Representing Information Flow and Knowledge Management in Product Design Using the Design Structure Matrix

by

Qi Dong

B. S. Mechanical Engineering

University of Kentucky, 1997

SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

AT THE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

FEBRUARY 1999

Signature of Author: ______________________________
Department of Mechanical Engineering
January 8, 1999

Certified by: ______________________________
Daniel E. Whitney
Senior Research Scientist
Center for Technology, Policy, and Industrial Development
Thesis Advisor

Accepted by: ______________________________
Ain. A. Sonin
Professor of Mechanical Engineering
Chairman, Committee for Graduate Students

Representing Information Flow and Knowledge Management in Product Design Using the Design Structure Matrix

By

Qi Dong

Submitted to the Department of Mechanical Engineering
On December 23, 1998 in Partial Fulfillment of the
Requirements for the Degree of Master of Science in
Mechanical Engineering

ABSTRACT

The design of complicated products such as automobiles requires timely coordination of many departments in an organization in order to win the time-to-market competition globally. Ford approaches this challenge with the Ford Product Development System (FPDS) based on systems engineering philosophy. Reusing the past engineering design knowledge helps to speed up the product development process. The Ford Direct Engineering (DE) program semi-automates the product design process and increases the knowledge re-use capability of the organization.

This thesis uses the Design Structure Matrix (DSM) method to study the design processes of three cases at Ford: the vehicle door internal subsystems, the throttle body system interface, and the throttle body assembly. The results of studies recommend the sequence of the design tasks for the DE programs. The existence of design iterations shows the current DE computer programs cannot automate the design process without involving human interactions. DSMs provide means to integrate individual DE programs, and suggest how to form teams based on the design issues rather than the traditional organization divisions. DSMs and their companion information databases can serve as browsers for design knowledge reuse, and define knowledge ownership. DSM can also quantitatively compare the complexity of two design alternatives. The three cases reveal that most knowledge captured is at lower level of the decomposition such as parts. System level knowledge is less understood and documented. This thesis also compares the DSM method to the Associativity Map method and the Datum Flow Chain method.

The accuracy and the effectiveness of the DSM method depend on the knowledge acquisition process and the data representation format. Incorporating timing with DSM improves the scheduling of tasks. This thesis discusses what knowledge DE or general Knowledge Based Engineering (KBE) should possess as well as the impact of the DE and KBE applications on the organization and the challenges they face.

Thesis Advisor: Daniel E. Whitney, Senior Research Scientist, Department of Mechanical Engineering

ACKNOWLEDGMENTS

This thesis work is made possible by the NSF cooperative agreement No. EEC-9529140. Massachusetts Institute of Technology Center for Innovation in Product Development and my research advisor Dr. D. E. Whitney provided me the opportunity to conduct the research with industries, specifically the Ford Motor Company.

For the completion of this thesis work, I owe great debts to my research advisor Dr. D. E. Whitney. Dr. Whitney not only offered me this research opportunity, but also gave me much needed advice and encouragement for the formation of the ideas in this thesis. In the process of this thesis work, I learned from him how to think with an open mind from all possible angles and dare to question everything. I will benefit from these lessons all my life.

I also would like to thank many managers and engineers at Ford Motor who provided me great assistance throughout my thesis research. The Direct Engineering group led by DE managers Ken Kuna and Pete Sferro at Ford provided the ground for this research work. For the completion of the case study on door design, Advanced Vehicle Technology engineer Darrell Kleinke played a central role. For the case study on throttle body, Kevin Deacon and Dave Wise provided me great help in learning and understanding the problems. They were also good audience for my ideas and discussions. Dave, I shall be able to spend more time practicing guitar now since the thesis writing is almost finished. I would also like to thank throttle body design engineers Denis Madson and Jim Rauch, accelerator control AVT engineer Jim Conrad, and air intake system engineer Jhun Lin for spending so much of their valuable time and helping me understand the details of the system level interaction. I would like to thank Bob Matulka and Dave Roggenkamp for helping me better understanding FPDS in the process of this thesis writing. I want to also extend my thanks to Visteon KBE manager Shuh-yuan Liou and his group for their help on understanding different approaches to the Knowledge Based Engineering. In addition, I want to thank Kevin Deacon, Pete Sferro, Darrell Kleinke, Bob Matulka, and David Roggenkamp for reviewing this thesis and providing me very valuable comments.

I also owe great debts to my father Shaojing, my mother Dalin, and my brother Huayi. Their love and understanding provided the emotional support much needed during the thesis work. I also must thank all my friends who gave me much encouragement and advice during my work. Thank you all.

TABLE OF CONTENTS

Part I Introduction and Background

Part II Thesis Research Methodology and Results

Part III Conclusions and Future Research

LIST OF FIGURES

LIST OF TABLES

Part I
Introduction and Background

CHAPTER 1 INTRODUCTION

When I was an engineering student co-op at General Electric Appliance division three years ago, I worked as a product design engineer in the refrigerator high-pressure-side system group for three months. One of my projects was to redesign the refrigerator condenser and implement the design changes. Having taken courses in Heat Transfer and Engineering Thermodynamics, I worked out the new condenser configuration in two days and turned it over to the CAD group. At that time, I thought to myself I could probably get this project done in a week. My manager had better find something else for me soon to keep me busy.

However, the implementation of the new design change turned out to be much more complicated than what I thought. The manufacturing engineers came to tell me that the bend radius I chose could not be made by the GE bending machines at Louisville, Kentucky. Although the vendor in Mexico could make the bends, and it was cheaper to outsource the condensers anyway, the unionized factory workers would go on strike if GE did so. The purchasing engineers came to tell me that if I could change the pipe diameter to another size, they might be able to get a lot of savings. After a long delay in the CAD group to create the drawings, the prototype of the new design was made. However, when I tried to put the new condenser onto a test refrigerator, I found that I forgot to consider the space for the compressor, and so the compressor was in the way of the condenser. The lab technician suggested folding the condenser into two-thirds of its length, but I had to do more calculations to make sure the heat transfer was okay. Besides, the additional height might interfere with other components in the assembly. I also had to talk to the manufacturer of the prototype to make sure they can make the bend. Also, what would an additional bend mean to the total cost of the product?

It turned out that this project together with several others not only kept me busy throughout my co-op assignment, but also were not completely implemented by the time I went back to school. I was a little frustrated but also became deeply interested in understanding and improve the design process of complex products. After I entered graduate school at MIT, my research advisor Dr. D. E. Whitney and the MIT Center for Innovation in Product Development introduced me to the product development process in the US automotive industry, specifically the Ford Motor Company. The issues associated with understanding the existing design processes and capturing and reusing the design knowledge provided me challenging research opportunities and insights to the design of a complex product.

1.1 Problem Statement

A complex product such as a vehicle usually consists of thousands of components. Product design in such a scale requires the collaboration of many people. In order to accomplish the product design in timely fashion, the product is usually decomposed into systems, subsystems, and parts. The engineers then form teams to work on various parts of the product. In the end, different part of the product are brought together to form the final product. This decomposition and integration process is the systems engineering approach to the product development of a complex product. It allows engineers to concentrate on their area of expertise, and deliver the product faster. However, this approach also faces challenges. Just like what I experienced during my co-op assignment, the technical aspect of designing a single component is usually well understood and relatively easy to manage. Problems occur when components are integrated into a system, and have to interact with other elements in the system. Such interaction include that with other physical components in the system for the purpose of meeting packaging requirements, power requirements, etc., and that with non-physical components such as the manufacturing facilities, the product cost, etc. These interactions often raise issues unexpected when designing a component without considering the behavior of the entire system. Hence the proper decomposition and integration of the complex product is the critical to the speed and the quality of the design of a complex product.

In addition to the problems the decomposition and integration process faces, the US automotive industry is also challenged by the fierce global competition. The requirements on the quality, cost, and the time-to-market of the product are rising high. Ford responds to the global challenge with the Ford Product Development System (FPDS) based on the Systems Engineering philosophy. Ford also realized the importance of reusing past design knowledge to improve the speed and quality of current design. The Ford Direct Engineering (DE) program intends to speed up the product design processes by semi-automating the design activities and increasing the knowledge re-use capability of the organization. However, the existing DE applications are on small subsystems in the vehicle such as the throttle body system. A car consists of thousands of these subsystems. Finding a way to integrate these DE applications is critical to speed up the overall design process of the entire vehicle. However, the top-down approach that FPDS proposes seems to conflict with DE's integration approach. Chapter 2 and 4 of this thesis has more detailed discussion on this point. Chapter 2, 3, and 4 also contain more detailed discussions on the FPDS, design knowledge reuse at Ford, and DE, respectively.

The goal of this thesis research is to understand and speed up the product development process at Ford Motor Company and maintain Ford's competitive advantage globally. The results and observations in this thesis should also be general enough to be able to benefit the design process of complex products in general.

1.2 Available Methods

This thesis employed the Design Structure Matrix method (DSM) to study the Ford product development process. DSM uses precedence matrices to record the information exchange in the design process, and further suggests the optimal sequence of completing tasks based on the information flow. The suggested sequence can help Ford engineers eliminate unnecessary design iterations, and deal with inevitable design iterations. Hence the design process can be more efficient. The DE programs can also incorporate the suggested sequence. Furthermore, DSM can be used to capture, store, and represent design knowledge for reuse. More detailed discussion of DSM method is in Chapter 5 and throughout the case studies in this thesis in Chapter 6, 7, and 8.

Two other methods are studied and compared to DSM. The first one is the Datum Flow Chain (DFC) method. DFC helps to map out the assembly relations based on the Key Characteristics involved in the assembly. The second method is the Associativity Map (AM). It is a method developed by the DE engineers to understand the relationship between parts in an assembly. Chapter 9 and 10 contain detailed discussions on DFC and AM, respectively.

1.3 Thesis Objectives

The objectives of this thesis work are as follows:

- Explore different means of representing product design knowledge using:
  --Design Structure Matrix (DSM)
  --Datum Flow Chain (DFC)
  --Associativity Map (AM)
- Study what information the DE programs should include at assembly level, as well as how to understand the system level interaction so that individual system design can be integrated into the whole vehicle design and the part centered design mentioned in Chapter 2 can be avoided.
- Compare the above three methods. Find out whether there is a particular method of modeling that is the best for the Ford Direct Engineering, or if all are needed.
- Categorize knowledge and rules in a design process, and explore means to effectively capture and represent them:
  --Local and non-local knowledge. Local knowledge is the knowledge that is kept and used in one organization. Non-local knowledge is the knowledge that exists in many different

organizations, and usually does not have one single owner. The information ownership is of this thesis's interest.

--Generic knowledge and variant knowledge. Generic knowledge is the knowledge that applies to all product models, such as all throttle design configurations. Variant knowledge is the knowledge that is specific to a particular product design configuration such as the throttle bodies using cam linkages.

- Capture and Understand information flow and coupling, and their implications on computerized Knowledge Based Engineering (including Ford Direct Engineering).
- Use DSM to provide means to link the DE current application islands.
- Explore the robustness of the Direct Engineering program. Understand how flexible the DE programs are to the change of product designs.

1.4 Research Settings

This research opportunity is supported by NSF cooperative agreement No. EEC-9529140, the MIT Center for Innovation in Product Development (CIPD), and Dr. D. E. Whitney. CIPD is a National Science Foundation funded engineering research center. Its objective is to develop and deploy breakthrough product development science, processes, and tools, in order to provide the US industries a privileged competitive position in the world.

This research was carried out in Ford Motor Company at Dearborn Michigan, and Visteon--an enterprise of the Ford Motor Company, Rawsonville plant at Rawsonville, Michigan. Visteon is newly separated from Ford and contains all the Ford's components divisions. The Visteon Rawsonville plant designs and manufactures a variety of automotive components, including alternators, fuel rails, intake manifolds, and throttle bodies.

The author spent two months in summer 1997 at Ford Advanced Vehicle Technology center, Dearborn, Michigan to study the vehicle door design process. In January 1998 and summer 1998, the author studied the design process of the throttle body and the Ford's Direct Engineering effort on throttle body mainly at Visteon Rawsonville plant. Many engineers and managers from various departments in Ford and Visteon provided supports and inputs for this research, as mentioned earlier in the "Acknowledgement" section.

1.5 Preview of the Thesis

The rest of the thesis is organized as follows:

Chapter 2 discusses how Ford approaches its product development processes using system engineering concept, including its corporate level practice--Ford Product Development System (FPDS). This chapter also presents the issues and concerns related to the FPDS, and show where this research work could benefit Ford's product development process.

Chapter 3 presents the current information reuse efforts at Ford and the need for new methods such as the Ford Direct Engineering and other systems engineering tools. The causes of the need for new methods are discussed. This chapter also explains where this thesis research work can help.

Chapter 4 describes the Ford Direct Engineering concept, and its approach in the throttle body design. The achievements of direct engineering and the issues associated are discussed. This chapter also explains where the thesis work can help.

Chapter 5 explains the first methodology used in this thesis work—the Design Structure Matrix (DSM). Previous work done by other researchers using this method is presented. The general steps the author took in using this method in this thesis are also described.

Chapter 6 presents the first case study using DSM—the vehicle door. The problem associated with the door design is discussed. The steps taken for the study are described. The results and findings from this case study are also presented.

Chapter 7 presents the throttle body system interface study using DSM. The main objective of this study was to find means of linking direct engineering applications written for various systems such as throttle body system. The systems that interact with throttle body system are described. The steps taken are listed. The results and observations from this study are also presented.

Chapter 8 presents the throttle body system component interface DSM. The main objective is to understand the interaction among the components in the throttle body assembly. The components in the throttle body are described, and the steps taken are listed. The results and observations are also presented, and also compare to that obtained from the throttle body system interface in Chapter 7. The comparison brought discussions on how well the systems engineering perspectives are carried out in throttle body system, how well design knowledge is captured, understood, and presented.

Chapter 9 describes the Associativity Map (AM) method developed by the throttle body Direct Engineering team. DSM and AM are compared. The advantages and disadvantages of both methods are discussed.

Chapter 10 introduces the Datum Flow Chain method and applied it to the throttle body assembly design. The information contained in the DFC and DSM are compared and discussed.

Chapter 11 concludes the research findings in this thesis. Possible future research directions are discussed including the DSM method, what knowledge DE and other Knowledge Based Engineering system (KBE) should contain, and the impact of DE or KBE on organizations.

CHAPTER 2 FORD SYSTEMS ENGINEERING AND FPDS

2.1 Systems Engineering at Ford

2.1.1 Why Does Ford Promote Systems Engineering

As the complexity of vehicle design increases and the global competition grows more fierce, Ford faces the challenges of improving the quality of the products, reducing the cost and time to market. Ford responds to the global challenge with a new product development system—the Ford Product Development System (FPDS)—based on the Systems Engineering principles. Ford promotes Systems Engineering for the reason of correcting its traditionally part-centered engineering process. Some of the causes of Ford's part-centered design process are:

1. Ford approaches the design of a product as complex as a car through decomposition. A car is broken down to thousands of systems, subsystems, and parts at last. The decomposition approach turns the design of a complex product into manageable sizes, and makes a difficult task possible through collaboration of teams and engineers. Each system represents an engineering department at Ford. Subsystems are assigned to design teams. Individual engineers usually are responsible for the design of various parts.

   However, the reason for which Ford divided the vehicle into the existing systems and subsystems is not the same in every case. Some of the divisions are historical, such as the engine power train system. Some of the divisions are based on the manufacturing processes, such as the sheet metal system team. Some of the divisions are based on the functions the customers would relate to the car such as the instrument panel team. This inconsistency in the organization division created difficulty for communication across the organization boundaries.

Although decomposition makes engineers' jobs easier, since Ford is a company that designs cars rather than systems and subsystems in a car, it is also important to be able to integrate the design of systems into a functional vehicle. Part-centered engineering process usually misses the system level interactions at the early design stage, and causes late design changes when parts are put together to form the product, and hence delays the product development process and results in sub-optimal product quality. For this reason, Systems Engineering principles are much needed to improve the current situation.

2. The engineers have been assigned to design single parts rather than subsystems and systems. The metrics against which the engineers' job performance is measured drives people to think part-centered. Ford's current purchasing department has cost target on components rather than system. Engineers are directly rewarded for making cheaper and better parts but not cheaper or better system. However, many times, the total cost of the assembly can be reduced through cost tradeoffs between parts in the assembly.

   For example, the federal law requires two mechanical energy sources in the throttle body to close the throttle plate. Most of the Ford throttle bodies use either a dual wound torsion spring or a combination of a single wound torsion spring and an extension spring. The dual wound spring seems to have many advantages. It needs less space and so reduces the under-hood packaging issues (this point will be discussed in more details in a later chapter). The dual wound spring also eliminates the assembly step to put the extension spring in. Furthermore, the spring supplier said the dual wound spring would cost less than the total cost of a single wound spring and an extension spring. However, the torsion spring engineer was concerned that the dual wound spring cost would look higher than the single wound torsion spring, and hence for his component, it would look like the cost increased. He did not want that to be on his performance review.

If the engineers are not directly rewarded for the cost reduction of a system, or even worried to be penalized for the local cost increase due to cost tradeoffs made in the system, the engineers are going to be very reluctant to look at the system. They would instead concentrate on single parts, and this attitude contributes to the part-centered mentality.

3. Parts are of much smaller scale than systems, and hence are much easier to manage. The engineers are more comfortable to work with parts rather than systems that require a lot of interactions with many people outside of the organization. One engineer actually literally told the author that he liked to work with parts without thinking too much about the whole car. Recognizing this weakness is caused by the natural limitation of human beings, Ford as an organization needs to teach and reward its engineers to think at systems level.

4. The availability of the advanced CAD software also contributes to the part-centered mindset. The contribution can be summarized into the following three points. First of all, the CAD software has made creating and editing drawings much easier and faster than the manual process. This advantage of CAD software on the other hand has misled engineers into thinking they do not have to consider carefully about the assembly before they draw the part, because any changes will be easy to make if the assembly does not work.

   Second, the CAD software gives us an illusion that creating those beautiful 3D solid objects on the screen is very important. If those parts appear so perfect on the screen, they ought to be perfect when put together into a system. The author's research advisor

D. E. Whitney once said before the CAD software, there used to be a position called the layout man. This person's job was to design the assembly and each component's position and relation in the assembly before part drawings were ever made. However, there isn't a position like this at Ford any more. The engineers put their attentions on the beautiful 3D objects on the screens rather than thinking of a larger picture.

Third, the current CAD software is very good at creating individual parts. However, they have very limited ability of checking assemblies. Dr. D. E. Whitney has proposed three kinds of assembly models in the CAD system [Whitney (1996)]. The first type is "a non-connective model that places the parts in a world coordinate system in the correct relative positions and orientations but otherwise taking no note of the fact that they are assembled to each other." The second type "captures constraints such as 'against' or 'aligned' that are applied by the designer to various surfaces or axes on parts after they are designed." The third type is "a connective model that makes use of mating features." Most CAD systems today do well on type one assembly, and some of the CAD system does type two model such as interference check. However, neither of the two types is satisfactory in analyzing assembly analytically in the computer. Many times, these assemblies in the computers work fine, but when the manufacturing process noise is introduced, the actual parts can not be assembled. Hence, the engineers put more credence in the part design than in the assembly design. The part-centered mentality is enhanced.

The above four points are only part of the many reasons that made Ford engineering process part-centered. Yet these reasons are enough to make one seek changes in the Ford product development process in order to compete in today's global market. After all, a car is a large system. Ford's goal is to make the best and the least costly system, but not the best and least costly automotive parts.

2.1.2 Ford's Systems Engineering Thinking

In order to change the traditional part-centered engineering process and achieve the following corporate goals, Ford promotes systems engineering in the corporation [Ford Design Institute (1998)] in order to:

- Improve the quality of the product;
- Better Manage the complex products and processes;
- Meet development time constraints without sacrificing quality;
- Reduce the cost of development.

Ford defines systems engineering as "a customer/requirements driven engineering and management process which transforms the voice of the customers into a feasible and verified product/process of appropriate configuration, capability and cost/price" [Ford Design Institute (1998)]. The following characteristics are summarized as system thinking [Ford Design Institute (1998)]:

- System is greater than sum of parts;
- System is no better than weakest link;
- Optimizing parts does not optimize the whole;
- Interactions determine performance of system;
- Can't fully understand whole by breaking down and analyzing parts—yet design is historically done by the engineers through analyzing parts.

2.2 Ford Product Development System (FPDS)

The system thinking initiative has resulted in the new Ford Product Development System (FPDS). FPDS suggests a "V" shaped product development process (Figure 2-1: FPDS V). The left leg of the V shows that the customer requirements are cascaded to system, subsystem, and components design requirements through a formalized process. This process is highly iterative. The iterations take place mostly during the design and cascading process with the support of CAE tools. While the system requirements are developed, verification methods and process are also developed in parallel. The verification stage, which is the right leg of the V, will be mostly linear, and serve only as a process of verifying the design on paper.

FPDS differs from the existing product development process at the following points:

- Customer/requirements are focus at every level of the design.
- Requirements are linked and traceable.
- More up-front planning and experimentation.
- Fewer changes downstream and less fire fighting.
- Shorter development time and reduced resource cost through more use of analytical tools up front.
- Optimized cost through the development of an integrated system.

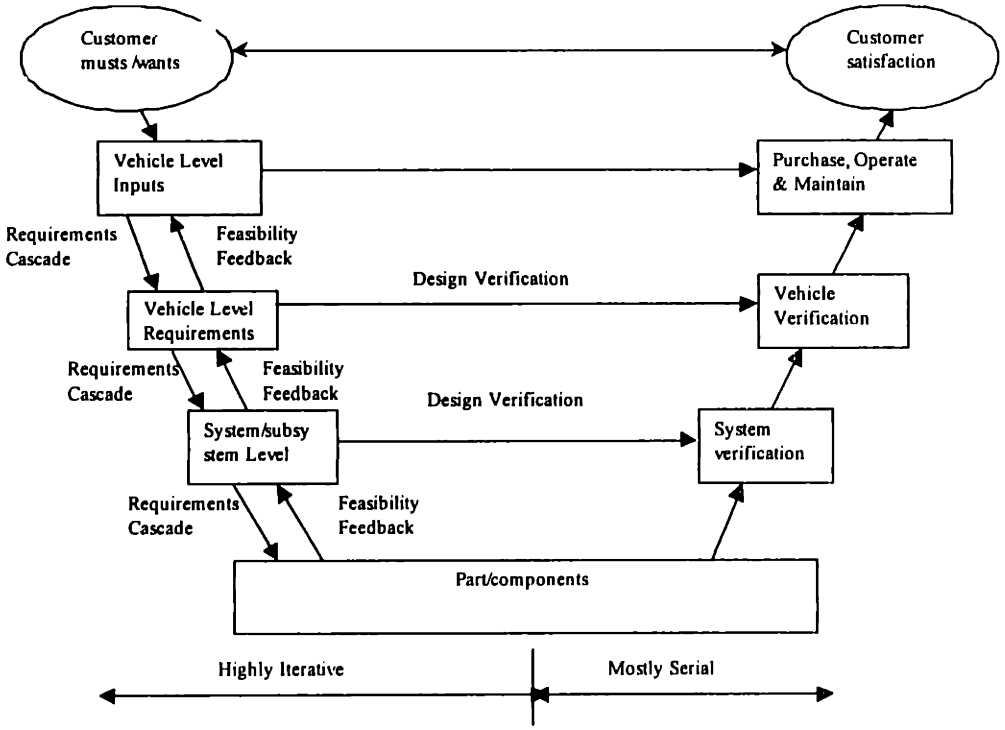


Figure 2-1 FPDS V

2.3 The Challenges that FPDS Faces

So far in our discussion, FPDS sounds like a good idea. In fact, more than 20 Ford vehicle programs have adopted FPDS in order to improve their product development process. However, when FPDS meets the real engineering and business world, it faces many challenges before entirely successful implementation. In this section, two of the challenges that relate to this thesis are discussed. The author would like to point out before going into the details of the challenges that FPDS is in fact a hypothesis that assumes Systems Engineering will improve Ford's product development process. The successes and challenges that FPDS faces are both great learning experiences for Ford to develop a better product development process.

2.3.1 Top-down vs. Bottom-up—Vertical Communication in the Organization

FPDS takes the top-down approach from the systems engineering method. The left side of the FPDS "V" model is a decomposition process. In the decomposition process, the voice of the customers and requirements from regulatory agencies and corporate standards are cascaded to vehicle level requirements, system level requirements, subsystem requirements, and finally requirements on the parts. However, manager and engineers have had difficulties to complete this top-down process as it is defined.

The vehicle level requirements exist but currently are inadequate to allow cascading to the lower system levels. A few examples of the content in a typical Vehicle Design Specification (VDS) are listed below. Due to proprietary reason, this thesis can not provide the year, model, and other details of this specific vehicle.

> "Making Country: USA
> Warranties: 36 months whole vehicle warranty, 60 months anti-corrosion warranty.
> Drive: Five-speed, manual transmission, rear wheel drive, limited slip differential.

> Lights: single halogen, complex surface lens headlights. Front fog lights.
> Wheels: 17x8.0 alloy wheels. Space-saver steel spare wheel."

The System Design Specifications (SDS) also exist for most of the systems and sub-systems in the vehicle. The following is a definition for SDS that the author found on the Ford Intranet:

> "A System Design Specification (SDS) is a comprehensive set of functional requirements which translate the wants of the customer, the corporation, the demands of regulatory bodies and the interfaces between systems and subsystems into measurable and verifiable generic engineering requirements. The sum total of these requirements equals the minimum performance required to ensure successful functional performance at the system or subsystem level. SDS requirements are stated generically; that is, they specify what the system must deliver but do not specify materials, structure or design. The SDS contains a description of generic system function, provides an authorized source for each requirement, lists all verification methods for each requirement and includes supporting details (graphics and diagrams) as needed to further explain requirements."

A typical SDS entry reads like the following. Due to proprietary reason, the author has hid the name of the component.

> "System A Clearance
> The nominal design clearance between system A and other attached system B components shall be 3.5-mm minimum, 2.0 mm under stack up conditions.
> Owner: XXXXX
> Last Modified: 00/00/00 00:00
> Requirement Priority: XXXXXX
> Verification Methods: DVM-0009-TH
> Requirement Sources: TBD"

The referred verification method code as well as other codes can be found in the rest of the SDS.

From reading the contents of the VDS and the SDS above, it can be deduced that the current SDS is developed independent of the VDS instead of being cascaded from VDS as suggested by FPDS. The existing VDS is not adequate to be cascaded down to lower levels. These observations were proved by authors of the SDS's for various subsystems and systems during interviews. Furthermore the SDS authors also indicated that they plan to combine SDS's together and make a useable VDS in the future. Hence the real process of developing the design specification at various level so far has taken an integration (bottom-up) process instead of the FPDS proposed decomposition (top-down) process.

So why don't people use the top-down approach when it appears so promising on the FPDS "V"? The author attempted to explain this observation with the following five points, which may be an incomplete list of all possible causes.

First, as this thesis has discussed earlier in this section, Ford has long held a part-centered engineering design tradition. Due to the sheer size of the corporation, it is hard to change from part-centered engineering process to systems engineering approach overnight. Ford has made tremendous effort to educate its engineers and managers. The result is that most Ford managers and engineers now understand the logic of the FPDS approach and the opportunities for improvements available. However knowing what FPDS is and why doing FPDS does not guarantee the engineers and managers to know how to implement FPDS. Currently at Ford, there isn't a clear process to follow in order to implement the top-down process in day to day work. The author observed during interviews that many departments try to develop their own FPDS processes on things that they can get their arms around. Therefore many independent FPDS islands emerged, and everyone hopes to integrate with the rest of the company in the future. If this situation lasts, the top-down process not only won't emerge itself, but also will be replaced by a true corporate-wide bottom-up approach. Furthermore, the FPDS islands will develop strong internal structure and become hard to merge with other islands. Ford FPDS communications manager David Roggenkamp believes that the key to the success of FPDS is to develop a set of templates which provide a frame work for people doing it 'the right way (top-down)'. The templates may be developed using a top-down approach, or they could be derived from a set of bottom-up details and then rationalized. The combination of both approaches may be most effective in reality.

Second, many of the FPDS enablers are not in place yet to help carry out the top-down approach, including C3P--the Computer Aided Design (CAD), Computer Aided Engineering (CAE), Computer Aided Manufacturing (CAM), and the Product Information Management (PIM). The FPDS literature suggests that C3P should provide all the analytical engineering tools needed to make early engineering decisions before the prototype is even built. Hence the re-design and fire fighting in the verification stage of the "V" will be mostly eliminated. C3P should also provide a system that can coordinate the engineering tasks according to the FPDS process. However, the reality is that C3P started earlier than FPDS. The two programs have been developed relatively independent of each other. Currently, engineers created data files in computers that had all the information about a product, but there wasn't any thought about the user interface with data model, and how the knowledge could be effectively retrieved. This weakness of the data model exhibits the character of a bottom-up approach, for people just created data blocks of low-level parts and systems and assumed users would use them automatically. Both C3P and FPDS teams have recognized these shortcomings and are working together to resolve them.

Third, the author observed that the communication between the upper level managers and the lower level engineers need to be improved. Ford high-level managers developed FPDS through benchmarking other world leading automotive industries. It sets time lines for each stage of the product development in order to help Ford stay competitive in the global market. However, even though people who developed the FPDS had 'real engineering world' experience and background, as argued earlier in the first point, the current FPDS is short of detailed action plans on how to meet those time lines. In order to find out how to improve the efficiency of the current design practice, the lower level engineers had to start with the parts and systems that they are familiar with, which resulted in the bottom-up picture. On the other hand, the higher level FPDS experts feel a great deal of difficulty struggling between the desire for greater detail (to allow engineers to know specifically what they are responsible of doing) along with the need for less detail (to remain manageable from a project management point of view). They feel things became unmanageable when trying to spell out all of the things that need to get done. Professor John Warfield portrays similar situations that exists in many large organizations [Warfield (1998)]:

> "What is being talked about in the lower levels of the organization are often highly detailed subjects. These subjects are never discussed at that level of detail in the higher levels of the organization...
>
> Empirical observations of groups who work at different levels in organizations have shown that the relationships that high-level people construct and apply among metaphors and categories simply do not correlate with the lower-level ideas that high-level people assume are encompassed within those metaphors and categories. The result is that the decisions and actions taken at high level in organizations often amaze the operating levels because they make so little sense and vice versa."

Hence, in order for the FPDS top-down approach to spread in the organization, better understanding and communication between the FPDS managers and the vehicle program engineers are needed. The methodology this thesis research has taken—the Design Structure Matrix (DSM)—can serve as a powerful tool to enhance the understanding between different levels of the organization, and provide a good overview to the engineering process. In addition, since as argued earlier in this section that Ford as a large-scale corporation has a large inertia to changes, setting practical goals and gradually evolve towards the benchmarking target may be an easier approach than trying to get to the bench marking target in a short period of time.

Fourth, component reuse also contributes to the bottom-up approach. Comparing its design with Toyota, Ford realized its cars are too styling-driven, and called for component reuse. Using past components brings known quality to the parts. Less time is spent on designing components and hence the development process will be faster. However, using existing components inevitably causes the engineers to design the system around the limited choices. Parts are considered before the system is designed, which causes the tendency of bottom-up approach.

Fifth, outsourcing many of the components and systems adds challenge to the top-down process. Suppliers are separate business units from Ford. Although as a big business, Ford has the power of altering some aspects of how the suppliers do the business, many times, Ford also has to accommodate the special conditions at the supplier side. Collaboration between Ford and its suppliers will cause Ford to think about the details of the subsystem and the components available from the suppliers before system design can be finalized.

The above five are only some but not all of the challenges that FPDS faces. It is worth debating whether a complete top-down approach is viable, or whether a mix if top-down and bottom up is the correct the approach.

2.3.2 Local and Non-local Knowledge—Horizontal Communication Between Organizations

The author defines the *local knowledge* as the knowledge that completely resides within the specific organization of interest, and *non-local knowledge* as the knowledge that is developed, kept, and controlled by organization(s) outside the specific organization of interest. Hence, local and non-local knowledge is a relative concept depending on the object of study.

The vehicle throttle body cam design provides a good example to demonstrate the above definitions. The throttle body cam is a mechanical linkage that opens and closes the throttle body plate, and regulates the amount of airflow into the engine. Since the cam is part of the throttle body, and is rigidly attached to the throttle body shaft, the cam design engineers need to understand the packaging of the throttle body assembly. The cam design engineers are part of the throttle body design team at Ford. Hence the knowledge about the throttle body assembly packaging is in the throttle body team, and can be called local knowledge of the throttle body team based on the above definition.

However, the cam design engineers also need to design the geometry of the cam curve which decides how much air flows into the engine in response to the amount that the driver steps on the gas pedal. The cam curve can vary the responsiveness of the car to the driver's foot rotation, and hence create different perceptions of the drive-ability of the car. Since the accelerator cable goes through the cam groove, the cam engineers need to work with the engineers in the accelerator control team to decide the cam groove geometry, cam curve, etc. The knowledge the cam engineers need from the accelerator control team is non-local knowledge of the throttle body team based on the above definition.

Any engineer with work experience will realize that engineering work can never escape the existence of the non-local knowledge. For a complex product such as a car, the collaboration between various organizations exists almost in all the component and subsystem design. Because not all the knowledge involved in a design is in the control of one local team, the communication between the systems and subsystems teams at the same level in the decomposition process is very important. The author defines this type of communication as *horizontal communication*. There are two reasons for which the horizontal communication at Ford should be of concern.

First, horizontal communication is needed by the day to day engineering design work. Ford has divided the vehicle into thousands of systems, subsystems and components. The component engineers that are in the same subsystem team need to communicate with each other to make sure the subsystem meets the requirements. The engineers in the subsystems teams need to communicate with one another to make sure the systems works. Due to the sheer size of Ford, different subsystem and system teams often reside at different locations that are far from each other (the situation may be a little better for the component engineers in the same subsystem team). The distance reduces the frequency and efficiency of communication. Meanwhile, Ford does not have a clear process of how inter-organizational communications are to be carried out. Most the time, experienced engineers know whom they need to talk to in different groups. However, when a new engineer joins the team, he/she just has to figure out the process by him/herself. Even worse, sometimes the experienced engineers are transferred to new positions before the new engineer gets trained. In a situation like this, the new engineer tries to find out clues in a spider web, and often cause delays and problems in the design project. For example, an engineer, who was responsible for the accelerator control design in one of the Ford trucks, went to a new position before the truck program was finished. A new engineer took over the job, and when the truck went into production, the factory line inspection found that the truck throttle cable was jumping out of the cam grove (the engineers call it "cable looping") when the pedal was suddenly released. It was discovered that the new engineer did not know that she had to check the cable-looping situation. The necessary horizontal communication was missing and caused a delay of the production and large stress on the engineering team.

Second, as mentioned earlier in section 2.3.1, the current FPDS development is in the systems and subsystems level. In order to integrate the SDS into vehicle level, Ford needs to understand how the systems at the same level interact with each other and create a common interface between systems. For example, during the author's interviews with the V-engine FPDS experts, they said they were concerned about how the V-engine FPDS would integrate with other systems such as the transmission system, in the power train department. The V-engine department made their FPDS process from the components up. They were worried if the transmission department used the top-down approach, and the transmission FPDS only has timelines without actions (see the discussion in point 1 and 3 in section 2.3.1), the action-packed V-engine FPDS would not be able to integrate with it. If Ford lets the current FPDS developments continue, it would have to inevitably face the questions of how to integrate all the system FPDS process in the future.

Based on the above two points, a method to understand the communication and information flow between systems in horizontal direction is needed. This thesis later will show that the Design Structure Matrix (DSM) is a very effective tool to resolve the problem of horizontal communication within an organization.

2.4 Chapter Summary

In order to change and traditionally part-centered engineering process and stay competitive globally, Ford proposed Ford Product Development System (FPDS) based on the systems engineering principles. FPDS suggests cascading the customer requirements into different levels of the decomposed vehicle design, and bringing most of the design iteration up front. In order to achieve complete success, FPDS faces the challenges of implementing the top-down design process, and enhance the communication between different design teams. These challenges are opportunities for Ford to re-examine the way it does engineering design and further improve its product development process.

CHAPTER 3 DESIGN INFORMATION REUSE AT FORD

3.1 Why Reuse Design Information?

In this thesis, the author defines *design information* of a particular product as the record of the decisions made on the design of the product as well as the reasons for the decisions throughout the product development and design process of the product. Reusing past design information has the following impact.

The first is that design information contains learned knowledge. The product design process is an iterative learning process. Engineers usually have to go through several trials to get the correct design. Design alternatives are usually evaluated carefully. For each decision made in the design, a lot of learning and knowledge are accumulated. For the same reason that we accept Newton's second law without going through the same process as he did to discover and prove it, what was learned in the past design is valuable for new designs. Design for Manufacturing and Design for Assembly theories are examples of using learned knowledge to improve new product designs.

Second, by extending the reasoning from above, we can conclude that reusing design information improves the efficiency of product development and design processes. For example, if previous design has shown an alternative is infeasible, the engineers don't have to spend time and energy to re-discover the same thing. They can just use the results from design history, providing the design history is correct and complete. The design information reuse frees us from reinventing the wheels every time. The time saved in product development and design stage can speed up the delivery of the product to the market, and hence have significant economic impact in engineering firms.

Third, reusing design knowledge greatly aids the redesign process. Many products today such as cars, home appliances, copiers, etc., have evolved from their previous models. The two consecutive product models usually only differ by changes and redesigns at parts of the product rather than the entire product. Chen, McGinnis, and Ullman suggest that if a complete and correct design history is available, the following situation will occur during redesign [Chen (1990)].

> "...not only will an engineer be able to understand how a design came into being, he will also be able to inspect the constraint dependencies and relations. These in turn should give insight on how the design will be affected by changing or modifying an existing constraint. The overall effect should be to decrease redesign time and improve the final design, since the designer will know both the source and reasoning behind all previously made design decisions."

Fourth, the engineer turn over rate makes recording and reusing design knowledge necessary. Ford encourages engineers to rotate among different positions. Usually when an engineer acquires deep understanding of a particular product, it is time for him/her to move on to the next position. In order to keep the efficiency of the product design and not to start from scratch, the new engineer should be able to use the knowledge learned by the engineers previously in the same position.

Thus, reusing design knowledge has signification impact on the product development and design processes. It is not surprising a company like Ford has paid close attention to this issue.

3.2 Ford's current means of knowledge re-use

The ideally complete and correct set of design knowledge as mentioned earlier by Chen's paper rarely exists in engineering firms [Chen (1990)]. Ford's existing means of design knowledge reuse are as follows.

3.2.1 CAD Drawings

When an engineering design is finished, the most common product of the design process is a collection of CAD drawings of the assembly of the product, its systems and subsystems, and parts. These drawings are saved in a corporate database. When the engineers later need to learn about previous products, they can retrieve the drawing from the database.

However, CAD drawings are unable to retain all the design information about a product, because much of the design information is unsuitable to be represented in geometric form. For example, the design intention of the product can not be stated merely by geometry. The choices of certain features over some the alternatives cannot not be explained by drawings. Hence, engineers can only extract limited amount of design information from CAD drawings. The limited amount of information may not provide enough help for later design, and the engineers will have to start over even on similar products.

3.2.2 Design Guide

Ford has initiated other non-geometric means of documenting the design information. Design guide is one of the many. Due to proprietary reason, the author was only able to see the design guides for throttle body. The observation made here may not apply to all other design guides.

Design guides are text documents that record the design and manufacturing rules concerning a specific product. Graphs and drawings are attached to the documents when necessary. The throttle body design guides consist of sections on the parts in the throttle body assembly, such as the cam design guide, the shaft design guide, the spring design guide, etc. The authors of these component design guides are experienced design engineers working on these components. The content of the design guide is inspected by the entire team before publishing, and is constantly being updated by the team.

The throttle body design guides serve as a very good complement to the CAD drawing. The design rules of each component learned from past design experiences are recorded. The design for manufacturing rules are particularly well documented in these design guides. Constantly updating the design guides provides opportunities for the team to summarize and share their learning.

However, the throttle body design guides have the following two weaknesses. First, the documentation is mostly on specific components. Later in this thesis, data shows that very limited knowledge has been recorded at the throttle body assembly and system interface level. The product the throttle body team makes is not the components but the throttle body assembly and the final product—the vehicle. Hence, the design guides are an incomplete collection of design knowledge. The main method used in this thesis work—the Design Structure Matrix—provides an effective means to record the knowledge at assembly level, as the reader will see later in this thesis.

Second, the design guides serve like a knowledge repository but are lack of easy knowledge browser. Past design knowledge is listed in pages of text and figures. In order to find a particular design rule for a component, the engineers have to read through the text to find out the information. Furthermore, if an engineer is working on an issue that concerns the throttle body assembly, he/she has to read through a large body of text in the design guide in order to find the complete information at the assembly level. Since human beings have limited short-term memory, it is possible that some design rules at the assembly level will be overlooked, and cause later design iteration. Therefore, an information browser that can effectively help the engineers to retrieve the information needed from the design guides is necessary. Later on in this thesis, the author will show that the Design Structure Matrix can serve this function very well.

3.2.3 Component Design Specification, System Design Specification, Vehicle Design Specification

As mentioned earlier in Chapter 2, the Component Design Specification (CDS), System Design Specification (SDS), and Vehicle Design Specification (VDS) are efforts of FPDS to decompose the customer requirements to engineering designs at every level of the vehicle. As a consequence, these documents provide design requirements rather than learned knowledge from past designs. Most of the requirements are general without specific suggestion of actions (See Chapter 2 of this thesis for the definition of SDS). Hence, these design specifications are records of assumptions and constraints on the design, and are not a complete record of the past design knowledge. On the other hand, while the design guides appear to lack information at subsystem and system level, the design specifications provide a connection between various subsystems and systems based on the design requirements. Yet this connection still lacks knowledge learned in the past at subsystem and system level.

3.2.4 Ford Intranet

With the development of computers and network, Ford has largely taken advantage of the corporate Intranet to retrieve and exchange information and data. Various organizations at Ford published design information on their web sites. However, the existence of Intranet does not necessarily mean the existence of complete and well-organized design knowledge for reuse for the following three reasons.

The first and most important reason is that Ford does not have a complete collection of past design knowledge. As discussed earlier in this chapter, and will be further proved later in this thesis, various aspects of the design are recorded in various forms. Some aspects of the design information, such as that at the assembly level, are not well documented at all. As any tool, Intranet can only be as good as the information going into it. Hence, without a complete collection of design knowledge, the Intranet can at most improve the speed of browsing the existing design information.

Second, web page design strongly affects the effectiveness of the Intranet. Various Ford departments and organizations have explored the best way of presenting their knowledge on the web effectively. More learning and experimentation are needed to enable the users to find information quickly.

Third, the effectiveness of the search engine largely affects the search results for design information. The previous version of the Ford Intranet search engine took the search words and listed out all the items that had these words in. A large amount of items in the search results are actually useless. It took the author more time to go through the search results than to just ask someone. In the summer of 1998, Ford changed the search engine to keyword search. The owner of each item published on the Intranet assigns key worlds to the item. Hence, the amount of search results is largely reduced. However, the price to pay is the risk of incomplete information. There is always the possibility that the keyword the authors can think of do not match those that the users can think of. Hence, it is possible that the users get even less design information out of the already incomplete information collection in the Intranet.

Fourth, the web information security check makes the information even less available to the users. Many Ford Intranet sites check the users' identification because some of the proprietary information is only available to those with security passes. Although this is not a problem this thesis can address, it just shows that the Intranet can only provide users a subset of all the information published.

In summary, for the above four reasons, the author concludes that having Intranet does not imply having a complete collection of design information for reuse. Further more, Intranet is far from a satisfactory tool for effective design information reuse. The work in this thesis, which will be presented later, suggests several new methods that can serve as design information browsers including the Design Structure Matrix method.

3.2.5 Knowledge Based Engineering (KBE) and Its Design Advisors

The Knowledge Based Engineering (KBE) at Visteon (formerly part of Ford) initially started working on incorporating design for manufacturing guidelines into the CAD software. The group developed computer codes that work with the Ford drafting software to check the manufacturing feasibility of a particular part's final geometry. These program codes are called *Design Advisors*, including Casting Design Advisor, Injection Molding Advisor, Sheet Metal Design Advisor, etc.

In comparison to the four design information reuse methods mentioned above, KBE design advisors present an attempt to actively apply the design information for the engineers. In other words, Design Advisors are automated design for manufacturing rule checkers. This is one step beyond merely providing a knowledge repository like the CAD drawings, the design guides, and design specifications. Comparing KBE Design Advisors with the Intranet, KBE actively use the design rules rather than passively wait for the engineers to find the design information.

However, the design advisors are still far from perfect design information reuse tools. First of all, since the Design Advisors are used in the CAD environment where information about the geometry of a part but not how parts relate to each other in an assembly exists, these Design Advisors can only be applied to parts. They do not provide any help at the assembly level. From the discussion in Chapter 2 in this thesis, a good design information reuse tool shall incorporate more than merely parts. Second, design advisors contain only rules of design for manufacturing, which is only part of all the design information needed for reuse. Third, design advisors are used on finished part drawings. In a corporation like Ford, CAD designers make the part drawings, which happens at the end of the product design process. The product design engineers do not use drafting software and do not create the details of parts. Hence, the Design Advisors help the CAD designers at the end of the design stage, and plays little role at the conceptual design stage.

The KBE group at Visteon has started expanding their efforts from the Design Advisors to a more intelligent computer-aided design system. Another group named Direct Engineering (DE) at Ford also approaches the design knowledge reuse issue from the same angle. The methodology and the products of the new KBE work and DE are different in details but very similar in their general ideas. The author has spent total six months over one year and half period with the DE group but only a number of meetings and demos with the KBE group. Hence, in stead of discussing the new KBE programs, the author will devote the next chapter to the DE approach. Many of the situations that DE faces are also challenges the new KBE work.

3.3 Chapter Summary

Reusing design information is necessary to improve the design efficiency and keep the continuation of the learning of the organization. Conventional design information reuse methods at Ford such as CAD drawings, design guides, Component Design Specification, System Design Specification, Vehicle Design Specification, Intranet, and traditional KBE approach are lack of system level knowledge. The users cannot retrieve the collection of knowledge effectively. Hence a better method of collecting, organizing, and representing the design information for reuse is needed in order to improve the efficiency and quality of the product development process, and strengthen the global competitiveness of the corporation. The Direct Engineering method in Chapter 4 will present the one of Ford's approach facing the above challenges.

CHAPTER 4 FORD DIRECT ENGINEERING

The details of Direct Engineering are regarded as Ford's proprietary material. It is not the emphasis of this thesis either. Hence the author will only present a brief overview of the Direct Engineering concept and discuss a few points in the Direct Engineering that relate to the work in this thesis.

4.1 Overview of Direct Engineering

Direct Engineering (DE) is a new developing semi-automated engineering design process at Ford. Its definition, vision, and principles are presented in Table 4-1 below.

Table 4-1 Ford Direct Engineering

***Direct Engineering**: A new engineering process supported and enabled by advanced technology, refocusing and realigning of the engineering activities. Direct Engineering integrates product and manufacturing engineering activities into a seamless and cohesive process.*

***Vision**: To provide an environment that allows an engineer to consider both product and manufacturing requirements throughout the design/development/manufacturing cycle, resulting in a Total Product Definition built upon the collective intellect of the organization.*

***Principles**: --A single unified concurrent engineering process*
--An integral knowledge management process
--Seamless knowledge delivery and application
--Rapid development of a total Product Definition
--An "engineering centric" environment
--The home organization owns and leads the effort

The DE concept is deployed through a computer based tool—the *DE Modeler*-- with takes advantage of existing and emerging computer technologies to assist engineers in developing the Total Product Definition for a component, subsystem or system. The *Total Product Definition* is the specification for a product which includes (but not limited to) product geometry, product bill of materials, manufacturing feasibility, process planning, cost information, etc.

4.2 Direct Engineering's Approach to Design Information Reuse

In comparison to the conventional design information reuse tools mentioned in Chapter 3, the Direct Engineering approaches design information reuse from different angles. Some of the differences that are of the interest of this thesis are discussed as follows.

First, DE tries to incorporate all the design information (The Total Product Definition) needed in the DE modeler. As discussed in Chapter 3, the conventional means of design information reuse such as the CAD drawings, the Component Design Specification, System Design Specification, and Vehicle Design specification, etc. each provides only part of the information that design engineers need for decision making. A DE modeler contains information and rules from all the conventional information reuse tools and even more. Hence when a design engineer is using the DE modeler to design a product, he/she appears to have all the design and manufacturing experts in the room, and can check the design against all learned knowledge as he/she proceeds. Unnecessary design iterations are eliminated, and the design information reuse becomes very easy and efficient. Further more, the politics and bureaucracy involved in communication and inquiry are taken out of the picture.

Second, DE goes beyond a knowledge repository of all the design information. Many times, even with the existence of a collection of design information for reuse, design engineers still need to not only be able to find out what information they need to know at what stage of the design process, but also where to get the information. It is fairly common that some of the information available is neglected and then later causes unnecessary design iterations. DE studies the design process and the logic behind decision making in a product design, and builds the DE modeler to prompt the engineer to follow the design process and use the design information available. Hence, design engineers do not have to worry about forgetting what information is needed at various stage of the design. The DE modeler leads the engineers through the process.

Third, DE is not just about making a computer tool—the DE modeler. Instead, it carries out an engineering process of capturing, managing, and representing design information for reuse. In order for the DE modeler to have the Total Product Definition, DE developed a process to collect and summarize the design information about a particular product. This knowledge capturing process provides communication opportunities among engineers and managers from various departments of the business. Often, problems are discovered and solutions are discussed in these DE meetings. The product design is thus improved even before the DE modeler does anything, because a better understanding towards the design and the process is achieved through the better communication. Furthermore, needing all the information about products requires DE to discover ways to manage the information database. Building the DE modeler also requires DE to consider appropriate user interface. Hence, unlike traditional design information reuse tools, DE is actually looking at the entire picture of the design information reuse process.

4.3 The Challenges that Direct Engineering Faces

Direct Engineering, if successfully implemented, provides a good tool for information reuse. However, DE needs to overcome many challenges in order to turn the ideas into reality. Some of the challenges that the author observed during the process of thesis research include:

- Capturing design information for the Total Product Definition
- Automating the engineering design process
- Flexibility of the DE applications
- Standards of DE applications
- Assumptions about the User's Knowledge
- Work redistribution

The rest of this section will discuss each of the above points in details.

4.3.1 Capturing Design Information for the Total Product Definition

As discussed in the last section, DE tries to capture all the design information in a product design. However, capturing design knowledge faces the following two issues.

First, as this thesis will show later in Chapter 7, a large amount of design information is in engineers' and managers' head. According to Pete Sferro, the acting manager of the Direct Engineering department and the senior staff technical specialist, many people feel their job security is threatened, if they give up their knowledge and let it be put it into a computer. DE needs to find out ways to ensure people's job security before it can obtain a complete collection of information needed for Total Product Definition.

Second, the amount of information involved in a product design is large and complex. In order to deal with this problem, DE needs methods to efficiently acquire information from engineers and managers. Furthermore, DE needs to have a means of organizing the information collected, and turn the information into logic relations, for the DE modeler is a computer program that only takes logic statements to judge the design. In the throttle body DE case, the DE team's current practice is to call for meetings among engineers to collect design information. An Associativity Map (AM), which will be discussed later in Chapter 9, is then created to guide the programming of DE modeler. The Design Structure Matrix (DSM) method used in this thesis research proves to be a more effective method for organizing the information and discover the logic (See Chapter 9).

Third, DE creates communication channels between different engineering organizations. Hence, very often, during the DE meetings, the engineers discover the existing way of doing work is not efficient. Many suggestions for improvements towards the process and the product emerge. On one hand, these suggestions for changes are part of the product the DE process expects. However, on the other hand, support from the management is usually insufficient to implement the changes suggested in the DE meetings. Hence the DE program constantly faces the dilemma whether they should reflect the existing information and process, or the way things should be.

4.3.2 Automating the Engineering Design Process

The goal of the DE modeler is to create a semi-automated design environment where all the design information is available to the design engineers who are using the program. However, automating the engineering design process using computer programs creates the following concerns.

First, it is worth debating what engineering process the DE modeler should use. In fact, the current Ford product design practice does not follow any documented process. There is no flow chart or formalized steps to follow when engineers design a product. Design engineers usually find information they need through experience, networking, and gradual learning. This situation is one of the causes of iterations and inefficiency in the design process, and is hence one of the problems DE tries to solve. However, by the same token, it is hard for the DE modeler to promote and formalize a single process when the traditional engineering process is largely random and informal.

If DE can find a process that is close to the practice of most design engineers, the engineers will accept the process easily. However, the existing design practice is far from perfect. During DE meetings, the communication between engineers often results in the recognition of the problems in the existing design process. Spending resources just to copy the existing problematic process into the computer appears unnecessary and uneconomic.

If DE engineers promote better design processes through the use of their modeler, many other problems appear. First of all, it is hard to decide the should-be process. The Design Structure Matrix in this thesis work provides a methodology to determine a more efficient design process based on information flow, which can be used to determine what process the DE modeler should take. Second, to promote a new design process is to change the way engineers work. This change takes more than just a computer tool to implement. Before the

DE modeler can gain full support from Ford top management, it is hard to expect engineers to welcome the new DE modeler.

Second, DE needs to put great effort in providing engineers learning and developing experience once the DE modeler is put into use. In his book Trapped in the Net, Professor Rochlin from University of California at Berkeley presented many examples of the impact of automated processes on the human reactions. He suggests:

> "Human learning takes place through action. Trial-and error defines limits, but its complement, trial-and-success is what builds judgement and confidence. To not be allowed to err is to not be allowed to learn; to not to be allowed to try at all is to be deprived of the motivation of learn" [Rochlin (1997)].

Hence, if the DE modeler could present a perfect design process to the engineers, the engineers would not have the opportunity of going through the process of learning how to think in designing a product. In short run, DE modeler takes out the learning curve that engineers have to go through and makes the design process more efficient. However, in the long run, the engineers may loose the ability of discovering and identifying opportunities for improvement. After all, what could be a better way of learning swimming than just actually get into the water? Even worse, the engineers may become unable to identify the right design process when a new product need to be designed and a new DE modeler needs to be constructed.

The DE engineers and managers have thought of this problem. Pete Sferro believes the ideal DE program should be hot-linked to the details that explain each event in the modeler to the user. This includes the logic, the math, past experience along with the reasoning and who is responsible for each rule or event. Engineers have immediate access to any and all knowledge of the DE system at any time. Furthermore, if the engineers wish to go outside the limits of the DE system knowledge domain and create something totally new, the disciplined DE process is there to ensure that new knowledge is captured and integrated into the based

DE system. DE should accelerate the learning curve of new qualified engineers and provide the fundamental base for the growth of new knowledge. From the observation of the existing DE programs, the author believes that a great deal of hard work is still needed by the DE engineers in order to realize the above visions.

Third, the implementation of DE is a gradual process. When some processes are switched to DE, many other parts of the organization may still be using the traditional processes. Based on his past work experience on work flow applications, George Roth at MIT argues that the hybrid processes in an organization may actually take longer time than the traditional process because many conflicts between the old and the new systems have to be worked out. Hence, automating the design process may not be the solution to all.

4.3.3 Flexibility of the DE Applications

As new technology emerges everyday, product design changes very quickly. The design process and logic behind all the judgements change according to the product configuration. Since the DE modeler takes a certain amount of time to construct, DE engineers should find out ways to make the modeler keep up with the design changes and be able to justify the investment of resource. For example, the throttle body DE modeler is made for aluminum straight bore configuration. Pretty soon, the throttle body design is going to change into plastic progressive bore, however the aluminum throttle body modeler is not yet put in use. Modularize the DE program and separating blocks of design rules may be part of the means of resolving this situation.

In the history of manufacturing automation, human beings have been proven to be more flexible than they were thought of. No robot can be built economically to have the same degree of flexibility as human hands. Hence, it is worth further investigating that what the proper degree of automation is in the design process. This investigation will be more difficult than that for the automation of factories because the economic impact of the design activities is not so clear-cut as the hardware. Also, the flexibility of human thoughts are harder to model and analyze than the manufacturing processes.

4.3.4 The Standards of DE Applications

The existing DE applications are on small systems in the car, such as throttle body, AC hose, etc. A car contains thousands of systems like these. Sooner or later, DE applications will face the problem of integrating individual DE application islands into the entire car. In order for the integration to be successful, DE applications have to be able to communicate with one another. Hence it is important to set standard interfaces among DE applications. The work in this thesis has studied the throttle body's interface with other systems, and provided help for the future integration of throttle body with other systems.

Since the entire Ford Motor Company is trying to implement the top-down design process proposed by FPDS, developing DE application islands and integrating them seems to be counter-acting. Ideally, DE as a knowledge reuse program should develop parallel to the corporate-wide product development process. Hence the question rises regarding whether DE should develop a huge program that designs the entire vehicle instead. Of course, it is also of concern whether such a huge program is feasible to make currently, since the understanding of the design process and design knowledge of the entire vehicle is currently very limited.

4.3.5 Assumptions on the User's Knowledge

The DE modelers have to make assumptions on what the users know. These assumptions may greatly affect the effectiveness of the DE modelers. For example, my research advisor D. E. Whitney observed that the AC Hose DE modeler assumes the user is an experienced AC hose engineer. The computer hardly gives hint on what the user needs to do step by step to complete the assembly design. Hence a person who is unfamiliar with the hose assembly is not able to use the modeler. The DE program managers state that the DE system requires a qualified engineer who is capable of truly handling the scope of a complex systematic engineering problem. Hence it is important for the DE team to provide appropriate training program to engineers. Otherwise, engineers may make mistakes in the decision making if they did not know the assumptions and restrictions made on a particular DE program.

4.3.6 Work Redistribution

On the surface, the implementation of DE modeler will reduce the workload of design engineers by improving the design efficiency. However, the reality may not be so ideal. Building the DE modeler actually moves the knowledge and logic involved in a design process from engineers' head into the computer database and program codes. The DE modeler requires people who are knowledgeable about the software to construct and maintain the modeler and database. The DE theory suggests expanding the role/responsibility of the engineers to encompass functional responsibility of capturing the knowledge, recording it into the corporate database, encoding it into the DE modeler, and functional maintenance of the system. Hence, if DE succeeds in achieving the above objectives, the workload of the engineers may not be reduced as much as it appears. In addition, the specialists in computer software and Information Technology (IT) are needed to create the corporate functional IT architecture (database structure, communication links, etc.) to support the DE systems. Therefore, it is worth evaluating whether the total workload of the design process is reduced and whether the workload shift is economic.

Furthermore, traditionally, design engineers at least hold a formal Bachelor's degree from engineering schools and are well trained in their thinking and engineering intuition. However, most software specialists do not have the understanding of the fine details of design and manufacturing processes. Hence, taking the thinking role off the engineers and letting the software specialists develop programs that mimic engineer's decision making processes seem to be a little risky.

4.5 Chapter Summary

Directing Engineering (DE) is a developing semi-automated engineering process at Ford. If successful, DE can provide means of effective knowledge reuse and abundant opportunities for improving product development process. However, it also faces the challenges of effectively capturing design information, automating the engineering design process, balance the flexibility of the DE applications, standardize DE applications, making correct assumptions about users, and the challenge of justify the workload redistribution.

The history of manufacturing automation has shown warning signs to any automation of processes. David A. Hounshell from Carnegie Mellon University has argued the inflexibility, the lack of standards, the breakdown and maintenance, and the redistribution of work in the plants are the main issues concerning the transfer machines [Hounshell (1996)]. The same concerns apply to the automation of the design process. Only when DE finds solutions to the above challenges, could DE realize its vision.

Part II

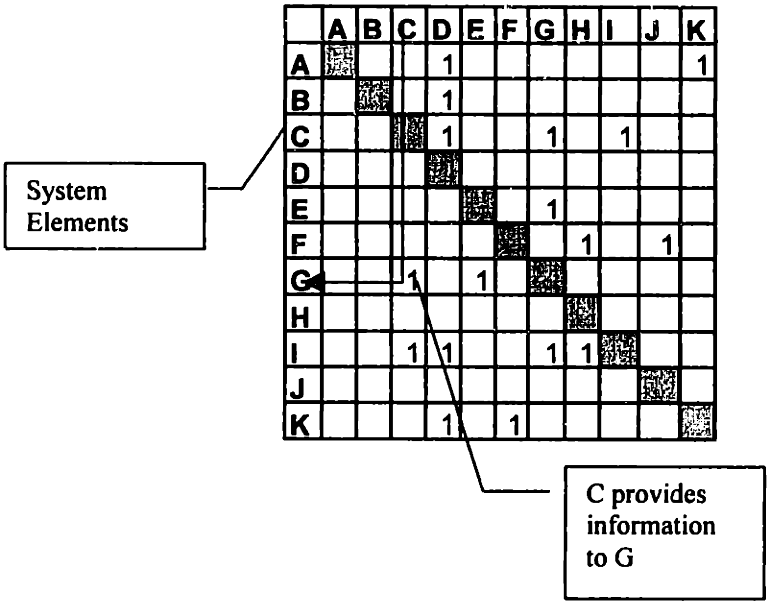
Thesis Research Methodology and Background

CHAPTER 5 DESIGN STRUCTURE MATRIX

5.1 Introduction to Design Structure Matrix Method

The Design Structure Matrix (DSM) is a tool that helps capture the important relations in a complex system such as that in a design project. The purpose of applying the DSM method to a design project is to understand the information flow and communication in the design process, and hence to seek improvements based on better understanding of the system. DSM has many similarities with PERT chart [Wiest (1977)]. However, later in the thesis, the author will show that the PERT chart cannot represent the iteration loops that commonly appear in a real design process. Thus DSM can present a more complete view in project planning.

An example of a binary DSM is shown in Figure 5-1. The author defines the *System Elements* as the title of the each row and column of the matrix. In this example, the system elements are *A, B...K*. The matrix is square. In other words, the rows and columns represent the same elements in the same order. Each entry in the DSM (marked by an "X" in this example) indicates that the element in the corresponding column provides inputs to the element in the corresponding row. For example, in Figure 5-1, an "X" exists in row *G* column *C*. Therefore, *G* needs information from *C*, i.e., *C=>G*. In a binary DSM, the diagonal elements do not have much meaning. Hence in this thesis, the author usually shades them without giving any value. However, a DSM does not have to be binary. Non-binary DSM will be discussed in the literature review later in this chapter.



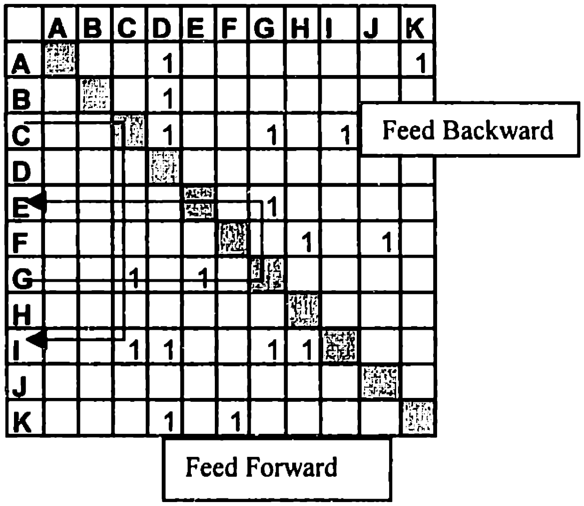
	A	B	C	D	E	F	G	H	I	J	K
A	▨			1							1
B		▨		1							
C			▨	1			1		1		
D				▨							
E					▨		1				
F						▨		1		1	
G			1		1		▨				
H								▨			
I			1	1			1	1	▨		
J										▨	
K				1		1					▨

Figure 5-1: A Binary Design Structure Matrix (Unpartitioned)

The matrix entries above and below the diagonal have different significance. Assume system elements *A* through *K* are the tasks we need to complete, and we are going to complete the tasks in the order of *A* through *K*. The matrix entries below the diagonal shows upstream tasks feeding information to downstream tasks. For example, in Figure 5-2, an "X" is in row *I* column *C*, which means element *C* provides inputs to element I. Since *C* is finished before *I*, as long as element *I* asks, information from *C* is ready and available. The below diagonal region is called *Feed Forward Region.*

Entries above the matrix diagonal have different effects. An entry above the diagonal shows upstream tasks requesting information from downstream tasks. For example, in Figure 5-2, an "X" exists in row *E* column *G*, meaning *E* needs inputs from *G*. However, *G* cannot be finished before *E*. Hence *E* has to make an estimate on *G*. When *G* is finished later on, the initial assumptions made about *G* have to be compared to the end result. If the initial assumptions on *G* are very different from the end results of *G*, *E* has to be reworked, and the new result of *G* has to be checked against the new assumed *G* until the assumptions and the results become close enough.

In a math or engineering problem, the above process is called iteration. Iterations are common in design projects. For example, on a vehicle car door, the window rubber seal surfaces contribute to the friction force on the glass, and hence the motor power requirements. Assume the seals are designed before the motor is chosen. Later in the design program, the engineers may find the final motor configuration cannot provide the required power, then the seal engineers have to redesign the seals to meet the motor power requirements. Or a different motor must be used. However, the new motor may have larger size than the old one and so the sheet metal design have to be changed. This situation is an example of design iteration. Design iterations create reworks, negotiation, meetings, etc. Design iterations require close communications between various parties involved and usually slow down the progress of the design process. Hence, in order to speed up the design process, we shall try to eliminate design iterations.

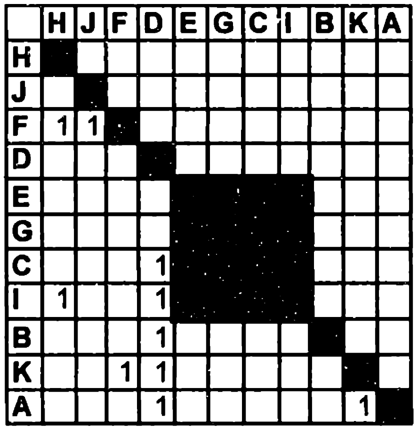


	A	B	C	D	E	F	G	H	I	J	K
A				1							1
B				1							
C				1			1		1		
D											
E							1				
F								1		1	
G			1		1						
H											
I			1	1			1	1			
J											
K				1		1					

Figure 5-2 Feed Backward and Feed Forward Regions

Since the above diagonal entries in a DSM are potentials for design iterations, we shall try to reduce the amount of entries in the feed backward region. When all the above diagonal marks cannot be eliminated, efforts should be put to move the above diagonal marks close to the diagonal. The closer an above diagonal mark is to the diagonal, the fewer system elements are involved in the design iteration, and hence the faster the iteration will be. The process of rearranging the order of the system elements in order to reduce the amount of the above diagonal elements or move them closer to the diagonal is called *partitioning the DSM.*

Figure 5-3 shows the same matrix in Figure 5-1 and Figure 5-2 after partitioning. The order of the system elements is changed. As a result, fewer elements are above the diagonal. However, several above diagonal elements cannot be moved below diagonal no matter how the order of rows and columns are rearranged. Hence, one design iteration loops exist, which consists of elements *E, G, C,* and *I.* If system elements *A* through *K* are the tasks to be completed, the most efficient way is to complete the tasks in the order suggested by the partitioned DSM.



	H	J	F	D	E	G	C	I	B	K	A
H	■										
J		■									
F	1	1	■								
D				■							
E					■	■	■	■			
G					■	■	■	■			
C				1	■	■	■	■			
I	1			1	■	■	■	■			
B				1					■		
K			1	1						■	
A				1						1	■

Figure 5-3 DSM after Partitioning

Besides iteration loops, the result of partitioning in Figure 5-3 also shows the order of completing tasks. For example, tasks *H* and *J* must be completed before *F* because *F* requires inputs from *H* and *J*. However, *H* and *J* can be completed independently in parallel because they do not require inputs from each other. The actual order of completing tasks is better illustrated by the digraph discussed in the next paragraph.

The relations depicted by a DSM can also be represented in the form of a graph with directional arrows, which is called a *digraph*. A digraph corresponds to the DSM in Figure 5-3 is shown in Figure 5-4. The digraph shows the order of completing tasks. The elements in level 1 shall be completed first, then those in level 2, and so on. Furthermore, the elements in the same level can be completed in parallel if they are not in design iteration loop. For example, *H, J,* and *D* can be done in parallel because they don't need input from one another. For the elements in the same level that are involved in design iteration, such as *I, C, E*, and *G* in this case, negotiation and collaboration are required. However, the iteration loop as a whole can still be done in parallel to the rest of the independent elements in the same level. For instance, *I, C, E*, and *G* as a group can be done in parallel with *F* and *B*. Every DSM has a corresponding digraph. Digraph is more effective in visualizing the order of completing tasks. However, when the number of tasks increases, the digraph becomes cumbersome. Then the DSM is more effective in finding the optimal order. More discussion on the comparison between DSM and digraph will be presented later in this thesis in the case study of comparing throttle body Associativity Map to DSM.

The partitioning result provides the first step and the mathematical result of optimization. Based on the experience and knowledge of the particular system, many other techniques can be used to deal with the iteration loops, and further optimize the system. In the rest of this section, the author will discuss the following methods briefly:

- assigning dependency values
- tearing
- de-coupling, and add-coupling.

Each method serves different purpose and provides different insights to the problem. The situation and the purpose of the study shall be evaluated in order to choose an appropriate method.

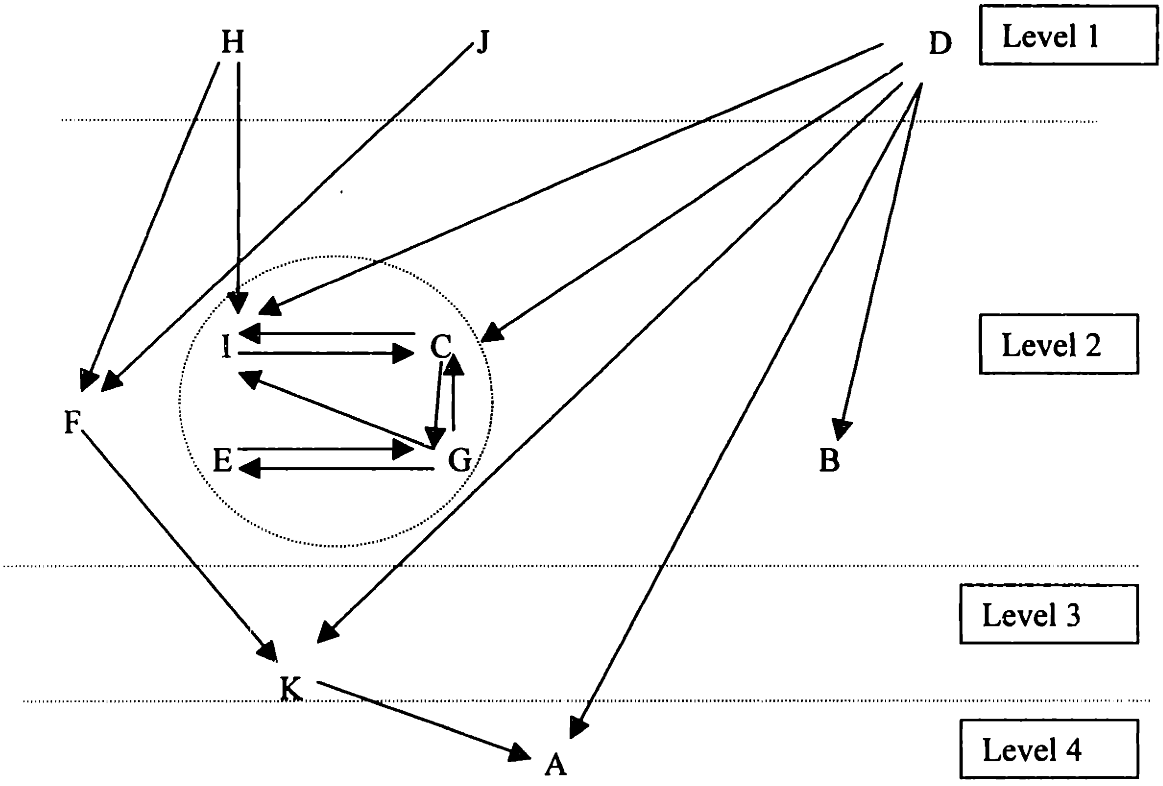
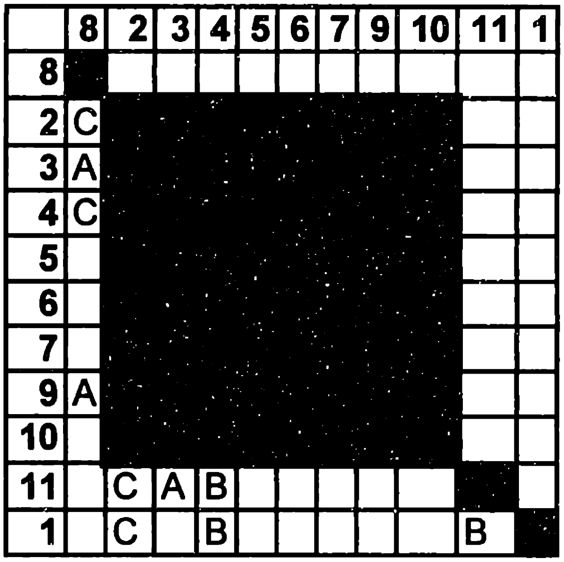


Figure 5-4 The Digraph Corresponding to the DSM in Figure 5-3

Assigning Dependency Values

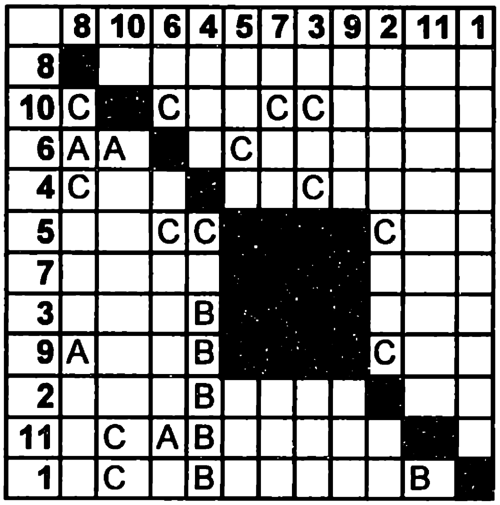
The relations in a matrix may have different weights. Some of the information flow may be critical for decision making, while some others may be ignored in order to make a decision quickly. Instead of considering all the relations with equal importance, one may ignore those less critical relations and only include the important relations in an iteration loop. By doing so, the size of the iteration loop may decrease, and the project can go through faster. Figure 5-5 shows an example of DSM with weighing factors. If all weighing factors are treated with the same importance, the partition result shows a large iteration loop among element 2, 3, 4, 5, 6, 7, 9, and 10. However, if the C relation is considered of little importance than A and B, the partition result shows a much smaller iteration loop (see Figure 5-6). Using weighing factors helps to identify the critical elements that dictate the iteration loops. The process can be organized to better around the critical elements, and hence the speed of the design process can be improved. A good example of this method is illustrated in McCord and Eppinger's paper [McCord (1993)]. Some other practices of using non-binary DSM are discussed in section 5.2.



	8	2	3	4	5	6	7	9	10	11	1
8	■										
2	C	■	■	■	■	■	■	■	■		
3	A	■	■	■	■	■	■	■	■		
4	C	■	■	■	■	■	■	■	■		
5		■	■	■	■	■	■	■	■		
6		■	■	■	■	■	■	■	■		
7		■	■	■	■	■	■	■	■		
9	A	■	■	■	■	■	■	■	■		
10		■	■	■	■	■	■	■	■		
11		C	A	B						■	
1		C		B						B	■

A: high dependency B: medium dependency C: low dependency

Figure 5-5 Sample DSM with Weighing Factor (after Partitioning)



	8	10	6	4	5	7	3	9	2	11	1
8	■										
10	C	■	C			C	C				
6	A	A	■		C						
4	C			■			C				
5			C	C	■	■	■	■	C		
7					■	■	■	■			
3				B	■	■	■	■			
9	A			B	■	■	■	■	C		
2				B					■		
11		C	A	B						■	
1		C		B						B	■

A: high dependency B: medium dependency C: low dependency

Figure 5-6: Sample DSM after Partitioning Based on the Dependency Value

Tearing

Instead of assigning dependency values, Dr. Steward suggested the tearing method to minimize the iteration time of the loop [Steward (1981)]. Dr. Steward defines tearing as to choose the "X's" so that if they were removed from the loop and the variables in the loop were re-ordered by partitioning, no "X's" would appear above the diagonal. This means having made these estimates, no additional estimates need to be made. Tearing is also a way to find out the critical-to-speed elements in the system. However, this is not the main method used in this thesis. The readers may find more details about this method in the referenced paper mentioned above.

De-coupling and Add-Coupling

Most of the time, there is a trade-off between the quality of the design and the speed of the design. De-coupling is a method to remove some of the relations (*X's*) between elements so that the iteration loops will be smaller. However, since some of the information transfer is missing, the quality of the design might be hurt. Add-coupling is to add information transfer (*X's*) to enlarge the iteration loop. The result is an increased design quality. Concurrent engineering is an example of add-couplings. Dr. Eppinger, Whitney, Smith, and Gebala have more detailed explanations on De-coupling and Add-coupling their paper [Eppinger (1994)].

5.2 Related Works

Besides the works mentioned above, many other researchers have explored the DSM methods from various angles. Ledet and Himmelblau provide detailed background introduction of the mathematics behind DSM [Ledet (1970)]. Professor John Warfield's Societal systems : planning, policy and complexity also contains sections on the fundamentals of structural mathematics [Warfield (1976)].

Many algorithms have been developed to partition the matrix mathematically. Ledet and Himmelblau presented the method of partitioning the matrix using the reachability matrix method [Ledet (1970)]. Professor Warfield has a step-by-step procedure on how to partition the matrix after obtaining the reachability matrix [Warfield (1973)]. Dr. D. Steward suggests a different method for partitioning, which is called the loop tracing procedure [Steward (1965)]. He has also developed software for partitioning. Dr. Rogers at NASA has developed a computer algorithm—DeMAID for partitioning [Rogers (1989), Rogers and Padula (1989), Rogers (1997)]. During this thesis work, the author has developed a visual BASIC program in Microsoft EXCEL based on the reachability matrix theory. The details of the program are in Appendix A.

Besides using the binary DSM, many researchers have explored assigning values to the DSM entries. Robert Smith and Steven Eppinger developed methods to predict the slow and rapid convergence of iteration within a project by assigning dependence measure to the interactions, and task times on the diagonals of the DSM [Smith (1997)]. Pimmler and Eppinger not only assigned the dependency level to the interactions, but also categorized the interaction in order to present more details of the design process [Pimmler (1994)]. Some the research work estimates the probability of completing a product development process over time based on DSM frame work [Krishnan (1997), Carrascosa (1998)].

During the course of this thesis research, the author found that DSM is not only a good tool for recording the relation between tasks and re-sequencing the tasks, but also a good tool for browsing learned design knowledge. In addition, the process of making DSMs involves understanding and capturing the relation between tasks, which requires knowledge on information management in an engineering design process. The following is a list of research work on this aspect. Chen, McGinnis, and Ullman give a very good overview of knowledge capturing, managing, and reuse [Chen (1990)]. Stauffer, Ullman, and Dietterich studied the engineer's thoughts during a design process using a technique from cognitive psychology, known as protocol analysis [Stauffer (1987)]. Khadilkar and Stauffer conducted experiments to determine the usefulness of design history information and to establish the need for providing the conceptual level design information for future reuse [Khadilkar (1996)]. Similar research is also done by Baya, Gevins, Baudin, Mabogunje, Toye, and Leifer [Baya (1992)]. Kuffner and Ullman studied and categorized the information requested by the design engineers in order to understand what information is needed by intelligent CAD systems [Kuffer (1990)]. Chovan and Waldron studied the information reuse provided by CAD drawings [Chovan (1990)].

5.3 General Procedure Used in This Thesis

The research work in this thesis took the following steps in general. This section provides general explanation for each step. The actual situation of each case study may vary in details, and the variations will be discusses later in each chapter.

Step 1: Define the system and its scope.
Step 2: List all the system elements.
Step 3: Study the Information Flow between System Elements.
Step 4: Build a matrix to represent the information flow and verify the matrix with engineers;

Step 5: Partition the matrix;
Step 6: Optimize the system;
Step 7: Give the matrix to the engineers and managers.

Step 1: Define the System and Its Scope.

Since the DSM is a tool that studies the design process as a system with many interacting elements, it is important to define the boundary of the system in order to focus the research work. Different system definition results in different output of the DSM, as the thesis will show later between the throttle body system interface DSM and the throttle body assembly DSM.

Step 2: List All the System Elements

Initially, the system elements can be chosen based on the existing project plans, engineers' suggestions, etc. In this thesis work, the author usually defines the initial set of system elements based on the reading of design documentation. However, experience shows that the initially defined system elements often need to be modified in the process of assigning interactions to them. This point will be further discussed later on in this section.

Step 3: Study the Information Flow between System Elements

The third step is to study the information flow between system elements. In this thesis research, the author usually read the design documents as well as interviewed experienced engineers who were working on the particular product. Interviews are just as important as reading design documents for two reasons. First, not all the knowledge is well captured by design documents. A large amount of information exists in engineers' heads. This point will be proved later in Chapter 7 based on the data in this research. Hence, it is important to extract the undocumented knowledge from the engineers.

Second, interviews seem to be an effective means of extracting knowledge from the engineers' mind compared to other methods. During the thesis research work, the author found that different engineers often had different views on how one element related to the other and how important the relation was. The causes of the differences could usually be one of the following two:

- The interaction was not direct. For example, the *Belt Seals* affect the motion of the window *Glass* by exerting seal drag force on the glass. The seal drag force and the motion of the glass are factors that affect the motor design. Thus the relation is

  Belt Seals => Glass => Motor.

  However, often, the engineers would say the belt seals affected the motor design due to their intuition from work experience. In this situation, the interviewer needs to have a good understanding on the content of the interaction, and explain to the engineers why a mark should be put in row *Glass* and Column *Belt Seals*, instead of row *Motor* and column *Belt Seals*. Although there is no mark between the *Belt Seals* and the *Motor*, the two will reach each other indirectly, which will show in the reachability matrix.

  Sometimes, the indirect interactions might not be found out during the interview. The interviewer needed to document the content of each interaction, and to go through them alone after the interview to sort out indirect relations. By doing so, the interviewer can also gain a good insight to the system, and find out the essence of the information flow.

- The engineers have different perspectives on the issues due to the difference of their work. In this case, the interviewer acted as a mediator. The interviewer should possess the knowledge of the system to some degree, and be able to discuss different viewpoints with different interviewees, or even go back and ask them again, until a common understanding is reached.

For the two reasons mentioned above, it is very important to interview individual engineers face to face for the content of the relations. However, since interviews are usually time-consuming and tedious, many researchers have proposed using survey sheets or collecting information from meetings. These proposed the methods have their advantages and disadvantages.

In some of the previous papers on DSM, the researchers used survey sheets to obtain the engineers' opinions. The DSM was constructed based on the majority of the votes or the higher level managers' views. The advantage of using survey sheets is time-efficient. However, many important details may be lost since the survey sheets don't give explanations on their choices. Also, bias due to work experience cannot be distinguished in a survey sheet.

Meetings open up the opportunities for discussion and understanding which is unavailable from the survey sheets. Meetings also reduced the amount of time for the data collector to get a consensus among engineers. However, it is inevitable that some of the engineers may feel uneasy to speak their mind due to peer pressure, or due to the influence of the rest of the group. Some important data may be lost this way.

Since the DSM is a tool to analyze the design project and to seek improvements, it is important that the data is accurate. The author feels that although talking to engineers in person is time consuming, the interviewer can usually gather accurate information and gain a very good insight to the system. However, when necessary, one may have to trade the speed of data collection with the quality of the data.

Step 2 and Step 3 are highly iterative. Deeper understanding of the system usually results in modification of the initial system elements. The system elements in this thesis research were modified many times during the interviews and documentation readings in order to represent the system accurately.

Step 4: Build a Matrix to Represent the Information Flow

A binary matrix can be built to represent the information flow between various system elements. One can build the matrix by hand. The DSM program in Appendix A will take the recorded relation and build an unpartitioned matrix.

Step 5: Partition the Matrix

The method of partitioning and literature has been mentioned in Section 5.1 and 5.2 already. The program in Appendix A can partition the DSMs mathematically.

Step 6: Optimize the system

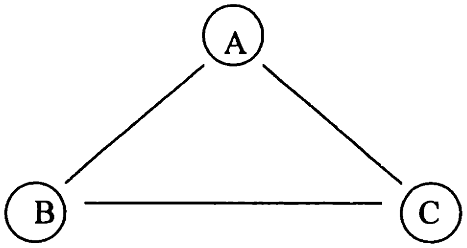
Partitioning provides only the mathematical understanding of the process. Further study can be done after partitioning, including assigning dependency values, tearing, de-coupling and add-coupling, and other non-binary DSM techniques. Section 5.1 and 5.2 have provided discussions on these methods.

Step 7: Give the matrix to the engineers and managers.

One of the goals of this thesis research work was to aid the design engineers and engineering managers to understand the design process better and approach the communication more systematically. Hence, the constructed DSM's are usually given to the engineers and manager to use. The usual response from the engineers and managers was that they were very impressed by the power of the tool. Furthermore, seeing the entire picture of the design process like never before makes them think over their current practice, and seek improvements.

5.4 Chapter Summary

Professor John Warfield has once proved that human beings can only deal with 7 things simultaneously without the outside aids [Warfield 1998]. These 7 things include 3 items and their 4 interactions (Figure 5-7). Engineering design projects involves much more than 7 relations. Hence, tools are needed in order to understand the design process and seek improvements. Design Structure Matrix is a simple yet powerful tool to study the design process and information reuse. The rest of the thesis will prove this point using empirical research evidences.



The 7 things to think about are:
A, B, C, A-B, A-C, B-C, and A-B-C.

Figure 5-7 The Seven Things

CHAPTER 6 STUDY VEHICLE DOOR DESIGN

6.1 Introduction to the Product

Before introducing the objectives of this case study, the author would like to explain the product studied first so the readers can understand the terminology used in the objectives statement. The product studied in this case is the Ford vehicle door. Looking from the outside, a door consists of the outer sheet metal panel, the glass, the seals, and the interior door panel that driver and passengers touch when they are inside the vehicle. The interior door panel is mounted on an inner sheet metal panel, which is welded together with the outer sheet metal panel (see Figure 6-1). At the front corner close to the windshield, a triangular shaped area, where the side-view mirror is mounted on the outer sheet metal panel, is called the Sail Panel (see Figure 6-2). At Ford, the sheet metal system team, who also designs the entire body frame, designs both the outer and the inner door panels.

Inside the space formed by the outer and inner sheet metal panels, various mechanisms and wires reside. The largest mechanism is the glass-moving mechanism that moves the glass up and down according to the driver or passengers' adjustment. Figure 6-1 shows the major components in a power window glass-moving mechanism. The regulator arms (X style in this figure) hold the window glass (see also Figure 6-3). A motor drives the regulator arm linkage and moves the glass up and down. One of the regulator arms links to the equalizer channel which moves within the below belt retainer to ensure the glass a leveled motion. The glass moves within a track (Glass Runs) along the A-pillar and B-pillar (see Figure 6-1).

Figure 6-4 shows the details of glass runs. At Ford, The moveable glass system team designs the glass-moving mechanism as well as the glass runs and the seals.

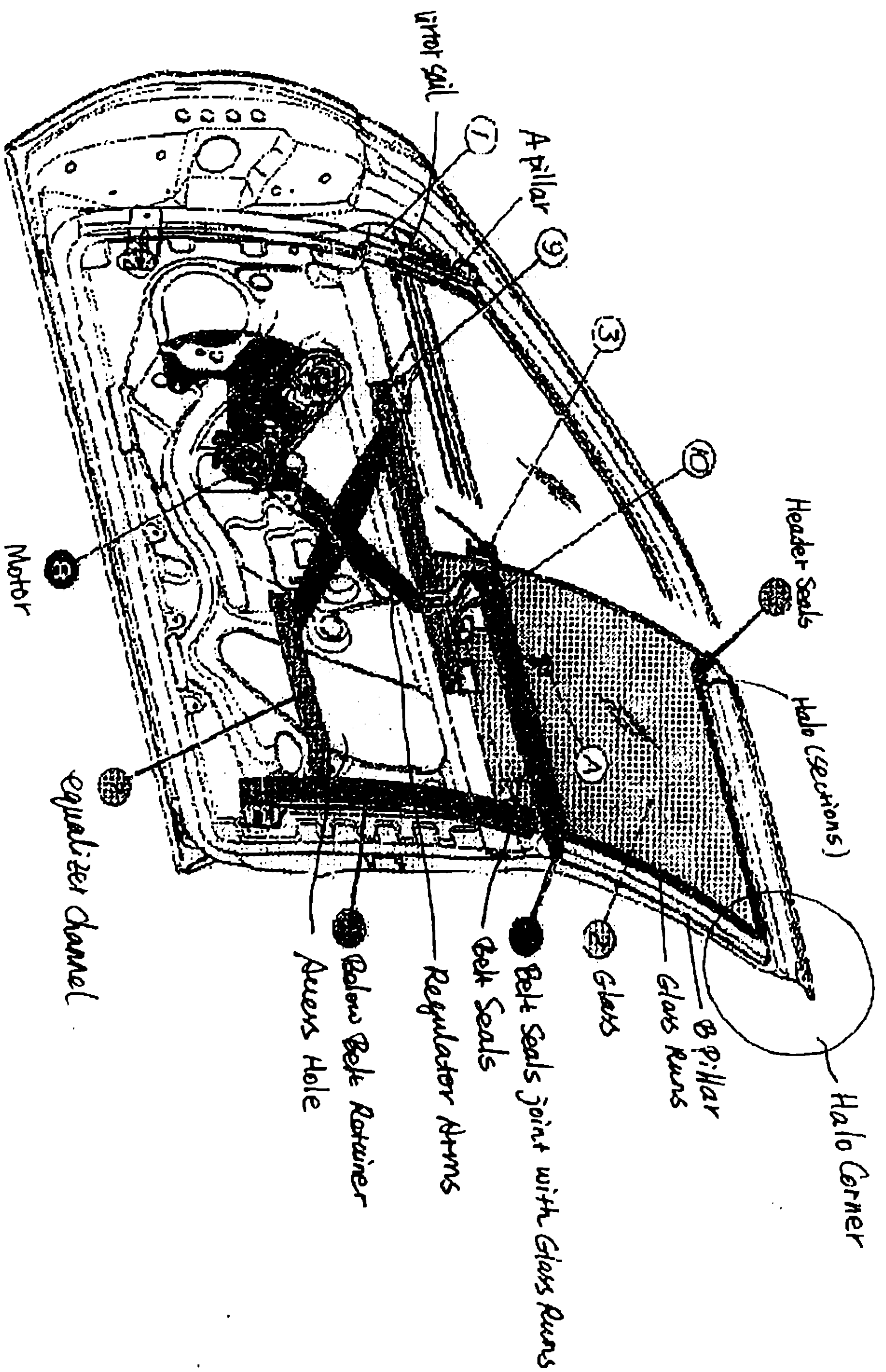


Figure 6-1 Major Component inside the Door

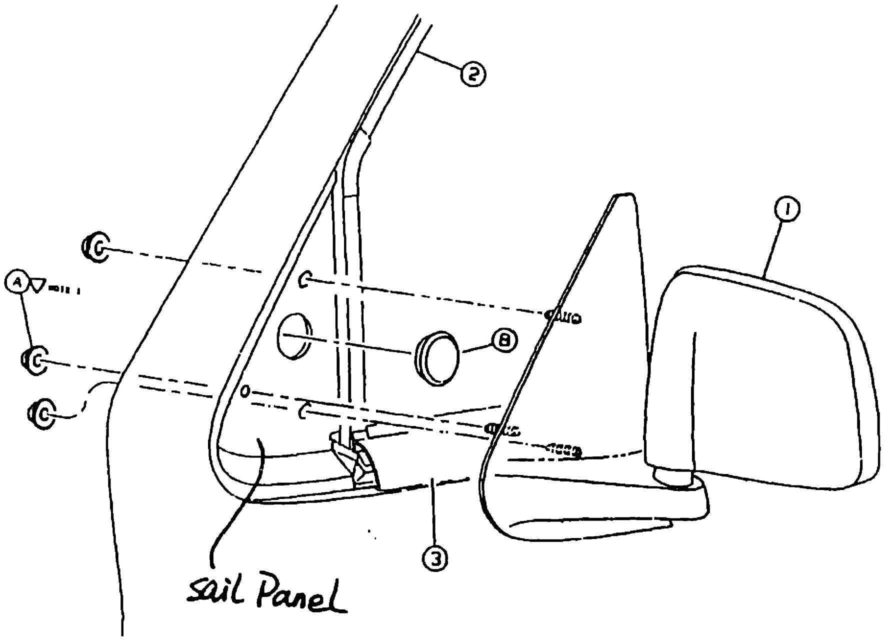


Figure 6-2: Sail Panel

Cross-Arm
Regulator

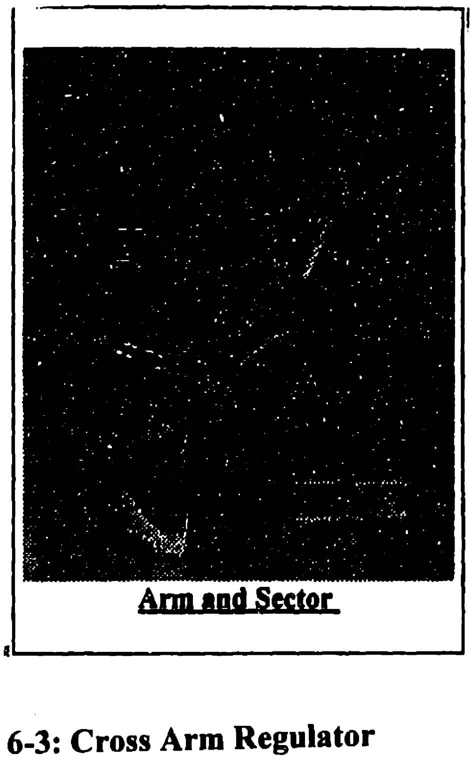
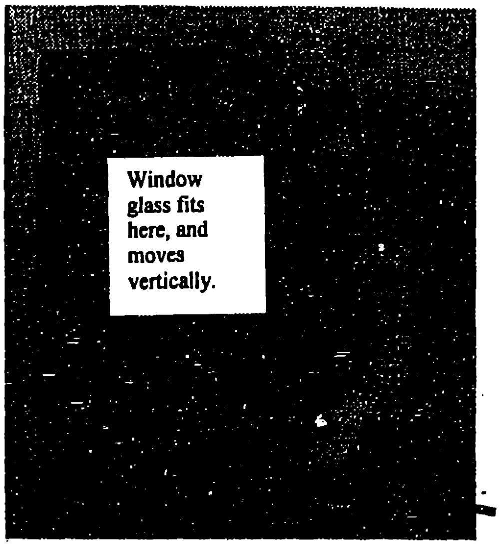
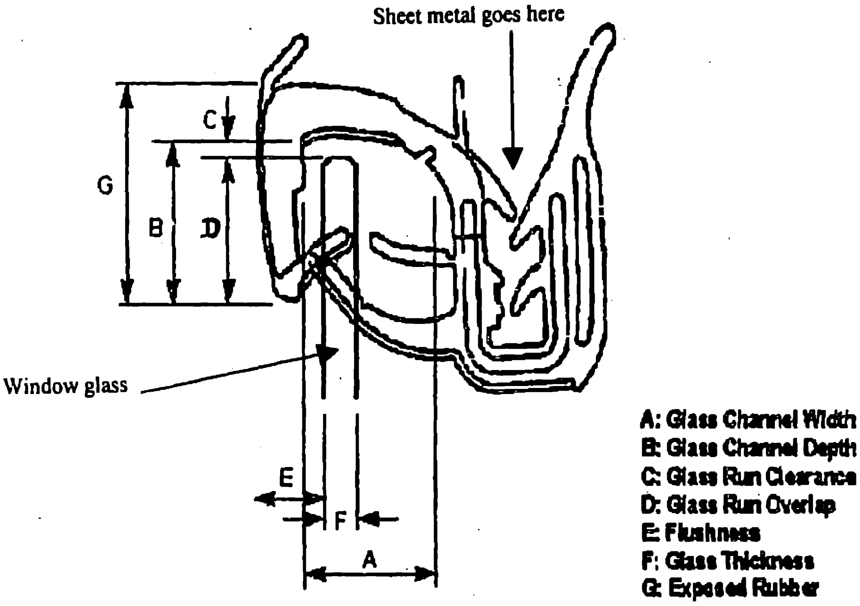


Figure 6-3: Cross Arm Regulator



This portion is inside the sheet metal panels.

Glass Runs



Cross section of Glass Run channels

Figure 6-4: Glass Runs

Besides the moveable glass system, the space between the inner and outer sheet metal panel also contains the speaker and various electronic wiring to operate the motor, door locks, speakers, etc. At Ford, the electrical system team designs the wiring inside the door.

6.2 Case Study Objectives

From the introduction in the last section, we can see different parts and subsystems of the door belong to different design teams and organizations in Ford. No one single team is assigned to design the door as a whole. Hence effort is needed to understand how to integrate the part and subsystem designs into a high quality door and improve the product development time.

This thesis research is part of the Ford Direct Engineering (DE)'s effort to apply system engineering thinking in the door design process. The author spent seven weeks in summer 1997 at Ford Advanced Vehicle Technology, Dearborn, Michigan, to study the door design process. Besides the overall research objectives listed in Chapter One, the following objectives are specific to this case study:

- Discover and document important relation specific to the door moveable glass system interfaces, which include the glass system, the electrical system, and the sheet metal system.
- Understand the door moveable glass system interface. Suggest a logical design sequence and identify the teamwork required to execute the design.

This study was based on the assumption of "parallel iteration" defined by Dr. Robert Smith and Dr. Steven Eppinger as "an extreme case where a number of development activities are underway at one time" [Smith (1997)]. Parallel iteration is a good assumption since Ford Motor Company has been designing doors for years. The tasks involved in door design and the task interrelationships are well known, and did not change during the course of this study.

6.3 Steps taken

Chapter Five in this thesis has described the general steps taken by this thesis research to apply the DSM method to various case studies. Here, the procedure specific to this case study is listed below, as well as the author's learning and observation made in each step.

1. Define the system and its scope;
2. Define the initial set of system elements;
3. Define the interactions between elements;
4. Re-define the system elements;
5. Categorize the contents of information flow;
6. Assigning dependency values to each relations;
7. Partition and re-group the tasks.

Step 1: Define the System and Its Scope

This research studied how the moveable glass system interfaces with other systems in the vehicle. The systems that interfaces with the moveable glass system mainly are the electrical system, and the sheet metal system.

Ford vehicle door designs have many variations. This research studied the following specific design configuration:

- The regulator was restricted to the cross-arm (scissors) style (see Figure 6-3).
- The motor sound quality was not considered.
- Frame doors only. Frame doors are those that have sheet metal frame around the glasses. Non-frame doors are like those doors on Mustangs that do not have sheet metal tracks around the glasses.
- Power window only.
- Front doors only.

Step 2: Define the Initial Set of System Elements

Ford Advanced Vehicle Technology Engineer Dr. Darrell Kleinke defined the initial set of system elements, based on his previous work experience and documentation. These initial system elements included the following:

- Shape of the Inner Panel at Regulator
- Belt Opening
- Pillars (sections)
- Equalizer Channel
- Halo Corners (upper)
- Halo (header section)
- Below Belt Retainer
- Wire Route
- Sharp Edges on the Sheet Metal
- Connector between the Motor and the harness
- Design of the Switch
- Mass of Wire
- Regulator Motor and Arms
- Belt Seals
- Glass Runs
- Access Hole Geometry
- Outer Panel
- Header Seals
- Sail Panel
- The Position of Wire Fasteners
- Power Supply
- Electrical Circuit Design including the Fuses, Resistance, Circuit Breaker
- Current Drawn to the Motor

However, during the process of understanding the interaction between the system elements (Step 3), this initial set of system elements were modified many times in order to represent the essence of the design issues effectively. This modification process is listed as Step 4 below. In fact, Step 3 and 4 are highly iterative and are carried out simultaneously.

Step 3: Define the Interactions between System Elements

The interactions between system elements in this case are mainly defined based on interviews with experienced product design engineers. Darrell Kleinke, an Advanced Vehicle Technology engineer, provided most of the data for the moveable glass system. Two other experienced electrical system design engineers also verified the data. The author also gathered information from engineering meetings. The rules about interviews discussed in Chapter 5 are learned from conducting these interviews.

Step 4: *Re-define the System Elements*

From the knowledge of the system, it then was discovered that some of the initially defined system elements were not detailed enough to represent the issues involved in the design process clearly. For example, the *Regulator (Motor and Arms)* listed in Step 2 was too broad an element. The issues related to regulator arms design and the regulator motor design are very different. Also, for the motor itself, the motor physical size, position, and orientation affect the physical size and location of other components inside the door panels, and hence are related to the packaging issue. The motor electrical features are related to the circuit design and wiring. Therefore, in order to accurately capture the design issues, the original *Regulator (Motor and Arms)* was divided into new system elements--*Regulator Arms*, *Motor Physical Features*, and *Motor Electrical Features*. Another example is the initially defined system element *Belt Seals*. Issues related to the belt seals varied very much. For instance, the show surface design is decided at the styling department. The seal lips and flange affect the seal drag force on the glass, and are related to the *Belt Opening*. The belt seals joint with the sail panel and the glass runs are appearance and wind noise concerns. Hence, the original *Belt Seals* was divided into *Belt Seals Lips and Flange, Belt Seals Show Surface, Belt Seals Joint with the Sail Panel, Belt Seals Joint with the Glass Runs.* Some other changes on the system elements were also done. These changes were made simultaneous with Step 3 as the author's understanding on the system developed. The new set of system elements is listed as follows. These system elements include not only physical parts and features, but also engineering parameters. They are a collection of items that engineers have to consider in order to deliver a quality product.

- Outer Panel Shape
- Halo (Header Cross Section)
- Belt Opening
- Pillars
- Sail Panel
- Halo Corners (Upper)

- Inner Panel Shape at Regulator
- Inner Panel Material at Regulator
- Belt Seals Joint with Sail Panel
- Glass Runs
- Below Belt Retainer
- Header Seals
- Belt Seals Joint with Glass Runs
- Equalizer Channel
- Motor Electrical Features
- Connector between the Motor and Harness
- Current Draw to the Motor
- Wire Size
- Wire Route
- Access Hole Geometry
- Sharp Edges on the Sheet Metal
- Belt Seals Show Surface
- Glass
- The Joint of Glass Runs and Header Seals
- Belt Seals Lips and Flange
- Regulator Arms
- Motor Physical Features
- Power Supply
- Electrical Circuit Design (fuses, resistance, circuit breaker)
- Switch Current Capacity
- Wire Length
- The Position of Wire Fasteners

Step 5: Categorize the Content of Information Flow

Previous research has been done to categorize the information exchange in a design process. The literature review in Chapter 5 has mentioned some of the existing work. In order to understand the nature of the vehicle door design process, the author and Ford engineers sought means to categorize the information content. Although Baya, Gevins, Baudin, Mabogunje, Toye, and Leifer categorized the design information into Descriptor, Subject Class, Criticality, and Level of Detail [Baya (1992)], and Dileep V. Khadilkar and Larry A. Stauffer used these categories to study an actual design process [Khadilkar(1996)], the above four categories do not apply to this case for the following reasons:

- This project only concerned the interface of the moveable glass system, the electrical system, and the sheet metal system, so most of the information falls in the "Relation" category in the "Descriptor".
- By the same token, most the information fell in the "Feature" and the "Connection" category in the "Subject Class."
- Since our project concerned only the existing design process of the door, the "Level of Detail" did not apply to our project at all.

Hence, the above four categories could not provide a useful insight of the information flow in the door design process.

Pimmler and Eppinger categorized the design information based on the nature of the information in one of their paper about the design of an automotive climate control system [Pimmler (1994)]. This approach seems to provide more useful insights to the door design information. Hence the author and Ford engineers defined the following information categories:

Spatial: the spatial-type interaction identifies needs for adjacency or orientation between two elements.

Function: the function-type interaction includes information related to the power and motion of the glass movement, the need for material and support under dynamic loads, and aerodynamic consideration such as wind noise and door rattling.

Appearance: the appearance-type of interaction includes information related to the appearance of the car, such as the smoothness of the joint of two components.

Step 6: Assigning Dependency Values to the Relations

Instead of using binary DSM, this research work attempted to assign dependency values to indicate the importance of the information flow between two elements. The following letters were used:

A: high dependency. The information in the column is essential to start the work of the element in the row.

B: medium dependency. The information in the column is needed in order to make a decision on the element in the row..

C: low dependency. The information in the column provides a good reference in making a decision to the element in the row. However, the decision on the element in the row can be made without the information from the element in the column.

O: optional path. The information flow path is not necessary, but it indicates a potential for add-coupling and improving the quality of design. This symbol may be used in conjunction with the above three symbols.

The dependency values are obtained from interviews with experienced product design engineers. The same rules mentioned in Step 3 apply. The interviewer needs to have a consistent rule of the dependency value assignment.

Step 7: Partitioning and Re-grouping the Tasks

After step 5 and 6, each entry of the DSM contains two dimensions of the information being exchanged. One dimension is the impact of the information exchange on the design process. The other is the category of the information. Hence, this thesis work took two different approaches to partition the DSM in order to show the two different dimensions of the data.

The first approach is to show the impact of the information exchange on the design process. From the definitions provided in Step 6, one can see that *A* and *B* type of information exchange has large impact on the design process and design decision making, while *C* and *O* types of information has relatively small impact. Hence, when the matrix is partitioned, *A* and *B* are treated as one and *C*'s and *O*'s as zero in a binary DSM. If an entry in the DSM contains *A* or *B*, this entry is treated as a *1*. The DSM partitioning program takes the data and partition the matrix based on the Warfield's reachability matrix method (Warfield 1973).

The second approach attempts to show how different types of information affect the decision making in the design process. The dependency values are treated as described in the last paragraph. Furthermore, individual DSMs are made for each information categories. The DSMs for individual categories are partitioned independently.

After partitioning the matrix using the mathematical method, further re-grouping of the tasks can be done by manually moving the rows and columns within an iteration loop. This step is based on both observation, and the knowledge and the experience of the system. More details on this step will be in the next section.

The last step is to find a name for each iteration group as proposed team divisions. Sometimes, in the process of doing so, new team divisions or new element divisions can be found based on the understanding of the system. Thus, the matrix can be further optimized.

6.4 Results and Discussion

The information flow based on the current sub-system team division is shown in Figure 6-5. Three different patterns indicate the three major design teams involved --the moveable glass system team, the electrical system team, and the sheet metal system team. As one may observe, many important information exchanges are outside of a system team's block. Quite a few of them are above diagonal. To deal with these design iterations, what the author observed in summer 1997 was that the Ford engineers held concurrent engineering team meetings that involve a large number of people from various organizations. These meetings did serve as a communication opportunity. However, due to the large scale, these meetings could not effectively and efficiently resolve design issues. Furthermore, many time, the person who could provide answers to the question raised was not present at the meeting because he/she has more important work to do than attending an ineffective meeting. Professor Whitney and Professor Eppinger called this situation "Concurrent Engineering in the Large," and argue that aids from other product development tools are needed in addition to concurrent engineering meetings [Eppinger (1994)]. DSM seems to be a very powerful tool to deal with concurrent engineering in the large.

The result of partitioning the DSM in full scale (approach one in Step 7 in the last section) is shown in Figure 6-6. The following can be observed from the DSM:

1. Four teams can be formed based on the partitioning results—the Belt and Above Belt Frame Design Team, the Glass and Its Track Design Team, the Power and Motion Team, and the Electrical Packaging Team. Team 1 includes not only the iteration loop but also the tasks that can be done independently before and after the iteration loop. Two engineers can be assigned to work on the *Belt Seals Joint with Glass Runs* and *the Sharp Edges on the Sheet Metal* independently. The *Outer Panel Shape* and the *Belt Seals Show Surface Design* belong to the Styling Department, so no teams are formed for them. However, The Styling Department's design decides the interior system engineers design, so these two elements are included in the matrix. The *Power Supply* is the battery design. Currently, all batteries are standard 12 volts, so there isn't much change on it.

2. Compared to the patterns in the second row and second column in Figure 6-5 and 6-6, the proposed team has broken down the original organization structure. Experts from various teams are brought together to form teams to deal with design tasks. The new teams are based on the issues in the design rather than the traditional manufacturing processes as that of the current teams. Thus, in stead of trying to concurrent all the issues in the design on one large meeting that involves everyone, the new team meetings can be concentrated on specific issues in the design, and hence should be more effective and efficient.

3. The above observations also tell us that all issues in the door design that has potential for large design iteration and time delay involve more than a single engineering team. Knowledge needed to resolve issues that are critical to the design progression resides in different organizations. Concurrent engineering meetings based on the partitioning results

from DSM is one way to deal with the situation. Direct Engineering program, when completed, should also provide tools to access knowledge resides in other organizations, and hence improve the design efficiency at system level. In addition, the DSMs obtained from this case study can serve as browsers to the knowledge. When an engineer works on one item in the DSM, he/she can find out in the DSMs about who to interact with and which organization that other person is in. Furthermore, when a complete database (Chapter 7 has more details) is built to record all the details of the interaction in a DSM, a DSM can act as a browser to the knowledge database for fast and accurate access.

4. Compared to the current teams, the proposed teams are of smaller sizes and hence involve fewer people. The communication between team members thus could be more efficient and manageable. Furthermore, since each team represents a design iteration, the smaller the team is, the smaller the iteration loop is. Hence, the proposed teams represent smaller design iterations, which means the design can be completed faster.

5. The proposed design teams capture all of the information transfer that is critical to the design iteration within a team (Figure 6-6). The current team divisions (Figure 6-5) leave a lot of the important information outside of the teams. From the system engineering prospective, teams should be divided so that the most important information exchange occur within teams, because communications between members of the same team are more effective than that between members of different organizations. Hence, the proposed design represents a better system engineering approach.

6. The partitioned DSM (Figure 6-6) not only shows a better way of dividing teams, but also shows which team needs to meet at what stage of the design process. For example, team 2 does not need to start until team 1 finished its design because team 2 has to wait for the inputs from team 1. Thus, there isn't a need for weekly large-scale concurrent engineering meetings any more. Based on the progress of the product design, members of different teams can meet to deal with their specific issues. When a team's work has not

started or already is finished, the members in that team can be freed up for other tasks. Thus engineering resources are used efficiently.

7. The elements in Team 3 in Figure 6-6 are the results of mathematical partitioning. However, Team 3 can be further divided into two smaller sub-teams based on the understanding of the system. Hence, iteration loops from mathematical partitioning can be further optimized based on engineering knowledge and understanding of the system.

The details of the information content is further studied by constructing three DSMs that are based on the three information categories—the spatial relations, the function relations, and the appearance relations, as described in approach two in step 7 in the last section. The DSMs for each information categories are in Figure 6-7, 6-8, and 6-9. Each individual matrix presents a different aspect of the system. The following observations can be made:

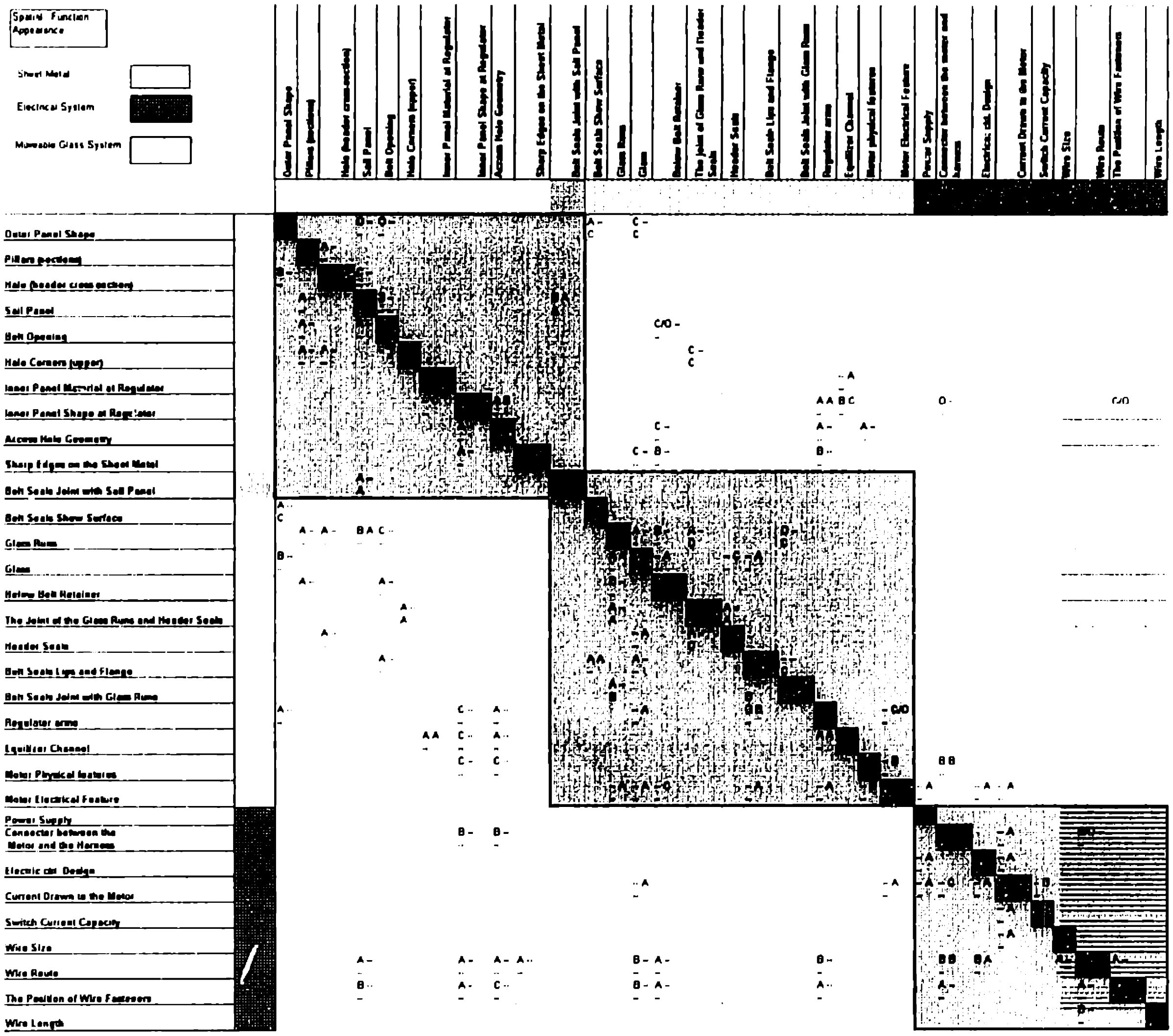
1. Compare the full-scale matrix (Figure 6-6) with the DSM based on the spatial relation (Figure 6-7), the elements from the beginning to the *Access Hole Geometry* have the same order and iteration loop grouping in both matrices. Hence, up to the *Access Hole Geometry*, the spatial relation dominates the system design, although other types of information also flow in the system.

2. Compare the full-scale matrix (Figure 6-6) with the system matrix based on the function relation (Figure 6-8), the system elements from the *Connector between the Motor and Harness*, and the *Switch Current Capacity* follow the same order and iteration loop division. So the second part of Team 3 follows the functional relation of the system although other types of information also flows in this part.

3. In the DSM based on the appearance relation, the marks are symmetrical about the matrix diagonal, except the relation between the *Glass* and the *Outer Panel Design*. However, for a joint of two components, the component design has higher priority than the joint

design. For example, the information flow from *Glass Runs* to *the Joint of the Glass Runs and Header Seals* is of high dependency, while the other direction is of low dependency. It seems like the functional requirements have higher priority than the appearance requirements in this case. Furthermore, the appearance relation appears mainly as an issue of the joint of two parts such as the joint between the glass runs and the belt seals, and does not play an important role in the door glass system interface.

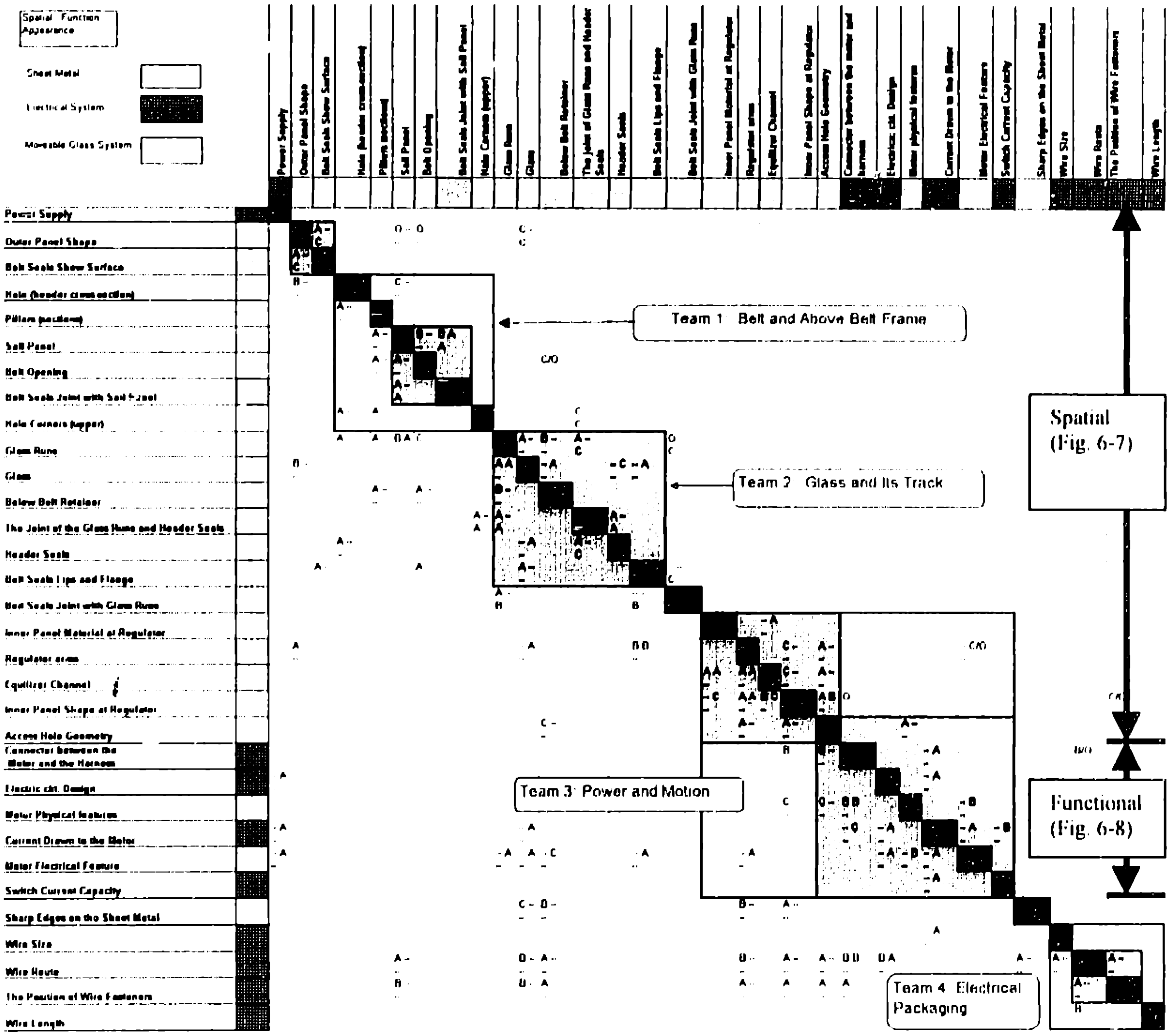
4. From the above three observations, we can conclude that the door design engineers first deals with the packaging issue (spatial relation), then the functional requirements. This order is different from the usual design methodology that requires the design to meet functional requirements first, then others. The difference is caused by the fact that the door design does not start from scratch. Most of the components are standard parts Ford buys from suppliers or makes in house. Between different vehicles, the shape of the door varies, but not the function of the doors. Hence, the engineers' first concern is how to package the available standard parts into the provided space, instead of how to design a mechanism to move the glass. The author believes this packaging-before-functional-requirements situation also applies to other designs where a lot of design reuse exists.



Note: *Belt joint with sail panel* belongs to both sheet metal team and moveable glass team.

A: high dependency. B: medium dependency. C: low dependency. O: optional path

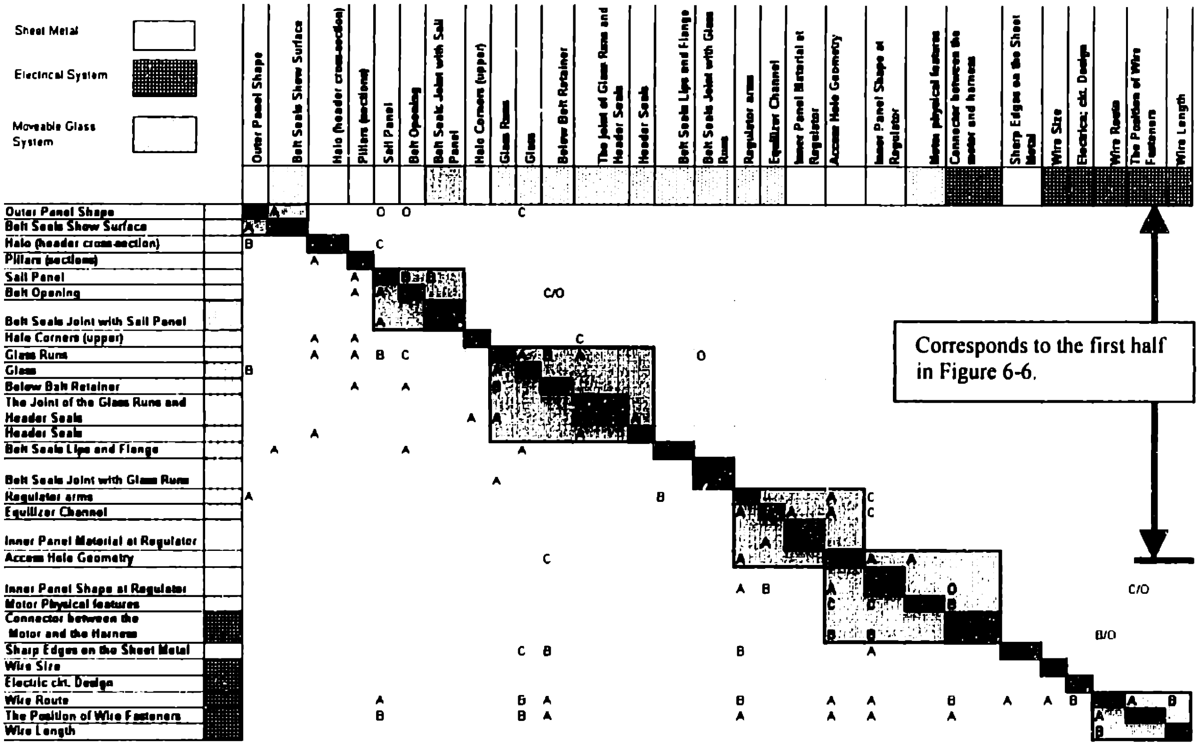
Figure 6-5 Current Vehicle Door Design Process



Note: *Belt joint with sail panel* belongs to both sheet metal team and moveable glass team.

A: high dependency. B: medium dependency. C: low dependency. O: optional path

Figure 6-6: Proposed Door Design Process after Partitioning the DSM

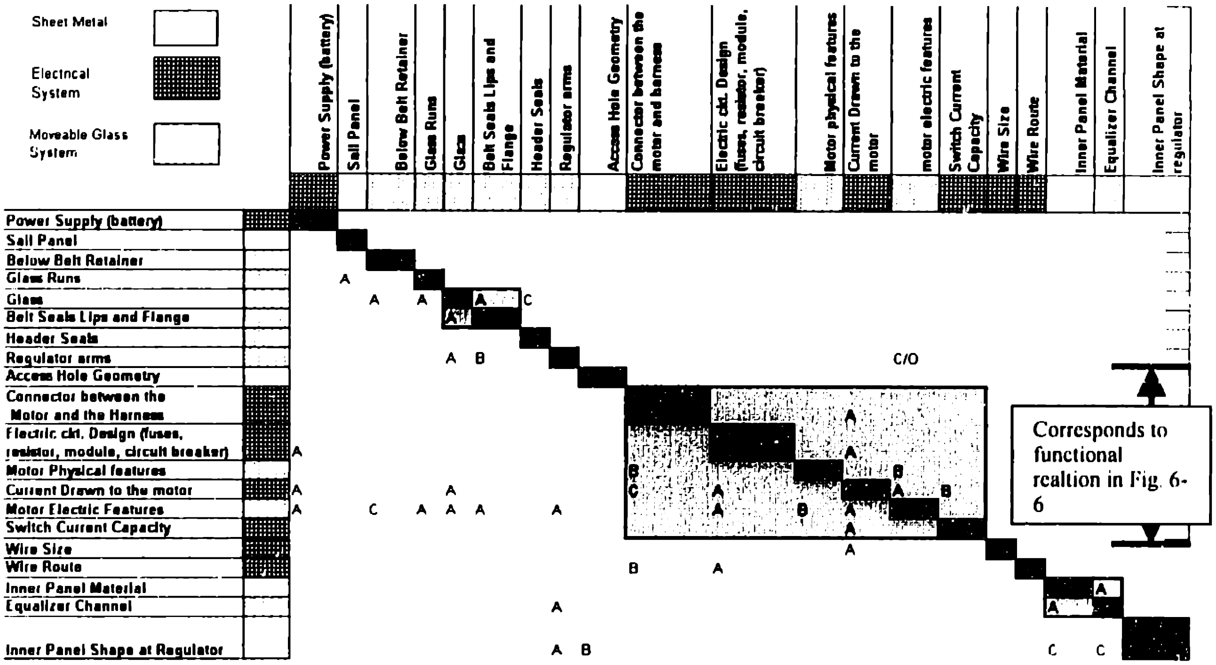


Corresponds to the first half in Figure 6-6.

Note: *Belt joint with sail panel* belongs to both sheet metal team and moveable glass team.

A: high dependency. B: medium dependency. C: low dependency. O: optional path

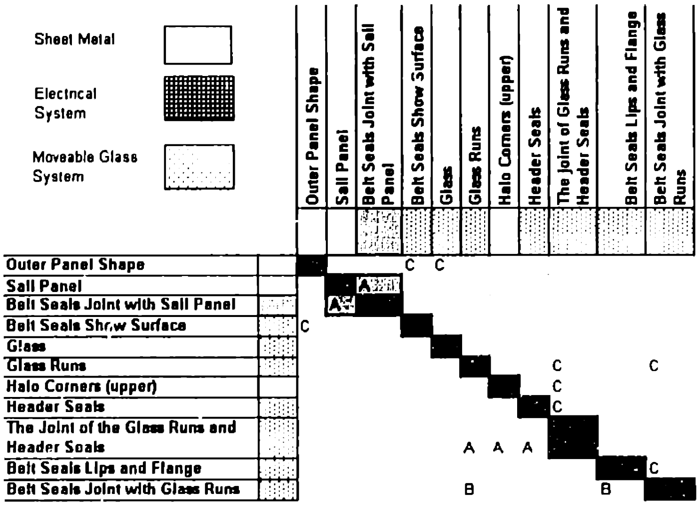
Figure 6-7 Door Design DSM Based on Spatial Relation



Note: *Belt joint with sail panel* belongs to both sheet metal team and moveable glass team.

A: high dependency. B: medium dependency. C: low dependency. O: optional path

Figure 6-8 Door Design DSM Based on Functional Relation



Note: *Belt joint with sail panel* belongs to both sheet metal team and moveable glass team.

A: high dependency. B: medium dependency. C: low dependency. O: optional path

Figure 6-9 Door Design DSM Based on Appearance Relation

6.5 Chapter Summary

Vehicle door design at Ford has traditionally been a collaborative effort of many design teams. This case study used DSM to discover and document how moveable glass system team interacts with the sheet metal team and the electrical system team in order to complete the door design. The information transferred during the door design process is recorded by the DSMs. The recorded information is also divided into spatial, functional, and appearance three categories.

The following observations are made from the case study:

1. The results of partitioning the DSMs suggest forming design teams that mix the experts from different organizations to deal with iteration causing design issues, so that to avoid the large scaled but ineffective concurrent engineering meetings that invite everyone on the vehicle programs.

2. The partitioned DSMs suggest design teams that are smaller than the existing teams. Hence the teamwork is more efficient and the speed of the design process can be improved.

3. The partitioned DSMs provide a map of the information flow in the design process, and only the iterations that are inevitable. Following the order suggested by the DSM to complete the tasks helps the engineers avoid unnecessary design iterations and improve the speed and the quality of the design.

4. The suggested order of completing tasks can also help the teams to better schedule their resources and plan out their work.

5. In most cases, design knowledge needed to resolve design issues resides in more than one team. The DSMs obtained can serve as a browser for the engineers to find out what need

to be done and whom they need to talk to.

6. Observations from the information categories show that door design engineers first consider packaging issues then functional issues. This is due to Ford reusing and outsourcing most of the components inside the door. In addition, appearance is not of important concern for the moveable glass system interface.

The DSMs obtained in this case study are posted on the Ford Intranet as a reference to engineers' work.

CHAPTER 7 STUDY THROTTLE BODY SYSTEM INTERFACE USING DSM

7.1 Introduction to the Product

The product studied in this case is the Ford mechanical throttle body. A throttle body is essentially a butterfly valve that controls the amount of airflow into the engine (see Figure 7-1). Upstream of the throttle body is the air intake system. Ambient air enters the air intake system, goes through the air filter, and approaches the throttle body plate. Downstream of the throttle body is the engine intake manifold, which the bottom of the throttle body is bolted onto.

The throttle body is comprised of seven major parts, which are the housing, the plate, the shaft, throttle return springs, cam and lever, throttle position sensor, and bearings. The housing is the largest part. It has a cylindrical bore, which is the main airflow passage. The bottom of the housing connects to the intake manifold using bolts. A brass plate—the throttle plate—is inside of the bore. The plate is rigidly screwed on the shaft (see the lower diagram in Figure 7-1), which is supported by two ball bearings connect to the housing. One end of the shaft connects to the cam and lever assembly. The other end of the shaft connects to the throttle position sensor (TPS), which is a potentiometer that varies its output voltage based on the angle that the shaft turns. When the throttle body is assembled in the vehicle, a cable (the accelerator cable) connects the gas pedal under the driver's foot to the cam on the throttle body. When the driver steps on the gas pedal, the cable pulls the cam and hence opens the plate according to the amount the driver steps on the pedal. The TPS measures the plate opening, and tells the fuel injector to provide the appropriate amount of fuel.

For safety reasons, the United States Federal Law requires two independent mechanical energy sources to close the throttle plate when the driver's foot is not on the pedal. Ford throttle body design uses two throttle return springs as the two energy sources. Figure 7-1 shows a torsion return spring between the cam and the housing. In this particular design configuration, the other spring is an extension spring that connects to the throttle lever and the engine bracket. As an alternative, some throttle design uses a dual wound torsion spring at the same location as the single wound spring in the picture. Also, some designs use a compression spring on the accelerator cable as another energy source. Later on in this chapter, the design of the dual wound spring and the single wound plus an extension spring will be compared using DSM.

Throttle body cam geometry determines the amount of the airflow into the engine in response to the driver's foot rotation. Hence varying the cam geometry can create different customer perception of the vehicle handling and performance feeling for the same engine. In some throttle body designs, a four-bar linkage is used instead of a cam. Alternatively, a single ball stud can also be used (single lever) linkage, which create a much less sophisticated air intake versus driver's foot rotation curve.

The cam or other linkages used are mounted on the lever. The lever also incorporates the stops for the plate wide-open and idle position, the speed control pin, the spring connections, etc.

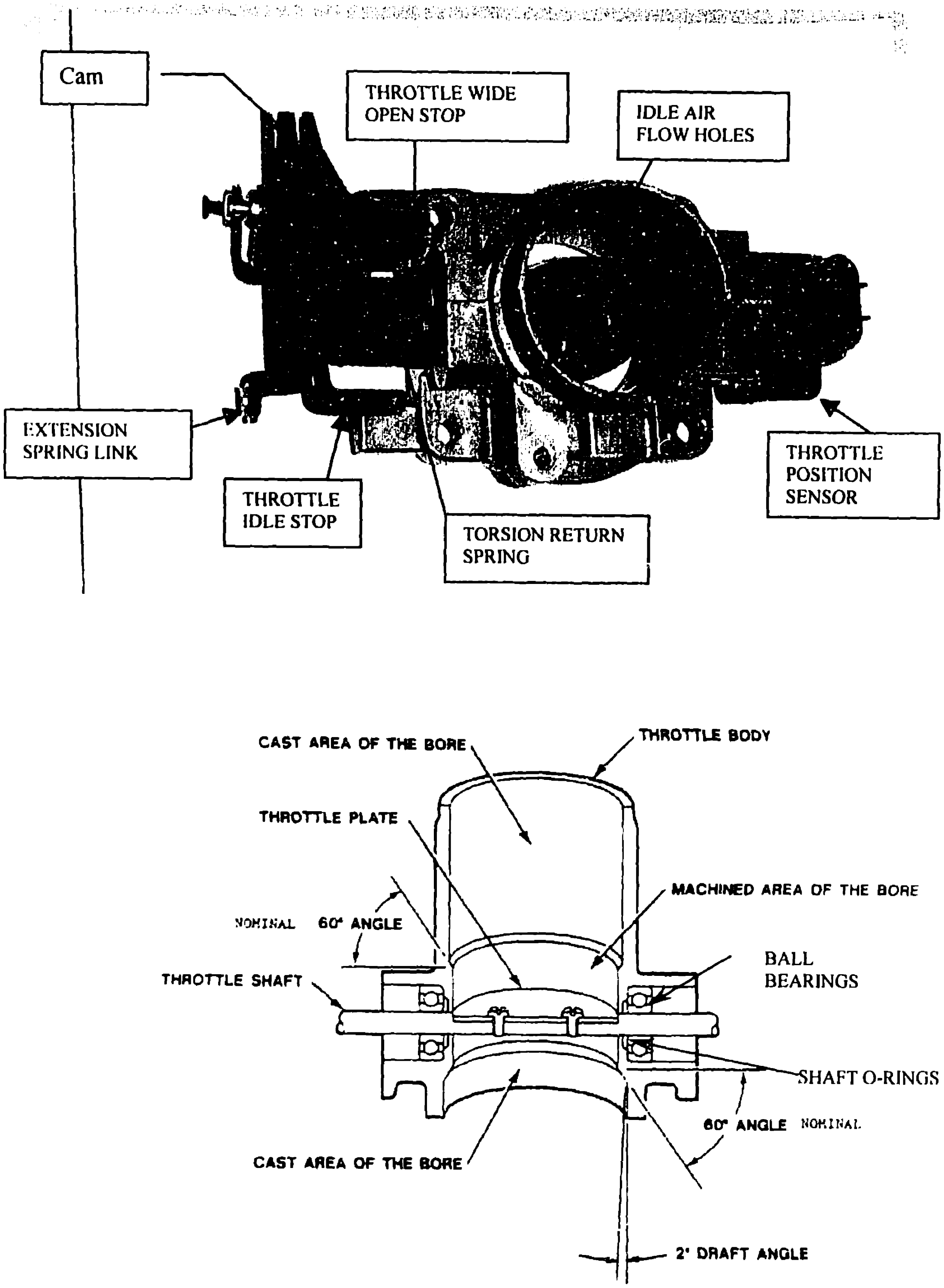


Figure 7-1 Throttle Body

7.2 A System Engineering View of the Throttle Body

From the introduction to the throttle body, we can see that the parts in the throttle body interact with one another in order to achieve the required function and performance. Furthermore, throttle body is a part of a larger system—the vehicle. As a unit, the throttle body system interacts with many other systems in order for the car to run normally. Jared Judson, a MIT Leaders for Manufacturing program graduate, has demonstrated the interaction and integration of throttle body system with many other systems in the car [Judson (1998)]. In Figure 7-2, the author of this thesis also drew a diagram of the system engineering view specific to the throttle body. Four systems directly interact with the throttle body, including the accelerator control system, the air induction system, the engine power train system, and the speed control system. Since the throttle body is of a much smaller scale than the above four systems, it is called throttle body subsystem in this diagram. Inside the circle that represents the throttle body subsystem, the throttle body components interact with one another too. In fact, inside the four systems mentioned above, the components in those systems also interact just like those in the throttle body. However, since this thesis work is concentrated on how throttle body interacts with the rest of the car, we are not concerned of how the components in other systems interact with themselves.

The diagram in Figure 7-2 shows that throttle body design engineers and managers should deal with two kinds of relations. The first one is the assembly level relation, which includes the interaction between all throttle body parts. The second one is the system level interaction, which includes the interaction between throttle body and the other four systems in the figure. The Ford Direct Engineering (see Chapter 4 of this thesis) has been experimenting to create a semi-automated design environment for the throttle body design, as well as a few other systems that are of the same scale. Hence, it is worth studying what information the DE programs should include at assembly level, as well as how to understand the system level interaction so that individual system design can be integrated into the whole vehicle design and the part centered design mentioned in chapter two can be avoided.

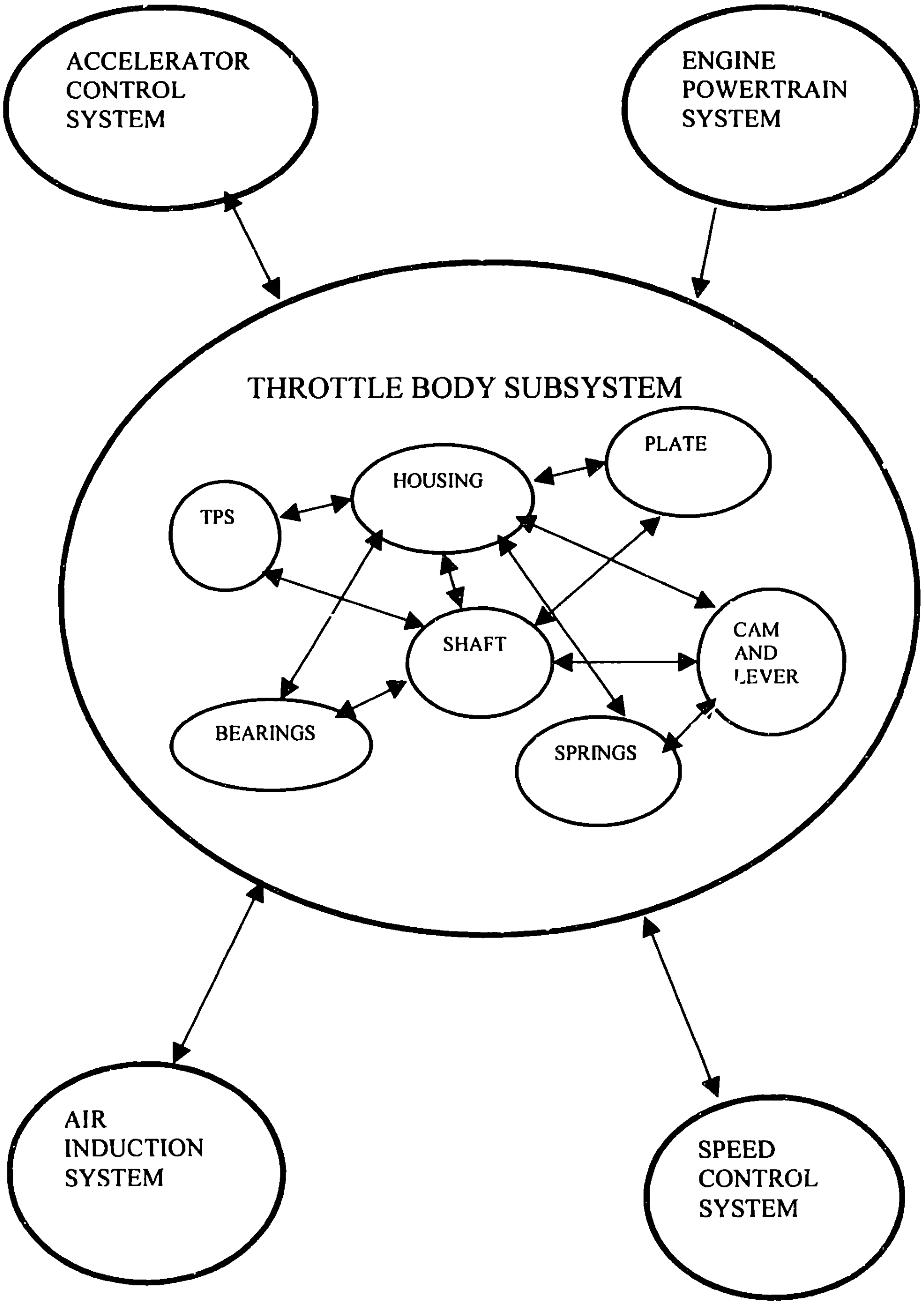


Figure 7-2 A System Engineering View of the Throttle Body

This chapter describes the research done to understand the throttle body system interface. The next chapter describes the study on the throttle body assembly level interactions. Both research work was conducted at Visteon Rawsonville plant and Ford Dearborn research centers. The research took place during January and May to August 1998.

7.3 Case Study Objectives

Besides the general objectives mentioned in Chapter 1 of this thesis, the following objectives are specific to studying the throttle body system interface:

- Discover and document important relations specific to the throttle body system interfaces, including the air intake system, engine power train system, speed control system, and the accelerator control system;
- Use DSM to integrate the throttle body system with the above four systems, and find a means of combining Direct Engineering application islands.
- Show knowledge and rules that apply to every TB or every version of a TB vs. variant knowledge;
- Show information flow and coupling in the throttle body system interfaces.
- Study whether Direct Engineering software can automate the entire design process or human interaction is necessary.
- Show knowledge ownership.
- Illustrate DSM to Ford engineers;

7.4 Steps Taken

This research took the following steps. Each step will be discussed in this section.

1. Define the system and its scope;
2. Define the elements involved in the system;
3. Define the interactions between system elements, and build an information database to record the interactions;
4. Choose a specific design configuration;
5. Partition the DSM;

6. Compare the DSM for various design configurations;
7. Study the Ford FPDS requirements.

Step1: Define the System and Its Scope

This research takes a system approach and mainly studies the interfaces of the throttle body with the air induction system, the engine power train system, the accelerator control system, and the speed control system. Individual component design is not of major concern here, and will be discussed in the next chapter.

Throttle body design has many variations. The design variations studied in this research include:

1. throttle linkage:
    - Cam
        - Over-molded on metal lever
        - composite cam and lever over-molded on a disk
        - composite cam and lever over-molded on the shaft
    - Single lever
    - Four-bar linkage
2. Throttle return springs
    - dual wounds
    - use extension springs
    - use compression springs (not compatible with cam)
3. Bore
    - Aluminum straight bore
    - plastic straight bore
    - plastic progressive bore
4. Use or not use idle air control valve

This study was based on the assumption of "parallel iteration" defined by Dr. Robert Smith and Steven Eppinger as "as extreme case where a number of development activities are underway at one time" [Smith (1997)]. Parallel iteration is a good assumption since Ford has been designing throttle body, air intake system, engine power train system, speed control system, and accelerator control system for many years. The tasks involved are well known, and did not change in the course of study.

Step 2: Define the Elements Involved in the System

At the beginning of this research, the author chose the initial set of system elements based on the Ford design documents on throttle body. These design documents include the Ford System Design Guides and the System Design Specifications for throttle body system, accelerator control system, speed control system, engine power train system, and the air induction system. The System Design Guides provide detailed information and rules on the design of a particular system, as well as all the components in that system. The System Design Specifications provide a higher level of requirements on a particular system, including federal regulations, clearance requirements with other systems, etc.

In order to define the system elements, the author highlighted the parts in the above documents that concern how throttle body system interacts with other systems. Then a set of elements, whose interaction reflected the system level interaction described by the design documents, was defined. Although the author consulted a few design engineers in order to understand the content of the design documents, the initial set of the system elements were mainly from reading the design documents. In the next step, when the author expanded the source of the information, the initial set of the system elements was modified so as to capture the system level interaction completely. In fact, step 2 and step 3 are highly iterative. Chapter 6 of this thesis also described the same situation for studying the door design process.

Due to the concern of the length of the chapter, the author will not list out all the system elements as Chapter 6 did. Interested readers can find the initial set of system elements in

Figure 7-4, and the final set in Figure 7-5, 6, and 7 later in this chapter. Nevertheless, the same observations on the nature of the system elements made in Chapter 6 apply here too. The initial and the final set of system elements include not only physical parts and features, but also engineering parameters. They are a collection of items that engineers have to consider in order to deliver a quality product.

Step 3: Define the Interactions between System Elements, and Build an Information Database to Record the Interactions

After the initial set of systems elements were defined, the first draft of the throttle body systems level DSM was built. The system interactions in the first draft of the DSM were from the design documents as described in the last step. In order to make the DSM reflect the actual design process, the author also took information from the following two sources:

1. Ford Design Engineers from each system. The engineers who provided major contribution to the data collection include:

- Denis Madson, who is an experienced throttle body (TB) design engineer, especially at the TB linkage design.
- Jim Conrad, who is a Ford Advanced Vehicle Technology engineer in accelerator controls system. He is also the author of accelerator control design guide and system design specification.
- Jhun Lin, who is an experienced air induction system engineer.
- Jim Rauch, who is an experienced TB design engineer.
- Kevin Deacon, who is an experienced TB design engineer, as well as a Ford Direct Engineering engineer.
- Dave Wise, who writes the program for the TB Direct Engineering application.

The interviews took three different forms. The first was to show the DSM to the engineers, and explain to them how it works. Then the engineers went through each column (or row) in the DSM with the author and corrected the entries. The author documented the content of each entry. This form of interview was very time consuming. Besides the reason explained in Chapter 5, the following also applies.

- It was obvious that even experienced engineers like those I interviewed had not systematically thought of all aspects of the design. The engineers responded to these interviews very positively, because these interviews made them think in the system point of view, and they also learned more through the discussions.

- Due to the amount of the information and details that are involved, it took average one hour to complete three to four columns in the DSM. The entire DSM contains from 37 to 40 columns/rows. Hence, it took many hours to complete the DSM with one engineer. The engineers had limited hours available each day, so the result was many days of interviews. For example, it took Denis Madson 7 days and average 1.5 hours per day to complete the matrix. And those 7 days was spread in two weeks due to Denis Madson's schedule. Despite all above, the author feels that this work is worthwhile because the data collected is accurate and hence reflects the real engineering design process.

2. Engineering meetings. By attending the regular engineering meetings, the author was able to identify issues from the meeting discussions, as well as collect information from open-ended talks and asking questions now and then. This form of data collection was a continuous effort over the entire period of rescarch.

During the process of collecting the knowledge on system design, a database was built to record the interactions between system elements using Microsoft EXCEL. A small portion of the database is shown in Figure 7-3. The system element that provides information is listed in the "From" column. The system element that receives information is listed in the "To" column. The "Content" column records the reason and the content of the information exchange. Due to proprietary reason, the details of the information content are omitted here. The "Source" column records the person or document that said the information exchange was needed. From the fifth column until the end, different design configurations are listed. When the interaction between a pair of system elements apply to the specific design choice, a *1* is put in the corresponding column. Using these columns, the author was able to sort out in the system element interactions specific to a design configuration and make DSMs for various throttle body designs.

From	To	Content	Source	Single lever	Overmolded cam	Composite cam	Cam + shaft	4-bar linkage	use torsional return spring	use extension returning spring
[redacted]	lever idle stop	Omitted due to propriet-ary reason	THROTTLE BODY DESIGN GUIDE-9E926-Air Intake Charge Throttle ASSEMBLY DESIGN CONSIDERATIONS E9	1	1	1	1	1	1	1
	lever idle stop		THROTTLE BODY DESIGN GUIDE-9E926-Air Intake Charge Throttle ASSEMBLY DESIGN CONSIDERATIONS E8	1	1	1	1	1	1	1
[redacted]	[redacted]		THROTTLE BODY DESIGN GUIDE-9E926-Air Intake Charge Throttle ASSEMBLY DESIGN CONSIDERATIONS E9	1	1	1	1	1	1	1
lever WOT stop	[redacted]		THROTTLE BODY DESIGN GUIDE-9E926-Air Intake Charge Throttle ASSEMBLY DESIGN CONSIDERATIONS E9	1	1	1	1	1	1	1
[redacted]	lever idle stop		THROTTLE BODY DESIGN GUIDE-9E926-Air Intake Charge Throttle ASSEMBLY DESIGN CONSIDERATIONS E9	1	1	1	1	1	1	1
[redacted]	[redacted]		THROTTLE BODY DESIGN GUIDE-9E926-Air Intake Charge Throttle ASSEMBLY DESIGN CONSIDERATIONS E8	1	1	1	1	1	1	1

Figure 7-3 The Database that Records Interactions between System Elements

Step 4: Choose a Specific Design Configuration

As mentioned in step 3, the information database shown in Figure 7-3 records what system interaction is specific to which system design configuration. Using the "sort" function in the Microsoft EXCEL, the author was able to get the list of system element interactions for a particular design configuration and make a DSM. Three design configurations are studied in this thesis. They are:

Configuration 1	**Configuration 2**	**Configuration 3**
• Use cam	• Use cam	• Use single lever
• Use dual wound return springs	• Use single torsion return spring and extension return springs	• Use dual wound return springs
• Use straight Aluminum bore	• Use straight Aluminum bore	• Use straight Aluminum bore
• Use idle air control valve	• Use idle air control valve	• Use idle air control valve

Configuration 1 and configuration 2 differ by the throttle return springs they use. Comparing the DSM of these two configurations, one can make some conclusions on the different design choices. Configuration 1 and configuration 3 differ by the linkage type. Configuration 1 uses cam and configuration 3 uses single lever (ball stud) design. The difference of the two designs can be found by comparing their DSM.

In fact, the database is built to include all the information possible for all throttle body configurations. Although this thesis only studied the above three configurations, it is very easy to sort the database for a different kind of combination of elements, and make a DSM for a new design configuration. Meanwhile, since the source of the knowledge is reserved in the database, the DSM is always ready to be updated or corrected.

Step 5: Partition the DSM

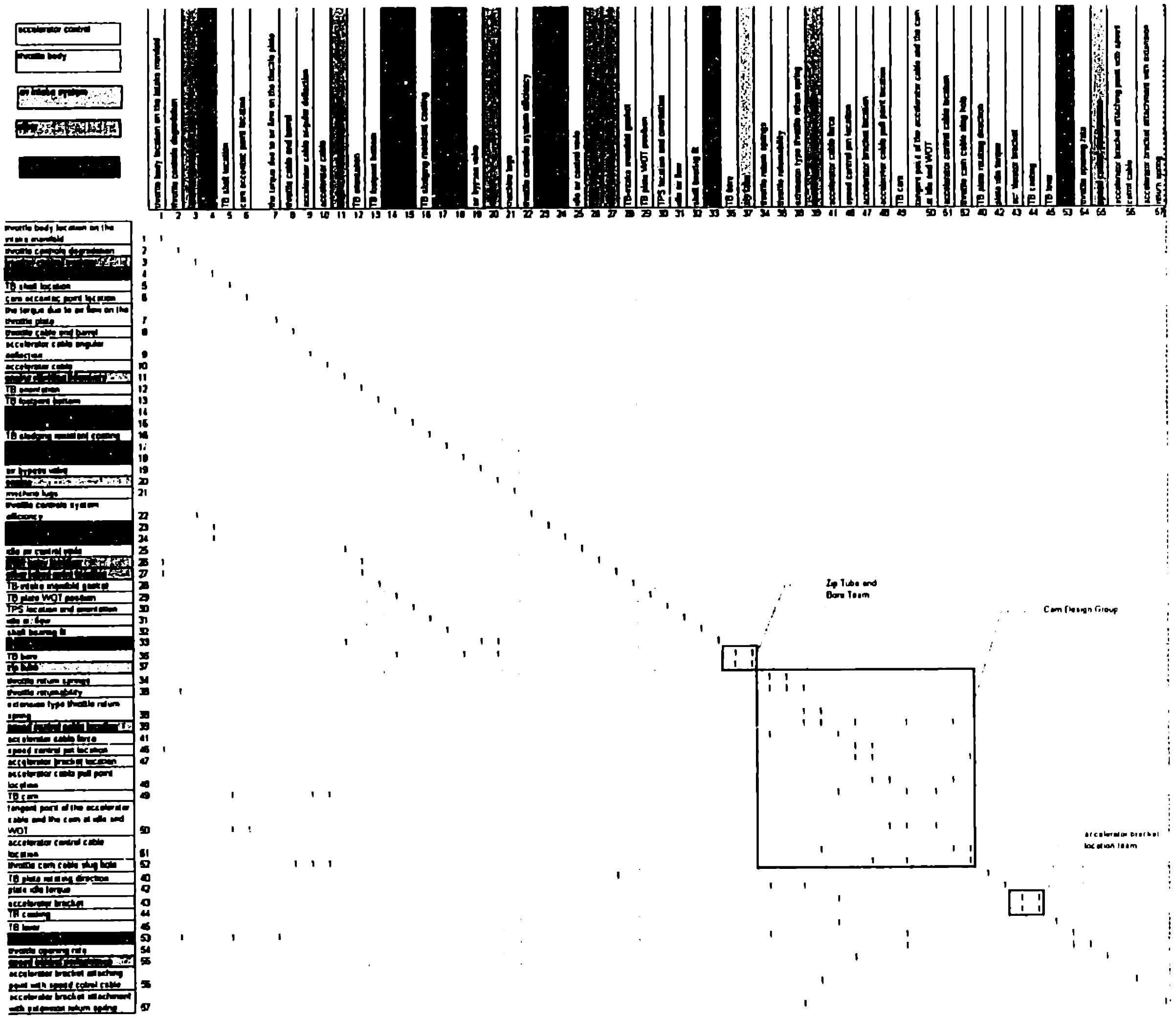
In this step, the Microsoft EXCEL Visual Basic program in Appendix A took the list of element relations from the last step, and partitioned the DSM. The partitioning program can only provide a mathematical result of the partitioning. In other words, the program can tell which elements are coupled and which are not, but it cannot tell what to do within a group of coupled elements. In the case of the throttle body system interface, the program leaves a large chuck of elements (about 26 out of total 40). In order to see the problem more clearly, the author further re-ordered the matrix manually. More discussion on the result will be in the next section of this chapter.

Step 6: Compare the DSM of Various Design Configurations

After the DSMs are partitioned, the DSM of configuration 1 and 2 are compared to see the effect of different throttle return spring designs. In addition, configuration 1 and 3 are compared to see the effect of the different linkage designs. The total number of the system elements, the total numbers of the iteration loops, the complexity of the iterations, and the total number of the *1*'s in each matrices are compared. The results of the comparisons will be discussed in the next section in this chapter.

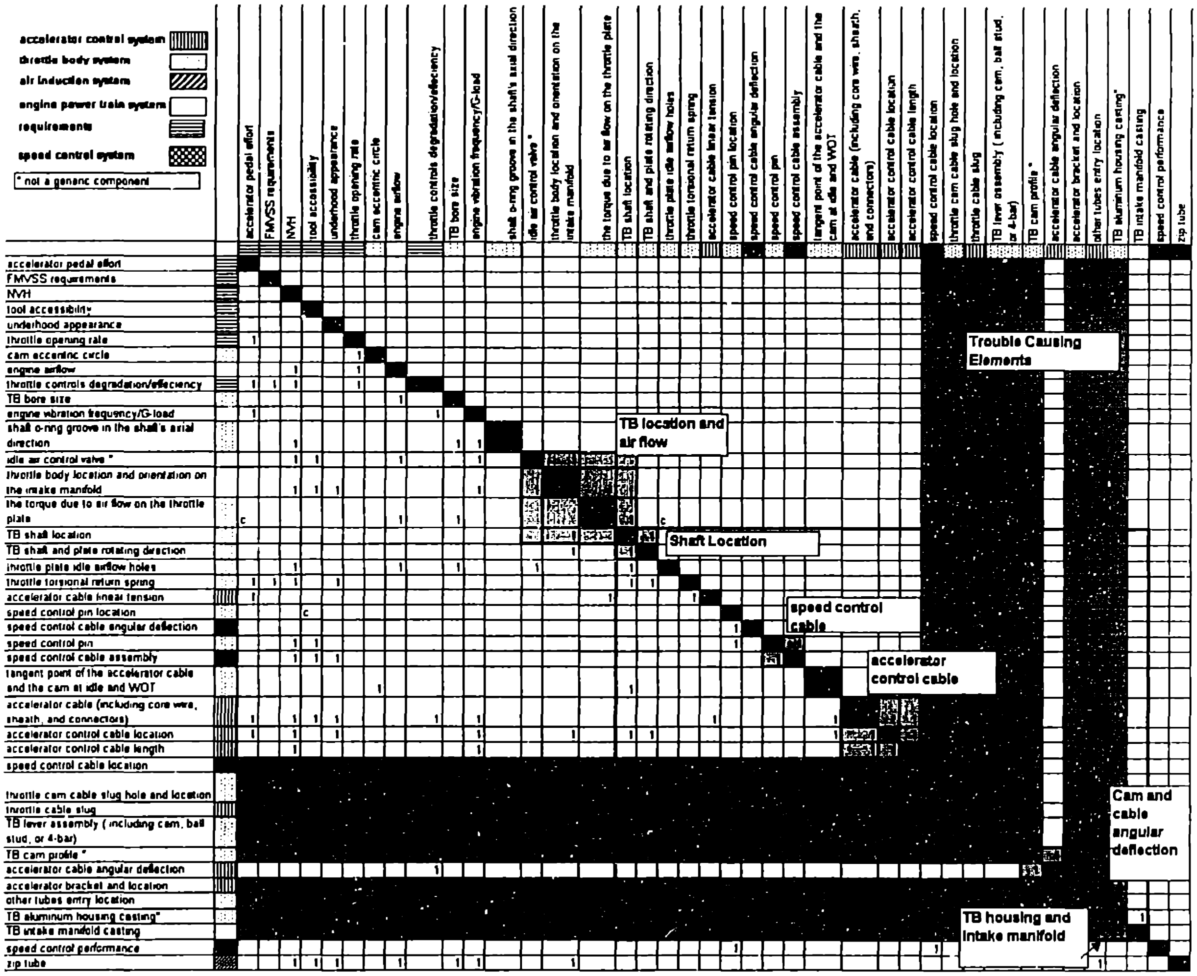
Step 7: Study the FPDS Requirements

In the information database, similar to how design configurations are studied, the 15 FPDS attributes are also listed as columns (not shown in Figure 7-3). Sorting the database based on the FPDS attributes provides a view to how the throttle body system interface design falls into the FPDS requirement categories. The results will be discussed in the next section.



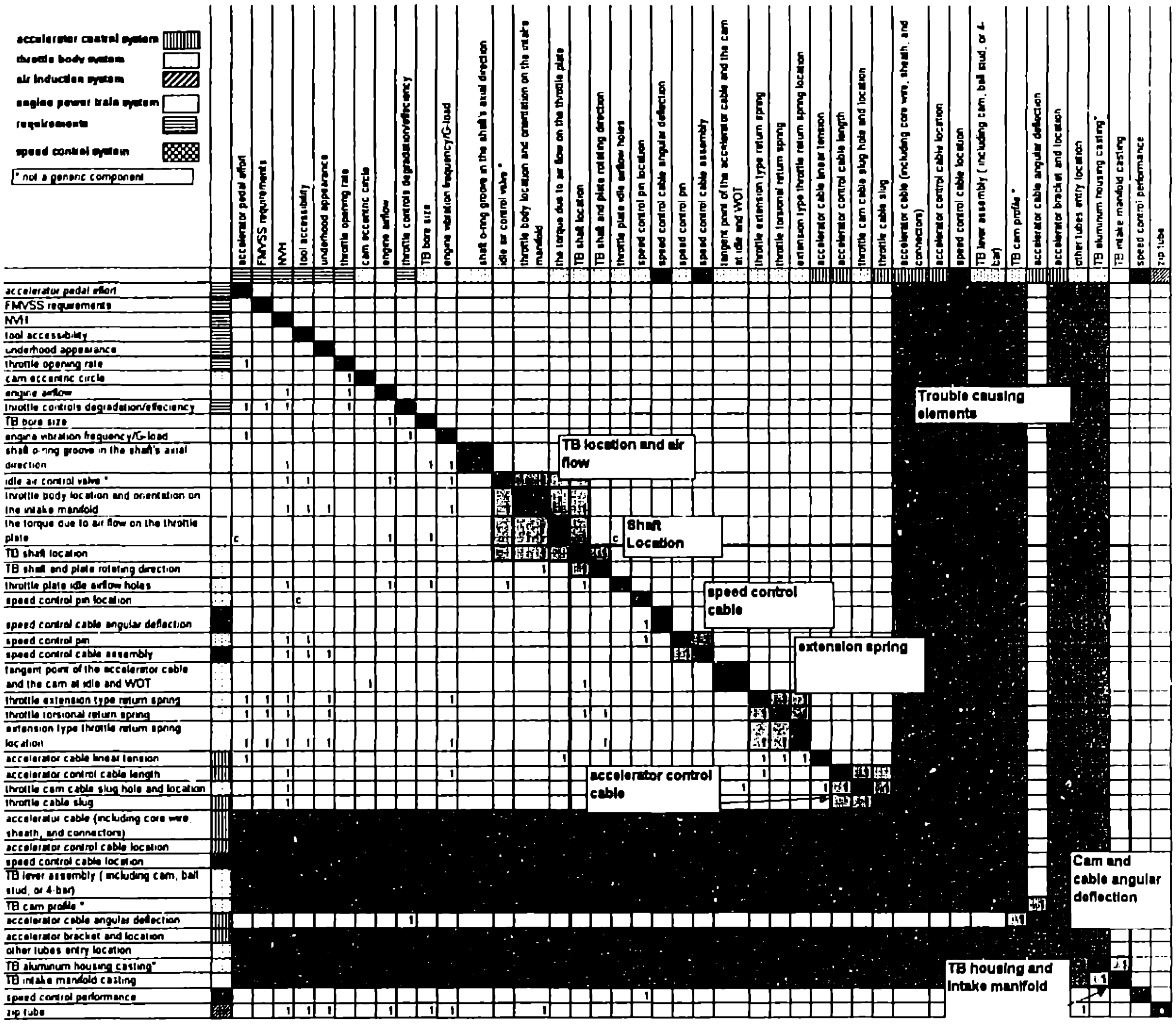
Design configuration: Cam, Dual wound spring, Aluminum straight bore, and use idle control valve.

Figure 7-4 the First Draft of the DSM from Reading the Design Documents



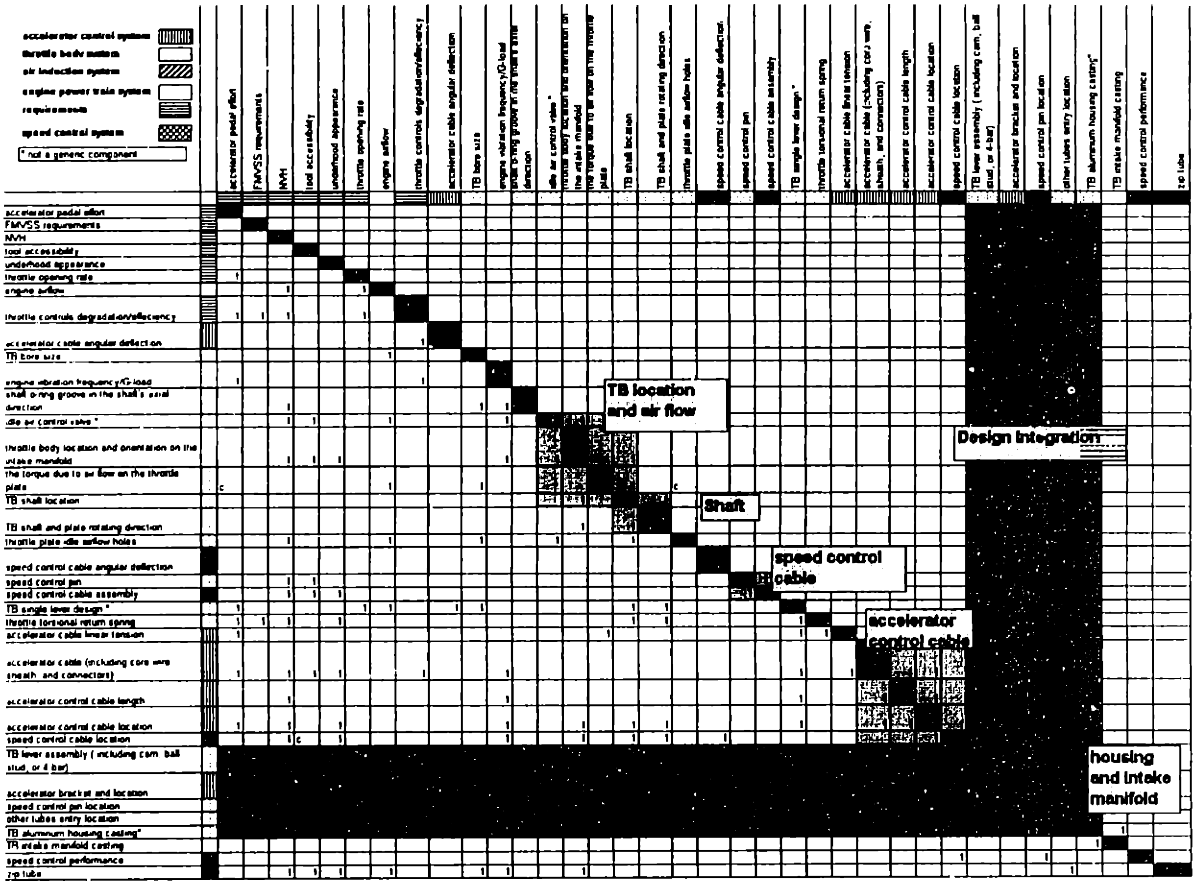
Design configuration 1: Use Cam, dual wound spring, Aluminum straight bore, idle air control valve

Figure 7-5 Throttle Body System Interface DSM (Design Configuration 1)



Design Configeration 2: use cam, torsion spring and extension spring, Aluminum straight bore, idle air control valve

Figure 7-6 Throttle Body System Interface DSM (Design Configuration 2)



Design configuration 3: use single lever, dual wound torsion springs, Aluminum Straight bore, idle air control valve.

Figure 7-7 Throttle Body System Interface DSM (Design Configuration 3)

7.5 Results and Discussion

The final DSMs for the three design configurations are shown in Figure 7-5, 7-6, and 7-7. The observations made will be discussed in six major topics:

1. Throttle body system interface;
2. Throttle body design variations;
3. Design knowledge capturing and reuse;
4. Local and non-local knowledge;
5. Implications to DE applications;
6. FPDS attributes.

7.5.1 Throttle Body System Interface

The following observations can be made about the throttle body system interface:

1. The major design iteration at system interface level exists between the acceleration control system, speed control system, and the throttle body system. In the DSMs in Figure 7-5, 6, and 7, system elements from "idle air control valve" to "TB aluminum housing casting" are involved in this large iteration loop, which includes about 65% of the total system elements.

2. Since the iteration loop mentioned above is so large, it is hard to get much improvement just by recognizing its existence. In order to get more useful information, the author further re-ordered the DSM manually. The results are shown in figure 7-5, 6, and 7. A smaller iteration loop is separated out, and is named "TB location and air flow." A design Team can be formed based on this loop. However, this iteration block still iterates with the rest of the large iteration loop mentioned in point 1 through "TB shaft location."

   The result of manual re-arrangements show that a few elements seem to have the most influence on the large design iteration loop. These elements are highlighted in darker gray in the DSMs. Any change to one of these elements will result in large design iterations.

Excluding these large-iteration-causing elements, the rest of the iteration loop can be further divided into smaller iteration blocks inside the larger design iteration loop, such as the "shaft location", "speed control cable" in Figure 7-5. Design teams can be formed around these iteration blocks.

By manually further separating the large iteration loop mentioned in point1, the throttle body design system interface is better understood, and smaller concurrent design teams can be formed to improve the efficiency of the design process.

3. Since the DSMs record the current throttle body design process, the existence of large iteration loops and large-iteration-causing elements raises the question whether the throttle body design process is optimal. Pete Sferro argues that sometimes the engineers are given too much design choices, and are creating unnecessary design iterations. For example, if housing casting design can be standardized, many guessing and iterations can be eliminated. This is also the reason why part standardization can speed up the design process.

4. By exposing the large iteration loops and the large-iteration-causing elements, the DSMs provide engineers an opportunity to see how the design is carried out as a whole. It prompts the engineers to re-think of the throttle body design itself. Maybe a better design can be done to reduce the number of iterations, and hence save development time.

5. The air intake system plays a minor role in the system interaction. The only interaction between the air intake system engineers and the throttle body engineers is the physical connection between the throttle body bore and the zip tube, which usually worth little discussion because of the standardized design.

6. Most system interactions are between the throttle body cam design and the accelerator control cable connection to the throttle body. The engineers in both system teams have to

work very closely together to decide the design. Engineer Dennis Madson agreed on this observation. Dennis had to attend many accelerator controls engineering meetings, and work closely with the accelerator control engineers for design changes. In addition, the accelerator control System Design Guide contains a part regarding the throttle body cam design.

7. The patterns in the second row and column of each DSM are highly mixed. Design teams of each system have to work closely in order to resolve issues in the design process at the system interface level. Based on the iteration loops, a few design teams are proposed to deal with issues in the design problem, such as the "TB location and airflow" team, the "shaft location" team, etc. (see Figure 7-5, 6, and 7). These proposed design teams are of small sizes, and hence can work more effectively. The experts are brought together to work on design issues in different design stages, and hence are more effective and efficient than having large concurrent engineering meetings. In other words, the same observations as point 2, 3, 4, and 5 in section 6.4 in this thesis also apply here.

8. Another interesting observation is that most of these large-iteration-causing elements belong to the throttle body itself. Then it should not be too difficult to manage the TB interface to other systems, assuming the engineers within TB communicate well with each other.

7.5.2 Throttle Body Design Variations

Figure 7-5, 6, and 7 represents the design process of three different configurations as listed in the last section. Table 7-1 shows a comparison among the three configurations regarding the total number of system elements, the matrix entries, and the large-iteration-causing elements.

Table 7-1 Comparison of the DSMs for the Three Design Configurations

	configuration 1	configuration 2	configuration 3
total number of system elements	40	42	37
total number of DSM entries (the 1's)	216	253	192
total number of large-iteration-causing-elements	8	8	5

Configuration 1—Figure 7-5; Configuration 2—Figure 7-6; Configuration 3 –Figure 7-7

Additional observations on the different design configurations can be made as follows.

Dual-wound Throttle Return Springs vs. Single-wound Throttle Return Spring and Extension Return Springs

Throttle body engineers have been long arguing the dual wound return springs are a better design than the extension return springs. Since configuration 1 and configuration 2's only difference is the return springs they have, comparing the DSM's for the two configurations provides a clear proof to the argument above from the system-interface point of view. The following observations can be made:

- Dual-wound spring design has fewer system elements. Configuration 2 has 42 system elements, and Configuration 1 has 40 system elements. The two additional system elements in Configuration 2 are "extension return spring locations" and "throttle extension type return spring". Both of the extra elements are due to the addition of extension spring as part of the design configuration. The rest of the system elements remain the same.

- Dual wound springs reduce the amount of work involved in engineering design. Configuration 2 has 253 *1*'s in the DSM, while configuration 1 has only 216. Using extension spring brings 37 more issues to think about at the throttle body system interface level.

- The presence of the extension spring makes the design more iterative. Configuration 2 has 8 large-iteration-causing elements, while configuration 1 has only 7. Configuration 2 also has an additional iteration loop caused by the extension spring design compare to configuration 1.

In conclusion, from a system interface point of view, dual wound spring creates a simpler system interface for the throttle body. For the first time ever, the DSMs provided Ford engineers a means to evaluate the complexity of the design quantitatively.

Single lever linkage vs. Cam design

The Accelerator Controls System Design Guide points out that single lever is cheaper than other types of linkage, including cam, four-bar linkage, and pin-and-slot linkage. Since the only difference between Configuration 1 and Configuration 3 is that Configuration 1 uses cam and Configuration 3 uses single lever, their DSM's provide justification to the system design benefit of the single lever linkage. The following observations are made:

- Single lever design requires fewer system elements. Configuration 3 has 36 system elements, and Configuration 1 has 40 system elements. The five additional system elements in Configuration 3 are "cam eccentric circle", "tangent point of the accelerator cable and the cam at idle and WOT", "TB cam profile", "throttle cam cable slug hole and location", and "throttle cam cable slug hole and location". The one system element unique in Configuration 1 is "single lever design". The single lever design replaces the TB cam profile. The rest of the additional system elements in Configuration 1 are caused by the involvement of cam design.

- Single lever design significantly reduces the amount of work involved in the engineering process at the throttle body interfaces. Configuration 1 has 216 1's in its DSM. Configuration 3 has 192 1's in its DSM. There are 24 fewer system interface issues to consider when a single lever design is used.

- The presence of the cam makes the design more iterative. Configuration 1 has 7 large-iteration-causing elements, while configuration 3 has only 5. The cam design also contains one more iteration block which is the “cam and cable angular deflection”.

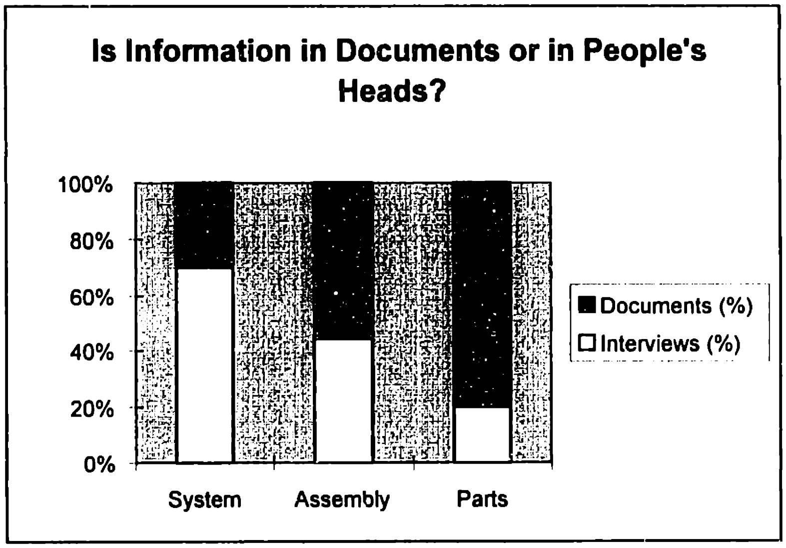
Hence, in conclusion, single lever design is not only cheaper, but also a much simpler system interface design problem. However, because the single lever cannot provide a sophisticated throttle opening curve, a complicated progression bore has to be used. The cam design only needs a simple straight bore. More tradeoff evaluations has to be done before making a design decision between the two.

7.5.3 Design Knowledge Capturing and Reuse

The following observations are made regarding the design knowledge capturing and re-use.

1. The system elements contain not only physical components in the system, but also functional requirements, performance requirements, etc. In fact, the system elements are chosen so that all the issues during the design process can be properly captured. The reason to include the function requirements in the DSM is to see how the requirements drive the system design. The existence of a mixed type of system elements indicates that the interactions between design teams can not be captured by geometry only. Hence, not all the information exchange can be carried by the CAD drawings of parts.

	Interviews (%)	Documents (%)
System	70	30
Assembly	44	56
Parts	20	80



Note: the numbers for the System and Assembly are from the information database. The number for the parts is an estimate. However, the trend is that as the scale of the design gets smaller, more knowledge is documented.

Figure 7-8 How Much Knowledge is Documented

2. The first draft of the DSM was made from the records in the design documents only (see Figure 7-4). It seems very simple and linear compared to the final DSMs shown in Figure 7-5, 6, and 7. A large amount of the information in the finished DSMs was added later through interviews with engineers and attending engineering meetings. Based on the "Source" column in the information database, the author was able to count how much knowledge is documented and how much is kept in engineers' head by counting how many interactions recorded in the database are from documents and how many are from interviews. Figure 7-8 shows the result in a bar chart form that is made by Professor D. Whitney and the author. In this chart, the number for the system level is from the

information database mentioned earlier in Figure 7-3. The number for the assembly level is from the information database built to study the throttle body assembly, which will be discussed in the next chapter. The number for the parts is an estimate. However, it is there to show the trend that as the design scale gets smaller, more knowledge is documented. The design process is poorly understood at system level. The engineering design documentation is mostly about the design of parts.

The consequence of the above observation is that a team has to go through learning curves constantly. At Ford, most engineers rotate among different positions to gain a variety of experiences and rise in the company. The average length of one rotation assignment at Ford is 2 years, which is just enough for the person to acquire expertise in the position. From Figure 7-8, we can see when the experienced engineer leaves his/her current position, he/she takes away a large portion of the system level knowledge from the team. The replacement has to go through a period of time before he/she can gain the same amount of system level knowledge. Furthermore, reading the design documents about the design of parts still cannot guarantee the new engineer know how to design the parts in the real world. The learning curve is further extended by unable to effectively use the existing design documents. This learning period can damage the quality and work efficiency of the team, and hence affect the speed of the product development process.

3. The DSMs tell us the need for information exchange between two elements. The database tells us what information need to be exchanged, why, and who said it. A database as such can serve to accumulate the learning in an organization, and as well as provide a complement to the existing imperfect design documents. When an experienced engineer leaves the team, the new engineer can use the information database to quickly catch up with the design process. The team can avoid making mistakes or delaying the design process.

4. The DSM together with the information database can be used as a good browser for knowledge learned. The existing Ford design documents made an effort to capture knowledge learned for future reuse. However, the current practice is to store all the knowledge in pages of documents without an easy user interface. In order for these documents to serve full function, an engineer has to be able to remember all the information as well as to be able to properly apply them when needed. When a design concerns the interface of several systems, an engineer has to read a large amount of documents. The reality is that no one has such a good memory. Hence, the incomplete documentation of the process (see the point 2 in this section) serves even less than it could. With the aid of DSMs, the engineers can easily find what inputs they need for their work, and what outputs they give. If the engineers need to know more about the inputs and outputs, they can go to the database to find out. The engineers no longer need to memorize everything in order to do a good job. The learned knowledge can be used more effectively.

7.5.4 Local/Non-local Knowledge

Local knowledge is the knowledge that is kept and controlled within a system team--throttle body system team in this case. Non-local knowledge is the knowledge that needs interactions between the throttle body team and other teams that interface with it. Non-local knowledge could be kept by all interacting system teams, or partially owned by one or more teams and is not well understood. Once the DSM is built, the 1's that involves two system elements with the same color indicate local knowledge. The 1's that involves two system elements with different colors indicate non-local knowledge. With the aid of the DSM and the information database, the engineers can easily find not only the local knowledge, but also the non-local knowledge.

7.5.5 Implications to DE Applications

Since all the DE applications are only on relatively small systems such as the throttle body, system interface DSM's can help to integrate separated applications. The throttle body system interface DSMs tells us how throttle body DE program should communicate with the DE programs made for the rest of the interfacing systems.

The existence of the iteration loops indicates the limit of the computer automation. The current technology makes the DE programs only possible to shorten the time it takes to complete an iteration cycle. Human interactions are still needed where iteration loops exist. In the future, it is still possible that a more sophisticated DE system (with a knowledge or inference engine that can do some iteration by itself) can handle the iterations.

Compare using the DSM as knowledge browser mentioned in section 7.5.2 to the DE and Knowledge Based Engineering (KBE) at Ford, DSM seems to be a low cost and passive knowledge based engineering system. DSM acts like a browser that provides users directions to use the existing information database. It does not judge the situation for the engineers. Engineers have full freedom in making design decisions. DE and KBE have more ambitious goals. They intend to incorporate the design knowledge in the computer program as design rules, while actively making decision for the users during their design, as well as providing a process for making the design. Chapter 11 in this thesis will discuss the engineers' responses to the automated design systems. Unless the computer system is excellent or the engineers are forced to use the computer system, the engineers usually do not like their freedom of making decisions to be taken away. It is still unclear whether an active or passive approach to the knowledge management is better.

7.5.6 FPDS Attributes

Figure 7-9 shows how much information exchange falls into each FPDS attribute. These bar diagrams are made through sorting the information database based on the FPDS attributes. The total number of the information exchange that belongs to a specific FPDS attribute is divided by the total number of the information exchanged recorded by the database to get the percentage ratio. The following observations are made on the FPDS attributes:

1. The FPDS percentages do not add up to 100 because each information transfer may belong to more than one FPDS attributes.

2. The packaging requirement is the dominant factor for the TB system interface, followed by NVH and performance requirements. Safety, Customer life cycle, and Product/Process compatibility are the third.

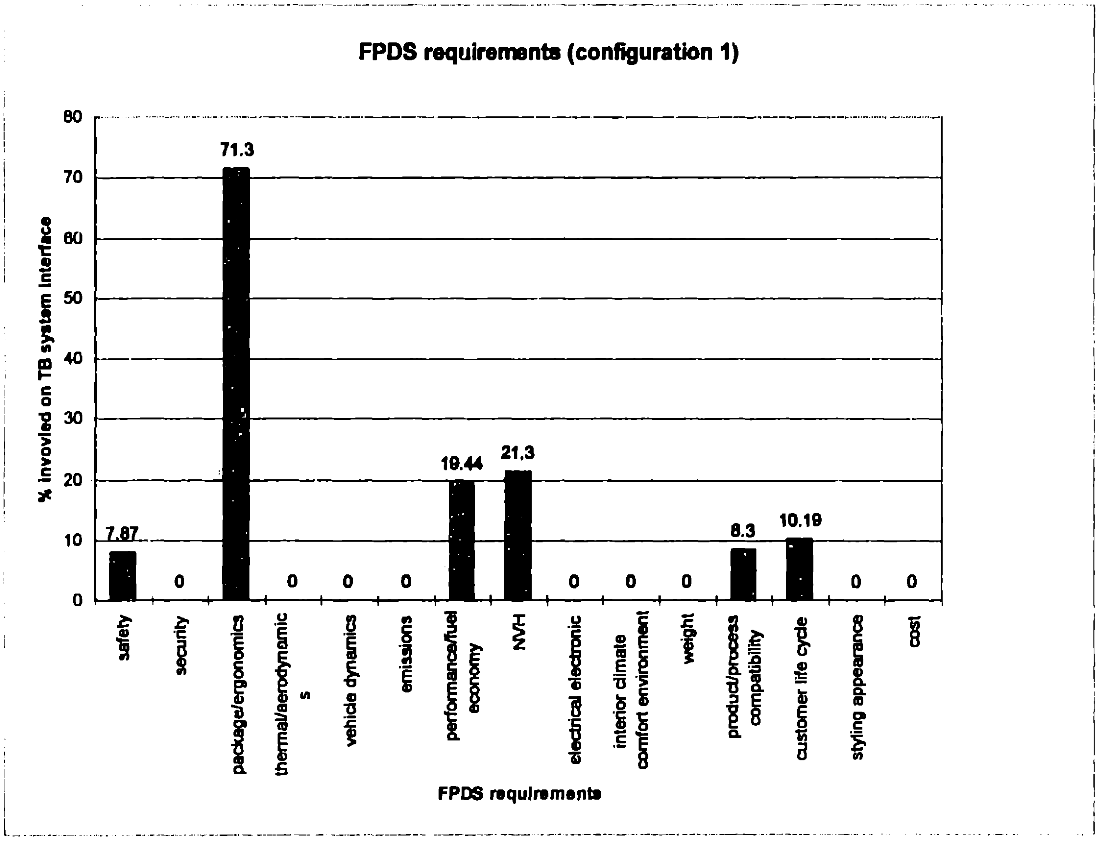
3. Nine out of the 15 FPDS requirements do not play any role in the TB system interfaces. Such requirements include vehicle security, thermo/areodynamics, vehicle dynamics, emissions, electrical/electronics, interior climate control environment, weight, styling/appearance, and cost. Some of the above requirements such as interior climate control environment naturally do not drive TB system design due to the location and function of TB. However, surprisingly, cost and weight did not appear explicitly in the TB system interface DSM. The absence of the Weight factor could also be due to the lack of ability to calculate the impact of the design change of one part on the system and the entire vehicle. Two possible reasons for the absence of cost factor are listed as follows:

   First, there isn't an existing method to calculate the cost of design change. The cost of a design at system interface level not only involves the cost of all the systems that interact with it, but also concerns the cost of material, manufacturing, assembly operator cost, transportation, etc. of all the systems involved. Hence the cost calculation is very complicated. At system level, no method has been documented. Hence, only very general guidelines are provided in design documents.

   Second, the design of a product such as throttle body is more function driven. Since the cost of the design cannot be checked immediately, it usually is checked after the design is made.

4. The absence of weight and cost mentioned above indicate that in order to effectively decompose the vehicle level requirements to system level, means of evaluating how designs meet some of the attributes are needed.

5. Furthermore, the lack of information about cost and weight in the DSM indicates that the current design documents and SDS are not closely related to the FPDS requirements. Ford faces the challenge of decomposing the FPDS attributes to lower system levels as mentioned in Chapter 2.



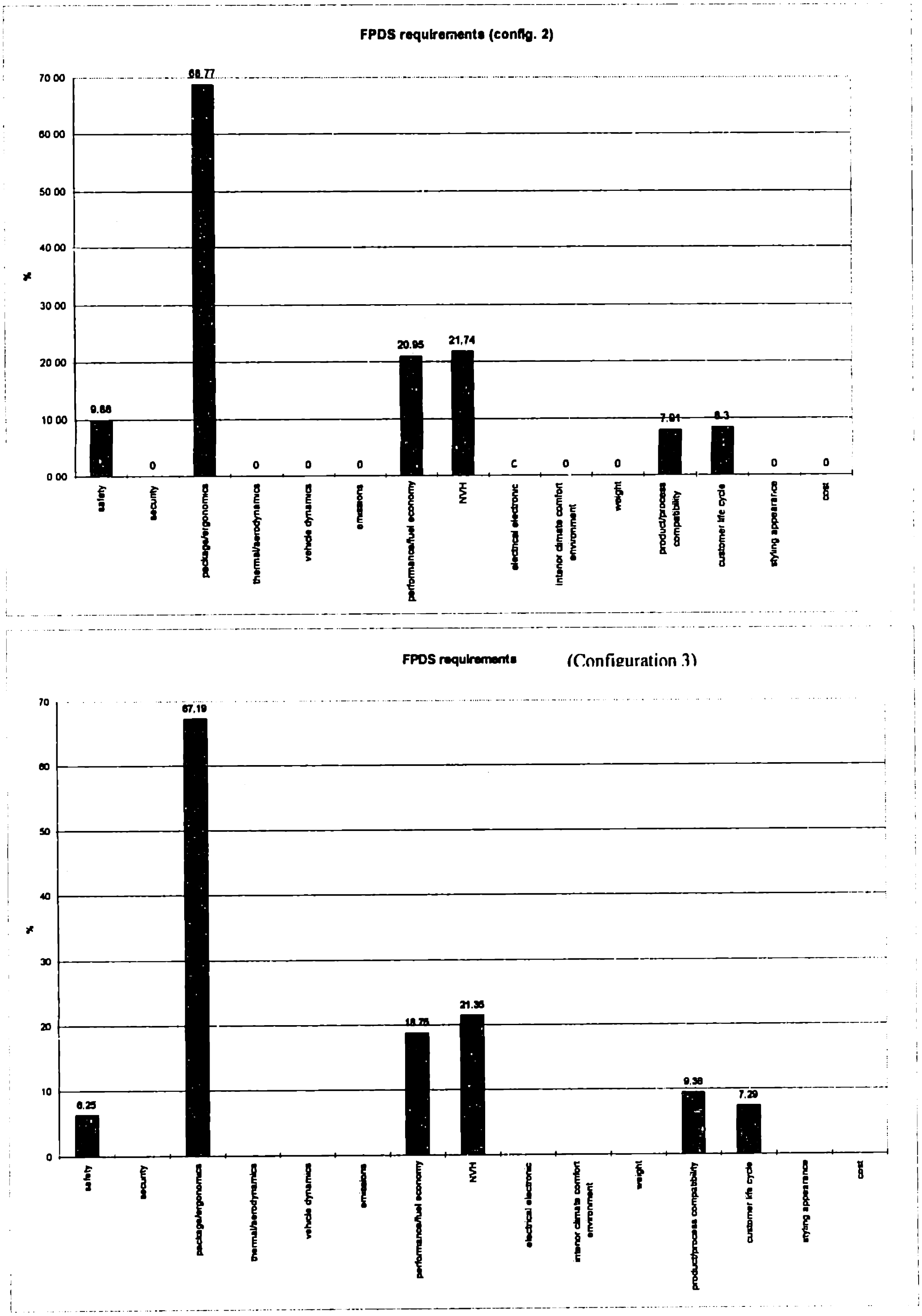


Figure 7-9 FPDS Attributes

7.6 Chapter Summary

This chapter presented the case study of using DSM to understand how the Ford throttle body interfaces with other systems around it, including the accelerator control system, engine power train system air induction system, and the speed control system. Figure 7-5, 6, and 7 show DSMs made for various throttle body design configurations. Figure 7-3 shows a small part of the database that records all the interactions included in the three DSMs mentioned above. The following observations are made from this case study.

About Throttle Body System Interface

1. The throttle body mainly interfaces with the accelerator control system and the speed control system. The air intake system plays a minor role.

2. The large iteration loops can be manually further divided into smaller iteration loops and large-iteration-causing loops to better understand the problem causes and to better manage the design process.

3. Some of the large-iteration-causing elements may be eliminated by standardizing parts for reuse.

4. Seeing the whole picture of the throttle body design process prompts engineers to seek improvements in the current design process.

5. The partitioned DSMs broke up the existing design teams, and suggest to bring experts from different organizations to form teams and deal with inevitable design iteration loops. New teams are built around issues in designs and are smaller. Hence, more effective product development process can be expected.

About Throttle Body Design Variations

DSM for the first time provided quantitative means to compare and evaluate design alternatives. In this case study, dual-wound throttle return spring leads a simpler design process compared to the single-wound plus an extension return springs design. Same applies when compare single lever linkage design to the cam design.

Design Knowledge Capture and Reuse

1. The system elements contain not only physical parts and features, but also engineering parameters that concerns the design. CAD drawings cannot carry all the information involved in the product design process.

2. Design knowledge is best documented at part level, and less documentation at assembly and system level. The organization loses efficiency when expert engineers leave the team and rotate to new position. DSM proves to be an effective way of keeping the system level knowledge in the organization.

3. DSM serves as good browser for reusing the captured knowledge. It is a passive KBE system. It does not provide design process to the user while DE and KBE do.

Local and Non-local Knowledge

DSM gives clear indication of local and non-local knowledge through the use of patterns for different organizations. Furthermore, DSM is also an effective tool to document and browse non-local knowledge.

Implications to DE Applications

1. DSM can help integrating the individual DE applications.
2. DSM indicates design iterations inevitably exist. The current DE program cannot automate the entire design process. Hence, human interactions are needed.

3. DSMs are passive knowledge browsers that do not make decisions for the users. DE and KBE applications also provide the design decision making process for the users. Which one is better cannot be said yet.

FPDS Attributes

Packaging issue dominates the TB system interfaces. Many FPDS attributes do not play roles in the TB system interface. Some of these attributes are not applicable to the TB system interfaces. The absence of Cost and Weight are due to lack of means to evaluate them at the system level. The SDS's are developed independent to the VDS.

CHAPTER 8 STUDY THROTTLE BODY ASSEMBLY DESIGN USING DSM

8.1 Objectives

Figure 7-2 has shown the system engineering's view of the throttle body system. Chapter 7 discussed using DSM to study the system interface of the throttle body system, which is how the throttle body bubble as a unit interacts with other system bubbles. This chapter looks at the details within the throttle body bubble, i.e. how the components in the throttle body interact with one another. Besides the general research objectives mentioned in chapter 1, the following objectives also apply to this case study:

- Discover and document important relation specific to the throttle body assembly design.
- Show what design rules the DE program needs to incorporate.
- Show information flow and coupling in the throttle body assembly.
- Compare the difference between the throttle body assembly design and the throttle body system interface.

8.2 Steps Taken

This research took the following steps:

1. Define the system and its scope;
2. Define the elements involved in the system;
3. Define the interactions between the system elements and build an information database to record the interactions;

4. Choose a particular design configuration;
5. Partition the matrix;
6. Compare the DSM with throttle body system interface DSM;
7. Study the FPDS requirements.

Step1: Define the System and Its Scope

This research studies the throttle body assembly design. Only the interaction between throttle body assembly components was considered. The system level interaction was not included here. The throttle body components include:

- Housing casting
- Lever and Cam
- TRC screw
- Shaft
- TPS
- Bearing
- Plate
- Spring
- O-rings

The final DSM does include two elements from the accelerator control system (the accelerator bracket) and the air induction system (the zip tube) because otherwise some of the relations cannot be recorded clearly.

The design configuration studied here is the same as the design configuration 1 in chapter 7, which includes cam, idle air control valve, dual-wound springs, and Aluminum straight bore. The purpose of choosing this specific configuration is to compare the throttle body system interface with the throttle body assembly design. Although it can be very easy to make DSMs to study other design configurations, the author think it is unnecessary for the purpose of this thesis, because the throttle body assembly DSMs are simple as will be illustrated later in the results and discussion section.

The "parallel iteration" defined by Dr. Robert Smith and Steven Eppinger still applies here [Smith (1997)]. Parallel iteration is defined as "an extreme case where a number of development activities are underway at one time." Since the Ford throttle body design process is well defined and did not change during the course of this study, this assumption applies here.

Step 2: Define the Elements Involved in the System

This step took the same form as that for studying the throttle body system interface, except that only the throttle body design documents were needed. For the details, please refer to the corresponding section in chapter 7.

Step 3: Define the Interactions between System Elements, and Build an Information Database to Record the Interactions

This step is also similar to that in understanding the throttle body system interface. The same group of experienced Ford design engineers were consulted. The author also attended engineering meetings to gather information. For the details of the process, please refer to the corresponding section in chapter 7. This step is highly iterative with step 2. In the end, a set of system elements that can well represent all the interactions between throttle body component designs were chosen. Furthermore, an information database that is similar to the one in figure 7-3 was built. For the details of the EXCEL database, please refer to the corresponding section in chapter 7. This information database contains information for all design configurations. Hence DSMs for other forms of the throttle body design can be easily made.

Step 4: Choose a Specific Design Configuration

Identical to that described the corresponding step in chapter 7, the Microsoft EXCEL "sort" function sorted out the list of system element interactions that are specific to the chosen design configuration: use cam, dual-wound spring, straight Aluminum bore, and idle control valve.

Step 5: Partition the Matrix

The program in Appendix A was used to partition the DSM. Contrast to the situation in chapter 7, the partition result needed very little manual manipulation due to the simplicity of the DSM, as the readers will see later in the results and discussion section.

Step 6: Compare the Assembly DSM Throttle Body System Interface DSM

The throttle body assembly DSM and the system interface DSM were compared. Many observations are drawn, as will be listed in the results and discussion section.

Step 7: Study the FPDS Requirements

The information database for the throttle body assembly also contains columns for the FPDS attributes. The database was sorted and provided a view of how the FPDS attributes drive the design of the throttle body assembly. The results will be in the next section.

8.3 Results and Discussion

The DSM for the throttle body assembly based on the current component design divisions is in Figure 8-1. The same DSM after partition is shown in Figure 8-2. Discussions on observations made will be carried out for the following two major topics:

1. Observation on the throttle body assembly DSM.
2. FPDS attributes.
3. Comparison between the throttle body assembly design and the throttle body system interface.

The observations on the design knowledge capturing and reuse, the local and non-local knowledge, and the implications to the DE applications are the same as that in chapter 7, and will not be repeated here.

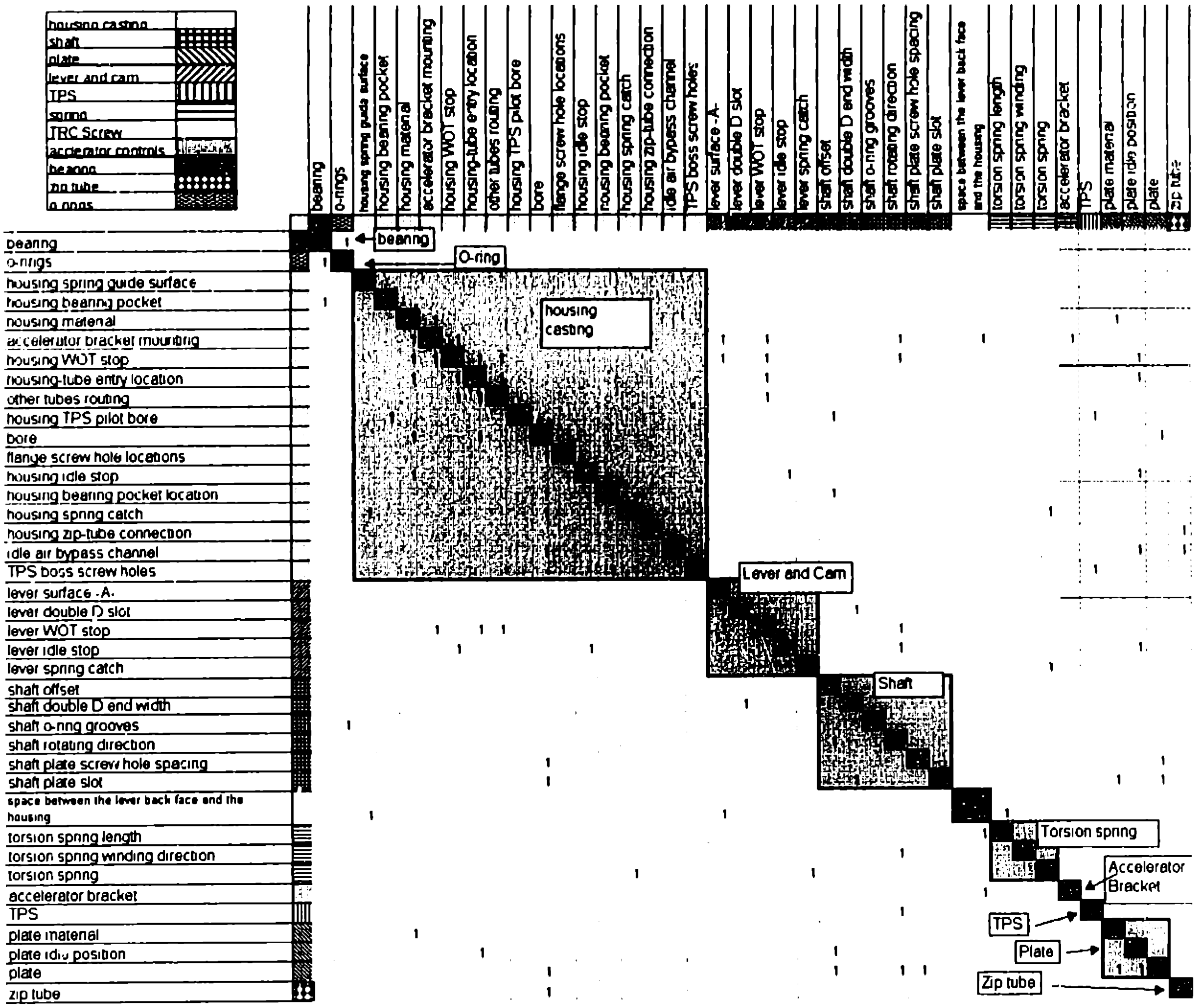


Figure 8-1 Throttle Body Assembly DSM Based on the Current Component Division

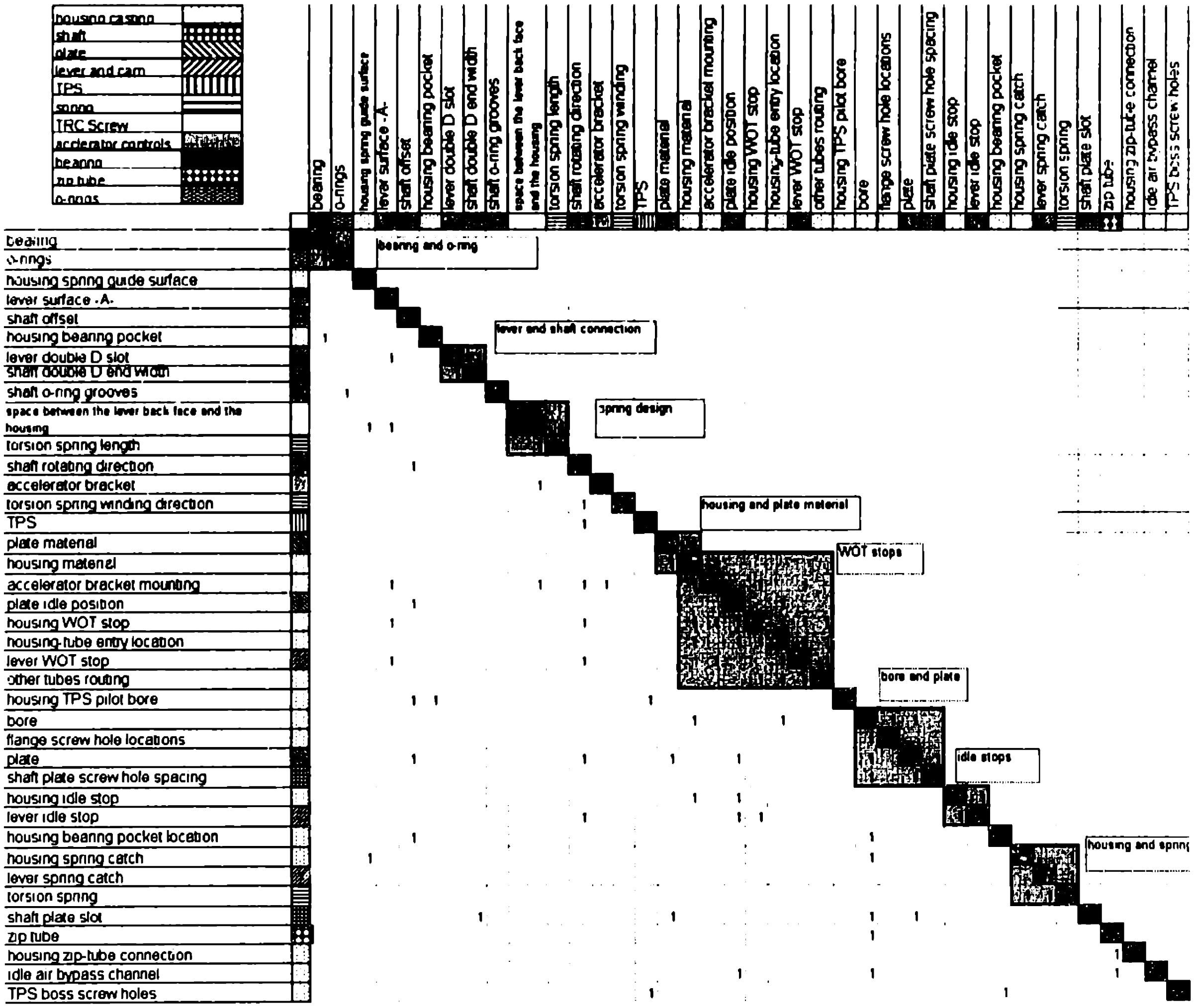


Figure 8-2 Throttle Body Assembly DSM after Partitioning

8.3.1 Observation on the Throttle Body Assembly DSM

The following observations can be made for the throttle body assembly design:

1. Figure 8-1 shows that the current component divisions need to interact closely with one another for the throttle body assembly design because there are many *1*'s in the DSM are above diagonal and these *1*'s are not captured by single design teams. Figure 8-2 proposes forming design teams around the design iterations loops. Smaller design teams

are formed, and all inter-organizational communications are captured by the proposed design teams. The second row and column in Figure 8-2 show a highly mixed pattern. Hence, close communication and collaboration among throttle body component design engineers are needed. More discussions regarding the proposed teams can be found in section 6.4 of this thesis.

2. The iteration loops in this case are small. Changing the design configuration may add or reduce a few iteration loops. However, since these loops are so small, the design process won't be significantly affected due to design changes. This is also the reason that the author did not make more DSMs like in Chapter 7 for different design configurations at the assembly level.

8.3.2 FPDS Attributes

Figure 8-3 shows the FPDS attributes distribution at the throttle body assembly level.

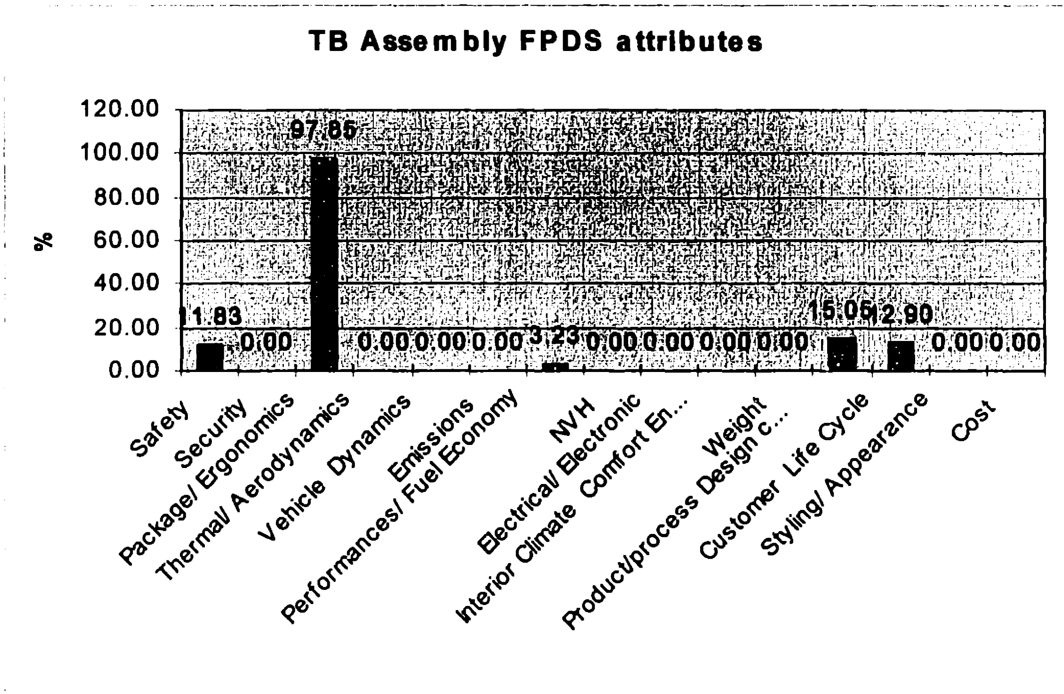


Figure 8-3 Throttle body Assembly FPDS Attributes Distribution

The percentage of each FPDS attribute does not add up to one because each information transfer may involve more than one FPDS attributes. The following observations are made:

1. The "Package/Ergonomics" dominates the throttle body assembly design. Next are the "Product/Process Compatibility", "the Customer Lifecycle", and the "Safety".

2. Many FPDS attributes have zero values in the chart in Figure 8-3. Some of the attributes do not play a role in the throttle body assembly design such as the "Security". The "Weight" and "Cost" do not appear for similar reason as discussed in chapter 7.

8.3.3 Comparison between the Throttle Body Assembly Design and the Throttle Body System Interface

Since the design configuration for the DSM in Figure 8-3 is the same as that in Figure 7-5, comparisons between the two reveal the differences between the throttle body system interface design and throttle body assembly design. The following observations can be made:

1. The throttle body system interface DSM contains 216 entries, while the assembly DSM contains only 93. The number of system elements are similar in both (approximately 40). Hence, the throttle body system interface is a more complicated problem than the throttle body assembly design itself. From the System Engineering point of view, we should divide teams in a way so that most interactions occur within a team rather than across the organization boundaries. In the case of throttle body, the organizations that interface with the throttle body team are resided as far as thirty minutes drive away. The current team division contributes to the inefficiency in the design process.

2. The iteration loops in the throttle body assembly DSM are a lot smaller than those in the system interface DSM. It appears that the throttle body design is simple and manageable. All complication occurs at the system interface level. This observation also supports the

System Engineering argument made in the last paragraph.

3. For the FPDS attributes, "Packaging/Ergonomics" dominates both the assembly and system interface design. Other FPDS attributes appear to have the same pattern in both DSMs except the "performance/fuel economy" and "NVH". These two attributes play important roles at the system interface because they dictate the spring and linkage designs, which are driven by the system interfaces.

8.4 Chapter Summary

The chapter uses DSM to study the throttle body assembly, i.e. how parts in the throttle body assembly interact with one another. The dual-wound spring, cam linkage, straight Aluminum bore, and idle control valve design configuration is used for comparison purpose.

The partitioned DSM in Figure 8-2 shows a better and more efficient design process than the current practice in Figure 8-1. Close communications between different component design teams are needed for the throttle body assembly design.

The distribution of FPDS requirements is in Figure 8-3. "Package / Ergonomics" dominates the TB assembly design. Many FPDS attributes have zero values for the same reasons discussed in Chapter 7.

Compare to the throttle body assembly system interface in Figure 7-5, the throttle body assembly design is much simpler and has fewer and smaller design iteration loops. The current organization division contributes largely to the inefficiency in the design process. Among FPDS attributes, the "NVH" and "Performance/fuel economy" dominate the system interface while do not play a major role at assembly level, for the design consideration at different levels are different.

CHAPTER 9 COMPARE DSM TO ASSOCIATIVITY MAP

Besides the Design Structure Matrix method used earlier, this thesis work also experimented with two other methods for studying the product development process, which are the Associativity Map and the Datum Flow Chain Method. This chapter will briefly introduce the Associativity Map, and compare it with the Design Structure Matrix Method. The next chapter will introduce the Datum Flow Chain method, and compare it with DSM.

9.1 Origin of the Associativity Map

The Associativity Map (AM) is a tool developed by the throttle body Direct Engineering team. Jared Judson has clearly stated in his thesis the origin of the AM [Judson (1998)]:

> "When the engineers developing the Throttle Body DESM Application realized they needed assembly knowledge, they went looking and found very little recorded information. They had to create their own, and in so doing developed a tool which was called the Associativity Map. The story of this map is a telling example of how assembly knowledge is maintained.
> … The Associativity Map was first developed using Post-it Notes and a large roll of paper. Later, it was moved to a computer file."

Hence, the DE activities prompted the engineers to think at system level. The need of a tool that can help the system thinking emerges. Associativity Map is a simple and visual tool that the DE engineers thought of. After it was developed at the throttle body team, many other DE teams tried to adopt this method to help them visualize the system level interaction.

9.2 Throttle Body Associativity Map

An early version of the Associativity Map is shown in Figure 9-1. Each block in the AM represents a feature on one of the throttle body components. The arrows between blocks represent the dependency of component design on one another. The blocks are grouped and colored differently to represent different components. Normally, the map is printed on a large sheet of paper for easy viewing.

Although AM has a good intention to capture system level interactions, Jared suggests in his thesis the following weaknesses:

1. The Associativity Map was difficult to print out because of the large size of the map.
2. It is also hard to keep up-to-date.
3. It was probably never completely correct, and has surely languished since.
4. The engineers contributing to the AM, which was the only record of the complex inter-part relationships outside of the Throttle Body DE Application code, saw it as an "extra" task and only infrequently used it to develop or understand the Throttle Body system model.

In addition, the author also observed that the engineers who program the throttle body DE application rarely refer to the AM. In fact, many times, the engineers said that they had to update the AM because they changed something in the DE program was changed. Hence, in reality, the AM at most served as a record rather that a guide for the DE program. This record is still not up-to-date and complete.

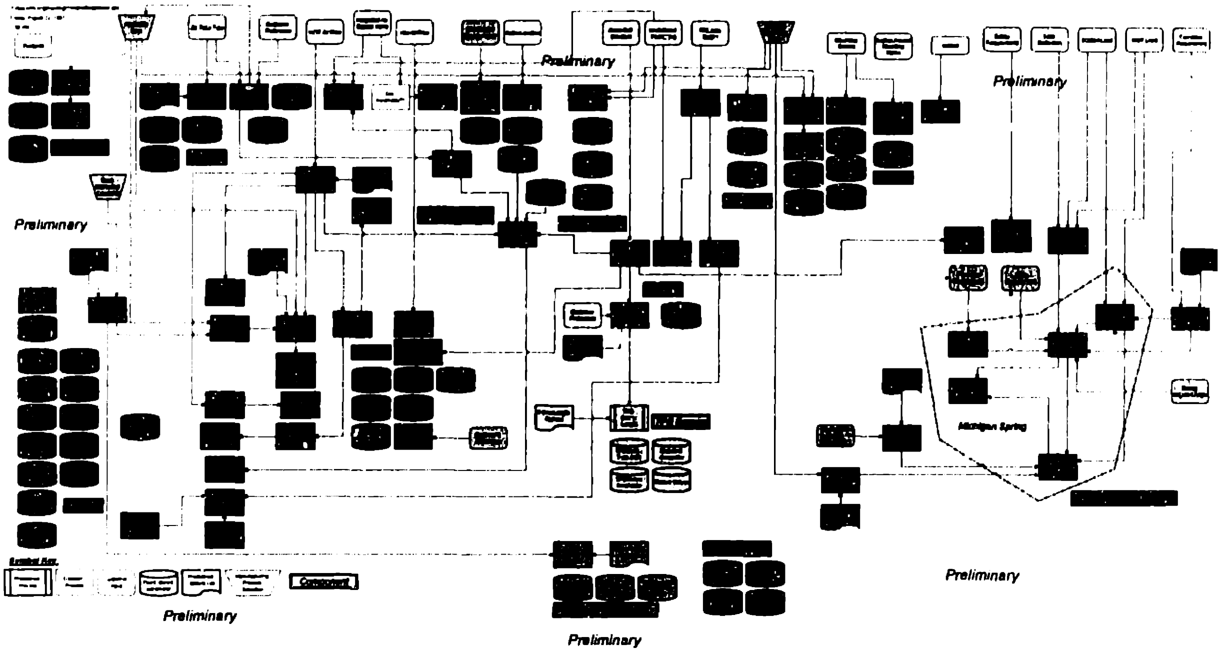


Figure 9-1 Throttle Body DSM

Source: Integrating Supplier Designed Components into a Semi-Automated product Development Environment by Jared Judson [Judson (1998)]

9.3 Compare Throttle Body DSM with Associativity Map

Each DSM has a corresponding digraph, and vise versa. With directional arrows, the AM is exactly a digraph (see Chapter 5 of this thesis), and hence can be turned into a DSM (Figure 9-2). In the discussion that follows, the AM DSM will be compared to the throttle body assembly DSM made from design documents (Figure 9-3), and the finished throttle body assembly DSM (Figure 8-1).

9.3.1 Associativity Map DSM vs. Throttle Body Assembly DSM from Design Documents

In the process of constructing the throttle body assembly DSM (see chapter 8), a DSM is built based on the information in the design documents. This DSM is in Figure 9-3. The final throttle body assembly DSM is built based on this DSM with additional information obtained by the author from interviewing engineers and attending engineering meetings. In Chapter 8, arguments were made to show the design documents were unable to capture all the interactions between throttle body components. Here, comparison between the DSM from the AM and the DSM from design documents can show us whether the AM is a good tool to capture all of the design knowledge. The following observations can be made between them:

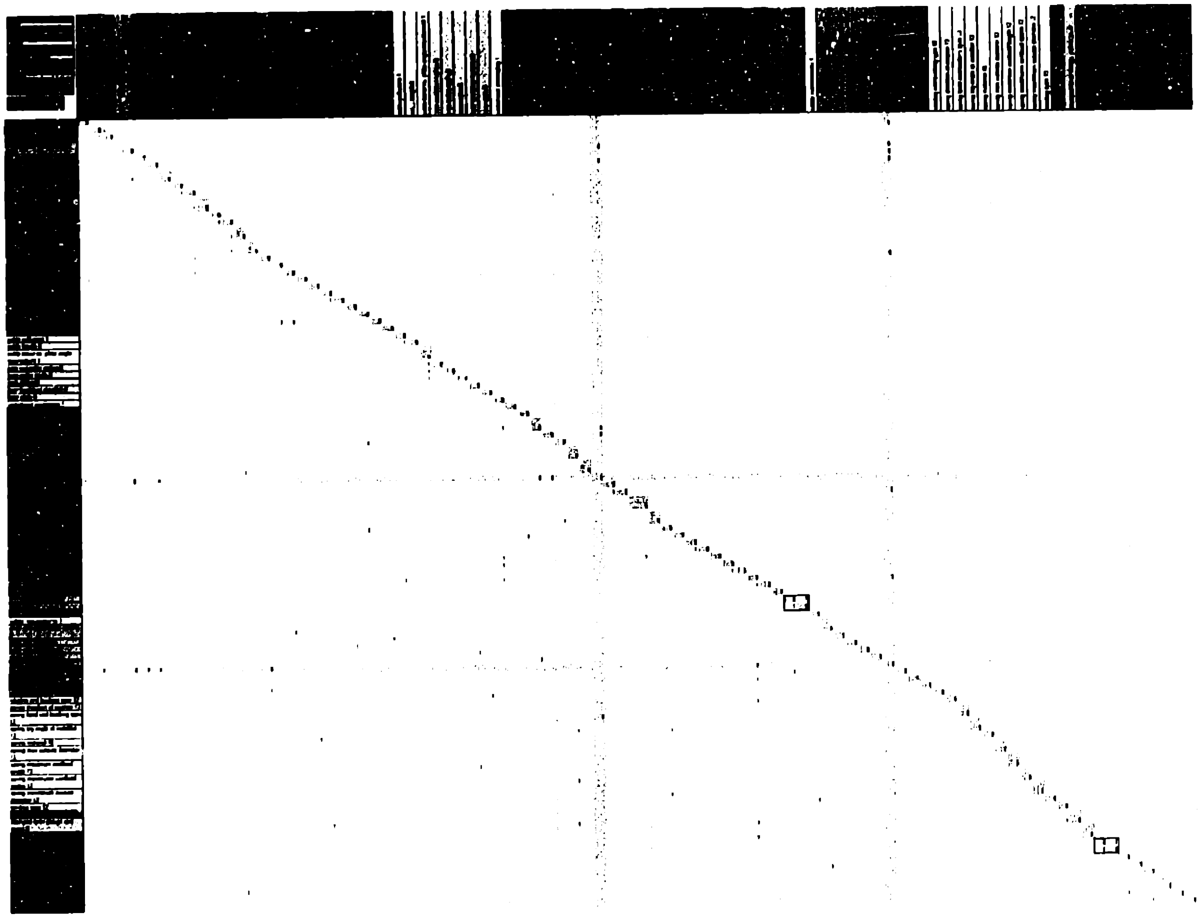


Figure 9-2 Associativity Map DSM

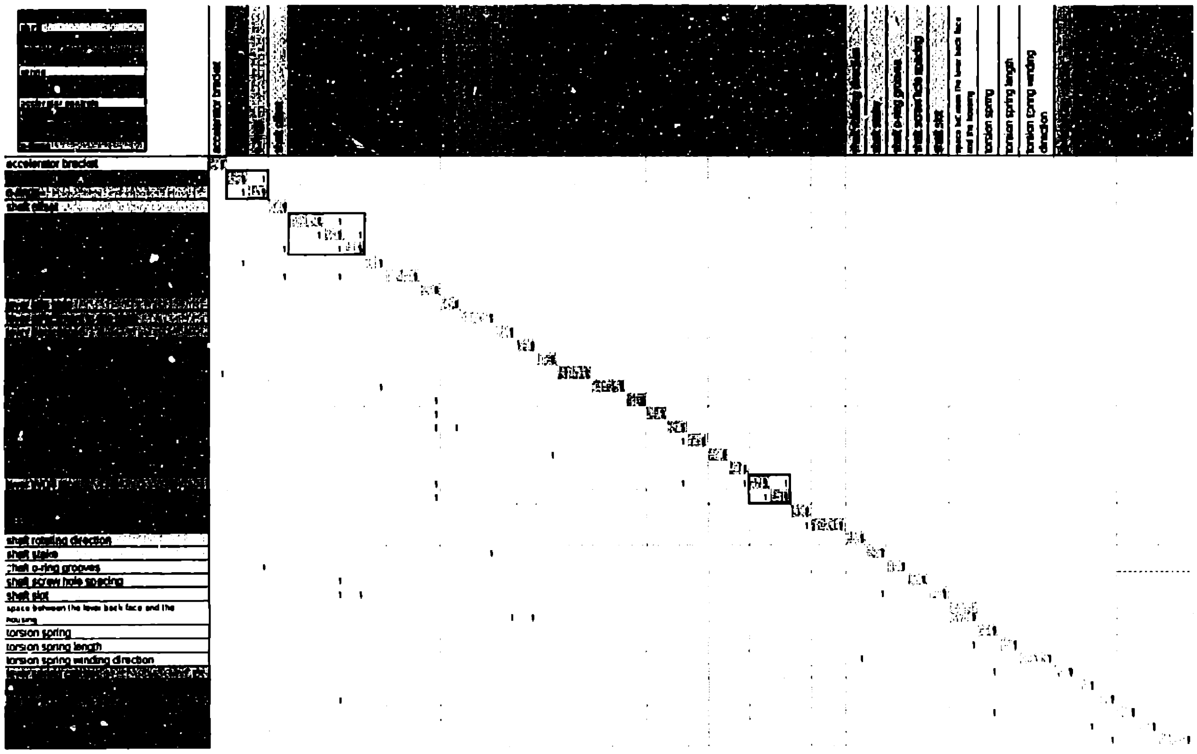


Figure 9-3 Throttle Body Assembly DSM from Design Documents

- The AM DSM does not seem to capture more information than the DSM from design documents. Table 9-1 shows the amount of information contained in each DSM. Although the AM DSM contains more 1's, the AM DSM also contains more system elements. Hence, we cannot say the AM DSM contains more information than the design guide DSM.

  Both DSMs have very few iteration loops, which are also very small. From Figure 8-1, we know the real process is far more complicated. Hence, the AM DSM does not seem to capture more design iteration than the design documents.

	# of system elements	# of 1's
Associativity Map DSM	89	88
Design Document DSM	41	41

Table 9-1 How Much Information is in AM DSM and the DSM from Design Documents

- The two gray bands in the AM DSM (Figure 9-2) are due to the consideration of the machining and assembly lines. The DSM from the design guide does not have these two bands, because the design guide did not emphasize the effect of the manufacturing facilities. Maybe the design guide should have added this information.

- The two DSMs have different iteration loops. After careful study, the author explored two possible causes:

1. The information collected in both methods is not complete. The information causing the iteration loops in the design guide DSM is missing in the AM. The AM contains manufacturing information that the design guide DSM lacks. Hence, this proves that tools can only be good if the inputs are good. The knowledge acquisition method is important.

When both DSMs capture complete information of the system, even if the two DSMs have different levels of details for their system elements, the information flow pattern and the location of the iteration loops in both DSMs should be the same. In other words, as long as the data is complete, DSM should provide a repeatable result for the location and the nature of the iteration loops.

2. The level of details of the system elements in both methods is different. The AM breaks components into features, because the AM tries to include the manufacturing process. The design guide DSM seems to keep things at component level, because when

the author constructed the DSM, manufacturing process of features are regarded as "within" a component, and was not included in the assembly DSM.

In general, the level of details of the system elements can affect the amount of iteration loops in a DSM in two possible ways. The first is that if you break down the elements in the loop further, the iteration loop goes away. An example of such is that when the author first constructed the DSM from design guide, there was an element called "lever WOT and idle stops". This element caused a 5-element iteration loop. When it was broken down into "lever WOT stop" and "lever idle stop", and the iteration loop went away.

The second situation is that the iteration loop is not breakable. For example, the O-ring engineer and the bearing engineer have to decide together what o-rings and bearings they are to use to seal the bearing properly. In this case, the system elements cannot be broken down any further, and iteration cannot be taken away.

However, for the above two DSMs, the iteration loops don't seem to disappear with the increase of the level of details. The iteration loops captured in the design document DSM contain information the AM DSM don't have, and vise versa. Hence, the second situation discussed above applies. The difference in iteration loops are due to the incomplete collection of information in both DSMs.

In conclusion, the AM was unable to capture more information than the design documents. Although it was an attempt to visualize the throttle body design process, better methodology is needed to aid the understanding of the design process and the DE program.

9.3.2 Associativity Map vs. Throttle Body Assembly DSM

From the previous discussion, we have learned that the AM did not capture all the information exchange in the design process completely. In Chapter 8, this thesis has shown that DSM was able to capture the design knowledge accurately and completely. The difference of the results from the two methods may be due to the following reasons:

1. When the TB team initiated their study to the TB using AM, they did not realize the existence of the two-way feedback communications. The first AM the author saw did not have any two-way arrows. After the author questioned the DE engineers, they realized the need, and started to add two-way arrows to the AM.

2. Failing to see the two-way communication may be due to the inexperience of the TB team at mapping out the process. However, as the chosen method, the AM does not promote the two-way thinking. In other words, there isn't a mechanism in the AM that makes the engineers to think in both ways when facing a pair of component relations. DSM, on the contrary, by default force the engineers to think of a pair of system elements in both ways. For example, when engineers go through a row that is titled "A", they think of how the rest of the system elements provide inputs to "A", including how "B" provides inputs to "A". When engineers later go through the row that is titled "B", they naturally think of how "A" provides inputs to "B". Hence, even if the engineers forgot to think of the communication from "A" to "B" earlier on, they will not miss it later on. Therefore, DSM is more helpful to the users.

3. Suppose the TB engineers learned their lessons, and added all the necessary two-way arrows to the AM. Suppose the TB engineers also found a complete list of elements that can represent the system. Then we can make a DSM from the AM. This DSM from the AM will look the same as the DSM in chapter 8, and hold the same amount of information about the design process. In this case, the AM and DSM are just different forms to represent the same information. Yet, compare DSM to the AM, DSM can further provide information on the process flow order, and iterations' occurrence, as well as identifying system elements that are cause large iterations. The AM cannot provide more information

other than the relations between each pair of elements.

4. To look at the problem from another side, A small portion of the complete throttle body assembly DSM (Figure 8-2) was turned into a digraph—an AM. The result is shown in Figure 9-4.

   The AM in figure 9-4 contains only 16 out of the 39 elements in the DSM, and already appears to be quite complicated. Hence the DSM is better at conveying more information visually. In addition, DSM can be partitioned and suggest a sequence of completing tasks based on the information exchange relations. The suggested sequence can be of use for the DE program to promote a better design process for the users. Compare to DSM, it is hard to sort out sequence from AM. Hence, DSM is more helpful for DE programs.

5. From the comparison between the AM DSM and the DSM from the design documents, we know that AM DSM does not contain all the information about the throttle body assembly design, while the DSM in Figure 8-2 does.

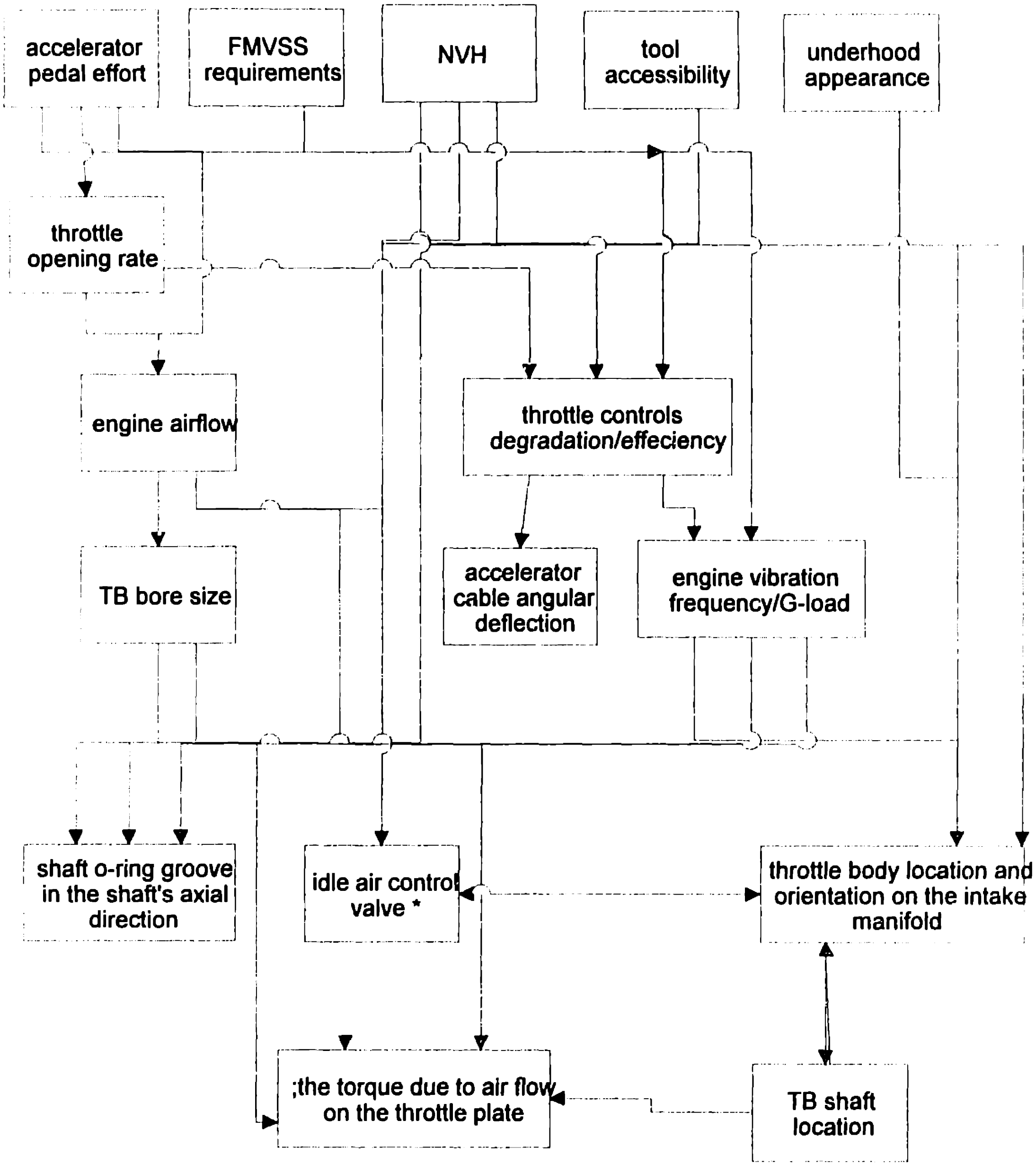


Figure 9-4 Associativity Map Converted from a Small Portion of the Throttle Body Assembly DSM in Figure 8-2

9.4 Chapter Summary

The existence of the Associativity Map proves that Ford engineers have realized the need to find a tool and capture the communication between component feature designs. However, the AM was not an effective method due to both the weakness of the method and the inexperience of the engineers. DSM can do all that AM does and more. Hence, DSM is a better method than AM.

CHAPTER 10 STUDY THROTTLE BODY ASSEMBLY USING DATUM FLOW CHAIN AND COMPARISON TO THE DSM METHOD

10.1 A Brief Introduction to Datum Flow Chain

Datum Flow Chain (DFC) is a methodology to design process for assemblies from top down. Dr. Dan Whitney and his PhD student at the time Dr. Krish Mantripragada first developed this method. Whitney and Mantripragada classified joints between parts into two classes [Mantripragada (1997)]:

- Mates convey dimensional location and constraint from one part to another. When all of a part's mates are complete, it should be constrained in all 6 degree of freedom unless free motion is part of its function.
- Contacts are redundant and provide strength or partial constraint.

Since the final assembly has to deliver the key Characteristics (KC), mates should directly associate with the KCs for the assembly. During the assembly process, the mates should be fastened together first and only then can the contacts be fastened. Whitney and Mantripragada suggest that if the distinctions between mates and contacts "can be expressed carefully and mathematically, then we can construct directed digraph representations for dimensional transfer from mate to mate in a declarative way, providing a basis for synthesizing constraint, location, and tolerance achievement" [Mantripragada (1998)]. This directed graph of mates is called the *Datum Flow Chain* (DFC). The intention of DFC is to express the designer's logical intent concerning how parts are to be related to each other geometrically to deliver the KCs repeatedly.

10.2 Objectives

Mantripragada and Whitney have applied DFC to case studies at Boeing [Mantripragada (1998)]. The work done in this thesis has the following objectives:

- Understand how to make a DFC for the throttle body assembly.
- Compare DFC and DSM to find out the similarity and differences of the two methods. Understand where they are applicable.

10.3 Results and Discussion

A picture of the throttle body is presented in Figure 7-1. The following is a list of steps taken to assemble the throttle body. A DFC is drawn for each step. Solid arrows indicate mates. Dashed lines indicates contact. The number on the lines indicates the degree of freedom constrained by the particular mate/contact. The coordinates on the line indicates which degree of freedom is constrained. Figure 10-1 shows the coordinates system used.

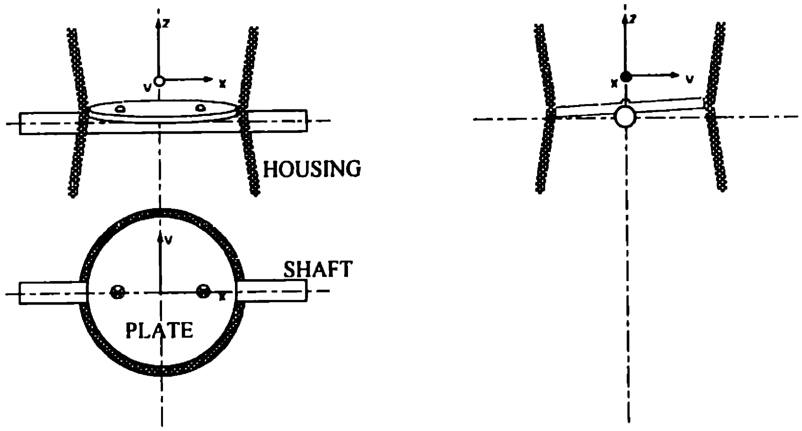


Figure 10-1 Coordinate Convention

Step 1: Press the ball bearings into the housing. Install TRC screw. The corresponding DFC is in figure 10-2.

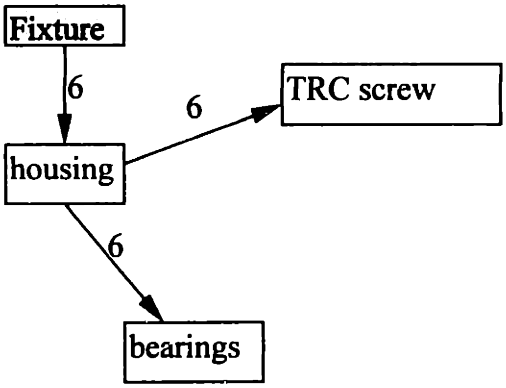


Figure 10-2: DFC for Step 1 in the Throttle Body Assembly Process

Step 2: Step1 and Step2 are actually done in parallel. In this step, the lever is pressed onto the shaft. This operation is called brackering. Figure 10-3 shows the DFC for this process.

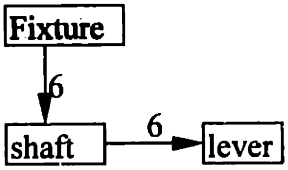


Figure 10-3: DFC for Step 2 in the Throttle Body Assembly Process

Step 3: Assemble spring and plate. Figure 10-4 shows the DFC for this step. In this DFC, the two sub-assemblies created in step 1 and 2 are now treated as rigid bodies. Although the fixture in step 2 is removed, the sub-assembly from step 2 does not change.

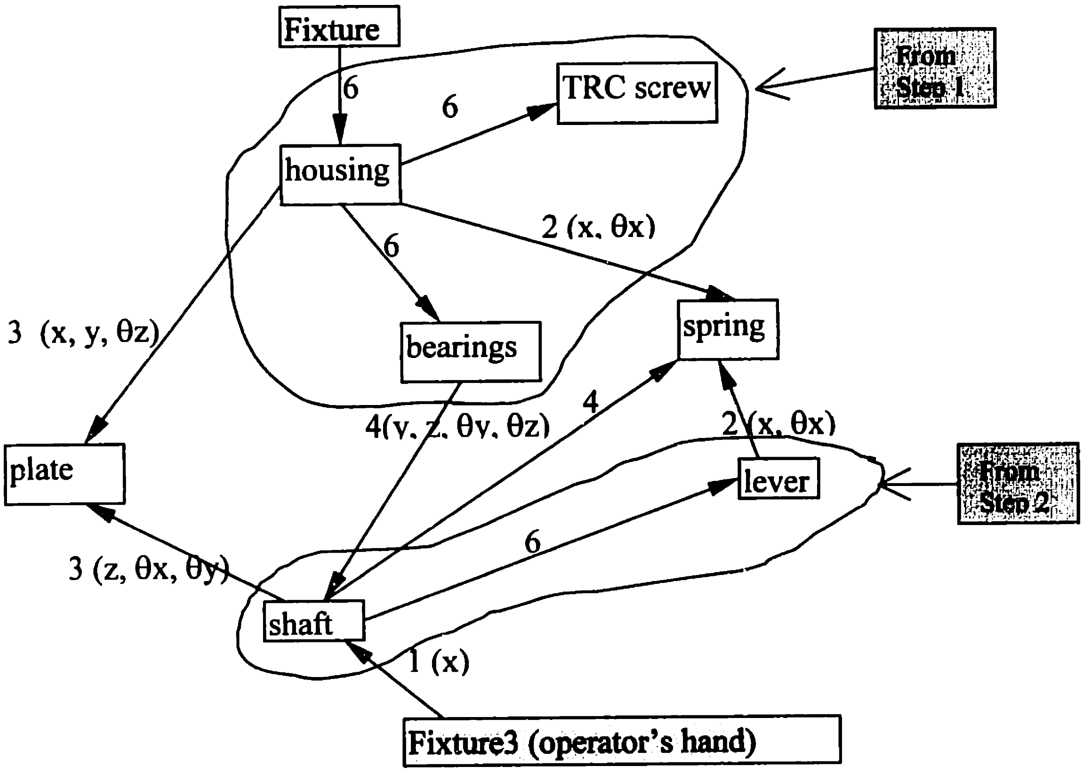


Figure 10-4 DFC for Step 3 in the Throttle Body Assembly Process

Step 4: Assemble throttle position sensor (TPS) and plate screw. The DFC is shown in Figure 10-5. Very interestingly, a two-way arrow exists between the plate and the shaft. The shaft constrains the plate's 5 degrees of freedom. However, the plate also constrains the shaft's X direction of motion by being inside the bore. The existence of the two-way arrows violates the DFC rule—the graph should be acyclic [Mantripragada (1998)]. The DFC theory is not mature enough to deal with this situation yet.

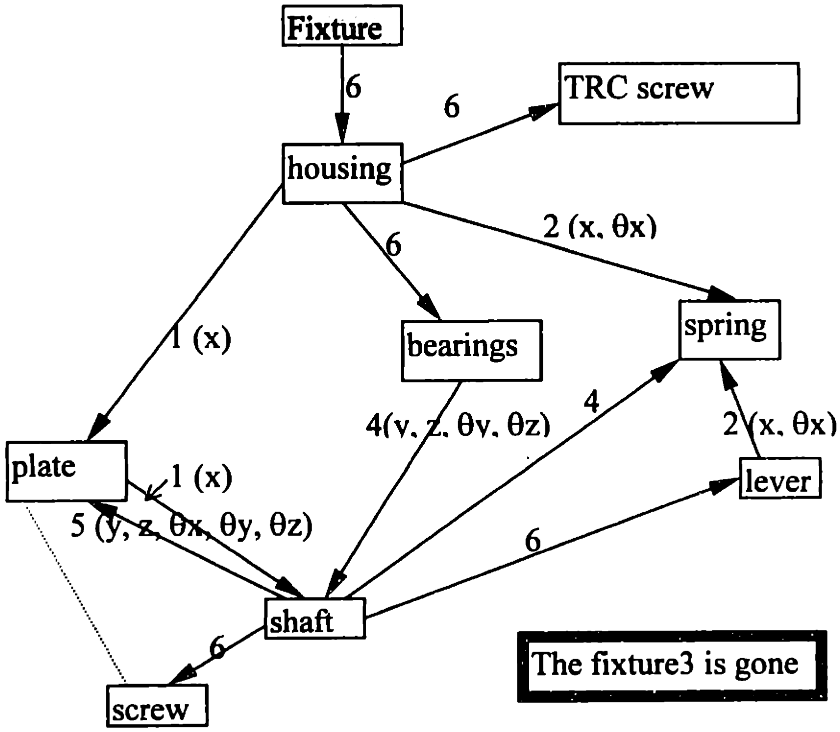


Figure 10-5 DFC for Step 4 in the Throttle Body Assembly Process

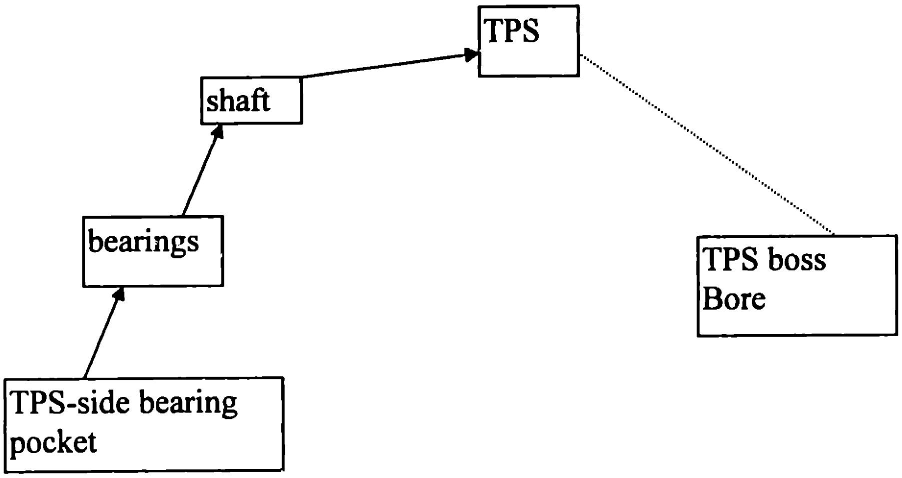
Also, from Figure 10-5, we noticed that by taking off Fixture 3 the relation between the plate and the shaft changed. However, sometimes, taking off the fixture does not change the relations in the DFC such as the case in Step 1 and Step 2. The conclusion is that when the fixture determines six degree of freedom of a part, taking off the fixture does not change the relation in the DFC. When the fixture determines fewer than six degrees of freedom of the part, taking off the fixture will change the relation in the DFC.

10.4 Compare Throttle Body Assembly DSM to DFC

The DFC and DSM for the throttle body assembly seem to contain very different information. The information in the DSM is about the design process and design intent. The information in the DFC is about the assembly and manufacturing method. Although ideally the above two types of information should correspond to each other in order to convey the design intention to the final product, observations from the DFC and DSM show that it is not always the case. Two examples can illustrate the point.

Example 1

Figure 10-6 is a partial DFC to show the relation of bearing pockets and the boss bore. It is very obvious to see that their concentricity needs to be maintained in order to keep the shaft in straight. Hence, the housing machining process needs to accommodate this relation.



TPS boss bore and TPS side bearing pocket have to keep concentricity.

Figure 10-6 Partial DFC to Show the Relation between the TPS Boss and the Bearing Pocket

An interesting observation from this partial DFC is that the bearing pockets decide the shaft's location. This is certainly the case for the assembly process. However, the DSM shows the engineers always decide the location of the shaft first based on many other factors. Then the location of the bearing pockets on the housing is decided by the shaft location. So for the design process, an arrow should be drawn from the shaft to the bearing pockets, which is completely opposite from that in Figure 10-6.

Example 2

Figure 10-7 is a partial DFC. In this figure, we can easily see the relative location of the bearing pockets and the bore is important. In the throttle body assembly, shaft is not mounted along the diameter of the plate but off by a small amount. The purpose is to provide extra plate-closing torque for safety reason. Hence, the bearing pockets have to be also off-centered by the same amount as the mounting holes on the plate so that the plate can fit into the bore without scratching the wall. Plate scratching the wall may cause sticking and binding situation and affect the safety of the vehicle. However, at early stage of the design process, the bore size and the bearing pockets are considered independently for their functionality. The relative location of the two only becomes an issue when coming to assembly process.

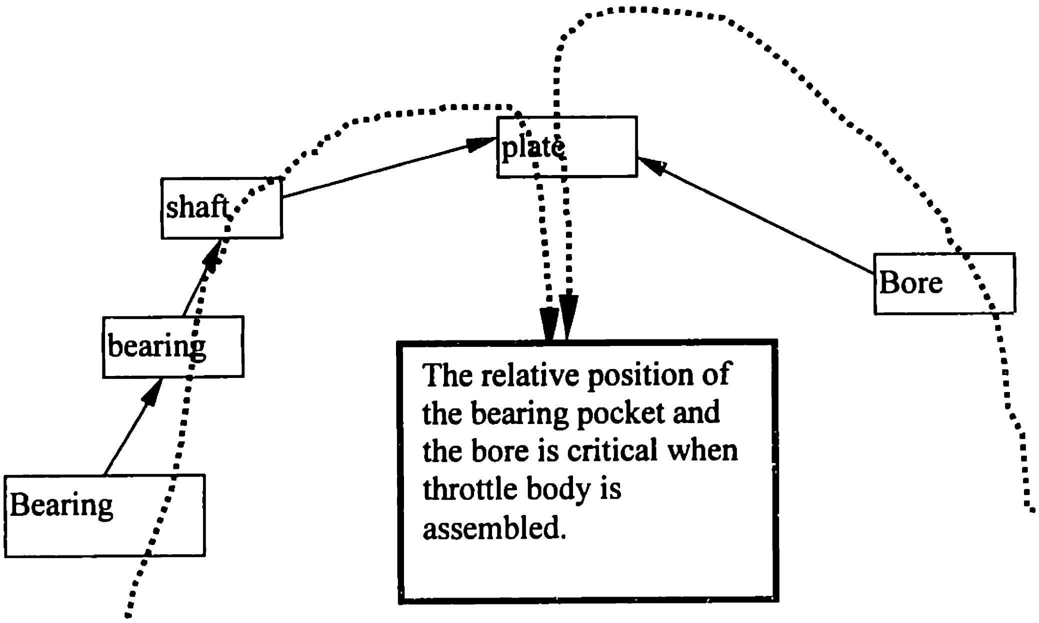


Figure 10-7: A Partial DFC to Show the Relation between the Bore and the Bearing Pocket

From the above two examples, we can see that many issues that the manufacturing engineers care are not of the design engineers concern. DFC is a good tool to identify these issues up-front during the process in order to achieve design for assembly and the KCs. However, DFC is insufficient to model the design process. DSM is good to model the design process and communication between organizations.

A few questions rise regarding DFC. Can DFC correctly represent the design process and design intent? Is it true that the manufacturing process and the design process naturally are concerned of different kinds of issues? How to decide what knowledge that should be possesses by the CAD designer and manufacturing engineers, and what different kind of knowledge should be possessed by design engineers? The two groups should definitely exchange necessary information to make the design better for manufacturing.

10.5 Chapter Summary

DFC is a new yet powerful tool to convey KC and design intentions into the assembly process. DFC concerns very different kinds of information comparing to the DSM. The two can be used together to better understand and improve the product development process.

Part III

Conclusions and Future Research

CHAPTER 11 CONCLUSIONS AND FUTURE RESEARCH

11.1 Conclusions

This thesis research is based upon the broad background of Ford Direct Engineering (DE). In order to improve the efficiency of the design process, DE strives to develop means of understanding the design processes, as well as providing tools for design knowledge reuse. This thesis research successfully used the Design Structure Matrix (DSM) methodology to represent the information transfer during the processes of design vehicle doors and throttle body. The following learning can be concluded from this research:

- The parameter based DSMs suggest the engineering teams to break the traditional organizational boundaries and form proper teams based on the issues in the design process.

- DSMs suggest proper sequence of tasks and team meeting schedule.

- DSMs accept the inevitable iterations in the design processes, and provide a means to visualize and understand them.

- DSMs serve as a browser for engineers to easily find the learned knowledge, as well as a road map for the engineering design processes.

- DSM defines the knowledge ownership. From the case studies, we see some knowledge is maintained within a design team—local knowledge, while some exist across the organization boundaries—non-local knowledge. Non-local knowledge is usually less well understood and maintained than local knowledge.

- DSMs provides the DE programs a means to connect individual application islands. They show what the knowledge needs to be transferred across organization boundaries.

- The current computer programs cannot carry on iterations in the design processes automatically. When an iteration loop exists in a DSM, human interaction is necessary.

- DSM also provides a way to quantify the complexity of the product design alternatives by comparing how much information exchange is needed in each design.

- The throttle body case study revealed that system level knowledge is poorly understood and documented. Most knowledge captured is about parts. Most system level knowledge is in experienced engineers' heads, which leaves the organization when the engineer changes his/her job. DSM proves to be an effective means to capture the system level knowledge and keep the learning with the organization.

- The database accompanying the DSMs also provides a means to find the knowledge that are generic to all variation of a product, as well as knowledge that a specific to a certain design configuration of the product.

- Compared to the Associativity Map (AM), DSM can not only capture the interaction between elements like AM, but also optimize the system and suggest the sequence of tasks to be completed based on the information flow. Furthermore, DSM prompts engineers to consider two way communications, and hence is naturally a better tool to model the design process.

- Compare to Datum Flow Chain (DFC), DSM captures the process by which engineers make design decisions. DFC represents the relation between parts from manufacture and assembly point of view. The two methods view the design process from different angles.

Ford engineers and mangers are excited about seeing their design process captured in such a systematic way. Both the car door and throttle body engineers keep a copy of the research result as a reference for their work. Ford has also named DSM their Best Practice in Product Development. The results of this thesis are also published on Ford Intranet as Lessons learned.

11.2 Future Research

The author has gained tremendous learning throughout the course of this thesis research. Several possible future research directions are discussed here.

11.2.1 More about DSM

DSM is a great tool to model the design processes and information transfers involved. However, as a method, it also faces many challenges. These challenges include data collection, data representation, and incorporating timing in a project.

Knowledge Acquisition

Chapter 5 has discussed the difficulty involved in data collection. Usually, the process involves many people from different areas. The amount of the information involved and the poor understanding of the process across organization boundaries often make the data collection process very time consuming. In chapter 5, the author argued about the strength and weakness of various methods. In addition, MIT SDM student Nader Sabbaghian has developed a web-based tool for collecting data over the Internet [Sabbaghian (1998)]. It will be worth studying how effective the tool is in practice. Also, other forms of tools that can aid the data collection process will also help DSM greatly. After all, DSM is just a modeling tool. The tool can only be as good as the inputs.

Data Representation

In this thesis work, the author has demonstrated that DSM can serve as a browser for the design knowledge. In their current forms, the DSMs and their corresponding databases are in EXCEL spread sheets. A few problems are associated with this state. When using them, the engineers need to first have the database and software on his/her computer. Second, the engineer needs to look up his/her design task in the DSM, and then look up the corresponding information content in the database manually. Third, the large size of the DSM makes it hard to read and publish. A user-friendly interface should be created using Internet and Java programming language. The database can be installed on a web server, and the user can just enter his/her design task. The program will look up the inputs and outputs for that particular task. If the user needs, the program can also search the database for the associated information. However, publishing the information of the design process concerns the security of information in the organization. The author feels this is an area more research and work can be done. In the same article by Nader Sabbaghian, he also used automatic e-mail system to inform the users for action [Sabbaghian (1998)].

Incorporating Timing with DSM

The DSMs in this thesis suggest the sequence of completing design tasks based on the information exchange. However, in a real design situation, information exchange is only part of the cause of the task sequencing. Due to the limited time for product design (time to Job1 in Ford's case), sometimes the tasks have to be completed without all the information inputs. The situation is shown below in figure 11-1. Tasks A, B, C, and D are sequenced according to the information flow. This sequence ensures each task gets all the inputs needed before it starts, and so the result of each task is guaranteed. However, the Gantt Chart shows that the total program time is shorter than the total time needed to complete all tasks.

	A	B	C	D
A	1			
B	1	1		
C		1	1	
D	1	1	1	1

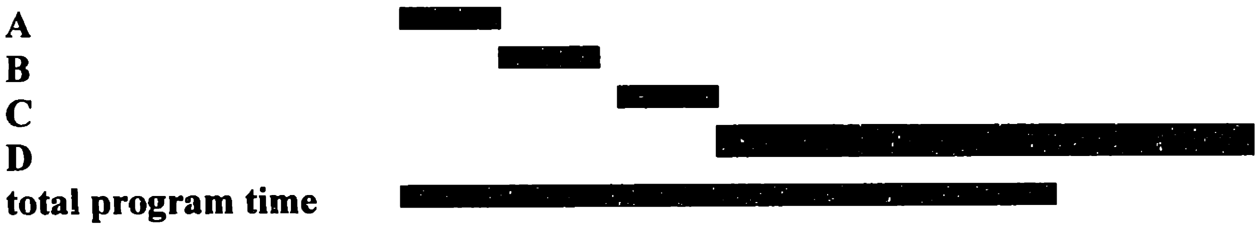


Figure 11-1 DSM and Project Timing

In other words, if C is to wait until all the necessary information is available, we can not meet the program deadline. In this case, if we decide to start C earlier, without waiting for all the information from A, B, and C to be available, we may be able to meet the deadline. The time chart will then look like:

Now the problem with timing is resolved, but does this guarantee a good product? Figure 11-2 shows the DSM corresponding to the last time line. As you can see, by starting D earlier, we are making assumptions on what B and C will input to D. Two 1's appeared above diagonal, indicating potential for later design iteration loops. Although in the time chart, it seems like we can meet the deadline, if B and C turned out to be quite different from what D assumed initially, then D has to start over again based on the new information from B and C. At this point, it is very hard to say if we can still meet the deadline as we have planned and deliver a good quality product.

	A	D	B	C
A	1			
D	1	1	■	■
B	1		1	
C		1		1

Figure 11-2: Re-arranged DSM Based on Meeting the Time Lines

Krishnan and Carrascosa have provided means to deal with this dilemma using the probability of change and the project's sensitivity to change [Krishnan (1997), Carrascosa (1998)]. The "assigning dependency value" method discussed in Chapter 5 and 6 of this thesis can also help this situation. The author believes more research is worth carrying out to better understand the decision making on sequencing tasks and to test the methods developed by the above researchers.

11.2.2 Discussions on What the Knowledge Based Engineering Should Provide

Direct Engineering (DE) is a type of Knowledge Based Engineering. It tries to capture the design knowledge for reuse in order to improve the speed and efficiency of the product development processes. Observations during the thesis research work have raised many interesting questions. These questions are worth further research and studying. They are summarized here by five main topics:

1. Is DE a copy of the current processes? Should it be?
2. What knowledge should be in DE?
3. How much knowledge is there in a product design?
4. How many kinds of knowledge are there? How to best utilize various kinds of knowledge?
5. The development process of DE.

The rest of this section provides detailed discussions on each topic.

Is DE a Copy of the Current Processes? Should it be?

During the process of programming the throttle body DE application, the DE engineers collected design rules from engineers. These rules drive the logic in the DE program. Besides extracting design rules from the current design practices, the DE programs also closely follow the current design processes hoping the organization can easily accept the program because it does not have to go through a learning curve of the tool. Hence, it seems that DE programs are more or less electronic copies of the current design practice. They save product development time in data transfer, phone calls, and communications, but they are no more than a more organized copy of the current practice. How can we improve the current process if the engineers are to use the computer program without even having to think about the existing process too much anymore?

Since the development of any DE and KBE costs companies a lot of time and the resource, the author believes that any DE or KBE application should strive to provide improvements to the existing process. However, suggesting new design processes put DE in a position to face the war of defending their view of a better process than the existing one. Implementation of the program may also become harder. The author thinks that the DSM work in this thesis may be an effective tool to deal with the above situation. The process of constructing the DSM can raise the awareness in engineering teams about the communication at system level and across organizational boundaries. The partitioned DSM shows how to best prioritize tasks based on the information transfer pattern, and hence suggest a convincing new process. DSMs can also show the iteration loops and areas issues rise, and hence prompt engineers to re-think of the existing process and seek improvements. Therefore, employing DSMs as the first step of DE development could be an effective resolution.

What Knowledge Should be in DE?

DE programs collect the design knowledge from design engineers and record them in the form of design rules. The DE programs also attempt to actively verify engineers' design and make design decisions based on the logic behind these design rules. Furthermore, the DE programs try to incorporate the process of designing a product into the software, as discussed in the previous point. In short, the current DE programs intend to not only incorporate design knowledge, but also the process knowledge needed to make engineering design decisions.

However, the author observed during the research period that the engineers were particularly interested in one part of the throttle DE program, which was the math calculation for the cam design. The rest of the throttle body DE program had a difficult time to attract the engineers. This observation seems to tell us that the engineers welcome the design knowledge, such as that the Cam calculation captures. However, they are not convinced that the process knowledge in the DE program is necessary. Furthermore, the engineers do not feel comfortable letting a computer make the design decision for them. They are more comfortable to let the computer programs do busy work.

The author believes that proposing a better process in the computer should ultimately improve the productivity of the design process. However, the engineers may not feel comfortable loosing so much control over the design decision making process at one time. The DSM work in this thesis may be an intermediate step between now and the final deployment of DE. DSM can serve as a passive knowledge browser for the engineers without making decisions for them. Partitioned DSM can also show the engineers the improved design process. After engineers gain faith in the knowledge on design and process in the DSM, they may be more acceptant to the DE programs.

How Much Knowledge is There in a Product Design

Table 7-1 contains a series of number that indicates the size and complexity of the DSMs. This provides a relative measure to the amount of information being exchanged in the throttle body design at system level. Chapter 8 also provided the similar numbers for the throttle body assembly. Throttle body is a simple subsystem compare to many others in the car. However, the numbers mentioned above are already very impressive. It is hard to imagine how much knowledge will be involved in the entire vehicle design. Therefore, it is necessary to seek means to deal with the challenge of the amount of the knowledge.

In fact, the amount of knowledge is a relative measure with respect to the users of the knowledge. For example, engineers do not need explanations to Newton's Second Law, while a history major may need more explanation to the law. The amount of knowledge needs to be transferred to the history major is a lot more than to the engineer. Hence, one solution to the above challenge is to design the DE and KBE applications with the assumption that the users already possess a defined amount of knowledge. Meanwhile, anyone who is to use the DE or KBE application will go through a training program that will ensure he/she possess the defined amount of knowledge. The best assumption on the users' knowledge may be assuming they have a Bachelor's Degree in engineering. Then the training program can teach them the specifics of the particular DE/KBE application. In this picture, the DSM work can serve as a database with a browser to teach the new engineers about the De/KBE process.

DSM can also be used to bring new engineers up to speed in the current design process. Right now at Ford, when a new engineer arrives, he/she usually reads design documents and asks other engineers in the team to learn about the design process. Since there isn't a defined process of learning for the new engineers, it usually takes a while before new engineers can get up to speed. Using DSMs to illustrate the knowledge and process to the engineers can shorten this learning period.

How Many Kinds of Design Knowledge are there? How to Best Utilize Various Kinds of Knowledge?

In a product design process, many types of knowledge exist for different purposes. For example, the knowledge concerning the cam geometry is design knowledge, and is needed at the early stage of the engineering design process. The knowledge concerning the injection molding draft angle is manufacturing knowledge, and is needed at the late stage of the design process when the CAD designers draw the detailed dimensions of the parts. Currently, Ford collects all the knowledge and put them in repositories such as design guides. It would be helpful for both the engineer's work and the DE programming to understand when what kind of knowledge is needed.

The Development Process of DE

Product design process can be viewed as a weave pattern shown in Figure 11-3. The design of a specific product consists of various aspects of the design such as Design for Manufacturing (DFM) and Design for Assembly (DFA) rules, and can be represented by the vertical lines in Figure 11-3. The horizontal lines in Figure 11-3 show that each aspect of the design applies to all products.

Direct Engineering develops its applications for specific products such as throttle body, air hose, etc. It takes the vertical lines in Figure 11-3 and goes through all aspects of the design every time it develops a DE application. Another group at Visteon—the Knowledge Based Engineering Group (KBE)—takes a different approach. KBE takes the horizontal lines in Figure 11-3 that represent different aspects of the design, and develops applications that can be used across all product lines such as the Design for Casting software.

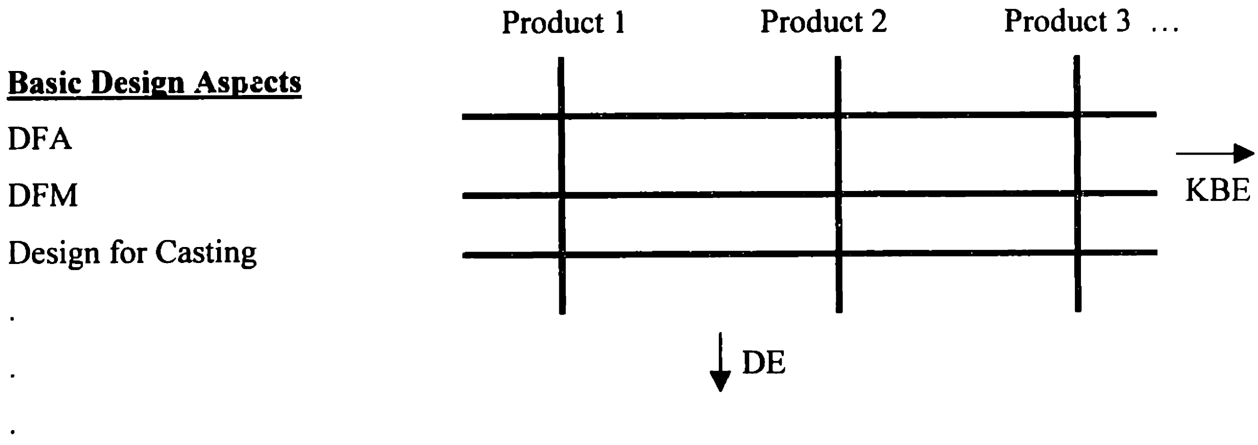


Figure 11-3 Product Design Weave

KBE's approach may be better than that DE takes. Taking the horizontal lines can help the engineers extract the rules of one aspect of the design at a time. The developed KBE applications can then be used for all products. The KBE applications' impact on the business is large, and the developed rules are generic and re-useable. Eventually, when all aspects of the design are well understood and made into KBE applications, the final set of KBE applications can be used on any product in the future. Hence they will be very flexible. DE's approach can produce a program that takes care of all aspect of a product at a time. In the long run, DE's approach will repeat its work on each aspect of the design for every product application it encounters, and hence does not appear very efficient. Furthermore, DE's approach also produces applications that are not flexible to design changes. Hence, the DE team may want to re-look at their approach and find ways to overcome the shortcomings. Maybe DE team can form groups to focus on various aspects of the design and learn simultaneously when applications for various products are developed.

11.2.3 Impact of the Knowledge Based Engineering Systems on the Organization

The example in 11.2.2 about the cam portion in the DE program has clearly indicated that a any KBE system will affect the existing way of doing things. People will accept the KBE system if the effect is good for them. However, they tend to hesitate and even reject the program if they are uncertain about it. The throttle body DE program is a good example.

Since the KBE system is going to take the work away from some people, resistance will exist. For example, the CAD designers at Ford think the DE program is going to take away their jobs, and took a defensive attitude towards the DE programs.

In addition, taking the knowledge away from people's minds and making it available to everyone make some people think they are going to lose their jobs once they lose the privilege to the knowledge. Hence resistance arises.

Another side of the story is that when computers automate the knowledge and process, engineers may lose the opportunity of learning from practice. The design process is also dependent on the performance of a few computers at critical moments. In addition, the workload shift also needs to be justified for business profit purpose. Chapter 4 has more detailed discussions on the above points. Hence it is very worth studying the impact of the KBE system on the organization, as well as how the response of the organization shapes the development of the KBE.

CHAPTER 12 REFERENCES

Baya, Vinod, Jody Gevins, Catherine Baudin, Ade Mabogunje, George Toye, and Larry Leifer. "An Experimental Study of Design Information Reuse." ASME Design Theory and Methodology. DE-vol. 42, 1992, pp141-147.

Carrascosa, Maria, Steven Eppinger, and Daniel Whitney. "Using the Design Structure Matrix to Estimate Product Development Time." DAC98-6013, Proceedings of the 1998 ASME Design Engineering Technical Conference. September 13-16, 1998 Atlanta, Georgia.

Chen, Aihua, Brian McGinni, and David Ullman. "Design History Knowledge Representation and Its Basic computer Implementation." American Society of Mechanical Engineers Design Engineering Division (Publication) DE Design Theory and Methodology. DTM '90 1990 ASME Design Technical Conferences - Presented at the 2nd International Conference on Design Theory and Methodology Sep 16-19 1990 v 27 1990 Chicago, IL.

Chovan, John D. and Manjula B. Waldron. "The Use of Function and Form When Reading Mechanical Engineering Drawings by Experts and Non-experts." American Society of Mechanical Engineers, Design Engineering Division (Publication) DE Design Theory and Methodology - DTM '90 1990 ASME Design Technical Conferences - Presented at the 2nd International Conference on Design Theory and Methodology Sep 16-19 1990 v 27 1990 Chicago, IL.

Eppinger, Steven D., Daniel E. Whitney, Robert P. Smith, and David A. Gebala. "A Model-based Method for Organizing Tasks in Product Development." Research in Engineering Design, 1994, vol. 6, pp1-13.

Ford Design Institute. Systems Engineering Fundamentals, Ver. 2, 1998.

Hounshell, David A. "Automation, Transfer Machinery, and Mass Production in the U.S. Automobile Industry in the Post-World War II Era." Presentation at the Annual Meeting of the Society for the History of Technology, London, United Kingdom, August 1996.

Judson, Jared. Integrating Supplier Designed Components into a Semi-Automated product Development Environment. MIT Master thesis. 1998.

Khadilkar, Dileep V. and Larry A Stauffer. "An Experimental Evaluation of Design Information Reuse During Conceptual Design." Journal of Engineering Design. Vol. 7, No. 4, 1996, pp331-339.

Krishnan, Viswanathan, Steven Eppinger, and Daniel Whitney. "A Model-based Framework to Overlap Product Development Activities." Management Science v 43 n 4 Apr 1997.

Kuffer, Tom A. and David G Ullman. "The Information Requests of Mechanical Design Engineers." Proceedings of the Design Theory and Methodology Conference. DE-Vol. 27 (New York, ASME), pp167-174, 1990.

Ledet, W. P. and D. M. Himmelblau. "Decomposition Procedures for the Solving of Large Scale Systems." Advances in Chemical Engineering. Vol. 8, 1970, pp. 185-254.

Mantripragada, Krish, and Daniel Whitney. Datum Flow Chain, Theory and Applications. Presentation. 1997.

Mantripragada, R. and Whitney, D. E., "The Datum Flow Chain," Research in Engineering Design, v 10, no 1. 1998.

McCord, Kent R. and Steven D. Eppinger. "Managing the Integration Problem in Concurrent Engineering," Working Paper, MIT Sloan School of Management. August, 1993.

Pimmler, Thomas U. and Steven D. Eppinger. "Integration Analysis of Product Decomposition." ASME Design Theory and Methodology Conference, Minneapolis, MN, September 1994, pp1-9.

Rochlin, Gene I. Trapped in the Net—The Unanticipated Consequences of Computerization. Princeton University Press. Princeton, New Jersey. 1997.

Rogers, J. L. DeMAID Program. Http://fmad-www.larc.nasa.gov/mdob/users/rogers/Current_research/DeMAID.html. August 1997.

Rogers, J. L. "DeMAID: A Design Manager's Aid for Intelligent Decomposition User's Guide." NASA Technical Memorandum, 101575, March 1989.

Rogers, J. L. and S. L. Padula. "An intelligent Advisor for the Design Manger." NASA Technical Memorandum, 101558, February 1989.

Sabbaghian, Nader. "Product Development Process Capture and Display using Web-based Technologies." IEEE International Conference on Systems Management & Cybernetics. 1998, pp.2664-2669.

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

Stauffer, L. A., D. G. Ullman, and T. G. Dietterich. "Protocol Analysis of Mechanical Engineers Design." International Conference on Engineering Design, ICED 87, Boston, MA 1987.

Steward, Donald V. "Partitioning and Tearing Systems of Equations." SIAM Journal of Numerical Analysis. Serial B, Vol. 2, No. 2, 1965, pp. 345-365.

Steward, Donald V. "The Design Structure System: A method for Managing the Design of Complex Systems." IEEE Transactions on Engineering Management. Vol. EM-28, No. 3. August 1981.

Warfield, John N. "Binary Matrices in System Modeling." IEEE Transactions on Systems, Man, and Cybernetics. Vol. SMC-3, No. 5, September 1973.

Warfield, John N. Societal systems : planning, policy and complexity. A Wiley-Interscience publication. New York, NY. 1976.

Warfield, John N. "Twenty Laws of Complexity: Science Applicable in Organizations." Systems Research and Behavioral Science, 1998.

Warfield, John N. "The Magic Number of Three—Plus or Minus Zero." Cybernetic and Systems 19, pp339-358. 1998.

Whitney, D. E. "The Potential for Assembly Modeling in Product Development and Manufacturing." 1996. http://web.mit.edu/ctpid/www/whitney/papers.html. Oct. 23, 1998.

Wiest J. D. and F. K. Levy. A Management Guide to PERT/CPM, Prentice-Hall, Englewood Cliffs, New Jersey, 2nd Ed., 1977.

APPENDIX A

A COMPUTER PROGRAM FOR BUILDING AND PARTITIONING THE DESIGN STRUCTURE MATRIX

This program is developed using Microsoft EXCEL Visual BASIC. Inputs have to be entered into the proper cells in the worksheets. The user instruction, user interface, as well as the actual program codes are recorded here.

User Instruction

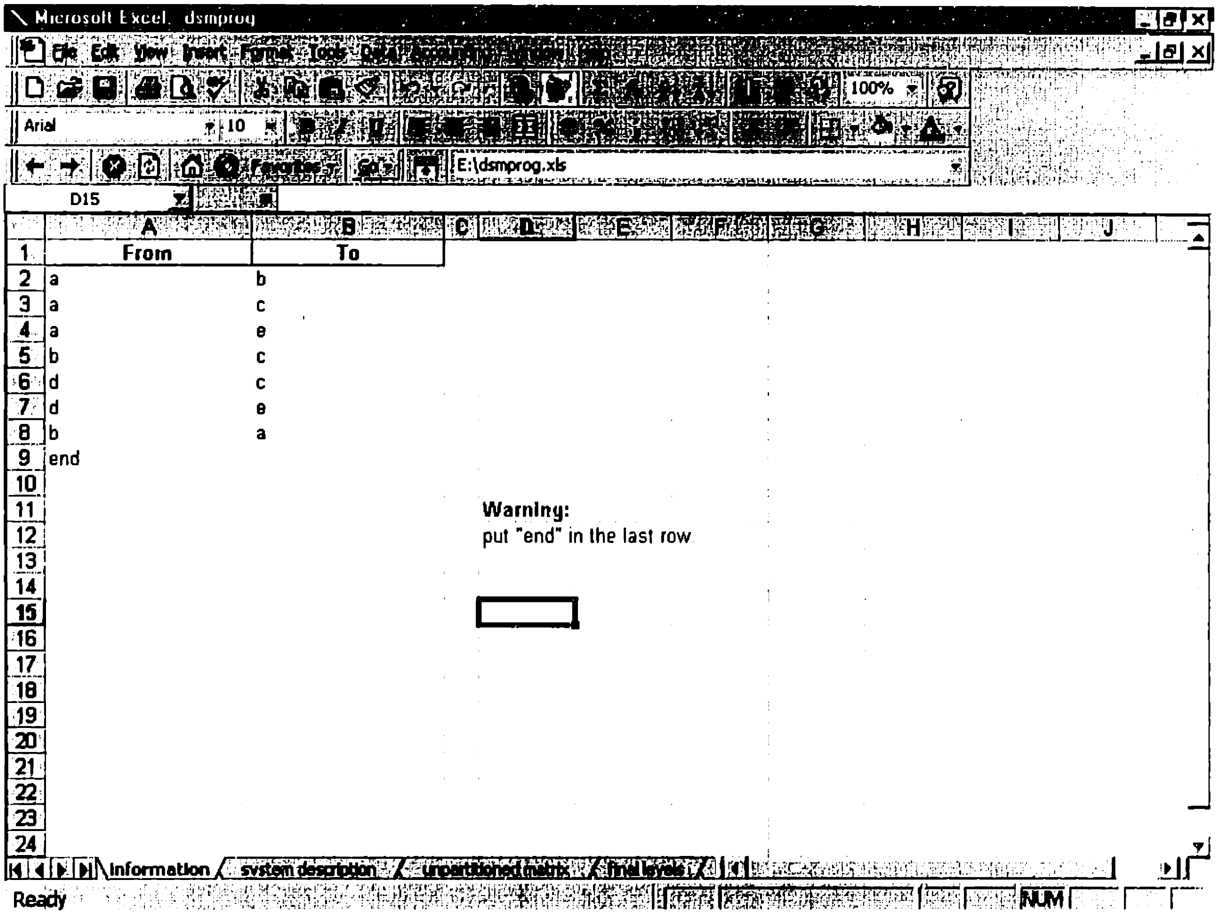
1. In the "**information**" sheet, enter all the existing relations. The elements that provide input goes into the "from" column, and the elements take the inputs go into the "to" column. At the end of the "From" column, type in "end".
2. Go to sheet "**system description**". In Cell 1B, type in the total number of system parameters. Starting from Cell 4A, type in the title of system parameters.
3. Go to sheet "**unpartitioned matrix**" and clear contents, except the ones in the gray cells.
4. Go to sheet "**final levels**" and clear contents except the gray cells.
5. Go to sheet "**unpartitioned matrix**". Click the button "build unpartitioned matrix and compute levels".

6. Based on the result in the "**final levels**" sheet, rearrange the unpartitioned matrix's rows and columns, and build the partitioned matrix.

The program may take a while if you have many system parameters. It took 2 minutes for 40 system parameters.

Sample DSM Partitioning Using the Program

This example contains five system elements, which are *a* through *e*. Their interactions are recorded in the first sheet "information" (see Figure A-1)

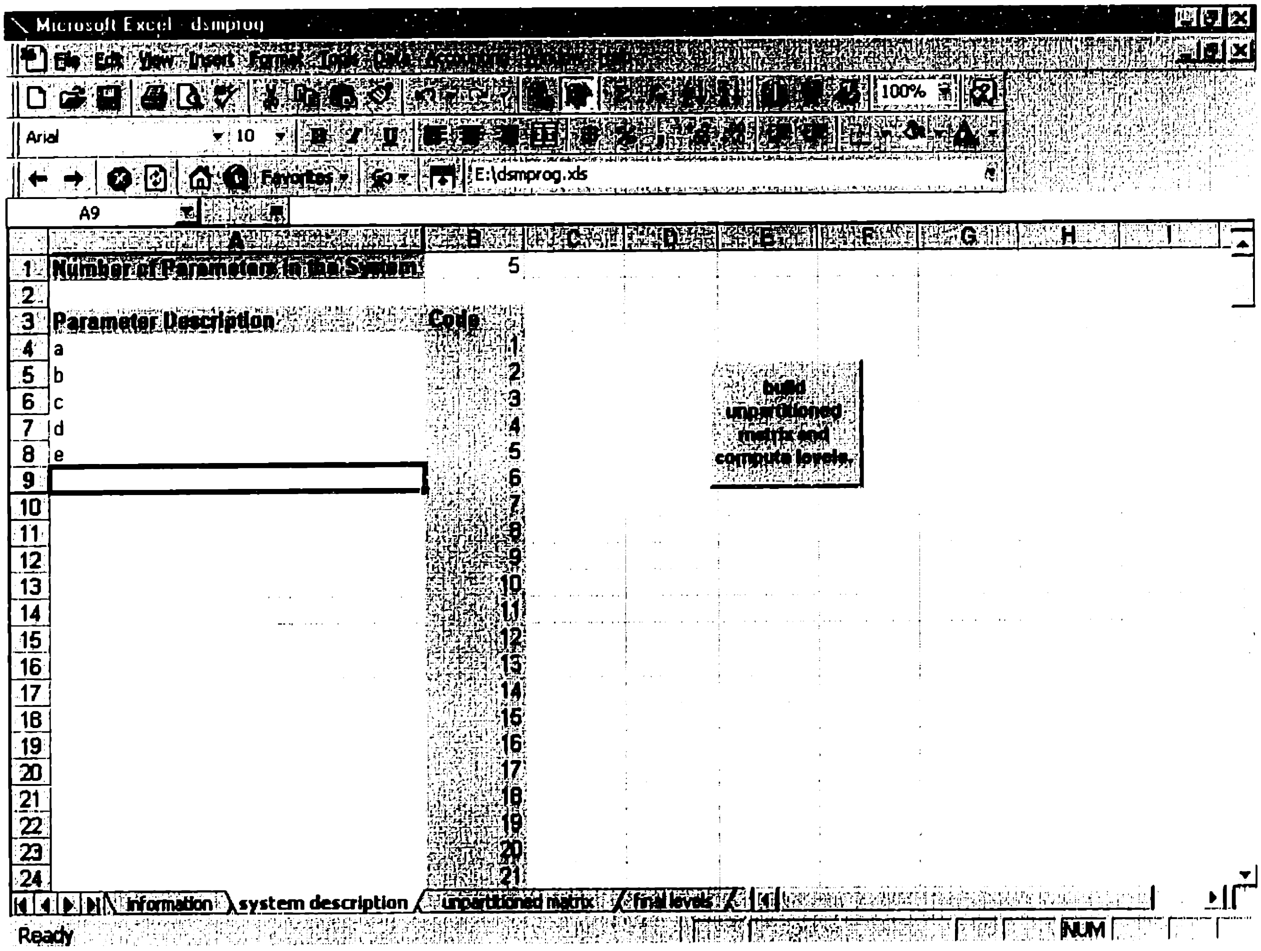


From	To
a	b
a	c
a	e
b	c
d	c
d	e
b	a
end	

Warning:
put "end" in the last row

Figure A-1 Sheet "Information"

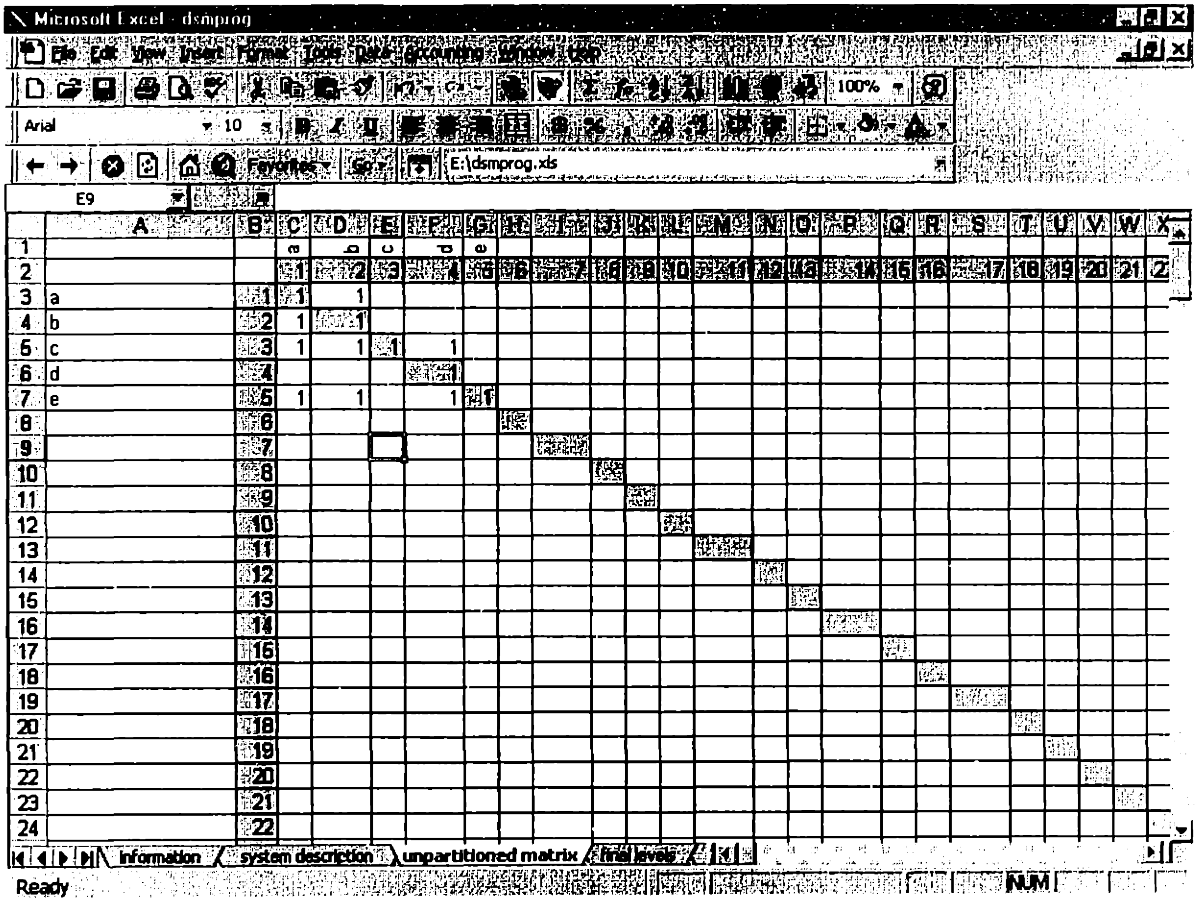
In the second sheet, the information regarding the 5 elements is entered. Then the user may click the bottom to the right of the sheet (see Figure A-2).



	A	B
1	Number of Parameters in the System	5
2		
3	Parameter Description	Code
4	a	1
5	b	2
6	c	3
7	d	4
8	e	5
9		6
10		7
11		8
12		9
13		10
14		11
15		12
16		13
17		14
18		15
19		16
20		17
21		18
22		19
23		20
24		21

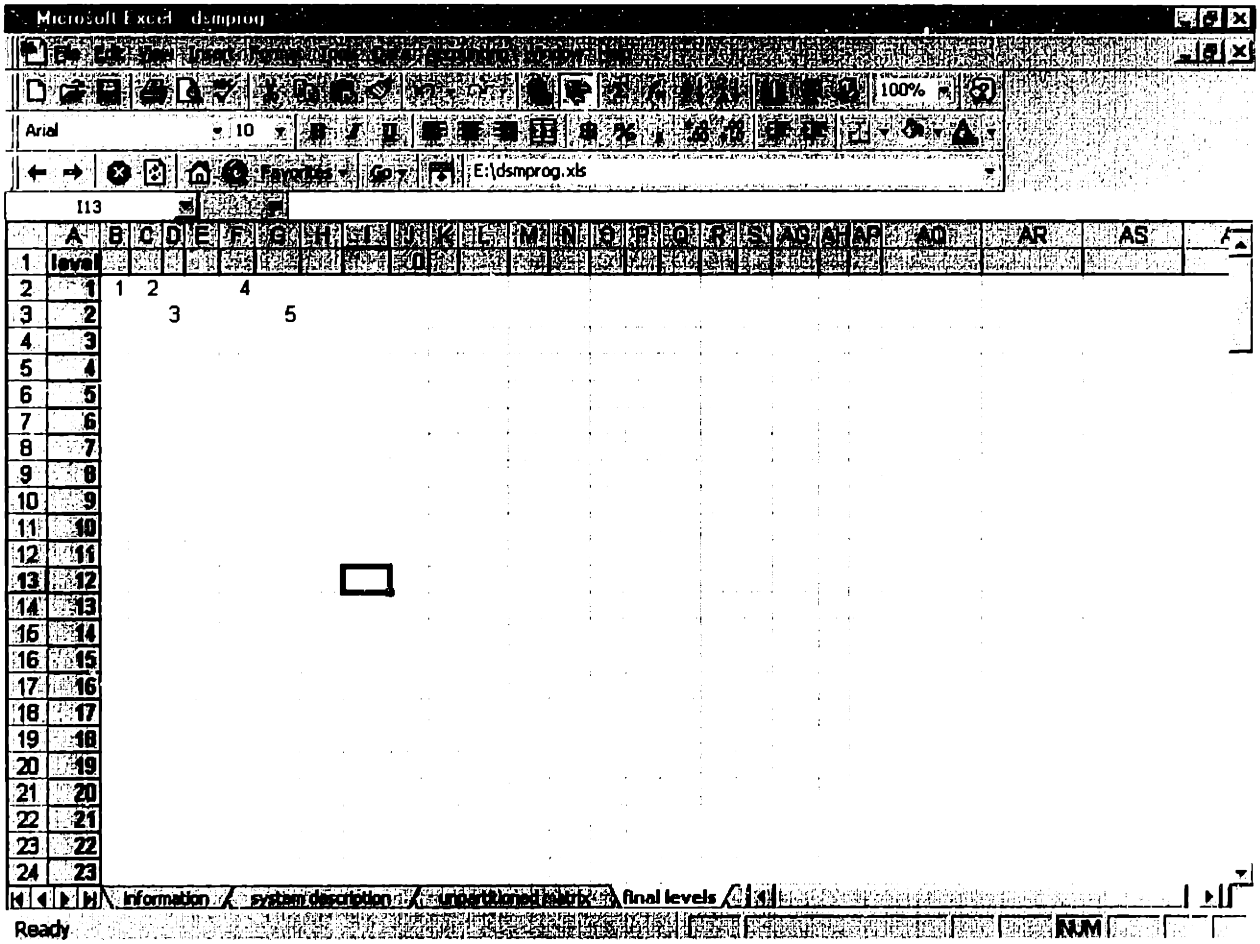
Figure A-2 Sheet "System Description"

The program then first build the unpartitioned matrix in the sheet "unpartitioned matrix" (see Figure A-3). The partition results are in the sheet "final levels" (see Figure A-4). The form of the partitioned matrix can be built based on the unpartitioned matrix and the partitioning result (Figure A-5).



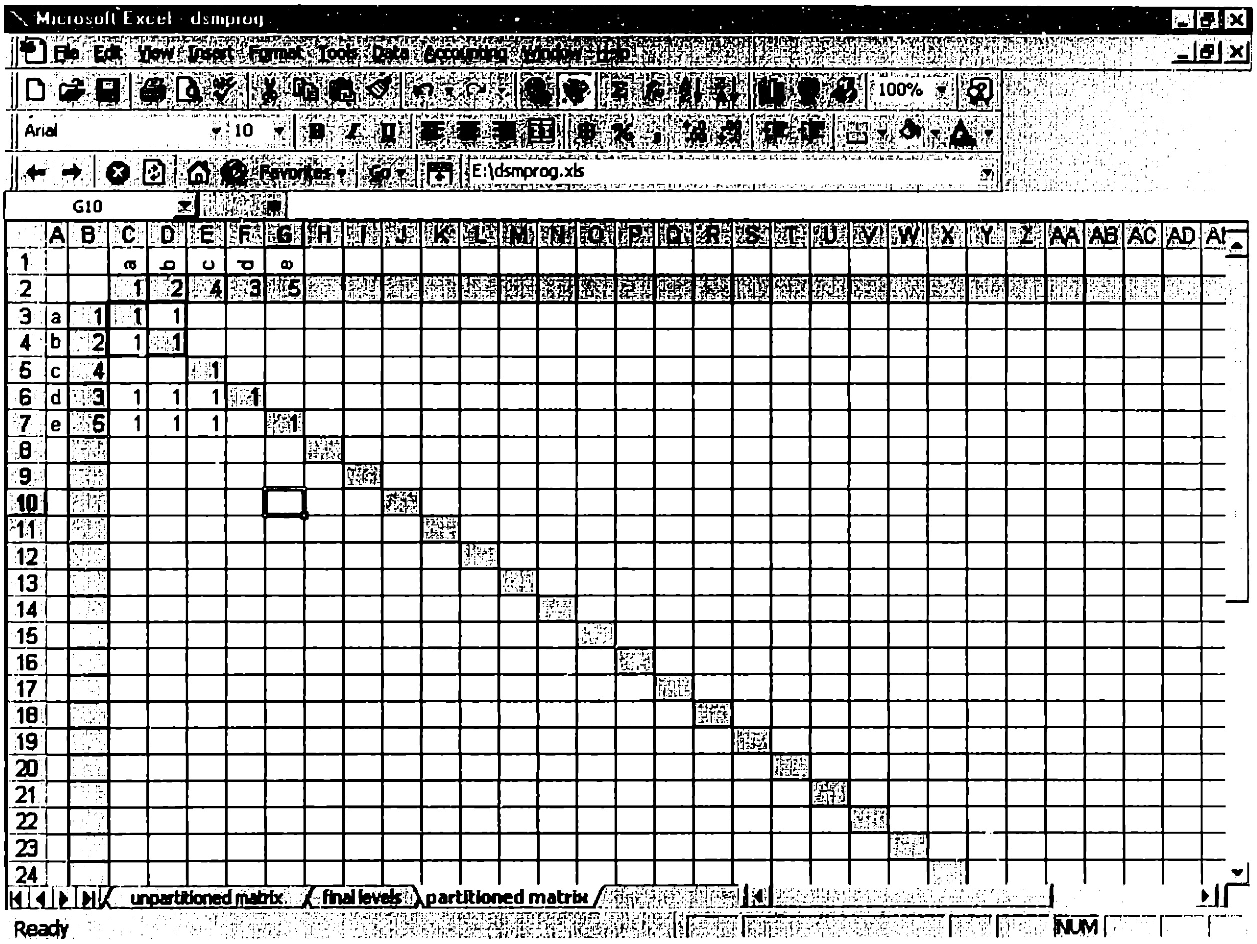
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1			a	b	c	d	e																	
2			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	2
3	a	1	1	1																				
4	b	2	1	1																				
5	c	3	1	1	1	1																		
6	d	4				1																		
7	e	5	1	1		1	1																	
8		6																						
9		7																						
10		8																						
11		9																						
12		10																						
13		11																						
14		12																						
15		13																						
16		14																						
17		15																						
18		16																						
19		17																						
20		18																						
21		19																						
22		20																						
23		21																						
24		22																						

Figure A-3 Sheet "Unpartitioned Matrix"



	A	B	C	D	E	F	G
1	level						
2	1	1	2			4	
3	2			3			5
4	3						
5	4						
6	5						
7	6						
8	7						
9	8						
10	9						
11	10						
12	11						
13	12						
14	13						
15	14						
16	15						
17	16						
18	17						
19	18						
20	19						
21	20						
22	21						
23	22						
24	23						

Figure A-4 Sheet "Final Levels"



	A	B	C	D	E	F	G
1			a	b	c	d	e
2			1	2	4	3	5
3	a	1	1	1			
4	b	2	1	1			
5	c	4			1		
6	d	3	1	1	1	1	
7	e	6	1	1	1		1

Figure A-5 Sheet "Partitioned Matrix"

Program Codes

```
'
' build_unpartitioned_matrix Macro
' Macro recorded 8/13/98 by Qi Dong
'
'


Public i As Integer 'i records the total number of system elements in the matrix
Public parameter(150) As String 'parameter array records all the system elements in the
matrix
'assume the system parameter won't be more than 150
Public R(150, 150) As Integer
```

```
Sub build_unpartitioned_matrix()
  Dim judge, k, m, judge1 As Integer
  Dim m1, m2 As Integer
  Dim counter, counter1 As Integer

   'find the number of parameters and assign the value to i
   Sheets("system description").Select
   Cells(1, 2).Select
   i = ActiveCell.Value
   Cells(4, 1).Select

   'assigning the system element titles to the parameter array
   For counter = 1 To i
      parameter(counter) = ActiveCell.Value
      Cells(4 + counter, 1).Select
   Next counter

   'set initial values of R(m,n)
   For counter = 1 To i
      For counter1 = 1 To i
         If counter = counter1 Then
            R(counter, counter1) = 1
         Else
            R(counter, counter1) = 0
         End If
      Next counter1
   Next counter

   'assigning the value of parameter relation to R(m,n)
   Sheets("information").Select
   judge = 1
   k = 2
   Do Until judge = 0
      judge1 = 1
      m = 1
      Cells(k, 1).Select
      Do Until judge1 = 0
         If Cells(k, 1).Value = parameter(m) Then
            judge1 = 0
         ElseIf m > i Then
            judge1 = 0
         Else
            m = m + 1
```

```
         End If
      Loop
      m1 = m
      Cells(k, 2).Select
      judge1 = 1
      m = 1
      Do Until judge1 = 0
         If Cells(k, 2).Value = parameter(m) Then
            judge1 = 0
         ElseIf m > i Then
            judge1 = 0
         Else
            m = m + 1
         End If
      Loop
      m2 = m
      R(m2, m1) = 1
      k = k + 1
      If Cells(k, 1).Value = "end" Then
         judge = 0
      End If
   Loop

'output un-partitioned matrix
Sheets("unpartitioned matrix").Select

'fill in matrix titles
For counter = 3 To i + 2
   Cells(counter, 1).Value = parameter(counter - 2)
   Cells(1, counter).Value = parameter(counter - 2)
Next counter

'fill in R(m,n) values in the matrix
   For counter = 1 To i
      For counter1 = 1 To i
         If R(counter, counter1) = 1 Then
            Cells(counter + 2, counter1 + 2).Value = R(counter, counter1)
         End If
      Next counter1
   Next counter

reachability_matrix
partition
```

```
level1

End Sub


'
' reachability_matrix Macro
' Macro recorded 8/17/98 by Qi Dong
'
'
Public reach(150, 150) As Integer 'reach contains the elements in the reachability matrix

Sub reachability_matrix()

   Dim counter, counter1, counter2 As Integer
   Dim temp1(150, 150), temp2(150, 150) As Integer

   For counter = 1 To i
      For counter1 = 1 To i
         temp2(counter, counter1) = R(counter, counter1)
      Next counter1
   Next counter

   'start calculating the reachability matrix
   For counter = 1 To i - 1
      For counter1 = 1 To i
         For counter2 = 1 To i
            temp1(counter1, counter2) = temp2(counter1, counter2)
         Next counter2
      Next counter1
      For counter1 = 1 To i
         k = 1
         Do Until k > i
            For counter2 = 1 To i
               temp2(counter1, k) = temp1(counter1, counter2) * R(counter2, k) +
temp2(counter1, k)
               If temp2(counter1, k) > 1 Then temp2(counter1, k) = 1
            Next counter2
            k = k + 1
         Loop
      Next counter1
   Next counter

   'now, the final temp2 array contains the answer for the reachability matrix
```

```
   'so give the values in temp2 to the reachability matrix
   For counter = 1 To i
      For counter1 = 1 To i
         reach(counter, counter1) = temp2(counter, counter1)
      Next counter1
   Next counter

End Sub



' partition Macro
' Macro recorded 8/17/98 by Qi Dong
'this macro partitions the matrix based on the reachability matrix result
'



Public row(150, 150) As Integer
'row(x,y) contains the reachability set of each element in DSM
'x indicates the element in the matrix this variable is associated with
'y indicates the order of the variable in the reachability set of element


Public column(150, 150) As Integer
'column(x,y) contains the antecedent set of each element in DSM
'x indicates the element in the matrix this variable is associated with
'y indicates the order of the variable in the antecedent set of element

Dim counter, counter1, k As Integer
Public total_row(150) As Integer 'total number of elements in the reachability set
Public total_column(150) As Integer  'total number of elements in the antecedent set

Sub partition()
For counter = 1 To i
   'assign values to the reachability set of element #counter
   k = 0
   For counter1 = 1 To i
      If reach(counter, counter1) <> 0 Then
         k = k + 1
         row(counter, k) = counter1
      End If
   Next counter1
```

```
  total_row(counter) = k
  'assign values to the antecedent sen of element #counter
  k = 0
  For counter1 = 1 To i
     If reach(counter1, counter) <> 0 Then
        k = k + 1
        column(counter, k) = counter1
     End If
  Next counter1
  total_column(counter) = k
Next counter

'check if the initial reachability and antecedent sets are correct by representing them on the
spreadsheets
'Sheets("reachability set").Select
'For counter = 1 To i
'  For counter1 = 1 To total_row(counter)
 '    Cells(counter, counter1).Value = row(counter, counter1)
  ' Next counter1
'Next counter

'Sheets("antecedent set").Select
'For counter = 1 To i
'  For counter1 = 1 To total_column(counter)
 '    Cells(counter, counter1).Value = column(counter, counter1)
  ' Next counter1
'Next counter

End Sub

Sub level1()
'find subsets
Dim j, counter, counter1, k, total, elevel As Integer
Dim taken(150), temp(150), c(150) As Integer 'taken(x)=1 means element x has already been
taken out at a level

For counter = 1 To 150
  taken(counter) = 0
  c(counter) = 1
Next counter

total = i
elevel = 0
```

```
Do Until total = 0

elevel = elevel + 1

For j = 1 To i 'j is the row number in the table
   signal = 0
   For counter = 1 To total_row(j)
      If taken(row(j, counter)) <> 1 Then
         For counter1 = 1 To total_column(j)
            If row(j, counter) = column(j, counter1) Then
               signal = 1 'signal=1 means the element in the row has a same one in the column
               Exit For
            Else
               signal = 0
            End If
         Next counter1
      ElseIf taken(row(j, counter)) = 1 Then
         signal = 1
      End If
      If signal = 0 Then
         Exit For
      End If
   Next counter

   k = 0

   If signal = 1 Then
      For counter = 1 To total_row(j)
         If taken(row(j, counter)) = 0 Then 'means this element is not taken out before
            k = k + 1
            total = total - 1
            temp(k) = row(j, counter)
            taken(row(j, counter)) = 2 'now the element is taken at this level
         End If
      Next counter
      'output to the sheet "final levels"
      Sheets("final levels").Select
      For counter = 1 To k
         Cells(elevel + 1, c(elevel) + counter).Value = temp(counter)
      Next counter
      c(elevel) = c(elevel) + 1 + k
   End If
```

```
Next j

Cells(1, 10).Value = total
For counter = 1 To i
   If taken(counter) = 2 Then
      taken(counter) = 1 'now marks then are all in the previous level
   End If
Next counter

Loop

End Sub
```