author's experience on recent vehicle programs, this is a realistic assumption. Because the As-Is-P DSM is incomplete in that not all Completion Criteria have been included (reference page 52 for explanation), it is impossible to correlate the simulation model to the actual time. As a result, this thesis is limited to comparing relative change in process mean time and standard deviation. However, we point out the As-Is-P average completion time is a bit shorter than the Actual Process Time. This offers as a good check in our reasoning, as we would expect the process with fewer tasks to take a bit shorter time to execute.

A Comment on Completion Time (Process Sheets) Vs. Completion Time (Actual)
The disparity between process sheet completion time and actual completion time has been observed in other work looking at Auto OEM's closures development process. Despite the disparity in times, programs do finish ahead of what many simulations have predicted, but at what expense? Engineers report that process sheets expect tasks to be complete the first time they are attempted and don't account for the iterative nature of complex systems development. Unplanned rework is not accounted for in process planning. Often, decisions are made that result in large amounts of rework and in such cases major tasks are reworked and minor tasks are left undone because there is no accounting for the additional time required. Therefore, the expense paid for causing significant rework could be measured in the tasks that must be overlooked and their potential impact.

Assessing the As-Is-P Process for Restructuring

Critical Tasks

To help in our understanding of the process and prioritizing improvement strategies, we follow techniques covered by Noor and created a chart identifying the critical tasks in the process. Figure 22 identifies the top seven critical tasks based on the number of dependencies for each of the tasks. The figure reports the seven tasks which have the largest amount of feeding dependencies. Each of the tasks identified feed the most information to other upstream and downstream tasks. In other words, the task creating the Front Door Assembly in CAD for D0

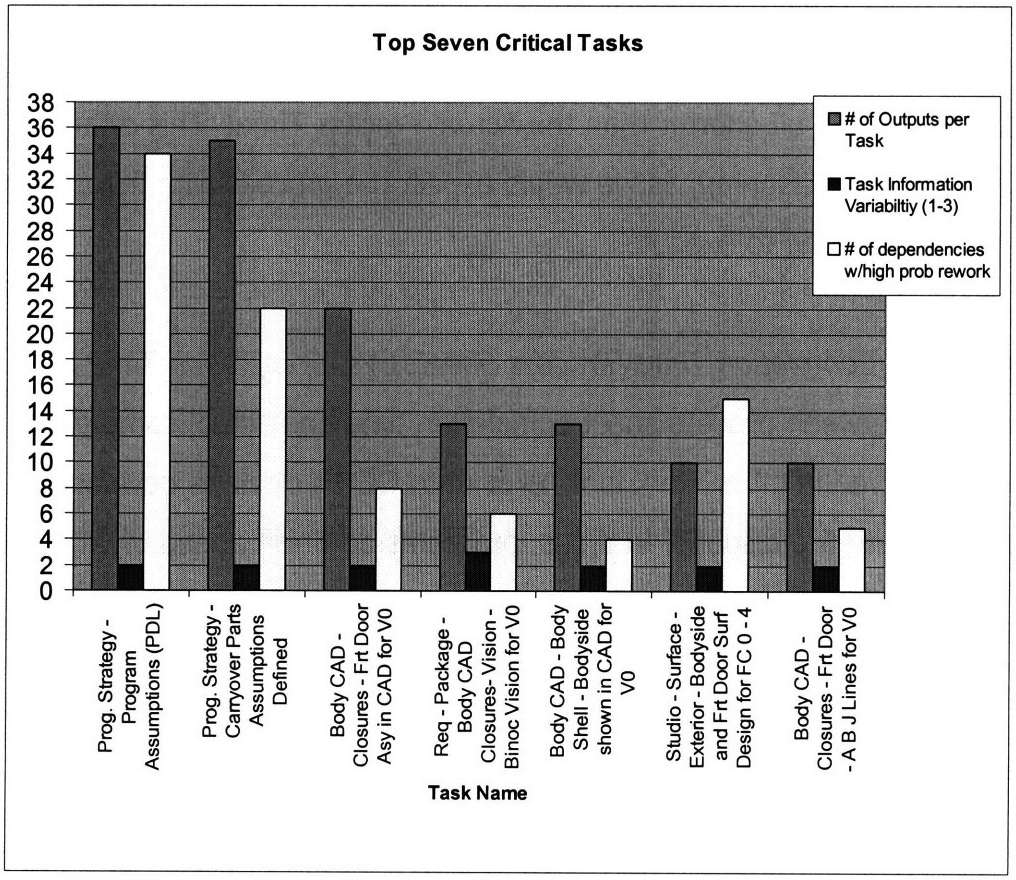


Figure 22: Top seven critical tasks with their respective Information Variability and number of high probability re-work dependencies.

feeds 22 other tasks. Because these tasks affect so many other tasks, their importance is very high. It is important to try and deliver these tasks on time and with high quality so as to limit their effect on the many other tasks they provide information to. We also include each task's Information Variability, a measure of a task's likelihood to change after it has been completed. Put another way, Information Variability can be described as how stable the deliverable from a given task is. Finally, this chart also contains the number of high probability of rework dependencies (Task Volatility of 6 or 9) for each of the top seven critical tasks.

In concert, we can use the task attributes in Figure 22 to help in prioritizing our process restructuring efforts. We look not only for tasks that affect a large number of other tasks, but also for tasks with a high impact based on probability of rework. With these things in mind, we can develop strategies to lower Information Variability or to lessen dependencies probability of rework.

Class 1 surface delivery is highlighted as a task with a fair amount of relationships and a high number of high rework probability dependencies. So, if the Class 1 changes, a large number of tasks will be affected and the degree of that effect could be great. There are a few directions we could move in to improve this situation:

- Reduce the likelihood of the Class 1 task deliverable to change. This strategy attacks the Information Variability of the task.
- Reduce sensitivity of tasks receiving the deliverable from the Class 1 task. This strategy attacks the task property Task Sensitivity.
- Increase Task Duration for the Class 1, allowing more time to mature the Class 1 task deliverable.

We could simply allow the Class 1 task more time to get the task deliverable correct. This would lead us to the assumption that with more time to complete the deliverable, the task's Information Variability would decrease. Because there is immense pressure to reduce product development time, extending task duration is not the first choice made if there are other options. In this case we can reduce the probability of rework felt by tasks receiving the deliverable. When the Class 1 surface file is released to the PD community, it is incorporated into a number of

CAD component and feasibility models, in addition to being assessed to Completion Criteria. The component files are very sensitive to the reception of the new surface, and a fair amount of time is spent just incorporating the new surface. The sensitivity comes from the quality of the surface released from the studio – often times it is does not meet the quality levels required for incorporation into component files. CAD surface files typically measure quality based on surface tangency and surface gaps. We suggest the addition of an intermediate task, Generate Intermediate Class 1 file. The studio will still release their Class 1 surface; however, instead of feeding directly to all other component files, it will feed an intermediate file where quality issues can be quickly addressed before being incorporated into many different files. Thus, we look to add one additional (short in duration task) to effectively eliminate the sensitivity to change that would normally be present. Figure 23 shows the current process and Figure 24 shows the process proposal as described above.

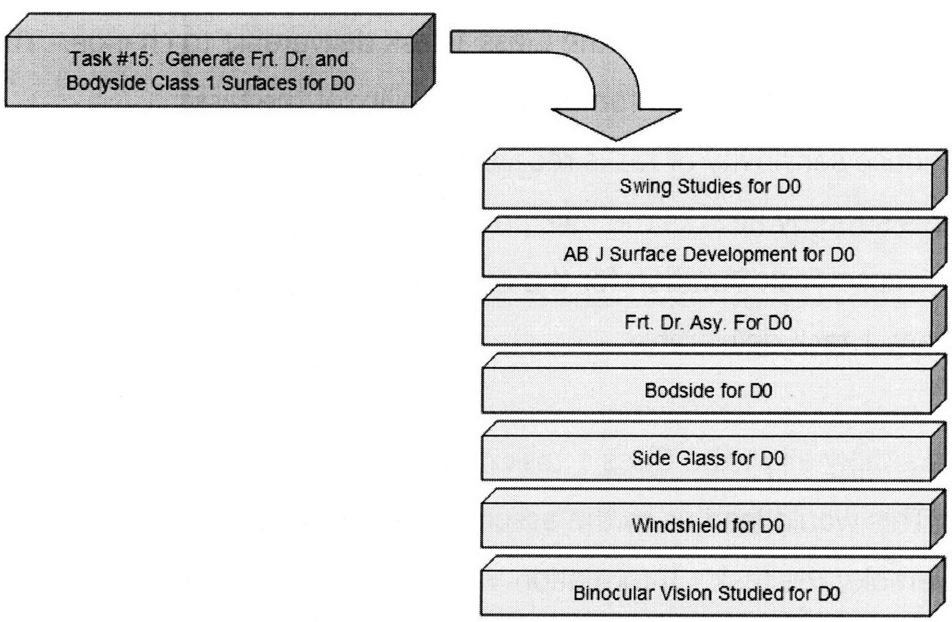


Figure23: Current process – Class 1 surface feeds numerous tasks with high sensitivity to the Class 1 surface. Source: Author

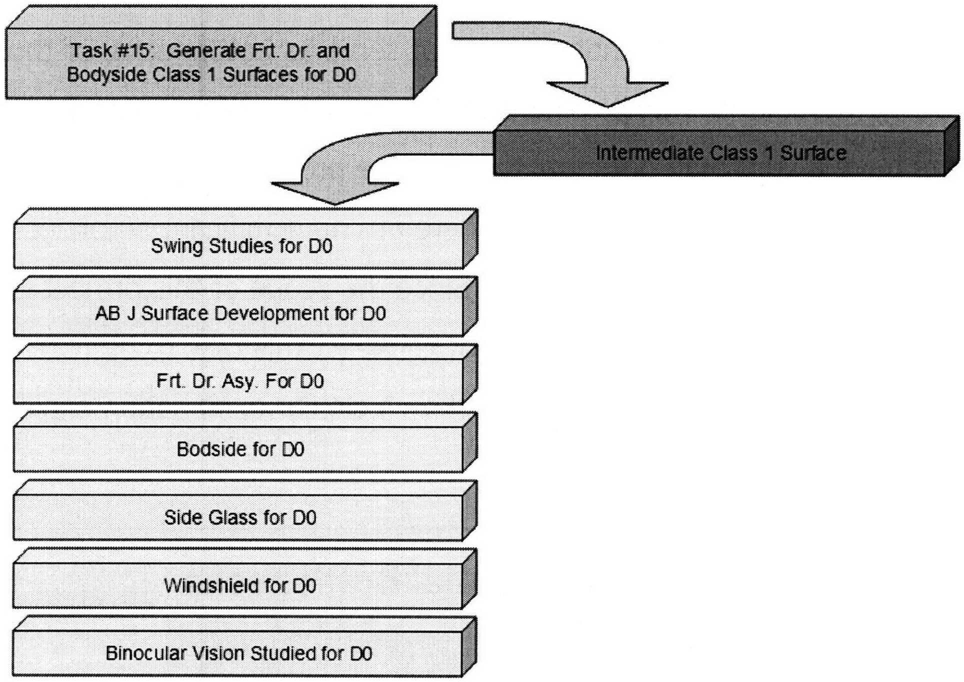


Figure 24: Introduction of a new short duration task which decreases sensitivity to follow on tasks. Source: Author

The results of adding the one short duration task to reduce multiple down stream sensitivities are shown in below. Again, we report the effectiveness of this strategy relative to the benchmark As-Is-P DSM:

- Net change in average process time: 18% Average Process Time Reduction
- Net change in process standard deviation: 40% Standard Deviation Decrease

Information Bottlenecks

We can also use the DSM to identify tasks that serve as bottlenecks, or those that require a large number of inputs. The significance of identifying such tasks is that these tasks have a very high potential to delay the process since they are waiting on the deliverables from so many tasks. There are no significant surprises revealed with regards to the bottlenecks. Since the scope of this project is centered on the front door closures, we would expect this task to be receiving a large amount of deliverables from other tasks. A similar explanation explains the high ranking of the Bodyside.

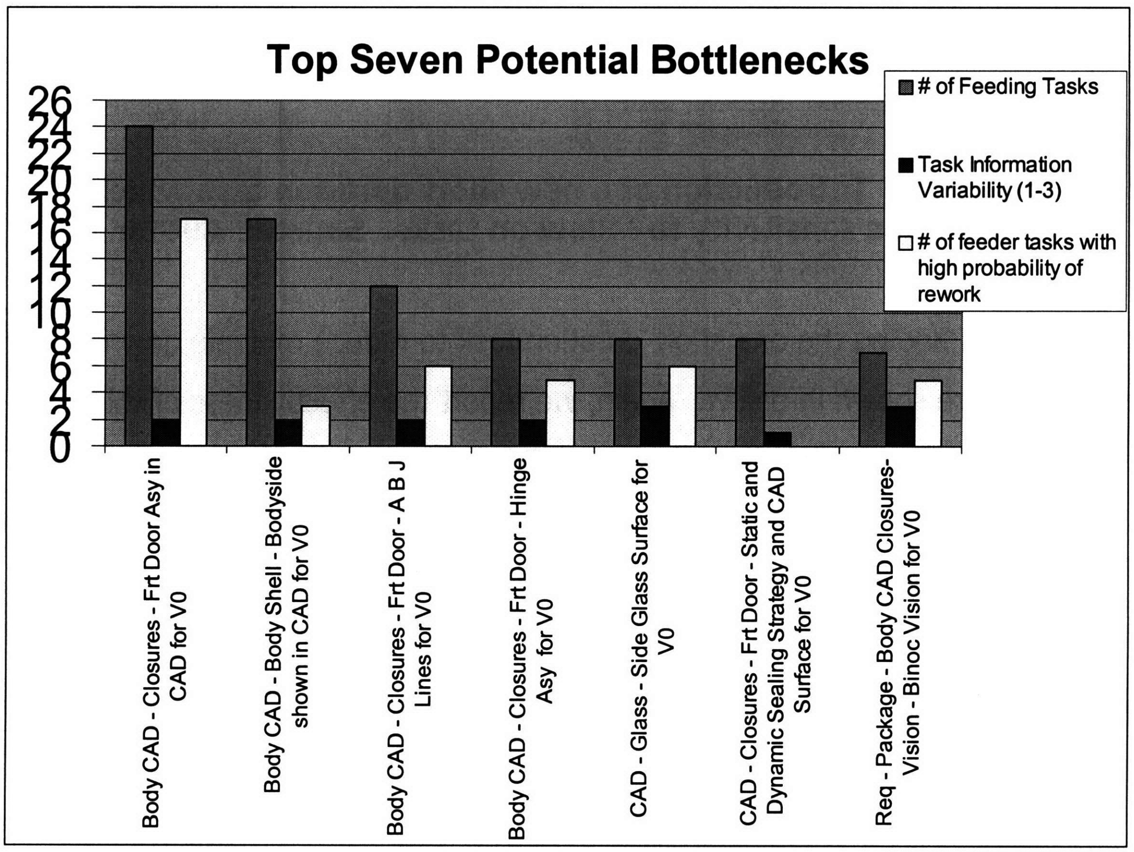


Figure 25: Top seven potential bottlenecks. Source: Author

When we consider the above two charts in unison, we can better develop our understanding tasks that are significant to the process. Tasks that appear on both charts above are those that have the greatest potential to delay the overall process, for they not only provide their deliverable to multiple tasks, but they also have the potential to wait for a significant time. Such tasks require effort to ensure that they are delivered on time and are as stable as possible. Cross-referencing both charts, we report the following tasks which appear in both Figures 22 and 25:

- Front Door Assembly
- Bodyside
- AB J Surface
- Binocular Vision Requirement

Using the Integrated Requirements Closures DSM to Understand C^2 Complexity

The above sections explaining how to identify Critical Tasks and Bottlenecks can also be applied to Completion Criteria alone. Because some Completion Criteria will be inherently harder to satisfy than others, it can be helpful to try and quantify each Completion Criterion's complexity. Doing so not only helps us better understand Completion Criteria, but can also assist in prioritizing efforts in process restructuring. One approach to do this would be to look at the number of inputs required for each C^2 and make the assumption that the greater the number of Inputs required then the greater the complexity. Doing so helps in our understanding of a given C^2 and can help in determining prioritization of C^2. Figure 26 shows the result of determining the total number of Inputs feeding different C^2 during the D0 design phases. For the D0 design phase, the Completion Criteria with the greatest number of inputs are (in order from greatest to least):

1. Package Requirement - Binocular Vision
2. System Specification - Door Swing Clearance

Note: The following requirements all had the same number of inputs feeding them

3. System Specification – Hinge Inclination

4. System Specification – Side Door Margins
5. Assembly – Instrument Panel Loading

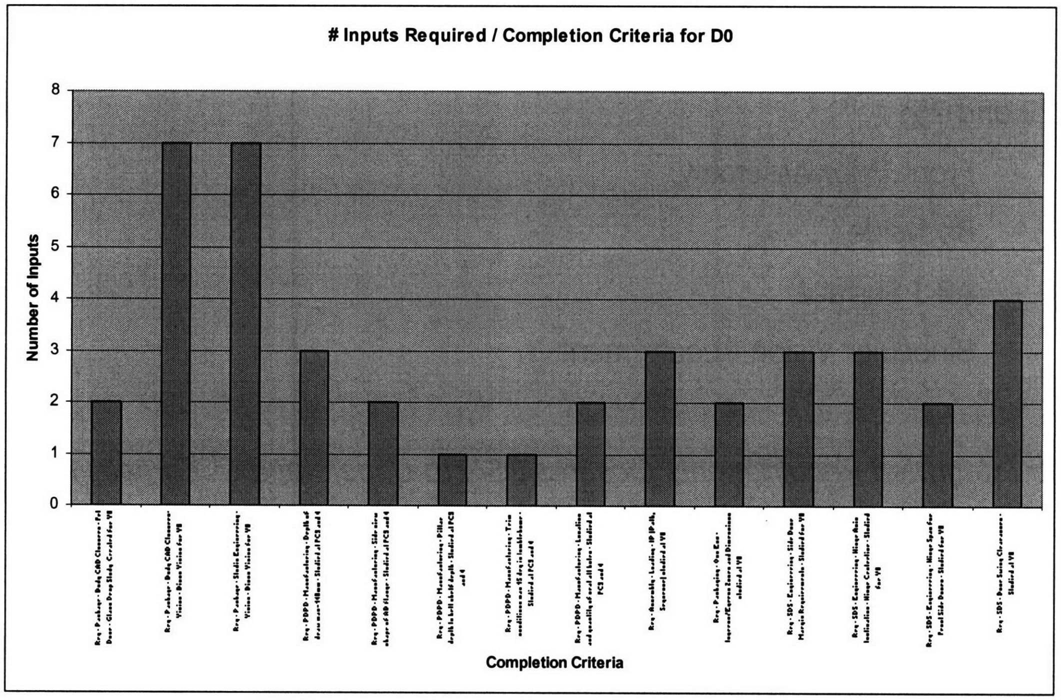


Figure 26: Number of Inputs per Requirement for D0. Source: Author

Some of these Completion Criteria appear as we might expect, for it is well known that Binocular Vision and Swing Clearance are both a function of many inputs. However, one unexpected ordering is the increase in inputs for the Side Door Margin requirement in D1 compared to D0. This is a result of the exterior ornamentation Class 1 surface being released after the sheet metal Class 1. Exterior ornamentation surfaces are not available in D0, becoming available in late D0 or D1. After speaking to Closures System Integrators, it was discovered that this is indeed a problem. Because most traditional door architectures are primarily sheet metal, this has traditionally not been an issue. However, with Sash Door architecture, Class 1 components above the shelf are non-sheet metal – they are exterior ornamentation components. The requirement for Side Door Margins, although studied in D0, is incomplete in its assessment, as the exterior ornamentation surfaces have not been released yet. With this understanding, we can now begin to asses the validity of checking Side Door Margin Requirements in the D0 design phase, knowing that the results are incomplete. In other words, we can make a conscious decision to not assess this requirement in D0; likewise, we could make the decision to pull the exterior ornamentation Class 1 release forward, if possible.

To assist in which decision to make, delay the assessment or pull the Class 1 release forward, we look at the potential impact. In this case, we observe that the Front Door Assembly has the single greatest amount of dependencies (22), therefore, long reaching consequences in its effect on development. These dependencies include both Inputs (information fed from other assemblies such as the Hinge Assembly) and Completion Criteria (information fed from assessments such as the maximum allowable depth of draw requirement). Because of its long reaching ability, it is desirable to have as much feasibility built into the design as possible. It is recommended that the Exterior Ornamentation surface release be adjusted from its current schedule and moved to be in synch with the sheet metal release.

In summary, the number of inputs feeding completion criteria may be used as a simple gage of a Completion Criterion's complexity. Process planners will find this useful because these complex tasks may require additional task duration or resources to ensure they are executed properly.

Process Restructuring Strategy

After the DSM was created for the D0 and D1 design phases, the model content and interactions were reviewed with Subject Matter Experts and an effort was made to improve the process and gage the effectiveness of proposed improvements through DSM simulation. The properties that are included in the

DSM were also reviewed with SMEs so they could understand exactly what we could attempt to effect. These properties are listed below:

- Task mean duration, best-case, and worse-case: This property represents the typical time, in days, a task takes to complete. A best and worst case value is also recorded.
- Information Variability (IV): This property represents the stability of a tasks deliverable, or the likelihood the task's deliverable will change after being initially released. It is rated on a scale of 1, 2, or 3. The definitions of what classifies tasks as 1,2, or 3 are consistent with those used by Zambito:
  - IV = 1: Deliverable is stable with a 25% or less chance of changing
  - IV = 2: Deliverable is not known to be stable or unstable and has a 25% - 75% chance of changing
  - IV = 3: Deliverable is not known to be stable or unstable and has a greater then 75% chance of changing
- Task Sensitivity (TS): Task sensitivity represents the sensitivity of a task to changes in that task's inputs. It is rated on a scale of 1, 2, or 3, with the following definitions for each:
  - TS = 1: Task is insensitive to most information changes
  - TS = 2: Task is sensitive to major information changes
  - TS = 3: Task is sensitive to most information changes
- Task Volatility (TV): This property represents the volatility of a dependent task relative to changes in information from input tasks. It is a two dimensional property resulting from the multiplication of IV and TS.
- Learning Curve: This property represents the amount of time required, as a percentage of the initial task duration, to repeat the task. Thus, a LC=0.60 requires 60% of the initial task duration when done subsequent times.

- Rework Impact: This property is reported as the percentage of a task that must be redone if the task's inputs change.

With these properties in mind, SMEs suggested strategies to improve the system with regards to one or multiple properties listed above. Table 4 depicts a summary of these discussions. Some of these strategies have already been discussed (Proposals #5 and #7), below we discuss additional proposals.

Proposal #	Effect to Task	Description of Proposal	Intended Task/System Effect
1	Modify Task	Standardize seal profiles	Decrese IV
			Decrease Prob RW
			Decrese TD
2	Modify Task	Standardize hinge assemblies and door hanging strategies	Decrese IV
			Decrease Prob RW
			Decrese TD
3	Modify Task	Incorporate early formability FEA and formability CAD parameter	Decrese IV
			Decrease Prob RW
4	Modify Task	Incorporate seal attatch point distance into door design guidelines	Decrese I.V.
			Decrease Prob RW
5	Eliminate Task	Eliminate redundant studies performed by multiple organizations	Decrease TD
6	Modify Task	Employ new technology for Binoc	Decrease IV
			Decrease TD
7	New Task	Employ Intermediate Class 1 Surface File	Decrease TS

Table 4: Process Improvement Proposals Summarized. Note: IV=Information Variability; RW=Rework; TD=Task Duration; TS=Task Sensitivity. Source: Author

Proposal #1 Discussion

This proposal came from one of the Closures System Integrators. Traditionally, each new program re-invents the seal profile they will use. This results in the reengineering associated with all aspects of the seal profile. Auto OEM is benefiting very little from past experience as they are essentially reinventing the wheel with each new program. Despite this large re-engineering effort, most seal profiles end up being very similar. With this in mind, it is possible to standardize most dimensions in the seal profile, leaving only minor tweaks available to "customize" the profile for a given design. This proposal does not require the addition of a task to the PD process. Instead it requires changing the inputs to a task and how the task is executed. The additional input to this task is the standardized profile and the different execution is to follow the standard profile. This proposal will lower the task's Information Variability, Probability of Re-work, and Task Duration. This proposal also helps to eliminate this task as a potential bottleneck (as pointed out in Figure 25).

Proposal #4 Discussion

One of the manufacturing requirements addresses the closeness of the seal attach points onto the Door Inner panel. If the seal must be attached to an inward facing radius on the Door Inner Panel, there must be a minimum distance between attach points to ensure the sheet metal hole punches do not intrude on each other (reference Figure 28). The proposal is to include the minimum required distance as a design guideline. Furthermore, this proposal requires incorporating the guideline into the component CAD file as a parameter. Incorporating the parameter helps to enforce the requirement and to flag violations – this essentially builds the manufacturing requirement right into the design. This proposal decreases task Information Variability and Probability or Re-work. It is worth noting that we do not look to remove this Completion Criterion from the D0 or D1 design phases. Ignoring this can have grave consequences, as this requirement acts as one factor in driving the shape of the Door Inner sheet metal and therefore the packaging of hardware inside the door.

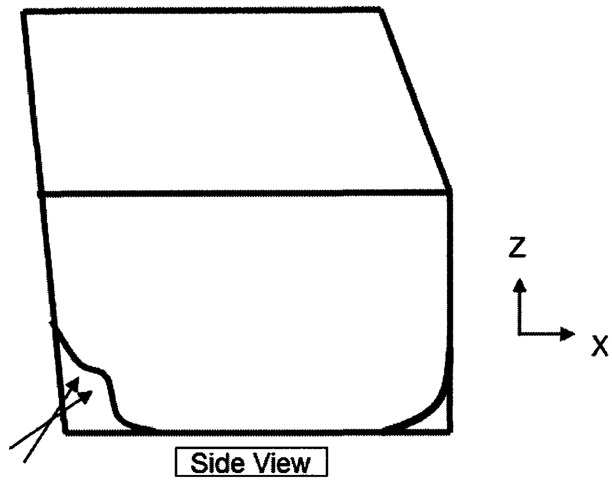


Figure 28: Side view of door inner with seal attach points (shown with arrows) on inward facing curved surface must have enough distance between them to facilitate the hole punches in the manufacturing dies. Source: Author

Proposal #5

SMEs and Managers all cited numerous identical requirements being assessed by different organizations. When this happens, resources are wasted performing duplicate tasks which are not accounted for in the process. Furthermore, engineering staff find themselves responding to different people and organizations for the same requirement. Our DSM captures one duplicate requirement; this proposal eliminates the duplicate requirement.

Proposal #6

This proposal entails changing the development and verification method for the task Assess Binocular Vision. As highlighted in Figures 22 and 25, assessing the Binocular Vision requirement requires a high number of inputs, and because of coupling with component development, can have a large effect on other tasks. This coupling is considered "intended" feedback and is inherent to the iterative nature of developing the vehicle in the A-pillar area. Thus, we do not look to eliminate the task. Because this task requires so many geometric inputs, it is time consuming and prone to human error in first time set-up. New technology development can be employed to decrease the task duration required for assessing this requirement and improve first time through assessments. The current task requires five days (nominal). Using KBE based tools, numerous assessments can be made in one day. The KBE based tool not only assesses the package of the geometry, but it also assesses the static stiffness properties of the resulting section, which is often a body structures target. This tool should be used by all programs as the verification tool used to assess the Binocular Vision Completion Criteria requirement.

Results of Proposals

The above strategies were incorporated into the As-Is-P DSM to create the Future Process DSM. In cases where tasks' properties (such as Information Variability or Task Duration) were reduced, a judgment based on experience was used to determine by how much. For some task durations, technology experts were consulted to determine effects of technology deployment. The collective effect on the system resulted in the following histogram (Figure 29):

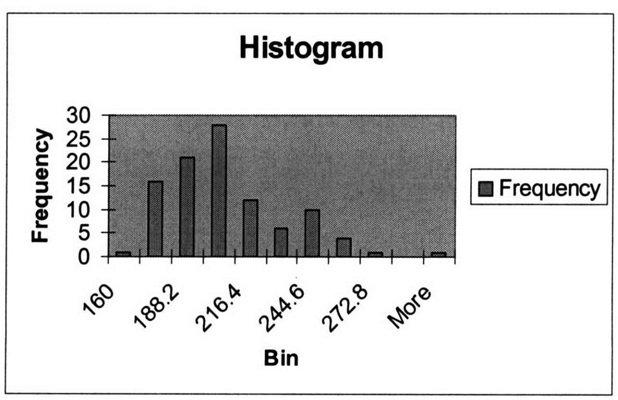


Figure 29: Histogram of Process Completion time considering all improvement proposals. Source: Author

- Net change in average process time: 29% Average Process Time Reduction

- Net change in process standard deviation: 60% Standard Deviation Reduction

Other Restructuring Techniques

The DSM offers itself as an effective analytical tool as well as a visual tool. As described in Chapter 4, marks that are made above the center diagonal of the matrix represent potential rework. These relationships can be planned or unplanned (de Weck, 2006). Generally, planned iteration is acceptable and unplanned iteration causes costly and unexpected rework. However, planned iteration could be considered planned waste. In this vein, any iteration should be evaluated to determine if it is truly needed. Also, it is possible to reduce or eliminate iteration by changing the process. In our assessment of the As-Is-P DSM, we identified a number of coupled tasks which represent planned iteration. These iterative marks generally took the shape of component to component relationships or Completion Criteria to component relationships. Examples include the coupled interaction of Hinge Assembly to Door Assembly and the Hinge Assembly to the Striker and Latch Assemblies. The development of these assemblies is inherently iterative and the current process intentionally plans for this iteration. Except for the already mentioned duplicate tasks, no unplanned marks were recognized. In an effort to reduce the effects of rework, planned or unplanned, DSM practitioners will often attack the outliers in a coupled process. In Figure 30 we see marks in the far upper right corner of the heavily coupled large block. These marks represent planned iteration in the form of Completion Criteria evaluating Inputs. One example is the Hinge Span C^2 requirement that specifies a minimum distance between hinges based on the size of the door they are attached to. The Inputs Hinge Assembly and Door Assembly are used to assess the requirement. The requirement is assessing the location of the Hinge Assemblies; therefore, it is one factor in the packaging of the Hinge Assemblies.

The Completion Criteria in the upper right corner were reviewed and determined to be appropriate. Although no Completion Criteria were evaluated to be removed, we still point out the value of having the matrix constructed and easily being able to visually identify planned and unplanned iteration.

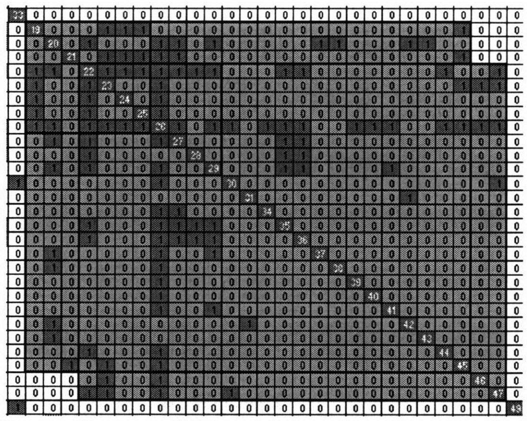


Figure 30: For Do, planned iteration is shown in the upper right corner of the matrix in the form of Inputs being evaluated by completion criteria. Source: Author

Unintended Rework

Although we have described the As-Is and As-Is-P DSMs to have no unintended rework, during the review of the final DSM instances were reported that did cause unintended rework.

Observing Figure 16 (the partitioned matrix) reveals two large macro processes representing D0 and D1. Each of these blocks is highly coupled within themselves; however, the two are essentially decoupled from each other. Information is fed forward from D0 but there are no marks representing tasks in D1 having any influence on tasks in D0. The de-coupling of the two major design phases is by design, intended to contain iteration within design phases. However, occasionally decisions made in subsequent design phases force development rearward to a prior design state of completion. In such cases, decisions made have such a large impact that the changes associated with them cannot be contained within the design phase where they are made.

SMEs at Auto OEM used a specific case to illustrate this. This instance involves the development of the AB flange. The AB flange is a key feature in the front door development; it affects a large number of components and interfaces (recall our analysis in the prior section identifying the AB J Surface task as both a Critical Input and a potential Bottleneck). This surface and resulting flange require a lot of development and drive the development of many other components. Additionally, the AB flange is evaluated to a number of Completion Criteria. Normally, the AB flange is matured to a point of minimal, if any at all, movement by the beginning of D1. The plane for the AB flange surface is pretty well frozen, but small movements along that plane may occur. These small movements are considered part of normal development and handled within D1. However, substantial movement of the AB profile or surface is difficult to contain in D1. Such large changes negate prior Completion Criteria assessments and can take the maturity of the design back to a D0-like maturity. The SME reported that one vehicle program made the decision to move the AB flange more than one inch, making the decision after the D1 design phase. This SME asked how we could model such an instance using the DSM so he could show the potential rework associated with making such a drastic change so late in development.

To replicate this in our DSM process model we add a task to the D1 design phase. At first response, one thinks to add a mark above the center diagonal from the AB Flange in D1 back to the AB Flange in D0. However, although the information in the D1 AB Flange did change, it was the result of a *decision* to change that Input. Therefore, we must add an additional task to capture this decision. The task is considered a decision, which is an Input, and is scripted as follows:

➢ Engineering – Closures – AB Flange – Consistent w/D0 Program Assumptions

The deliverable for this task would be worded as follows:

➢ D1 AB flange surface generated with consistent assumptions from D0 design phase.

Further detail could be added to quantify what "consistent" means. For example, consistent could mean that the surface is not moved from the D0 position greater than some defined distance. The point to make here is that this *Decision* is what will cause the design to be pushed back to D0 or stay in D1. If, say, the Decision is to move the AB outside of the specified tolerance, we can quickly assess the impact by making a mark from the Decision node feeding back to the D0 AB Flange. This, in turn, triggers not only the re-generation of the AB flange task in D0, but potentially anything else fed by the D0 AB flange will. The additional task representing the decision to change the AB flange significantly was added to the As-Is-P DSM and then re-partitioned to determine observe the effect. Figure 31 shows the result.

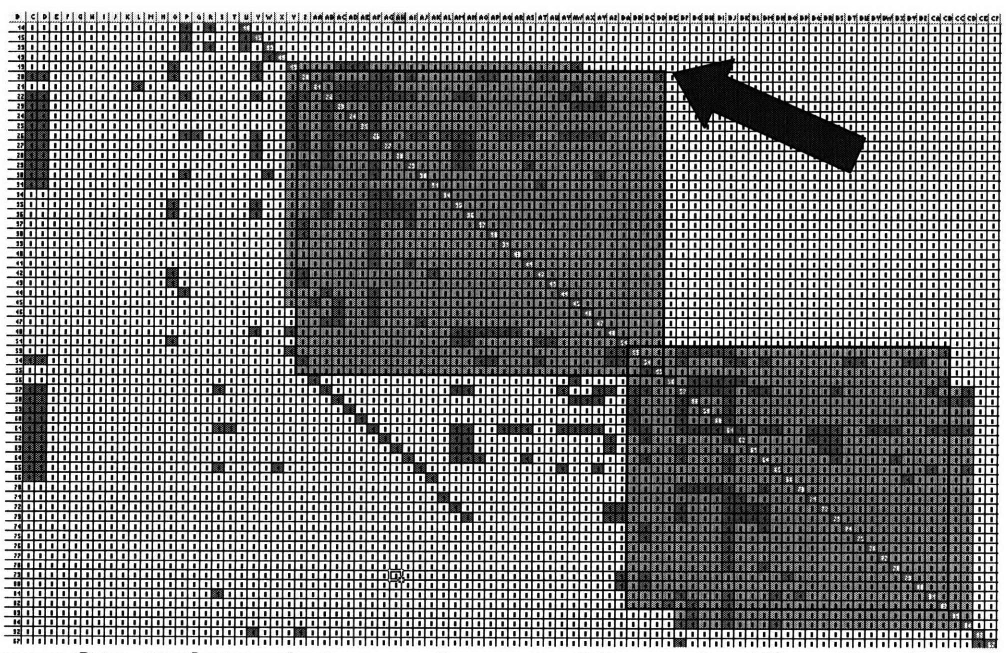


Figure 31: Unintended rework created from significantly changing the D1 AB flange. Source: Author

As we expect, there is now a feed back loop (highlighted with the red arrow in Figure 31) to the D0 design phase. Although this is readily witnessed in Figure 31, in practice decision makers do not have such clear illustrations or models to show the potential impact of their decisions. In this case, the DSM clearly shows the feed back – the next step in further understanding the impact of deciding to move the AB flange would be to identify each Completion Criteria and Input affected by the D0 AB Flange. Additional, quantitative effects on mean process time and standard deviation can be evaluated through simulation.

The result of having a model capable of capturing potential rework is better decision making. When PD teams are faced with a tough decision, they can use the DSM to easily capture the potential negative and positive effects to the system.

Conclusions

Recall our stated purpose explained in the beginning of the prior chapter, "... to demonstrate how to develop a process that delivers product definition (Inputs) to satisfaction (Completion Criteria)." We have shown how to architect the tasks of a DSM in the format of Inputs and Completion Criteria. This unique binning strategy enables us to identify the information created within product development as well as assess this information to requirements. We have also presented a number of ways to better understand a process by identifying Critical Inputs and Bottlenecks while considering Information Variability and Rework Probabilities. In concert, these techniques allow us to develop a process that evaluates Inputs at the appropriate time in the development cycle. Furthermore, we showed how to create a DSM spreadsheet that is handy to management and to those not intimately familiar with the creation of a DSM.

The techniques presented in this work were demonstrated on the closures system through the D0 and D1 design phases. Although the closures system was chosen as the product to model, the techniques in architecting a DSM and analyzing it can be applied to nearly any product development process. Techniques were applied to understand and prioritize our efforts. It was pointed out that process improvement can be made by attacking a number of different task attributes or by creating entirely new tasks.

Strategies were developed with closures SMEs and these ideas were measured using simulation and referencing the base line DSM. The process proposals suggested in this paper suggest a potential of nearly 30% process reduction time, given the benchmark. Recognizing that the DSM created for this work is not complete in that it does not contain all completion criteria, we can easily say that actual process improvements would be less then 30%. None the less, this paper has demonstrated how to apply techniques to process structuring to improve the overall process.

Given that this work represents a proof-of-concept model, being evaluated for further application, we conclude that the process modeling techniques presented herein are effective and the model should be further developed. The DSM model used for this work can easily be expanded to include as many Inputs and Completion Criteria as necessary.

Auto OEM does not currently use process models that account for the iterative nature inherent to complex system development. The DSM represents a model that could be employed to accurately capture the product development process. Because of Auto OEM's limited experience with such models, a training regimen is suggested for those involved in DSM projects. Recent work in Auto OEM's PD group includes the use of Value Stream Mapping. Each project begins with a slide show introduction of basic VSM concepts to educate participants. This same educational technique could be applied to DSM projects at Auto OEM.

Chapter Summary

Chapter 12 represented the more technical side of this paper. At the beginning of the chapter we laid out our analysis approach. This was followed by a detailed discussion on the nature of interactions in the As-Is and As-Is-P model. As we expected, the mature process of Auto OEM yielded little benefit from partitioning. Simulation of the As-Is-P DSM gave us a benchmark to measure the effectiveness of our proposals.

A number of ways to assess tasks were presented in this chapter. We highlighted the discovery of Critical Tasks and potential Bottlenecks, for example. Sharing the model with SMEs from Auto OEM allowed the generation of a number of process proposals. These proposals were incorporated into the DSM and simulation was used to determine a 30% decrease in average process time and an improvement in standard deviation. This large improvement is admittedly very aggressive, as our DSM does not have all Completion Criteria accounted for.

SMEs also explained how major rework occurs. We incorporate their experiences in the DSM and partitioned the DSM to reveal the rework they reported. This showed how the DSM can be used to educate decision makers on the potential impact of their choices.

Chapter 7: Reflection and Future Work

Chapter Summary

In this final chapter we reflect upon a few practical lessons learned from this project. These lessons are highlighted for those planning a DSM oriented project. Finally, we suggest related future areas of work.

DSM Construction Process Reflection

Constructing a rigorous process model at a large company is no easy task. Engineers and managers are busy folks. They are constantly on the run and it would seem that many are more controlled *by* their schedule rather than in control of their schedule. That said, the engineering staff that helped to support this effort certainly participated and contributed their expertise and time. There is a certain bit of explaining one must do in order to gain full support of staff members. Staff are often caught off guard when asked specific questions regarding exactly how long a task takes. Similarly, because they are not familiar with the DSM, a fair amount of explaining was required. To help educate PD members on DSM practices, an internal company website could be developed. This site would have basic instructional material and summaries of prior company DSM projects. Also, company process engineers were not familiar with the DSM. They are familiar and comfortable with Gantt Charts and Work Plans – the DSM looks foreign to them. We would like to think that this project, running on the heels of a recently finished DSM project, is helping to get the word out.

Information gathering is tedious and not fun. Future DSM practitioners should be warned, matrices of significant size will test your patience and eye sight. Keeping track of information in the DSM can initially be extremely difficult - as tasks are gathered and de-conflicted, task identities will change making for a lot of update work. This author suggests recording task dependencies and later populating the

DSM with marks, after the greater portion of de-conflicting has taken place. Also, simple cartoons or illustrations (such as those shown in Figure 18) can help in keeping track of relationships.

Task durations can be difficult to peg down. Most engineers working at Auto OEM divide their time among numerous vehicle programs. Thus, they are rarely dedicated to one task and the time required to complete that task. Additionally, caution should be expressed when gathering task duration from the person who is performing that task. Few people want to have additional time removed from schedules and they may pad their task duration. It is best to try and sample a few sources to arrive at an average task completion time.

Future Work

The next logical step for future work is to further develop this model's content. More technical work that can be done is the incorporation of resources into the DSM. Being able to add and subtract personnel to a task, and understand the impact to the system would be helpful. Resource optimization in the DSM could be very helpful to planners.

One of the most important observations we can make is that the methodology applied this paper can be applied to any number of product development projects. In this work, we have centered on automotive closures; however, nearly any product development process may benefit from the approaches presented.

Chapter Summary

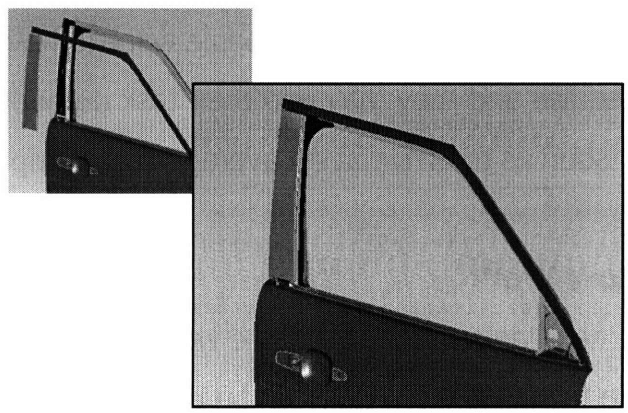
This chapter captured helpful practical advice that was gained through experience developing this paper. DSM users may want to consider this advice while undertaking their own DSM projects. Finally, we suggested areas that might be further developed in the future.

Appendix A: Closures Illustrations and BOM

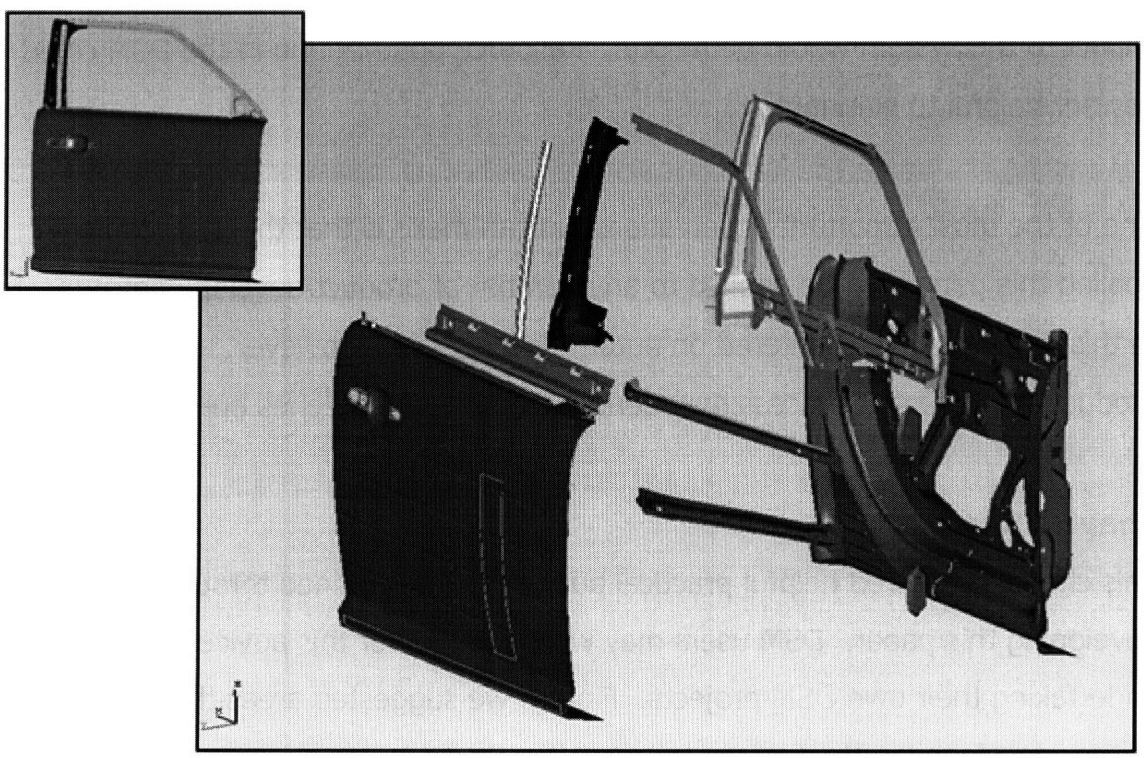
Sash Door Architecture

Sash Door Architecture Front Door Assembly typically consists of the following sheet metal components:

- Door Inner
- Door Outer
- Above the Belt Halo Assembly
- Belt Reinforcement Outer
- Hinge Reinforcement (2X)
- Latch Reinforcement
- Side Impact Beam
- Appilque at B-Pillar
- Appilque at A-Pillar
- Sail Reinforcement



Appilques at A and B-Pillars

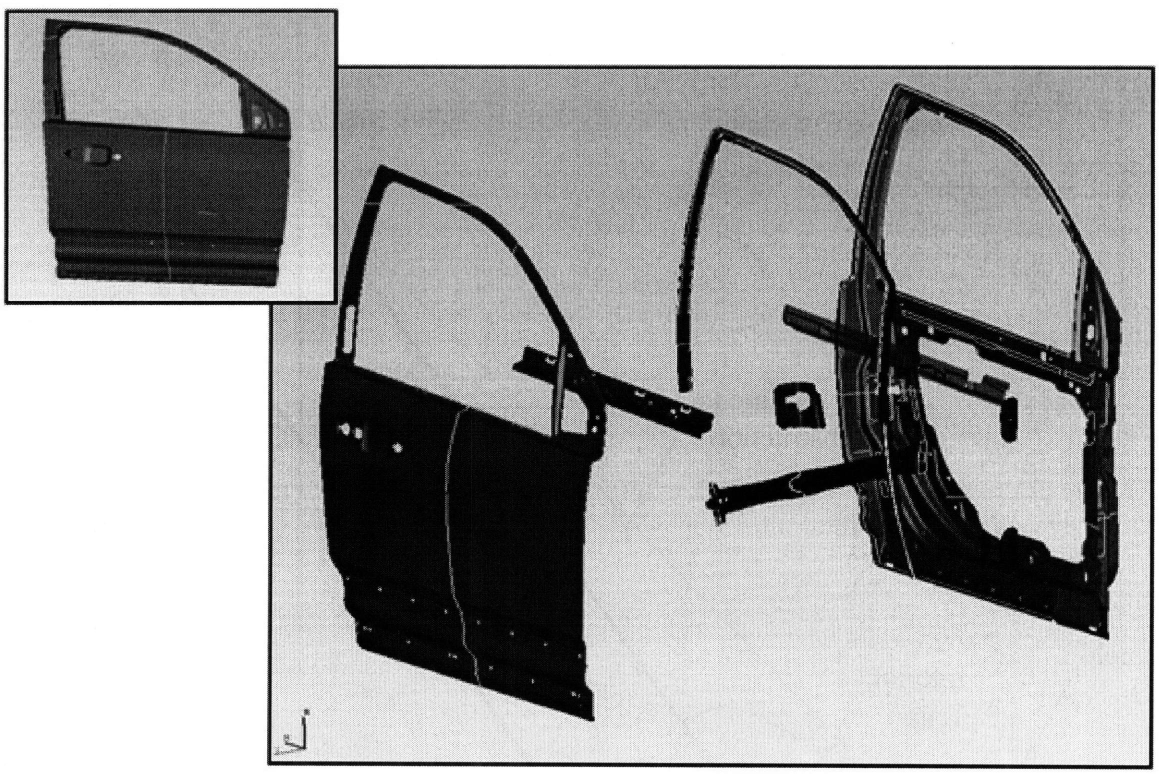


Typical Sash Door Assembly Components

Stamped Frame Architecture

Stamped Frame Architecture Front Door Assembly typically consists of the following sheet metal components:

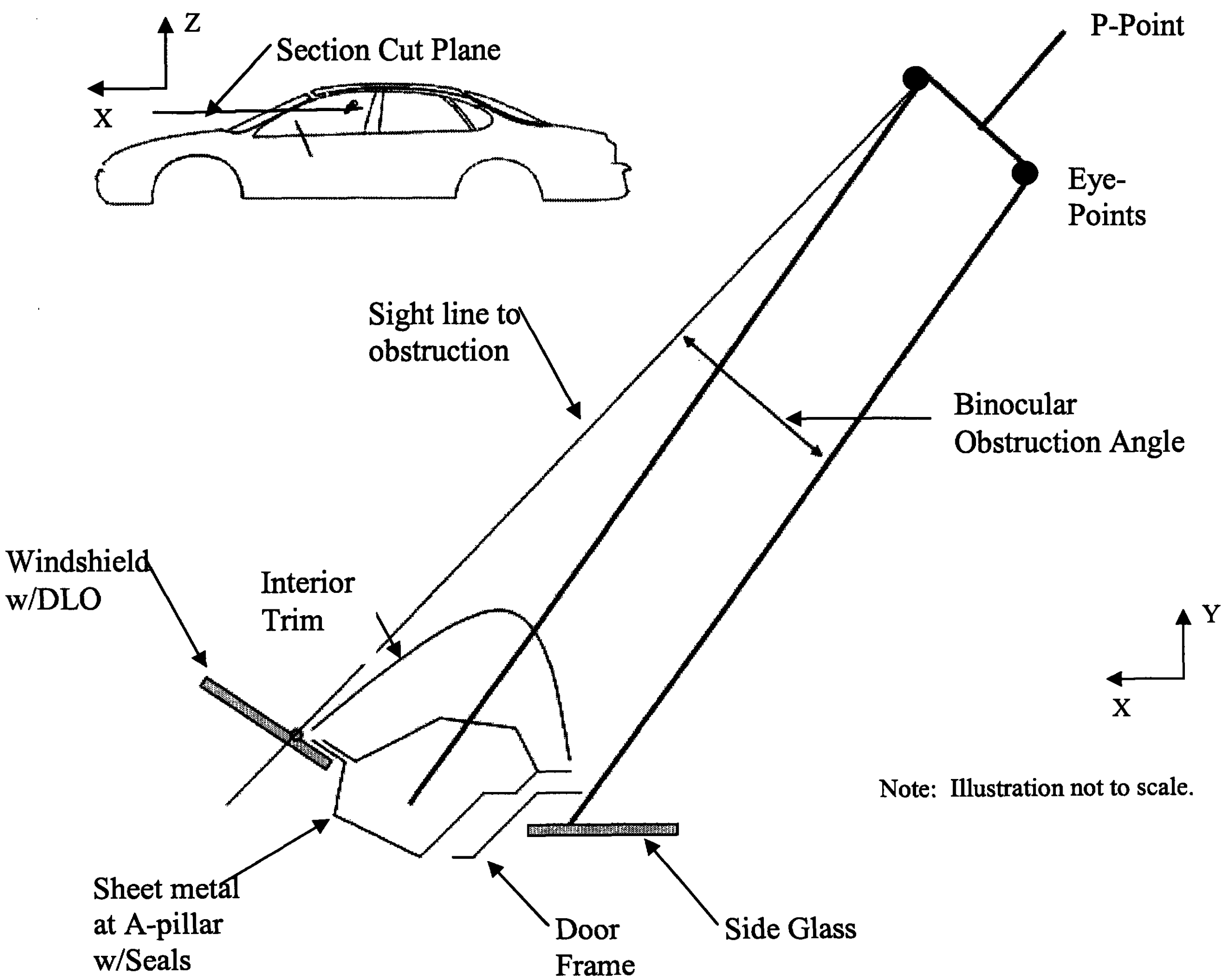
- Door Inner
- Door Outer
- Belt Reinforcement Inner
- Belt Reinforcement Outer
- Hinge Reinforcement (2X)
- Latch Reinforcement
- Side Impact Beam



Typical Stamped Door Architecture Assembly

Appendix B: Binocular Obstruction Study Requirement

A-pillar binocular obstruction requirements represent the vision space needed in the area of the A-pillar. The Society of Automotive Engineers specifies measuring to this requirement (SAE J1050) by developing a section, cut parallel to the Two Up Vehicle Attitude (position of the vehicle body w/respect to the ground at standard loaded condition) from the driver's right and left eye points. The SAE requirement indicates the maximum allowable A-pillar obstruction to be be less than 7.50 degrees on the driver's side and less than 6.00 on the passenger side. A typical study requires the gathering of program data in the area of the A-pillar, cutting the section, and comparing the resulting angle to the requirement. The illustration below depicts the approximate cut plane and a typical section cut to measure A-pillar Binocular Obstruction on driver's side.



Appendix C: References

Zambito, Antonino P., "Using the Design Structure Matrix to Streamline Automotive Hood System Development," SM Thesis. MIT Libraries, Cambridge, MA: 2000.

Noor, Jehanzeb G., "A Comprehensive Approach to Complex System Product Development: Operations Management Tools applied to Automotive Design," SM Thesis. MIT Libraries, Cambridge, MA: 2007

McGill, Eric A., "Optimizing the Closures Development Process Using the Design Structure Matrix," SM Thesis, MIT Libraries, Cambridge, MA: 2005

"Describing and Measuring the Driver's Field of View," SAE J1050, January 2003

Eppinger, Steven et al. "A Model-Based Method for Organizing Tasks in Product Development." Research in Engineering Design, 6.1 (1994) 2-8.

Eppinger, S. D., Uhlrich, K. T., Product Design and Development, 2^{nd} ed., the McGraw-HCompanies, New York, NY: 2000

Browning, Tyson R. "Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions." IEEE Transactions on Engineering Management 48.3 (2001) 292-307.

McManus, Hugh., "Lean Engineering: Doing the Right Thing Right." Queen's University Belfast, Northern Ireland, UK: 2005

"Task Based DSMS" ESD.763 – *System Project Management* Lecture 3, September 19th, 2006.