

# Developing the Requirements for an Assembly Advisor

By

Brian A. Landau

B.S. Mechanical Engineering

Carnegie Mellon University, 1998

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

**MASTER OF SCIENCE**

AT THE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

FEBRUARY 2000

© 2000 Massachusetts Institute of Technology. All rights reserved.

Signature of Author: \_\_\_\_\_

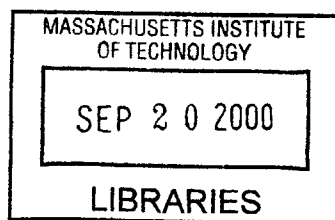
Department of Mechanical Engineering  
January 21, 2000

Certified by: \_\_\_\_\_

Daniel E. Whitney  
Senior Research Scientist  
Center for Technology, Policy, and Industrial Development

Accepted by: \_\_\_\_\_

Ain A. Sonin  
Professor of Mechanical Engineering  
Chairman, Committee for Graduate Students





# Developing the Requirements for an Assembly Advisor

By

Brian A. Landau

Submitted to the Department of Mechanical Engineering on  
January 21, 2000 in Partial Fulfillment of the Requirements  
for the Degree of Master of Science in Mechanical  
Engineering

## ABSTRACT

The design and development of any complex product requires a thorough understanding of how individual parts, components and sub-systems interact with one another. Very few product-development firms have the resources or the ability to effectively identify, interpret, and transfer this type of knowledge back to the earliest possible stage(s) of design. As a result, many firms take a reactive—rather than a proactive—approach to design, manufacturing and most particularly assembly.

The purpose of this research is to help identify ways to promote proactive design—in particular, design for assembly. The information here will be used to help knowledge-based engineering groups fashion an assembly advisor that can be used in all types of industries. In particular, the thesis explores four different pathways, which help to further understand the requirements for an assembly advisor: (1) determine what restrictions exist on knowledge-based advisors; (2) identify what types of assembly information already exist in both industry and academia; (3) describe how this information is currently represented; and (4) identify what information should be made available, regardless of what currently exists.

The advisor, when deployed, is intended to provide the designer with important feedback from the point of view of assembly, while still complementing existing design practices. The advisor will help a designer, during the design stage, look at assembly issues more closely while still promoting the importance of other issues—issues that drive the design of a product.

A large amount of information collected and used in this report came from a number of formal and informal discussions with experienced designers, engineers and software developers, and from analyzing various automotive sub-systems.

Thesis Supervisor: Daniel E. Whitney

Title: Senior Research Scientist, Center for Technology, Policy, and Industrial  
Development and Lecturer, Department of Mechanical Engineering

# TABLE OF CONTENTS:

---

<b>1. OBJECTIVES FOR AND APPROACH TO THESIS .....</b>	<b>8</b>
1.1 Introduction: The Need to Bring Assembly to the Level of Design .....	8
1.2 Research Objectives: .....	10
1.3 A Brief Introduction to Design for Assembly (DFA) and the Role of an Assembly Advisor:.....	11
1.4 Approach for Gathering Information Relevant to Research:.....	12
1.5 Roadmap for Thesis: .....	13
1.6 Final Thought: The Impact of Assembly on Product Design .....	15
<b>2. GENERAL KBE THEORY AND METHODS .....</b>	<b>18</b>
2.1 An Introduction to the Theory of Knowledge-Based Engineering:.....	18
2.2 What is a Knowledge-Based System? .....	18
2.3 The Advantages of Knowledge-Based Systems:.....	19
2.4 The Limitations of Knowledge-Based Systems: .....	20
2.5 The Type of Environment Needed to Successfully Develop and Deploy a Knowledge-Based System:.....	21
2.6 Are Knowledge-Based Systems Necessary for “Good” Design?.....	23
2.7 Chapter Summary and Conclusions: .....	29
<b>3. VISTEON/KBE METHODS.....</b>	<b>31</b>
3.1 Introduction: .....	31
3.2 Evolutionary Rise of Advisor Development within Ford/Visteon: .....	31
3.3 Background Information about the KBE Department: .....	35
3.4 KBE Strategy and Philosophy:.....	36
3.5 Process for Developing a Typical KBE Advisor:.....	39
3.6 Top-down/Bottom-up Processes: .....	41
3.7 Structure of a Typical KBE Advisor: .....	45
3.8 Types of KBE Advisors: .....	47
3.9 Chapter Summary:.....	50
<b>4. GENERAL DFA OBJECTIVES.....</b>	<b>52</b>
4.1 Roadmap for Chapter .....	52
4.2 Setting the Stage for Design for Assembly .....	52
4.3 What is Design for Assembly and How Does It Work in the Design Environment?.....	53
4.4 Limitations with Design for Assembly.....	55
4.5 Motivation for Placing Assembly Knowledge in a Hierarchical Structure and Its Role in “Design Space” .....	57
4.6 Classifying DFA Knowledge—General Framework.....	61
4.7 Classifying DFA Knowledge—Detailed Framework.....	64
4.8 Integral and Modular Architectures and Their Roles in DFA: .....	71
4.9 Chapter Summary.....	81
<b>5. DESIGN CONSIDERATIONS AT FORD/VISTEON AND THEIR INFLUENCE ON ASSEMBLY .....</b>	<b>83</b>
5.1 Introduction: .....	83
5.2 Lean and Mass Production and Their Contributions to Both System Engineering and Design for Assembly:.....	84

5.3 The Use of the Lean Production System and System Engineering at Ford: .....	87
5.4 The Flow of Information from Concept to Customer:.....	90
5.5 Collecting and Transferring Knowledge Using Design Guidelines: .....	92
5.6 The Primary Drivers in the Design of Different Vehicle Sub-Systems and Their Contribution to Assembly:.....	101
5.7 Chapter Summary:.....	106
<b>6. HVAC (HEATING, VENTING, AIR, AND COOLING) CASE STUDY .....</b>	<b>108</b>
6.1 Introduction:.....	108
6.2 Motivation: Why Analyze this Particular Vehicle Sub-System from the Standpoint of Assembly? .....	109
6.3 Background Information: .....	110
6.4 Different HVAC Architectures: .....	116
6.5 Results from the Analysis: .....	118
6.6 Chapter Summary:.....	137
<b>7. KBE TRADEOFFS AND PITFALLS .....</b>	<b>138</b>
7.1 Introduction:.....	138
7.2 KBE Boundary Conditions:.....	139
7.3 The KBE Strategy: Gaining Positive Word of Mouth.....	140
7.4 KBE Challenges:.....	143
7.5 Case Study: The Impact that Packaging and Product Architecture Have on the Design of KBE Advisors .....	146
7.6 Chapter Summary.....	153
<b>8. KBE/DFA TRADEOFFS AND PITFALLS .....</b>	<b>155</b>
8.1 Introduction:.....	155
8.2 The Issue of Breadth versus Depth:.....	156
8.3 KBE-DFM/A's Approach to the Development of an Assembly Advisor: .....	157
8.4 A General View of BDI-DFA™: .....	160
8.5 The Use of BDI-DFA™ in the Ford/Visteon Environment: .....	163
8.6 Conclusions and Chapter Summary: .....	167
<b>9. RECOMMENDATIONS AND REQUIREMENTS FOR AN ASSEMBLY ADVISOR.....</b>	<b>169</b>
9.1 Introduction:.....	169
9.2 A Brief Summary of the Previous Eight Chapters: .....	169
9.3 Recommendations and Requirements: .....	173
9.4 Potential Research Projects: .....	183
9.5 Chapter Summary:.....	186
<b>REFERENCES.....</b>	<b>188</b>
<b>SUGGESTED READING MATERIAL .....</b>	<b>191</b>
<b>APPENDIX.....</b>	<b>193</b>

# TABLES AND FIGURES:

---

<b>Figure 2.1:</b> Maintaining and Nurturing Knowledge in Different Company Environments .....	26
<b>Table 2.1:</b> The Use of Expert Systems in Different Company Environments .....	28
<b>Figure 3.1:</b> Evolution of Software Advisors .....	32
<b>Figure 3.2:</b> Transferring Knowledge .....	34
<b>Figure 3.3:</b> Segmenting KBE Knowledge .....	37
<b>Figure 3.4:</b> KBE Approach to Software Development .....	42
<b>Figure 3.5:</b> Bottom-up approach used by KBE.....	44
<b>Figure 3.6:</b> Structure of a KBE Advisor .....	46
<b>Table 3.1:</b> Characteristics of Various KBE Tools.....	48
<b>Figure 4.1:</b> The “Complexity Insertion” Loop.....	56
<b>Figure 4.2:</b> CDW-27 HVAC Designs.....	59
<b>Figure 4.3:</b> Design Drivers with Each HVAC Design Seen in Figure 4.2.....	60
<b>Figure 4.4:</b> Assessing Design for Assembly Rules .....	62
<b>Figure 4.5:</b> DFA Knowledge at a Deeper Level of Understanding.....	65
<b>Figure 4.6:</b> How Effective is DFA by Itself?.....	74
<b>Table 4.1:</b> Advantages/Disadvantages of Integral/Modular Architectures .....	77
<b>Figure 4.8:</b> General Example Showing How a Function Map Works .....	78
<b>Figure 4.9:</b> Function Map for Rotato™ Vegetable Peeler .....	80
<b>Figure 5.1:</b> Associating Process Development with Rework and Knowledge Transfer .....	91
<b>Table 5.1:</b> General Attributes of Component Design Manuals.....	94
<b>Figure 5.2:</b> Part-Selection Matrix for the Heater Core Design .....	96
<b>Figure 5.3:</b> Bar Chart from IP Matrix, Showing Assembly as the Second Highest Priority .....	99
<b>Figure 5.4:</b> Design Drivers for Various Product Designs .....	102
<b>Figure 5.5:</b> Fuel Tank Designs.....	105
<b>Figure 5.6:</b> Blower Motor Compartment for Two Different HVAC Units.....	106
<b>Figure 6.1:</b> Typical HVAC Unit (Picture taken from <u>The Auto Book</u> , 3 <sup>rd</sup> Ed. by W. Crouse and D. Anglin) .....	110
<b>Figure 6.2:</b> Schematic of an HVAC Unit, Showing Airflow through System (Picture taken from <u>The Auto Book</u> , 3 <sup>rd</sup> Ed. by W. Crouse and D. Anglin) .....	113
<b>Figure 6.3:</b> Typical Location of an HVAC Unit inside a Vehicle (Picture taken from <u>The Auto Book</u> , 3 <sup>rd</sup> Ed. by W. Crouse and D. Anglin).....	116
<b>Table 6.1:</b> Bill of Materials for Each of the Four HVAC Units.....	119
<b>Table 6.2:</b> Comparing Different Aspects of Each Unit.....	124
<b>Figure 6.4:</b> Heater Core Cover Designs for the Ford Windstar and the Ford Taurus .....	126
<b>Figure 6.5:</b> Blower Motor Compartment for the Ford Taurus and the Ford Windstar Units .....	127
<b>Figure 6.6:</b> Attaching the Blend Door and the Evaporator Core to the HVAC Housing (in the Ford Taurus and Windstar Models).....	130
<b>Figure 6.7:</b> Attaching the Air-Intake Compartment to the Windstar Housing.....	131
<b>Figure 6.8:</b> Door Assembly for Ford Windstar Unit.....	132
<b>Figure 6.9:</b> Heater Core Assembly in the Toyota Camry and Ford CDW-27.....	134
<b>Figure 6.10:</b> Different Mating Plane Configurations .....	136
<b>Figure 7.1:</b> Cause-Effect Diagram Illustrating KBE Strategy .....	141
<b>Figure 7.2:</b> Challenging Areas that Face KBE.....	143
<b>Figure 7.3:</b> Types of Information Acquired by KBE .....	144
<b>Figure 7.4:</b> Process Used by the Two Versions of the KBE-HVAC Advisor.....	149
<b>Table 7.1:</b> Products with an Endoskeleton and an Exoskeleton.....	153
<b>Figure 8.1:</b> The Assembly Advisor Dilemma.....	156
<b>Figure 8.2:</b> KBE Assembly Advisor.....	158

<b>Figure 8.3: Effective Use of DFA Suggestions .....</b>	<b>164</b>
<b>Figure 8.4: The DFA Dilemma.....</b>	<b>166</b>
<b>Figure 9.1: Key Aspects of an Assembly Advisor and Their Overlapping Responsibilities .....</b>	<b>170</b>
<b>Figure 9.2: Possible Ways to Link the User with Both System-Level and Part-Center Design .....</b>	<b>187</b>

# CHAPTER I

---

## OBJECTIVES FOR AND APPROACH TO THESIS

### 1.1 Introduction: The Need to Bring Assembly to the Level of Design

Assembly may be considered one of the most underrated actions in the fabrication of any product, yet the design and preparation of a product for assembly is probably one of the most challenging. The act of assembling parts together is an instinctive action that humans learn at a very young age (Redford and Chal, Design for Assembly: Principles and Practices). However, it is the act of designing parts for assembly that, in most, cases is far from inherent. The process of what to do may not always be clear, and the outcome at times is far from what is normally expected<sup>1</sup>. In order to tackle these issues with a clearer understanding, a different approach needs to be adopted—one that can systematically control and even help predict overlooked or non-linear relationships that make their way all throughout the design and development process, right down to assembly, itself. This approach is commonly called system-level or top-down design, and it will be considered throughout this thesis from the point of view of assembly.

System-level design is a type of methodology that requires a firm understanding about each layer of a product's design, from the individual features on a part to the various relationships that exist between parts, components and sub-systems—and, of course, the combinations that go across these three levels. As the reader can most probably infer, just from reading this definition, using this methodology to design a product for assembly, especially for one that is as complicated as an automobile<sup>2</sup>, requires an extensive amount of preparation and a great deal of understanding about the product.

---

<sup>1</sup> D. Whitney, MIT Course 2.875: "Mechanical Assembly and Its Role in Product Development"

<sup>2</sup> If the reader considers that the average automobile has approximately 15,000 parts, with each part having an average of two mechanical links to other parts, then the total number of mechanical links in an automobile is 15,000. However, this only tells a small part of the story: P. Smith and D. Reinersten mention in their book Developing Products in Half the Time that the complexity of A project "grows by compounding, rather than simply growing linearly, as more requirements are added." (pg. 66) The authors also mention that "as the number of elements [i.e., parts, features on parts, etc.] in a product grows, each element has to be designed more and more carefully to maintain overall system goals." (pg. 68)



To understand the potential benefits of system-level design, it is best to first understand how most products are currently designed and where existing methods fall short of their goals. In most cases, the process of designing a product begins at the part level, the state at which individual parts are made and tested, and ends at the system level, the state at which various parts, components, sub-systems and conditions external to the system interact with one another. This is normally called the bottom-up approach to design, or simply part-centered design—the antithesis of top-down or system-level design. Although using this technique within a design environment can allow vast amounts of information to be regulated at an effective rate for human comprehension, there are certain limitations to this procedure—limitations that the author will now go into<sup>3</sup>.

Generally speaking, the detailed design of a product follows the sequence of steps listed below<sup>4</sup>:

1. Use a CAD/CAM/CAE tool to design individual parts; “assemble” parts to each other; and analyze parts and/or assemblies in a variety of ways, exposing potential problem areas related to assembly, manufacturing and/or performance.
2. Build a model of each part; assemble the model parts together; evaluate the manufacture-ability of each part and the assemble-ability and performance of the model assembly.
3. Simulate the production of the final product; detect and correct problems witnessed during the simulation.
4. Begin actual production of the product; detect and correct problems witnessed during actual production.

At every step of the process and across the entire process, a bottom-up approach is used. Local-level concerns are attacked first (e.g., the design of individual parts, the analysis of individual parts, etc.), followed by global-level concerns (e.g., the assembly and analysis of designed parts using a CAD tool, the assembly and analysis of modeled parts during the prototype build, etc.) In many cases, this is considered a trial-by-error approach to design, with, of course, a number of operations, or checkpoints, infused into the process to control most levels of uncertainty.<sup>5</sup> However, what these checkpoints do, besides

---

<sup>3</sup> Ford, like many other companies, is breaking away from this traditional methodology, and moving toward a methodology that aligns itself well with system-level design. Within the past few years, Ford has been applying FPDS (Ford Product Development System) to most parts of the Organization. FPDS is a system-management approach to the design and development of customer-driven vehicles. The development system provides ways to associate customer requests with the design of a vehicle. More about FPDS will be discussed in Chapter V.

<sup>4</sup> In Ford’s case, this is the back half of FPDS. As Chapter V will mention, FPDS has not influenced this part of product design yet.

<sup>5</sup> The prototype build can probably be considered the biggest checkpoint of the entire design/development process. What it basically does is reduce the impact of any uncalculated errors that can go from design to actual production.

what they are intentionally aimed to do, is increase both the time and cost for design and development of a product.

With the ever-increasing need to develop products in even a shorter period of time than they presently are and at less cost, many product-development companies are shifting focus and responsibility from the later stages to the earlier stages of product design. In the process, these companies are trying to reduce the number of checkpoints described above.<sup>6</sup> Doing so, however, requires a shift in thinking—a shift from the more traditional approach to a new, organizationally driven approach that will complement the increasing need for quickly designed and inexpensively developed products. Such a methodology requires not only mapping out different aspects of the overall process but also converting as much subjective information into a form that can be quantified, normalized (for comparative purposes) and used repeatedly from project to project.

The motivational and challenging aspects of system-level design and particularly design for assembly will set the groundwork for the author's research objectives, discussed in the next section of this thesis. After that, the author will then talk about design for assembly, what is it, what it is able to do for designers, and how it can be applied to a knowledge-based advisor. From there, the author will discuss the "roadmap" for his research project—conforming the idea of system-level design and design for assembly into a structured and systematic plan of action, which can be effectively implemented and used within product-development companies like Ford Motor Co. and Visteon. Principles from knowledge-based engineering (KBE) will be used to establish the framework for how such an organizational plan can be quickly and efficiently integrated into these companies, so as to make the greatest impact in all stages of design and development.

## 1.2 Research Objectives:

To some degree, assembly is taken into account during the design stage in any product-development company. But the questions that one has to ponder over are, how much is it considered at that level; what type of approach is used when it is considered; and is the existing approach necessarily the best one? These questions and others will be raised and answered within the context of this thesis. Even though the author uses situations and products from Visteon to promote design for assembly practices, the reader should make every effort to expand these ideas beyond Visteon's "walls" and even perhaps beyond the scope of the automobile industry.

---

<sup>6</sup> In one meeting that the author attended in the summer of 1998, in Dearborn, MI, a highly regarded engineer from Visteon's Electronic Division spoke about the future of the Organization and its plans to separate from Ford Motor Co. During one part of his talk, he mentioned that Visteon is making every effort to eliminate all testing equipment from its factory floors, relying on the quality of its parts and assembly of those parts to generate reliable products each and every time. Clearly, such a claim will require Visteon's products to be designed using a system-level approach rather than a trial-by-error approach, where testing equipment is a necessary part of the process.

The purpose of this thesis is to identify and understand the method(s) that Visteon uses to design vehicle sub-systems for assembly, and then to put this information into the framework of an assembly advisor for general use—not just for Visteon’s purposes. The thesis will discuss how assembly knowledge is currently represented and managed in both academia and industry; where these methods fall short of their desired goals; and what factors need to be considered in order to have such a tool be accepted in an industrial setting.

Visteon understands the importance of design for assembly and is exploring new avenues to promote this technique to even further heights within the Organization. For more than a year, Visteon’s Knowledge-Based Engineering (KBE) department has been the carrier of this new methodology, creating software solutions to improve the very process of assembly by design. The information contained in this thesis is designed to help KBE add to its own knowledge about these practices and to make it think about design for assembly from a larger scope. The author also complements this information with some knowledge about the development and implementation of different KBE software tools, gained from a number of discussions with technical experts within KBE’s ranks.

The intent of this thesis is to make the reader aware of system-level design, design for assembly and the importance and benefits of each one. The thesis will answer the question, why is it important to tailor a design to the conditions related to assembly, rather than to modify these conditions to the design? Such a shift in methodology can allow changes to be made in a shorter time and at less cost, but such thinking also requires a good understanding about all of the necessary interactions that make the parts of a product assemble together correctly and easily each and every time—something that very few companies are able to do intuitively. As a result, it is imperative to the development of this methodology to construct a process, which can identify, manage and easily represent all of the necessary relationships in a design, during every part of the design process.

### 1.3 A Brief Introduction to Design for Assembly (DFA) and the Role of an Assembly Advisor:

Throughout this thesis, the author will explain where the current methodology of design for assembly stands within the context of product design and knowledge-based engineering (KBE); what needs to be done in order to extend DFA to higher levels of system interaction; and what obstacles stand in the way of making these goals realistic within a user/advisor environment. As will be explained in Chapter IV, DFA is a “tool,” which is mostly used by designers to assess the assemble-ability of a design, during product design. DFA “tools” can come in a variety of forms—from simple rules and guidelines to complicated evaluation and simulation software<sup>7</sup>. However, none of these “tools” have the ability to see how DFA principles impact the physical and functional world of a product (i.e., the system-focused environment of design). As a result, most

---

<sup>7</sup> One part of the Appendix will include some of the present software tools available to the public sector.

DFA applications approach design from the point of view of features, parts, and interfaces (i.e., from the local design region), where very little product knowledge is needed for an evaluation.

The assembly advisor needs to act as a carrier of both system-level and local-level design principles. To make such an advisor, especially within a KBE environment, requires the understanding of key parameters of product design (e.g., product architecture), process design (e.g., different assembly sequences as a function of cost, time, etc.), and the links between the two (e.g., weight and size of a “growing” assembly with respect to various handling operations). The author lists some of the things that the advisor should deal with:

- Approach each assembly type using a different set of stipulations; assembly evaluations should be based on given assembly architectures.
- Provide designers with suggestions that are detailed enough to be used in particular product applications. (Suggestions should provide insight without taking away designer creativity.)
- Understand the importance of assembly and where it fits in the design of different products. (Assembly may not be the primary driver for a design, so it is important for the advisor to identify different tradeoffs and present them to the user.)
- Encourage more interaction between designers and manufacturers. (The tool should not be intended to isolate one group from the other, like many DFA “tools” do.)

#### 1.4 Approach for Gathering Information Relevant to Research:

The author spent approximately 18 months collecting and synthesizing all the information seen in this thesis. A little less than half of that time was consumed talking directly to a number of designers, manufacturing and quality engineers, industrial engineers, technical consultants, and software developers within Visteon. The remaining time was spent at MIT learning about the uses of design for assembly and how it is perceived in academia by attending lectures, reading a number of books, journal articles and web pages about the subject, and getting exposure to a software tool called DFMA™ by Boothroyd-Dewhurst, Inc.

During the course of this research project, the author made four visits to the Dearborn, MI area, spending that time primarily learning about how assembly is considered at the design level. The author mainly took the role of a consultant, interviewing different people within the Visteon community from various departments. Most of the author’s time was spent talking to people within the KBE department, at the Danou Technical Center (DTC). In addition, the author made one or more visits to the following facilities, speaking to experts about their respected responsibilities and projects and seeing how different vehicle sub-systems are assembled on the factory floor<sup>8</sup>:

---

<sup>8</sup> For a complete list of Visteon facilities, refer to <<http://www.visteon.com/about/facilities/region2.html>>

NAME OF FACILITY	KEY PRODUCTS PRODUCED
Chesterfield	Seats, foam pads, sew covers, kit builds, and headrest builds
Climate Control Systems	Air handling subsystems, refrigerant transport subsystems, powertrain cooling subsystems, heater coolant subsystems, heater coolant subsystems, and control subsystems
Rawsonville	Fuel and air charging systems, fuel injectors, windshield wiper motors, alternators, and fuel pumps
Saline	Instrument panels and consoles
Sheldon Road	Heater and air-conditioning assemblies, aluminum heater cores, aluminum radiators, and heater and air-conditioning controls
Technical Center—Allen Park	Interior car and truck systems and components, and electronics driver information systems and audio products
Technical Center—Glendale	Exterior lighting, and powertrain air induction and fuel vapor storage systems

Although the author spent some time at each of these facilities, most of the information included in this thesis came from paying particular attention to how air-handling subsystems and instrument panels are made at the Sheldon Road and Saline Plants, respectively. For the most part, the author had a great deal of contact with the people at these facilities and exposure to the products that they design and make.

Detailed information about the design of both air-handling subsystems and instrument panels were collected from design manuals and Ford/Visteon internal web pages. In addition, general information regarding other departments and their products were also collected from the internal web system, for comparative purposes.

### 1.5 Roadmap for Thesis:

The remaining part of this thesis will be broken up into nine sections. The first eight sections (Chapters II – VIII) will look at design for assembly and the requirements for an assembly advisor, each in a different light; while the last section (Chapter IX) will be

dedicated to gathering all of this information and representing it in the form of listed recommendations and requirements.

## **Chapter II: General KBE Theory and Methods**

The purpose of this chapter is to help the reader understand the theory behind knowledge-based tools, the types of information that they support, as well as their benefits and limitations in different environments. This chapter helps set the framework for how Visteon's KBE Department develops so-called "expert systems" and what it needs to consider when making an assembly advisor.

## **Chapter III: Visteon/KBE Methods**

This chapter will take the information from the previous chapter and bring it into an environment where issues like time for implementation, ease of use, and limited resources and information are very important and must be considered at every level. In this chapter, the author will discuss the different factors which helped ignite the idea of knowledge-based engineering within Ford and Visteon and the strategic approach that the Department takes when developing each of its software tools, from feasibility testing to final release and post-implementation studies. Furthermore, the author will explore many of the challenges that Visteon/KBE faces, and some of the efforts that it is now exercising to help alleviate many of these problems. (A detailed description of Visteon/KBE challenges will be covered in Chapter VI.) Most importantly, the author will discuss some of the general comments made by the users of these tools and the actions taken by Visteon/KBE in response to those comments. All of this information will set the framework for what is needed when developing a typical advisor, let alone one that is intended for assembly analysis.

## **Chapter IV: General DFA Objectives**

In the last chapter, the framework for an assembly advisor was discussed; however, the information needed to support this tool was not. This very issue brings the reader to the next topic of discussion, and probably one of the most important topics in this thesis, the key objectives for general design for assembly (DFA). In this chapter, the author discusses the general practice of design for assembly, talked about and used in academia and in industry; some of the important issues (i.e., dilemmas) that DFA fails to point out and why; and the ways in which DFA can tackle these dilemmas. The author will pay close attention to how most DFA literature segments assembly, where the main focus has primarily been, and why it has been there. The author will then show a more detailed segmentation of the same topology and discuss why it is important to look at assembly from this point of view. Most of the chapter is based on the framework of this topology. Each of its parts will be described and illustrated in some cases with brief examples using real products.

## **Chapter V: Assembly Considerations within Ford/Visteon**

In many large companies like Ford and Visteon, where complex products are constantly produced, several issues have to be considered and weighed simultaneously. The reader may not be surprised to know that assembly is considered a relatively small driver in the

design of most products. If this is the case, then what are the issues and how do they affect design for assembly? The points that will be covered in this chapter include, what issues drive the design of most vehicle subsystems; how do these drivers actually affect the architectural design of these subsystems, particularly in the context of assembly design; and how do both Ford and Visteon deal with “capturing” and expressing this information for later use. The author will answer these questions using important pieces of information collected from assembling and disassembling different subsystems, reading various design manuals, and speaking with several designers and engineers across Visteon.

### **Chapter VI: HVAC Case Study**

This chapter will study various vehicle Heating, Venting, Air and Cooling (HVAC) units; how they are designed; what factors drive their design; how these factors may conflict with good assembly practices; and how the units can be better designed for easier assembly. The purpose of this chapter is to apply DFA in a realistic environment where a number of issues, not just within design for assembly, “fight” for “space.”

### **Chapter VII: KBE Tradeoffs and Pitfalls**

This chapter will look deeply into some of the challenges that the KBE Department has faced and is continuing to face in their quest to develop software tools that follow their customer’s liking. In particular, two KBE tools will be used to show how different tools are made to accommodate different customer needs.

### **Chapter VIII: DFA/KBE Tradeoffs and Pitfalls**

This chapter will be broken down into two separate sections. The first section will briefly describe and differentiate various kinds of DFA software tools—how they approach each analysis and what types of information they require to make an analysis. The second section will go into what the DFA/KBE team is doing now to develop its own assembly advisor for use within Visteon and what they have and perhaps have not considered for its development.

### **Chapter IX: Recommendations and Requirements for an Assembly Advisor**

This chapter will bring together all of the pivotal information from the previous chapters and arrange it in a way that can provide the reader with a strong list of recommendations for developing an assembly advisor. The author takes the view that these recommendations are not solely for use within Visteon, Ford, or even within the automobile industry; the information provided here is arranged in a way that can be used for all industries which design and fabricate mechanical devices.

## **1.6 Final Thought: The Impact of Assembly on Product Design**

Assembly is the one of the most critical processes in the development of any product. It is at this point where inconsistencies, poor judgement calls, and oversights truly make

themselves known. D. Whitney gives his own reasons for why assembly is so crucial in product design<sup>9</sup>:

- Assembly is inherently integrative; it is at this juncture where parts, people, departments, and even companies come together.
- It is where quality is actually determined.

In general, assembly can make or break the entire development process. Even though assembly, by itself, has little bearing on the entire cost of a product, it can definitely instigate huge costs in other ways. Consistent problems with product assembly can result in huge delays and tremendous costs in rework, troubleshooting, and redesign.

For some time, it was perceived that these types of problems could be virtually eliminated with the extensive use of automated assembly equipment. The thought was that even poorly designed products could be manufactured and assembled without concern if the most sophisticated assembly machines were used. Many huge and well-known manufacturing companies, including General Motors (GM) and Polaroid<sup>10</sup>, learned an expensive lesson from this so-called misperception. P. Ingrassia and J. White vividly capture this misnomer in their book Comeback: The Fall and Rise of the American Automobile Industry. In the book, the authors discuss the tremendous amount of money that GM spent to help automate its factories and deliver cars and trucks in a shorter period of time than its Japanese competitors. “Between 1980 and 1985, GM spent a staggering \$42 billion on new factories, tools, equipment, and products.” The results were far from what were expected by the Company: cars and trucks were made in longer time than they previously were and with poorer quality; GM’s capital funds were diminishing substantially; and its relatively large portion of the automobile market was quickly deteriorating.

Ingrassia and White go on to explain the reasons for these unexpected turnarounds in productivity and in market share, citing a number of examples. They make this observation:

“Compounding the technical problems with Hamtramck’s robots was the complex design of GM’s cars. Hamtramck’s cars were overloaded with electronic gadgetry and hard to build. The front and rear bumper of a Cadillac Seville had more than 460 parts, and took thirty-three minutes of labor to put together. Two years after it opened, Hamtramck put a

---

<sup>9</sup> MIT Course 2.875

<sup>10</sup> The difficulties that Polaroid had using Sony robots to help assemble its cameras were discussed in the MIT Course 2.874, “Integration in Product and Process Design.” The point that was made by the professor was that Polaroid, at the time, failed to see the implications of having products not geared for robot assembly. D. Whitney added that Sony helped Polaroid redesign the cameras for robot assembly; and Polaroid, from there, learned how to do design future cameras this way. Polaroid learned a very important lesson from this situation: it is not always possible to design or redesign a production process for a given design.



stunning 100 hours of labor—five times as much as Toyota—in building each car. In 1985, the year the plant started production, GM’s spending on “maintenance and repairs” jumped by 18 percent to \$5.4 billion from \$4.6 billion the year before.” (pp. 111-112)

This very quotation really explains what has been said all along within this chapter: assembly is an important aspect in product development, and should be considered at every level of development, especially at the level of design.

# CHAPTER II

---

## GENERAL KBE THEORY AND METHODS

### 2.1 An Introduction to the Theory of Knowledge-Based Engineering:

The purpose of this chapter is to understand the general characteristics of software advisors; the way in which they store various forms of knowledge; and the limitations or boundaries that they all “endure” when storing, managing and using knowledge for various applications. The information from this will help provide a better understanding of what types of information can be collected for use in an assembly advisor and how it can be represented in a knowledge-base format.

In addition to these fundamental issues, the chapter will also challenge the uses and expectations of knowledge-based systems in the design environment. In essence, the chapter will take a step back and look at the issues important to the understanding of when a knowledge-based system is necessary to design, and when it is not. The author will briefly demonstrate how other companies, like Toyota, design easy to manufacture and assemble products with less computing power.

### 2.2 What is a Knowledge-Based System?

Knowledge-based systems, or expert systems as they are more commonly called, “can be defined as an intelligent program that has the capability to solve difficult real-life problems using a knowledge base and inference procedures.”<sup>11</sup> Ideally, an expert system should be able to identify problem areas, assess existing tradeoffs (and the implications that come from any one of these possible decisions), and learn from previous experiences—much like a human would do. The expert system must be made to communicate with other computer software packages, as well.

Expert systems store and utilize two types of knowledge in their knowledge base: (1) “raw” information such as rules, formulas, and assumptions, and (2) information that

---

<sup>11</sup> Refer to Manufacturing Engineering and Technology, 3<sup>rd</sup> Ed., by S. Kalpakjian

comes from inquisitive understanding and experiences. Information within knowledge base is normally expressed in the form of if-then computer algorithms, with each statement linked to a question or set of questions for the user to answer, directly or indirectly<sup>12</sup>

Visteon's Knowledge-Based Engineering Department develops the framework for each of its "expert" advisors in the same way, using data libraries and an if-then decision tree to store and manage knowledge, respectively. (More will be said about this in Chapter III.) On a recent project, known as the fastener library, this two-part knowledge structure was very easy to see explicitly. The fastener library is an interactive database which designers can use to select the correct fastener for a particular headlamp application. During the selection process, the designer is able to literally go through a decision tree, answering questions that will help narrow down the number of available fasteners. Although this is a very simple case, it still shows the general framework for expert systems at any level of complexity.

### 2.3 The Advantages of Knowledge-Based Systems:

Expert systems have enormous benefits in the industrial community, which is probably why research in this field of artificial intelligence (AI) has grown in leaps and bounds in the last few years. Because they are able to "consume" large amounts of information for analysis and provide precise results in a speedy timeframe, expert systems have found homes in different areas of engineering, law and business. The author has listed some particular reasons why expert systems are so beneficial, especially when placed in the right setting. (This list came exclusively from the web site mentioned earlier in footnote #12.)

- Expert systems do not forget, but humans may
- Many copies of an expert system can be made, but training new human experts is time consuming and expensive
- If there is a maze of rules, then the expert system can "unravel" the maze<sup>13</sup>
- Although expert systems are expensive to build and maintain, they are inexpensive to operate

---

<sup>12</sup> To learn more about expert systems, refer to [http://www.bus.orst.edu/faculty/brownc/es\\_tutor/es\\_tutor.htm#5-AD](http://www.bus.orst.edu/faculty/brownc/es_tutor/es_tutor.htm#5-AD), a web page, which cites key information about expert systems. The web page is titled "Introduction to Artificial Intelligence and Expert Systems" by C. Brown and D. O'Leary, and it is used as a teaching aid. Much of the material discussed in this Chapter came from this web site.

<sup>13</sup> In actuality, this is not true. Before the "maze of rules," as the authors call it here, can even be "unraveled" by the expert system, it must first be clarified and organized by the expert-development team, so that a logic sequence can be defined. What the authors may have meant by this statement was that an expert system can evaluate different factors simultaneously if the system's knowledge base has been designed to make a wide comparison. (This normally means that the factors under consideration have to be compared using common "dimensions.")

- Development and maintenance costs can be spread over many users (if, of course, there are many users)
- With expert systems, similar transactions can be handled in the same way; the system will make comparable recommendations for similar situations
- An expert system can provide permanent documentation of the decision process
- An expert system can review all the transactions, while a human expert can only review a sample

The ability of expert systems to retain and process huge amounts of information quickly and provide consistent results each time is the reason why they have almost unlimited potential. However, there are many factors, which provide a definite separation between what an expert system can do and what an expert can do—and this will be discussed in the next section.

## 2.4 The Limitations of Knowledge-Based Systems:

Although expert systems supply enormous advantages within different environments—for example, missile guidance systems, modeling and simulation of production facilities, computer-aided design, process planning, and production scheduling—they do have a variety of limitations. Many of them come from the fact that human reasoning is very hard to deduce, and recreate in the form of a decision tree.

Some of the limitations that hinder expert systems from making rational decisions are as follows (Again, refer to the web page mentioned in footnote #12):

- Unable to use common sense
- Unable to use creativity to solve unusual problems
- Unable to learn from past experiences
- Unable to use case-based reasoning
- Unable to use sensory experience (dependent on symbolic input)
- Unable to recognize when no answer exists or when the problem is outside area of expertise

Although these limitations definitely exist at some level today, their barriers are quickly eroding as several technical universities, like MIT, Carnegie Mellon University (CMU), and Stanford University, discover new findings in machine learning, case-based reasoning, shared learning (using autonomous robots, for example), and expert-system design. There are a number of web sites that the reader can go to learn about these areas of research, at these and other universities.<sup>14</sup>

---

<sup>14</sup> The author found the following web sites rather interesting, although the search was not quite extensive: Stanford University—Knowledge System Laboratory <<http://www.ksl.stanford.edu/>>; CMU—Machine Learning Group <<http://www.cs.cmu.edu/groups/ml/ml.html>>; and CMU—AI and Cognition <<http://www.cs.cmu.edu/groups/cognition/cognition.html>>.

## 2.5 The Type of Environment Needed to Successfully Develop and Deploy a Knowledge-Based System:

### 2.5.1 From the Side of Development:

As mentioned already, knowledge-based systems have a number of advantages, especially in product development, but the potential only exists when the information that they store and process is fashioned for computer recognition. C. Brown and D. O'Leary mention that "Computer programs are best in those situations where there is a structure that is noted as previously existing or can be elicited." (For the final time, refer to the web page mentioned in footnote #12.) The Visteon Knowledge-Based Engineering (KBE) Department, as will be described in Chapter III, is constantly seeking out this type of structure in each project that it takes on. For instance, KBE looks for any iterative steps in an existing process, which can be automated. In addition, the "raw" information itself must have some quantitative characteristics so that it can be reasonably compared and the results themselves can be presented consistently over time and from project to project. But, as one can see, this limits what KBE systems can do; many processes and decision are not based on sequential cause-effect relationships. More will be said about this topic later in the thesis.

However, before development should even begin, many other issues have to be considered first—issues such as the cost for developing the tool and the expected results from the tool. These and other issues need to be carefully weighed in order to see if the tool can make a significant contribution to an existing process. In addition, the tool must fit the organization's core methodology and be adaptable to changing conditions. These issues will be discussed in the next sub-section.

### 2.5.2 From the Side of Deployment:

Three issues come to mind when thinking about the deployment of an expert system into a user environment: (1) the actual "fit" of the system into the existing process; (2) the level of user-friendliness provided by the system; and (3) the user's level of confidence with the system's results. The first two issues have to be considered during project development, and can be controlled by the expert-system development team. By first looking at how designers interact with the tools that they use, the development team can develop an advisor that complements the designer's environment. The designer can use the advisor, developing new designs quicker and more efficiently, without severely changing his work patterns. The third issue, however, is not necessarily a variable which can be explicitly acted on by the team; raising the user's level of trust in the system requires nothing more than successful user interaction over time.

"Fitting" an expert system into a process requires the information through the process to remain unchanged. For instance, an expert system must be able to take in information from one part of the process, analyze the information, and present the results in a format that is recognizable at the next level. In essence, the system must not only be user

friendly but “process friendly” as well. Alternatively, if an expert system is to be developed in a manner, which changes the way existing procedures are done, then the system must encapsulate the entire process and prove its benefit(s).

Likewise, the user interface must be made to comply with what the user likes in order for the system to gain immediate acceptance. This means that it must be compatible with what the user is comfortable using, and the sequence of steps in which information is requested must be asked in a logical way. Even though details like this are not prevalent during development, they do have a significant impact on customer perception.

One of the biggest challenges that deployment teams face when implementing a new expert system into a user environment is trying to convince the user of the potential benefits and the credibility of its results. In certain professions where wrong answers can lead to disastrous misfortune for the company and even for the end customer, this concern is no doubt valid. In these types of situations, the development team must be sure of the validity of the system and convey those thoughts to the user.

The author will communicate these ideas and others using an example of an expert system, which was deployed in the medical environment.

### 2.6.3 Example of an Expert System in a User Environment:

Mycin, a medical advisor created at Stanford University during the mid 1970s, is used to identify the cause of a patient’s infections. Mycin interacts with the physician by “asking” him or her various questions about the patient’s physical condition. Based on these answers, Mycin navigates down the heuristic structure contained within its knowledge base to converge to a diagnosis and possible treatment.

However, “When Mycin was first created, many physicians didn’t trust its diagnosis. There was no way for them to know whether or not Mycin’s reasoning was correct.”<sup>15</sup> Originally, Mycin showed the validity of its diagnosis by attaching a confidence score to it—but this was not enough to calm most physicians’ worries. As a result, an extra feature was later added to Mycin to provide the doctor with a transcript of the logic that Mycin used to come up with its conclusions. (To learn more about the general characteristics of expert systems and Mycin, refer to <http://www-formal.stanford.edu/jmc/someneed/someneed.html>.)

This example illustrates several points: (1) expert systems provide suggestions by following a decision-tree structure; (2) they require user interaction; and (3) they must provide ways to validate their suggestions. Many of these issues will be followed up further in the next chapter, where the author will describe how Visteon’s Knowledge-Based System Department addresses them.

---

<sup>15</sup> Refer to <http://web.mit.edu/STS001/www/Team7/mycin.htm>

## 2.6 Are Knowledge-Based Systems Necessary for “Good” Design?

It is not always the case that software tools and expert systems are absolutely needed to help improve the flow and growth of knowledge. In many cases, it is a matter of changing the way people think about how they design and develop products and how they use their experiences (or the experiences of others) to help improve existing designs. In other words, how a company is set up to allow information to be effectively grown, shared and utilized at all levels can really be the determining factor in good product design—regardless of what tools are used or how sophisticated they are.

Several people that the author had talked to at Visteon conveyed this idea, although indirectly. When asked what they felt needed to be done in order to improve product development and the quality of the products at Visteon, here is what they had to say:

“It’s easy...relatively easy to take cost out of a particular product by saying, ‘Here’s the design, now, can we use a cheaper material? Can we skip a step?’ It’s much harder to say, ‘How am I, in the essence of this design, baking cost into it?’ Some...some of that goes back to some of the things that we talked about at first, you know, ‘Do I have any orientation changes? Do I have to have...how many individual processing steps do I have to have? How many part variations do I have to have because that drives the level of inventory that I have to carry or to be able to accommodate?’ I mean it explodes factorially.”

Superintendent, Truck Line Management  
Sterling Axle Plant

“I wish somebody could do this—develop some kind of business model so that all engineers from all different levels have to get involved in or share the responsibility of controlling or, better yet, reducing costs. This hasn’t happened, yet. Especially within a large company, most engineers don’t really know how they put cost into a product. In this world, they are so competitive. I think, that everybody should have this as part of their job...to reduce cost, to get more profit out of their designs. No matter how good a product functions, no matter how quick it can be put to market...if you can’t make money from it, then you’re buried.”

“But when I talk to design engineers, the first thing that comes out is, ‘I don’t have any limits from Purchasing. I don’t have to think about this because that is not my job.’”

Technical Consultant  
Knowledge-Based Engineering

“You know, if everybody didn’t have to sign their name to it so that it was their part...they could tell their kids that they made that part, they designed that part...it would probably take a lot less time.”

Manufacturing and Quality Engineer  
Saline Plant

As each of these comments appear to show, sometimes it is not a matter of implementing new technology to improve a process, but a matter of changing the way people think about design. These quotations also show that in most cases designers seem to “battle” with one another on certain issues—issues that can either enhance or constrain each designer’s space to work. The quotations also express the need for more system management, where decisions are made to improve, for example, the performance of the system, rather than individual components. In other words, decisions should be made at the system level, and then they should be followed down to the part level in order to understand how negotiations should be properly managed. The Superintendent from the Sterling Axle Plant mentioned that Ford and Visteon have not been able to clearly follow this kind of information flow down to the part level yet—although efforts are being made. He also mentioned (as it is stipulated in the first quotation above) that this would require a change in thinking on the designers’ part. As he put it, designers need to understand how cost is automatically ingrained into a product design before it ever happens.

The next two parts of this section will discuss how U.S. and Japanese companies both approach the aspect of learning (i.e., retaining, nurturing, and overlapping or sharing knowledge), and perhaps reasons why expert systems are “acceptable” in one type of learning environment as opposed to another.

### **2.6.1 Knowledge Retention and Management in U.S. and Japanese Companies:**

Knowledge-based systems have become such fascinating tools, especially at Ford and Visteon, because of their extraordinary ability to retain huge amounts of information. In a sense, knowledge-based systems act like a highly interactive database where designers and engineers can go to for particular bits of information quickly and easily. Many of the employees at Ford and Visteon would say that this is a far better method for finding and extracting information than what previously existed—relying on the knowledge and experience of one expert, especially in an environment where employees are constantly moving from job assignment to job assignment. However, other companies, such as those in Japan, still encourage hands-on learning, cross-functional learning, and shared learning—all without the use of knowledge-based systems to accentuate the rate of knowledge growth and transfer.

To better understand this concept, the author will now delve into the different ways in which most North American and Japanese companies attempt to strategically reach the pinnacle of knowledge in product development. Although there is some synergy



between the different types of methods used in North American companies and in Japanese companies, there are some very obvious differences. P. Smith and D. Reinersten spend some time in their book, Developing Products in Half the Time, discussing these differences. Here is a sample of what the authors had to say about collecting and retaining knowledge through experience in both North American and Japanese industries:

“In automobile development the Japanese have a considerable cycle-time advantage over North America and Europe... [Kim Clark and others at the Harvard Business School] have found that on average it takes Japanese companies forty-three months to develop a car compared to sixty-two months in North America or Europe, for a ratio of about three cycles in Japan to two elsewhere. Projected out, this means that the Japanese can introduce and gain experience from nine generations of products while we are left with only six.” (pg. 69)

To further add to this piece of interesting information, the authors then made the following remark:

“Contrary to the faster-smaller steps approach, the Japanese steps in automobile development happens to be faster but not smaller than U.S. steps. Therefore, the Japanese gain a clear-cut victory every time they get another generation ahead. But even if their steps were smaller, they would still have an advantage because of the learning that takes place from introducing a product.” (pg. 70)

The Japanese companies put themselves in a position where knowledge and experience can be gained at a faster rate than their competitors. However, the question now becomes, how are they able to learn and retain what they learn at such a high rate? The Japanese do this by making sure to develop what they have learned to the point where it reaches a mastery level. As D. Whitney pointed out to the author several times, Japanese companies, like Toyota and Sony, encourage system-engineering practices (i.e., the integration of product, process and supply-chain management). This, however, requires continuous and extensive communication between different functional groups; and a “universal” commitment by the employees to help the Organization grow, as a whole. (System engineering cannot be successfully applied to an organization without first having the right cultural mindset in place.)<sup>16</sup> With this type of strategy, there is no need to go back and relearn pivotal pieces of information that may be important for later projects. Smith and Reinersten mention this very issue, as shown in the quotation below:

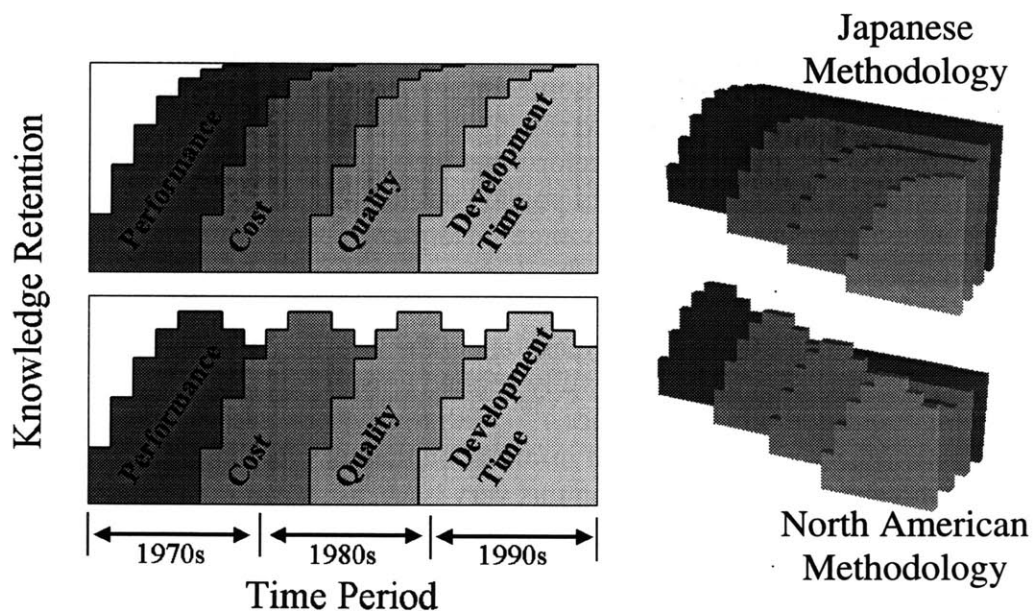
---

<sup>16</sup> D. Whitney also mentioned that in the case of Toyota, employees on a project usually stay together for the course of the entire project and perhaps beyond that time. This allows each member of the project to learn about different aspects of vehicle design and their levels of interaction. Such a strategy falls in line with system engineering.

“In this regard we learn from the Japanese, who cultivate the ability to learn from one technique well, then absorb it and move on to the next without losing earlier gains...The reason their current emphasis on product quality seems relatively low compared with that in North America is that they have for the most part mastered quality.” (pg. 9)

Figure 2.1 illustrates how both North American and Japanese companies maintain knowledge in particular areas of product design along the course of several years. The reader should pay special attention to how the four areas of product development—performance, cost, quality and development time—overlap one another in each case. For the most effective utilization of knowledge, from layer to layer and across different layers, it is necessary to leave a large “footprint”—as one might infer.

The main point that the reader should get from this figure is that most Japanese automotive companies continue to develop ways to master different regions of product development, even when emphasis might move to another area. They not only collect this knowledge, but also maintain and nurture it, as well. Most North American companies, on the other hand, use more of a serial approach to “raise the bar” of product development—meaning that if one issue becomes important for a given period of time, then other issues tend to be sacrificed. Using this type of methodology means simply that issues are scarcely reaching their saturation point and are rarely used in an efficient and effective manner at different times. As a result, the experience and knowledge that is reached and mastered at one point eventually falters and diminishes in magnitude, with respect to other areas in product development.



**Figure 2.1:** Maintaining and Nurturing Knowledge in Different Company Environments

This figure also helps to explain why some product-improvement methods move through North American companies like a craze rather than like an essential staple in the future development of product efficiency. For the most part, most North American companies implement new practices which can be isolated into one area of product development. For instance, as Figure 2.1 shows, “Development Time” has emerged as the descendent of “Quality Improvement” during the 1990s. This is one of the reasons why Design for Assembly (DFA) has become such an attractive methodology in product development—reducing the time to design, manufacture, and of course assemble products. However, this is where DFA reaches its limitation; it does not consider performance or quality issues to a great degree. This issue will be discussed further in Chapter IV, General DFA Objectives.

### 2.6.2 Different Company Approaches to Product Design and How They Tie in with Expert Systems:

D. Whitney mentions that many Japanese companies, like Toyota and Sony, use far less computing power than several of their North American competitors.<sup>17</sup> (In fact, he goes as far as to say that Toyota, in particular, uses about one-half of the computing power that Ford uses.) Many times, it comes down to how different companies retain and use knowledge for maximum effectiveness. This may be due to several things, such as how the organization is presently structured to allow for knowledge transfer, and in the area of product development, what are the key drivers in design and what tactics are enforced to allow those drivers to be recognized. In Toyota’s case, designers and engineers rotate jobs when projects are fully completed (about once every six years). This allows them to gain the maximum amount of consumable information at all levels of one project and to extend that knowledge effectively to other projects.

Different companies have different strategies for designing products; and in many cases, it is this distinction that can determine whether or not expert systems are a good match for any product-development company. Based on personal communications with D. Whitney, it appears that companies, which take a system-level view to design, tend to find less need for expert systems—at least as a major contributor to design work. D. Whitney made no comment about the opposite design strategy—whether or not part-centered design promotes, to some level, the need for expert systems, especially in a competitive environment like the automobile industry. However, based on the author’s research and findings at Ford/Visteon, this statement appears to have some degree of truth.

At any level, expert systems are capable of handling particular areas of design, but only in those cases when the key contributors can be de-coupled and isolated from other parts of design. For instance, when asked to design a product or part of a product, which emphasizes and satisfies a variety of different issues, most likely, designers will use their

---

<sup>17</sup> Dr. Whitney brought up this comment during an informal meeting with five of his graduate students (including the author); the meeting was held in September/October of 1998.

experiences, intuitive thinking and especially their creativity to come up with a successful design. Conversely, placing an expert system in this type of setting would most probably lead to failure due to the various limitations that affect all expert systems—namely the inability to process a complex nest of information that is relatively hard to segment and place in a top-down manner.

The chart below shows how some companies emphasize product design (particularly design for assembly) in different ways and what is the likelihood that an expert system can be used to map out either of these strategies. (More will be said about the issues revolving around design for assembly in Chapter IV.)

COMPANY EXAMPLES	DFA STRATEGY (EXAMPLES)		APPLICABLE ENVIRONMENT FOR EXPERT SYSTEM?
Ford Visteon	<ul style="list-style-type: none"> <li>• Consider consolidating one or more parts into one final part</li> <li>• Replace screws and other threaded fasteners with snap tabs</li> <li>• Provide chamfers for easy insertion</li> </ul>	Part-Centered DFA Strategy	HIGH Acceptability
Denso <sup>18</sup>	<ul style="list-style-type: none"> <li>• Consider variety during assembly process, not during fabrication process</li> <li>• “Identify sources of model/variety changeover time or cost and seek ways to eliminate them through product design”</li> <li>• Design products to utilize reconfigurable tooling</li> </ul>	System-Centered DFA Strategy	LOW Acceptability

**Table 2.1:** The Use of Expert Systems in Different Company Environments

Of course, the author is in no way trying to state that expert systems can only be used where principles and practices revolve around part-centered design or design for assembly; he is merely trying to say that this type of strategy is more applicable to expert systems. Let the author also add that expert systems, like the ones mentioned here, can encapsulate a significantly large part of product design (or in this case, design-for-assembly) when it is considered separate from the entire development process. As the author will explain in Chapter VIII, KBE is looking to develop an advisor that can capture several existing principles of design for assembly, including the theories of part

<sup>18</sup> Refer to Article “Nippondenso Co. Ltd: A Case Study of Strategic Product Design” by D. Whitney

gripping, handling and presentation, part reduction, and part mating, to name a few. However, as the Department may quickly realize, the principles behind assembly by design cannot be entirely encapsulated and placed inside an expert system; assembly is intricately coupled with the product and fabrication processes and with the supply chain. (C. Fine, in his book Clockspeed, explains that the type of architecture used to develop a system or sub-system can determine which parts will be manufactured in-house or purchased by outside suppliers—the “make/buy” decision process.)

## 2.7 Chapter Summary and Conclusions:

In this chapter, the author defined what an expert system is and described how information is represented in its knowledge base. Particularly, the representation of knowledge is very important to the development of an assembly advisor and provides a key to how assembly knowledge should be represented. For example, as the reader will understand more clearly in the next chapter, present DFA rules and techniques are not formatted for use in a knowledge-based environment. The general principles of DFA are context free, making the designer evaluate his own design for assemble-ability and using these guidelines wherever they may apply. During the latter part of this process, the designer searches his own “knowledge base” for certain pieces of information about the product (e.g., initial design stipulations, such as product size and shape; material selection; carryover-part selection, etc.), so that he can understand how far he can apply these rules. Designers usually have to understand in what ways DFA rules can be applied to a design without “disturbing” the parameters and conditions initially put in place. (In Chapter IV, a simple case study is provided showing how the DFA principle of top-down assembly can offset several of the conditions already put in place for the air-handling system of a car. Some of these conditions include the elimination of air and water leaks; the easy removal of injection-molded parts from a die; and the use of vibration welding, during assembly, to permanently seal parts together, to name a few.)

In addition, there is another problem that plagues assembly advisors: because of their rigid inference structure, assembly advisors are not suitable for very early concept-level work. The number of decisions, that need to be made about a product at this level, can be almost overwhelming for humans, let alone for expert systems. And in many cases, the connections between these decisions make the overall decision process virtually impossible to describe with an if-then computer algorithm. For example, many decisions, especially those at the earliest phase of design, cannot be done sequentially. Additionally, their effects may not show up until much later in the process. Designers take part in a number of tightly coupled serial, parallel, iterative and non-iterative tasks, before finally converging to a final result in their part of a design. If-then statements are not capable of tackling such tasks. (That is why user interaction is imperative: the user is needed to determine the validity of the expert system’s decisions and to use these results to evaluate a design from a more global view.) As a result, the expert advisor may only be feasible at a point in the design process when most decisions have already been made, and the process from there can be clearly defined.

In the next part of the chapter, the author then discussed what type of factors must be in place for an expert system to be properly developed and effectively deployed in a user environment. Issues such as “fitting” the tool to an existing process; developing a user-friendly interface; and raising the user’s level of confidence in the tool were all briefly discussed. Many of these issues were talked about from a very high level for the reason that it is rather difficult to discuss the details of any one of these issues without reference to any particular cases. In the next chapter, expert systems will be looked at more thoroughly in the context of a user environment—more particularly within the Ford/Visteon organization.

At the chapter’s conclusion, the author spoke about expert systems in a design environment, one segment of the user environment, showing some of the conditions that need to be in place for its utilization. The author discussed how not all companies feel that expert systems are essential for proficiently quick product design. Knowledge systems definitely have their advantages, especially in the automobile industry where product complexity is huge. However, some companies believe that perhaps the best way to learn about a product is through long-term contact with particular projects. In other words, these companies feel that knowledge can be gained across different areas of design by staying on projects through their entirety. As a result, a cohesive understanding about all levels of product design can be consumed.

An association was then made between a given approach to design—be it part-centered design, system-level design, or a combination of the two—and its level of acceptability toward the use of an expert system. An example was provided using two different approaches to design for assembly that are utilized by two separate companies. The author attempted to prove that a “part-centered” environment would look at expert systems more favorably than a “system-level” environment would because of the limitations that inhibit tool-type flexibility and the various organizational structures within each environment. Such insight will help to show where expert systems can be used most effectively in the design-for-assembly environment.

# CHAPTER III

---

## VISTEON/KBE METHODS

### 3.1 Introduction:

The purpose of this chapter is to understand the various factors that both promote and inhibit the development of a typical “expert” advisor, within the context of a commercial environment. Knowledge and experiences from Visteon’s Knowledge-Based Engineering (KBE) Department will be used to help understand these issues more clearly.

In general, this chapter will focus on the following topics:

- Understand what the initial framework should be for a typical expert advisor, based on various circumstances
- Understand the various limitations that exist when developing any type of advisor
- Evaluate the pathway that KBE takes, and see if it is a reasonably sound strategy to follow
- Identify what decisions impact the use of a KBE advisor in a commercial environment
- See how all of these factors impact the development and use of an assembly advisor

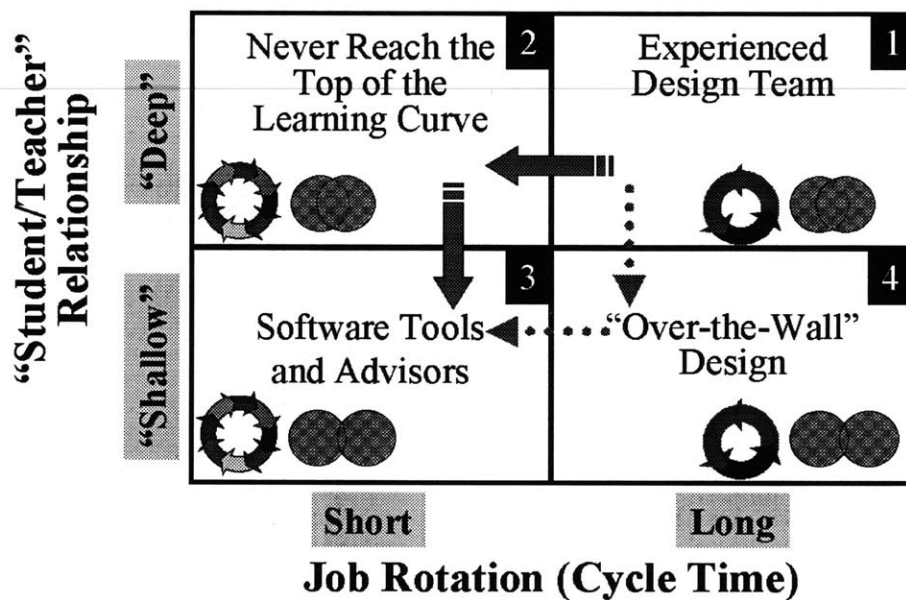
### 3.2 Evolutionary Rise of Advisor Development within Ford/Visteon:

There are many reasons why software advisors have gained such popular demand in recent years, especially within Ford and Visteon. Software advisors, in general, help to store and manage, what would be considered, sporadic information in an easy-to-access environment (when, of course, it is applicable to the application). The shift in

methodology from “on-the-job” learning to knowledge-based learning is due in large part to two factors that have almost become a trend within the Ford/Visteon community<sup>19</sup>:

- Time between job rotations is rather small. (Each job requirement lasts 2 to 3 years before the next job rotation—a relatively small time when compared to Toyota’s which is approximately 6 years per job assignment.)
- Lack of interactive learning from experienced personnel.

The combination of these two factors means less information is being gathered and transferred from experienced to novice designers and engineers. The following matrix (Figure 3.1) attempts to show how the evolutionary demand for advisors first originated:



*Pathway Taken by Ford/Visteon:*  
 .....➔ Weak Pathway  
 ➔ Strong Pathway

*Job Rotation:*  
 (Sun icon) Long Cycle Time  
 (Gear icon) Short Cycle Time

*“Student/Teacher” Relationship:*  
 (Sun icon) “Deep”  
 (Gear icon) “Shallow”

**Figure 3.1: Evolution of Software Advisors**

The figure should be followed in the order in which the four cells are numbered. Each cell shows two types of information: (1) the types of circumstances or conditions instilled at each phase (shown pictorially at the bottom of every cell), and (2) their respective effects (shown at the top of every cell in text form). The pathway taken at Ford and Visteon, shown with arrows going from cell to cell, was influenced by first quickening the time between job rotations and then reducing the level of knowledge sharing between

<sup>19</sup> Most of the information from this section came from a discussion with a Visteon manager who had worked for Ford for nearly 30 years.



individuals on a project. The author will now go into a more detailed description of each phase, and how they became the institution for the evolution of knowledge-based tools.

### Quadrant 1: Experienced Design Team

Originally, knowledge was nurtured and contained within a design group. The team usually stayed together for at least 10 years, internally gaining and sharing knowledge as it worked. New, inexperienced members assigned to the team worked along with the experienced members, acquiring knowledge from them and from project experiences. A rich source of knowledge, in this case, was transferred from one member to the next. The location of vital design information was time dependent, meaning that it was based on job cycles, turnover rates, and the ability to effectively transfer knowledge to a new group of individuals.

### Quadrant 2: Never Reach the Top of the Learning Curve

Starting a few decades back, the time between job transitions decreased from 10 years to 2 or 3 years. One of the most obvious reasons for this change came from the lack of attention or dedication most people had when working with a team for such an extended amount of time. It was found out that most workers wanted to learn a variety of different things, from different perspectives, and did not want to be dedicated to one specific task for a relatively long period of time. Ford's reaction to all of this was simply to reduce the time between jobs. However, this led to an unstable situation. Workers were not learning enough to be experienced in a particular discipline.

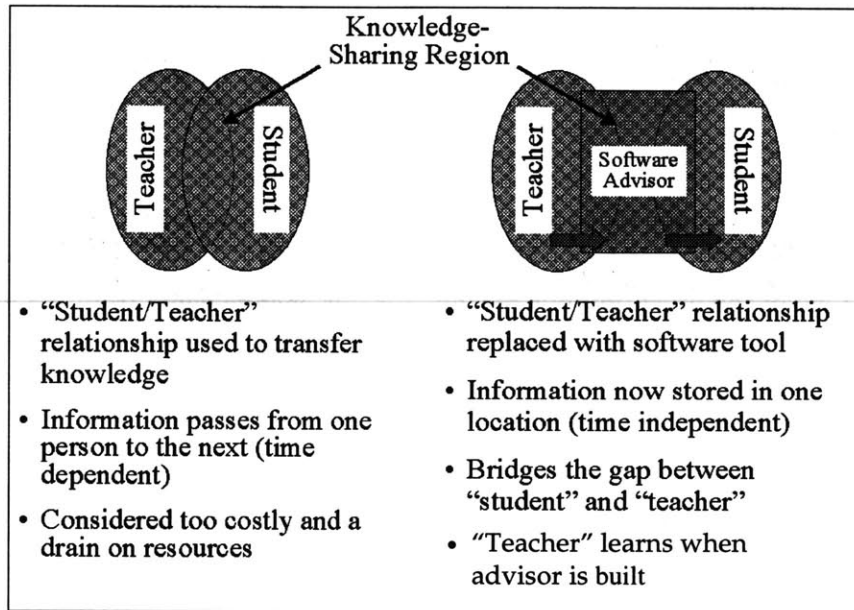
### Quadrant 3: Software Tools and Advisors

In addition to this move, Ford decided to limit the amount of cross-functional learning because it was a costly process that did not demonstrate a deep level of financial importance. The Company felt that by doing this, it would, in effect, save money. Ford's thinking was simple: let each person do a separate job, thereby reducing the number of job overlaps and allowing the work to be "attacked" more evenly.

However, this decision compounded the former decision, making the entire situation very unstable. Because many Ford employees were switching job functions at an accelerated pace, it became more and more difficult to track down and learn from personnel with expertise in special areas of design and manufacturing. In addition, people were not staying on a project long enough to fully understand the implications of their decisions; they did not have any past experiences in certain areas to fall back on. (Toyota, as D. Whitney mentioned, tries to avoid this situation by having designers stay on projects for a longer time—usually from the beginning to the end of a project.) To remedy this problem, the development of knowledge-based systems became increasingly more desirable than it had ever been before.

In this case, the gap between "student" and "teacher" is bridged with a knowledge-based system. In addition, the knowledge is consistently stored in one place, the knowledge-based system, making the whole act of knowledge transfer time independent. (The term "time independent," as it is used here, means that whatever knowledge is stored in

knowledge base is not affected by outside factors, such as the rate at which people rotate jobs or learn. There is a concern, however, for how useful this knowledge will be over time if it is not updated. This is a separate issue that the author will talk about later in this thesis.) Figure 3.2 highlights the important points that were just made about the “old” and “new” approaches to learning within Ford and Visteon, Quadrants I and III, respectively.



**Figure 3.2:** Transferring Knowledge

#### Quadrant 4: “Over the Wall” Design

For the most part, Ford went from Quadrant 1 to 2, and from Quadrant 2 to 3. Of course, the conditions described in Quadrant 4 do show themselves to some degree within Ford and Visteon.<sup>20</sup> (Ford is trying to correct this matter by bring into its community practices like system-level design, concurrent engineering<sup>21</sup> and direct engineering<sup>22</sup>—strategies that make up part of the framework for the Ford 2000 Project.) Regardless of which pathway Ford followed over the years, “Over the Wall” Design is an unstable practice

<sup>20</sup> Again, this information came from the discussion that the author had with a manager with almost 30 years of work experience with both Ford and Visteon. The sequence of events, which he described to the author, is consistent with the story told here.

<sup>21</sup> Kalpakjian, in *Manufacturing Engineering and Technology, 3<sup>rd</sup> Ed.*, defines concurrent engineering, or simultaneous engineering, as “a systematic approach integrating the design and manufacture of products with the view of optimizing all elements involved in the life cycle of the product.” (pg. 13) C. Fine notes in his book *Clockspeed*, that concurrent engineering has recently evolved beyond the scope of product and process design, adding on a third dimension--supply-chain management.

<sup>22</sup> Kalpakjian defines direct engineering as an extension of concurrent engineering, which “utilizes a database representing the engineering logic [including consequences] used in the design of each part of a product.” (Refer to *Manufacturing Engineering and Technology, 3<sup>rd</sup> Ed.*, pg. 13)

and places new designers and engineers in a situation where learning is not really promoted. As a result, this alternate pathway (Quadrant 1 to 4 and Quadrant 4 to 3) would have most probably led to the same end result that was originally described with the first pathway.

### 3.3 Background Information about the KBE Department:

The Knowledge-Based Engineering (KBE) Department develops expert advisors for designers and engineers within the Visteon community. KBE is a relatively new department that is constantly working to develop new tools that will make both the process of designing and manufacturing products faster, simpler, and more efficient.

KBE was first developed six years ago within Ford. (At that time, KBE was called DFM/A, most probably for the reasons that design for manufacturing and design for assembly were both gaining influence at Ford. However, as the reader will learn in greater detail, KBE first descended down the path of design for manufacturing. The question of why this was the case may be based on the fact that assembly knowledge is harder to contain than manufacturing knowledge.) KBE established itself as a department close to two years ago, developing products solely for the needs of the Ford/Visteon community. Recently however, KBE has been promoting its tools to customers outside of its own community, selling them to several other automotive companies<sup>23</sup>.

The Department has over 40 employees, from different disciplines, developing tools and providing customer support. Many are software developers, while others are consultants in the areas of lean manufacturing, plastic molding, and CAD modeling.

KBE is broken down into eight key sections, as shown below:

1. Planning and Development
2. Intelligent Subsystems
3. DFM/DFA Tools and Methods
4. Intelligent CAD/CAM/CAE
5. Integrated ECAD/MCAD
6. External Applications
7. Intelligent Components
8. Expert Systems

---

<sup>23</sup> The author does not really understand how such a marketing strategy can give Visteon an advantage over other automobile suppliers. If KBE sell its “expert” tools to various automotive part manufacturers so that they can design products faster and cheaper, then in time Visteon may lose much of its competitive advantage. This is a critical time for Visteon as it seeks out sub-system contracts from vehicle assemblers other than Ford. For now, it enlarges KBE’s customer base; the outcome for Visteon, however, is yet to be decided.

Although the sections act independently from one another, they do collaborate extensively to develop new products for its customers and to learn from different experiences and strategies. The Department is still small enough where it can keep the divisional walls from restricting information flow and informal contact.

### 3.4 KBE Strategy and Philosophy:

KBE “develop[s] and deploy[s] integrated Knowledge Based Engineering in key design, engineering, and manufacturing applications, to give Visteon a measurable competitive advantage in terms of cost, quality and speed.”<sup>24</sup> KBE evaluates different customer issues, at various levels, in order to create tools which can complement, rather than conflict with, the environment that they will be used in.

KBE develops software tools to accommodate its customer’s needs without ever seriously affecting the existing process used by the customer. It does not consider itself to be an in-house, strategy consulting organization; it does not develop software tools that enforce process changes. KBE’s primary mission is to develop tools that will help optimize the present processes. As was explained to the author, KBE does not have the resources and the customer support to make such system-level changes.

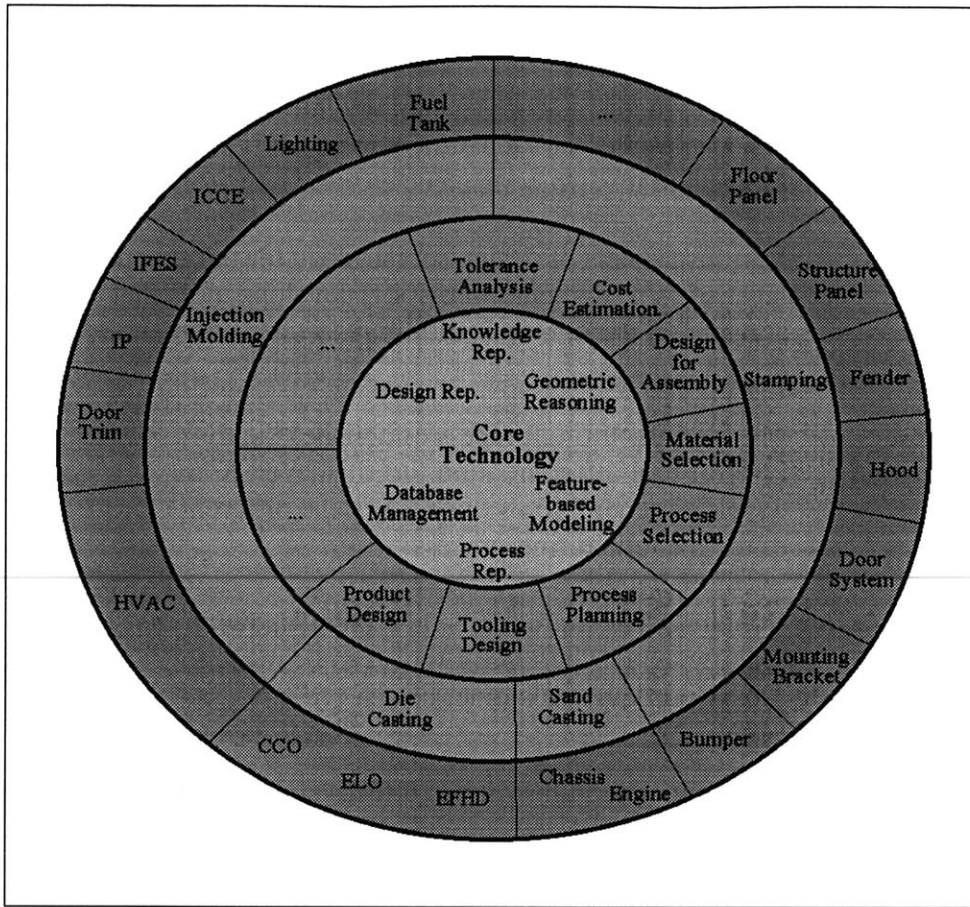
In addition, KBE has to be sure that its tools are flexible enough to conform to ever-changing conditions. KBE is definitely concerned with the product life of each its tools, and for obvious reasons. If each tool has a product life that is shorter than its development life, then KBE puts itself in a difficult position:

- The customers may lose confidence in the product or with KBE’s work.
- The number of requests that come in to KBE to modify existing tools may grow at an overwhelming rate. (This would certainly be the case if KBE provides a long-term service for each of its customers, after deployment.)
- The chance of reusing a tool for different projects to accommodate different customer needs may become an unlikely option.

The Department looks at collecting raw requirements in order to fashion a pathway for each of its applications. The figure below (Figure 3.3) shows how KBE’s methodology is layered to support the tools that it develops. (This figure was recreated with permission from KBE.)

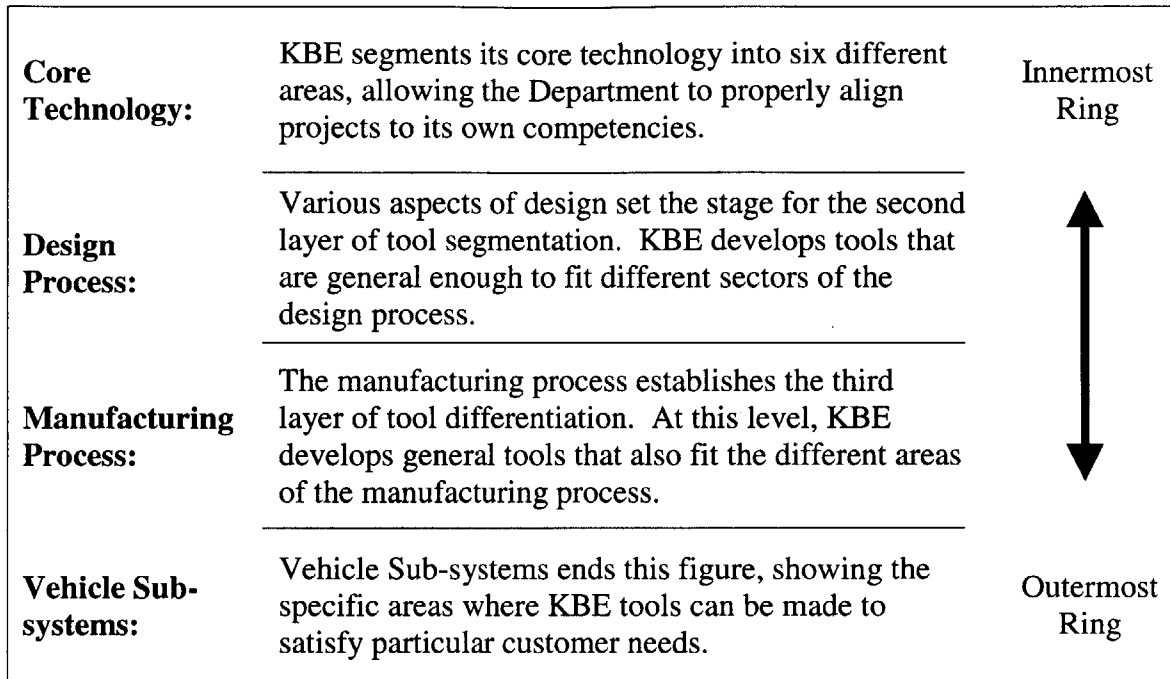
---

<sup>24</sup> Quotation came from a Ford/Visteon internal web page—KBE’s business web site.



**Figure 3.3: Segmenting KBE Knowledge**

Figure 3.3 shows how KBE knowledge is broken down and aligned with Ford’s product development process. The figure is shown in four layers (or rings)—KBE Core Technology, Design Process, Manufacturing Process, and Vehicle Sub-systems—each of which are described below:



KBE uses this map to better understand how it can effectively position itself for greater success. Although it is very general, it does provide KBE with a certain level of direction in which it can follow. In particular, the map helps to answer the following questions:

1. Where has KBE's focus been presently (based on every layer of Figure 3.3) and where would it like to change its focus in the future?
2. What types of projects does KBE need to promote more of in the future in order to develop a broader product portfolio for itself?
3. Where can projects and tools be integrated for effective utilization of KBE's time and resources?
4. How can KBE make tools that are aligned with its customers' needs and with its own core competencies?
5. What is the best way for KBE to use those core technologies that have been dormant?

Generally speaking, KBE develops two types of tools: one type that is for general evaluations in design and manufacturing and another type that is specific to vehicle sub-systems. In many cases, KBE incorporates general-type tools into its specific-type tools to give its customers a "complete" evaluation package that can serve their needs even better. The author will speak about the two classifications of advisory tools later in this chapter; but the point to be made here is that KBE looks to integrate its different modules to give its customers custom-made software solutions. This is a major part of KBE's strategy, and its strategy map, as Figure 3.3 attempts to show.

### 3.5 Process for Developing a Typical KBE Advisor:

KBE goes through a detailed process of identifying customer requests (which support its own resources and goals); acquiring the right amount of knowledge, both raw and deductive knowledge, for each project; and properly deploying the tools for expeditious use. The total process can be generally described using an 8-step approach:

1. **IDENTIFYING** the general needs of the customer
2. **DECOMPOSING** those needs into specific requests
3. **FILTERING** the customer requests (i.e., performing a feasibility study on all requests to determine if they are within KBE's scope)
4. **CONVERTING** the remaining requests into the framework of an advisor type
5. **ACQUIRING** pertinent information from the design guides
6. **SORTING** the information into different categories with the help of an assigned customer team. (The customer team comes from all functional aspects of the design/development process. KBE normally calls this team the cross-functional team.)
7. **DEVELOPING** a knowledge hierarchy (with the help of the cross-functional team)
8. **LINKING** the information using conditional statements (again, with the help of the cross-functional team)

Many of the challenges that KBE faces come from trying to do each of these activities as quickly and efficiently as possible. In many cases, certain steps have to be repeated so that the project in question is ready to pass through each stage of development. More will be said about the limitations and challenges that KBE confronts each time it does a project in Chapter XII, KBE Tradeoffs and Pitfalls.

The eight-step process mentioned here is nothing more than a generic template that the author constructed based on the information provided by KBE managers and software developers. Because most of the projects taken by KBE are manageable for a team of two to three people, which can be worked on in a period of one year or less, many of these developers see the projects through from beginning to end. As a result, they provide a rich source of information for how a typical project is done. The description to follow for each of the eight steps is based on information collected from discussions with these developers.

**IDENTIFYING**      General requests come in to KBE from customers who wish to improve a particular part of a process that they are responsible for. In many cases, the requests themselves are seldom structured for a quick assessment (i.e., feasibility test) by KBE. The requests are collected at this point, prepared for the decomposition phase of the process.

- DECOMPOSING** At this stage of the process, KBE interacts with the potential customer so that both understand each other, and the goals for the request. It is the responsibility of KBE to help the customer frame the request in a way that can be effectively evaluated. KBE does this by explaining to the customer what it is able to do—the types of projects that it normally works on, the size and scope of these projects, etc. If the goals of the request are not aligned with KBE’s goals and strategies, then KBE will be reluctant to do the project.
- FILTERING** Each request is structured for evaluation and then put through a feasibility test.<sup>25</sup> The test takes into consideration factors like project-development time, alignment with KBE core strategy, and available resources, to name a few. If the project is not feasible at the time of its evaluation, then the project will be “filed away” until the Department is in a better position to actually complete the project, in a reasonable amount of time.
- CONVERTING** If the process is considered feasible, a formal strategy is developed and a development team (i.e., cross-functional team) is formed. The framework for the potential tool is also drafted using the core knowledge from KBE’s viewpoint.
- ACQUIRING** At this stage, the “raw” information needed for the project is obtained from design manuals, and other types of resources. The technical experts from the customer side of the development team are asked to interpret the information contained in these manuals and to help fill in any missing information. (Normally, the manuals do not supply enough information for KBE’s needs; that is why a functional team is placed on each of these projects. More will be said about what the manuals supply and do not supply in Chapter V.)
- SORTING** At this stage, the “raw” information is sorted, leaving behind any material that is not relevant to the project or that cannot be used in an advisor environment. This sorting process is naturally done with the help of the technical experts from the customer side.

---

<sup>25</sup> As the author understands it, the process for configuring each request for a feasibility test is not very structured. As the reader will see in Chapter VII, KBE does not have a generic set-up in place for extracting key aspects from each request and seeing if they align themselves well with KBE’s own goals and skills.



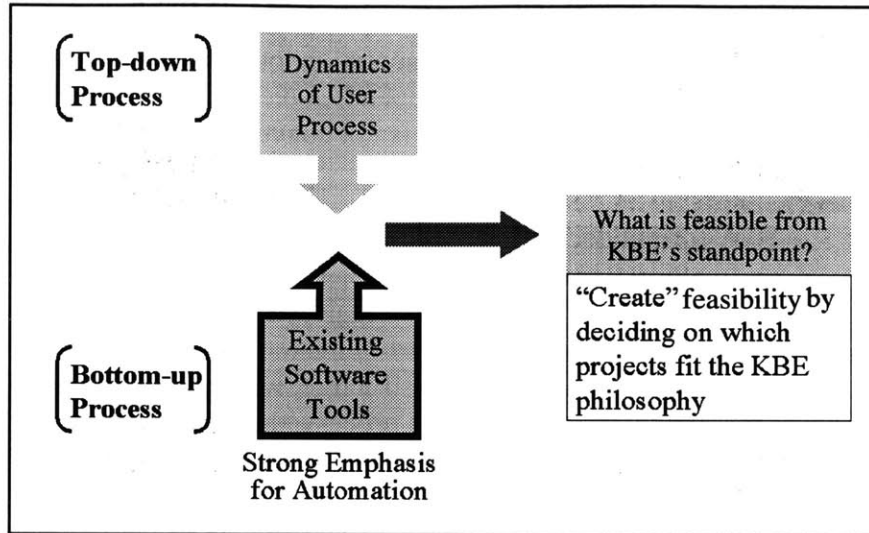
**DEVELOPING** At this point, the “raw” knowledge is put into a hierarchical structure for later implementation into the tool’s knowledge base. This work is primarily dependent on what the experts have to say. Most of this information is not conveyed in any of the design manuals.

**LINKING** A formal decision tree is made with the help of the cross-functional team. The “tree” consists of conditions that need to be considered at every level. Again, the technical experts bring in this information. This part of the tool’s knowledge base is considered the inference structure, which was discussed in the previous chapter. In this process, the heuristics are gathered and organized based on the expert team’s understanding of the various conditions that must be considered during actual design.

### 3.6 Top-down/Bottom-up Processes:

In many cases, KBE uses a combination of top-down and bottom-up techniques to determine the scope and direction of a project. The top-down process, as defined here, includes all possible ways to enhance concurrent-design practices. This requires that KBE first identifies and understands the system dynamics of the user’s design environment and then develops and incorporates a KBE system to complement it. The top-down process also looks to see how an “expert” system will affect a user environment, before a project is completed. The cross-functional team helps KBE to see all of these different conditions as tool development begins. “Top down” tools, as the author describes them, are aimed to encourage cross-functional communication, product analysis from an architectural level, and interactive learning, to name a few. The bottom-up process, on the other hand, is used whenever existing parts of the design process are slow and cumbersome. KBE develops many of its tools to optimize existing design procedures so that faster iterations can be done; however, they may not necessarily improve overall design performance or provide the type of flexibility needed to “absorb” different design and packaging conditions. Automating methodical steps, which are dispersed throughout a design process, is one way of applying bottom-up design principles. As will be described later in this chapter and in part of Chapter VII, the bottom-up approach can lead to some unexpected and undesired results.

Figure 3.4 attempts to show this two-in-one process and some of the key bits of information extracted from each approach.



**Figure 3.4: KBE Approach to Software Development**

The two processes bombard one another, resulting in a power struggle for supremacy. (Based on the author's evaluation of different KBE tools and methods, the bottom-up process seems to be considered the mightier of the two. For the most part, KBE looks to automate existing parts of a process whenever possible in order to significantly improve design time. Work like this can provide so-called quick solutions to problems, something that KBE always looks for in every project it considers. More will be said about this later in this chapter and in Chapter VII.) Regardless of which process "wins" this "battle," each customer criterion must conform to the pre-established conditions set forth by KBE. If the criteria are flexible enough to be shaped within this structure, then the tool can be made.

Each process—top-down and bottom-up—will be discussed in the following subsections—even though KBE almost considers them simultaneously. These processes are both based on the user's environment, not on KBE's.

### 3.6.1 Top-down Approach:

In order to develop any type of "expert" advisor that accommodates the customer's request and complements the end user's environment, KBE must first follow and understand the existing user process. To do this, KBE interacts, on a regular basis, with its customers, defining the requests in detail; acquiring essential pieces of "raw" and deductive information to be stored and managed in knowledge base; and keeping the developing projects in check with the needs of the users. (At the beginning of every project, KBE develops a cross-functional team, from the customer group to help perform these activities.) KBE emphasizes strong customer support, to complement a pull-system strategy.

KBE relies on the experience of its customers to see where a good fit can be made. Heavy customer interaction is crucial at this level because, in most cases, the overall design process or particular pieces of the design process are not normally documented. (If a process does not follow a systematic structure, for whatever reason(s), then it may be very difficult to describe in an instruction-type format. This is one reason why design processes are seldom seen in design manuals.) For this very reason, KBE has not been able to attack this level of tool development with the same kind of consistency and order each and every time it works on a new project.

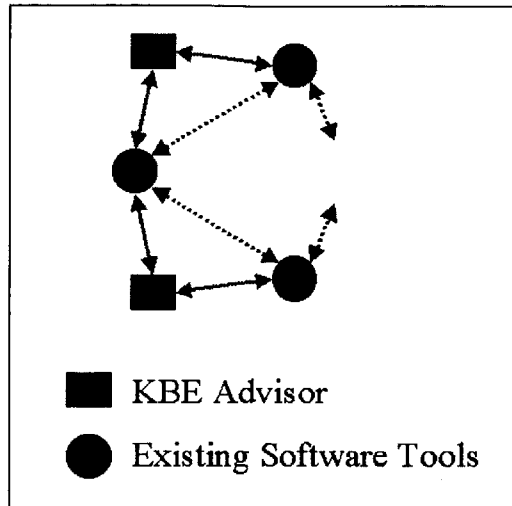
At this level, KBE sees where its tools can fit into the existing user process with the minimum amount of user confrontation. KBE does this by first gathering the necessary logic “hidden” behind the actual process and then using that knowledge to construct an information-based flow diagram. The flow diagram, at a relatively high level, shows who is involved in the process and at what level; how information is passed from level to level; and what conditions need to be in play for information to pass.

### 3.6.2 Bottom-up Approach:

Conversely, KBE also looks at the user process from a lower or a more detailed level of development. When going through this stage of development, KBE looks for answers to particular questions, such as:

- What tools are already at the user’s disposal?
- Where is there repeatability in the process?
- Can an existing KBE tool be used to give the customer what he wants, rather than going through the long process of creating a new tool?

KBE advisors are developed to help strengthen the existing links between different design and development tools. Figure 3.5 attempts to show the general idea behind KBE’s bottom-up approach. The dotted lines, connecting software tools to each other, represent the “weak” links, which KBE advisors intend to enhance. In many cases, these “weak” lines characterize manual processes (e.g., manually entering data tabulated from one tool into another for further evaluation). For many of its projects, KBE tries to find ways to automate existing manual processes (ones that can be considered repetitious) and to link the intended advisor to the CAD tool, using parametric modeling, for instance.



**Figure 3.5:** Bottom-up approach used by KBE

In situations where links are “open ended,” such as when needed information can only be gathered from human insight, KBE is limited. (Figure 3.5 shows this with arrows connected only at one point.) This goes back to the idea that expert tools are limited to only information that can be shown in a structural framework.

In many cases, KBE projects focus on the process of automating manual procedures, which are highly repeatable and require little deductive reasoning. By automating many of the procedures, the customer, in most cases, can see immediate results: design time is reduced because the computer algorithm, inside the advisor, processes the information faster than the user can. This also allows the user to go through several more design iterations, thus providing an error-free or near error-free design to the people at the next level of development.

In addition, KBE also emphasizes parametric modeling whenever packaging is an issue in the design process. As part of a parametric model within CAD, “hidden” relationships (usually in the form of mathematical equations) exist between features on parts and parts in an assembly to promote quicker design changes. These parametric links can include any combination of points, lines and surfaces on parts. For instance, consider two parts in space with a defined link. The link states that the normal distance between these two parts is always a certain fixed distance. If one part is then moved in CAD space, then the other part will move accordingly so that the defined parametric link is followed.

Parametric modeling can be a constructive tool for designers, allowing the advisor, rather than the designer, to make immediate changes to a design once an initial change has been made by the designer. However, if used without caution, parametric modeling can also limit designer use. In other words, too many parametric links can over-define a model and limit the type of design changes that are necessary for various exogenous conditions,

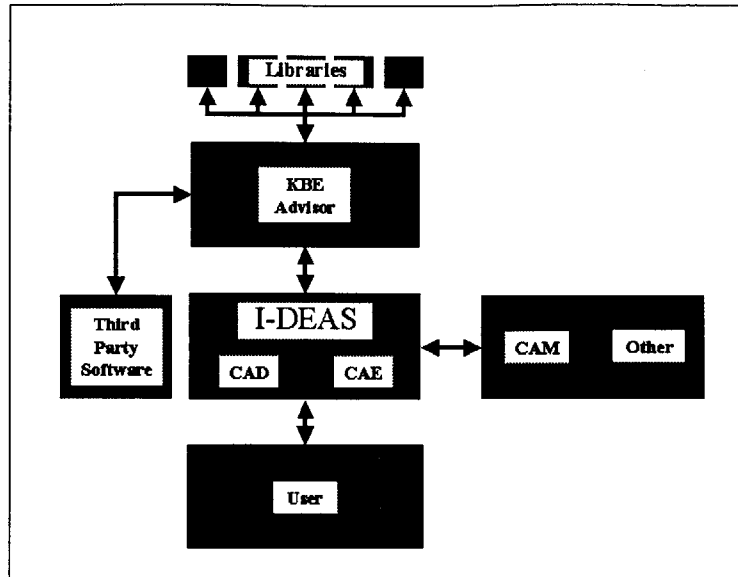
like packaging and assembly. The bottom-up approach comes into play when the KBE development teams fail to fully understand how far to parameterize different CAD models. The teams have to keep in mind, as they expand the models, system-level conditions—conditions that are most probably not defined mathematically. Each development team has to understand the extremes, in which the model may be stretched, twisted, increased or decreased in overall size or any combination of the three, so performance is not hindered in any way. Extreme parametric modeling may provide the designer with only certain pathways to follow, such as only changing the overall size of the model. The top-down approach would consider both detailed rules and system-level dynamics so that a balance is struck between user flexibility and rapid design changes. The following example shows how tools can be made obsolete by considering only defined, mathematical parameters.

The KBE-HVAC team—which develops advisory tools to help promote the easy design of air-handling systems (HVAC units)—was bombarded by this issue. The original KBE-HVAC advisor (Version 1) was highly parametric, leading to the belief that this tool would significantly reduce design time. However, what the team later recognized, with feedback from the customer, was that this version would not be able to work well in an environment where package space is constantly changing in shape and size. In addition to that, the designers who were using the system wanted to have more flexibility. (In many cases, experienced designers are cautious of tools that drive the user down a particular pathway. Designers of this sort like to use tools that enhance their work without inhibiting their creativity.) As a result, a second version was made, which had far fewer parametric links to the CAD models and allowed the user more flexibility with the design. (More will be said about the KBE-HVAC team’s findings in Chapter VII.)

### 3.7 Structure of a Typical KBE Advisor:

The general structure of a typical expert system was briefly described in Chapter II. This chapter will speak about the common structure of a typical KBE advisor, and how it is linked to the user’s CAD package.

Figure 3.6 shows how a KBE advisor is linked to I-DEAS—the CAD/CAE suite that most Ford and Visteon designers and engineers use—and to third-party software packages. The advisor, itself, is also linked to a number of libraries, which store the “raw” information (e.g., rules, guidelines, assumptions, etc.) necessary for the advisor. The advisor contains the “logic” (i.e., the decision tree), which works in conjunction with the libraries’ “knowledge.” The user interacts with the advisor through the graphical user interface (GUI) provided by I-DEAS.



**Figure 3.6:** Structure of a KBE Advisor

KBE develops these links (as was mentioned in Section 3.6.2) and customizes the CAD/CAE interface in order to support specific functions held by the KBE-linked advisor. The standards for incorporating new icons into the existing I-DEAS framework are rigidly set and tightly followed by KBE. (Generally speaking, the standards describe how new windows and icons should be designed in order to “complement” the appearance of I-DEAS’ user interface.) KBE strongly feels that any additions or changes to the existing user interface must be tightly knit with the user’s CAD/CAE environment—and it feels this way for the following reasons:

- To give the customer the perception that the advisor is actually part of the I-DEAS software tool.
- To provide the user with the same level of comfort that he or she had with the user interface before the advisor was linked to I-DEAS.

Each KBE advisor has three areas where flexibility can be either enhanced or sacrificed: the libraries, the decision-tree structure, and the open interface to different third-party software package(s). KBE uses a plug-in/plug-out architecture when developing its advisors in order to support the various changes and demands that they will undergo. In particular, this type of architecture is useful for adding, removing or modifying libraries.

KBE strongly emphasizes not only “complete”<sup>26</sup> solution packages, but also flexibility in order to customize the tool so that it can accommodate the user’s changing requirements

<sup>26</sup> A “complete” software tool is one that provides a thorough solution to a problem, or at the very least the necessary material needed for the designer to make a critical decision. An example of a software tool that does not provide a “complete” solution is one that gives general suggestions, without any quantitative information to back up its recommendations.

and needs. This is where the tool can become quickly obsolete. (This is one of several important issues that will be discussed in Chapter VII.)

Although this type of computer architecture is modular by nature, it does have one particular flaw: it cannot support changes to the decision-tree, without reconfiguring the computer algorithms. If the stream of decisions recorded in the advisor no longer agrees with the actual process used by the end user, then the tool, itself, becomes obsolete. This issue also makes it rather difficult for KBE to develop generic suites for customers inside and outside of the Ford/Visteon community. (KBE is now experiencing this challenge, as it begins to sell its products outside of Ford and Visteon.) The only suggestion that the author could provide is a system that allows the user to develop his own decision tree by “asking” him certain questions about the design process; a precedence relationship can be constructed from the answers to these questions. However, at this time, the author does not know whether this type of architecture can be developed. It would require first understanding the common and uncommon characteristics of each decision process over time so that the right questions can be asked.

### 3.8 Types of KBE Advisors:

The type of advisors that KBE develops and the general characteristics that distinguish one type from another will be discussed in this section.

Many of the KBE advisors can be classified using three primary factors:

1. Is the advisor capable of performing generic tasks for a wide range of products (process dependent) or specific tasks for certain products (product dependent)? (GENERIC VERSUS PRODUCT SPECIFIC)
2. Is the advisor linked or not linked to the CAD modeler? (ASSOCIATED WITH GEOMETRY VERSUS NOT ASSOCIATED WITH GEOMETRY)
3. Is the advisor an evaluation tool (dynamic tool) or a tool that generates features on parts that conform to standard guidelines? (EVALUATION VERSUS GENERATION)

The following chart (Table 3.1) shows some of the KBE tools that have already been developed and deployed, with their respective distinguishing characteristics.

Advisor Name:	Generic/Specific		Association with Geometry		Evaluation/Generation	
	Generic (Process Dependent)	Product Specific (Product Dependent)	Associated with Geometry	Not Associated with Geometry	Evaluation Tool	Tool which Generates Geometry
Design Advisor for Die Casting	X		X		X	X
Design Advisor for Injection Molding	X		X		X	X
E-MDRC (Electronics Manufacturing Design Rules Checker)	X		X		X	
Expert Gating		X	X		X	
KBE-AIS (Air Induction System)		X	X		X	X
KBE-Fuel Tank		X	X		X	
KBE-HVAC (Heating, Venting, Air and Cooling)		X	X		X	X
KBE-ICCE (Interior Climate & Comfort Engineering)		X	X		X	
KBE-IP (Instrument Panel)		X	X		X	X
KBE-Lens Synthesis		X	X			X
KBE-Lighting		X	X		X	X
KBE-Stamping	X		X		X	X
KBE-Steering		X	X		X	X
KBE-Wiper		X	X		X	X
Processing Advisor for Injection Molding	X			X	X	
Shape Tool <sup>27</sup>	X		X		X	X
Stretch Analysis <sup>28</sup>	X		X		X	
Cost Estimation	X			X	X	
Tooling Feasibility	X		X		X	

**Table 3.1:** Characteristics of Various KBE Tools

<sup>27</sup> The Shape Tool is used to predict the reliability of solder joints on printed circuit boards for electronic products.

<sup>28</sup> The Stretch-Analysis Tool is used to identify the high stretch areas that may cause vinyl tearing on door trims or IP covers.



KBE makes every effort to develop tools that can be reused across different projects, and whenever possible, linked with other KBE tools. In the future, generic tools, like the process advisor for injection molding, will have the ability to be linked to product-specific tools, like the KBE-HVAC advisor, allowing KBE to deploy complete suites.

Table 3.1 is a very useful chart because it tells the reader what type of advisors KBE emphasizes and what associations exist between each of the three categories. In addition, there are certain development procedures that KBE naturally follows, based on the customer's requirements and the structure-type for the advisor, and these are implicitly shown in the chart.

When an advisor is generic and not directly associated with any geometry, there is a tendency to make the advisor a web-based tool. In many situations like this, people who do not use CAD software to do their work (e.g., product engineers and manufacturing engineers) normally use this type of advisor. For instance, the Processing Advisor for Injection Molding is an interactive tool used by manufacturing engineers and tooling specialists to “troubleshoot” problem areas. The Cost Estimation tool is another type of web-based tool used by product engineers to determine material and labor costs (among other things), which may prove useful when negotiating with outside contractors.

When the advisor is intended to be generic and directly associated with various geometric models, KBE moves cautiously while keeping the project scope in constant check. Projects that have these types of initial parameters can quickly and easily grow into time-consuming assignments, if not watched over carefully. KBE makes sure to look at a few selected aspects of a process which can be evaluated in closed form (i.e., very few exogenous variables enter the evaluation, such as many of the system-dynamic phenomena that occur during injection molding.). KBE understands that in order for its customer to be pleased with the product, it has to provide a total solution—even if it means looking at one or two stages of the entire process. For instance, the Design Advisor for Injection Molding focuses on four key aspects of injection molding:

1. Identifies edges having insufficient draft angles
2. Identifies edges having insufficient rounds (fillets)
3. Conforms to changing standards (i.e., minimum draft angle can be changed by user)
4. Regenerates CAD model to conform to standards

However, the tool does not look at larger issues—issues that require an expert to be on hand to determine the efficiency of an existing design. The author lists a few of these larger issues below<sup>29</sup>:

---

<sup>29</sup> Refer to <<http://www.cadtek.com/moldflow.htm>> to learn about Moldflow, an injection-molding advisor made by Cadteck. Moldflow is able to simulate the flow of any number of plastics and provide suggestions to recognized problems.

- Regenerating a design to reduce the number of sink marks during actual cooling
- Regenerating a design so that cooling time is reduced
- Estimating the cost of an injection molded tool for a designed part
- Determining the location of weld lines.

KBE has developed in what seems to be an even distribution of product specific and generic tools; however, it is obviously clear that it also makes tools to be dependent on geometric models. Most of the tools that it makes are in fact for designers, making another point—most designers use tools that can “interact” with their CAD models. If KBE was to develop an assembly advisor that was generic and associated with CAD models, then it would have no other choice but to focus on particular aspects of assembly (e.g., interference, assembly sequence, fastener types, etc.)

In addition, most of KBE’s tools not only evaluate a given product design but also help modify or regenerate it. The question now becomes, does KBE intend to make an assembly advisor which can do the same thing? This depends on a number of issues, including in what part of the design process does KBE plan to implement this tool and what type of dependency (process or product specific or some combination of the two) will the tool have? Regeneration may require the tool to be deployed at a later stage in design than evaluation. Parts need to be fully defined in CAD space before a tool can regenerate them for assembly; this also means that the advisor would have to contain information about the model (e.g., surfaces, edges, holes, pins, etc.) before regeneration could even be considered. More will be said about this in Chapter IX.

### 3.9 Chapter Summary:

Chapter III discussed the requirements for knowledge-based tools in the Ford/Visteon user environment. The chapter began with a discussion of how these tools became such an influential piece to Ford/Visteon’s new development strategy. Two conditions, in particular, were identified and explained as reasons for this wave of development—less time on individual projects and more job segmentation. As these circumstances naturally emerged, the Organization, as a whole, found itself having a difficult time locating and retaining essential pieces of information—information which was scattering all about as fast as its people were rotating job assignments. At present, Ford/Visteon is now developing interactive databases, or knowledge-based tools as they have been called here, to provide its employees with the right knowledge at the right time. Essential pieces of information can now be contained and managed for interactive learning and proficient design activities; the question is, how far can these tools go before they reach a natural set of limitations? Generally speaking, the natural limitations ingrained within present “expert” tools are all characteristics that fall short of human understanding. They include, describing many of the ideas that come from assembling a physical mock-up of a design (e.g., the actual feel of attaching two or more parts together); developing learning algorithms that work in the same way that humans learn, etc.

The next part of this chapter was devoted to how Visteon's Knowledge-Based Engineering (KBE) Department tackles some of the issues briefly discussed in Chapter II, using some examples provided by KBE. In particular, this section of the chapter discussed the Department's philosophy and strategy for taking on and developing certain projects. It was noted that KBE evaluates each project at different levels in order to see how well they fit the Department's approach to development and marketing. Right now, KBE is at a position where it has more projects than it can handle, and must be selective with the projects that it takes. As a result, the KBE must see to it that there is synergy between the project requests and the Department's core philosophy and strategy.

The remaining part of this Chapter talked about the common structure of these tools and the different characteristics, which set them apart. In most cases, each KBE advisor is linked to a CAD/CAE tool. (In cases where it is not, the advisor is merely a web-based tool.) In general, a KBE advisor is used to extract "raw" information from its libraries during any needed time; provide a bridge between the CAD/CAE tool and third party software packages; and deliver information to the user in a familiar user environment.

Each KBE advisor was then segmented and looked at in three different ways: (1) as a tool that is process dependent or product dependent; (2) as a tool that is associated or not with geometry; and (3) as a tool that evaluates or generates geometry. These characteristics are all considered depending on a variety of circumstances—the type of user and the type of resources and information available to KBE to develop the tool. The structure and characteristics described here will be used to help understand what type of framework is feasible for an assembly advisor.

In Chapter II, the author discussed the framework of an advisor; in this chapter, the author described how KBE develops expert systems for particular design applications and how it goes about getting essential information, which is necessary, for each tool's structure. The next chapter will go a step further: it will discuss what type of information exists in the world of design for assembly and what can be taken from this to make an successful assembly advisor.

# CHAPTER IV

---

## GENERAL DFA OBJECTIVES

### 4.1 Roadmap for Chapter

This chapter will discuss different aspects of design for assembly (DFA) and how it is used today. In general, the following topics will be discussed:

- What is design for assembly and how does it work in the design environment?
- What are some of the limitations with DFA?
- How can DFA be segmented, categorized and placed within product knowledge?
- What is the role of product architecture in assembly design?

The purpose here is to see what has already been done in the area of DFA; what existing information can possibly be incorporated into an assembly advisor; and what areas of DFA need to be explored and developed further in order to make assembly analysis more useful.

### 4.2 Setting the Stage for Design for Assembly

The Industrial Revolution and the introductions of both mass production and the assembly line all promoted, in their own way, the idea of “baking” assembly into the earliest possible stages of design. (The name most commonly used nowadays to describe this process is Design for Assembly, or simply DFA.)<sup>30</sup> However, the process of formally collecting, classifying, and quantifying this type of information into a useable format has only been done, with serious consideration, within the last few decades.

---

<sup>30</sup> In an article by G. Boothroyd, “Design for Assembly—The Key to Design for Manufacture,” the author makes the claim that Design for Assembly may have formally originated before or during the Renaissance Period.

Generally speaking, DFA rules have existed to help designers, at the very least, think about the general issues that affect assembly during the initial design of a product. By thinking about such guidelines as assemble from above, use standard fasteners, and reduce the total number parts to be assembled, the designer automatically helps develop assembly-efficient products. In balance with other issues, DFA can also be used to improve a product's quality, manufacture-ability, and assemble-ability, to name just a few.

Many product-development companies are now looking at DFA to help analyze product designs during the concept-development process. Many available software tools help to determine the assemble-ability of a product and give suggestions where needed before a physical prototype is ever made. In addition, they allow designers to think about design in a different way, which can be used when designing new products or parts to products.

However, many of these tools rely on information that comes from a detailed design and a selected assembly sequence. As a result, any type of suggestion given by a tool would be used to "correct" any problem areas, from a "local" level. (For example, use snap-fit fasteners instead of screws.) The advisor would not be able to evaluate a given product architecture, from the standpoint of assembly, and provide suggestions to improve the design or suggest an alternative design. To have a tool make both local and global decisions would require the tool to have "knowledge" about the product itself. It would have to be able to answer questions, such as:

- What are the primary drivers for the design, and how do they affect the design?
- Where does the designer have the flexibility to make necessary changes?
- What are the effects from making changes to a design?
- What design changes have more reason to be implemented than others? (In other words, the advisor must understand what issues have precedence over others.)

All of these ideas will be described in more detail throughout this chapter. Although this chapter is saturated with information, there is still even more that the reader can learn about by reading a number of different books, journal articles, and research papers on the subject. The author provides a list of literary resources in the Suggested Reading Material Section; some have been cited throughout the thesis to convey particular thoughts.

### 4.3 What is Design for Assembly and How Does It Work in the Design Environment?

Design for Assembly (DFA) "is based on the primary principles of maximizing assembly efficiency and minimizing product complexity in the early stages of design."<sup>31</sup> Design

---

<sup>31</sup> Definition provided by a Ford internal web page

for assembly, for the most part, is a loosely structured methodology, consisting of nothing more than simple design rules, for evaluating existing designs and determining where improvements can possibly be made. The rules, themselves, act independently from one another and from the product that they are intended to evaluate and improve. It is up to the designers to use these rules in the most effective manner, which they see possible, for improving their designs.

The majority of DFA guidelines found in books, manuals, and journal articles associated with this topic, state the following, in some form or another:<sup>32</sup>

- All DFA procedures address part geometry and pair-wise, inter-part spatial relationships

The fact that most DFA knowledge is applied to part geometry—rather than to sub-system or system-level “geometry,” for example—makes this practice conducive to bottom-up design and analysis. When DFA is used in this way, problems can arise immediately or over time and in ways that are unexpected. For example, assembly designs that consist of several mating relationships per part will most likely require the construction and analysis of a prototype: for complex assemblies, it is very difficult to see how slight changes at the part or feature level can influence the whole assembly. Assembly problems can almost certainly be found in vehicle sub-systems, like instrument panels (particularly registers and airbag modules) and car doors that require two or more surfaces to be flush with one another.

- Most DFA methods emphasize reducing part count

Most DFA methods emphasize part consolidation and, at some level, an integral architecture. Part consolidation does reduce part count (and with that, assembly time and cost) but it also drives product development down a narrow pathway. DFA has a tendency to make designers think about circumstances revolving around individual products, rather than around groups of similar products (commonly called product families). But this could be remedied in several ways; this chapter and Chapter IX will spend some time dealing with this issue.

- Most design suggestions are based directly on different types of assembly methods—manually-, automatically- or robotically-driven operations.

Most DFA suggestions are intended for improving production efficiency at the factory floor (i.e., time and cost for manufacturing and assembling products); however very few of these suggestions are made to improve the level of efficiency at the design level. Many DFA tools and procedures lose sight of how design changes can affect the scope of a product family and the time for implementing new products. Therefore, design for assembly should consider how design changes can both improve and inhibit every level of development, from design to fabrication, and it should consider these issues equally.

---

<sup>32</sup> Refer to Redford and Chal, Design for Assembly: Principles and Practices (1994)

## 4.4 Limitations with Design for Assembly

Although current DFA practices are becoming influential parts of the design process within product-development companies around the globe, they do have their limitations. Many of them can be associated with a lack of knowledge about different products and a lack of system-level understanding about different processes. The following list shows some of the more specific issues regarding the gap between DFA and actual assembly:

- No attempt has been made to tackle the problem associated with assembly organization or concept design. (An assessment, using current DFA methods, cannot be made without a detailed design.)

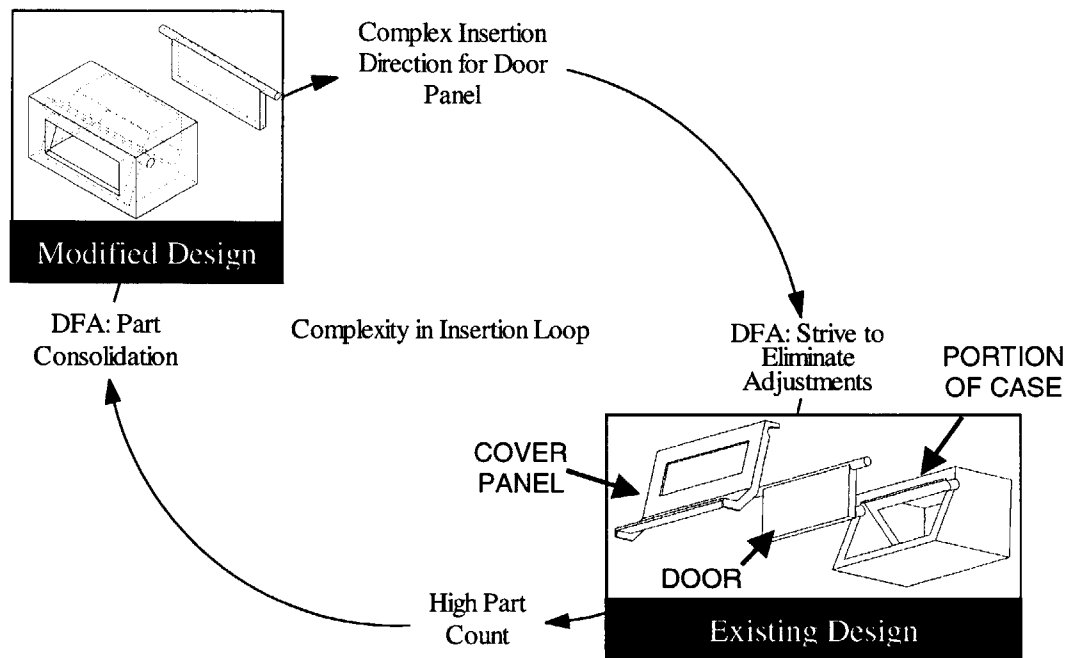
Current DFA methods are used mostly to evaluate rather than instigate design changes for the purpose of assembly. The way that DFA is presently fashioned, recommendations are delivered only after a detailed design is presented. These recommendations are normally provided in some type of quantitative form so that every evaluation is done with a high level of repeatability.

DFA, at this time, is weakly linked to the earliest stages of design—the concept stage. Most, if not all, designers will say that they keep assembly in mind when designing parts, but very few will ever claim that they follow a particular strategy for designing their products for assembly. Design for assembly, at the concept level, mostly consists of nothing more than general design rules (Refer to Figure 4.4 for a list of these rules); it is up to the designers to determine where and how to use these rules in the framework of their designs. This information—the knowledge that designers use to take these static rules and use them in real applications—needs to be the main topic of consideration if DFA is to be made to accommodate any level of proactive design.

- Information Loops—where one suggested action can bring the user back to another needed change. DFA does not comment on some of the conflicting issues associated with assembly guidelines.

As with any method that focuses on improving a particular manufacturing process, design for assembly can bring its user down a pathway that either makes little or no sense or repeats itself without end. Because design for assembly, as a method, has no way of understanding all of the issues associated with design and how they interact with assembly, it is up to the user to make sure that every decision made is in check with the product's function and physical constraints. The rules themselves can at some level conflict with one another, when used in various applications. It is the designers' responsibility to know which rules are important; when they need to be considered in a design; and which rules can simply be eliminated without severely affecting the quality of assembly. It is also up to the designers to know when to “break” information loops (i.e. a combination of suggestions that force the designer into a never-ending process of repeated decisions) when they exist by understanding

the key tradeoffs that allow the loop to propagate in the first place. The following example attempts to explain information loops a little more clearly.



**Figure 4.1:** The “Complexity Insertion” Loop

Figure 4.1 describes one of the pitfalls with the DFA theory of part consolidation and its impinging effect on the rule of easy insertion. In essence, this diagram shows the conflict between two different rules: one which emphasizes part separation and another which emphasizes part consolidation. This causal loop structure was developed using the HVAC design fitted for the 1999 Ford Windstar. (More will be said about this unit and similar units in Chapter VI.) The loop focuses on the existing design of one of the door panels and its interaction with the main casing. The existing design for assembly involves the casing, the door panel and a cover mount. The loop shows that by integrating the cover mount with the main casing, the door panel’s entrance into its assigned compartment becomes a complicated operation. Complexity in insertion, in this particular case, is due to two main factors: (1) the shaft must now be aligned with and inserted into two holes from the inside of the casing, and (2) the panel might have to be positioned and repositioned several times in order to pass through a small opening used to restrict the panel’s range of motion.

- Suggestions do not consider assembly balance, material constraints, product development time, intended functional requirements, and many other issues.

Current DFA theory is very focused on assembly time and cost (issues relevant to actual fabrication); however the theory never really considers other facets of product development. For instance, design for assembly does not consider how its



recommendations affect design time and product portfolio management. This is due to the fact that DFA evaluates designs on an individual basis; it is up to designers to consider other products within the questioned product's portfolio so that knowledge can be shared and utilized effectively. It is not difficult to see that DFA, if used blindly, can lead designers down the path of sequential product development—and ultimately to the process of excessive reengineering.

In addition, DFA has no way of identifying and connecting all of the factors associated with a product—for example, functional requirements; physical conditions external to the product; the natural sequence for assembly; the strategic placement of parts into the assembly so that assembly operations are quick and easy (i.e., assembly balance); product life (resulting in design for disassembly); etc.—and seeing how its recommendations affect all of them.

- Guidelines are not presented in any hierarchical fashion. In other words, DFA does not expose any explicit or implicit links between different sets of suggestions.

Because DFA is weakly linked to product knowledge, it provides suggestions that are not very specific and approaches each product rather conservatively, by looking at the part and components levels. In addition, the rules are not ranked according to how they affect different product architectures, methods of design, manufacturing and assembly, and constraints, to name a few. The remaining sections of this chapter show what type of hierarchy can be developed to evaluate existing rules and to show where more emphasis needs to be placed. Chapter IX will show in clearer detail how links between the traditional methodology and the methodology emphasized here (system-level design) can be made.

#### 4.5 Motivation for Placing Assembly Knowledge in a Hierarchical Structure and Its Role in “Design Space”

For the most part, it is essential to determine which rules can be used merely for the purpose of making a suggestion (low association with overall product design) and which rules can be used for driving a design down a particular pathway, such as integral product design (high association with overall product design). DFA guidelines can take two forms: (1) is the rule general or specific, qualitative or quantitative, etc., and (2) is the rule really conducive to the product that is being analyzed? In most cases, DFA rules are general and do not consider the nature of the design, the process, and the global/local interactions that come with any design change. (The underlying philosophy of DFA, as D. Whitney underlined, is coding and classification. In other words, the rules are intended to be context free. However, this philosophy conflicts with how information is presumed to be formatted for use within an advisor's knowledge base.) As a result, some rules conflict in reasoning with others, or with the requirements and/or the functionality of the pre-determined design specifications. Therefore, it is important to look at the following questions, as a means of refashioning these rules:

- How does one determine which rules have a higher precedence over others?
- Will this hierarchical structure be the same for different products?
- Can this hierarchical structure be determined by evaluating how often each rule is used from product to product? Is there some other method that can be used?

Most textbooks and manuals, regarding the subject of DFA, do not classify rules in any way other than to fit them into a manual or an automatic assembly situation. G. Boothroyd mentions this limitation in his book, Assembly Automation and Product Design, confirming what has already been suggested in the thesis so far:

Although functioning well as general rules to follow when design for assembly is carried out, guidelines are insufficient in themselves for a number of reasons. First, guidelines provide no means by which to evaluate a design quantitatively for its ease of assembly. Second, there is no relative ranking of all the guidelines that can be used by the designer to indicate which guidelines result in the greatest improvements in handling and assembly; there is no way to estimate the improvement resulting from the elimination of a part of from the redesign of a part for handling, etc. It is, then, impossible for the designer to know which guidelines to emphasize during the design of a product.

Unfortunately, Boothroyd does provide any clear-cut approach for ranking these guidelines to fit specific assembly applications; he merely points this idea out to the reader.

Some rules have a major impact on the design while others make slight improvements to the design. DFA manuals do not show how some rules require more product knowledge in order to be implemented successfully. For instance, when trying to design an HVAC unit that allows the doors to be easy aligned and assembled to the main assembly, using a top-down approach, one must understand that this requires changing the configurations of all mating surfaces. Figure 4.2 below shows the two illustrations, depicting this design change: Figure 4.2-A shows the existing design of the Ford CDW-27 HVAC unit, while Figure 4.2-B shows a possible redesign of the unit, for the purpose of making the door-to-casing assembly much easier. (Refer to the HVAC Case Study in J. Nevins and D. Whitney's book, Concurrent Design of Products and Processes: A Strategy for the Next Generation in Manufacturing.)

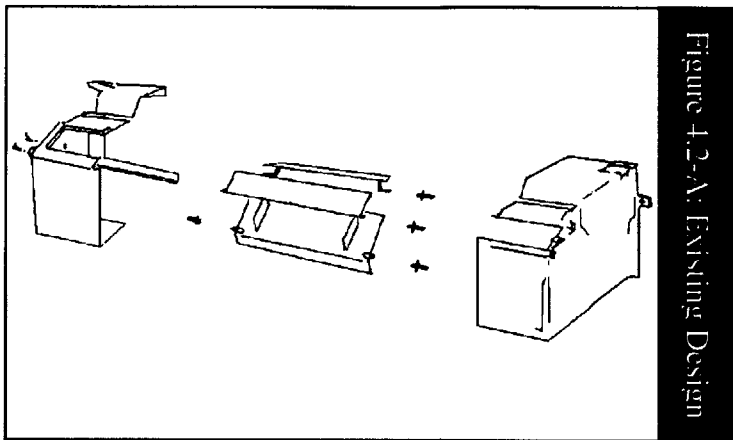


Figure 4.2-A: Existing Design

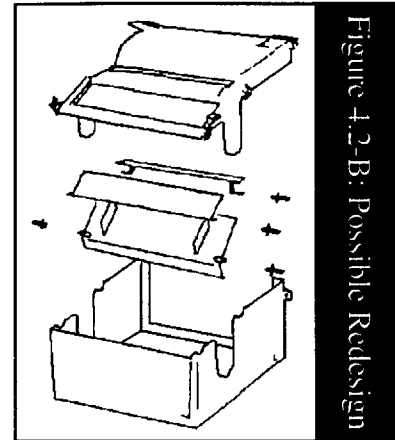


Figure 4.2-B: Possible Redesign

**Figure 4.2:** CDW-27 HVAC Designs

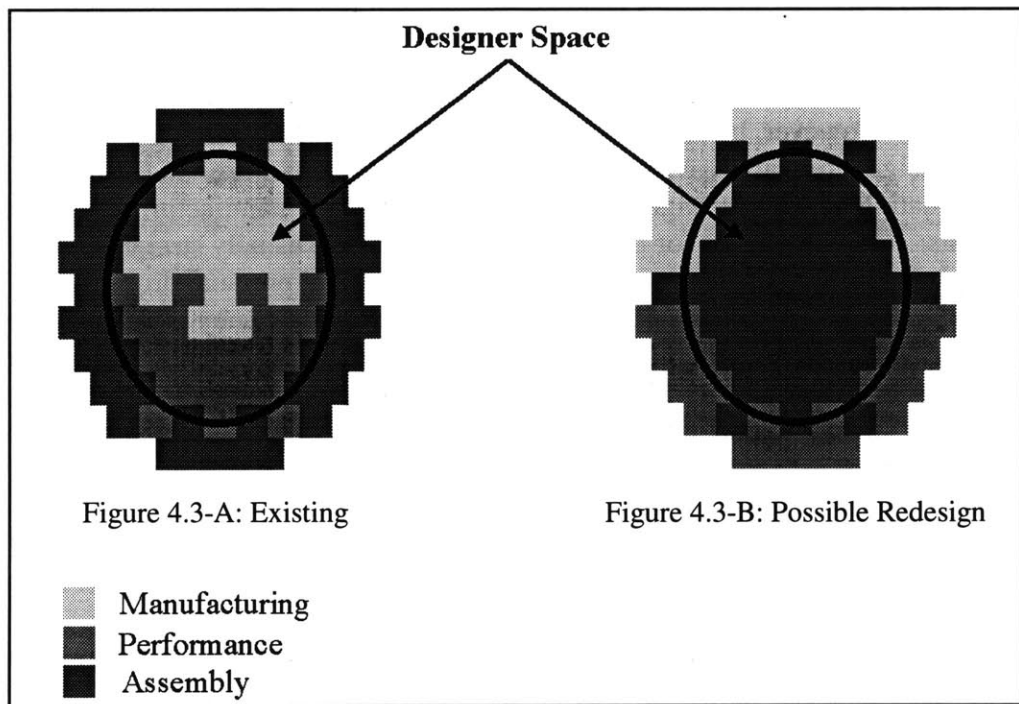
From the point of view of assembly, the latter design is definitely an improvement from the former one. However, by changing the design, even in this manner, the issues, which govern the design, are rearranged in a different hierarchical structure. In the original design, the mating plane is arranged in a way to fulfill certain manufacturing and airflow requirements. (Mating planes for products of this nature are usually straight so that the two casing halves can be mated properly without any residual pressure drops during operation.) Although the revised design is almost primarily driven by good assembly practices (i.e., the doors and top casing half are all inserted in a top down manner), it makes the design susceptible to other problems—problems related to manufacturing and performance. The undulating configuration for the new mating design complicates the tool design and increases the risk of air and water leaks.

A list of reasons why such a design suggestion was not implemented in the CDW-27 system are shown below:

- Cost for redesigning the mold outweighs the savings generated from an easier assembly technique (COST)
- The mating seam is not straight, thus enhancing the risk of water and air leaks (AIR PERFORMANCE).
- For injection-molding, the mating seams should be straight and the casing halves should be equal in size (MANUFACTURING)
- An undulating seam is usually not an option whenever vibration welding is used (EXISTING ASSEMBLY PROCESS)
- The limited capability of the CAD solid modeler (COMPUTER RESOURCES)

Like physical packaging space, all design considerations have to be “shaped” in a way to “fit” the most important issues into a limited amount of design space. Design space is a non-physical environment, which is associated with physical aspects of a design, such as

parts or features of parts. Designers mention design space when determining the amount of flexibility they have designing products to their liking. They use their expertise and creativity to solve problems and to reach demanding standards. The design space is bounded by certain design parameters (e.g., performance criteria, carryover selection, material selection, and selected manufacturing processes, to name a few). Designers must stay within these boundaries and remain focused on the key design drivers, which will ultimately shape the design and allow it to meet the given design requirements. However, not all design drivers “agree” with one another; many of them do conflict at some level. As a result, the most important issues, which are highly emphasized for meeting a project’s goals, are considered first for design space; lower-level issues remain outside this region, exposing them to potential design or fabrication risks. Figure 4.3 shows how design issues are considered and arranged in design space for both the existing and revised product designs shown in Figure 4.2.



**Figure 4.3:** Design Drivers with Each HVAC Design Seen in Figure 4.2

As Figure 4.3 attempts to illustrate, when certain factors are considered over other factors, for a limited design space, the low-priority issues usually get “pushed out” of this design region. In the case of the CDW-27 unit, issues regarding the performance of the entire unit and the manufacture-ability of the housing now move outside the designer’s scope of preference. By doing this, they become exposed to certain quality and performance risks.

Design space may be bounded by the physical limitations of the unit, the manufacturing/assembly equipment used to make the unit, and the conflicting issues that arise when trying to consider different aspects of the design. As a result, designers must rank the various drivers that are considered in a design from the highest to the lowest priority, and severe any issues from the design that do not “fit” within their appropriate design region.

#### 4.6 Classifying DFA Knowledge—General Framework

For the most part, existing DFA rules found in most academic manuals can be placed in one of the following five categories, in order from the lowest to the highest need for knowledge about a product<sup>33</sup>:

- Part orientation
- Part Handling
- Part Insertion
- Assembly Sequence
- Product Architecture

In terms of how it is used here, product knowledge encompasses not only knowledge about the product, but the interacting processes used to help develop the product, from the manufacturing/assembly process to the financial matters related to it. As the author has mentioned already, current DFA focuses more on part-to-part assembly rather than on product assembly—the interactions that exist between all of the parts that make up a product. The author attempts to illustrate this point by taking a number of different design rules and determining where they fit within these five different categories.

The grid shown below in Figure 4.4 contains a list of various DFA rules, which can be found in most books, manuals, and articles about this subject. (All of the DFA principles contained within this grid came from Redford and Chal’s book, Design for Assembly: Principles and Practices. Other books were also considered, but the author found no new material that could be added to the grid’s list.)

Each rule is assessed by how it fits each of the following areas: part handling and traveling, part orientation, part alignment and insertion, assembly sequence and product architecture. The author independently weighed each rule on three different levels—a strong, moderate or weak level of association to each one of the five categories. (These strength judgements were made exclusively by the author; no references were used to make these decisions—just the author’s experience and understanding about DFA and interpretation of each rule.) The key can be found at the bottom of the grid.

---

<sup>33</sup> The author developed this five-point framework based on his review of all general DFA rules. In G. Boothroyd’s book, Assembly Automation and Product Design, and in similar books, DFA rules are segmented into two categories: handling and insertion. D. Whitney noted that he uses this five-point system when teaching his assembly course, 2.875.

RULE:	PRODUCT KNOWLEDGE				
	Part Handling and Traveling	Part Orientation	Part Alignment and Insertion	Assembly Sequence	Product Architecture
Low friction between parts and track is undesirable for purpose of transportation	STRONG	WEAK	WEAK	WEAK	WEAK
Thin parts do not push effectively	STRONG	WEAK	WEAK	WEAK	WEAK
Abrasive parts wear away track material and may change critical dimensions on parts	STRONG	WEAK	WEAK	WEAK	WEAK
Sticky parts require reduced conveying velocities so that parts do not stick together	STRONG	WEAK	WEAK	WEAK	WEAK
Larger parts are conveyed at a slower rate	STRONG	WEAK	WEAK	WEAK	WEAK
Minimize the number of very stable attitudes that the part can take on the track	STRONG	WEAK	WEAK	WEAK	WEAK
Design parts so that they do not interact positively (i.e., tangle) with one another	STRONG	WEAK	WEAK	WEAK	WEAK
Loss of output depends on size of features—may not help to increase velocity of conveyor system	STRONG	WEAK	WEAK	WEAK	WEAK
Accept parts in an orientation that can give the highest output	STRONG	MODERATE	WEAK	WEAK	WEAK
Design parts to be handled by “universal” gripping machines	STRONG	WEAK	WEAK	WEAK	WEAK
Ensure that there is good access on the gripper	STRONG	WEAK	WEAK	WEAK	WEAK
Less orientation usually implies more output	STRONG	MODERATE	WEAK	WEAK	WEAK
Maximize part symmetry if possible or make parts obviously asymmetrical	STRONG	MODERATE	WEAK	WEAK	WEAK
Try to avoid dealing with small parts or small features on large parts	MODERATE	WEAK	STRONG	WEAK	WEAK
Choose the largest open tolerances that can be used	STRONG	WEAK	STRONG	WEAK	WEAK
Assist alignment by general rule of chamfers, tapers, etc.	STRONG	WEAK	STRONG	WEAK	WEAK
Eliminate crossed assembly surfaces	STRONG	WEAK	STRONG	WEAK	WEAK
Avoid precise assembly surfaces (i.e., close fits)	STRONG	WEAK	STRONG	WEAK	WEAK
Contact points on cylindrical surfaces should be avoided	STRONG	WEAK	STRONG	WEAK	WEAK
Design parts so that they can be released as soon as possible once insertion has started	STRONG	WEAK	STRONG	WEAK	WEAK
Unattainable insertion occurs when the grip site cannot be maintained up to the point of insertion	STRONG	WEAK	STRONG	WEAK	WEAK
If a further part needs to be located relative to the current part, ensure that the final location of the current part is accurate (Rule of Constraint)	STRONG	WEAK	STRONG	MODERATE	WEAK
Ensure that the probability of successful insertion is high before releasing the part	STRONG	WEAK	STRONG	MODERATE	WEAK
Try to avoid simultaneous insertion operations	STRONG	WEAK	STRONG	MODERATE	WEAK
Insert from above	STRONG	WEAK	STRONG	MODERATE	WEAK
Having several choices at an early part of assembly is more important than at a later stage <sup>34</sup> (Immediate Commitment)	STRONG	WEAK	STRONG	MODERATE	WEAK
Good product design can help ensure that the most expensive items are assembled first <sup>35</sup>	STRONG	WEAK	STRONG	MODERATE	WEAK
It is more sensible to customize product at as late a stage as possible (Delayed commitment)	STRONG	WEAK	STRONG	MODERATE	WEAK
Standardize parts and interfaces	STRONG	WEAK	STRONG	MODERATE	WEAK
Reduce part count and part type	STRONG	WEAK	STRONG	MODERATE	WEAK
SCORE (showing where DFA rules apply)	12.0	4.0	14.0	8.5	5.0

 STRONG association with respected category 1.0 point  
 MODERATE association with respected category 0.5 point  
 WEAK or NO association with respected category 0.0 point

**Figure 4.4:** Assessing Design for Assembly Rules

<sup>34</sup> This falls in line with the natural process of assembling a product. The industrial engineer (the person who usually sets up the assembly process at the factory floor) has the ability to change the order of different assembly operations at the beginning of the process because he is not confined to the physical limitations of the assembly as it is being developed. As more and more parts are attached to the assembly, the industrial engineer has fewer options to consider due to part obstruction or the weight and size of the “growing” assembly.

<sup>35</sup> This assumes that the expensive parts are protected by the parts around them. For example, the evaporator core of an HVAC unit is protected by the plastic casing, which is assembled around it. However, one has to consider that as these delicate parts move through the assembly process, they are subjected to several flips and movements. The neighboring parts must do a very good job of containing these delicate parts and restraining them from any possible movement.)

This figure shows how assembly knowledge can be grouped together into separate categories and subcategories, ranked (according to the degree of product knowledge that is required for analysis), and eventually placed into a hierarchical framework. The example also shows, as was mentioned earlier, where assembly knowledge stands now and what it focuses on.

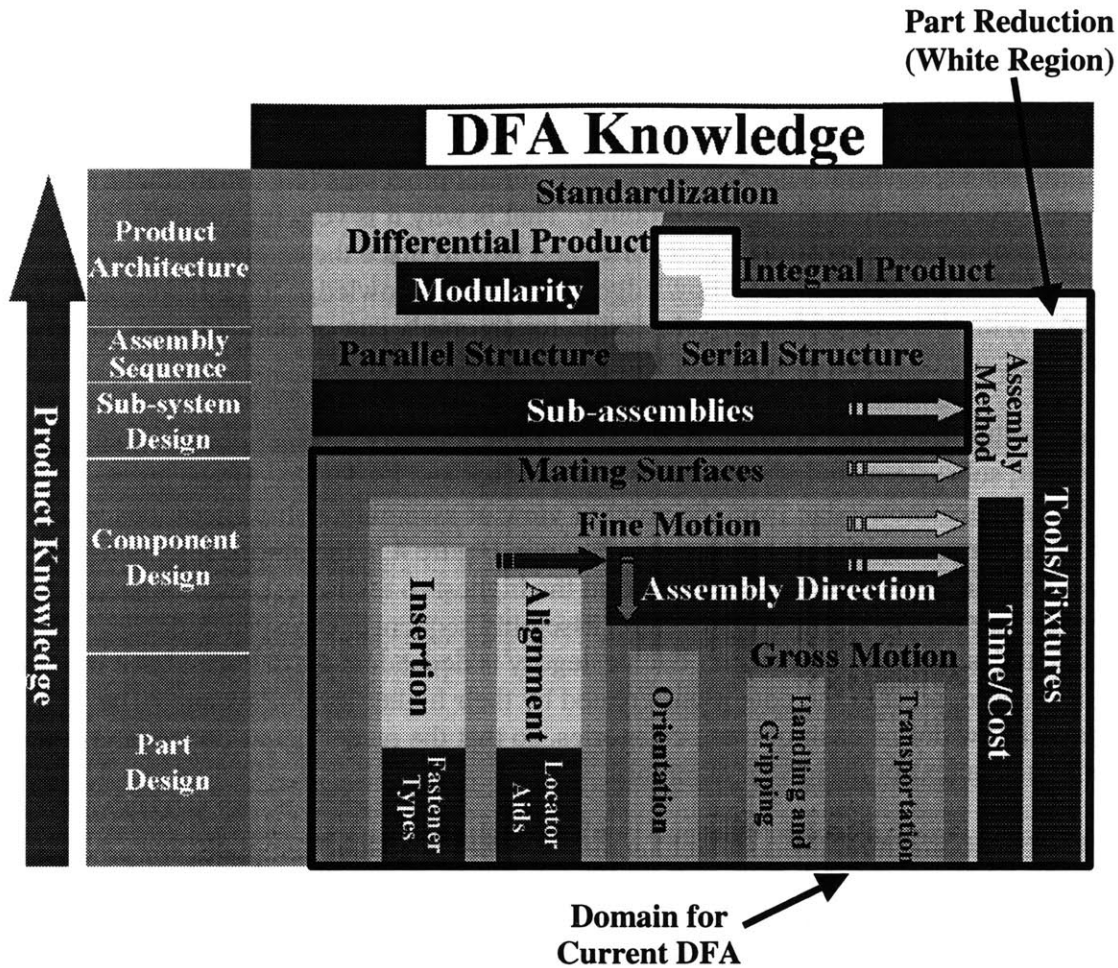
The score at the bottom of each column shows where there is the greatest and weakest emphasis in DFA. Of course, the reader should not be completely persuaded by these results, due to the author's sole judgement about each rule and where they stand in the context of DFA and the limited number of rules collected in a few books. For the moment, ignoring the fact that these issues can certainly affect the outcome of the results, the fact is that there is still a lack of DFA knowledge at the level above part and component design—primarily because such information is based on the products that the rules are suppose to analyze.

The reader may notice that that the column noted as “Part Orientation” has the lowest score. The author believes that this is due to the fact that there are few rules that need to be mentioned when improving a design for better part orientation. In other words, just a few rules can completely exhaust this topic.

The reader should also notice that the rules governing “Part Location and Insertion” dictate the rules driving “Assembly Sequence.” For some of the cells that are colored green in the “Part Location and Insertion” column, a yellow (or green) cell follows, in the “Assembly Sequence” column.

There has not been much emphasis in design for assembly when it concerns a product's architecture, as the reader can see from Figure 4.4. The reason, as mentioned earlier in this chapter, is that such understanding requires knowing a great deal about the product, in question. DFA rules and guidelines cannot be created to encompass different architectures and situations. The only way that they can is by developing a consistent set of stipulations that can be measured and used across different product architectures. However, this requires the evaluation of different assembly conditions as they pertain to different product types. More specifically, it requires that each of these rules be weighed according to different assembly conditions and product types (some rules may not have as much importance in some situations than they do in others) and that more rules be developed with a particular emphasis on product architecture. This, of course, requires first developing a set of characteristics that differentiate each architecture type within a given product portfolio, and then understanding how assembly is different in each case. The author does not have any particular suggestions on how this can be done for all product portfolios, but provides some insight in Chapter VI.

To begin the process of associating assembly rules to various applications, different products have to be analyzed based on different types of architectures, mating planes, assembly sequences, assembly directions, and all other factors that encapsulate assembly. All of these issues should be grouped together and thought about systematically, so that

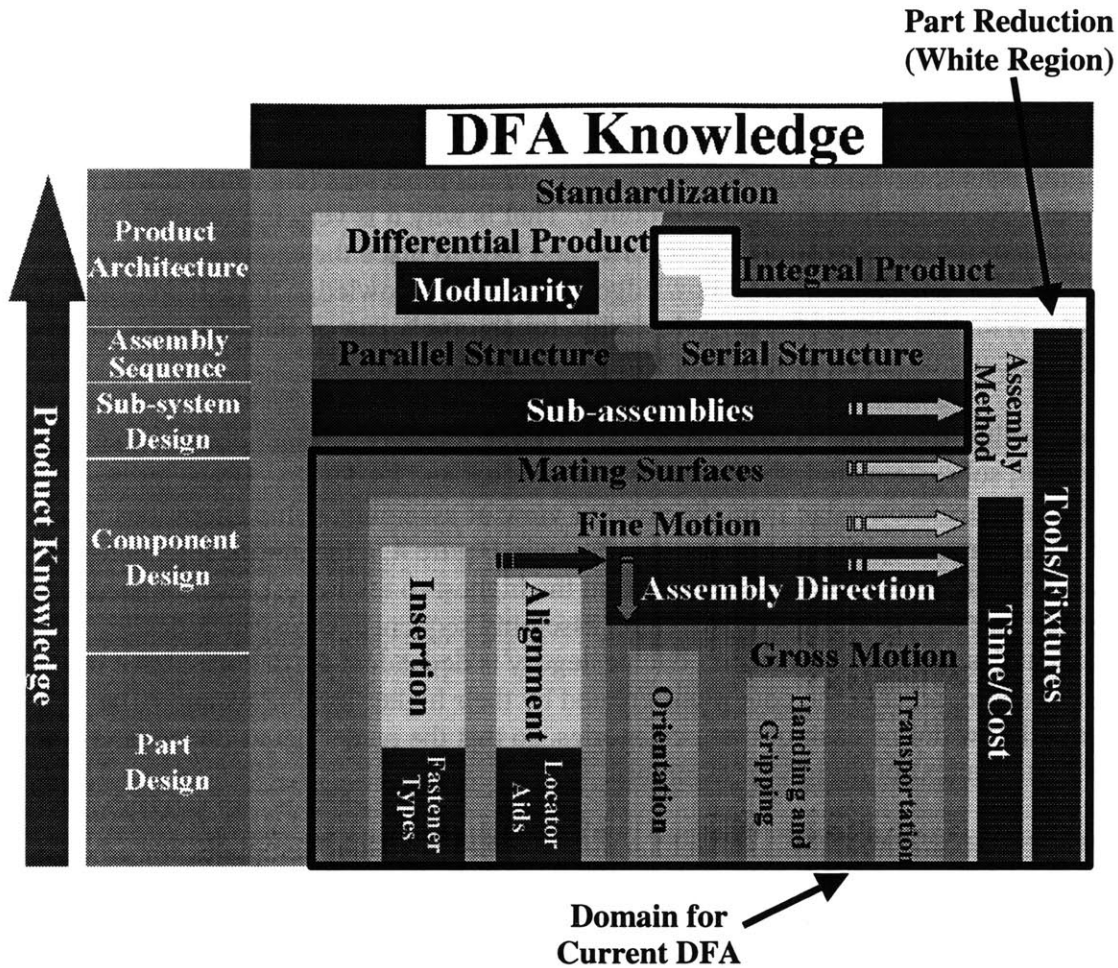


**Figure 4.5: DFA Knowledge at a Deeper Level of Understanding**

Figure 4.5 includes different types of DFA knowledge, at a very high level of abstraction. DFA knowledge, as it is suggested here, is based on the following information: general design rules (such as those listed in Figure 4.4), ways in which different products are organized for assembly, and methods for analyzing the assemble-ability of a product (such as evaluating the time and cost for assembling a product). The figure tries to emphasize how different areas and levels of product design can have an effect on or can be affected by different assembly actions. In other words, it tries to show how and where a physical product is tied in with assembly. The author will use this picture to cover three topics of discussion: (1) what was the reasoning for this structure; (2) what areas of this structure are covered by existing DFA methods and theory; and (3) in what ways can “untouched” areas of DFA be exposed and linked to the real design environment. Because there is so much information contained in this figure, the author will speak about these topics individually within the next few sub-sections.

#### 4.7.1 The Structure of Figure 4.5 and Its Underlying Principles





**Figure 4.5:** DFA Knowledge at a Deeper Level of Understanding

Figure 4.5 includes different types of DFA knowledge, at a very high level of abstraction. DFA knowledge, as it is suggested here, is based on the following information: general design rules (such as those listed in Figure 4.4), ways in which different products are organized for assembly, and methods for analyzing the assemble-ability of a product (such as evaluating the time and cost for assembling a product). The figure tries to emphasize how different areas and levels of product design can have an effect on or can be affected by different assembly actions. In other words, it tries to show how and where a physical product is tied in with assembly. The author will use this picture to cover three topics of discussion: (1) what was the reasoning for this structure; (2) what areas of this structure are covered by existing DFA methods and theory; and (3) in what ways can “untouched” areas of DFA be exposed and linked to the real design environment. Because there is so much information contained in this figure, the author will speak about these topics individually within the next few sub-sections.

#### 4.7.1 The Structure of Figure 4.5 and Its Underlying Principles

DFA knowledge is a very broad topic that goes far beyond the borders of the physical product, which it helps to create. DFA can be made to consider not only the parts of a product, but the different levels of interaction between parts, components, sub-systems and systems, as well as all of the “outside” factors (e.g., performance requirements, packaging issues, material selection, etc.) and essential processes (e.g., manufacturing and assembly) that help make up the product. That is why it is very important to place these types of issues into different categories and then to place the categories into a framework that allows assembly knowledge and product knowledge to be linked with one another. With this type of framework, design for assembly can be defined more clearly and strategically utilized for optimum effectiveness.

The author linked product knowledge to the different levels of product design—the part, component, subsystem, and system levels of a product and its design. However, to consider product knowledge from the point of view of assembly, some alterations were made to this organizational structure. As shown in Figure 4.5, the author segmented product knowledge into five categories, from the highest to the lowest level of need in design for assembly—product architecture, assembly sequence, sub-system design, component design, and part design. Each category is dependent on the one above it; they each, in a sense, react to the conditions created by their higher level associate. The author will now define each of these categories, so that the reader better understands their place in the context of design for assembly.

- **Product Architecture:** Product architectures can be considered templates for product design. Using a product architecture helps designers develop or refashion products, while keeping in mind issues that are important to each product’s function. Product architectures are used to provide a good starting point for incremental product development, preventing unnecessary reengineering, and avoiding excessive costs.
- **Assembly Sequence:** Assembly sequence structures consider the order of assembly for each part that incrementally makes up a product by evaluating different conditions and constraints at both the process and product levels. Some of these issues include the number of orientations for parts and assemblies, the line of action for assembly, part interferences (particularly, as a function of assembly order), assembly equipment (such as fixtures, power screwdrivers, hoppers, etc.), material/part flow, and the layout of the assembly floor. In addition, the assembly sequence structure is strongly dependent on product architecture; the type of architecture used for a product determines the limited number of ways in which a product can be broken up into separate assemblies.
- **Sub-system Design:** Sub-system design considers each sub-division of the product (i.e. sub-assemblies); the mating surfaces to other sub-assemblies, components and parts; the tools and fixtures necessary for assembly; and the type of assembly method that will be used. (Sub-system design considers more than just the physical sub-assemblies; it considers the interactions between various parts of the entire assembly

and the process for assembly.) At this level, there is a dichotomy from system-level design to part-centered design.

Note: Sub-assemblies have a product architecture and an assembly sequence of their own. As a result, product knowledge required from the level of sub-system design would include the two categories previously mentioned. (Information is allowed to flow up and down at most levels of product knowledge for the purpose of retrieving global-level information and passing it down to local-levels, as is the case with sub-system design. This will be explained more when the author discusses the reason for the arrows.)

- **Component Design:** A component is defined as two or more parts that come together to make a stable sub-assembly. At the component level, part-to-part interactions are considered as well as the interactions between components and sub-systems. Issues like mating-surface configurations and tolerances really begin to get the designer's notice at this level of assembly design.
- **Part Design:** At the level of part design, all of the individual elements are considered. It is at this level that assembly features, such as chamfers and snap fits, are emphasized.

Collectively, these categories help bring some order to DFA. (Most if not all of the issues associated with DFA are shown to the right in Figure 4.5.) Some areas of DFA are associated with only certain levels of product knowledge, while others are linked to two or more different levels. For example, designing locator aids for assembly is normally considered at the part-design level, but the use of standardization in assembly can be used at any level of a product, from the fasteners to the architecture. (The author also suggests that the reader refer to Figure 4.4 to see how different DFA rules apply to product knowledge—some rules emphasize a piece of product knowledge while others span the entire scope.)

In some cases, certain areas of DFA knowledge are segmented even further. Two good examples of this are gross and fine motion. Gross motion can be segmented into orientation, handling and gripping, and transportation; likewise, fine motion can be divided into insertion and alignment.

The arrows dispersed throughout the figure attempt to illustrate a high connectivity between certain facets of DFA and others. Each arrow places an emphasis on how information flows through the different fields and what issues need to be considered “upstream” from these arrows. Where arrows are not drawn, this implies that the flow of information is not concentrated in any one direction; in other words, assembly information can flow up or down with respect to product knowledge.

In some areas of design for assembly, there are no definitive borders that clearly separate different aspects of its knowledge. For instance, it is not clearly understood where

differential architecture segments itself from integral architecture. In such cases, boundaries are replaced with some type of range or scope for measurement. (Figure 4.5 shows this with blending borders. In particular, look at the blending borders between Differential Product/Integral Product and Parallel Structure/Serial Structure.)

The author will now speak about some of the underlying principles that are hidden away in this figure. All the important principles will be discussed from the highest level of product knowledge down to the lowest level.

#### Product Architecture: Integral versus Differential Product Design

Product architecture can be described, in general terms, by the degree of integrality or differentiation used in the product scheme. Most books and journal articles place modular and integral design against one another, mentioning that they are absolute opposites. This is a misconception, however. It is not always true that products are segmented into individual parts and components for the sole purpose of developing a modular product portfolio. A product may have to be segmented into different ways for simple reasons, some which include design for assemble-ability, design for manufacture-ability, part carryover, functional requirements, and designer reasoning<sup>36</sup>. Segmentation in this manner is called differential product design. Modular product design is a special case of this. (To pass this idea along to Figure 4.5, the author had “Modularity” completely encapsulated by “Differential Product”.)

Because modular and integral architectures both have legitimate reasons for improving product designs, particularly in assembly, they will be talked about more in Section 4.8.

#### Assembly Sequence: Parallel versus Serial Structure

The next level of assembly design is to see how products can be designed for different assembly schemes. The two general types of assembly schemes, as mentioned by Redford and Chal, are serial and parallel operations. These cases and any mixture of the two can be illustrated in a precedence diagram. (A precedence diagram shows when parts need to be introduced into the assembly mixture with respect to other parts.)

In a parallel product structure, the parts of a product can be placed in any order without any concern for additional assembly space. An example of this would be the assembly of different electronic components to a PC board.

Any product that uses a serial assembly structure has one unique solution (i.e., one possible order for assembly). In most cases, it is due to the geometry of the individual parts in an assembly and what type of assembly space they provide for any in-coming parts. For example, any product that uses a “parts in a box”<sup>37</sup> design must have, at some

---

<sup>36</sup> Many designers have a natural tendency to design parts for individual functions, rather than to design parts that incorporate several functions. This is one of the reasons why part consolidation has become such an inspiring practice in product development companies.

<sup>37</sup> The phrase “Parts in a Box” was created by D. Whitney in order to help classify and understand different types of product designs with respect to assembly.

level, a serial-assembly structure. Generally speaking, the assembly process must follow this order: (1) parts are inserted into one or more sections of the “box”; (2) all sections of the “box” are assembled together to create a complete “box”; and (3) parts are assembled to the outer section of the “box”. In steps 1 and 3, a combination of parallel and serial assembly operations can apply; step 2 separates steps 1 and 3 from each other, making this part of the assembly process serial.

Why is this distinction important? It allows the designer to understand the limitations that are placed on the design from the point of serial and parallel assembly sequences. A completely parallel assembly structure gives an industrial engineer—one who determines the assembly sequence, the placement of equipment, and the flow of parts and materials on the factory floor—the ability to really make a contribution. He has the flexibility to rearrange the operations for better assembly output. Serial product structures limit the amount of reorganization that the industrial engineer can make to the assembly. If a product has a purely designed serial structure (e.g., unstable assembly operations that require two hands to handle and are immediately followed by a reorientation), the industrial engineer has to work around that by incorporating additional assembly procedures or equipment. Designers should be aware of the implicit levels of parallel and serial assembly in their designs, and whenever possible, try to make the design conducive to parallel assembly.

#### Assembly Method: Manually-, Automatically-, or Robotically-Driven Assembly Operations

Any assembly method that is used is determined by the “available” assembly sequences and several other contributing factors, such as the number of orientations required before, during and after each assembly step; the type of tools and fixtures used; and the level of fine motion required for inserting parts. (Figure 4.5 shows these different attributes with arrows flowing into “Assembly Method”.) In general, deciding on what the “appropriate” assembly method is for a given product requires information about the production volume<sup>38</sup>, the level of difficulty for assembly, and the type of assembly equipment that is committed to the product. The second factor is determined by how sophisticated the individual assembly operations and motions (gross and fine motions) are. (Complex assembly operations can require the use of fixtures. For example, refer to Figure 4.2-A. In order to assemble the three doors to each casing half, a fixture of some sort must be used. To do this operation without one, as the author had experienced, requires some fine handling in order to align each of the five parts together.)

#### Time and Cost Evaluation

The type of assembly method used (whether it is manually-, automatically-, or robotically driven or any combination of the three) can determine what type of scoring system should be used to assess the time and cost for actual assembly. The assembly method, in a sense, provides the framework for analysis. (Based on this statement, one can say that current

---

<sup>38</sup> One vital piece of information that is not shown in Figure 4.5 is production volume. This is the number of products produced yearly. It influences the type of assembly method used. Small volumes push for manual assembly, while high volumes push for high-speed assembly.

DFA evaluation methods are mostly process driven rather than product driven, as Figure 4.4 tries to illustrate.)

#### Fine Motion versus Gross Motion

The two types of assembly actions or motions are shown in Figure 4.5—fine motion and gross motion—and they are considered individually at different levels of product knowledge. Fine motion—which encompasses all motions for aligning and inserting parts into an assembly—requires a relatively high level knowledge about the given product in order to be used effectively. Gross motion, on the other hand, which requires all operations leading to fine motion, can be considered at the part-design level. Gross motion does not need to consider actual assembly operations or part-to-part interactions; therefore, it can be considered without any contribution from other parts of the product. Although gross motion, in some sense, is product independent, it is mostly process dependent. Factors like type of assembly equipment used for transporting, containing and orienting parts contribute significantly to the evaluation of gross motion.

#### 4.7.2 What Does DFA Emphasize Based on Figure 4.5?

As Figure 4.4 helped to show, DFA rules normally emphasize part and component design, with very open-ended suggestions for remaining levels of assembly design. The evaluation tools that exist for analyzing the assemble-ability of a product also fall into the part and component regime, with very little emphasis beyond that. For the most part, DFA emphasizes issues revolving around part insertion, part alignment, and part orientation (three of the five levels shown in Figure 4.4). It does make some sparse jumps to product architecture, but this is done indirectly when suggesting the consolidation of two or more parts. (Even this type of suggestion is somewhat unfocused because part consolidation by DFA standards mostly involves replacing threaded fasteners with plastic snap tabs—not looking at the product beyond the features on the individual parts. Beyond that requires product knowledge and even user-based knowledge.)

As was mentioned earlier, DFA evaluations and rules are process rather than product dependent. The author mentions some reasons why this trend was most likely followed during the development of DFA:

- Manufacturing and assembly processes can be classified into a selected number of categories. Product designs and architectures, however, can vary enormously and, as result, are difficult to classify collectively.
- There are a finite number of steps in any assembly operation, many of which are explicit and repeatable. The steps that go into designing a product are not always clear and methodical as in actual assembly, making the whole design process difficult to analyze. (This very topic was mentioned in Chapter III; there the author discussed why KBE/Visteon leans so heavily on the expertise of the cross-functional team.)

- Design for assembly is intended to bring information regarding assembly techniques closer to design. It does not consider, however, how to best fit this information into the product development process.
- From an industry perspective, design for assembly should provide ways for reducing assembly time and cost. Time and cost metrics, however, are process dependent and do not consider the nature of the product or the process for designing the product. Unfortunately, at this time, very few metrics exist which are dependent on the product, basically because it is very difficult to quantify product knowledge.

DFA will not be able to do an actual product analysis without “knowing” more about the given product. Right now, DFA provides quick “solutions” by focusing on the part and component levels (as they pertain to a given assembly process) and making recommendations from there.

#### 4.8 Integral and Modular Architectures and Their Roles in DFA:

Because product architecture is a very important issue in product design, it should be equally important in design for assembly. Unfortunately, DFA has not fully arrived at this level yet. Current DFA methods implicitly suggest that products be designed with a high level of integrality. The author says that this is done “implicitly” because DFA makes its recommendations at the part level. However, modular design is also taking notice in realm of design for assembly. Unlike integral design, which tries to eliminate assembly operations entirely, modular design attempts to use assembly to its advantage to expand existing product portfolios.

The author has devoted this section to integral and modular architectures and their roles in DFA for the following reasons:

- Both provide an equal assortment of advantages and disadvantages in design for assembly (but in what areas of design and development?)
- Both architecture types look at assembly differently, and both require different levels of knowledge for effective utilization. (See Chapter II for details.)
- More emphasis needs to be placed on product architecture—the root of product design—in order to understand how DFA can be used most effectively.
- Current DFA practices do not consider how their suggestions implicitly lead to altercations in product design.

##### 4.8.1 Part Consolidation and Its Role in Integral Product Design:

For many, design for assembly means simply part consolidation. Logically speaking, this makes sense: fewer parts mean fewer assembly operations and a lower assembly cost per product. Other reasons for consolidating parts include<sup>39</sup>:

---

<sup>39</sup> Redford and Chal, Design for Assembly: Principles and Practices (1994)

- Reduction in assembly time and cost
- Potential for less failures
- Potential for less-in-process inspections
- Higher product reliability
- Lower manufacturing costs (on a per-product basis) <sup>40</sup>
- Faster implementations
- Practical requirements of being able to assemble the product

The process for determining if two or more parts can be consolidated into one, or if the parts themselves, can be removed from the assembly altogether is rather simple to describe. To do this, one must know about the functional importance of each part in the assembly. If a part in question has enough functional importance to be in an assembly, then one must ask, does it have enough reason to stand by itself, as one lone part? The answer may be due to various reasons. For instance, the part may need to move relative to another part, or it may require a different type of material for whatever reason(s). G. Boothroyd, in an article titled “Design for Assembly—The Key to Design for Manufacture”, mentions a three-part questionnaire, to determine if part consolidation or part removal can be successfully accomplished.

1. During the operation of the product, does this part move bodily with respect to all parts already assembled?
2. For fundamental reasons, does the part have to be of a different material from all other parts already assembled?
3. Does the part have to be separate from all other parts already assembled because otherwise assembly or disassembly of other separate parts could not be carried out?

Although DFA makes the designer consider the functional requirements of each part, it does not consider how changes in the design, even at the part level, can effect the functional parameters of the entire system. D. Whitney points out, using an example taken from Redford and Chal’s book, Design for Assembly: Principles and Practices, how part reduction and consolidation can effect the performance of a pneumatic pump design. In this design, a plunger moves up and down at a rate that is dependent on the air flowing in and out of the pump, the spring that the plunger glances off of, and the weight of the plunger. These properties have to be in balance with one another in order for the plunger to reciprocate up and down at the required velocity and acceleration. D. Whitney mentions that by integrating parts to the plunger for the purpose of reducing assembly operations affect its rate of displacement—due to the added weight of the plunger.

---

<sup>40</sup> One of the controversies that has emerged with the use of DFA (part consolidation) is that while it reduces the cost for assembly it increases the cost for manufacturing. (Some experts, including Boothroyd, say that part consolidation, if applied wisely, can decrease the overall cost for fabrication.) However, by consolidating parts, the molds to make those parts, increases in complexity and in cost. All of these factors, of course, have to be considered for each separate application.



Although the result from part reduction and consolidation can be immediately beneficial (making it a strong carrier for other potential projects), one must consider the implications that can be associated with such actions during the lifetime of this product and the development of later ones.<sup>41</sup> Many of these so-called side effects can navigate their way through later stages of product development, manufacturing and quality control. For instance, part sharing and part reuse may also need to be considered even before asking the question, can this part be consolidated with that part?

Part consolidation, for many of the reasons described earlier, is a good technique to use, but when used without caution or concern can trap any firm into the mindset of “one product at a time”. (M. Meyer and A. Lehnerd mention this in their book “The Power of Product Platforms”.) Part consolidation enforces the idea of making unique parts to accommodate one product. Rarely do people, who use this technique, consider other issues, which can take into account other products or a product family. As a result, whenever new products are developed to conform to various needs and requirements, the design process essentially begins from scratch, thus increasing the time and cost for development. How can parts be reused or made to serve particular standards when part consolidation pushes them to be unique? (As mentioned by D. Whitney, a possible rationale for this is that the functions involved are common, so a common consolidated part can be used. However, one has to consider all levels of a part’s function and its location in space relative to other parts, which it is being compared to.) That is the question that needs to be answered every time this technique is used.

Part consolidation may be considered one of the balancing forces in any design process, keeping the designer in check with his own natural tendencies to design more parts than necessary. In addition, designers generally like to do the following:

- Redesign parts rather than look for and use existing parts
- Design parts without fully understanding the relationship between cost and feature design
- Design products that the designer likes and possibly not what the customer likes

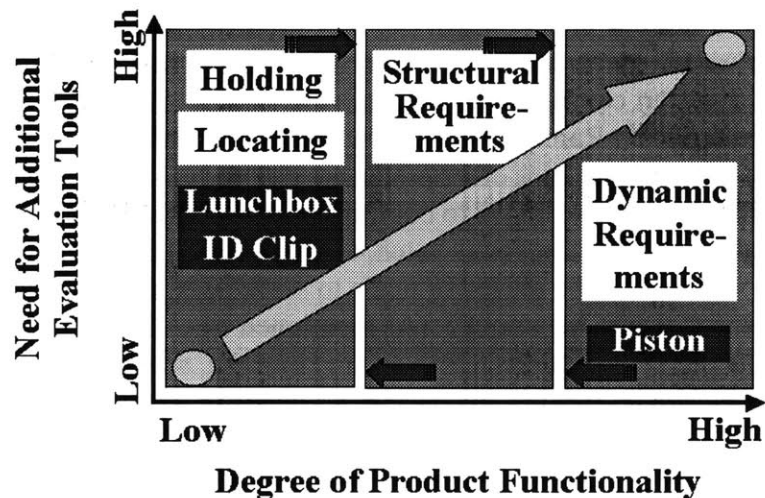
DFA is really no more than a technique for checking the efficiency of any design, with respect to cost. The rules, themselves, are merely suggestions, which can be used by the designer if he feels that the change will not affect existing parameters that are important in the design.

#### 4.8.2 How effective is DFA/Part Consolidation by itself?

---

<sup>41</sup> It can also be said that DFA and part consolidation have such an immediate impact in any manufacturing organization because their suggestions deeply affect the actions made on the factory floor, and not at the design level. N. Reppenning and J. Sterman in an article titled “Getting Quality the Old-Fashioned Way: Self-Confirming Attributions in the Dynamics of Process Improvement” make the statement that it is more difficult to see immediate improvement in product-development because results, on average, take twice as long as they do in manufacturing.

In many cases, the act of consolidating parts for the purpose of improving the assembleability of a design is simply not enough. This act is usually followed by some type of functional analysis on the entire system: one must determine if all the necessary system-level functions for that design are still in check with all of the pre-established requirements. The level of analysis can depend on the severity of each risk under evaluation, and the nature of each product function. The following illustration (Figure 4.6) attempts to show how part consolidation is related to product functionality.



**Figure 4.6:** How Effective is DFA by Itself?

Figure 4.6 is broken down into three types of mechanical “products,” and they are listed below:

1. Products that merely hold two or more parts together; products in this classification are normally subjected to small loads. Examples include lunchboxes and ID clips.
2. Products that are required to withstand excessive loading, such as a suspension bridge.
3. Products that are required to move, such as a piston or pneumatic pump.

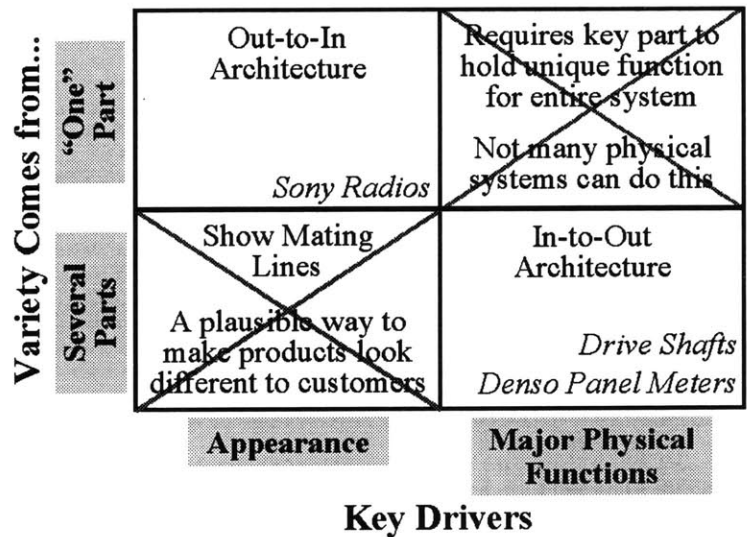
When considering “simple” products, like the lunchbox, where the functional requirements are small in number, part consolidation can be used without the need for further analysis. However, for those products with a relatively high number of functional demands (e.g., structural, thermal, and dynamic requirements), certain analysis tools may be required for checking the integrity of the product.

#### 4.8.3 Modular Design and Its Role in DFA:

Like DFA, modularization has been gaining a great deal of popularity in the realm of product development. Products using a modular design are dictated by a different set of circumstances than products with an integral design. Rather than having the various

functions confined to a selected number of parts, as in the case of any integral design, the functions are spread across an array of parts for many reasons. By setting certain standard parameters, such as particular part-to-part interfaces, various combinatorial mixes can be successfully accomplished, allowing different functional conditions to be met with each type.

Modular design, as the author has seen, can be segmented into two or more ways. The following graph (Figure 4.7) shows this segmentation. (For this case, the term “appearance” is identified as a separate parameter from other product functions.) Sony Walkman radios and Denso panel meters are used as examples to help demonstrate the different classifications of modular architecture. Sony develops product mixture by changing the outer casing (what is seen by the customer), while keeping the internal mechanisms relatively untouched. (The author calls this an out-to-in architecture, as shown in quadrant 2, meaning that variation comes from changing the outer surface of the product.) In this case, variation comes from one part; any new members to the product family would require making new cover designs. To put it simply: each new product is defined by one pivotal part—in this case, the cover design. For the Denso panel meters variation comes from the combination of several parts assembled together to match a particular performance requirement. (The author calls this an in-to-out architecture, because the changing functional requirements are satisfied by the internal parts, and more particularly by their interactions with one another.)



**Figure 4.7:** Type of Modular Designs

The remaining two cells, which are seen crossed out, show other types of potential modular designs, which are seldom used because of various mechanical limitations. Some of these limitations are described in the cells, themselves. (Note: The lower-left quadrant is intended to show how several parts, which define the outer surface of the assembly, can act as an “eye sore” to customers. Most customers like products that have

an integral shape to them; they do not want to see mating lines passing all along the product.)

#### **4.8.4 Advantages and Disadvantages with Each Type of Architecture:**

Although one can find several advantages for emphasizing one type of architecture over another, or visa versa, the fact is that the architecture that is chosen must “fit” the product’s intended conditions (e.g., packaging space, reusability, etc.) Many books and articles about product development describe the advantages and disadvantages for both modular and integral architectures. However, what they rarely show is how these advantages/disadvantages impact different areas of design and fabrication. For instance, where, in particular, does modularity and integrality have significant impacts on manufacturing, assembly, product development, and quality control, to name just a few? Under what conditions should one consider an integral architecture or a modular architecture? Not only should the advantages and disadvantages of each be identified, but also under what conditions do they have the biggest impact.

The following table attempts to take some of the already known pros and cons of each and classify them into four categories: product development, manufacturing, assembly and quality control.<sup>42</sup> (Classifying the advantages and disadvantages of modular and integral architectures into only four categories may be considered incomplete; but for the purposes of this example, it will suffice.)

---

<sup>42</sup> Ulrich, K. “The Role of Product Architecture in the Manufacturing Firm”

	MODULAR ARCHITECTURE		INTEGRAL ARCHITECTURE	
	Advantages	Disadvantages	Advantages	Disadvantages
<b>MANUFACTURING:</b>	<ul style="list-style-type: none"> <li>• Low tooling costs</li> <li>• Lots of tools</li> <li>• Short set-up times</li> </ul>	<ul style="list-style-type: none"> <li>• Lots of parts to keep track of</li> </ul>	<ul style="list-style-type: none"> <li>• Reduced material flow</li> <li>• Low inventory</li> </ul>	<ul style="list-style-type: none"> <li>• High tooling costs</li> </ul>
<b>ASSEMBLY:</b>	<ul style="list-style-type: none"> <li>• Use standard fixtures</li> </ul>	<ul style="list-style-type: none"> <li>• High part count</li> <li>• Requires assembly</li> </ul>	<ul style="list-style-type: none"> <li>• Low part count</li> <li>• Low assembly costs</li> </ul>	<ul style="list-style-type: none"> <li>• Need for unique fixtures</li> </ul>
<b>PRODUCT DEVELOPMENT:</b>	<ul style="list-style-type: none"> <li>• High product variety</li> <li>• Long product family life</li> <li>• Component sharing and component reuse</li> </ul>		<ul style="list-style-type: none"> <li>• Efficiency in space, weight, or energy consumption</li> </ul>	<ul style="list-style-type: none"> <li>• Limited product variety</li> </ul>
<b>QUALITY CONTROL:</b>		<ul style="list-style-type: none"> <li>• Visible character lines</li> </ul>	<ul style="list-style-type: none"> <li>• Improved product reliability</li> </ul>	<ul style="list-style-type: none"> <li>• Performance problems difficult to find and de-couple</li> </ul>

**Table 4.1:** Advantages/Disadvantages of Integral/Modular Architectures

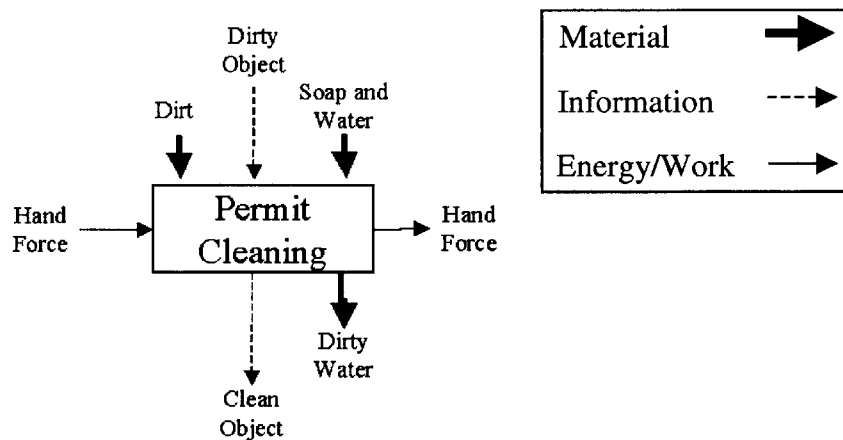
It is important to think about these issues when creating a new design, or if possible, modifying an existing design. Breaking down the advantages and disadvantages of both modular and integral architectures into categories like those shown in Table 4.1 can give product-development companies a better idea of which type of architecture is more suitable for their needs.

#### 4.8.5: The Use of Function Maps to Leverage Part Integration with Part Modularization and Standardization:

To better understand the tradeoffs between integral and modular designs, especially in different design applications, a function map is recommended. A function map is a type of tool that designers can use to really understand how function and design are associated with each other. Functions maps are very versatile: they can be used during concept design or redesign. They can help the designer consider the basic principles of design, allowing him to see beyond traditional standards. For example, when used properly, function maps can make a designer question the pre-established segmentation of most assemblies.<sup>43</sup>

<sup>43</sup> During one lecture in MIT Course 2.74, "Optimal Product Design," the professor constructed, with the help of the class, a simple function map for an automobile. Near the end of this project, the professor

Function maps help to trace the flow of various types of information (electrical impulses, radio signals, etc.), material (any type of solid, liquid or gas) and energy (kinetic energy, thermal energy, noise, etc.) entering and leaving a system. The following example helps to explain this in more detail, while also showing how function maps work in general. In this example, the system is defined as any physical object that requires cleaning. (Figure 4.8 illustrates this by having dirt (material) enter the system.) The person, who sees that the object is dirty (information), begins to clean it with soap and water (material). In order for the person to successfully clean the object, however, he has to do work; in other words, he has to hold, wash, and dry the object. The result is a clean object (information) and a container of dirty water (material).



**Figure 4.8:** General Example Showing How a Function Map Works

Function maps must also follow the law of mass conservation and energy conservation. In this example, dirt enters the system and leaves with the soap and water.

This map, of course, can be expanded to show different sub-systems and the flow of material, information and energy from one sub-system to the next. In Figure 4.8, the primary inputs and outputs are shown, and nothing more. The level of detail that one wishes to show in a function map is dependent on the complexity of the system in question, the amount of time allowed to develop the map, and the type of information that is required for a successful analysis. The designer has to determine what specific material, information and energy flows are important to describe and document and what sub-functions have an important role in the design.

The next example shows a function map with more detail. In this example, the function map describes the use of the Rotato™, a hand-held device used for peeling vegetables

---

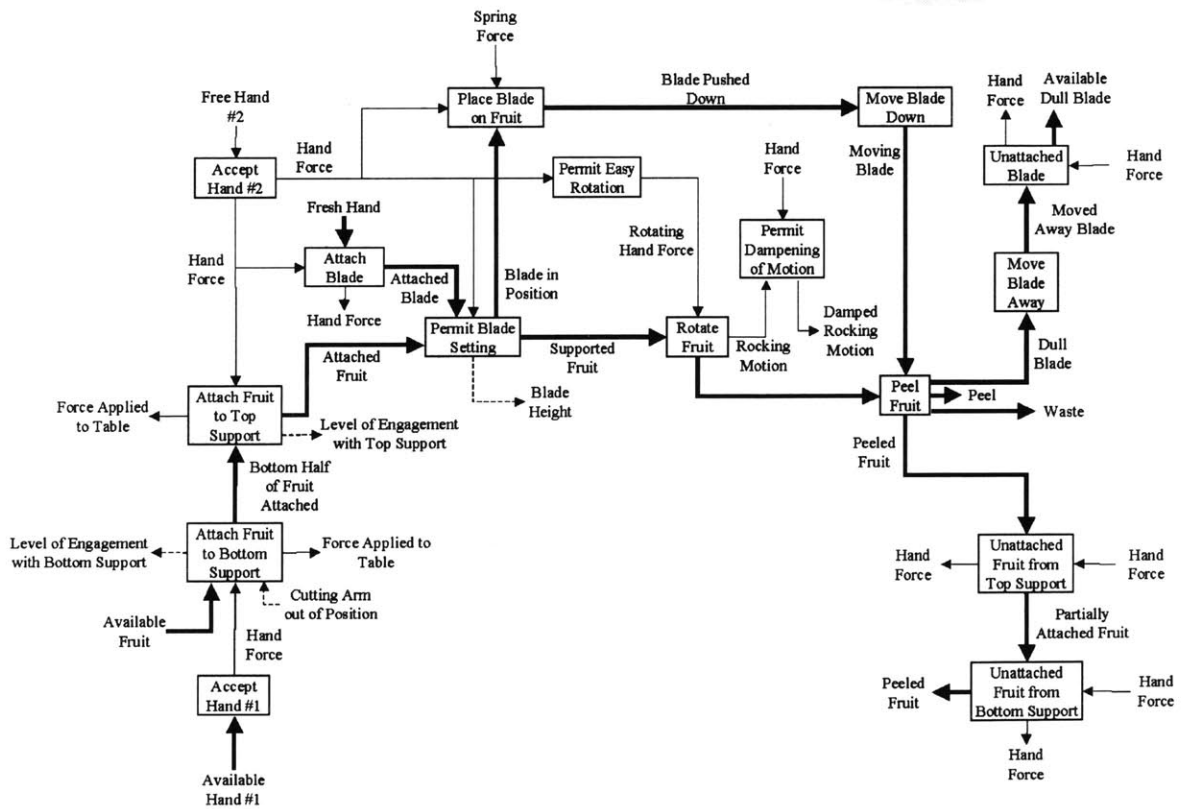
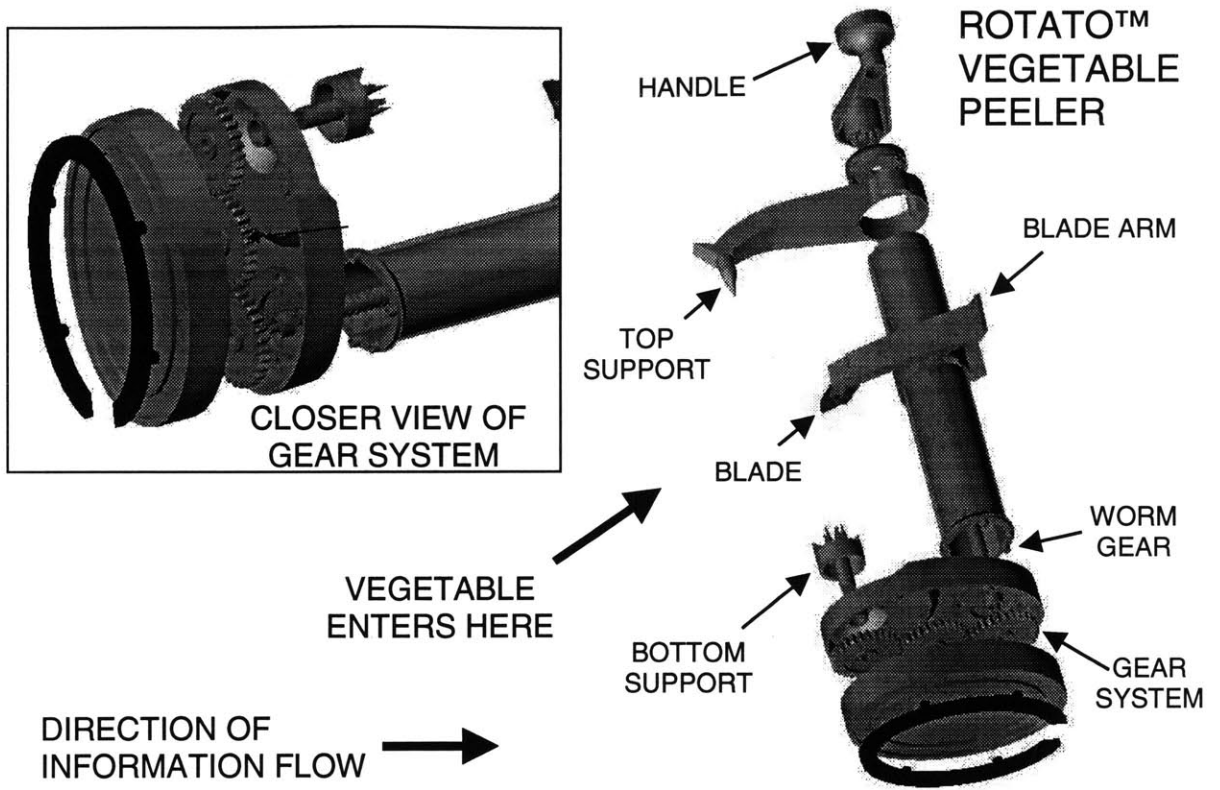
pointed out that some of major sections of an automobile can be integrated into one section. He mentioned, as an example, that the engine and transmission share many common functions and should actually be designed as one unit.

(and fruits).<sup>44</sup> The device works by first mounting a vegetable onto both the bottom and top supports of the unit, as seen in Figure 4.9. (Tiny prongs on each support drive into the vegetable when a force is applied to the vegetable by the user's hand.) The blade arm is moved up along the main shaft until the blade is aligned with the top part of the vegetable. The blade is then set against the surface of the vegetable by a pretension spring. The person operates the device by turning the handle. With an internal worm gear and a gear system, the motion of the handle is translated to the blade arm and to the vegetable. The blade moves down the shaft while the vegetable rotates in place. These actions, together, allow the Rotato™, with the user's help, to perform the necessary peeling operations.

Below the CAD drawings of the Rotato™ is a function map that describes all of the actions just mentioned and others, which may not be obvious to the first-time-user of this device. The map shows that the user is required to hold and position various parts of the device; this means that ergonomic issues play a big part in this design. However, as the reader may see from the drawing below, the unit does not provide adequate areas for gripping or handling. For example, the shaft provides the biggest gripping region. The user, however, has to adjust his hand from the shaft during the middle of the peeling process because the blade arm will eventually run into it as it moves down the shaft. It also provides a way to test possible redesign scenarios. For instance, at a deep enough level in the function map, one can see where certain parts do not provide any use to the key functions and sub-functions. This product has over 20 different parts (not including fasteners). When analyzing this product, it was found that some parts were not carrying any significant functional value; they were merely used as covers. Obviously, part consolidation could be applied in these situations.

---

<sup>44</sup> This function map was done as part of a class project for MIT Course 2.74 that the author was involved in. In addition to the function, the project team surveyed different people after having them use the product; the results were compared against the function map to see which sub-functions were crucial to the design



**Figure 4.9:** Function Map for Rotato™ Vegetable Peeler



Function maps bring order and organizational thinking to the very early levels of concept design or redesign. They allow the designer to see and understand where common and distinct (yet important) functions exist across a design, and provide him with, at least, a direction to follow when attempting to implement design changes (such as part consolidation). Function maps are most effective when they are used to evaluate different models of the same product family. At this level, products can really be seen for their common characteristics. By looking at each map and pinpointing areas where the sub-functions are the same, the designer has a better idea of where standards can be best applied and where modular characteristics can be pursued. In addition, metrics can be attached to the function map. Designers can record the change in value of a particular flow as it crosses through a sub-function. These values can then be attached to various components to determine which model types can be used in different products. As will be discussed in Chapter IX, such a tool can leverage many of the properties that exist in current DFA evaluation tools. (Much of the information presented here about function maps came from K. Otto and K. Wood's unpublished book Product Design: Techniques in Reverse Engineering, Systematic Design, and New Product Development.)

## 4.9 Chapter Summary

This chapter spoke about design for assembly (DFA) and its impact on design and the actual act of assembly. DFA is used to help designers keep in mind issues that are important for making assembly-feasible products. However, current DFA methods are loosely organized, requiring the expertise of the designer to take any of its recommendations and use them in a manner that he sees fit for his work. The assembly advisor when developed and deployed, however, is intended to provide the designer with product specific advice that he can use without additional evaluation.

An attempt was made to show how a hierarchical stepladder of DFA knowledge could be represented. First, general information about assembly and design for assembly were separated into individual components. Next, the components were placed on a product-knowledge grid to determine their relationship with different levels of a product and with each other. Finally, the hierarchical structure was used to show where current DFA methods exist now and where improvements need to be made.

The last part of the chapter discussed part consolidation/integral architecture and modular architecture and their role on design for assembly. Current DFA practices emphasize part consolidation and ultimately drive a product to an integral architecture. However, there may be more benefit to keeping parts differentiated from others, perhaps for the purpose of using some type of modular design. Several issues need to be weighed against each other to determine the right type of architecture to use, provided some given conditions. The ultimate point is that part consolidation is only one choice, and in some cases, it may not be the best choice. The assembly advisor will also need to consider this.

Chapter V will discuss how assembly is considered at Ford and Visteon. The main purpose of this next chapter is to understand what type of user environment do designers and feasibility engineers work in when developing a product. This information will prove useful as the assembly advisor is defined. The assembly advisor, of course, has to complement the work environment that it will be placed in so that designers and engineers are more accepting of the tool and its results.

# CHAPTER V

---

## DESIGN CONSIDERATIONS AT FORD/VISTEON AND THEIR INFLUENCE ON ASSEMBLY

### 5.1 Introduction:

This chapter will explain, in brief detail, how, where and at what level assembly is considered at Ford<sup>45</sup>. Once these questions are answered, the chapter will then explain why assembly is considered in this way by looking at how different vehicle sub-systems are generally designed and manufactured. (The reader should note that the sub-systems, which the author has focused on, were mostly designed, manufactured and assembled by Visteon. There are situations, of course, in different product business units where certain specifications for a sub-system are given to second-tier suppliers (Visteon suppliers); the suppliers then design, develop and deliver the product around these specifications.) This will require that the following questions be answered at some level of detail:

- What contributing factors determine the pathway that designers choose for their designs? Why do they even consider these factors over others?
- Where does a lot of the information that designers use come from? How does this information influence the way designers think about their designs?
- What are the roles of the manufacturing and industrial engineers? How do they fit into the process of designing vehicle sub-systems?
- How is knowledge at the level of manufacturing and assembly presented to designers as they continue work on existing projects or plan for future projects? Is this necessarily the most effective way?

The author should note one very important detail: because Ford is such a large company with approximately three-hundred thousand employees worldwide<sup>46</sup>, designing, testing, buying, manufacturing and assembling literally hundreds of thousands of different

---

<sup>45</sup> For the most part, within the context of this chapter, the author considers Ford to mean both Ford and Visteon.

<sup>46</sup> This number came from a recent issue of Fortune Magazine ("Fortune 500" Issue, April 26, 1999)

parts<sup>47</sup>, the author does not want to give the reader the impression that the information contained within this chapter describes how vehicle sub-systems are designed and manufactured within Ford. In helping to write this chapter, the author focused particularly on two types of vehicle sub-systems—instrument panel (IP) and console sub-systems and heating, venting, air and cooling (HVAC) sub-systems—and how each is generally designed.

This chapter will help the reader understand some of the reasons why assembly (as mentioned in the previous chapter) is not considered a major design driver at Ford. In essence, Chapter V will show how the theory of design for assembly really applies in a design environment where thousands of parts are designed (and in many cases, redesigned) to conform to a limited and somewhat inflexible physical (package) space. It will also consider many of the issues that designers and engineers at every level have to contend with besides making products that are relatively easy to manufacture and assemble and why some of these issues constrict many of the practices considered important for system-level engineering.

## 5.2 Lean and Mass Production and Their Contributions to Both System Engineering and Design for Assembly:

Lean- and mass-production methods are normally compared to one another at the level of manufacturing and assembly, but their differences originate at the level of design. One way to see how the two methods differentiate from one another, particularly at the level of design, is by determining how information is transferred between people across the same level of an organization; across different levels of an organization; and across different organizations all together.

Mass production, as defined by Kalpakjian in Manufacturing Engineering and Technology, is the production of goods in “quantities of 100,000 and over, [requiring] special purpose machinery, called dedicated machines, and automated equipment for transferring materials and parts.” Lean production, on the other hand, is defined as<sup>48</sup>:

---

<sup>47</sup> Currently, Ford uses 24 different platforms to produce over 50 cars and trucks. (Refer to Thinking Beyond Lean by M. Cusumano and K. Nobeoka.) If the reader assumes, for now, that Ford produces a total of 48 cars and trucks (an average of two vehicles per platform) and that each car and truck consists of approximately 15,000 unique parts, then Ford buys or manufactures and assembles approximately 360,000 unique parts every four years—the average life-span of a part in a vehicle. However, this number represents the extreme case; Ford, like many other companies that develop a wide variety of products, emphasizes part reuse and part sharing to minimize and control product costs, design and manufacturing times, inventories levels, etc. A more realistic number would be approximately 275,000. This assumes, however, that 50% of the parts from each vehicle are common to it and its “sibling” vehicle; 25% are common across all platforms; and the remaining 25% are unique to each vehicle. (These figures are based on what the author feels is a reasonable distribution of common and unique parts commonly found in most vehicles.)

<sup>48</sup> Refer to the article “Lean Production: The End of Management Whack-a-Hole” by Michel Baudin (Dated 3/16/99). The article can be seen in its entirety at the following web site: <[http://www.mmt-inst.com/End\\_of\\_management\\_whack\\_a\\_mole.html](http://www.mmt-inst.com/End_of_management_whack_a_mole.html)>.

“The pursuit of concurrent improvement in all measures of manufacturing performance by the elimination of waste through projects that change the physical organization of work on the shop floor, logistics and production control throughout the supply chain, and the way human effort is applied in both production and support tasks.”

The basic properties for each method are listed in the table shown below<sup>49</sup>. All of the properties, as displayed within each of the two columns, have been arranged in such a manner to allow the reader to see the striking differences between the two production methods.

MASS PRODUCTION	LEAN PRODUCTION
<ul style="list-style-type: none"> <li>• High volumes/large production capacity</li> <li>• Vertical integration</li> <li>• Assembly workers performing a single task</li> <li>• High division of labor</li> <li>• Dedicated assembly tools</li> <li>• Large batch sizes</li> <li>• Large levels of inventory</li> </ul>	<ul style="list-style-type: none"> <li>• Just-in-time (JIT) techniques</li> <li>• Strong relationship between OEM<sup>50</sup> and outside suppliers</li> <li>• Assembly workers performing multiple tasks</li> <li>• Synergy between management and labor force</li> <li>• Flexible tools (Short time required for changing dies)</li> <li>• Small batch sizes</li> <li>• Small levels of inventory</li> </ul>

These two lists essentially show how both mass- and lean production practices differ at the factory level; it is at this level that both production methods show their greatest impact and certainly their most distinguishable properties. In general, mass production is used whenever large quantities of manufactured goods are required in a relatively short period of time, while lean manufacturing is used whenever flexibility is absolutely essential (e.g., when developing small batches of different products in a short period of time). These ideas should be considered when reviewing each production type in the table above.

These methods are also distinguishable by the way that they approach and solve production problems. With mass production, problems at the production level are naturally expected: the production process is infused with several testing and rework stations, which provide nothing more than quick fixes to problems that are, in many cases, deeply fixed in the product design and development processes. With lean production, on the other hand, the philosophy is quite the opposite: “minimize the

<sup>49</sup>Refer to <<http://web.mit.edu/2.996/www/lecture4/tsld006.htm>>, MIT Course 2.996: Integrated Product and Process Design.

<sup>50</sup> Original Equipment Manufacturer

number of mistakes made during the production process in order to achieve a perfect product the first time.”<sup>51</sup> To accomplish this feat, checks must be made—not merely at the production level—but at the design level.

In The Machine that Changed the World, P. Womack, D. Jones and D. Roos describe where rework is mostly emphasized and why in each of the two production methods, by looking at the number of workers assigned to a project, over the life of the project. Here is what the authors had to say:

“The result is a striking difference in the timing of the effort devoted to a project. In the best Japanese lean projects, the numbers of people involved are the highest at the very outset. All the relevant specialists are present, and the shusa’s<sup>52</sup> job is to force the group to confront all the difficult trade-offs they’ll have to make to agree on the project. As development proceeds, the number of people involved drops as some specialties, such as marketing assessment and product planning, are no longer needed.”

“By contrast, in many mass-production design exercises, the number of people involved is very small at the outset but grows to a peak very close to the time of launch, as hundreds or even thousands of extra bodies are brought in to resolve problems that should have been cleared up in the beginning. The process is very similar to what we saw in the assembly plant: The mass-producer keeps the line moving at all costs but ends up doing massive amounts of rework at the end, while the lean producer spends more effort up front correcting problems before they multiply and ends up with much less total effort and higher quality in the end.” (Pg. 115)

As the passage mentions, each production method has quite a different approach to product design, prototype development, and final production. With mass production, problems are solved only after they emerge on the production floor; this is considered the extreme case of reactive design and development. With lean production, however, problems are solved before they occur on the production floor, making this the extreme case for proactive design and development. In most automotive companies, a combination of the two methods is used.

---

<sup>51</sup> Refer to the web site <<http://two.mit.edu/V110/N40/car.40n.html>>. The web site shows a copy of an interesting article published in the MIT Tech (one of MIT’s major school newspapers) on Oct. 5, 1990. The title of the article is “MIT Professors Publish Car Study.” The article describes the results of an extensive automotive study conducted by Womack, Jones and Roos, which were descriptively presented in their book The Machine that Changed the World. For a more in-depth study of lean manufacturing, the reader should refer to the actual book or similar books. The author provides a list of books about lean implementation strategies in the Suggested Reading Material Section.

<sup>52</sup> The shusa is the leader of a project

Although lean production has several more benefits than mass production (e.g., more flexibility and less generated waste and cost per product manufactured) the fact is that both production methods have an equal assortment of challenging tasks that must be continuously looked over and satisfied. With mass production, corrections are constantly made without a real understanding of how these corrections influence other parts of the process. As a result, it is imperative that measures be constantly and immediately taken to correct problems—even those that emerge unexpectedly, without any reason—as quickly as they arise. With lean production, the emphasis is on system-level design, in order to anticipate potential problems before they happen. This requires that the interactions between parts, components, and sub-systems be traced out before they are manufactured and that communication between members on a project never be restricted.

All of the important points made, up to this point, about lean and mass production are summarized in the following table:

MASS PRODUCTION	LEAN PRODUCTION
<ul style="list-style-type: none"> <li>• Reactive approach to design and development</li> <li>• Infuse rework stations into production process</li> <li>• Increase the number of employees over the course of development</li> <li>• Rework happens at the production level</li> <li>• “Over the wall” design</li> </ul>	<ul style="list-style-type: none"> <li>• Proactive approach to design and development</li> <li>• Design products so that they can be assembled correctly the first time</li> <li>• Decrease the number of employees over the course of development</li> <li>• Rework happens at the design level</li> <li>• System-level design</li> </ul>

### 5.3 The Use of the Lean Production System and System Engineering at Ford:

Over the years, several books and papers have been written about the lean- and mass-production systems, and how they are commonly associated with Japanese and U.S. automotive companies, respectively. However, in recent years, this distinction has been somewhat blurred, as more U.S. car companies are following the techniques found in the lean production system.<sup>53</sup> Ford, for instance, has developed a way to help improve up-front preparation, design and experimentation with the use of the Ford Product Development System (FPDS). The overall objective to this system is to help link customer requirements with the design of various vehicle sub-systems, through a formalized set of procedures and with the help of CAE tools.<sup>54</sup> (It should be equally noted that, at this point, Ford has not yet perfected a way to completely translate the

<sup>53</sup> In the last ten years, America’s “Big Three”—GM, Ford, and Chrysler—have used the techniques commonly conveyed in the Toyota Production System to reestablish themselves in the automobile industry. One book that mentions this in detail is M. Cusumano and K. Noeboka’s book, Thinking Beyond Lean.

<sup>54</sup> Refer to Q. Dong’s Master-of-Science Thesis, “Representing Information Flow and Knowledge Management in Product Design Using the Design Structure Matrix”, February of 1999, Department of Mechanical Engineering, MIT.

information that it collects from its customer surveys and interviews to the level of the individual parts of a vehicle. This comment was made by a superintendent at Visteon's Sterling Axle Plant.)

Although Ford, in particular, has made an extensive effort to structure its design process to meet multi-level requirements (e.g., customer requirements through FPDS and component sharing and standardization through the Ford 2000 program), there still seems to be an insufficient amount of information pushing its way to design from manufacturing and assembly. "Over the wall" design still appears to be used whenever any vehicle sub-system is considered for manufacture-ability and assemble-ability. In other words, lean techniques have not been fully implemented at the connecting point between design and manufacturing. The following quotations from a manufacturing engineer help to confirm this—and also explain why this process is normally followed:

"There's no...there's no real simultaneous effort put into that to make sure that everything can be assembled as it's designed...no. But also you got to remember that we're using basically the same components for everything that we have out there...and we have a history of having lived with that for a long time. So we know pretty well how to assemble it."

"But there are other issues like, on the Windstar...It really doesn't surface until we...they ramp-up production. They don't tend to identify these things early on. And...it would be nice if you could come up with something to help solve that. To me, that's a big issue. I mean, the biggest issue that I've found, in all of our programs, is the lack of the assembly plant in telling us what the issues are until we ramp up before production. And then, all of a sudden, we get overwhelmed with issues that they want fixed right now."

At Visteon, designers are not totally aware of many of the issues that get identified during production. The reason(s) for this is/are due to any one of the items shown below. (Note: Key excerpts from an extensive interview, which the author had with an industrial engineer (IE) and a manufacturing engineer (ME) at the Sheldon Road Plant, have been included below each item to support the author's reasoning. The manufacturing engineer works with the designers and tooling manufacturers to come up with a feasible design—one which can be manufactured and assembled at a relatively low cost. The manufacturing engineer also develops a working prototype of the design so that any manufacturing-, assembly- and performance-related problems can be identified and corrected before actual production begins. The industrial engineer determines the assembly layout—in other words, the sequence of assembly operations, the location of equipment and workers, and the type of assembly method that will be used, to name a few.)

- The problem is not considered serious enough to be brought to the designer's attention. In this situation, corrective measures are made to the process—not to the



product. (If an assembly issue is, in fact, very serious—for example, if it affects the performance of the product—then the problem and its solution are documented in the next version of the vehicle sub-system’s design guide.)

ME: “Generally, if there's something that's glaring and outstanding...it'll be put in the design guide. If not, then...it's not.”

- There are no formal guidelines that show how particular vehicle sub-systems can be designed for the purpose of manufacturing and assembly. The manufacturing engineers rely on direct contact with the designers. However, it appears to be that not all of the key manufacture-ability and assemble-ability issues fully make their way to the designers and into the designs, themselves. Designers may not have the time or the know-how to address these issues.

ME: “Well, we've developed certain rules, but they are not really published. We have certain rules on what's worked and what hasn't...what's feasible. But there is nothing published on that. It's generally...we work with the designers, and the designers work up-front to eliminate everything we know is a problem.”

- The role of the manufacturing and industrial engineer, at the time of design, is to determine whether or not any of the designer’s work can be feasibly manufactured and assembled. For the most part, they are there to critique the designer’s work—not to persuade him to make certain design decisions.

IE: “From the Sheldon Road standpoint, no we don't get involved in the packaging<sup>55</sup>. It's primarily the design engineers that do that. But they deal directly with the program office to gain their package space.”

ME: “That's all basically set before we get into it. We have a certain package to work with, and they...they set the architecture and make it work within, and then we get involved. And then they say basically to us, ‘Can you make it work?’ And we say, ‘Hey, no we can't. We've got to have this...etc.’ But we're limited to package space. So the basic thing again, it's set up-front by the designer and the design engineer.”

- The product design normally drives the process design. The people at the production level must see to it that some type of production process be made for the existing design, regardless of how efficient the process is.

---

<sup>55</sup> Packaging space is the physical region that a sub-system is limited to. In a vehicle design, each sub-system is given a general package space; designers have to make sure that each sub-system design accommodates this space—and does not go beyond its borders. Packaging space will be talked about more throughout the rest of this thesis.

IE: “We actually had a problem putting on the blower motor cooling tubes so we got a heat light over there, over the actual parts, to make them softer...so they would go in the housing easier.”

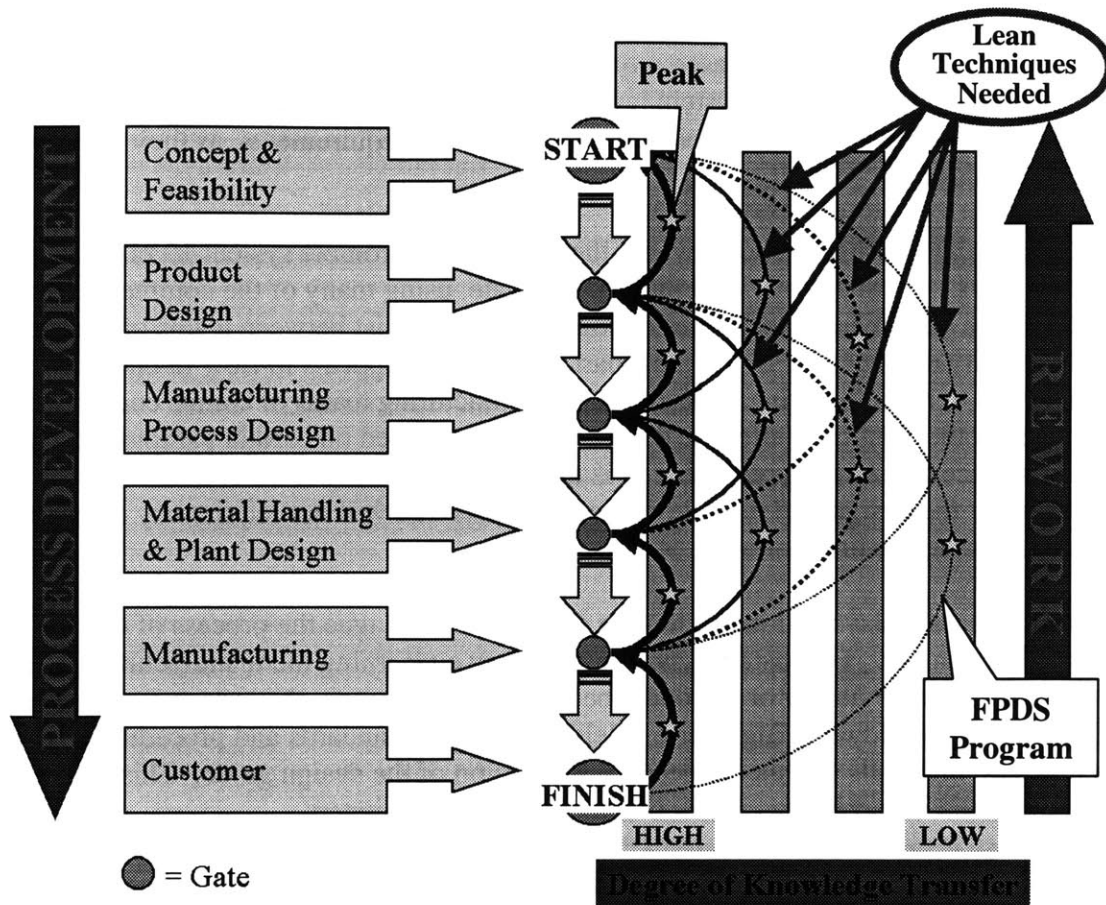
#### 5.4 The Flow of Information from Concept to Customer:

The figure below (Figure 5.1) attempts to summarize what has been mentioned up to this point within this chapter by illustrating how communication flows up and down through the information pipeline at Ford.<sup>56</sup> The development process flows from the top of the figure down to the bottom (as shown by the arrow at the far left) while the rework process flows in the opposite direction (as shown by the arrow at the far right). The semi-circular arrows, as a whole, represent the rework process. These arrows also show where information goes when a problem arises and needs to be corrected. In general, the figure tries to illustrate one very important point: the bigger the gap between various stages of the development process, the less likely it is that information—important information—will make its way back to the various places where it is essential for product improvement.

The most likely and least likely used information pipelines (and all others in between) can be identified in one of two ways. The first method is rather simple: look at the thickness of each line in question—a thick line means a relatively high degree of knowledge transfer while a thin dotted line means a relatively low degree of knowledge transfer. The second method requires finding the peak of the line and seeing where it falls relative to any one of the four rectangular regions shown in the figure.

---

<sup>56</sup> Figure 5.1 was recreated from a figure found on a Ford internal web page. This figure, however, was altered to convey a point, which was not made by the original one—the rate at which information flows between any two steps in the process is based on their proximity.



**Figure 5.1:** Associating Process Development with Rework and Knowledge Transfer

At many points in the development process, gates (or checkpoints) exist. These gates are used to help those working on a project best understand how far along the project actually is with respect to how far along it should be. The gates also help a project move along from one part of the process to the next. This is important for the reason that it helps to prevent extensive repetition within a development stage. (The diagram does not clearly show this type of iteration process, but it is certainly there.)

The gate process does have one obvious disadvantage—although there may be others. Each gate acts in a way like a one-way valve, where information flows only in one direction. Although this is not completely true, there is some evidence that Ford, in particular, has some difficulty relaying important information back to the design level.

Figure 5.1 shows where Ford is now enforcing a greater amount of knowledge transfer: the FPDS (Ford Product Development System) program is helping Ford align customer needs with product design. (Visteon is implementing its own product-design strategy called VPDS (Visteon Product Design System) that uses many of the techniques in FPDS

while elaborating on others.) This product-development program works by identifying at what level different customer-based requirements impact product design—from the vehicle level down to the sub-system level and beyond. D. Whitney mentioned that FPDS is used, at this time, to simply check-off different requirements, as they are met in the vehicle design<sup>57</sup>.

In addition, the figure also shows where there is need to promote system-level initiatives in other ways. As Section 5.3 attempts to illustrate, using many of the comments from Ford/Visteon employees, weak lines of communication exist between teams of designers and feasibility engineers. For the most part, information appears to flow in one direction, from design to manufacturing. Design and manufacturing teams of course communicate with one another, but only after the design team has decided on certain parameters (e.g., mating surfaces, fastener types, etc.). It is up to manufacturing team to determine the feasibility of the design and provide safe suggestions that will not require any elaborate modifications to the original work.

In several cases, delays and inefficiencies often seem to plague the process of quick knowledge transfer. For instance, whenever issues concerning the manufacture-ability and/or the assemble-ability of a product become severe enough, they are documented using WERS (Worldwide Engineering Release System) standards and procedures. The documented materials are later added to next version of the design guide. This, however, takes time, and in most cases is still overlooked by the designers.

In the next chapter, many of these information gaps will be verified in another way—by evaluating all of the different assembly “mistakes” that went into, go into, and still continue to be in several of the climate control units designed and fabricated at Ford. Figures and comments from engineers working at the Climate Control Organization will be included to clarify and support key points.

The author has emphasized the fact that there seems to be an insufficient amount of knowledge transfer during the iterative/rework process, but as the author will demonstrate in the next chapter, there also seems to be the same lack of knowledge sharing within a product family. In other words, it seems to be that certain attributes, which have been proven to effectively improve assembly in one product design, are not extensively utilized in other products. This will be demonstrated in Chapter VI by showing the different favorable and unfavorable characteristics in each HVAC unit analyzed by the author, from the point of view of assembly.

## 5.5 Collecting and Transferring Knowledge Using Design Guidelines:

Each product unit within Ford and Visteon has its own set of design manuals to be used as reference by the designers; they provide both novice and expert designers with

---

<sup>57</sup> D. Whitney made this comment during a discussion with the author and the manager of the KBE-DFMA team. The discussion took place at the Danou Technical Center in June of 1998.

relevant information on particular components of a sub-system—from various points of view, including assembly, appearance, packaging, and serviceability. The guidelines are, in a sense, collections of information from designers, industrial engineers, manufacturing engineers, and product engineers, from both the past and present; the guidelines help alert designers of past design mistakes so that they are not repeated in the new designs.

The information contained within the manuals pertains mostly to part and component design, and rarely considers design at any higher level than that. The manuals provide both text and general illustrations about key components in the design of a sub-system. The following table goes on to show what most of these manuals provide and do not provide:

Provides:	Does <u>NOT</u> Provide:
<ul style="list-style-type: none"> <li>• Detailed drawings of various components</li> <li>• Brief description about the purpose of each component</li> <li>• Reasons for enforcing certain guidelines (In many cases, this is true.)</li> <li>• Problems to avoid from past experiences</li> <li>• Standard structure that is followed by all design guides</li> <li>• Equations, charts and tables to help evaluate the performance, structural and dynamic integrity, etc. of various components</li> <li>• Purpose for each rule</li> <li>• In the case of the IP Sub-System Manuals: An easy-to-read table that summarizes the manual. The contents of the table includes a brief description of every design rule; the parts of the sub-system that they focus on; and the design issues that they emphasize (e.g., appearance, cost, maintenance, etc.)</li> </ul>	<ul style="list-style-type: none"> <li>• Complete exploded-view drawings</li> <li>• Direct cause-effect relationships with each design decision</li> <li>• Date when rule was first documented in guide (Outdated and obsolete rules are sometimes used, without warning, in current designs.)</li> <li>• Sequence of steps or method for designing a vehicle sub-system</li> <li>• Design for manufacturing and assembly beyond parts and features</li> <li>• Explicit and implicit tradeoffs that need to be made by designers, and the weight of each design decision</li> <li>• Regions of the vehicle sub-system where designers have the ability to be creative in their work</li> <li>• An index or some type of search capability</li> </ul>

Table 5.1: General Attributes of Component Design Manuals

The manuals are not intended to replace the experienced designer; the manuals are simply used as reference materials<sup>58</sup>. They do not capture, in any way, all the knowledge and experience that a designer of ten or more years has. This is why KBE makes sure that, with every project, an experienced cross-functional team is formed. KBE, as mentioned earlier, goes through a knowledge acquisition process, which includes gathering the necessary information from the design guides and the logic needed to implement these

<sup>58</sup> Most designers will use the manuals as a supplemental aid. Designers tend to find it quicker and easier to acquire important information by conversing with colleagues. As it was mentioned to the author, the manuals are not necessarily set up so that specific details can be easily found.

rules from the cross-functional team. Therefore, it is evident that the design guides will not show a detailed process for how to use the raw information that they contain. That type of information can only be extracted from an experienced team.

Although all design manuals developed and used within Ford follow a generic format, many of them provide information in different ways. The general requirements used when writing these guides focus particularly on how the information should be formatted, but again, it does this with a great deal of inconsistency.

The next two sub-sections will discuss particular aspects of the climate control and instrument panel design manuals. (Each sub-section will look at one or two particular areas in each of the two manuals. By no means does either section provide a complete overview of the manuals.) Although both types of manuals follow the general criteria and share many of the characteristics mentioned here, they do have different ways of representing information. The author will not discuss what type of raw information is commonly found in either manual; instead the author will determine how the two manuals manage and present this information to its readers.

### **5.5.1 Climate Control (CCO) Design Manuals—Steps for Selecting Components for HVAC Sub-systems**

In many areas of the CCO manual, there is an attempt to describe the logical sequence of steps required to either select or design a specific component. However, the information is provided in text-format (i.e., the step-by-step process is described in a “story-like” fashion). The manual definitely shows some cause-effect relationships, by describing to the reader what type of standard components are available for him to choose from, at each level of the decision-making process.

The author suggests that a decision-tree or some other type of graphical aid be used instead. The designer could then see how each decision leads him down a particular pathway; he could see and understand the logic behind each decision, at any level, faster and more clearly than in the format currently used by the manual. Reading information in this way can also give the designer the ability to see how his choices affect certain design parameters and how they limit future choices.

Figure 5.2 shows a small example of this idea (although it definitely lacks the finer qualities of a decision-based structure). Here decision-based information concerning the heater core and its connecting attributes are laid out graphically in a simple matrix. In the actual manuals, the information is expressed entirely with words.

Heater Core Dimensions	6" X 8"		7" X 8"		6.5" X 6"	
Connector Tank Size	SHORT	TALL	SHORT	TALL	SHORT	
Position of Connector Tubes	SIDE	TOP	SIDE	TOP	SIDE	TOP
Tube End Size	OVAL	3/4" 5/8" OVAL	OVAL	3/4" 5/8" OVAL	OVAL	3/4" 5/8" OVAL
Outer Diameter for Tube Routing	3/4"	5/8"	2/3"	3/4"	5/8"	2/3"

**Figure 5.2:** Part-Selection Matrix for the Heater Core Design

Figure 5.2 is segmented into three columns—one for each set of heater core dimensions. (This pivotal characteristic is the root of the decision process.) The arrows show the sequence of steps for selecting particular features on the heater core. The shaded areas represent a break in the association, along different points in the decision-making process. (In other words, previously mentioned characteristics for the heater core do not influence the selection process below these shaded areas.)

The design guides also show that the tube configurations must fit within a particular box size—although it is not shown in Figure 5.2. This information is based on the heater core size and whether or not the tubes stem from the side or from the top of the tank.

Similar information about part dimensions can also be found in the manuals for the blower motor, the evaporator core and other types of sub-assemblies. Therefore, a part-selection matrix can be created for each of these HVAC components, as well.

Some of other advantages with this idea were discussed briefly at the beginning of this sub-section; the author extends this list below:

- Designers prefer to see information in the form of tables and graphs
- Information is relatively easy to find
- General ideas behind most tables and graphs can be inherently understood
- Raw information can be easily up-dated, assuming that the heuristic information does not need to be altered
- Tables and graphs allow knowledge-based systems to be developed with greater ease



However, even after a reformatting the text-based information into an easy-to-read table, the part-selection matrix lacks some very important qualities. The matrix, like the manuals, does not provide any substantial support or guidance as the designer makes his way through the selection process. (This point was also mentioned in Section 5.5.) The matrix merely points out what part dimensions are available at each level of the selection process and what part dimensions are available at the next stage. However, it does not provide reasons for selecting one dimension over another (tradeoffs) or consequences after the decision has been made (cause-effect relationships). Including these parameters needs to be the next step in the development of this selection structure.

### 5.5.2 Instrument Panel (IP) Design Manual—Associating Design Issues with Key Components

At the beginning of each Instrument Panel (IP) design manual, there is a grid, which gives a description of each rule; the reason why the rule should be followed (based on past experiences); the specific area(s) of the design that the rule affects; and the type of rule it is. The following table helps to show what each of these grids looks like:

DESCRIPTION OF GUIDELINE	GUIDELINE TYPE	COMPONENT TYPE	REASON FOR RULE	DATE OF INTRODUCTION INTO MANUAL
Use standard fasteners	Assembly (ASSY) Manufacturing (MFG)	Ashtray (ASH) Glove Box (GLVBX)	Reduce the number of unnecessary fasten types	Month/Year
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

Although the author does not know the main purpose for presenting the rules in this fashion, the grid does allow certain amounts information to be collected—information that developers of these manuals may not have been aware of. In the Appendix, found at the back of the thesis, is a two-dimensional array, showing the number of times a particular type of rule, influencing a specific area of the product, was mentioned in these manuals. Some bar charts are also included to illustrate certain issues, which will be described later. Each guideline type (and its associated acronym) is displayed at the top of the array, while each component type (and its associated acronym) is shown at the far left of the array. Each number shown in the main part of the array represents the number of times a rule, in association with each component, was mentioned in the design manuals. (Note: many of the rules shown in the manuals were linked with more than one type of guideline and/or component.) The different colored cells merely allow the reader to easily see high and low values across the array; the key at the bottom left of the chart

shows what each color represents numerically. Five different manuals, describing major sections of the instrument panel, were collectively used to form this matrix.

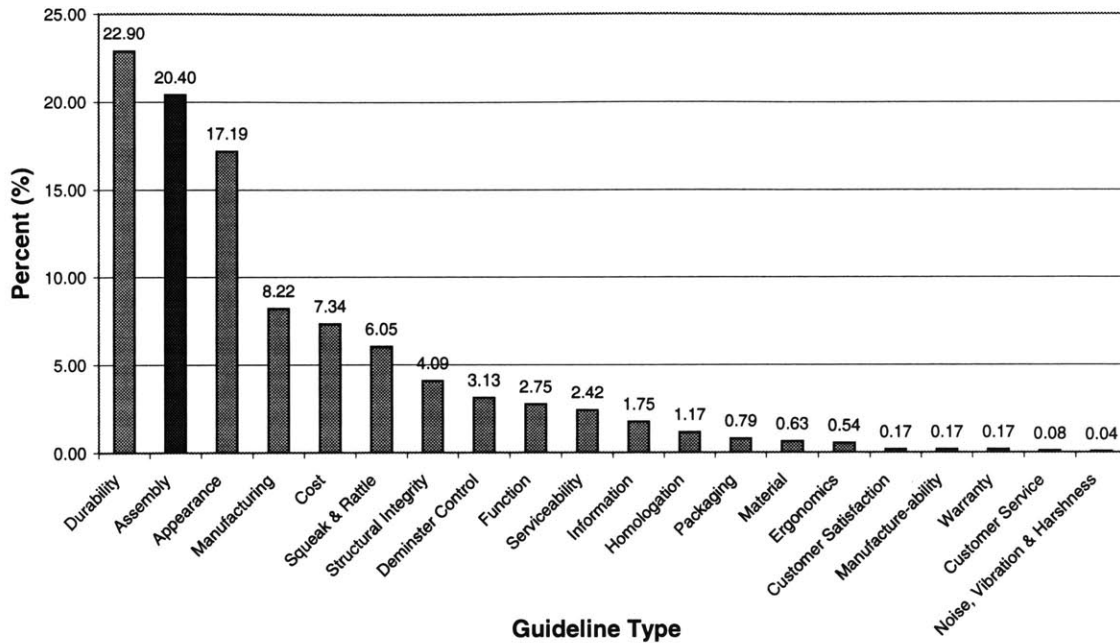
From this matrix, the following bits of information can be extracted:

- What rules and components have the highest (and the lowest) correlation with other rules and components? (The Appendix includes two correlation matrices—one for components and another for guideline types. An explanation for developing and understanding a matrix of this kind is also included in the Appendix.)
- What specific areas of the product are emphasized the most (and the least) in the design guides?
- What issues drive the design of different sub-systems based on the rules shown in these manuals?
- Are these trends consistent with the views of the designers?

The initial reason for developing the matrix using the information from the IP manuals was to see how various relationships and trends compared with the information collected and classified in other manuals, from other product units. The author felt that this method of comparison would be faster and more efficient than attempting to collect the same type of information from surveys, distributed to various designers and engineers in each product unit. Unfortunately, this type of format has only been seen in the IP guides. (The author had reviewed the guides for bumper, chassis, head and rear lamp, fuel tank, and HVAC designs—but was unsuccessful finding a similar type of grid.)

Although IP designers generally speak about how the appearance and the package space drive the design of any instrument panel, this does not seem evident from the matrix (See Figure 5.3 below—a bar chart showing where most design rules are commonly classified.)

## Design Guidelines for Instrument Panels



**Figure 5.3:** Bar Chart from IP Matrix, Showing Assembly as the Second Highest Priority

Some of the reasons for this incompatibility are listed below:

- The criteria for entering a rule into a design manual may not depend on what the primary drivers are for that product.<sup>59</sup> The types of rules that appear to be enforced are those which can be captured and fully utilized at the part and component levels. For example, assembly rules can be found across different levels of product design, but have been traditionally defined at these levels. (Refer to Figure 4.4) Many of the assembly rules found in these guides look at parts and components and usually as separate entities. Packaging, on the hand, is relatively hard to localize into a few component-specific rules without looking at the interactions of various parts, components, and sub-systems. In other words, packaging requires system-level thinking.
- The design guides do not deal very well with issues from a system level—especially if conditions change over the course of a project. Packaging would be a good example. Packaging rules could be made for these types of manuals if the packaging conditions for every vehicle and vehicle sub-system never changed. To explain the dynamics of packaging would require a system-level

<sup>59</sup> As it was described to the author, all guidelines found in design manuals are for the purpose of “reminding” the designer, or keeping him aware, of recent or repeating issues that can be avoided in new designs.

analysis of all parts, components, and sub-systems inside a vehicle—something that the manuals are not able to show. This probably explains why packaging appears to be a relatively minor issue in the manuals, as shown in Figure 5.3.

- The definitions for all guideline types and the criteria for categorizing each of them were not clearly written in any of the manuals. This makes the author believe that his definition for assembly, manufacturing, packaging, etc. may not be consistent with the definitions implicitly used in the manuals.

The most interesting point about Figure 5.3 is that it does not completely reflect the way IP designers think about design. For instance, after speaking to a primary designer<sup>60</sup>, a secondary designer<sup>61</sup>, a manufacturing engineer and an industrial engineer in the IP business unit, it became obvious that the two driving factors in the design of an IP are appearance and packaging. (Refer to Figure 5.4) Assembly is usually considered an after thought in the design of an IP unit; when the primary issues are completely considered, the design is analyzed for manufacture-ability and assemble-ability. However, Figure 5.3 shows a different order of precedence for appearance, packaging and assembly. Out of twenty different design issues, packaging was ranked 12<sup>th</sup>, appearance was ranked 3<sup>rd</sup>, and assembly was ranked 2<sup>nd</sup>. What does this contradiction between what designers and engineers say and what the manuals allude to? The designers and engineers base their answers on all levels of an instrument panel design; they understand how each design issue plays itself out over the course of a project and what it eventually does to “shape” the design of an IP. The manuals, however, have limited “understanding.” The rules that they contain look no further than the individual parts and components that make up an IP; they do not consider such things as design decisions and tradeoffs, time-dependent restrictions and outcomes (packaging conditions, for example, change over the course of a project), and non-linear conditions that seem to make themselves known way after design. Simply stated, part/component knowledge is usually written down in manuals while system-level knowledge remains in the heads of technical experts.

Such a system-level analysis, like the one described above, can be helpful, not only to designers, but also to Visteon KBE. The Department can use the information shown in the three matrices to identify where concerns are in the design of any product. In other words, it can be used to pinpoint problem areas, or areas of high risk and concern, and to find out where emphasis should be placed, when picking out potential projects. Although this method will not provide KBE with any particular instructions to follow, it can certainly be useful when developing a core suite of tools for different product business units.

---

<sup>60</sup> A primary designer is a designer with several years of experience, who oversees the design of a vehicle sub-system. The primary designer, who the author spoke with, had nine years of experience designing instrument panels for different Ford vehicles.

<sup>61</sup> A secondary designer usually has less experience than a primary designer does. (The secondary designer, who the author spoke with, had three to four years of design experience at Ford.) A secondary designer works on a particular part of the vehicle sub-system, and his work is usually overseen by a primary designer.

Unfortunately, there is a problem trying to pursue this idea further. As was mentioned earlier, design guides from different product units do not use the format found in the IP manuals. In order for KBE to go further with this approach, the following would need to be done:

- Determine the requirements for placing specific rules in various categories (i.e., a term like “assembly” can encompass a number of issues), and use these requirements consistently from product unit to product unit.
- Different rules from different design manuals must be evaluated and classified in the same way that the rules in the IP manuals were done. (This may require a lot of effort from the people who developed and helped develop these manuals.)

## 5.6 The Primary Drivers in the Design of Different Vehicle Sub-Systems and Their Contribution to Assembly:

Assembly is no doubt an important aspect in the design of any product<sup>62</sup>, but in many products that have hundreds or even thousands of parts, assembly is not considered the most important one. Due to the surge in technology in product development (e.g., CAD/CAE/CAM) and manufacturing (e.g., robotic equipment), certain design drivers, like assembly, are beginning to have less of an impact on design, while others are appearing to strengthen their presence. This section will discuss what issues are most important now in the design of different vehicle sub-systems at Ford, and how they influence assembly.

What drives the design of a sub-system in many cases varies from product unit to product unit, due to the many different factors that constrain it. In turn, this can affect the way in which products are assembled. For instance, any instrument panel is primarily driven by appearance: the end user simply does not want to see any apparent mating seams. This means that there should be no more than one or two parts that make up the outer assembly. For the purpose of designing an instrument panel that is consistent with the needs of the customer, designers are driven toward the use of an integral architecture. For other products, like the air conditioning (A/C) unit and the engine, which are out of the customer’s sight, visible mating seams are not much of an issue.

In most, if not all cases, packaging and performance<sup>63</sup> issues (at some level) influence how designers will approach a project and ultimately what the vehicle sub-system will look like. Figure 5.4 shows the primary drivers for various products, as told to the

---

<sup>62</sup> As D. Whitney mentions at the beginning of his course 2.875, “Mechanical Assembly and Its Role on Product Development”, very few products consist of just one part. In other words, most products have to be designed, at some level, with assembly in mind.

<sup>63</sup> Performance issues represent the functional attributes of different vehicle sub-systems; examples include the rate at which air flows through an HVAC unit, the number of candelas that radiate from a head lamp, and the appearance of an instrument panel to a customer, to name a few.

author, by designers and engineers in different product units. Generally speaking, assembly and manufacturing are considered important issues by themselves, but when considered with other issues and restrictions, they usually end up having a relatively low level of importance. (No designer or engineer would ever say that the design of a particular vehicle sub-system is driven purely by manufacturing and/or assembly.) As mentioned earlier in this chapter, assembly and manufacturing procedures appear to be important issues only after the design is completed, leaving these matters to the manufacturing and industrial engineers, and to similar technical experts.

VARIOUS VEHICLE SUB-SYSTEMS

		Fuel Tank	Head-lamp	HVAC	IP	Radio
DESIGN DRIVERS	Appearance		X		X	X
	Assembly					
	Manufacturing					
	Packaging	X	X	X	X	X
	Performance		X	X		X

X = Issues that have the highest level of consideration in a design

**Figure 5.4:** Design Drivers for Various Product Designs

Here are some quotations from designers, manufacturing engineers, industrial engineers and Visteon/KBE technical consultants, supporting the information found in Figure 5.4.

RADIO

---

“For example, the radio is driven by packaging and shaping...”  
—DFA Consultant

“I’m not familiar with the radio itself...you know, what’s the design cycle. But I’ve been told that for different vehicles they may use the same radio. So all the packaging has to be the same...it all has to have the same room for this one radio so that it can fit properly into each car.”

—DFA Consultant

---

HEADLAMP	<p>“The problem with the headlamp is that it’s driven by aesthetics...so...it’s rather simple [to analyze from the point of view of design for assembly]”</p> <p style="text-align: right;">—DFA Consultant</p>
<hr/>	
FUEL TANK	<p>“For the fuel tank, there is no styling, okay...just packaging.”</p> <p style="text-align: right;">—DFA Consultant</p>
<hr/>	
HVAC	<p>“...I can’t stress it, package drives the whole thing.”</p> <p style="text-align: right;">—Primary Designer, Air Handling</p> <p>“...the first thing I want to do is to design the best performing system I can to perform all our climate-control requirements</p> <p style="text-align: right;">—Primary Designer, Air Handling</p>
<hr/>	
IP	<p>“So I would say for an IP, the two, if I were given two to choose from: packaging and styling.”</p> <p style="text-align: right;">—Manufacturing Engineer, Saline Plant</p> <p>“...but we’re really driven by the stylists, which we can’t control. I mean, or we have very minimal control. The only time that we have any control is when...the feasibility guys from foam say, ‘We can’t make this part. You’ve got an unfeasible design.’”</p> <p style="text-align: right;">—Industrial Engineer, Saline Plant</p>

---

Designers of any product usually think about two or three important factors, and keep these in the highest regard. In other words, they design products to accommodate these requirements and limitations. Afterwards, they make design changes, where they are needed, by reacting to other issues (of secondary or tertiary importance) whenever they arise. This point was made in Chapter III, when the term “design space” was mentioned and defined.

Because packaging space is considered one of the most important issues in the design of an automobile, it is described separately in the following sub-section. In almost every discussion that the author had with designers and engineers working in different vehicle programs and on different sub-systems, packaging space was considered one of the most emphasized yet one of the least understood drivers in automotive design.<sup>64</sup> Packaging is one of the few drivers in design, which cannot be isolated and explained in detail at any

---

<sup>64</sup> In one sense, packaging space is very much like assembly and the conflicts that it has to face—the magnitude of a change is not directly related to the magnitude of the effect. For this reason, it is at times difficult to find an optimal solution to a problem.

one level. (This point was mentioned in Section 5.5.2.) Furthermore, packaging is probably the biggest contributor to design- and manufacturing-related delays. (If packaging was not an issue in the design and manufacture of a vehicle, then all parts could be made standard and the time to design and manufacture a new car or truck would take weeks or days instead of years or months. In addition, vehicles could be designed around a completely modular architecture, so that product variety can be obtained using a combination of standard parts, much like what is done in the computer industry.)

### 5.6.1 Packaging Space and Its Influence on Integral Design:

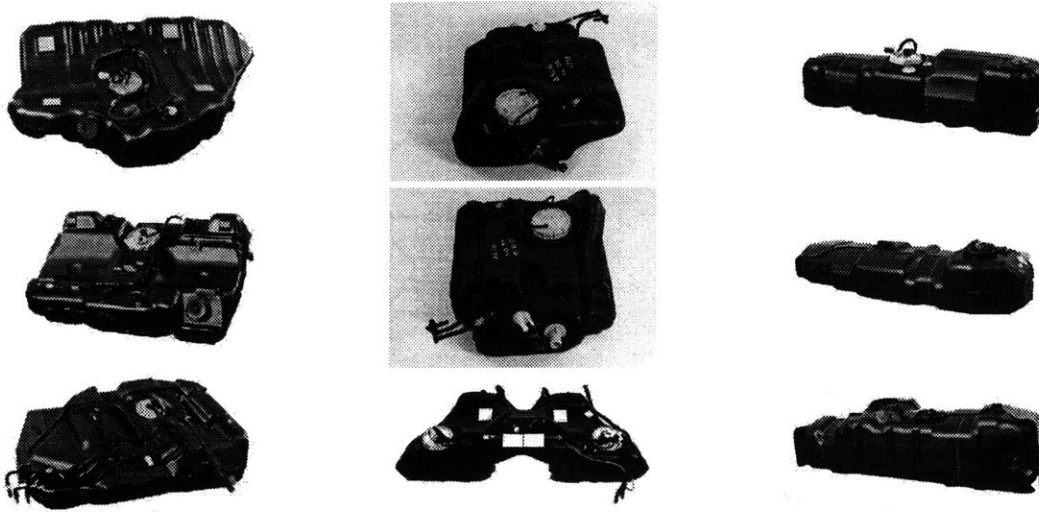
There has been constant technological development in the automobile industry: in the past twenty years, cars have become more and more sophisticated, with new gadgets and safety equipment. They have more sophisticated features inside them, enhancing the driving performance and the driver's comfort level. But in addition to what has been going on "behind the sheet metal," today's cars are small compared to their predecessors. As a result, car companies today are faced with a huge challenge—packaging more parts inside smaller size vehicles. Because of this situation, the demands placed on packaging space and department-to-department communication have increased dramatically<sup>65</sup>. These actions have also promoted the need for part consolidation, not necessarily for the purpose of assembly, but for conserving volume.

With the advances in plastics and blow-molding and injection-molding procedures, the demand for part consolidation can be successfully satisfied. Products, which can be made from some type of plastic material, can now be designed with a larger number of features (in order to contain several functions in one part) and shaped in a way that can fit their packaging configuration. In addition, products or parts of products, which can be shaped into a number of convoluted patterns without affecting their intended performance (i.e., products that act primarily as containers), are able to adapt to radical and immediate changes—especially if they are manufactured by plastic-molding equipment. Take for instance the design of various fuel tanks, wiper fluid tanks, and HVAC casings (all of which act as containers). Other plastic-made products, however, like instrument panels and door interiors, have critical dimensions to them, which need to be conserved throughout the design process. Figure 5.5 shows some of the different configurations used in different fuel tank designs—some are long and slender while others are box-like.

---

<sup>65</sup> Mary Walton's book *Car* really brings some of the issues revolving around package space in an automobile to life. The book describes, in great detail, how the 1996 Taurus was redesigned, from a "human perspective".





**Figure 5.5: Fuel Tank Designs**

As a result, the designs associated with blow-molded or injection-molded parts and products are usually the last ones to be “frozen” before actual fabrication. The time at which certain sections of any vehicle sub-system are frozen depends greatly on the cost for tooling and manufacturing. Most “last minute” changes are usually done to injection-molded parts because slight modifications can be made to the existing molds, in a short period of time and at relatively low cost.

With packaging requirements being what they are and with new materials and manufacturing equipment helping to alleviate this concern, integral design has made a huge impact on the way vehicles are made today. However, packaging space, in some situations can force the dislocation of specific functional requirements and assign each of them to individual parts. (Generally speaking, this is why designers in most vehicle programs are balancing packaging and performance requirements.)

Figure 5.6 provides an example of how packaging space can, in fact, force part dislocation—the opposite of what has been mentioned. The example shows two different blower motor compartments, one for the Windstar HVAC unit (left) and another for the Taurus HVAC unit (right). The reader will notice that the Windstar unit has fewer parts—one door, as opposed to two doors, for example. In both cases, one vacuum motor is used to move the door(s). In the Taurus design, links and a cam are used to move both doors simultaneously.

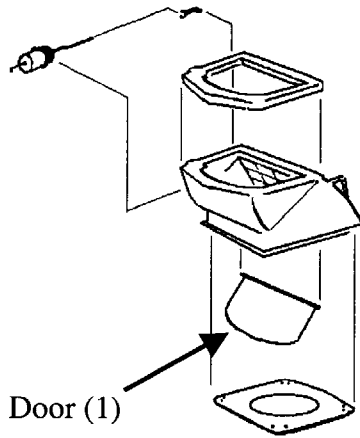


Figure 5.6-A: Windstar Unit

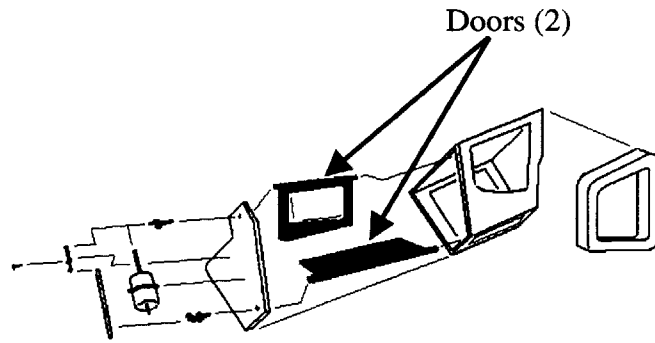


Figure 5.6-B: Taurus Unit

**Figure 5.6: Blower Motor Compartment for Two Different HVAC Units**

Can the Taurus design be made to look like the Windstar design, thus making the assembly simpler? Unfortunately, no. The reason is that the Taurus has a “tighter” packaging space than the Windstar for reasons that are not known to the author. As a result, performance was hindered—the limited packaging space did not allow for an adequate supply of air to enter the blower motor from one side of the unit. The design evidently had to be altered in order to satisfy the performance requirements given to the design team. A second air passageway was designed from another side of the compartment. However, when this was done, a second door needed to be included, as well, thus increasing the total part count.

Consolidating the two doors into one would not work in this case—doing so would violate the packaging and performance conditions that were initially set. This example not only shows why it is important to understand the product in question before even attempting to use part consolidation, it also shows the strong connection between packaging, performance and overall design for an HVAC unit.

**5.7 Chapter Summary:**

The purpose of this chapter was to identify the key drivers in the design of most vehicle sub-systems and to see how they either influence or restrict design for assembly. The chapter was broken up into three general topics:

1. Where in the system does Ford apply Lean Practices; where does Ford still need to enforce these practices?
2. In what way is written knowledge stored and transferred at Ford—and how can these methods be improved? Furthermore, how is this information

- different than the information stored in the minds of designers, engineers and other technical experts?
3. What are the major drivers in the design of most vehicle sub-system and how do they stand with respect to design for assembly?

Assembly is one of the few issues that is not approached systematically, at the level of design. This is due to the fact that assembly is explicitly considered only after a design is completed and the design meets certain packaging and performance requirements. When beginning a project, designers are given a set of requirements provided in a Product Direction Letter (PDL). The PDL helps designers understand what type of characteristics should be considered first in their work. For the most part, the PDL lists certain packaging and performance specifications; however, issues like assembly and manufacturing are rarely mentioned. As a result, assembly is not really an issue that drives the development of any vehicle sub-system; it is mostly considered during a feasibility test or prototype build.

The point that should be made here is that in order for an assembly advisor to really make a contribution to any design, it must be able to complement many of the issues that drive the design in the first place. Furthermore, the advisor should be able to identify the different characteristics that drive any of the designs that it plans to analyze. By recognizing these attributes, from design to design, the advisor is able to give specific recommendations that consider not only assembly but also particular attributes about the design.

In the next chapter, many of the issues revolving around product design will be examined more closely, from the level of the end product. In Chapter VI, the author will investigate how four different HVAC units are designed and assembled. A few examples using different parts of the Ford Windstar and Ford Taurus units were dispersed throughout the first five chapters of this thesis. In Chapter VI, a number of different and interesting characteristics about these and other units will be exploited.

# CHAPTER VI

---

## HVAC (HEATING, VENTING, AIR, AND COOLING) CASE STUDY

### 6.1 Introduction:

Chapter VI will compare and discuss some of the typical assembly architectures commonly found across different HVAC (Heating, Venting, Air and Cooling) designs. The chapter will talk about these architecture types using four current HVAC units as examples—the Ford Windstar, the Ford Taurus, the Toyota Camry, and the Ford CDW-27 (European design). (In actuality, Visteon designs the HVAC units for Ford, and Denso designs them for Toyota. Throughout this chapter, the author will use the name “Ford” to mean both Ford and Visteon and “Toyota” to mean both Toyota and Denso.) Pictures showing important areas of each unit will be provided in order to help the author better describe and compare certain key aspects of each assembly.

The layout for this chapter will be as follows:

1. Describe what a typical HVAC unit does and how it does it.
2. Describe how various surrounding factors (e.g., internal/external packaging requirements, architecture type, performance requirements, etc.) influence the design of a unit and its assembly.
3. Compare each of the four units with one another from the points of view of different architecture types, assembly methods, and company practices. (The reader should keep in mind that three of the four units are Ford designs while the remaining one is a Toyota design.)

The purpose of this chapter is to apply some of the ideas, which have already been expressed in this thesis, to an actual product. The reader should be alert to the fact that this chapter will not focus on the design of individual parts and components and their somewhat modest implication on assembly. The focus here will be on the design of each casing and their central significance on assembly, as a whole. As the reader will soon

realize, the casings have the most impact on how assembly is considered: each of the unit's 30 or more components (not including fasteners) are either assembled to the interior surfaces or to the exterior surfaces of the main casing.

## 6.2 Motivation: Why Analyze this Particular Vehicle Sub-System from the Standpoint of Assembly?

Before delving into any part of the analysis, it is important to first understand why this particular sub-system was chosen. HVAC units were looked at from the perspective of assembly because the results from the analysis can have a significant and immediate impact on the next generation of HVAC designs. (The author also explains, in some of the points below, why the casing is the most critical element in the design of an HVAC unit and why it is a major consideration in this analysis.):

- Very little has been done to improve the overall design of any of these units from the aspect of assembly. Several of these units have been optimized at the part/component level, but very little has been done beyond that. The actual results from the analysis, which will be seen later in this chapter, come from evaluating the entire design and identifying where certain changes can be made to improve assembly, as a whole.
- The casing, in many ways, is the key to the design of any HVAC unit. The casing locates every component in 3-dimensional space; instigates the choice of assembly; contains several architectural parameters; and navigates air through the unit and out to the passenger compartment. A typical unit can be designed more effectively for assembly by simply modifying the casing design—in particular, reconfiguring the mating lines that segment the casing into separate pieces. But any modifications to the casing will have to keep in mind many of the other functional requirements just mentioned.
- The casing is one of the few components in an HVAC unit that is flexible from the perspective of design<sup>66</sup>. As a result, it can be redesigned without seriously influencing the cost of the entire product.
- Several of the suggestions mentioned in this analysis can be used for other vehicle sub-systems that have a “flexible” casing and a number of components attached to it. Some examples include the fuel tank and washer fluid tank (with peripheral parts).

---

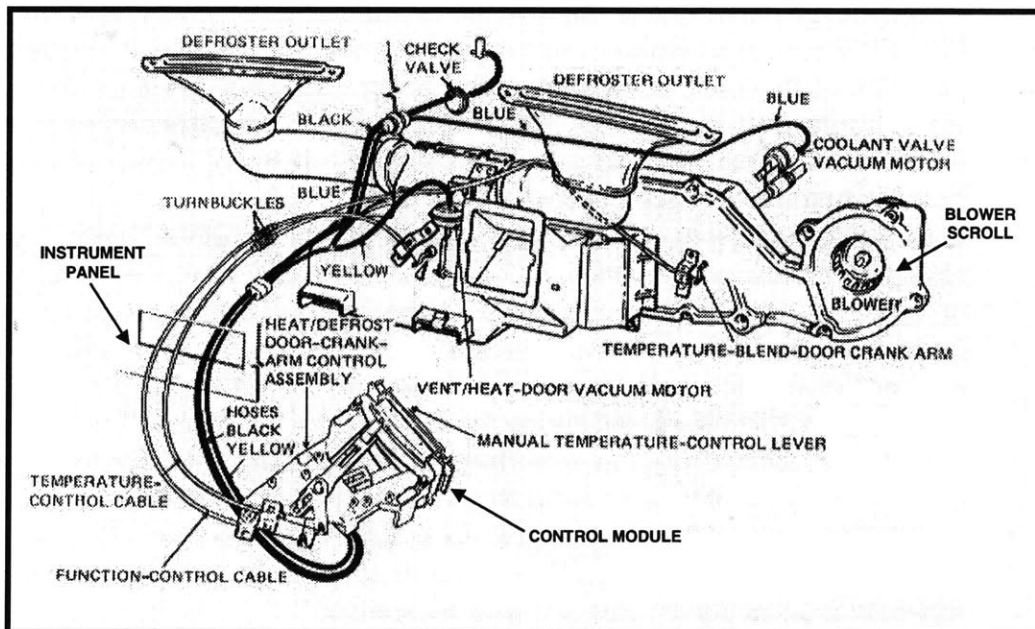
<sup>66</sup> In fact, the casing is the one component that gives each unit a distinct demeanor. Designers of HVAC units are constantly reusing various components, which have standard dimensions and performance characteristics (e.g., heater cores, evaporator cores and blower motors), but are rarely successful reusing casing designs (and for that matter the doors that go into the various casings). As a result, each casing design has to be modified for every new car model. For the most part, this is due to packaging space; a slight change in an existing package space can significantly affect the design of the casing.

## 6.3 Background Information:

This section will make a few comments about two key aspects in the design of an air handling system—performance and packaging measures in particular—before discussing the results from the analysis. The following sub-sections will describe (1) what are the main functions of a typical HVAC unit and how do they relate to the design, and (2) how does external and internal package requirements limit designer flexibility.

### 6.3.1 General Characteristics and Components:

A typical HVAC unit, as shown in Figure 6.1, provides warm, cool and re-circulated air into the passenger compartment of a vehicle. The air enters the passenger compartment in one of three ways: through the floor ducts, through the registers, or through the defroster vents. The unit relies on outside air or cabin air to provide these functions to the vehicle occupants.


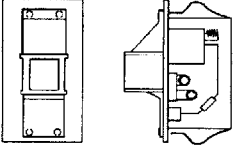

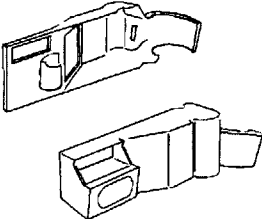
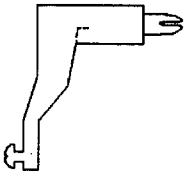



**Figure 6.1:** Typical HVAC Unit (Picture taken from The Auto Book, 3<sup>rd</sup> Ed. by W. Crouse and D. Anglin<sup>67</sup>)



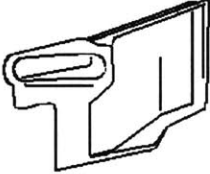


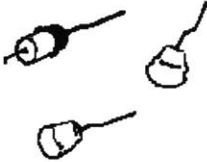
The user can control the rate of flow and the temperature of the air that enters the cabin through a control module found on the instrument panel. As Figure 6.1 shows, the control module actuates the air handling system through a series of cables and vacuum hoses. The cables drive the temperature-blend door while the vacuum hoses operate the vacuum motors, which initiate motion to the doors that they are attached to.

<sup>67</sup> Although this figure came from a book that was published in 1985, the fact is that HVAC designs have not changed dramatically in the past ten or more years. Many of the features shown in this figure and in Figure 6.2 have been retained and used in current designs.

Typically, HVAC units have the following components:

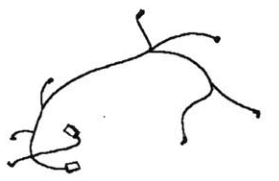
Blower Motor		<p>The blower motor includes a small electric fan that pushes cabin air or outside air through the unit; the speed of the fan is set by the user, with the help of the control module. The motor, itself, is cooled by the air that it pushes through the unit; a cooling tube is normally connected to both the motor and blower scroll, allowing some of the moving air to be redirected to the electric motor.</p>
Blower Motor Resistor		<p>The blower motor resistor makes use of “high resistance wire coils placed in the blower circuit to reduce the voltage drop across the blower motor.”<sup>68</sup></p>
Blower Scroll Assembly (including Air Collimator)		<p>The blower scroll assembly funnels air into and out of the blower motor. The shape of the scroll allows the circulating air to move to the evaporator chamber with very little turbulence.</p>
Casing		<p>The casing directs air through the unit to be heated, cooled or re-circulated; reduces unnecessary pressure drops and water leaks; and locates most components during assembly.</p>
Door Crank		<p>The door cranks convert translation from the vacuum motors to rotation for the doors.</p>
Electric Actuator		<p>The electric actuator controls the position of the temperature-blend door. (This component has for some time replaced the temperature-control cable that is shown in Figure 6.1.)</p>

<sup>68</sup> Component Design Guide, HVAC System. ©Ford Motor Co., 1992

Evaporator Core		The evaporator core, with its radiator-like construction, consumes heat and moisture from the passing refrigerant flow. It works in conjunction with a compressor, condenser, receiver, high-pressure inlet, and low-pressure outlet, completing the entire refrigeration cycle.
Heater Core		The heater core is a small, radiator-like component that is attached to the engine cooling system by two hoses. Hot coolant flows through the hoses and the heater core, transmitting heat from the heater core to the passing air of the HVAC system.
Heater Core Cover		The heater core cover allows the heater to be accessible, if service is needed. <sup>69</sup>
Temperature-Blend Door		The temperature-blend door allows hot and ambient air to be mixed together. The ratio of hot-to-ambient air is dependent on the position of the door. (Note: other types of doors and their functions will be discussed in the next subsection.)
Thermal Blanket		The thermal blanket is used to reduce thermal energy loss, as hot or cool running refrigerant makes its way to the designated heat exchanger—the heater core or evaporator core.
Vacuum Motor		The vacuum motors actuate many of the doors found in an HVAC unit. Vacuum is supplied to the vacuum motors from the engine intake manifold when the engine is running.

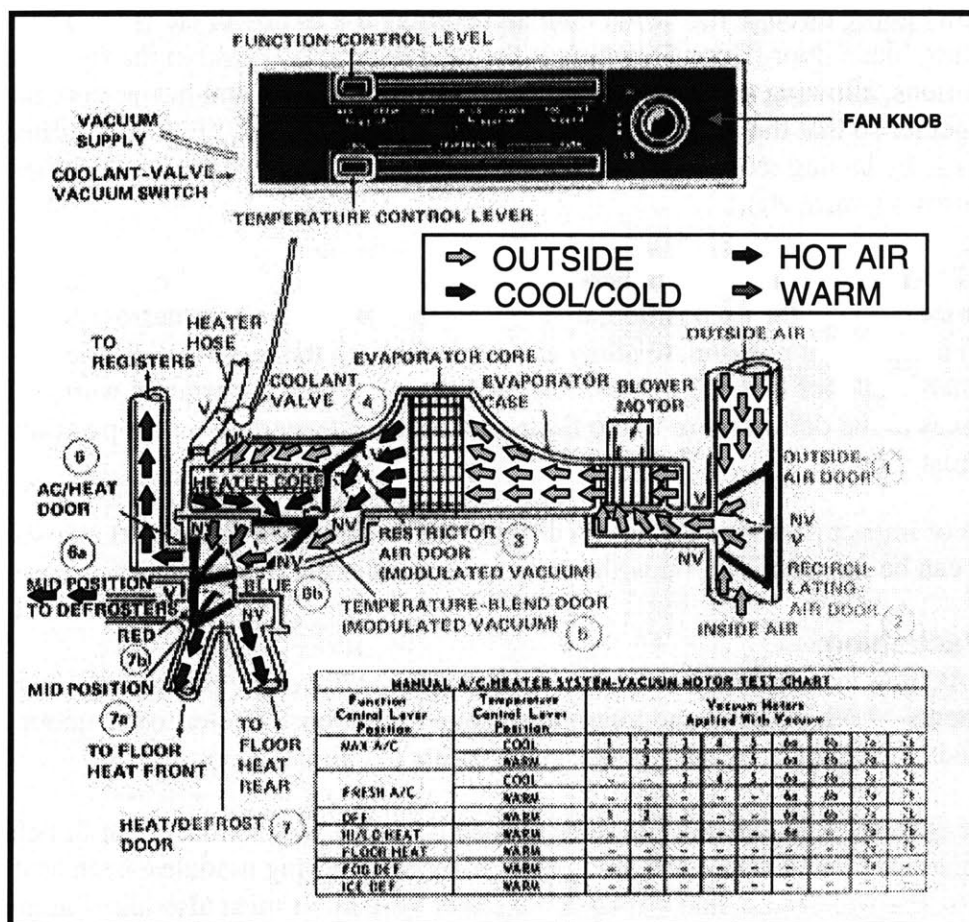
<sup>69</sup> Heater cores have a product life of about 5 or 6 years, which is somewhat short with respect to the life of a typical car. As a result, most units have a heater core cover so that a faulty heater core can be easily replaced with a new one—all without having to remove the entire unit from the vehicle. Because of where the unit is positioned in the car, behind the instrument panel, serviceability is a very important issue when designing an HVAC unit. (As a note, evaporator cores are usually sealed within the HVAC's main housing for two reasons: (1) they have a longer product life than heater cores and (2) having a removable cover for the evaporator core may allow condensation to enter the passenger compartment.)



Wire Harness (Vacuum Hoses and Cables)		The wire harness connects the control module to different components on the unit so that different levels of heating, venting, air and cooling can be set and regulated by the user.
---	---	--

### 6.3.2 Performance Characteristics:

Figure 6.2 shows a schematic of how air enters, navigates through, and exits the system; the figure also shows how air is heated, cooled, and brought to a desired temperature level. Arrows were added to the figure to allow the reader to better visualize the air-handling process, throughout the entire system. Each arrow is marked with one of four colors to illustrate where hot (red), warm (orange) and cold (blue) air is produced. Yellow-filled arrows represent airflow at an ambient temperature condition.



**Figure 6.2:** Schematic of an HVAC Unit, Showing Airflow through System (Picture taken from *The Auto Book*, 3<sup>rd</sup> Ed. by W. Crouse and D. Anglin)

During operation, the unit draws in air from either inside or outside the vehicle. As the figure indicates, outside air enters the system when the outside air door (Door 1) is opened and the re-circulating door (Door 2) is closed, and cabin air enters the system when the two doors are in the opposite position. The position of both doors is controlled by the function-control level, which is illustrated at the top of the figure.

The blower motor pushes the entering air through the evaporator case into the evaporator core, at a speed that is determined by the fan knob. (See the top of Figure 6.2.) The air goes through the evaporator core, regardless of whether or not the air is intended to be cooled. (In the case of this figure, the air is shown being cooled; the arrows change colors, from yellow to blue, as the air enters and leaves the evaporator core.)

After passing through the evaporator core, the air encounters the restrictor air door (Door 3), where it is either directed to the heater core or bypassed. (The figure shows ambient air (yellow) going through the heater core and coming out as hot air (red).)<sup>70</sup> The temperature-blend door (Door 5), which is the next door to be found in the figure, can vary positions, allowing a certain amount of evaporator-core air and heater-core air to blend together so that the desired degree of warm air is produced. (Figure 6.2 illustrates this process by having red arrows (heated air) and yellow arrows (ambient air) yield orange arrows (warm air).)

The air then meets the air-conditioner-heat door (Door 6), which can be situated in one of two positions: (1) in the up position, to allow air to move to the heat-defrost door (Door 7), or (2) in the down position, to allow air to exit through the registers. If the air-conditioner-heat door is in the up position, then the air, which is normally warmed in this case, moves to the defrosters or to the floor-heat outlets, depending on the position of the heat-defrost door.

The type of impact performance has on design will be discussed in the next sub-section, where it can be fully understood with respect to several other design issues.

### 6.3.3 Packaging:

The HVAC unit as mentioned in the previous chapter is driven by packaging requirements—both external and internal package limitations. (Refer to the quotations mentioned in Chapter V by designers and engineers in Climate Control.)

From the external side, the unit has to be shaped in such a way so that it can fit behind the instrument panel and console and along side many neighboring modules, such as the glove box, fire wall, radio, and airbag deployment system. It must also allow ample floor

---

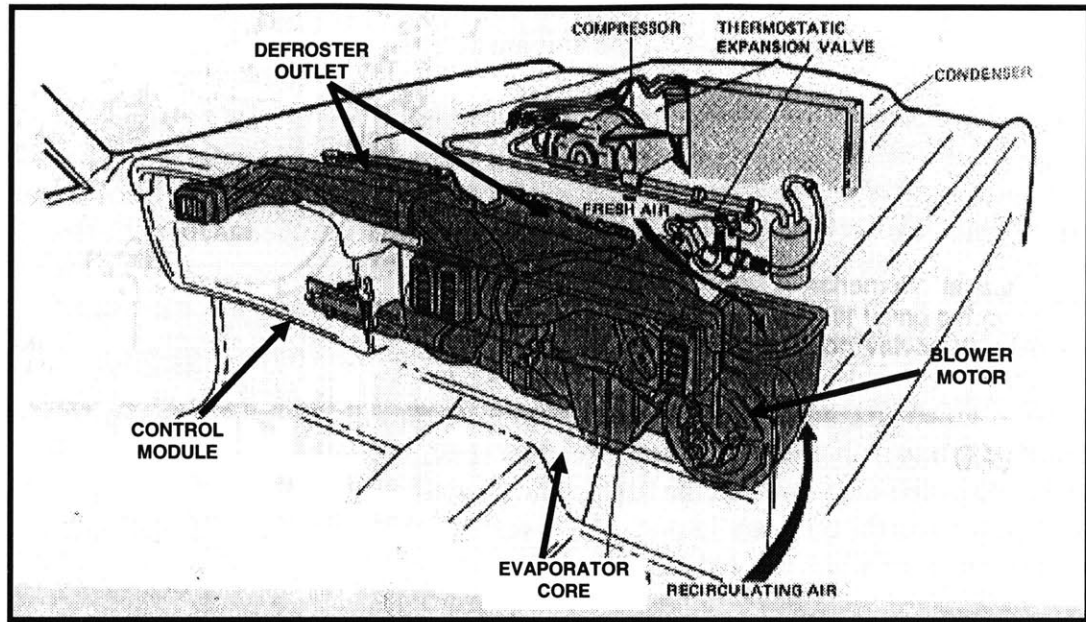
<sup>70</sup> While reviewing this portion of the figure, the reader may wonder why the arrows entering the heater core are not blue, corresponding to the flow of air exiting the evaporator core. For the most part, HVAC units are designed so that either the evaporator core or heater core is in operation, but never both. If the desired effect is warm/hot air, then ambient air would enter the heater core, as is shown Figure 6.2. (Note: When the air conditioner is running in a vehicle, the coolant valve (4) automatically closes off the flow of hot coolant from the engine cooling system.)

space for the front passenger. Figure 6.3 shows the location of a typical HVAC unit. Some of noteworthy components from the unit are also identified.

In addition, the location/position of several components are determined by the relative location/position of various other components; this is how certain performance measures are instigated at the level of design. As one would see when opening up an HVAC unit, the components are not tightly nestled together; placing each component inside one of these units is as much a packaging exercise as it is a performance exercise. Because the components are moderately spread out inside the unit for the reasons just mentioned, the unit, itself, becomes rather large.

For those components, which nestle themselves inside the unit, a certain amount of space is needed. Certain components, in fact, limit how small the unit can actually be. For example, the evaporator core and the basin, which captures and collects the condensation that is produced by the core, are important measures in the design of an HVAC and normally dictate the width and height of the unit.

The desired airflow performance will also drive the size of the unit, making it perhaps bigger than the given package space. Direct performance can be measured by how quickly the air in a cabin can be cooled, heated or refreshed. This all depends on the rate at which air is discharged from the HVAC unit and emitted into the cabin. Performance has a direct impact on the design of the unit—in particular, the size of the blower motor and heat exchangers (assuming size has a direct impact on performance) and the cross-sectional area of the airflow canal. Increasing the size of any one of these pieces of the design will undoubtedly increase overall performance. (However, one has to consider the fact that many of the parameters in the design of an HVAC are tightly coupled with other parameters. For example, if the airflow canal is enlarged so that more air can pass through it at any one given time, then many of the doors that attach to the casing will need to be enlarged, as well.)



**Figure 6.3:** Typical Location of an HVAC Unit inside a Vehicle (Picture taken from The Auto Book, 3<sup>rd</sup> Ed. by W. Crouse and D. Anglin)

The casing, doors, and evaporator core tubes are designed and redesigned to accommodate each unique package space. In most cases, the package space that is allotted to an HVAC unit is different with each vehicle. (Package space is also unique in left- and right-hand drive vehicles. The package space in a left-hand drive vehicle may not always be the mirror image of the package space in the alternative right-hand drive vehicle.) As a result, Visteon designs, manufactures and assembles over 40 individual HVAC units for Ford cars and trucks. Most of the internal and external components are carried over from a previous model and/or are shared with existing models, but the casing and other components, which are considered relatively inexpensive to redesign, are unique to every packaging situation.

#### 6.4 Different HVAC Architectures:

In many cases, HVAC units are categorized by architecture type. The type of HVAC unit used in an automobile strongly depends on the packaging space allotted to the unit. Currently, there are three main architectures considered in the design of an HVAC unit—the handed system, the semi-center mount system, and the split system.

- **Handed System:**  
In a handed system, the main components—the blower motor and heat exchangers—are spread out from right to left in the vehicle, much like the way it is shown in Figure 6.3. The air therefore flows through the unit in a straight manner, passing through the evaporator core, the heater core (if desired), and one of three designated

routes to the passenger compartment. The handed system is very sensitive to different packaging needs; very seldom can this type of design be reused from vehicle to vehicle.

- **Semi-Center Mount System:**

The semi-center mount system distinguishes itself from the other systems by having the heat exchangers and doors stacked vertically in the main section of the unit. The blower motor is adjoined to the main section from either the left or right side, depending on the specified package condition. (This system tries to be more applicable to both left-hand and right-hand vehicles, by taking advantage of its semi-symmetric architecture.) The air flows through the system and out to the ducts, in an S-shaped pattern, rather than straight across the unit as is shown in Figure 6.2.

Note: The semi-center mount system stems from the center mount system—a system that tries to take full advantage of symmetry and reusability in both left-hand and right-hand drive vehicles. In the center mount system, the blower motor is placed in the middle of the unit, making the system completely symmetric.

As was mentioned to the author by a designer in Climate Control, center mount units are rarely used in U.S. vehicle models. Here is his reason:

“A true center mount system will go in any vehicle, okay? But typically we don’t do a true center mount in the States. They do some in Europe, but we don’t do them here because the vehicles are much larger here, and they require more airflow, more capacity, more A/C ability, performance-wise, than they do in the smaller European type cars. And...in order to get that capacity, the package grows so much that the fore-aft dimension and also the up-down, the vertical dimension gets so great that you can’t accommodate it in the center stack area of the instrument panel. There’s not enough real estate, okay? In the European or Japanese communities, their systems are smaller, they can...they’re more compact, they can keep the instrument panel at a smaller dimension...the fore-aft and the vertical...they can keep them down to a reasonable number.”

- **Split System:**

The split system takes advantage of some of the engine’s compartment space. In this type of system, the evaporator core and air inlet compartment are situated just behind the fire wall in the engine compartment, while the rest of the unit is positioned behind the passenger compartment, behind the instrument panel. (Because of the intervention of the firewall, however, the system is divided into two or more items, with connecting portals to allow air to pass. This makes final assembly rather difficult, especially when connecting the joints to each section of the unit.) Not only does this design reduce the need for a significantly large package space behind the instrument panel; it also reduces the level of inlet noise heard by the occupants and eliminates the possibility of condensation leaking into the passenger compartment.

However, Ford is now moving away from this system; the designer, who made the comment earlier about center mount systems, had this to say:

“...they’re trying to get away from that...they’re trying to minimize the model package space in the engine compartment area. That’s...just by design, they’re trying to change the way the engine compartment layout is and the types of windshields, and the overall design of the vehicle. They want to minimize the amount of items that are put out in the engine compartment.”

Each of these architectures, as was alluded to in the definitions above, is driven primarily by package space. When beginning a new project, designers select one of these architectures based on the package space that they are given. Modifications are made to the selected architecture during the project, as the package space for each vehicle sub-systems becomes more defined.

The point that makes all of this relevant to assembly is that each architecture type can have an impact on how the mating planes are situated along the casing. (Of course, the architecture is not the only attribute in design that drives the position and location of the mating lines. As was mentioned in Section 4.5, with the example of the CDW-27 model, cost, performance, fixed manufacturing and assembly methods, and computer resources, to name a few can also dictate the segmentation of the casing. The architecture helps to define the shape of the product, and with that, its partition into various pieces.) The mating planes, in turn, affect how the doors, heat exchangers, blower motor, and many other components are assembled to the unit. The advantages/disadvantages of some of these architecture types, from the perspective of assembly, will be discussed in the next section.

## 6.5 Results from the Analysis:

As mentioned earlier, the author took apart and reassembled four different HVAC units for the purpose of analyzing each unit from the standpoint of assembly. An exploded view drawing of each unit is shown on the next few pages. The reader may want to refer these pages, in addition to Table 6.1 (Bill of Materials), when reading the comments, which the author makes. The table below identifies many, but not all, of the parts shown in the drawings. Fasteners, clips and links are not listed in the table.

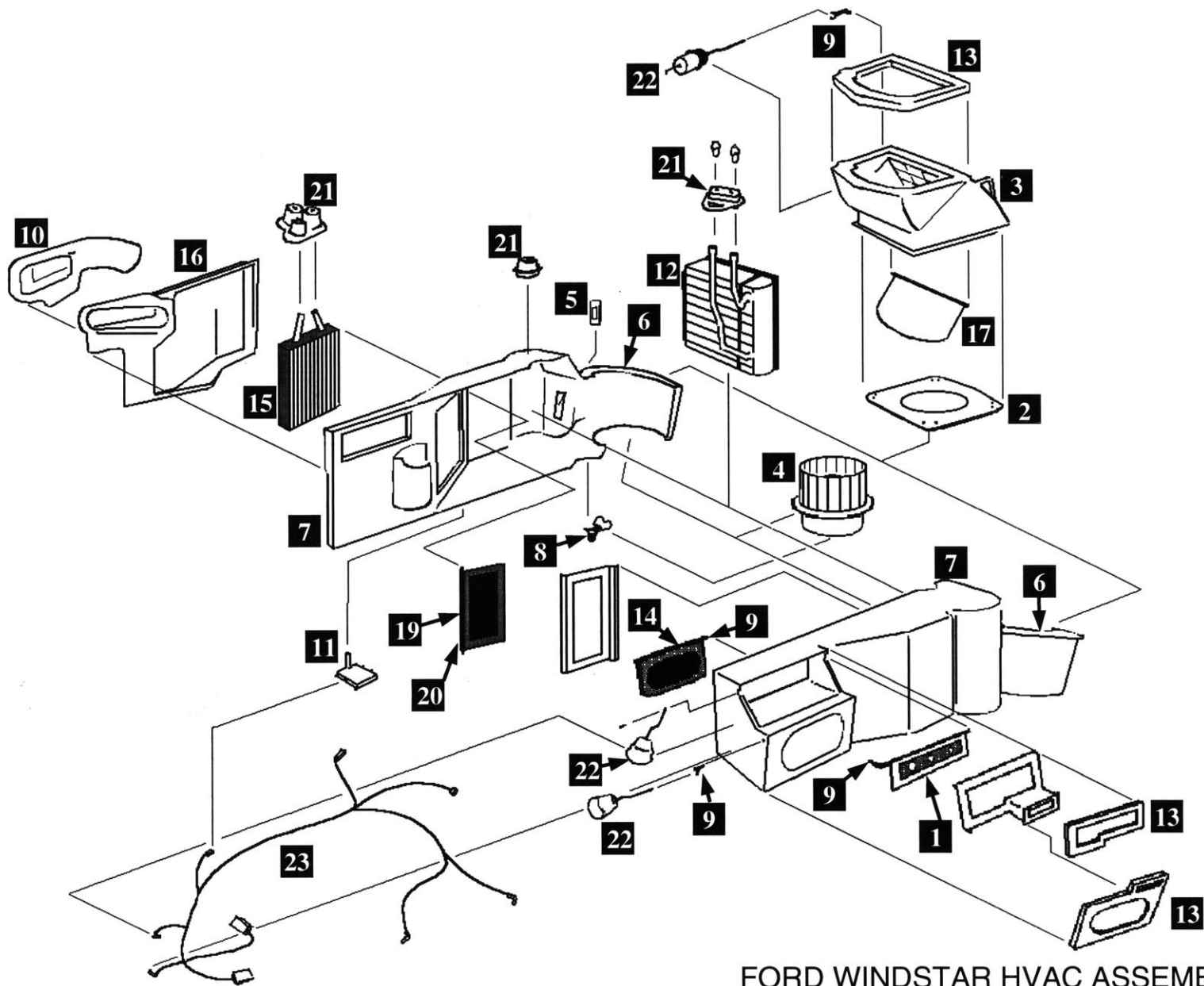
PART #	PART	QUANTITY			
		WINDSTAR	TAURUS	CAMRY	CDW-27
1	A/C Heat Door	1	N/S	1 (I/W #14)	1 (I/W #14)
2	Air Collimator	1	1 (I/W #6)	1 (I/W #6)	1 (I/W #6)
3	Air-Inlet Compartment	1	2	1	N/S
4	Blower Motor	1	1	1	1
5	Blower Motor Resistor	1	1	1	1
6	Blower Motor Scroll	2 (I/W #7)	2 (I/W #7)	2	2 (I/W #7)
7	Casing	2	2	2	4+
8	Cooling Hose	1	1	1	N/S
9	Door Crank	3	3+	4	2+
10	Duct	1	1	1	N/S
11	Electric Actuator	1	1	3	1
12	Evaporator Core	1	1	1	N/S
13	Foam Cushion	3	1	4	N/S
14	Heat-Defroster Door	1	N/S	1	1
15	Heater Core	1	1	1	1
16	Heater Core Cover	1	1	0	0
17	Outside-Air Door	1	2	1	N/S
18	Receptacle for Condensation	0	0	1	N/S
19	Restrictor-Air Door	1 (I/W #20)	1 (I/W #20)	1	1
20	Temperature-Blend Door	1	2	1	1
21	Thermal Blanket	3	3	1	1+
22	Vacuum Motor	3	2	0	1+
23	Wire Harness	1	1	N/S	N/S

N/S = Part is not shown in drawing

I/W (Part #) = Part is integrated with another part, which is identified by the part number

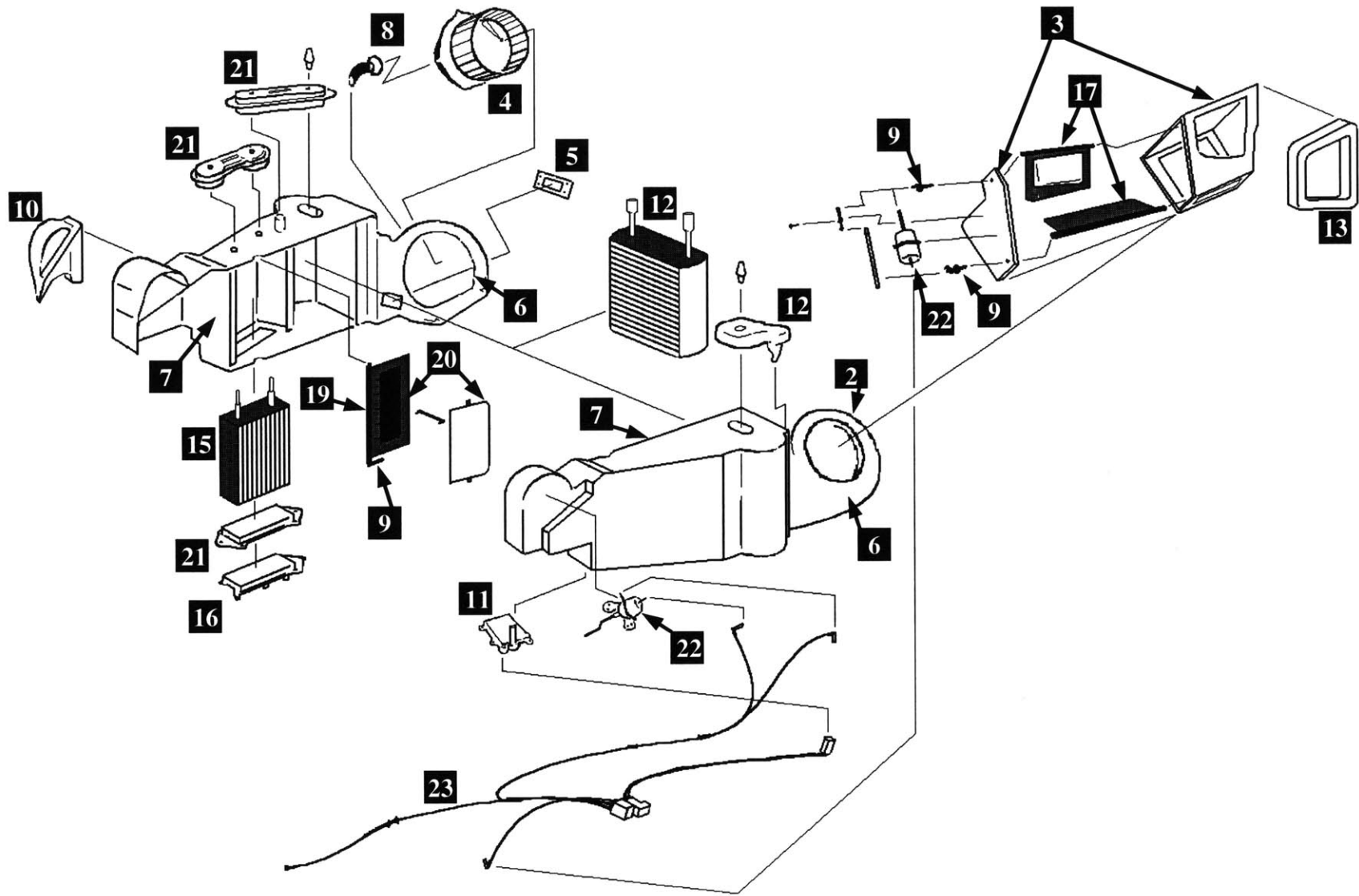
+ = There could be more of the same part type in the actual product, but they are not shown in the drawing

**Table 6.1: Bill of Materials for Each of the Four HVAC Units**

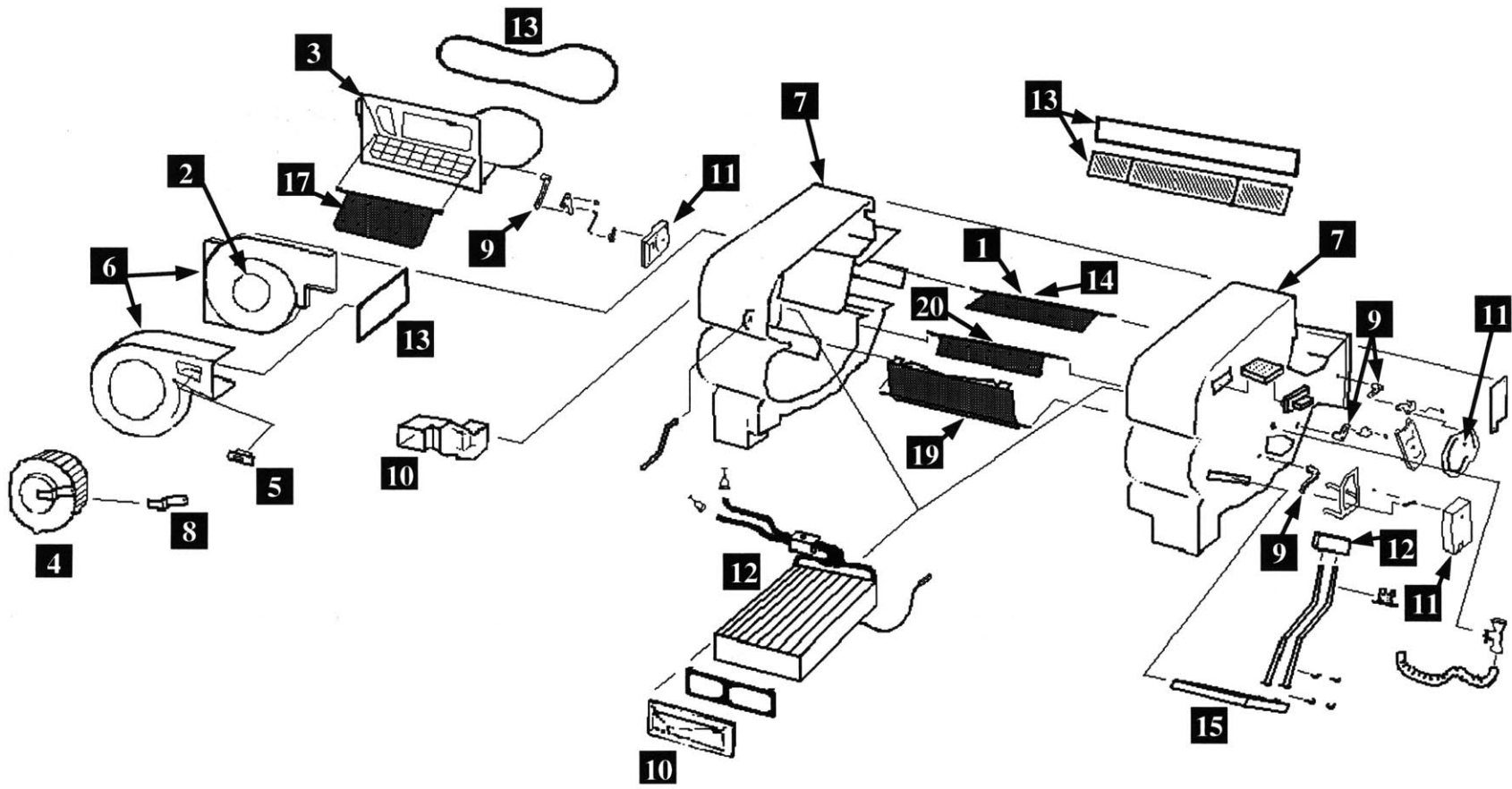


FORD WINDSTAR HVAC ASSEMBLY



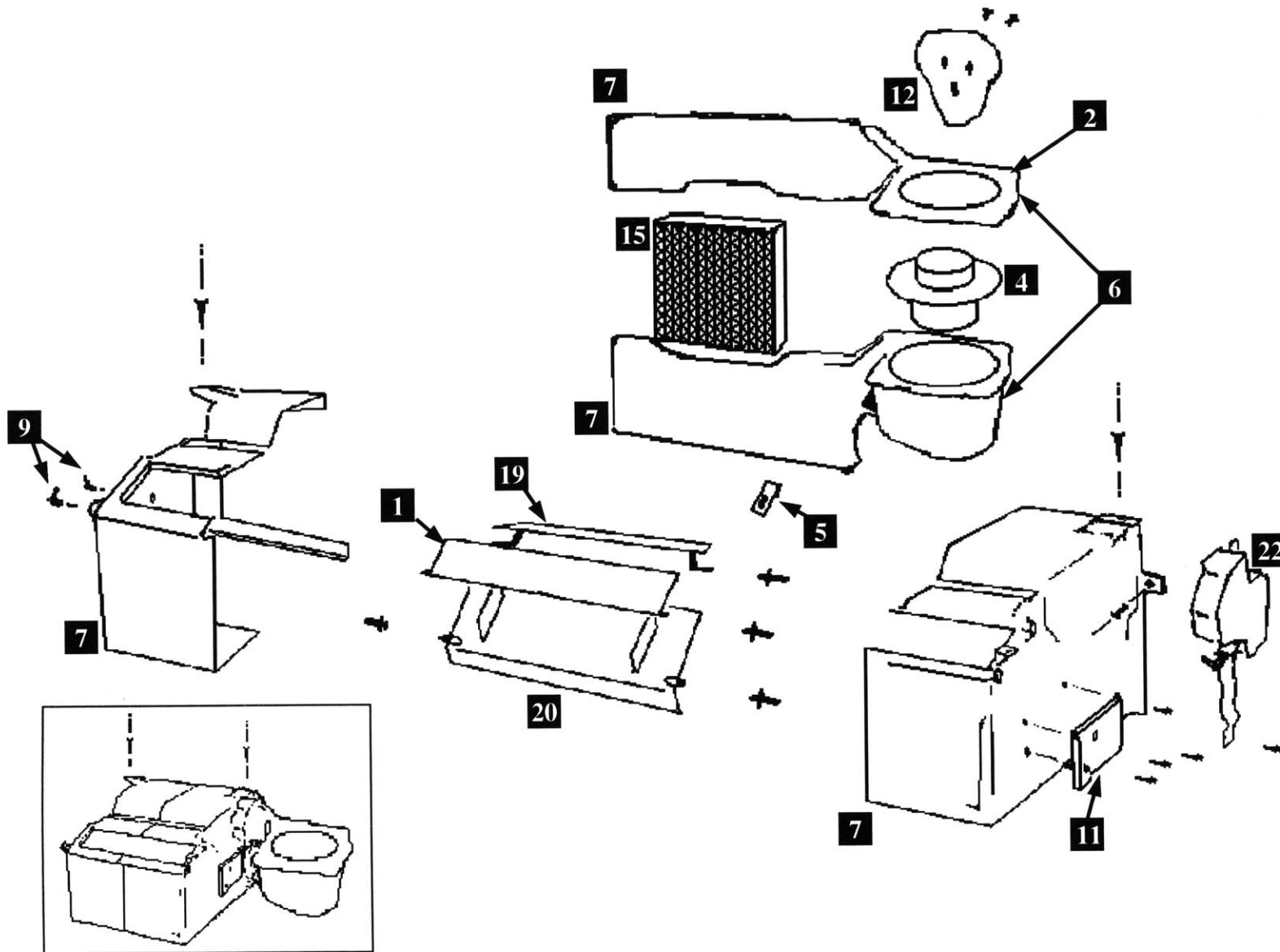


FORD TAURUS HVAC ASSEMBLY



122

TOYOTA CAMRY HVAC ASSEMBLY



FORD CDW-27 HVAC ASSEMBLY

To begin, the author has identified some of ways in which the four units can be characterized and has placed them in following grid (Table 6.2). The most interesting point that one can take from this figure is how differently Ford (Visteon) and Toyota (Denso) approach design for assembly in the case of HVAC systems: Ford emphasizes snap fits while Toyota emphasizes threaded fasteners. (Another interesting note to add is that Toyota uses fewer screws and snaps per part than Ford.)

	<b>FORD WINDSTAR</b>	<b>FORD TAURUS</b>	<b>TOYOTA CAMRY</b>	<b>FORD CDW-27</b>
ARCHITECTURE TYPE	Handed System	Handed System	Semi-Center Mount System	Semi-Center Mount System
NUMBER OF COMPONENTS/ PARTS	33	32	54	21 <sup>71</sup>
NUMBER OF SCREWS <sup>72</sup>	20	24	36	N/A
NUMBER OF SNAP-FITS	23	23	11	N/A

N/A = Not Available

**Table 6.2:** Comparing Different Aspects of Each Unit

Both the Ford Windstar and Taurus units are handed systems, while the Toyota Camry and Ford CDW-27 units are semi-center mount systems. The similarities are evident in the exploded view drawings. (For an explanation about each architecture type, refer to Section 6.4.) Although these units share some common characteristics at the product-architecture level, each one carries a unique set of traits for assembly (e.g., the location and arrangement of mating lines, the assembly direction for various parts/components, the number of orientations that the unit goes through before assembly is complete, etc.) These matters will be discussed in more detail in the remaining part of this section.

### 6.5.1 Taurus versus Windstar:

In general, the Taurus unit is easier to assemble (and disassemble) than the Windstar unit. The following comments strictly come from looking at the products, seeing how they are assembled, and making comparative notes along the way. The author will examine each of the obvious differences, and the advantages and disadvantages of each feature.

<sup>71</sup> The author did not receive a complete Ford CDW-27 unit; as the reader may be able to realize, major components, such as the evaporator core, basin, wire harness, and some vacuum motors are not included in the picture. Nevertheless, the author found the casing design and split planes interesting, and will mention more about them later in the Chapter.

<sup>72</sup> Ford uses torx screws in many of their units, while Toyota appears to use Phillip-head screws. (In fact, Ford discourages the use of posi or Phillip-head screws in the assembly of an HVAC unit, as is mentioned in many of the Climate Control guidelines.)

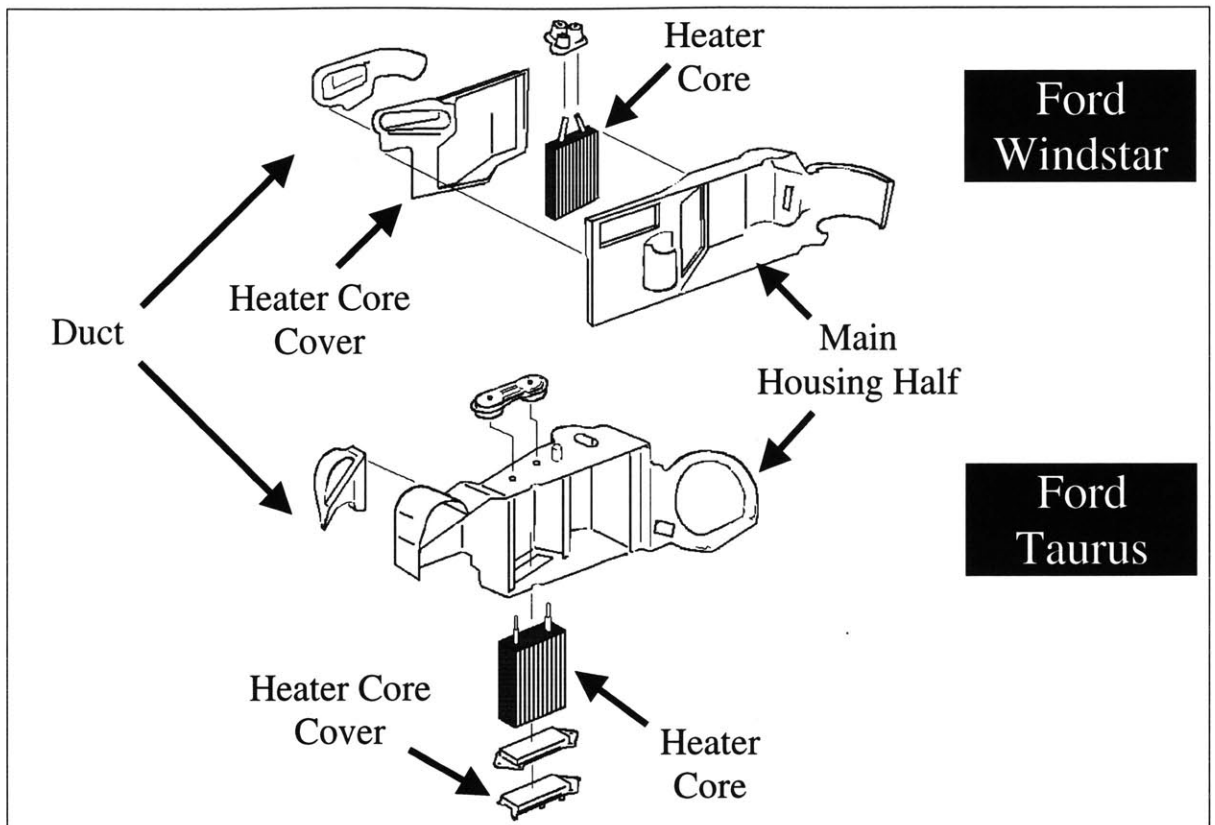
### Heater Core Cover Design:

Figure 6.4 helps to illustrate the points that will be made about the heater core cover designs for both the Windstar and Taurus units.

Windstar: The heater core cover for the Windstar encompasses the largest side of the heater core. The cover attaches directly to one half of the main housing with a few screws. The cover's purpose is to hold the heater core in place and also to allow it to be accessible for service. A duct is then attached to the heater core cover to allow hot air to enter the passenger cabin.

Taurus: The main-housing half holds and covers a majority of the heater core. The heater core is inserted into the main housing from the bottom (as Figure 6.4 illustrates). A thermal blanket and the heater core cover, respectively, are attached to the main-housing half, covering the remaining exposed area of the heater core. The heater core cover, in this case, is rather small, relative to the heater core cover used for the Windstar—making it more time and cost efficient from both a manufacturing and an assembly standpoint.

The way in which the heater core is mounted to the casing in the Taurus model allows it to sit more securely than in the Windstar model. The heater core, in the Taurus design, does not fall out of position after the casing is rotated in any particular way; the same cannot be said for the Windstar design. In addition, the duct is attached in a separate place from the heater core cover, allowing the heater core to be serviced in less time.



**Figure 6.4:** Heater Core Cover Designs for the Ford Windstar and the Ford Taurus

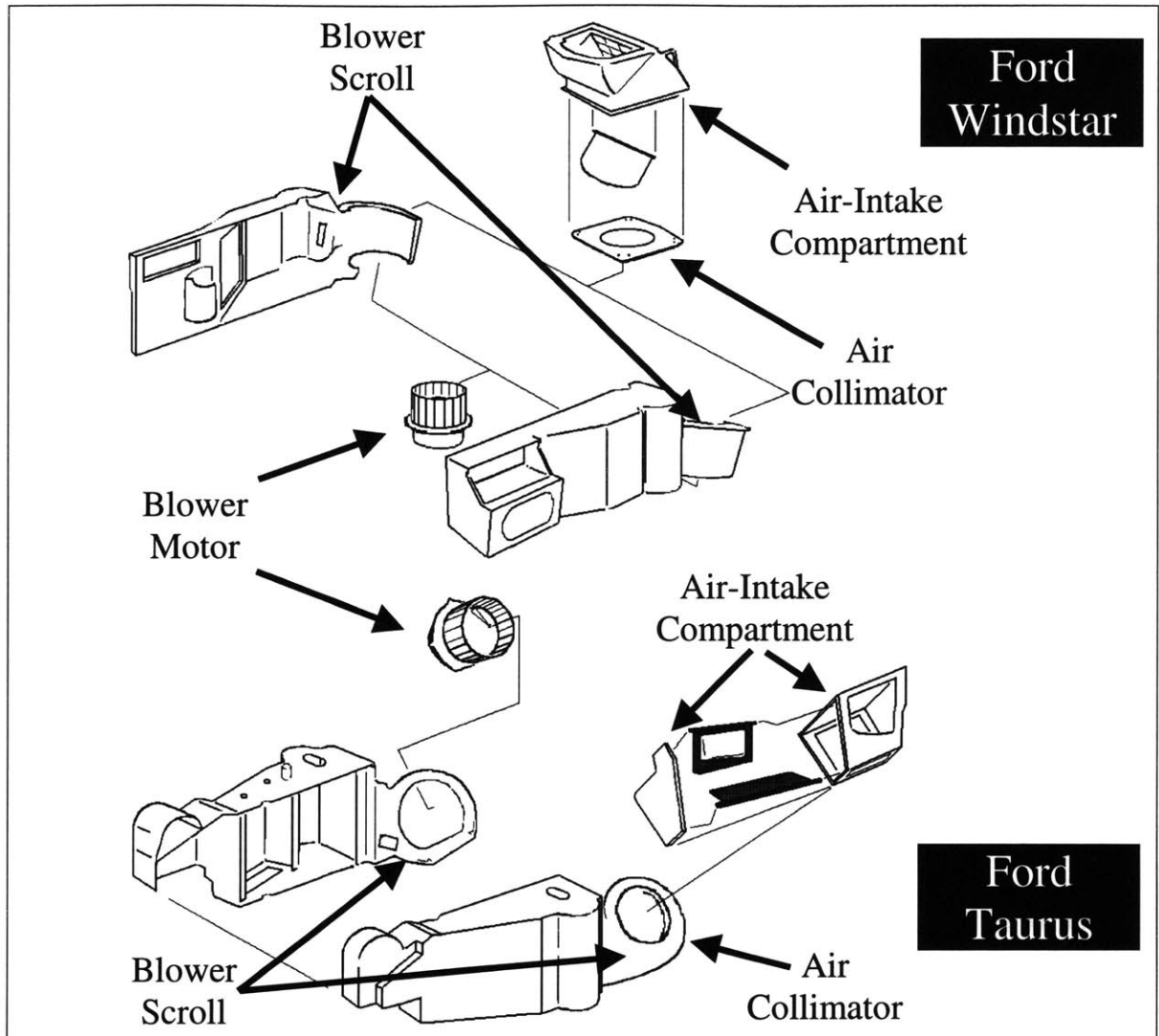
**Blower Motor Scroll:**

Figure 6.5 shows an exploded view of the blower motor compartment for both the Windstar and Taurus units. From a first glance at the two units, one can see that the Windstar has all of the shown parts (with the exception of the door) jointly assembled to the two casing halves, while the Taurus has none. Such complexities, as seen in the Windstar, may give rise to inconsistency during part mating or instability during handling.

Windstar: The blower motor in the Windstar unit is attached to the main housing only after both housing halves are assembled.

Taurus: The blower motor in the Taurus model is attached to only one half of the housing, allowing for more stability during and after assembly and alleviating the concern for buckling or improper mating of bendable parts, such as the casing, air collimator, and air-intake compartment. (The fully assembled Windstar unit, which the author received, showed the air collimator lifting away from the blower scroll. With the Taurus model, such problems are avoided because assembly is simplified to a set of

simple part-to-part attachments, rather than a nest of three or more parts that simultaneously locate each other.)



**Figure 6.5:** Blower Motor Compartment for the Ford Taurus and the Ford Windstar Units

The Windstar design is rather poor for the following reasons:

- Both parts of the housing need to be properly mated in order to locate the blower motor correctly.
- Any large-scale inconsistencies in the manufacturing of the casing halves can result in poor assembly and quite possibly inadequate heating and cooling performance. (For example, air and/or water leaks can propagate where buckling exists, as was the case in the Windstar unit the author received.)

- With the two halves mated together before the blower motor is attached, assembly becomes rather awkward and slow. (The unit at this point is already large and bulky.)

Assembly is easier in the Taurus design because the blower motor is attached to just one half of the housing. Orientation and handling is not as arduous because, for the time being, only half the casing is used. In addition, the Taurus design can absorb variation a lot better—the blower motor is located by and attached to just one part.

**Air Collimator:**

Refer to Figure 6.5 again to understand the comments made here.

Windstar: In the Windstar unit, a separate air collimator is attached to the top of the blower motor scroll. The air collimator is attached to the blower scroll in the same way that the blower motor is—the two casing halves simultaneously locate the part for attachment. And because of how it is attached, the same problems as mentioned before can arise.

Taurus: In the Taurus unit, the air collimator is actually integrated with the bottom right casing half—taking advantage of all the benefits of part integration.

**Note:** The author has made a number of comments about the poor design of the Ford Windstar unit; however, one must consider the reason(s) for these design decisions. In the case of this design, many of decisions reflected the assembly process that was used to join the sections of housing together.

The Windstar housing is joined together with a process called vibration welding. Vibration welding allows each housing to be sealed in a very short period of time, relative to the time that it takes to assemble it using a combination of snap-tabs and screws. To begin, a bead of plastic material is placed around the entire edge of one housing section. Next, the two sections are placed together, one on top of the other, inside a vibration-welding machine. The machine vibrates the two halves at a high frequency, melting and joining the plastic all around the mating surface. When the process is complete, the housing sections are tightly sealed to one another, making it virtually impossible to separate the two without destroying some part of the housing.

In order for the process to be effective, the mating surface must be straight. In other words, it must stay within a designated three-dimensional plane that most likely cuts the unit in half. Notice that the Windstar unit is split in this manner. Vibration welding requires straight mating surfaces for two reasons: (1) so that a strong seal can be made all along the welding region and (2) out of plane surfaces can disrupt the vibrating process (i.e., protruding surfaces and features can bang into one another, resulting in a poor weld and the destruction of the parts).

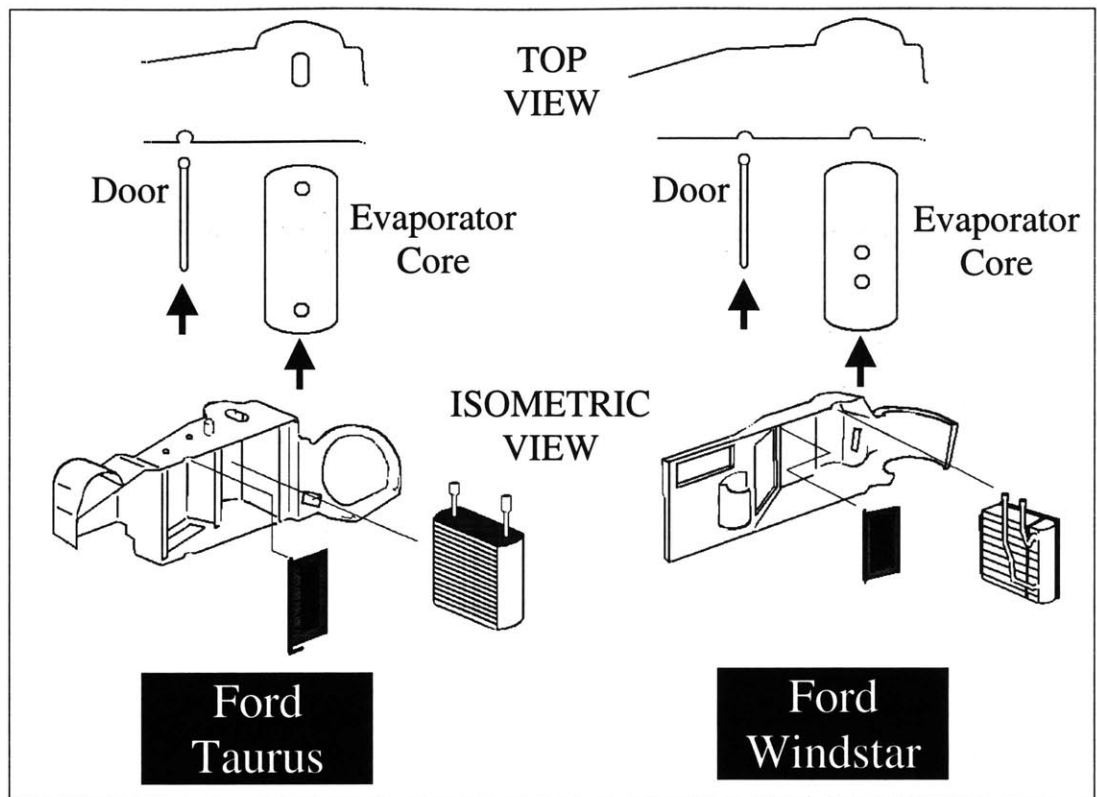


The Taurus, on the other hand, uses a combination of snap fits, screws and tongue-and-groove joints to assemble the casing halves together. Because the unit is not assembled using vibration welding, the designer can create a mating line that takes advantage of design for assembly. (Notice also that the Taurus unit is split in two directions, one that cuts through the evaporator casing and another that cuts through the blower motor scroll.)

#### Blend Door and Evaporator Core Assembly:

The Taurus design allows the evaporator core and the blend door to be relatively stable when attached to one of the housing sections. This allows the operator to prepare the assembly for the next operation without ever having to hold or reposition these parts. The blend door snaps into a deep notch found at the top and bottom of the mating line; the notch holds the door in place while still allowing the door to rotate along its axis. The evaporator core fits snugly into its own compartment; a hole at the top of the casing allows one of the two tubes to poke through, preventing the cooling unit from falling out. (Because of the location of the hole, the evaporator core has to be tilted slightly. Although this requires an extra operation, the whole task takes only a few seconds.)

The Windstar design, on the other hand, does not allow the each part to be secure after the same general operations are performed. Semi-circular holes at the seam seat the blend door and one of the two tubes on the evaporator core, but the holes do not allow either part to be fixed in place. Securing the two parts happens only after the second casing half is attached. The figure below helps to illustrate this; each unit is shown from both an isometric view and a top view.

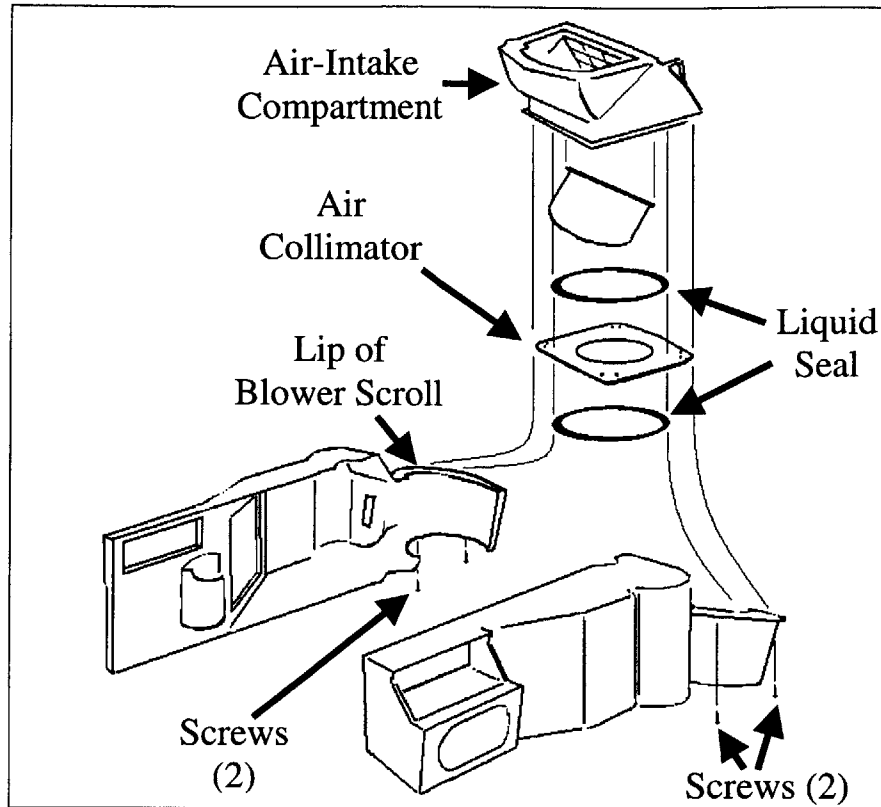


**Figure 6.6:** Attaching the Blend Door and the Evaporator Core to the HVAC Housing (in the Ford Taurus and Windstar Models)

### 6.5.2 Separate Comments about the Windstar, which Relate to the Comments Made in the Previous Sub-Section:

#### Air Intake Compartment:

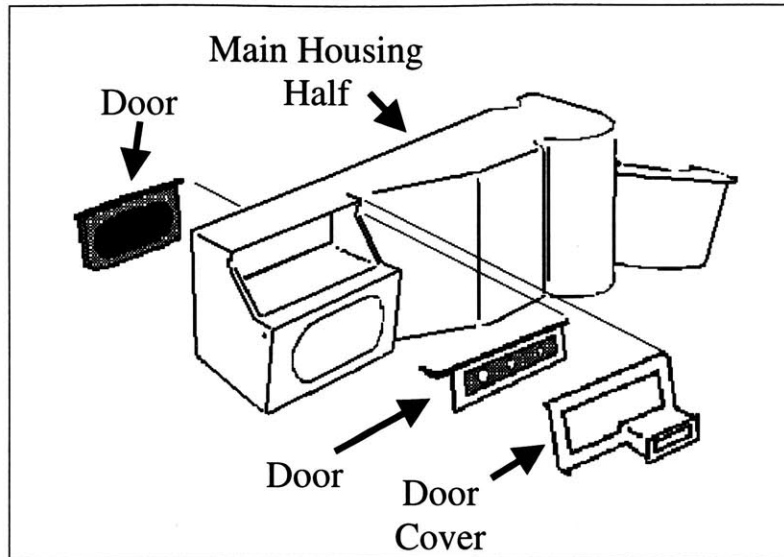
In the Windstar, both casing halves locate the air intake compartment. This is done by first mating the air collimator to the main assembly and then attaching the air intake compartment to the air collimator. However, both the parts are not firmly attached to the main casing, at this time; they are in fact loosely located to it, and to each other. The actual joining process happens when the unit is turned completely upside down and four screws are driven through the lip of the blower scroll, the air collimator and finally the air-intake compartment. This assembly technique violates one of the most crucial rules in design for assembly—try to avoid rotating an unstable sub-assembly in order to either attach additional parts to it or to drive in screws. Actually, during this part of the assembly process, a sealant is applied to the top of the blower scroll and to the top of the air collimator in order to temporarily adhere the sub-assembly to the main assembly, and to prepare the whole unit for rotation and final assembly. (The liquid seal also helps to prevent air leaks from propagating.) Figure 6.7 helps to illustrate some of the points made here.



**Figure 6.7:** Attaching the Air-Intake Compartment to the Windstar Housing

**Heat-Defrost Door Assembly:**

As Figure 6.7 shows, two doors are assembled to one casing half, but from opposite directions—one from the outside and one from the inside of the casing. For the inner door assembly, the worker has to deflect the casing slightly in order to fit the door into its final location. However, many of these assembly tasks occur when the plastic is still warm and quite malleable. For the outer door, assembly requires a two-step process: (1) rest the door over the door opening, and (2) place the door cover over the door and sonic weld it to the casing.



**Figure 6.8:** Door Assembly for Ford Windstar Unit

Based simply on the design, the author feels that it would be best to have both doors assemble from the outer direction. The existing door cover, of course, will have to be enlarged to encompass both doors. But this set-up allows both doors to be assembled quickly and easily without any intermediate step, such as orienting the casing to a new position, or bending the casing to accommodate either door.

### 6.5.3 Toyota Camry versus Ford Models:

The Camry design has many benefits over the Ford designs just mentioned; the author will discuss them below.

#### Few Assembly Directions:

One of the first things that an observer may pick up on when looking through the different exploded-view drawings is that the Camry has many of its parts assembled from one direction. For instance, most of the assembly activity occurs on the outer part of the one casing shown at the bottom right of the picture.

#### Few Attachment Points Needed for Wire Harness Assembly:

The most interesting thing about the Camry unit is that the wire harness requires very few assembly operations (handling, orienting, and attaching); the unit takes advantage of springs, links and cams to transfer mechanical energy from one or two electric actuators to four or more doors. As a result, a complicated wire harness, composed of different vacuum hoses, cables and wires, is reduced to just a few wires. (Ford, however, tries to avoid links; the Company sees this as extra parts that will add to manufacturing cost and assembly time.<sup>73</sup>)

<sup>73</sup> This information can be found in any one of the CCO design manuals written by Ford Motor Co.

The Camry idea eliminates the several orientations and attachment points. In most cases, wire harnesses require the most time for assembly; they require being attached to several parts on the unit at different locations. As a result, the unit needs to be oriented in a number of different positions so that the wire harness can be attached properly to the designated areas. Of course, at this time, the unit is almost completely assembled, making the wire harness assembly slow and cumbersome due to size and weight of the unit.

#### Integrated Cooling Hose for the Blower Motor:

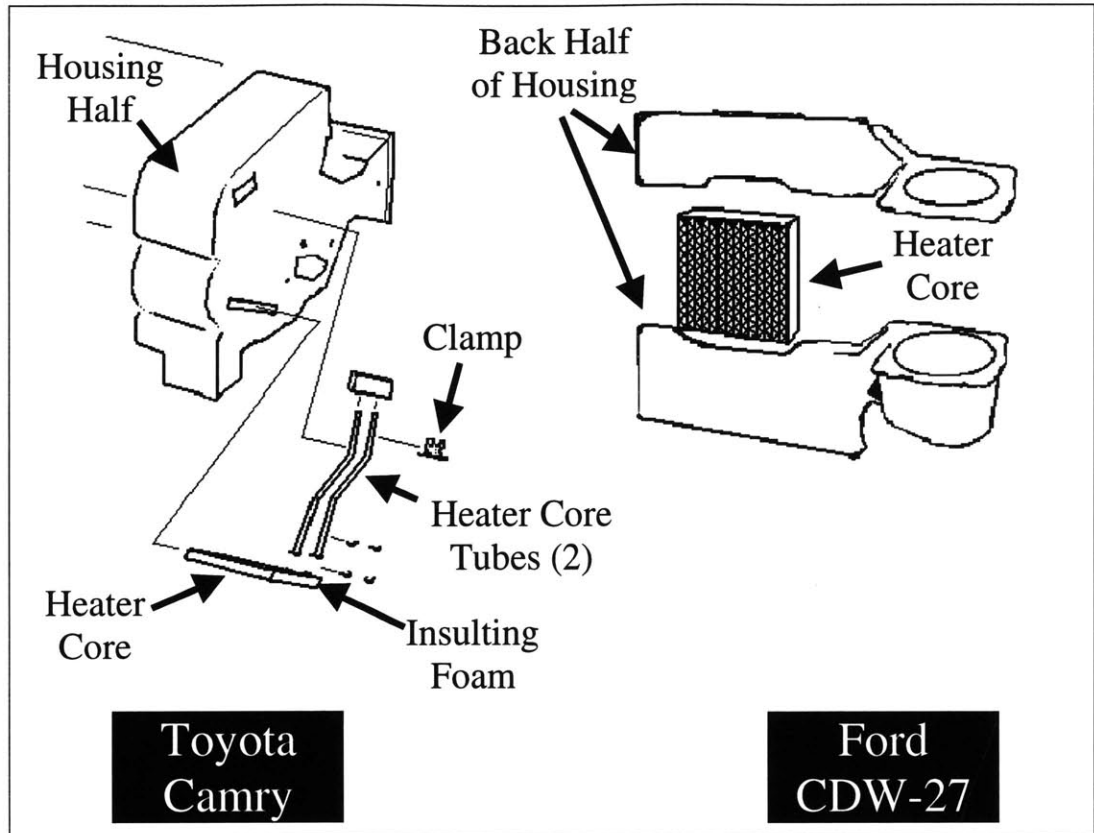
The blower motor on the Camry has a plastic cooling hose, which is combined with the blower motor to make assembly even easier. The plastic, integrated hose attaches to the blower motor at one end, while at the other end, the hose is aligned with an opening at the top of the blower motor scroll. When the blower motor is aligned with the blower motor scroll, so is the cooling hose to its respected attachment location. In both the Taurus and Windstar units, a separate rubber hose is attached to both the blower motor and to the blower scroll. Unfortunately, due to the hose's flexible structure, this assembly step takes some time.

#### Elimination of the Heater Core Cover:

Unlike the Taurus and Windstar units, the Camry does not have a heater core cover. The core goes into the main housing in rather the same way as the heater core in the Taurus model. A piece of foam is adhered to the tank of the heater core, and this provides the necessary insulation. The core is kept in place by the heater core tubes; the tubes are clamped to the side of the casing, removing the heater core's sixth and final degree of freedom.

On another note, the CDW-27 model does not have either a heater core cover or any type of portal for removing the heater core, if it fails. The heater core is attached to the two back sections of the casing. The back section is then attached to the front two casing sections with clips. (To prevent air leaks, the sections are attached using tongue-and-groove joints and corking.) In order to remove the heater core from the unit, the back part of the housing has to be completely removed. This means that the air-inlet compartment, blower motor, wire harness, and other items must also be removed. It is definitely clear that this unit was not designed to have the heater core easily serviced.

Figure 6.9 helps to illustrate the points that were made about the Camry and CDW-27 units.



**Figure 6.9:** Heater Core Assembly in the Toyota Camry and Ford CDW-27

**More Screws and Less Snap fits:**

The Camry unit has approximately twice as many screw fasteners as the Taurus or Windstar. (See Table 6.2) As a result, the Camry unit is relatively easy to disassemble and assemble without any destruction to any of the parts. This may be important to Toyota because it allows most, if not all, parts to be serviced. Ford has taken a different approach to design for serviceability. It has decided that only key components, those that have a short life span, will be made accessible; the rest will be sealed permanently inside the housing. Ford feels that it would cost vehicle owners less money to have the unit entirely replaced than have the existing unit fixed. (For parts that are not made accessible, the labor involved in removing the damaged part from the vehicle and from the unit is an intensive and costly project.) The reader may ask, why is it that Ford does not use more screws in its HVAC designs? For the most part, Ford believes in many of the traditional views of DFA: at all possible, screws should be replaced with snap-tabs in order to reduce assembly time and cost.

**Segregation of the Blower Motor Assembly:**

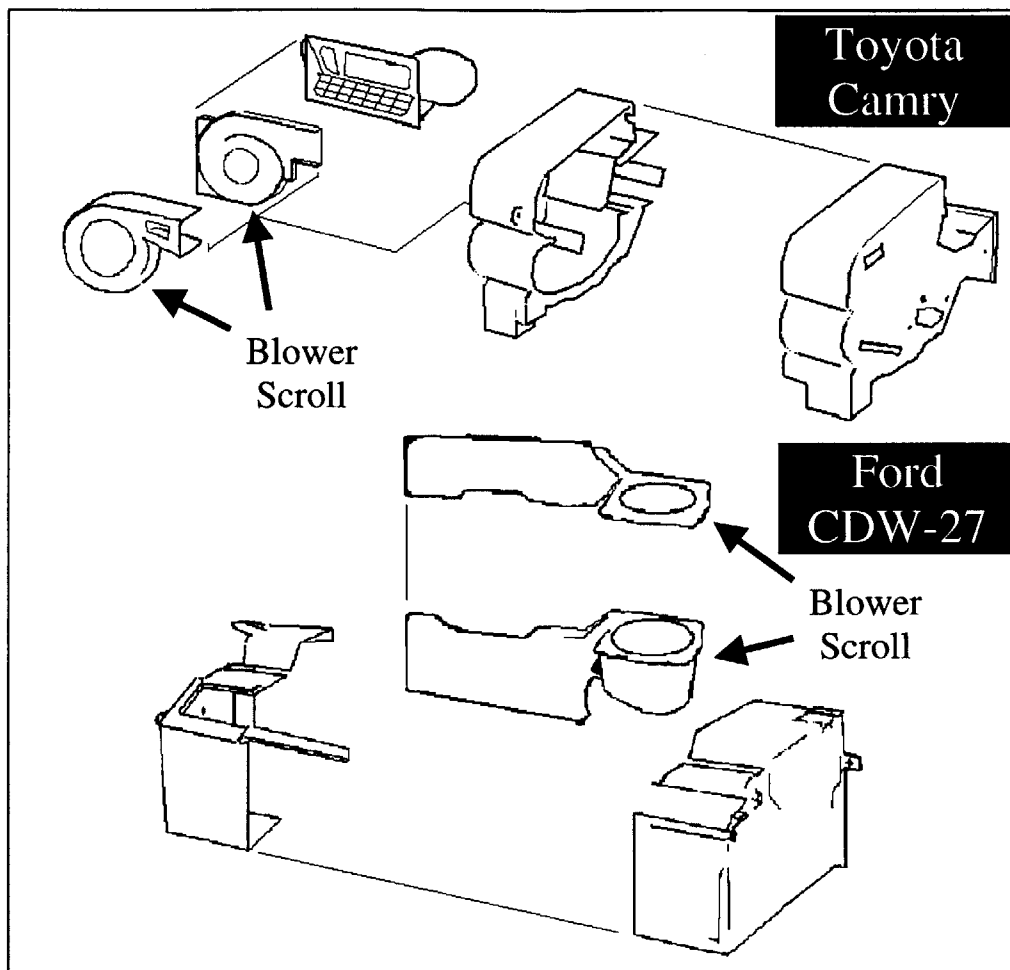
The Toyota Camry keeps the blower motor sub-assembly separate from the rest of the assembly. The figure below shows an exploded view of the different casing sections for

the Camry and the Ford CDW-27 units. Notice that in the CDW-27 model, the blower motor scroll is integrated with part of the main housing. (In actuality, the blower motor scroll is segmented into two parts, and each part is tied with a section of the main housing.) Having the mating plates oriented in this fashion, however, makes each part rather thin and long, making each part rather difficult to handle or set properly during assembly. In many cases, it is better to make each part somewhat box-like, thereby placing the center of gravity relatively close to the center.

In addition, segregating the blower motor assembly from the main assembly allows the following benefits to be reaped:

1. The main assembly does not become as large and heavy as before.
2. The blower motor sub-assembly can be made relatively quickly. (Orientation can be done without the main assembly, making assembly easier.)

The Toyota Camry unit captures all of these ideas. The last step in the assembly process is simply attaching the blower motor sub-assembly to the main assembly. This provides balance in two ways—handling of the individual assemblies and the sequence of the entire assembly.



**Figure 6.10:** Different Mating Plane Configurations

Based on what was said about the Camry and the CDW-27 units, it should be noted that part integration (and part dissociation) is an important issue to consider, but it should always be considered in the context of the actual assembly of a product. What will be the sequence of steps for assembling this product? Where will there be a need to orient the assembly for whatever reason? How big will the assembly grow at each stage of its assembly? How much will it weigh at each stage? These issues should be considered and weighed against the physical requirements placed on the units (e.g., pressure drops and water leaks) when designing split planes.

#### 6.5.4. Handed System versus Semi-Center Mount System:

Both the handed and semi-center mount designs have an equal number of advantages and disadvantages from the point of view of assembly. The handed system allows many of the doors to be located and in some cases fully assembled to one half of the main housing; the doors in the semi-center mount system rely on both halves of the housing to



make assembly possible. The reader should refer to the four units to see how the split planes are oriented and how the doors are assembled. This goes back to a comment made earlier: the architecture dictates the orientation of the split planes, and the split planes dictate the manner in which key components are assembled.

The semi-center mount system does provide some advantages of its own: it allows many components to be assembled in one direction (as was seen with the Toyota Camry) and it allows assembly to be done using small and well-balanced sections of casing.

## 6.6 Chapter Summary:

This chapter identified and discussed several of the essential characteristics that designers and engineers consider in the design of an HVAC system. In particular, it described how packaging and performance issues drive the type of architecture used in each application and how assembly is considered from there. The split planes, which are constrained by the physical characteristics of the architecture, determine the way many of the components are assembled to the unit. Depending on the orientation of the split plane, the handling of the casing sections and the insertion of many of the components for assembly can all be done rather easily. Therefore, more emphasis needs to be applied to the design of different HVAC architectures and their resulting split planes, in order for assemble-ability to really be mastered.

Many of the driving factors in the design of HVAC systems will be revisited in Chapter VIII, when the author discusses some of the challenges that Visteon's Knowledge-Based Engineering (KBE) Dept. faced as it developed an advisor for Climate Control designers. The reader should keep this chapter in mind when reading Chapter VIII.

In the next chapter, the author will discuss some of the challenges and problems that the Knowledge-Based Engineering Dept. confronts each time it takes on a different project and some of the problems that it may face in the future. The Chapter will identify some of the ways in which the Department is planning to overcome some of its current difficulties. In addition, the author adds a few comments about KBE's current strategy, stemming from the information provided in Chapter III, and what issues the Dept. should be aware of as it gains more customer acceptance and develops advisors that attack more system-level problems.

# CHAPTER VII

---

## KBE TRADEOFFS AND PITFALLS

### 7.1 Introduction:

This chapter is in a sense a continuation of Chapter III, Viston/KBE Methods. In Chapter III, the author discussed how the Knowledge-Based Engineering Dept. at Visteon develops customer-focused software advisors for Ford/Visteon designers and engineers. Chapter VII will reevaluate these ideas, but in a different way: this chapter will look at some of the challenges that KBE faces now and can potentially face in the future, by considering its existing strategy, philosophy and methods.

The chapter will be broken down into the following key areas:

- **Gaining Customer Acceptance:** How does KBE gain positive word of mouth within the Ford/Visteon community; and what are some of the negative implications from this process? These questions will be answered using some of the ideas from system-dynamics theory.
- **Time-Consuming Procedures:** What steps when generating KBE software are normally the most time-consuming and why? What is KBE planning to do to alleviate some of these time-related difficulties? What challenges will it face along the way?
- **Case Study:** Two advisors, the KBE-HVAC and KBE-wiper systems, will be compared and discussed. The author found these two advisors interesting because they each have a unique way of managing knowledge, based on the type of products and product architectures which they analyze. In other words, the case study shows how different product architecture types can influence the way that an advisor is developed.

The purpose of this chapter is to help KBE better understand some of the challenges that it faces at both the process and advisor levels. If KBE plans to help Ford and Visteon revolutionize the way product and process information is managed, it must find ways to

improve and, if necessary, reinvent its own development process. This chapter does not intend to provide KBE, or any of its readers, with detailed, step-by-step solutions to these problems and challenges. What the chapter actually intends to do is identify anticipated problem areas so that KBE can be alert to these issues, and their implications, as it moves on to bigger, more complicated projects—one of them being the KBE assembly advisor. Furthermore, the case studies prove useful—not just to KBE but to an advisor-development team wishing to create a process advisor. By identifying and comparing the many limiting factors that drive each product-specific advisory tool, a set of conditions can be developed for a process advisor, which encompasses most, if not all, products and product architectures.

## 7.2 KBE Boundary Conditions:

From KBE's point of view, developing an advisory tool requires an explicit understanding of the various factors, which in many cases limit the scope of the tool's functional development. Many of these ideas were mentioned in Chapter III. The author has provided a summary of those ideas below, which can be used as a reference for the rest of this chapter:

- **Time Limitation:** Most of the requests accepted by KBE usually take no longer than a year to complete. (Most customers will not wait any longer than that for a fully functional tool.)
- **Existing Software Tools:** KBE must be able to make tools that can complement existing software tools, which designers and engineers use across Visteon.
- **Existing Design/Development Processes:** KBE tools must conform to the existing procedures used by its end users.
- **Existing User Interface:** The user interface must complement that of I-DEAS, Pro-Engineer, or any other type of software system used by the customer.
- **Tangible Knowledge/Quantifiable Data:** Subjective information—information that cannot be represented in an advisor's knowledge base—does not suit many of the systems that KBE develops. (Refer to Chapter II, General KBE Theory and Methods, for more details.)
- **Customer Needs and Wants:** Each customer tends to have different needs and wants, resulting in a unique set of conditions for each project.
- **KBE Resources:** KBE is limited by the number of software developers and expert consultants on staff and by the number of licensed software tools at its disposal. (This issue is particularly true in the case of the developing assembly advisor.)
- **KBE Philosophy and Direction:** KBE wants to maintain good word of mouth. To do so, requires developing successful tools. This must be kept in mind when evaluating the feasibility of any request. In many cases, it means going for easy-to-handle projects first—the “low hanging fruit” theory—before tackling any of the larger, system-level projects.

### 7.3 The KBE Strategy: Gaining Positive Word of Mouth

As was mentioned in Chapter III, KBE emphasizes the development of “complete” software tools, from the very start of the project. But it also must consider doing this in a timely fashion. Most projects, which KBE considers, must require no more than a year to develop and deploy. (As one KBE manager explained it, most customers are reluctant to wait more than a year for one of these tailor-made tools. The development process is time consuming for both the KBE developers and the expert designers and engineers (i.e., the customers), who help on the projects. The customers provide critical information and feedback on the developing projects while still working on their own assignments. KBE has found out very quickly that projects have to fit the schedule of both the KBE developers and customers.) For projects, which are completed on or near the project due date and which provide a real impact to the customers, positive word of mouth (PWOM) usually results, spreading its way to other potential customers.

To help sustain and promote further PWOM, KBE must evaluate every new project based on how it aligns itself with KBE’s existing methods and strategy. (Refer to the set of requirements listed in Section 7.2.) For the most part, KBE looks to capture the “lowest hanging fruit” in every new pool of customer requests. Capturing the “lowest hanging fruit” is one method used by managers and management teams to get quick wins, especially in situations where immediate results are necessary. In E. Rasiel’s book, The McKinsey Way<sup>74</sup>, the “low hanging fruit” ideology is described in the following way:

“Sometimes in the middle of the problem-solving process, opportunities arise to get an easy win, to make immediate improvements, even before the overall problem has been solved... They create little victories for you and your team. They boost morale and give you added credibility by showing anybody who may be watching that you’re on the ball and mean business.” (Pg. 36)

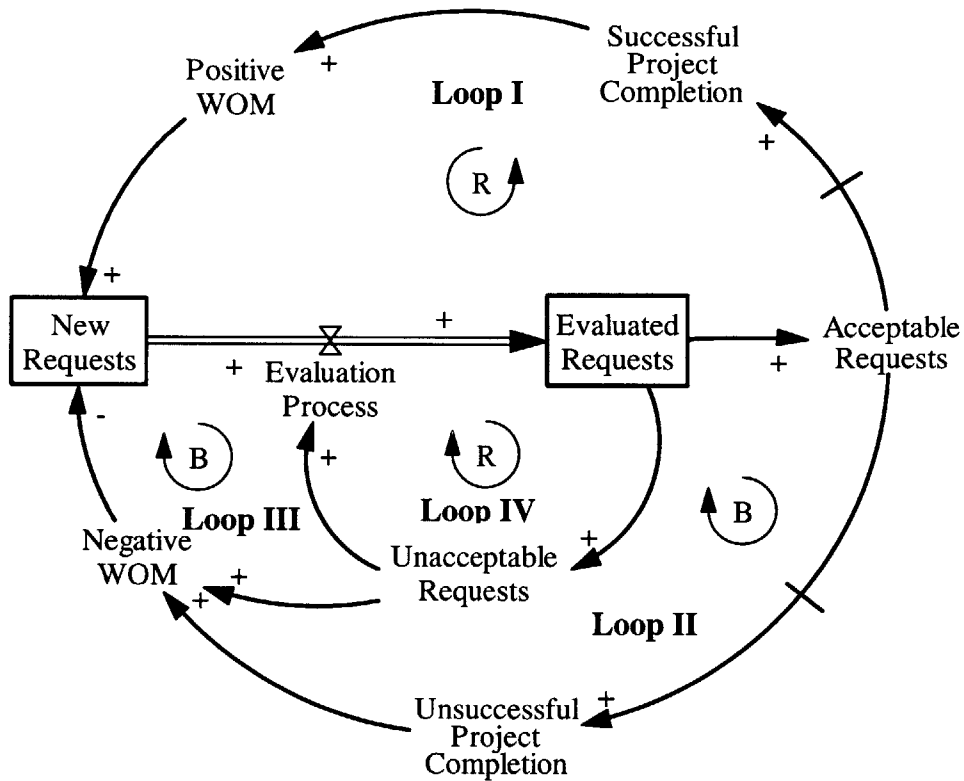
McKinsey use this ideology for the reasons just explained, and for the most part so does KBE.

However, doing this without raising the bar of improvement can result in a number of undesired consequences. KBE must always learn to improve on what it has already learned from past projects in order to raise its level of competency. Doing this requires that both KBE developers and managers understand the key, underlying principles in each project, share the knowledge that they have learned, and find ways to use this shared knowledge to develop more sophisticated tools for a bigger pool of customers. If KBE, as a whole, fails to do this, then it may find itself limited to only a few project types and customers. As a result, KBE may see the number of customer requests dwindle over

---

<sup>74</sup> McKinsey is a business management consulting firm, and one of the most respected in the consulting community. Rasiel was a McKinsey consultant, who worked for the firm from 1989 to 1992, and he describes his experiences in this book.

time. Figure 7.1 illustrates this situation, and others, using a series of interrelated cause-effect loops.



**Figure 7.1: Cause-Effect Diagram Illustrating KBE Strategy**

Before talking about what the figure actually describes, it is important to first explain the causal-loop structure actually works. The few paragraphs below show how casual-loops diagrams are normally represented. This information came primarily from J. Sterman’s unpublished textbook, Business Dynamics: Systems Thinking and Modeling for a Complex World. (The textbook is used in MIT course 15.874, System Dynamics in Business Policy.)

Each variable that has a box around it is called a “stock,” while each variable that describes the link between any two “stocks” is called a “flow.” The combination of the two is called a “stock-and-flow” structure. A “stock-and-flow” structure illustrates a special type of a cause-effect process.

A “stock” can be thought of as a reservoir, which holds information of some kind before it moves on to next stage. The information held in a reservoir is statically controlled, until it is ready to flow through the “pipeline” (the link between the two “stocks”) to the receiving reservoir. As the information runs through the “pipeline,” it takes on a dynamic form. A “valve” helps regulate the flow’s dynamic condition. (To see an

example of a value, look for the term “Evaluation Process” in the figure above; take notice of what it represents in the figure.)

Plus and minus signs show the type of cause-effect relationship that exists between two variables. A plus sign means that a positive relationship exists, while a negative sign means the opposite. The author provides some examples using the figure above for clarification. With all else being equal, an increase in the number of “Unsuccessful Project Completions” will most likely increase “Negative Word of Mouth (NWOM).” As a result, a POSITIVE relationship exists between these two variables; hence, a plus sign is shown next to the latter term. In another example, an increase in “Negative Word of Mouth (NWOM)” will most likely decrease the number of new requests—of course, with all else considered being equal. As a result, a NEGATIVE relationship exists between the two variables, as is illustrated with a minus sign.

In this structure there are a total of four loops, two that are reinforcing loops (R) and two that are balancing loops (B). A reinforcing loop characterizes a set of actions that allow for growth (or decay)—sometimes an unlimited amount of growth (or decay) if the loop is not restricted by certain real-life conditions. A balancing loop describes a set of actions, which are constrained by certain limiting factors. To determine if a loop is either a reinforcing or balancing loop, count the number of plus signs. If the number is even, then the loop is a reinforcing loop; if it is odd, then the loop is a balancing loop.

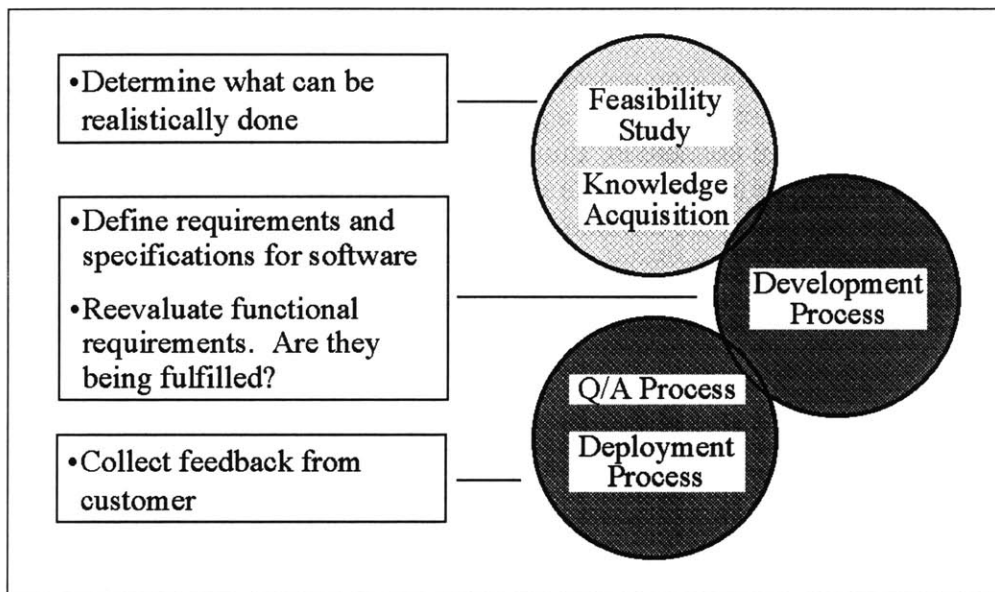
Although it is clear that KBE is fully aware of the positive results that come from using the “low-hanging fruit” theory (loop I), there does not seem to be conclusive evidence, at least in the author’s mind, that the Department is fully aware of some its negative implications. Negative word of mouth can develop in two ways: (1) from taking on a project that might not have a high-enough success rate (loop II), and (2) from not taking on enough challenging requests (loop III). These issues can promote NWOM and thus hinder the number of new requests that come in. This, of course, assumes that KBE continues to enforce the “low hanging fruit” method and not challenge itself along every step of the way to again raise that bar of improvement.

Another way that KBE can lose customer approval is by prolonging the evaluation process (loop IV). In many cases, the evaluation process is slower than the rate at which projects enter the system; as a result, the number of unevaluated requests, making its way through the “pipeline,” grows. In addition, a feedback loop ingrained into the evaluation process (loop IV) accentuates the delay. Projects, which do not “pass” the evaluation process the first time through, are normally “filed away.” This can happen for a variety of reasons: they do not align themselves well with KBE’s core strategy; KBE does not have the time or resources to take on these types of projects; similar projects are already being worked on; etc. After time, these projects will be removed from their “file cabinets” and re-evaluated. If the projects match KBE’s needs and resources during re-evaluation, they will be assigned to different development teams. But all of these decisions take time, and perhaps more time than the customers expect, developing this idea of negative word of mouth.

If these issues are not raised within the Department, then KBE may find itself desperately searching for new business. KBE may realize that the total requests on a yearly basis may rise (perhaps quickly at first), slowly reach a pinnacle, and then finally fall. These actions are all based on the so-called power struggle that goes on between PWOM and NWOM over time.<sup>75</sup>

#### 7.4 KBE Challenges:

KBE has to consider a number of different issues when fashioning any type of software tool for a customer's particular needs—many of which go beyond the limitations of the tool, itself. Many of these challenges can be classified into three key areas: the feasibility study and the knowledge acquisition; the development process; and the Q/A (question and answer) and deployment processes.



**Figure 7.2:** Challenging Areas that Face KBE

KBE has identified these so-called problem areas and time delays, but does not have any type of formal plan to isolate and attack them. The feasibility evaluation was already discussed in the previous section; more will be said about other aspects of the development process throughout this section.

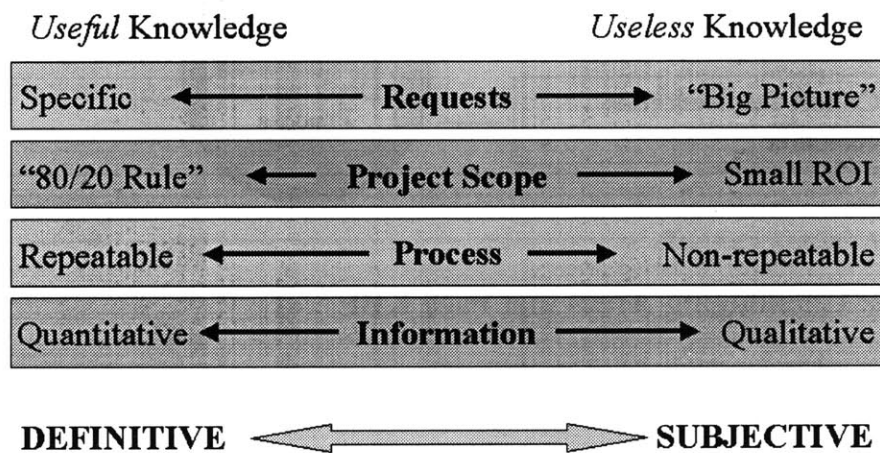
<sup>75</sup> KBE does not keep a formal record of the number of requests that it receives or the number of requests that it accepts, on a week-to-week or a month-to-month basis. Such information would definitely be helpful in trying to understand how the dynamic conditions illustrated in Figure 7.1 play themselves out in a real environment. A KBE manager, however, did mention that the gap between the number of requests received and accepted had stabilized in recent months; before then, it was steadily increasing.

When evaluating any request or going through the process of acquiring key bits of information for the potential advisor, KBE looks for particular types of information. The types of information that KBE looks to acquire supports the following doctrines:

- The particular request has to fall in line with the 80/20 rule, where 80% of all the results come from 20% of the total effort made.
- Each project must result in a “complete” software tool. (In many cases, this is done by looking for ways to automate existing manual processes.)
- The information acquired and used must be quantifiable, rather than subjective.

These ideas can be assorted and placed into two categories: customer-focused requirements and knowledge-base requirements. The former was discussed in detail in Sections 7.2 and 7.3; while the latter will be discussed in the remaining part of this section, in the context of KBE’s knowledge acquisition process.

During the knowledge acquisition process, KBE, with the help of its customer team, must determine what information is useful to a project and to the advisor. As was mentioned in Chapter II, advisors are not able to “think out of the box” like humans can. In other words, expert systems are not able to use subjective information to solve open-ended problems. Figure 7.3 shows the distinction between definitive (useful) and subjective (useless) knowledge; this figure helps to understand how KBE approaches the knowledge-acquisition process for any of its advisors.



**Figure 7.3:** Types of Information Acquired by KBE

Requests:

The project requests that come to KBE must be specific in their objectives. Many times the customers are not aware of what KBE can actually do and, as a result, do not put any bounds on their requests. The customers, at time, tend to forget or do not realize what KBE is capable of doing. KBE does not look to optimize an entire process; it looks at



particular parts of a process and optimizes each of them individually, using custom-built or commercial-bought software. One KBE manager had this to say, regarding the general requests that the Department receives:

“Well, in the past, they mainly gave us the ‘big picture’... what they wanted from a very general point of view. For instance, we get people from different SBUs constantly calling us up and saying, ‘Hey, we want to have something that can help support this...you know, that help us design a product quickly and easily.’...That’s how...when we get general requests like that, we know that this stage is going to take a lot of time. We want to make sure that what their saying is truly what they need...and if it is, we have to be sure that we can provide it to them.”

To help improve the process of identifying feasible requests, in the quickest allowable time, KBE is now looking to develop a standard request form for all of its potential customers to complete. However, as it was brought to the author’s attention, creating a form that, in a sense, converts subjective information and requests into a framework that can be helpful to KBE during the feasibility study is rather difficult. One must also keep in mind that the request form must be able to accommodate various customer requests.

#### Project Scope:

Because each project is time dependent and customer driven, KBE emphasizes the use of the “80/20 Rule” in each of the projects that it works on. In other words, KBE develops tools that will improve a significant part of a process, in the time that it is given to develop the tool. The tool may not be able to improve an entire process; however, in many cases, it is not worth the additional effort or time to develop tools, which consider the remaining part of a process.

However, finding out how the “80/20 Rule” can be applied to each evaluated project is usually a challenging, time-consuming process. It requires that the project be clearly defined (as explained in the previous category) and that it aligns itself well with the core competencies of the Department.

#### Process:

With each project, KBE looks to mimic many of the small processes that designers and engineers methodically do. This requires finding repeatable steps in a process and finding ways to automate them, so that designers and engineers can use their time and efforts more efficiently to do things that the advisor is not able to do. However, most processes are ingrained with various non-linear characteristics, which cannot be simply diagrammed as a decision tree or automated. Situations like this occur when designing products to meet certain assembly and packaging needs. The case study at the end of this chapter will illustrate that automation is not necessarily the best way to tackle a design problem, especially when non-linear “forces” like package space need to be considered at all times.

Information:

One of the most important steps for KBE during the feasibility and development processes is to understand what design rules really mean. Each design rule that is taken under consideration must be understood quantitatively. This ideology falls in line with what has been already mentioned in Chapter II: expert systems are not able to think subjectively; they need a quantitative method for comparing decisions throughout a process. Extracting this type of information, especially from design manuals, is not always easy to do; in many circumstances, KBE relies on the expertise of designers and engineers to further define these types of rules. A KBE manager had this to say, regarding this issue:

“You have all of these rules...a lot of them may be very suggestive. For instance, [pointing to a headlamp, which was on his table] ‘This headlamp needs to look good.’ How do you determine that?”

“Yeah, those things are too subjective. We’ll ask, ‘What do you mean by that? What type of specifications, parameters, and criteria are we talking about here?’ Can those things be reasoned out? How do we determine that? So for those things, it takes a long time to do.”

To improve productivity, KBE has hired more software developers and technical consultants. Its employee count has grown considerably and is now approximately at 40 to 45 people. By hiring more personnel, KBE hopes to reduce or eliminate the problems that it faces throughout its on-going development practices. However, increasing the hiring rate and the number of projects per year may also introduce new issues—issues that can develop internally without extensive departmental communication and synchronization. For example, software sharing and standard documentation of past work can help avoid any unnecessary redundancy. KBE is now realizing the impact that this can have on their practices, and is now looking for ways to improve knowledge sharing within its own ranks.

## 7.5 Case Study: The Impact that Packaging and Product Architecture Have on the Design of KBE Advisors

As was mentioned in both Chapters V and VI, packaging is a major issue in the design of various vehicle sub-systems. In many ways, it dictates the shape (i.e. architecture) of many components and sub-systems and the position of several others. It also has a major impact on how each KBE advisor is developed. Each KBE development team has to consider the nature of the projects that it works on; often this means that each tool must be tailor-made to accommodate the unique set of requirements set forth by the products, which the tools intend to evaluate.

This section of the chapter will describe how package requirements and product architecture drove the design of two KBE advisors, the KBE-HVAC and KBE-wiper systems. Both development teams had different sets of conditions and limitations that

they needed to consider based on the fundamental characteristics of the products that the advisors intended to analyze. Some of the challenging aspects of each project will be discussed as well as reasons why the KBE-wiper system provided more user flexibility.

The results that come out of this case study can be beneficial to KBE and other types of organizations, which plan to develop a generic process advisor. The case study describes the advantages and disadvantages of parametric modeling in a design environment and explains why it is important to understand the fundamental parameters of a product and its architecture even before designing an advisor.

### 7.5.1 KBE-HVAC Project:

The KBE-HVAC advisor was developed to help designers make HVAC systems “on the fly,” by eliminating processing delays that normally occur between design and analysis and by introducing the designer to parametric modeling. The KBE-HVAC made two versions of the advisor: the first version was highly parametric while the second version was “loosely” parametric. There were two reasons why a second version was made: (1) the first version was unable to consider all types of packaging conditions that designers face, and (2) experienced designers wanted a tool that can generate suggestions—not a tool that drives the designer down a particular pathway.

The nature of parametric modeling was described in Section 3.6.2 while explaining Visteon KBE’s top-down/bottom-up approach for developing KBE tools. However, to summarize, parametric modeling is normally used in a CAD environment to locate features on a part or parts with respect to other parts in 3-dimensional space using a series of “links”. (Part-to-part relationships will mostly be discussed in the context of this section.) These “links” may require two selected surfaces on separate parts to be parallel or perpendicular to one another; in addition, the “link” may require that the same two surfaces be placed a certain distance from one another (no more than X-units away, but no less than Y-units away). Of course, “links” can also be placed on two points, two edges, one point/one edge, one point/one surface, and one edge/one surface. Parametric modeling allows design changes to be made quickly. The designer does not have to position each part once a change is made to the design; all he has to do is move one part, and the rest will follow, based on the established links between the different parts.

The first version of the KBE-HVAC advisor used a high degree of parametric modeling to automate a wide variety of operations. The designer first selects a set of key components for his design (e.g., a specific evaporator core, heater core and blower motor). Based on the model components selected and the location/position of each component in space, relative to one another, the advisor then brings up a computer-generated model of the casing—some type of a semi-center mount system or a handed system. (Note: The advisor has to consider the different variations of each architecture type as well. For instance, the semi-center mount system can be sub-divided into left-side and right-side systems. The reader may recall from Section 6.3 that the semi-center mount system is defined by the location of the blower motor relative to the main housing.) After all the key parts are in place, and the casing is generated, the designer

finally modifies the design to accommodate the given packaging boundaries that are set for the unit.

A representative from the KBE-HVAC team had this to say about Version 1:

“You select the components you want. Those standard components have different geometrical configurations and have different assembly sequences. You select them, and the system will select the design for you. You can...design the housing by selecting a set of components...It [The system] brings up an architecture...and every geometry that relates to that architecture.”

Unfortunately, with this system, the designer is very limited in terms of what he can do to the parametric model. He is able to change the overall size of the model (i.e., scale the model up or down) and slightly reposition certain parts of the unit (e.g., the blower scroll can be rotated with respect to the body of the unit, but only marginally). The model cannot be radically manipulated so that its shape is completely distorted. The blower motor scroll, for instance, cannot be moved to the other side of the unit, nor can the evaporator case be twisted or bent. The parametric links, which define the model, do not allow such changes to happen. In other words, the model takes away some of the needed flexibility from the designer. If the package conditions change beyond the scope of the model, then the designer has no way of using the model successfully.

For the second version, the KBE-HVAC team rearranged the way the advisor helped the designer develop an HVAC design. In this case, the packaging conditions were considered first, paving the way for the remaining sets of decisions. The parameters for this model were not as well defined as they were in the previous version (in other words, less parametric links were used) for two reasons: (1) to allow the designers to make changes to the model, to accommodate the changing packaging space, and (2) to create a mesh, in quicker time, for a CFD (computational flow dynamic) analysis. The same KBE-HVAC representative had this to say about Version 2:

“Now, with the current process, we can design the whole thing on the fly...so that the architecture is determined by the design [parameters, such as the packaging region].”

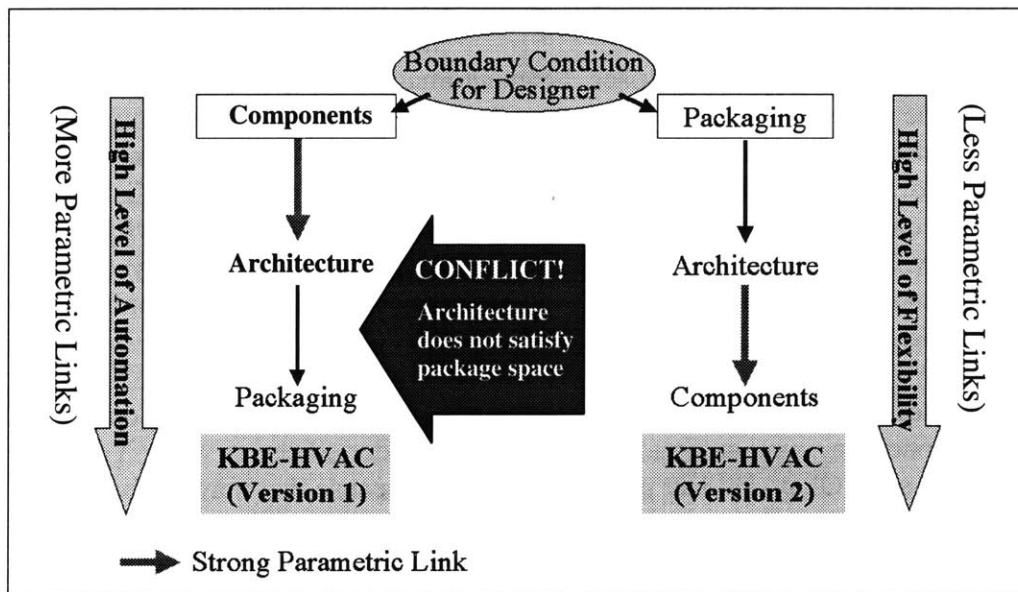
“Say you have a blank sheet. You...you start from scratch. You only have the packaging space...basically...so it’s usually before everything...even the architecture, itself. The architecture doesn’t matter that much. You...you can package your design in that space. That’s it.”

“Using the current version, we do the design. When you make it parametric...our components will still work for this second version. The only thing is that...this time, the boundary conditions are the package

space. In the first case, you can make it bigger...but you have to make the whole case [bigger].”

“You can still change it, make it bigger. That’s easy to do. That, you can still do. But...I mean, this new approach...relies more on the designer. It is essentially...the designer can do anything that he wants. We don’t control that much. Of course, that limits many of the automatic features that we have.”

The technique used by each version is shown below in Figure 7.4; included in the figure is an illustration that shows why the first version did not accommodate the designer’s needs. The boundary condition is defined as the driving characteristic, which all other attributes acknowledge and fall under. For example, in Version 1, the design is defined by the selected components and their connection to each other; the architecture and packaging conditions are next to follow, but never change the way that the components are tied together. In Version 2, the complete opposite is true: the packaging condition (i.e., a set of boundaries that define the given region of space for the unit) drives the design of the casing and the location of the components in that casing.



**Figure 7.4:** Process Used by the Two Versions of the KBE-HVAC Advisor

The first version shows the process that was used to develop a highly parametric model, which is strongly linked to the architecture of the unit; while the second version illustrates how a highly flexible system was developed to accommodate the designer’s needs. In both cases, there is a strong link between the architecture and the components that go into the architecture. The first version had difficulties in the design environment because it did not follow the natural process in which most vehicle sub-systems are

designed—first packaging, second architecture, and third components. The difficulty really emerged when designers tried to fit their parametric models into the given package space and were unsuccessful. What the development team learned from this was that packaging drives the design of the architecture; as a result, a new version was developed, which allowed the designs to be package focused, and not component focused.

From this experience, it is also important to determine early on what is, in fact, the proper mixture of user flexibility and automatic processing/parametric modeling. This is a settlement that each KBE team must decide on, with the help of the customer. No tool exists that has both high flexibility and software automation; there must be a compromise between the two issues. The KBE-HVAC team is currently trying to find ways to enhance both automation and flexibility, but this is very complicated. It requires the use of sophisticated algorithms to understand the relationship of every part in the design, especially when extreme changes are made. For example, if a part was moved and/or rotated in such a way that the net effect would be seen throughout the system, then the underlying algorithms would have to find a “global solution,” which would meet each link’s requirements. The algorithms may also have to be written to use different sets of rules and links for different design changes. Here is what the KBE-HVAC representative had to say, regarding all of this:

“If you give me a package space, I can automatically put [the components] in some position for you. I mean, sure I can give it a position for you, but that doesn’t mean it will satisfy your packaging space. So you normally want to...we want to be able to put [them] in the same position so that [they] can satisfy your package space, but that is difficult. That’s why we...we could never claim that we would be able to create this for you. We can automatically place anything for you in any place, but the only thing is...does this mean that the design will break down?”

“We need to consider the packaging space...because when you put one component in...you automatically have to consider others, as well. You cannot say, ‘Okay, I’ll just put this one here,’ because that may not be the appropriate place to put that particular component. So...this really becomes a big packaging problem. So we are in search of good algorithms to do this automatically. But...as far as I know, it is a very complicated issue.”

In the first version, the KBE-HVAC advisor used a very high degree of parametric modeling; this allowed many design changes to be done rather quickly. Unfortunately, it also limited the number of ways in which designers could change or improve a design. This is a critical point especially when one considers how package space can change over the course of a project; in a sense, the model needs to be as flexible as it is.

In the second version, more flexibility was created to accommodate the changing requirements for packaging space; however, with this move, a large degree of automation

was lost. In other words, changes to a design may take longer in this version than they would in the first version. This is one of the tradeoffs that both KBE and its customers must collectively consider.

In the next sub-section, the author will show what type of advisor expert and novice designers prefer, adding a different dimension to this paradox.

### 7.5.2 KBE-Wiper Project:

Unlike the KBE-HVAC advisor, the KBE-Wiper advisor uses a very low degree of parametric modeling,. The intent here was to give designers—experienced designers, mostly—the flexibility that they needed to make a variety of wiper modules. The reasons being, there exists approximately 53 to 56 different types of wiper designs, and designers, as mentioned earlier, prefer tools that give suggestions rather than make decisions for them. (Packaging space also has to be considered when designing a wiper module—like it is for most other modules in a car.)

The KBE-Wiper advisor uses a skeleton structure (an X-bar/link design, where X represents some number) to represent a wiper module. (A similar type of methodology is described in D. Whitney’s work, regarding assembly design. In his course, 2.875 “Mechanical Assembly and Its Role in Product Development, D. Whitney speaks about how assembly data can be captured without the need for feature recognition, by simply looking at the assembly features—specifically, their location, orientation and assembly direction in space. The KBE-Wiper system uses simplified models to capture and represent the most important aspects of the design of a wiper module.) Even in this simple representation, certain important kinematic properties (e.g., angle or rotation, angular velocity and angular acceleration for each bar and link as a function of crank angle) can be extracted and then compared to the given kinematic requirements. General CAD models are imported, and then inserted into the “stick” structure at their respective location and position. (The imported models are parametrically linked to various coordinate systems found on the wiper model.) Each imported model is represented as a simple cylinder; the designer can make changes to the length and/or the radius of each model. (More detailed CAD parts can be imported later, if necessary.)

The designer can use the model to check for any part-to-part interference at various wiper positions. The model can be readjusted for different package requirements, by using a “rubber-band” method: the designer simply moves different parts of the module to other locations (after locking in some parts and relative motions); the advisor then recalculates the wiper’s performance values.

The general process for developing a typical wiper blade in the KBE-Wiper advisor is as follows:

1. ENTER the number of linkages for the wiper design
2. ENTER constant properties (e.g., coefficient of friction)

3. RECEIVE kinematic values for wiper design (e.g., wiper force, angle of rotation, angular velocity, angular acceleration, as a function of crank-angle)
4. ENTER the location of the center lines for each shaft
5. ENTER the size of each bar, the location of all center lines, and the restricted range of motion for each shaft
6. ENTER the location of “ball points” (the points at the end of the bars)
7. ENTER the lines that will represent linkages (Connect the different “ball points” with a line)
8. RECEIVE coordinate systems for each represented bar
9. ENTER general CAD models from part library to be attached to the skeleton structure

The process used by the KBE-Wiper system is more flexible and follows more of a top-down approach than the KBE-HVAC system. The reason why the KBE-Wiper has these characteristics is because the wiper design can be simplified to a combination of bars and links. For more complicated shapes, as in the case of any HVAC casing, it is rather difficult to make such a simplification possible. (The designs cannot be represented as simple shapes.)

### 7.5.3 Overview of Parametric Modeling:

This case study brought up an interesting point that deserves to be talked about: products that derive their shape from packaging (products that have an “exoskeleton”—a container that holds parts in position) are usually difficult to automate, while those that derive their shape from a combination of links (products that have an “endoskeleton”) are relatively easy to automate. The table below explains why this is the case.



BASIC FRAMEWORK	Exoskeleton	Endoskeleton
TYPE OF MECHANISM	Containers (i.e., "Parts in a Box")	Linkages
EXAMPLES	<ul style="list-style-type: none"> <li>• HVAC Units</li> <li>• Cars</li> </ul>	<ul style="list-style-type: none"> <li>• Wiper Modules</li> <li>• Sliding Door Mechanisms</li> <li>• Trusses that are under-constrained (i.e., trusses with mechanisms)</li> </ul>
WHAT DOES PACKAGING INFLUENCE?	The shape of the container; and the location/position of each component with respect to the container and to other components	The position of each link with respect to its neighboring links.  (Note: The location of each link is naturally confined to a certain region of space because each link is physically connected to other links.)
CAN AUTOMATION AND FLEXIBILITY CO-EXIST IN A KBE ENVIRONMENT?	<p>PERHAPS, BUT DIFFICULT:</p> <p>One has to consider both the implicit and explicit relationships that exist.</p> <p>For example, a part may be physically attached to a container, but its location and orientation are dictated by other parts.</p>	<p>YES, VERY REALISTIC:</p> <p>A fixed number of explicit parameters (per link) are all that need to be considered when developing the algorithms</p>

**Table 7.1:** Products with an Endoskeleton and an Exoskeleton

## 7.6 Chapter Summary

This chapter spoke about some of the different challenges that KBE endlessly faces when developing advisors to fit a particular customer need and are in line with KBE's strategy and philosophy. The chapter looked at these challenges from two perspectives: (1) the process for acquiring information, developing the advisor, and deploying the advisor; and (2) the limitations that are infused within each advisor.

In addition, a case study was included to show how packaging can drive the development of advisors in different directions. Two advisors were considered for this case study, the KBE-HVAC and KBE-wiper systems. Each system used parametric modeling to a different degree based on the complexity of the designs that they intend to evaluate. A note was also made about the conflicting issues between automation and designer flexibility: how each is considered depends on the type of designers who will be using the system and the degree in which packaging space tends to change over time.

The next chapter will look at the particular challenges that one group within KBE, the KBE-DFM/A team, faces as it tries to design and develop an assembly advisor, which will be used to evaluate most vehicle sub-systems designed at Visteon. Many of these challenges come from the issues mentioned in this chapter, while others come from the lack of assembly knowledge that exists today.

# CHAPTER VIII

---

## KBE/DFA TRADEOFFS AND PITFALLS

### 8.1 Introduction:

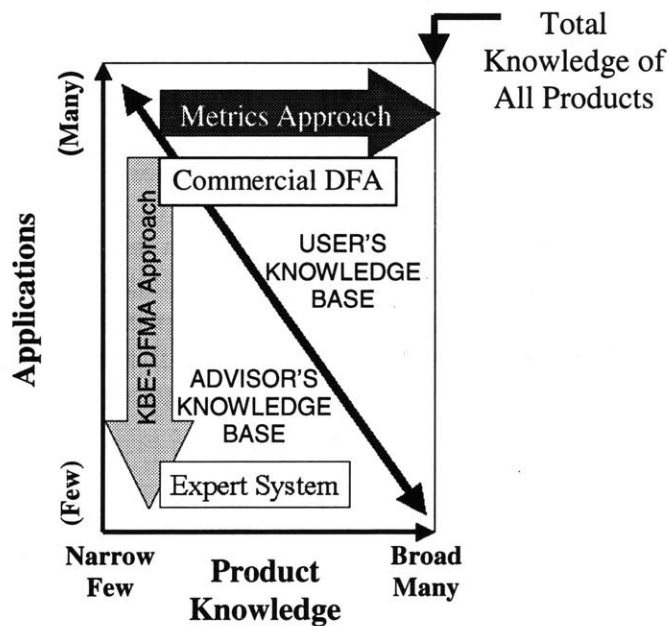
Chapter VIII will discuss the approach that KBE's DFMA (Design for Manufacturing and Assembly) section is taking to develop an assembly advisor that will be used within Visteon. In addition, the chapter will look at some of the challenges the Section faces as it goes through the process of determining their own requirements for this advisor. KBE-DFMA is planning to apply the same type of principles that many of the other KBE sections use when developing product- or process-specific advisors. Will this approach be feasible for an assembly advisor? The chapter will answer this question by explaining what KBE-DFMA's work will and will not be able to accomplish—and why.

This chapter will discuss the following topics:

- What is the better approach for an assembly advisor, breadth or depth? Assembly contributes to several products in different ways. How will assembly advisor be made to tackle this problem? Expert systems focus on particular processes and/or products, limiting their use to only particular applications. Current DFA methods and tools can be used to evaluate the assemble-ability of various products, but require the expertise of the user to help make the evaluation successful.
- What is KBE-DFMA's approach to the development of an assembly advisor? KBE-DFMA has to deal with the issue of breadth versus depth, but in addition, it has to contend with many of the challenges that the Department, as a whole, faces.
- What types of commercial tools and methods are currently available for assembly analysis? It is important to identify what current methods and practices lack and build them up from there.

## 8.2 The Issue of Breadth versus Depth:

To make an assembly advisor that is general enough to encompass all types of design applications (with detailed suggestions) requires knowledge about each product that it analyzes. This, however, is almost unrealistic, at least during initial development. The practical development of an assembly advisor requires “stretching” knowledge to either fit a wide range of products or to fill the needs of one specific product or product family. Using the former approach usually results in general, indecisive suggestions that are loosely linked to the design under analysis. (This is the pathway that most of the technical literature on the subject of DFA normally follows, as was mentioned extensively in Chapter IV.) Using the latter method, on the other hand, results in a detailed set of suggestions that can be quickly used in an application, but with a price: the tool, itself, will be limited to only a few applications and products. This then brings up the question, what is better—breadth or depth? Figure 8.1 helps to illustrate this conflict.



**Figure 8.1:** The Assembly Advisor Dilemma

The figure above shows one way that most of today’s process advisors can be organized. All advisors contain some level of competency about the products that they analyze (product knowledge) and some range of usefulness (applications). However, in most cases, these two defining characteristics usually follow opposite paths. The following two examples describe this in more detail.

### Breadth, but No Depth:

Most commercial software tools used for assembly analysis can be applied to a variety of applications, but they also require assistance from the user to provide them with key information about the products. In this situation, most information about the products are extracted from the user’s “knowledge base,” and not the from advisor’s knowledge base.

(The arrow running diagonally across the figure defines the location where the advisor's knowledge base will no longer serve a useful purpose, for a given application; it is up to the user to continue the analysis from there.) Furthermore, because the recommendations, provided by the tools, are mostly product independent, the user is required to return back to his own "knowledge base" to find ways to use these suggestions in the most effective way he knows how. (The author uses the term "Commercial DFA" to describe these types of tools; the reader can see where these tools are placed in Figure 8.1.)

#### Depth, but No Breadth:

Expert systems contain enormous amounts of information about individual products and the process for designing them; the user has to do very little to help the expert system converge to a specific answer. However, these types of advisors are applicable to only a few products or applications. Expert systems can only follow the heuristic framework contained within their knowledge base; therefore, they are unable to use "out of the box" thinking to analyze products, which may be out of their scope.

However, the conflict, which the breadth-versus-depth paradigm helps to create, can be eliminated, to an extent, with the use of measurable standards and metrics. For example, some design-for-assembly tools use cost and time for assembly to measure and compare the assemble-ability of different products, while others use a point system to make a comparable assessment. This idea, however, needs to be pursued further; the type of metrics used in design for assembly today do not successfully resolve the conflict shown in Figure 8.1. (More will be mentioned about this in Section 8.3.)

KBE-DFMA plans to create an assembly advisor by first, purchasing different commercial tools ("manual" tools as shown in Figure 8.1), which evaluate different aspects of assembly; second, linking them together so that a "complete" software tool is created; and third, customizing the tool (i.e., developing the tool's knowledge base) so that it can be used for various applications. (Figure 8.1 illustrates this plan with the large arrow inscribed with the expression "KBE-DFMA Approach.") The process will be gradual, as KBE-DFMA finds ways to include more and more product-dependent information into the advisor's knowledge base. More will be said about this in the next section.

### 8.3 KBE-DFM/A's Approach to the Development of an Assembly Advisor:

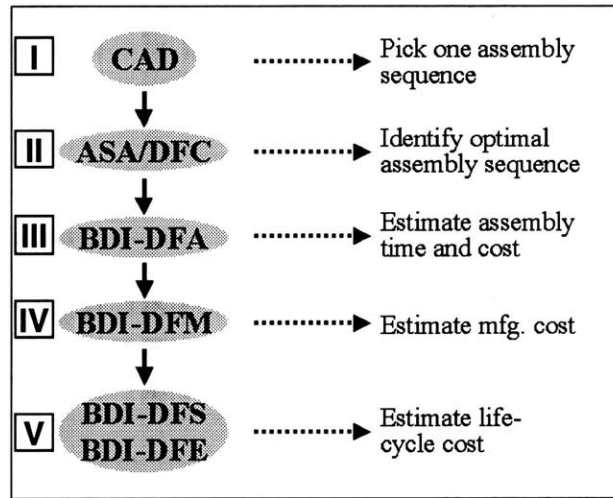
At this time, the KBE-DFMA team is using commercial software from Boothroyd-Dewhurst, Inc.<sup>76</sup> (BDI) to evaluate different vehicle sub-systems at Visteon; the team is using the software as part of a service to its customers, but plans to incorporate the software into its own custom-built assembly advisor.

---

<sup>76</sup> Refer to <http://www.dfma.com> for more information about the Company and its software tools; The web site also provides some case studies, showing how these tools have helped reduce overall product cost, in certain applications.

BDI currently sells four different software tools: BDI-DFA™ (Design for Assembly); BDI-DFE™ (Design for Environment); BDI-DFM™ (Design for Manufacture-ability); and BDI-DFS™ (Design for Serviceability). The tools, when used together, provide the user with a “complete” cost evaluation of a given product design, and in turn, reveal areas of the design where costs are high. Designers can then look at these high-cost areas and try to find ways to improve them.

Figure 8.2 illustrates one KBE team member’s idea for what this future advisor will include and the sequence of steps that the user will follow when using it.



**Figure 8.2:** KBE Assembly Advisor

**I. CAD (Computer-Aided Design) Environment:**

The advisor is intended for the designers’ use. This is a logical choice: designers usually have the most influence on a design; and they are very knowledgeable about the design process and about the products that they help design. However, in order for the advisor to be fully accepted, it must be able to “fit” within the designers’ CAD environment. This allows information to be quickly transferred from the CAD tool to the advisor, and visa versa.

**II. ASA (Assembly Sequence Analysis)/DFC (Datum Flow Chain):**

KBE-DFMA appears very interested in incorporating a suite of assembly evaluations tools that were developed over the years by D. Whitney, his colleagues at Charles-Draper Laboratories, and his students at MIT.<sup>77</sup> The suite includes a Datum Flow Chain (DFC) editor and an Assembly Sequence Analysis (ASA) module. The DFC editor provides the following:

<sup>77</sup> See the Suggested Reading Material Section for a list of papers and journal articles about these tools.

- “[A] graphical representation of the designer’s strategy to locate parts with respect to each other, which amounts to specifying the underlying structure of dimensional and datum references on parts constituting assembly.”<sup>78</sup>
- A plan for generating assembly sequences.

The ASA module provides the user with all possible assembly sequences based on the assembly parameters illustrated in the DFC editor and listed by the user.

### III. BDI-DFA™:

BDI-DFA evaluates a given design using time and cost metrics. Before it can do this, however, it requires some particular information about the product (e.g., the weight, size and general shape of the individual parts) and the process for assembling the product (e.g., the sequence of assembly). The assembly sequence, of course, will come from the assembly sequence generator. (More will be said about BDI-DFA in the next few sections. This software tool is a key contributor to the current methodology of design for assembly; therefore it deserves to be looked at separately.)

### IV. BDI-DFM™

The cost analysis from BDI-DFA will be used in conjunction with the cost analysis from BDI-DFM to provide the user with a total cost structure for fabricating a product. The user can use the results from these two tools to consider ways to optimize the total cost for fabrication. (The “hidden” methodologies behind design for assembly and design for manufacture-ability normally conflict with one another. For example, DFA can indirectly increase the cost for manufacturing parts, and DFM can indirectly increase the cost for assembling the entire product. Unfortunately, at this time, no metric exists, which considers both the costs for manufacturing and assembly. The user, in a sense, must find a way to reduce the total fabrication cost, based on these two separate results. More will be said about this later in this section.)

### V. BDI-DFS™/BDI-DFE™:

To provide the user with a complete cost evaluation of a product design, KBE-DFMA is planning to include both BDI-DFS and BDI-DFE into the assembly advisor.

Currently, all of these tools require information to be manually entered. One of the things that KBE-DFMA wishes to do is link these packages together and to the CAD tool so that evaluations, and more importantly design revisions, can be done in quicker time. The author has identified some areas where these tools have the potential to be coupled:

#### Link between Commercial Packages and CAD Models:

The BDI suite is currently unable to read and interpret CAD files (e.g., IGES files). As a result, the user must enter information about each part in the assembly (e.g., size, shape, texture, and material), the assembly sequence, and assembly directions and orientations.

---

<sup>78</sup> Refer to paper “Integrated Computer Tools for Top-Down Assembly Design and Analysis” by R. Mantripragada, J. Adams, S. Rhee and D. Whitney, Dept. of Mechanical Engineering at MIT.

KBE-DFMA is primarily looking to link I-DEAS to the BDI suite, thus making the whole evaluation process, from set-up to execution, simpler to do for the user.

The Datum Flow Chain (DFC) editor and Assembly Sequence Analysis (ASA) module require the designer to understand how constraints are distributed throughout the assembly and what factors limit the generation of some sequences, by referring to his drawing or model. The designer, in a sense, closes the gap between what information the computer tool needs and what information the drawing or model illustrates.

**Link between BDI-DFA™ and BDI-DFM™:**

BDI's DFA/DFM software tools are "loosely" linked with one another. In other words, the information gathered from one tool's evaluation needs to be entered in by hand in order for the other tool to make its evaluation. Both tools are used together to help the user determine the optimum cost for manufacturing and assembling a product. However, one must understand that there is tradeoff between part consolidation and part simplicity. Even though part consolidation can help reduce the time and cost for assembling a product, it simultaneously increases the complexity of each part and the cost for manufacturing them. Likewise, part dissociation can help push manufacturing issues down while, at the same time, push assembly issues up. The BDI suite is not able to optimize the cost of a design from both a manufacturing and assembling standpoint; decisions to do this are in the hand of the user.

Although the author recognizes the high level of complexity for linking BDI-DFA with BDI-DFM, this is an area that still needs to be explored. (Linking the two packages in this way may require the use of a new set of parameters that link assembly and manufacturing metrics together; more information about the products under analysis, which the existing tools are not able to do; or the simple understanding that, in most cases, cost considerations for manufacturing significantly outweigh those for assembly.) If attacked successfully, the benefit will be quicker iterations between design for manufacturing and assembly. However, one must be cautious to the fact that BDI-DFA shows a bias toward part consolidation.

**Link between Advisor and CAE (Computer-Aided Engineering) Tool:**

What the framework in Figure 8.1 does not show is how its recommendations will affect the functional intentions of the intended product. In order to determine if changes to a design will affect the primary functional requirements of that design, it will be important to evaluate the model using some type of Computer-Aided-Engineering (CAE) software. From KBE-DFMA's point of view, and from the point of view of its customers, it might be necessary to link not only the BDI suit to I-DEAS, but also to a CAE package. This will allow the end user to make quick changes within the product's scope. In other words, the iterations will be quick, without serious delays.

#### **8.4 A General View of BDI-DFA™:**



#### 8.4.1 What does it do and how does it do it?

Generally speaking, BDI-DFA is a tool to help designers think about assembly in their designs. The software guides the user through the assembly by “asking” him questions about the nature of each part: Do they necessarily have to remain dislocated with other parts? Does the functionality that each one holds disallow it to be consolidated with another part? The tools simply make the user think about the significance of each part, under evaluation.

BDI-DFA and tools like it are popular in the product-development community because they convert subjective, qualitative information, into a form that can be quantified and compared for at a useful level. In the case of BDI-DFA, cost and time metrics are used to make this determination. Once an evaluation is made about a design, the results are calculated and shown to the user so that he can compare it with previous evaluations. There has been some controversy, however, as to how accurate the time and cost figures are compared to the actual ones. But this is really not an important issue, if one is using the results for comparing one evaluation to another.

However, there is concern for what is used to quantify and compare different evaluations. Do time and cost assessments provide the only keys to good design-for-assembly practices? The assemble-ability of a design may not only be determined by how much it costs to assemble a product, or by how much time it takes to assemble it. Of course, these issues have a significant effect on the outcome of any design, but by just looking at these parameters, in isolation, the designers can be limited to only particular decisions and pathways.

D. Whitney made the following statement about this issue in an e-mail to the author (Jan. 7, 1999):

“You recall from the assembly class [MIT Course 2.875: ‘Mechanical Assembly and Its Role in Product Development’] last fall that we emphasized the system nature of assembly and that you can’t focus too much on any one aspect, such as the time to assemble each part. DFA focuses on assembly, and no one really does, so it is important that someone direct attention toward assembly issues. However, the time it takes to assemble a part may be the least important thing about that part. There may be good functional reasons why a part is the way it is, as well as lots of reasons why you might want to redesign it.”

BDI-DFA does not “look” at issues, which the designer needs to consider—issues that enforce a particular design path. If any evaluation is made without first considering the product and the various processes that exist to make up that product, then any suggestions which come from that evaluation may oppose certain existing conditions. Issues, such as those that follow, should be brought up during this kind of assessment:

- How does this assembly fit the type of architecture that is to be supported?

- How does this assembly fit the assembly scheme used on the factory floor today? Is it conducive for manual or automatic assembly, or a combination of both?
- How will a change now affect improvements or variations in designs for the future?
- Which parts are manufactured by suppliers? How does this affect the chain of assembly? Are any critical characteristics being shared between the assembler and its suppliers?

Other assembly methods face the same fundamental problems that the Boothroyd-Dewhurst Method faces; some examples include the IP Stuttgart Method, the Lucas DFA Evaluation Method, and the Hitachi Assemblability Evaluation Method (AEM). The IP Stuttgart Method determines the cost for assembling parts together using robot equipment; it uses statistical analysis methods to piece together general information about robot equipment and various product designs. The Lucas and Hitachi methods use assembly indices to evaluate the level of difficulty for individual assembly operations and to determine, in a quantifiable way, the overall assemble-ability of the product. Other methods use fixed point system to determine the assemble-ability of a product. (For a complete description and comparison of these different methods, refer to Redford and Chal's book, Design for Assembly: Principles and Practices, Chapters 6 through 9.)

#### 8.4.2 Why does KBE-DFMA use BDI-DFMA™?

Presently, the BDI-DFMA is the only tool that KBE-DFMA uses to evaluate assemblies. Although it is not the only assembly-analysis tool on the market, it does provide some advantages for KBE-DFMA and the entire Department<sup>79</sup>. As one member of the section said, BDI-DFMA:

- Is, in general, the “best” commercial software package for assembly. (Prior to purchasing the software, the Section did an extensive study on all of the different assessment tools for assembly, and this was the conclusion.)
- Is user friendly.
- Includes learning seminars.
- Already exists within Ford. (A good relationship between Ford and BDI was already established prior to KBE's purchase.)
- Is relatively inexpensive (compared to other software packages).

---

<sup>79</sup> BDI has a significant competitive advantage in this industry. BDI has sued many of its competitors for copyright/patent infringement. (Some such cases are mentioned in the Suggested Reading Material Section.) For example, Sapphire was one company, which sold an assembly-analysis tool similar to BDI-DFMA. BDI learned about this and sued the Company. Sapphire lost the court case to BDI and is no longer in business. Much of the information tied into its software tool is in fact protected under copyright/patent laws; even the name “DFMA” is a trademark of BDI.

## 8.5 The Use of BDI-DFA™ in the Ford/Visteon Environment:

As mentioned earlier, KBE-DFMA uses BDI-DFA to evaluate the assemble-ability of a design, within the confines of assembly cost and time. When the team consults with a designer or a team of designers, it normally asks the same type of questions that the software “asks” its user. Can these two parts be combined into one part? Can these screw fasteners be replaced with snap-tabs or push-fits?

At this time, KBE-DFMA has evaluated six vehicle sub-systems using BDI-DFA:

- Two fuel tanks
- One car radio
- One major sub-assembly for a car seat
- Two headlamps
- One instrument panel/console sub-system

It plans to seek out other projects, from other product units, as more and more department managers become aware of KBE-DFMA’s work. The team is now planning to do the same type of evaluation on an HVAC unit. (This is one of the reasons why KBE-DFMA is interested in the HVAC analysis described in Chapter VI.)

The level of success for each project was dependent on several factors: the total number of parts/components for assembly; the number of parts/components developed in-house; the time that it takes do a DFA analysis; and the amount of time spent with designers and engineers. (Success, in this case, is valued as the number of suggestions made by the tool that can significantly impact the time for and cost of assembly, and the number of these suggestions, which can be immediately implemented into the existing design.)

### Number of Parts:

If an existing product design consists of too few or too many parts, then it will be rather difficult to make a successful evaluation. In the former case, examining a product with few parts does not give the evaluator too much “room” to make any suggestions; consolidating the assembly any further may inhibit some of the product’s intended functionality. In the latter case, a product like a car seat, which has hundreds of parts, may be too complicated to analyze in one month, the general time allotted to KBE-DFMA to analyze an assembly. However, this all depends on how well the assembly is defined. Without a physical assembly or a CAD drawing to see and analyze, the project takes longer (e.g., two months for a seat sub-assembly with 30 parts).

### In-house Components:

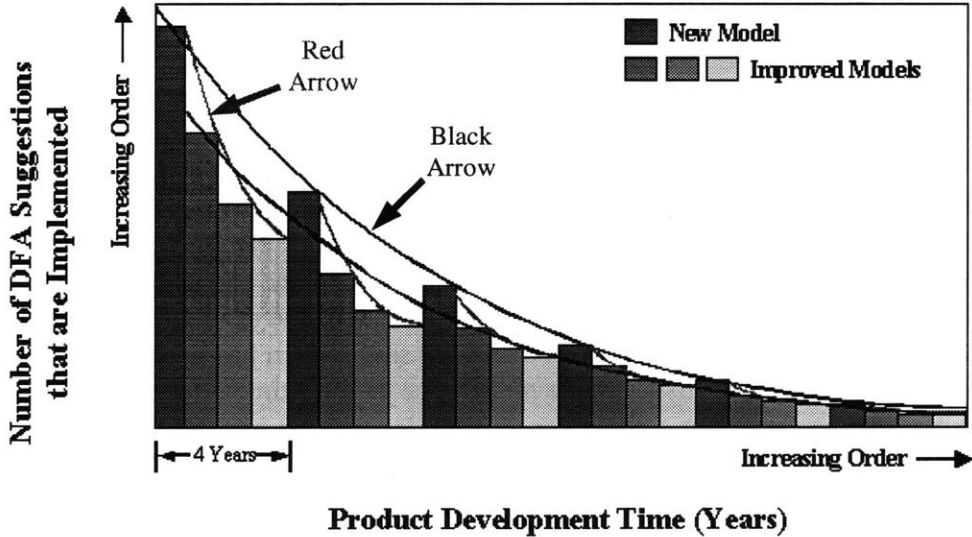
In almost all of the vehicle sub-systems that KBE-DFMA has analyzed, thus far, many of the parts, components and even sub-assemblies have been made by outside suppliers of Visteon. The success of each project is determined by the total number of Visteon-made parts, however; the team focuses its attention on these parts, knowing that it has the ability to make useful recommendations, which can be immediately used by the client.

KBE-DFMA does not have the capability to quickly change the way parts, components and sub-assemblies are made throughout the supply chain; therefore, it focuses on what it can help to change.

**Time Period for Evaluation:**

The success of any DFA analysis is strongly dependent on when the assessment is being done, relative to the life of the product. The most effective time to do it is when a product is going to be redesigned—not simply modified. At this stage in a product’s life, the designer has the ability to make extensive, and sometimes dramatic, design changes. Because the designer is starting with a “clean canvas,” he has a great deal of room to be creative. KBE-DFMA can also use this time to its advantage, by making design suggestions and seeing them implemented immediately into an actual product. In a sense, the team has to be aware of a product’s “DFA cycle.”

The “DFA cycle” coincides with the life span of a product, as can be shown in Figure 8.3. The figure tries to show when DFA suggestions can be effectively used throughout the course of a product’s lifetime—design, improvement and redesign. (Because the team has done only six projects, at this time, and does not have the resources or time to keep track of past projects, the figure simply shows how the “DFA cycle” could perhaps develop over time. A representative from the KBE-DFMA team acknowledged the reasoning behind this chart, however.)



**Figure 8.3: Effective Use of DFA Suggestions**

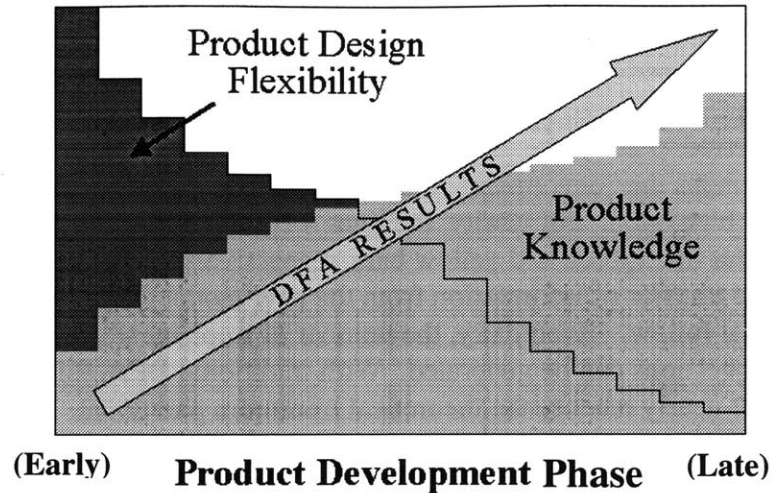
Each bar represents one year. Different colored bars are used to illustrate distinct stages of product evolution. (Refer to the key at the top of the figure.) The figure assumes a product life of 4 years for a typical vehicle sub-system, before redesign.

The graph shows two families of curves, both of which decay exponentially. (Exponential curves are used because they illustrate how humans learn over time.) Each set of curves is depicted in Figure 8.3 with black or red lines. The black lines show that knowledge is transferred to all future models. For example, two black lines are shown in Figure 8.3, one that maps the “new model” trend (red bars) and another that maps the “next model” trend (blue bars). For clarity, the lines were not shown for the two improved models after that (green and yellow bars, respectively). The red lines, on the other hand, show the transfer of information from the new model to each of the three improved models that follow. Eventually, the number of implemented suggestions will fall until no new suggestions can be generated. With traditional DFA, this happens when the design reaches or nearly reaches the theoretical minimum part count.

The purpose for these curves is to show that most improvements are made at the beginning of a product’s life, during design or redesign. During the time when a model is being completely redesigned, the designers and engineers have the ability to implement new ideas—to use new technologies and methods. They have the flexibility in their work to use DFA and DFM methods to reduce costs. However, as time goes by and designers and engineers try to find new ways to improve their existing model, they find themselves restricted to a limited design space, due to many of the parameters, which were initially set in place during the new-model stage. These parameters are based on the continuous improvement of the design for function, cost, and manufacture-ability, to name a few. This is one reason why KBE-DFMA is very selective with its projects; it wants to be sure that its recommendations can be used at a time when all types of improvements can be made to a product.

However, one very important problem arises when KBE-DFMA tries to analyze products during redesign: it does not have enough information about the products to make a good assessment. This shows one of the limitations of the BDI-DFA tool, and for that matter, all DFA tools. To provide the user with a detailed analysis, BDI-DFA requests specific parameters from the design. In a sense, DFA tools, like BDI-DFA, are used for checking a design after it has been developed rather than during the time it is being developed. No assembly advisor may be able to overcome this problem, but improvements can be made to these tools to allow them to incrementally move closer to the very beginning of concept design. This requires two things: (1) standardizing different parts of the design and the process for designing those parts, and (2) incorporating those standards, in the form of quantifiable information into knowledge base.

These issues bring out a very interesting paradox about design for assembly and BDI-DFA: how can suggestions be made, without having either enough product knowledge to make an assessment or enough flexibility to implement those suggestions. Figure 8.4 illustrates this matter of product knowledge versus product flexibility.



**Figure 8.4:** The DFA Dilemma

In the beginning stages of product development, there is a great deal of flexibility but not very much product knowledge. At this point, DFA is able to see its recommendations be used right away; however, it has no way of making these recommendations due to the limited amount of product knowledge presently available to it. As the project continues through time, a number of conditions are set in place, defining the design but also limiting the number of decisions that can be made about it. Near the end of the product development timeline, one can see that there is an enormous amount of product knowledge, which can be used for a DFA evaluation, but very little “room” to use the results from the evaluation to improve the developing design.

One way to avoid this conflict is to evaluate an existing model and to apply the results to a refreshed model. In many cases, refreshed models contain many of the best ideas used in past models, in addition to new, creative ideas, which may not be found elsewhere. KBE-DFMA can use this to its advantage. It can make recommendations beyond the part/component level and provide them to the clients, knowing that they will be used (or at least considered) in the near future, when the product is redesigned and made to be significantly improved. (This situation shows itself in Figure 8.3. Notice that the number of used recommendations increases significantly from new model design to new model design. Much of this is due to the fact that many ideas from existing product cycles are carried over to new product cycles.)

#### Face Time:

When assessing a design, a large percentage of the time is used to discuss possible ideas with designers, who know a great about the design. The designers know where changes can be made and where they cannot be made. (For example, they understand why it may be important, for a particular reason, to stay with screw fasteners rather than change to snap-tabs.) In a community like Ford or Visteon, it is rather difficult for the KBE-DFM/A team to keep in close contact with these designers; their services and experiences

are constantly needed in other areas. For this reason, the KBE-DFM/A has a challenging time trying to evaluate complicated assemblies that have, for instance, hundreds of parts.

In response to this dilemma, the author asked this question: If a majority of the time is used by KBE-DFMA trying to extract knowledge about a design, from a design team, then maybe it would be better to have the designers use the software tool? KBE-DFMA's role would be to show them how to use it. In this manner, the evaluation can be quicker and more useful to a designer or design team.

When this question was brought to the attention of the KBE-DFMA manager, he said simply that although designers have a great deal of experience in design, they are seldom aware of many of the processing issues that influence a design. Many of the members of the KBE-DFMA team have backgrounds in lean manufacturing, injection molding, and a variety of other process-related issues. Their expertise is imperative when evaluating any product with the BDI-DFA software.

However, this brings up another important question: if the software can only be used by people who understand manufacturing and assembly, why is KBE-DFMA planning to make an advisor, which includes BDI-DFA, for designers? The same manager mentioned that the process for developing the advisor comes in two parts: (1) "educating" designers about assembly and the software and (2) providing the designers with the tools to make the evaluations themselves.

Because of these external factors and limitations, KBE-DFMA's recommendations go no further than making changes from one type of fastener to another. Rarely do these suggestions considerably alter the design. As one member of the team mentioned to me, approximately 70% of most implemented suggestions deal with reducing the number of fasteners or substituting screws with snap-tabs. These types of suggestions can be used because they modify the tool (i.e., mold) only slightly and reduce the overall assembly time considerably. In general, suggestions that focus primarily on fastener types are safe to use because they do not affect the function of the product directly. In addition, they do not require the analyst to understand the product's function, except at the part level. (It is important to first understand why existing parts in a design are not joined together and made as one part before recommending part integration.)

## 8.6 Conclusions and Chapter Summary:

For now, KBE-DFMA is providing its clients with a service; it is using the Boothroyd-Dewhurst, Inc. suite to evaluate the manufacture-ability, assemble-ability, and service-ability of given vehicle sub-systems. The team is performing DFM and DFA evaluations on a number of products so that it can develop an assembly knowledge base. However, the difficult challenge is yet to come, when it goes through the process of developing the framework for a KBE assembly advisor.

The team has already acknowledged the fact that design for assembly is that more than direct-cost analysis and is now trying to find ways to include other aspects of it into their

work. What they may realize, from later customer requests, is that one assembly advisor may not be able to provide sufficient results for every type of application. The team may have to determine where segmentations exist in the world of assembly advisors, and what the drivers are for these divisions.

In the next and final chapter of this thesis, many of the ideas brought out from this work—the general methods and practices used in design for assembly, the typical framework and characteristics of a knowledge-based advisor, and the limitations of each in an industrial environment—will be used to understand the requirements for an assembly advisor. Questions like, what are the most important aspects of assembly that the advisor should have, what is a good plan of attack for making such an advisor, and what should it be able to accomplish in the world of assembly, will be answered.



# CHAPTER IX

---

## RECOMMENDATIONS AND REQUIREMENTS FOR AN ASSEMBLY ADVISOR

### 9.1 Introduction:

Chapter IX is the final chapter of this thesis. It will contain information that will prove useful to interested readers who plan to develop an assembly advisor. The chapter will list several recommendations and requirements using much of the information provided throughout the thesis. This chapter will not present the user with a list of detailed instructions for creating an assembly advisor; what it will provide, however, is a set of critical requirements that many books, papers and journal articles concerning design for assembly do not consider. Many of the guidelines are written in a general manner so that the most widespread conditions are met. It is up to the user to consider these guidelines in his design environment.

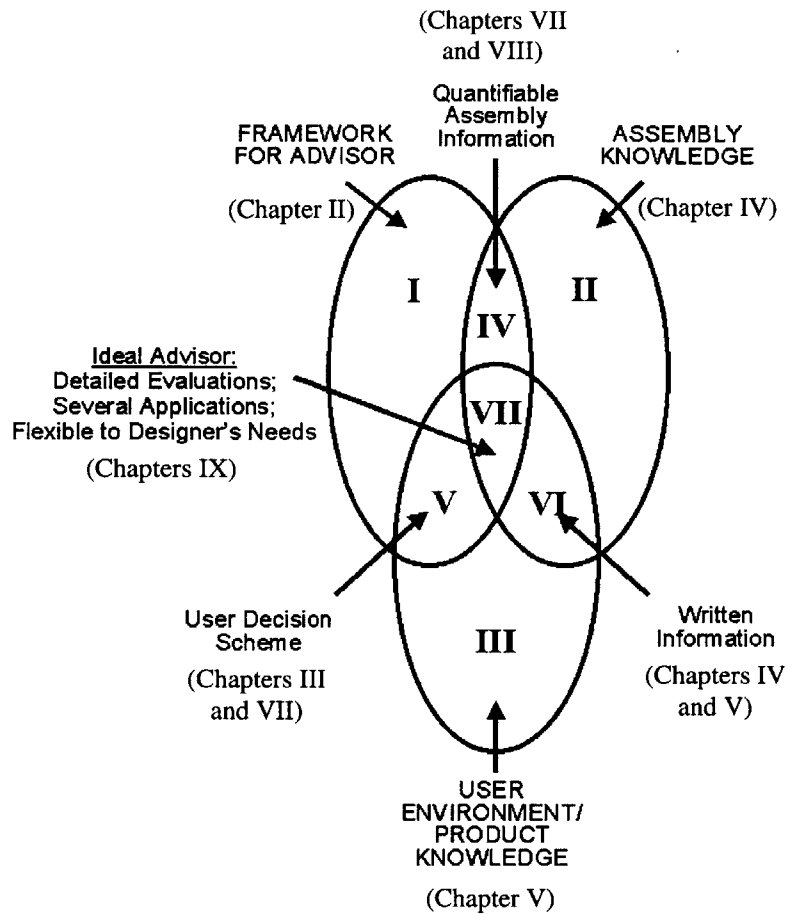
This chapter will be broken down into three sections:

1. A brief summary of the previous eight chapters for the purpose of understanding how they all contribute to the development of this chapter.
2. A list of requirements and suggestions for the development of an assembly advisor (as just mentioned in the introductory paragraph).
3. A list of suggestions for possible research projects relevant to this work.

### 9.2 A Brief Summary of the Previous Eight Chapters:

Collectively, Chapters I through VIII provided information regarding existing design practices; limitations with expert advisors and their development/deployment, particularly within a commercial environment; limitations with existing design-for-assembly theories; and general challenges that exist with the acquisition and management of knowledge in a design environment, to name a few. These chapters touched on three basic points that are pivotal to an assembly advisor: (1) the basic framework for the

advisor, (2) the representation of assembly knowledge in the advisor's knowledge base, and (3) the necessary requirements for user interaction in a design environment. Although these points are equally important from an individual standpoint, the reader should not dismiss their overlapping responsibilities. These interactions help distinguish assembly advisors from other types of advisors, and also show what areas of development need to be considered. The following figure (Figure 9.1) help to identify these interactions.



**Figure 9.1: Key Aspects of an Assembly Advisor and Their Overlapping Responsibilities**

The three key regions of this figure (Regions I, II, and III) were described explicitly in Chapters II, IV, and V—General KBE Theory and Methods, General DFA Objectives, and Assembly Considerations within Ford/Visteon, respectively. Their interactions (Regions IV, V, VI, and VII) were discussed mostly from the perspective of Visteon/KBE, in Chapters III, VII, and VIII—Visteon/KBE Methods, KBE Tradeoffs and Pitfalls, and DFA/KBE Tradeoffs and Pitfalls, respectively. The remaining part of this section will discuss this figure in more detail by explaining the importance of each of these seven regions. The figure above and the description below will also provide a useful summary of the thesis, at this point.

### I. Framework for Advisor:

The framework of a typical advisor is built around two knowledge-base components that interact with one another. The first component contains the “raw” information (e.g., equations, assumptions, numeric values for given conditions, etc.) necessary for analysis, while the second component contains the decision-making process, necessary for attacking a design problem in a systematic way. The two components “communicate” with each other when tradeoffs have to be made.

An “expert” advisor tries to replicate human thinking and understanding, but is limited to only certain types of applications—ones that can be described explicitly with a decision-tree structure and analyzed using only quantifiable data. (The reader may notice that these restrictions are illustrated in Figure 9.1 as the overlapping regions that exist between “Framework for Advisor” and its two neighboring aspects—“Assembly Knowledge” and “User Environment/Product Knowledge”.)

Decision-tree structures (ones that use if-then statements to set up a decision) limit themselves to only certain types of processes. They have to be highly sequential in structure; and be bounded to a given set of stipulations. Furthermore, they can never change over time; logic behind the process has to be made standard. The tree structure should be applied to processes that are well known and understood and do not require the use of an expert to be on hand. The processes should be very methodical and predictable.

### II. Assembly Knowledge:

Assembly knowledge comes in a variety of forms—from the subjective rules that designers refer to while working on a project to the equations and charts that they use to assess the assemble-ability of a design. (Refer to Regions IV and VI) Most assembly information, however, is process dependent. In other words, it focuses on the process for assembling a given product, but not the product itself. As a result, many of the existing DFA methods require extensive information from the user so that each evaluation is customized for different product types. That is why “Product Knowledge” is considered separately from “Assembly Knowledge” and placed in the same category with “User Environment”.

### III. User Environment/Product Knowledge:

The user environment is probably the most complicated aspect to consider during development of an assembly advisor. It encompasses all of the decisions necessary for designing a complete product or part of a product (e.g., functional characteristics; fabrication and assembly requirements; cost for design, manufacturing and assembly; existing materials, tools and parts; etc.). The decision process, in most cases, is very elaborate, requiring intuitive understanding of a variety of issues in a few simultaneous steps; rarely can it be completely encapsulated with a decision-tree structure. For a specific part of a process, which is consistent and repeatable, the inquisitive reasoning behind it can be recorded and placed inside an inference engine for use in an expert advisor. (See Region V)

The author is in no way suggesting that the intended assembly advisor be able to extract every piece of information that is confined to the user environment. However, the advisor should be developed in such a way that it “understands” its place in product development. If, for instance, packaging is the most important issue in a design, then the advisor should “know” what type of suggestions it can recommend. This issue was brought up particularly in Section 4.5: Motivation for Placing Assembly Knowledge in a Hierarchical Structure and Its Role in “Design Space”. An “Ideal” assembly advisor would be able to recognize how its suggestions can be used in a design without changing its primary functions, but this is not to be expected from an advisor at this time.

#### IV. Quantifiable Assembly Information:

In terms of design for assembly, quantifiable information means any information, which can help to make a comparison about different assembly designs. Chapter VIII spoke about some of the existing methods for evaluating designs on the basis of assembly (e.g., time and cost evaluations, variable- and fixed-point systems, etc.) The variable point system allows the user to develop a precedence relationship between various DFA rules, by pointing out which rules are the most important in his design and which ones are not. Before making an evaluation, the designer normally ranks each rule accordingly and enters the information into the advisor. The rules are then used to evaluate each part for assembly; at the end, an assemble-ability score is tabulated for the design, in question. The fixed point system, on the other hand, does not provide the user with this capability. The ranking system is fixed by the developers of the tool for all product types. Currently, these methods provide the only means for transferring assembly knowledge to the level of an advisor.

#### V. User Decision Scheme:

As briefly mentioned earlier, the user decision scheme illustrates the decision-making process for particular parts of design work in an explicit manner. Only those decision schemes, which can be drawn out in their entirety, encompassing all the important aspects of each process, can make their way from “User Environment” to “Framework for Advisor”. At this time, they are unable to consider non-linear (i.e., loosely coupled cause/effect) relationships; work should be done to consider these types of situations, perhaps using some of the techniques commonly found in the theory of system dynamics. This is especially important in the development of an assembly advisor because many of the cause-effect relationships, which affect assembly, are not always apparent. (Reasons for this were described in Chapter I.)

#### VI. Written Information:

In Chapter V, the author spoke about how Ford and Visteon capture component-level knowledge, by placing common rules for design; basic steps that have and have not worked in the past; and formulas and data sheets, defining specific performance characteristics into written manuals. Most of the information is written in a subjective way—meaning that the user has to rely on his knowledge and past experiences to apply the information to individual projects. What these guidelines do not capture is the type

of system-level thinking required to properly align “Assembly Knowledge” with “User Environment”.

#### VII. “Ideal” Advisor:

The “ideal” assembly advisor would be able to consider all three aspects of development, (i.e., framework, assembly knowledge, and environment) in a collective manner. It would provide the user with detailed suggestions for a variety of products. By merely providing suggestions, the advisor would allow the user to make decisions without necessarily inhibiting creativity. These three issues—generating detailed suggestions; analyzing a variety of products, with different assembly architectures; and providing the user with insight as well as flexibility—work against one another, in many respects. These conflicting characteristics all fall within the category of breadth versus depth (which was discussed in Section 8.2). The reasons for this are also summarized below:

- Commercial DFA tools assess each design using general assembly principles. This allows a wide variety of products to be analyzed but requires the expertise of the designer to see suggestions implemented in every case. (The tools do not contain information about the products in their database; if they did, they would be required to have information about all products.) As a result, the tools are very flexible but do not provide any deep insight into specific applications. (FLEXIBLE BUT GENERAL)
- Expert advisors evaluate specific aspects of a particular type of product. They have within their knowledge base information about the products and a logic structure for converging to specific results. However, the logic structure does not provide the system with the flexibility to evaluate various product types. (All products are designed with a different set of conditions in mind.) As a result, the user is able to gain specific insight from the tool, but only for certain product types. (SPECIFIC BUT INFLEXIBLE)

The next section will deal with different ways to at least reduce these conflicts (if complete synergy is not a realistic option).

### 9.3 Recommendations and Requirements:

This section will describe what key characteristics are important in the development of an assembly advisor that falls within Region VII of Figure 9.1. For clarity, the section has been broken down into sub-sections—Key Characteristics for an Assembly Advisor; The Need for Product Knowledge in an Assembly Advisor; Knowledge-Base that Complements Design for Assembly; User Requirements in a Design Environment; Ways to Link System-Level Design with Part-Centric Design using an Assembly Advisor; Storing and Managing Information for Later Use; and Illustrating the Decision-Making Process for Design for Assembly.

#### 9.3.1 Key Characteristics for an Assembly Advisor:

This section provides a list of key characteristics (essentially the requirements) for an assembly advisor. All but the last two suggestions are considered feasible for today's KBE developers. Many of the items here will be discussed in more detail in the following sub-sections.

- An assembly advisor should provide suggestions that are more than generic. Many DFA tools today not “fill the gap” between providing a suggestion and implementing a suggestion. The designer, in most cases, is the one who does this. If an assembly advisor is to ever get acceptance in the design community, it must provide useful information. However, for the advisor to give deeper recommendations, requires that it know more about the products that it analyzes. (The following items help to explain ways to do this.)
- An assembly advisor should analyze different product types using a different set of assembly principles for each case. The key to differentiating products is by determining what type of assembly architecture they have (e.g., “parts in a box” like an HVAC unit, links like a wiper module, arrays like PC boards, etc.). Different assembly principles apply to each type of architecture. For instance, top down assembly may be easier to apply with arrays than with “parts in a box;” therefore, the advisor should enforce this rule more so in the former than in the latter architecture type.
- An assembly advisor should evaluate assemblies beyond the part and component level. By definition, assembly deals with the collection of all parts in a system, not just the interaction of a few parts. An advisor that deals with product architecture, assembly sequences, and mating surfaces of key parts (like the housing for the HVAC unit) encourages designers to use top-down concepts. By thinking in this way, designers can anticipate assembly problems before they even reach the factory floor.
- An assembly advisor should identify where part consolidation and separation is most useful and perhaps needed in a design. (Refer to Section 4.8) In other words, it should help the designer know where to utilize integral design and modular design. To do this will require the use of a function map, a “tool” which decomposes a system into its essential functions and sub-functions and traces the flow of material, information and energy throughout the system. Function maps should also be used to examine product portfolios; this allows the designer to move away from a “one product at a time” methodology, and to see what choices are beneficial for the entire product family. (Refer to Section 4.4)
- An assembly advisor should rate the importance of each function identified in the function structure. The designer can then assess the importance of each rated function in order to determine which ones need to remain and which ones need to be eliminated or possibly coupled with other functions.

- An assembly advisor should keep track of all the different combinatorial designs, if modularity is a key consideration. (For example, Ford drive shafts are highly modular; parts can be mixed and matched to produce over one-hundred different types of the product.)
- An assembly advisor should consider the best sequence of assembly for a given design and determine how the specified sequence relates to the finer points of DFA—part insertion, orientation and handling, to name a few. The assembly sequence module can be used as an interactive rule checker for part- and component-level guidelines. The module, for example, can alert the designer to excessive product orientations and elaborate insertion directions.
- An assembly advisor should contain and utilize general information about specific assembly equipment (tools and fixtures) and assembly ergonomics. For example, the advisor should warn the designer when a designed part is too heavy to be lifted by an assembly worker, where a “blind” assembly has the potential to exist, and where a specific fixture is needed. An advisor with this capability really pushes the designer to know more about the environment in which his product will be assembled.
- An assembly advisor should have in its knowledge base a general hierarchical structure for organizing assembly knowledge, something similar to what was shown in Figure 4.5. (Refer also to page 56) This knowledge structure should be accessible to the designer, in perhaps pictorial form, so that he can see what aspects are important in his design. The structure should also be used as an interactive learning tool to assess where past emphasis on design for assembly had been placed. (The structure can be used, after the evaluation, to determine how many suggestions were considered and utilized at each level of DFA knowledge.)
- An assembly advisor should identify conflicting DFA rules that are used in an evaluation, and should alert the designer to them. (Refer to Section 4.4) An example of two rules that conflict in reasoning with one another are “design parts so that they are symmetric” and “design parts so that they are asymmetric.” The first rule helps to reduce the number of handling procedures for a part before it makes its way into the assembly; while the second rule helps to eliminate improper part connections. When to use either rule, however, depends on the nature of the product. If such a conflict is recognized, the advisor should request further instructions from the designer on what to do.
- An assembly advisor should identify surfaces that are used purely for mating two or more parts together and those that have a functional importance. (This becomes relatively important especially for package-driven items, like HVAC units, where the split planes can dictate the way an assembly is arranged. In the case of the Ford CDW-27 and Toyota Camry, the split planes are arranged in such a way that a fixture is needed to assemble the main doors to the casing halves. When designing products,

like the HVAC, emphasis should be placed on the orientation of the split planes, and the advisor should be alert to this.)

- An assembly advisor should allow the user to have the ability to update assembly and product information that is stored in knowledge base and to restructure the precedence relationships of some rules, if necessary. This allows the advisor to be flexible to the designer's needs and to the "needs" of the product design. It also makes the advisor applicable to different product evaluations.
- An assembly advisor should act as interactive-learning tool, as well as an evaluation tool, for designers. Designers should be able to access information concerning the advisor's reasoning and decision-making abilities so that he can learn from the tool. (It is also important for the "expert" advisor to have retrieving capabilities like this so that the designer can question the end result, if necessary.)
- An assembly advisor should keep track of all decisions made and allow the user to see those decisions (and perhaps possible outcomes) on a real-time basis (i.e., as he is going through the evaluation study). This makes the designer aware of his thought process, and allows him to see if certain decisions have brought him to an earlier set of steps (cyclical decision structures). In addition, this will allow the user to see information that is not normally found in the design manuals. (Refer to Section 5.5)
- An assembly advisor should collect information regarding past evaluations (e.g., the products, end results, and decision process) and make it retrievable to the user by typing in key parameters that reflect the current evaluation. A summary chart should also be included providing a list of recommendations (and reasons why they were used and not used) on past products with a similar architecture type.
- An assembly advisor should allow designers the freedom to be creative in their work but also prevent them from making serious design mistakes, with regard to assembly. The advisor should recognize what assembly attributes need to be maintained in a design and which ones are more or less optional. Part creativity should be allowed but only when part sharing or reuse is not an optional route to take. (This eliminates unnecessary part variety.)
- An assembly advisor should be used as a tool to help designers work more closely with industrial engineers, manufacturing engineers, and other technical experts. The advisor should be used to allow designers to know more about the conditions on the assembly floor (e.g., equipment type, assembly layout, etc.) and to allow the engineers to see the current development of the design so that quicker recommendations can be made.
- An assembly advisor should show how parts are constrained by other parts, and alert the designer of any constraint loops that exist in the assembly. (Constraint loops should be avoided because they require extremely accurate assembly equipment and



procedures. For example, in the Windstar HVAC unit, the housing assembly located the air collimator. If the housing is not accurately assembled, then the air collimator may not assemble correctly to the housing. The unit that the author received did actually show the air collimator lifting away in certain regions from the housing.)

- An assembly advisor should avoid the use of decision tree structures. As was mentioned in the introductory chapter and in several other chapters that followed, assembly is non-linear (cause and effect are not necessarily placed one after the other) and highly dependent on “outside” attributes, like packaging, performance, materials, manufacturing, assembly equipment, etc. (The latter point was brought out in Section 4.5, when suggesting a possible redesign of the CDW-27.)
- An assembly advisor should be able to “read” information from CAD models. Andreasen et. al. in Design for Assembly, 2<sup>nd</sup> Ed. say that “[i]f we could allow the expert system to interrogate the CAD product description directly (in the system around 80% of rule conditions may be satisfied by geometry alone), its performance and reliability might be much improved” over direct user input. (Pg. 164)
- An assembly advisor should consider cost in its evaluation, but from the perspective of both assembly and manufacturing. (Refer to Section 8.3) By considering it purely from an assembly perspective, suggestions tend to emphasize part consolidation without considering its impact on tooling costs.
- An assembly advisor should provide the designer with drawings of the entire assembly (i.e., exploded view drawings or models). The designer then can then see how his decisions affect the models and the assembly build.
- An “ideal” assembly advisor should be able to recognize how its suggestions can be used in a design without changing its primary functions. It would understand how its suggestions affect the product down to form of the individual parts.
- An “ideal” assembly advisor should be able to read in information from other advisors so that decisions are not just from the point of view of assembly. This would require having expert systems that can interact with one another and to be able to restructure their own decision structures, neither of which traditional expert systems are able to do.

### 9.3.2. The Need for Product Knowledge in an Assembly Advisor:

One of the most important ideas to come out of this thesis is that assembly cannot be done in isolation, nor can it be viewed in the same way for different products. (Refer to Section 2.6.2) Unlike other processes, such as injection molding, die-cast molding, and sheet metal forming, design for assembly requires the analysis of all parts, individually and collectively, which make up a product or a piece of a product. Most manufacturing processes, like the ones just mentioned, are confined to individual parts; they are not entangled with the complexities of part-to-part interaction and assembly dynamics (e.g.,

product orientation throughout the assembly process), to name a few. Visteon/KBE has already witnessed this occurrence with its existing process advisors—many of which look at individual manufacturing procedures. The Department is now seeing some of the challenging aspects of assembly as it begins development of an assembly advisor.

Several assembly analysis tools exist today (many of which are sold commercially) for the purpose of evaluating different aspects of assembly. These tools provide the user with a virtual simulation of an assembly process, a general evaluation of an assembly (using a point system or a cost/time study), a connectivity map (i.e., a liaison or datum flow chain diagram), etc. (Refer to Appendix) However, none of these tools help to differentiate different assembly architectures—one of the key characteristics for an “ideal” assembly advisor.<sup>80</sup> In the case study provided in Chapter VII, the author described how KBE advisors must be made to accommodate different product types (e.g., mechanisms like wiper modules and window actuators; “parts-in-a box” like HVAC units and automobiles; and arrays like PC boards and keypads, to name a few). (Refer to Section 7.5.3 and Section 8.6) An “ideal” assembly advisor should “recognize” this dichotomy, and furthermore, “understand” how assembly differs with each product type.

Developing an assembly advisor in this fashion requires that the advisor have two dimensions to it—a process advisor and a product-specific advisor. (KBE does not have an advisor within its portfolio that has this property. Refer to Table 3.1 for details. Such an advisor may prove to be a challenge for the Department, at least at this time.) The reader should note that a “product-specific advisor” as mentioned here, is an advisor which goes no further than “understanding” the differences between various assembly architectures; it is not intended to analyze the detailed geometry of a product. This, in fact, would be a turnaround from what traditional assembly analysis tools do—analyze the assemble-ability of a product from the part and component levels and let the user deal with issues related to assembly sequence and product architecture. (Refer to Section 8.2)

### 9.3.3 User Requirements in a Design Environment:

This sub-section will look at user requirements from three different perspectives: Providing the Right Amount of User Flexibility; Identifying the User’s Design Space within the Assembly Advisor Environment; and Encouraging Concurrent Engineering Practices with the Use of an Assembly Advisor.

#### Providing the Right Amount of User Flexibility:

---

<sup>80</sup> KBE is trying to develop an assembly advisor tool by linking commercial assembly tools together and to the CAD system that the advisor is going to run on. This may not be the best way to develop an assembly advisor, however. As was mentioned in Chapter III, this is considered a bottom-up approach to the design and development of an advisor: the advisor is developed without information regarding existing design practices at Visteon or different types of products. Of course, the reader has to understand that KBE has a limited number of resources and a demanding schedule for each advisor project. Developing an advisor in the top-down manner would require an extensive understanding about several of the points mentioned in Section 9.2.

Generally speaking, designers and engineers seek advisors that provide detailed suggestions, which reflect the design in question. An advisor that evaluates rather than regenerates a product is preferred because the former advisor provokes a greater degree of user interaction and designer creativity. However, it should be pointed out that designers invite the idea of product regeneration if it means having the advisor perform a trivial task that is very methodical in nature. The reader should note that engineers primarily evaluate the feasibility of a design and deliver their results to the designers. As a result, engineers would only need an evaluation tool for their type of work.

For novice designers, less flexibility and more automation (i.e., regeneration, parametric modeling, etc.) is needed from the advisor. This may mean that an entirely different advisor is needed for the novice user—one that offsets the requirements set by experienced designers; or it may mean that an “adjustable” advisor is needed—one that allows certain processes to be done by the user or by the advisor, depending on the user’s level of experience. (Refer to Section 7.5) However, for advisors that are highly generative and inflexible, it may be important to include a feature that validates the advisor’s reasoning to the user; such a feature will allow for faster reassurance from the user about the advisor’s capabilities. (Refer to Mycin case study in Chapter II.)

#### Identifying the User’s Design Space within the Assembly Advisor Environment:

Identifying the user’s design space is a very important aspect to design; it allows the user to understand how various functional requirements and processing limitations interact together and come out geometrically in the design. In their book Design for Assembly, 2<sup>nd</sup> Ed., M.M. Andreasen, S. Kähler and T. Lund state that that “a component is completely determined by specifying form, material, dimensions, surface quality and tolerances.” (pg. 131) And for the most part, designers, especially at the concept level, are interested in the form of the components, namely the surface configurations. Design space, in a sense, can be identified by this attribute. The designer has the ability to manipulate surfaces (or parts of surfaces) that are not constrained by function or by other means. Andreasen et. al. refers to these as “free surfaces.” (The authors also mention “connecting surfaces,” surfaces that allow part-to-part mating, and “assembly surfaces,” surfaces that are used to transport and handle parts.) Design space is essential to the successful deployment of design for assembly. (This issue was mentioned in Chapter VIII—KBE/DFA Tradeoffs and Pitfalls.) The question is how can an assembly advisor illustrate reasons for why a product is designed in the manner that it shown? The author provides some suggestions below:

- A set of color-coded features, parts, and components that reflect the various reasons for the product’s initial construction (e.g., shape and size).
- A connectivity map that shows how different product and process requirements influence individual parameters of a design.

Encouraging Concurrent Engineering Practices with the Use of an Assembly Advisor: Even with the influences of lean manufacturing and world-class production, many product-development companies, like Ford and Visteon, still have difficulties aligning

design with manufacturing and assembly. (Refer to Section 5.3) This idea was mentioned all throughout Chapter V, with quotations from a manufacturing engineer and an industrial engineer at Visteon's Sheldon Road Facility. Engineers react to a designer's work by determining its feasibility from both a manufacturing and an assembly standpoint; the results from the evaluation are later presented to the designer. The process is rather slow and inconsistent, and as a result, many problems are identified during actual production.

To help eliminate this dilemma, the author suggests that the assembly advisor be developed in such a way so that both designers and manufacturing/industrial engineers can use it. In particular, it should provide direct Internet access to the both groups, allowing up-to-date, "interactive" information to be transferred from one computer system to the next. (Refer to Section 5.3) (The information that is sent should be able read through a shared medium. Designers use different computer tools than engineers; as a result, the certain files may not be compatible with each system.) Two examples are provided below:

- The designer can submit a "simplified" version of his CAD model (i.e., the entire assembly in exploded view) to the engineers, allowing them to move parts along their intended line of action, to rotate the entire model, and to enlarge any aspect of it. The engineers can then submit their ideas to the designers, by placing text boxes on each aspect of the design that they have concern with. (A two dimensional drawing does not provide the same kind of substance as a three-dimensional model with interactive features and capabilities.)
- An "interactive" floor plan can be used by both groups (designers and engineers) to show how a design (even before it is complete) can be set up for actual assembly. This tool can show the position of the assembly and testing equipment and workers, the state of the product as it goes through each assembly operation, the time for each operation, the number of orientations through the entire process, the flow of materials into the process, etc. The designer can make revisions to the design based on what he sees from this set-up and from the industrial engineers suggestions. In addition, the industrial engineer is more prepared for set up as the design stage comes to an end.

#### **9.3.4 Ways to Link System-Level Design with Part-Centric Design using an Assembly Advisor:**

The author points out three ways in which part-centric design can be linked with system-level design: (1) by using functionality maps to determine where part separation and consolidation is important in a design; (2) by encouraging the earlier use of exploded-view models; and (3) by using the assembly sequence analysis tool, developed by D. Whitney, et. al., to evaluate all part- and component-level assembly issues.

##### **1. Function Maps:**

Many assembly evaluation tools that exist today, like BDI-DFA™, “ask” the user to identify common functions across a product for the purpose of applying part-consolidation theory. Current DFA methods do not provide a systematic approach to part integration; it is the responsibility of the user to know where the theory can be applied. The author suggests that these tools be used in conjunction with some type of function-map generator/editor. Function maps track the flow of energy, information and material through a given product design. They also show where part dislocation, integration, modularization and standardization are key to the design.<sup>81</sup> This can be done relatively early in the design process; it also provides the user with an organizational approach to design for assembly. (Refer to Section 4.8.5 for details and examples.) (K. Otto, a professor at MIT—Center for Innovation in Product Development, does research in this area of product design.)

2. Exploded-View Drawings:

In most cases, designers at Ford/Visteon do not develop exploded-view drawings of their designs. (The author asked the manager of the KBE-DFMA section if he had seen any designer at Visteon create sketches in this manner to help him with his work; his answer was no.) Even rough sketches can be helpful to designers for a variety of reasons. First, it can allow designers to consider each design as a whole, not just from the point of view of individual parts and components. Second, designers can immediately see where insertion and handling difficulties may arise. Third, it can provide the designer with key information (e.g., precedence relationships) for determining the possible sequence(s) for assembly. Fourth, exploded-view drawings of different products within a product family can be looked at and compared for the best design-for-assembly practices. (The author’s suggestions for the design of HVAC units were based mostly on the four exploded-view drawings shown in Chapter VI.) Fifth, exploded-view drawings can help designers determine what characteristics differentiate products within a product family, from the standpoint of assembly.

3. Assembly Sequence Analysis Tool:

Many of the existing rules for design for assembly can be considered within the assembly sequence analysis tool developed by D. Whitney, et. al. (Chapter VIII briefly describes this tool.) As was mentioned in Chapter IV, many of these design rules are based purely on fine- and gross-motion assembly applications (e.g., part location, insertion and handling). The analysis tool can be used to formally link part- and component-level design with actual assembly design. For instance, it can tell the user the fewest number of orientations that a given design will go through, during actual assembly. (However, the number of orientations for an assembly may not be as critical as the “placement” of these orientations during actual assembly. For example, assemblies that can be easily held by a

---

<sup>81</sup> Functionality maps can be used to identify common flows across different product designs; this can allow the user to see where part sharing, reuse, and standardization can be effectively applied. Using the functionality maps in this way can undoubtedly promote more system-level thinking

worker are immune to various orientations. What the analysis tool should consider is the weight, size and general shape of the assembly as a function of assembly operations.) The tool can be used to determine the direction of parts as they enter the assembly, relative to the assembly's position in space. It can also be used to evaluate different assembly sequences for optimal cost and time, bringing another dimension to design for assembly. (Traditional DFA methods and tools evaluate the time and cost for a single assembly sequence.)

Not only does the assembly sequence analysis tool provide a way to realistically use general DFA guidelines, it also provides the user with a way to evaluate different product types, from the perspective of assembly. The user can look at a variety of issues—from assembly time and cost to the placement of orientations throughout the assembly-sequence chain—to determine which of these has the most impact on design. Using the tool for several products can provide the user with an understanding of design for assembly from the perspective of assembly architecture and assembly sequence and their relationship with part/component insertion, orientation and handling.

### 9.3.5 Storing and Managing Information for Later Use:

The advisor should keep track of all suggestions made about past product evaluations as well as the key parameters, which describe the product and its architecture. In addition, the advisor should keep track of the decisions made for coming to different results, and they should be represented in some type of graphical manner, perhaps with decision maps. Most of the information stored in manuals does not provide this type of content. (That is the reason why KBE systems take so long to develop; they are based on “raw” information (e.g., equations, rules, etc.) and a logic base.) Having this information readily available can also allow KBE teams to use this knowledge to create other systems.

As was mentioned in Chapter II, expert systems are unable to learn from past evaluations; what they are able to do, however, is store huge amounts of information. This can be considered a benefit to the user. Key results from past product evaluations can be stored in the advisor's knowledge base for later use; with the help of a search engine, the user can look for past designs with similar assembly architectures and see what was done in those cases. By bridging the gap between stored knowledge and present applications, the user completes the learning cycle; he finds ways to apply the advisor's knowledge into his own projects. Because the information is presented in a way that is organized and linked to product types, the user quickly understands how assembly influences products in different ways.

### 9.3.6 Illustrating the Decision-Making Process for Design for Assembly

KBE might find it difficult to develop a universal decision tree for an assembly advisor. There are two reasons for this: (1) assembly, as mentioned in Chapter I, is non-linear and, at times, counterintuitive in nature; and (2) assembly is so much apart of each product, in

question, and the process for designing and fabricating it.<sup>82</sup> To capture these ideas, the author suggests using many of the techniques from system-dynamics theory; the theory can help designers better understand the consequences that emerge, no matter how subtle, from individually made decisions. Of course, the next step would be to compare different cause-effect diagrams with each other to see where there are common inferences.

## 9.4 Potential Research Projects:

This section will discuss some of the potential research projects that would accentuate what has already been said in this thesis. They include System-Level Knowledge in a Design Environment; Package Space and Its Impact on Design and Assembly; Part/Component Sharing and Reuse and Their Impact on Assembly Architecture; Design for Assembly and Manufacturing; and KBE Method for Knowledge Acquisition and Advisor Development.

### 9.4.1 System-Level Knowledge in a Design Environment:

#### Design-for-Assembly Rules:

Design-for-assembly rules are context free so that the rules can be applied to different assembly applications. Designers apply these rules in the context of their own work; designers use their own knowledge about product design to incorporate these rules to specific projects. Research should be done to capture designer knowledge—the knowledge that designers use to apply general assembly rules to their designs. Chapter IV showed that there is a hierarchy to what rules exist already, but that was shown from the point of view of product design (i.e., each level of a product, from the individual parts to the architecture). More has to be done in this area, of course, as was mentioned in that chapter. However, what the author describes here is another type of hierarchical ladder, which needs to be developed—one that shows the different levels of general and specific design rules.

**Identifying What Type of System-Level Information Currently Exists in Design Manuals:** Chapter V showed how current knowledge from the Instrument Panel and Console design manuals were used; an array was developed, showing the connection between the types of rules found in the manuals and the different parts of the sub-system. The author recommends that this method be followed for all design manuals. To do this, however, requires that all written rules be placed into different design categories, such as assembly, manufacturing, packaging, serviceability, etc. The purpose of this is to understand what drives the design of different sub-systems (if there is a direct correlation between what designers say and what the manuals suggest) and what types of information are normally documented. The information that emerges from this array can also provide designers with a better idea of what needs to be emphasized in design and in later manuals.

---

<sup>82</sup> The second reason helps to explain why assembly analysis tools that use a fixed point system are not conducive for all types of assemblies. Tools that allow the user to determine which assembly rules have a higher precedence over others have wider range of applicability. In a sense, such a tool would allow the user to alter the decision tree structure from application to application.

### 9.4.2 Package Space and Its Impact on Design and Assembly:

Packaging plays a significant part in the design of most vehicle sub-systems. As mentioned in Chapter V, packaging also influences the way products are designed for assembly. The details behind packaging are rather complicated and dynamic in nature (i.e., packaging conditions for different vehicle sub-systems can directly or indirectly affect one another and can change over the course of a project design). In fact, most design time is spent understanding these conditions and their impact on different vehicle sub-system designs.

What can be done to understand the non-linear, system-dynamic attributes of package space in a vehicle design? The author suggests that the “vehicle freezing” process be looked at over the course of a project design. By following this process through, one could see what aspects of a typical vehicle are frozen first, second, third, etc.; how the “freezing” process affects the design of different sub-systems; and ultimately how packaging influences design for assembly. By identifying which vehicle attributes are frozen over time, the evaluator can then see which attributes drive other attributes and how, and then determine if the order of freezing is arranged correctly.

The evaluator may even find that this process is not necessarily the best way to develop packaging regions for each vehicle sub-system. It may also help to see if package space can be fixed for most of these sub-systems, allowing more components to be made standard. For example, standard packaging space can eliminate the need for 40+ unique HVAC casings; the casings can be designed to just satisfy all of the necessary requirements for heating, air and cooling. In addition, less time will be needed to develop the sub-systems because all of the requirements have already been set at the very beginning. In fact, Ford may be able to share not only parts and components across different vehicles but whole sub-systems as well.

### 9.4.3 Part/Component Sharing and Reuse and Their Impact on Assembly Architecture:

Part/Component sharing and reuse play important roles in the design, and ultimately the assembly, of a vehicle sub-system. However, neither of these ideas have been fully exploited at Ford/Visteon. (The Instrument Panel and Console Organization has recently developed a team of designers and engineers to where more part sharing and reuse can be done; the team is also trying to reduce the number of unique part designs.) The author provides the following set of procedures for identifying common and unique parts across products within a product portfolio.

1. Identify all of the different components commonly found in a particular sub-system, such as an HVAC unit.
2. Identify all of the different model types for a given year.
3. Develop a matrix listing the common components on one side and the model types on the other side.



4. In each cell, note the component model for each designated sub-system model.

After this is done, the evaluator should then note the key parameters (e.g., physical properties, performance characteristics, geometric characteristics, etc.) that differentiate each component model from one another. Identifying these characteristics, no matter how trivial, and making note of them in the matrix can allow one to determine where standards should be set. Ultimately, an interactive database can be developed, which allows designers to find existing components for new projects simply by entering in certain key parameters.

As a note, a third dimension can be added to this matrix—the model year for the vehicle sub-system designs. The evaluator can then see how the part sharing and reuse was carried through over the course of a few years.

#### **9.4.4 Design for Manufacturing and Assembly:**

At this time, there is no real synergy between design for manufacturing, which says, “Design parts for easy manufacturing,” and design for assembly, which says, “Design parts for easy assembly.” For example, BDI-DFMA™ analyzes a product from the standpoint of assembly and manufacturing; however, the tool does not consider these two collectively. It is up to the designer to see how to balance the cost for manufacturing with the cost of assembly by reconfiguring the design. This usually requires the designer to evaluate his design in an iterative manner until he is satisfied with the results from both evaluations. An interesting research project would be to see how DFA and DFM “interact” with one another, by developing a set of time and cost metrics that consider both methods. In a sense, this project would help to determine where global minimization exists along the manufacturing/assembly-cost curve. Such results would help to answer the classic question in design for manufacturing and assembly, which is better, part differentiation or part consolidation?

#### **9.4.5 KBE Method for Knowledge Acquisition and Advisor Development:**

**Evaluating the Entire Visteon/KBE Process:**

KBE has difficulties developing customer-focused advisors in a timely fashion for a variety of reasons, which were mentioned in Chapter VII. The author suggests a research project that focuses, in detail, on how KBE approaches each project in its entirety. The project would pay attention to each aspect of a project, determining where the most time is spent and why. Such a research project may tap into some of the reasons why the acquisition and management of knowledge are both challenging processes, especially for any knowledge-based engineering organization. The project may also show what is needed from the customer so that KBE systems can be developed in shorter time, with the same level of effectiveness.

**Keeping Track of Customer Requests:**

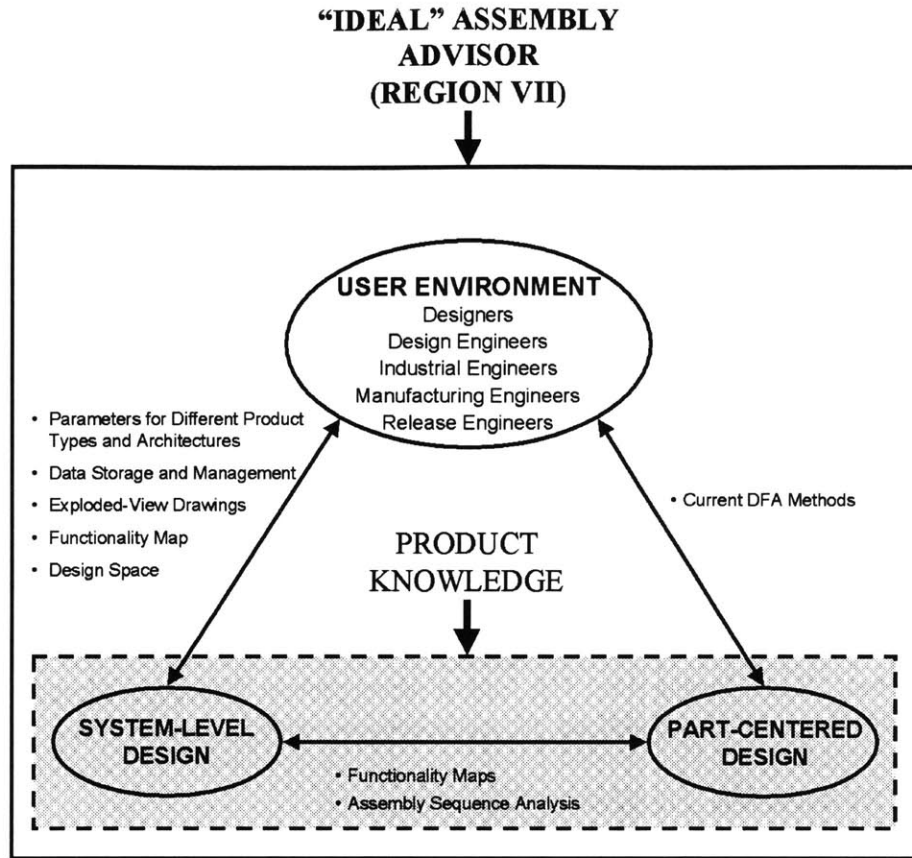
KBE does not keep track of what requests it receives on a week-to-week or a month-to-month basis, nor does it keep track of what requests it accepts during that same time

period. Keeping record of these events can be beneficial for two reasons: (1) KBE can determine how well it is handling in-coming requests (by looking at the gap between requests received and requests accepted, over time); and (2) KBE can see what most customers are interested in having. The results from this record can also be used to understand how customer perception commonly works in a dynamic environment, as mentioned in Chapter VI.

### 9.5 Chapter Summary:

This chapter mentioned the three key “ingredients” necessary for the development of an assembly advisor—the framework for a typical advisor, general assembly knowledge, and the user environment (which included product knowledge). However, the chapter did not stop there; it went on to discuss how these pieces interact with one another and where the “ideal” assembly advisor fits into this picture. The regions of knowledge space where these interactions occur are very important; they show the limitations of existing DFA methods and tools. Work has to be done to align these three major regions of assembly-advisor knowledge with one another so that their interactions are restriction free.

The chapter went on to show ways in which these three regions of knowledge space can work together in harmony. It mostly spoke about ways to link system-level design with part-centric design, and then to have these two methodologies work collectively with the user in the environment of an assembly advisor. The figure below summarizes what was said in Section 9.3.



**Figure 9.2:** Possible Ways to Link the User with Both System-Level and Part-Center Design

The chapter was not particularly interested in how to connect the user environment with part-centered design because current DFA methods and tools do this already. (Refer to Chapters IV and VIII) What this chapter was mostly focused on was describing ways to link system-level design with part-center design and system-level design with the user environment. The real difficulty comes when applying links to system-level and part-center design, due to the number of limitations that plague existing expert systems as mentioned in Chapter II.

# REFERENCES

---

- Andreasen, M., S. Kahler and T. Lund. Design for Assembly, 2<sup>nd</sup> Ed. New York: Springer-Verlag, 1988.
- Baudin, Michel. "Lean Production: The End of Management Whack-A-Mole." 16 Mar. 1999. [http://www.mmt-inst.com/End\\_of\\_management\\_whack\\_a\\_mole.html](http://www.mmt-inst.com/End_of_management_whack_a_mole.html).
- Boothroyd-Dewhurst, Inc. <http://www.dfma.com>
- Boothroyd, Geoffrey. Assembly Automation and Product Design. New York: Marcel Dekker, Inc., 1992.
- Boothroyd, Geoffrey and Peter Dewhurst. (1987). "Design for Assembly—The Key to Design for Manufacture." The International Journal of Advanced Manufacturing Technology. Vol 2 (3), 3-11. Springer-Verlag, Inc.
- Boothroyd, Geoffrey and Peter Dewhurst. "Product Design for Manufacture and Assembly." Manufacturing Engineering. Apr. 1988, 42-46. Society of Manufacturing Engineers.
- Brown, Carol and Daniel O'Leary. "Introduction to Artificial Intelligence." [http://www.bus.orst.edu/faculty/brownc/es\\_tutor/es\\_tutor.htm#5-AD](http://www.bus.orst.edu/faculty/brownc/es_tutor/es_tutor.htm#5-AD). 7 July 1999.
- Crouse, William H. and Donald L. Anglin. The Auto Book, 3<sup>rd</sup> Ed. New York: McGraw-Hill, 1984.
- Cusumano, Michael A. and Kentaro Nobeoka. Thinking Beyond Lean: How Multi-Project Management is Transforming Product Development at Toyota and Other Companies. New York: The Free Press, 1998.

Cusumano, Michael A. and Richard W. Selby. Microsoft Secrets: How the World's Most Powerful Software Company Creates Technology, Shapes Markets, and Manages People. New York: Simon & Schuster, 1998.

Dong, Qi. "Representing Information Flow and Knowledge Management in Product Design Using the Design Structure Matrix." Thesis, M.S. Department of Mechanical Engineering, MIT. Feb. 1999.

Fine, Charles H. Clockspeed: Winning Industry Control in the Age of Temporary Advantage. Reading, Massachusetts: Perseus Books, 1998.

"Fortune 500 Ranked by Performance." Fortune, 26 Apr. 1999, pg. F-32.

Harris, Crystal and Edu Morales. "Mycin."  
<http://web.mit.edu/STS001/www/Team7/mycin.htm>.

Ingrassia, Paul and Joseph B. White. Comeback: The Fall and Rise of the American Automobile Industry. New York: Simon & Schuster, 1994.

Jones, Daniel T., Daniel Roos, and James P. Womack. The Machine that Changed the World: The Story of Lean Production. New York: First HarperPerennial, 1991.

Kalpakjian, Serope. Manufacturing Engineering and Technology, 3<sup>rd</sup> Edition. Reading, Massachusetts: Addison-Wesley Publishing Co., 1995.

"Lean Production." MIT Course 2.996: Integrated Product and Process Design.  
<http://web.mit.edu/2.996/www/lecture4/tsld006.htm>.

Lehnerd, A.P and M.H. Meyer. Power of Product Platforms: Building Value & Cost Leadership. New York: The Free Press, 1997, pp. 1-22.

Mantripragada, R., J. Adams, S. Rhee and D. Whitney. "Integrated Computer Tools for Top-Down Assembly Design and Analysis." Dept. of Mechanical Engineering at MIT.

McCarthy, John. "Some Expert System Need Common Sense." <http://www-formal.stanford.edu/jmc/someneed/someneed.html>.

"MoldFlow" by CadTeck. <http://www.cadtek.com/moldflow.htm>.

Nevins, James and Daniel Whitney. Concurrent Design of Products and Processes: A Strategy for the Next Generation in Manufacturing. New York: McGraw-Hill, 1989.

- Otto, Kevin and Kristin Wood. Product Design: Techniques in Reverse Engineering, Systematic Design and New Product Development. Textbook for MIT Course 2.74: Optimal Product Design. (Unpublished)
- Rasiel, Ethan M. The McKinsey Way: Using the Techniques of the World's Top Strategic Consultants to Help You and Your Business. New York: McGraw-Hill, 1999.
- Redford, A. and J. Chal. Design for Assembly: Principles and Practice. London: McGraw-Hill, 1994
- Repenning, Nelson P. and John D. Sterman. "Getting Quality the Old-Fashioned Way: Self Confirming Attributions in the Dynamics of Process Improvement." Prepared for National Research Council workshop on Improving Theory and Research on Quality Enhancement in Organizations, Sept. 1997.
- Smith, Preston G. and Donald G. Reinertsen. Developing Products in Half the Time. New York: Van Nostrand Reinhold, 1995.
- Sterman, John. Business Dynamic: System Thinking and Modeling for a Complex World. Textbook for MIT-Sloan Course 15.874. (Unpublished)
- "Visteon—About Us—Facilities." <http://www.visteon.com/about/facilities/region2.html>
- Ulrich, Karl. (1995). "The Role of Product Architecture in the Manufacturing Firm." Research Policy. Vol. 24, 419-440. Elsevier Science Publishers.
- Walton, Mary. Car: A Drama of the American Workplace. New York: W.W. Norton & Co., 1997
- Whitney, Daniel. "Nippondenso Co. Ltd: A Case Study of Strategic Product Design." Research in Engineering Design. (1993) 5: 1-20. Springer-Verlag.

# SUGGESTED READING MATERIAL

---

## (DESIGN FOR ASSEMBLY)

Boothroyd, G. and L, Alting. (1992). "Design for Assembly and Disassembly." *Annals of the CIRP*, 41, 625-636.

Boothroyd, G., C.R. Poli and L.E. Murch. *Automatic Assembly*. New York: Marcel Dekker, Inc., 1982.

Butala, P. and M. Makovac. "The Role of DFA in Product Development—An Industrial Case Study." *Intelligent Assembly and Disassembly*. A Proceeding Volume from the IFAC Workshop. Bled, Slovenia, 21-23 May 1998, pp. 129-141.

Dewhurst, Peter. "Evaluation of Instrument Panel Designs for Cost of Manufacture and Environmental Impact." <http://www.dfma.com/news/panel.html>. 3 Jan. 2000.

Johnson, Richard F. "Using DFA to Enhance Value Engineering/Value Analysis Workshop Outcomes." Presented at the 12<sup>th</sup> Annual International forum on DFMA. Newport, RI, June 1997. <http://www.dfma.com/news/Johnson.html>. 3 Aug. 1998.

Nof, S., W. Wilhelm, and H.J. Warnecke. *Industrial Assembly*. London: Chapman & Hall, 1997. Chapter 3.

Ranky, Paul. *Concurrent/Simultaneous Engineering (Methods, Tools & Case Studies)*. United Kingdom: CIMware Ltd., 1994. Chapter 5.

Whitney, D. E. "Manufacturing by Design." *Harvard Business Review*. Vol 66, No. 4. 83-91. July-Aug. 1988.

Whitney, D.E. "The Potential for Assembly Modeling in Product Development and Manufacturing." 1996. <http://web.mit.edu/ctpid/www/Whitney/papers.html>. 3 Aug. 1998.

(LEAN MANUFACTURING)

Hayes, R.H. and G.P. Pisano. "Beyond World-Class: The New Manufacturing Strategy." Harvard Business Review. Jan.-Feb. 1994.

Jones, Daniel and James Womack. Lean Thinking: Banish Waste and Create Wealth in Your Corporation. New York: Simon & Schuster, 1996.

(ASSEMBLY SEQUENCE ANALYSIS)

Baldwin, D.F., T.E. Abell, M.C. Lui, T.L. DeFazio, and D.E. Whitney. (1991). "An Integrated Computer Aid for Generating and Evaluating Assembly Sequences for Mechanical Products." IEEE Transaction on Robotics and Automation. Vol. 7, No. 1 (February), pp. 78-94.

Mantripragada, R. and D.E. Whitney. "Datum Flow Chain: Theory and Applications." Presentation. 1997.

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

(BOOTHROYD-DEWHURST, INC COURT CASES)

BOOTHROYD DEWHURST, INC. Plaintiff v. CORRADO POLI, Defendant, CIVIL ACTION NO. 89-1650 (12 June 1991)

BOOTHROYD DEWHURST, INC. v. CORRADO POLI, CIVIL ACTION NO. 89-01650-F (15 Jan. 1991)

BOOTHROYD DEWHURST, INC., Plaintiff, v. THE BOARD OF TRUSTEES OF THE LELAND STANFORD JUNIOR UNIVERSITY, Defendant, C.A. No. 92-0075 P (8 July 1993)



# APPENDIX

---



## ASSEMBLY ANALYSIS TOOLS

SOFTWARE TOOL	DESCRIPTION	COMPANY Web Page
Boothroyd-Dewhurst, Inc— Design for Assembly	Time and cost estimation for assessing product design	Boothroyd-Dewhurst, Inc. <a href="http://www.dfma.com">www.dfma.com</a>
Deneb/ASSEMBLY	3D simulation of process design	Deneb <a href="http://www.deneb.com">www.deneb.com</a>
Deneb/ERGO	3D simulation of human tasks	
Taylor II Simulation	Simulation and analysis of processes in logistics manufacturing and material handling.	Taylor Enterprise Dynamics <a href="http://www.taylorii.com">www.taylorii.com</a>
Teamset—Design for Assembly	Point system for assessing product design	Teamset <a href="http://www.teamset.com">www.teamset.com</a>
VALISYS/Assembly	3-D tolerance stack-up analysis	Tecnomatix Technologies Ltd. <a href="http://www.tecnomatix.com">www.tecnomatix.com</a>
General Assembly Solution	“Line balancing, throughput optimization, logistics planning and cost estimations.”	
Final Assembly Solution	Line design, line balancing, logistics planning, product assembly analysis, time estimation, etc.	
WITNESS	Simulation and analysis of logistics planning	The Lanner Group <a href="http://www.lanner.com">www.lanner.com</a>



NUMBER OF STATED RULES THAT APPLY TO SPECIFIC COMPONENTS AND RULE TYPES

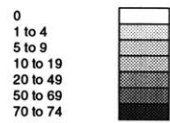
Guideline Types For Instrument Panels

Components

		DUR	ASSY	APP	MFG	COST	SQUEK	STRUC	DIM CTL	FUNC	SERV	INFO	HOMOL	PKG	MATL	ERGO	CUST SATIS	MFGAB	WARR	CUST SERV	NVH
		Instrument Panel	I/P	65	69	69	39	22	14	17	10	1	4	6	2	1	0	2	1	2	0
Floor Console	CONS	74	67	38	19	21	14	15	10	3	6	5	3	2	2	0	1	0	0	1	0
Defroster Trim	DRTR	76	66	41	17	17	13	13	7	1	9	5	4	0	2	0	0	0	0	0	0
Finish Panel	F/P	66	51	44	14	17	14	14	7	4	6	4	4	2	2	0	1	0	2	0	0
Ashtray	ASH	59	50	35	13	14	12	11	7	15	5	4	4	3	2	3	0	0	0	0	0
Cup Holder	CUP	57	50	31	13	14	11	11	7	16	5	4	4	8	1	1	0	0	0	0	0
Glove Box	GLVBX	54	43	26	15	12	9	11	7	3	4	4	4	0	2	0	0	0	0	0	0
Airbag	A/BAG	22	27	29	9	9	5	3	7	2	4	4	1	0	3	0	0	0	0	0	0
Soft Foam Instrument Panel	I/P.soft	3	5	6	98	1	2	1	0	0	0	0	1	0	0	1	0	0	0	0	0
Finish Panel (Fastener)	F/P.fasten	11	15	7	2	3	5	0	0	0	4	1	0	0	0	0	0	0	0	0	0
Air Conditioning Unit	A/C	6	7	16	0	3	0	0	0	2	3	0	0	0	0	0	1	2	0	0	0
Instrument Panel (Fastener)	I/P.fasten	13	12	1	1	3	4	0	0	1	2	1	0	0	0	0	0	0	0	0	0
Glove Box (Fastener)	GLVBX.fasten	10	8	2	2	6	4	0	0	0	1	1	0	0	0	0	0	0	0	0	0
Glove Box (Door)	GLVBX.door	3	1	11	5	4	4	0	2	0	0	0	0	0	1	0	0	0	0	0	0
Glove Box (Latch)	GLVBX.latch	2	3	5	0	5	2	0	2	8	1	0	0	0	1	2	0	0	0	0	0
Glove Box (Bin)	GLVBX.bin	3	2	4	1	7	4	0	0	1	1	0	0	1	0	3	0	0	0	0	0
Ashtray & Cup Holder (Fastener)	ASH/CUP.fasten	8	8	1	1	3	2	0	0	0	1	1	0	0	0	0	0	0	0	0	0
Airbag fastener	A/BAG.fasten	5	6	0	1	3	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0
Glove Box (Striker)	GLVBX.strkr	1	0	4	0	2	3	1	5	2	0	0	0	0	0	0	0	0	0	0	0
Glove Box (Bumper)	GLVBX.bumper	1	0	5	0	2	5	0	0	2	0	0	0	0	0	0	0	0	0	0	0
Glove Box (Hinge)	GLVBX.hinge	2	0	3	0	2	4	1	3	0	0	0	0	0	0	0	0	0	0	0	0
Defroster Grille	GRILLE	1	0	9	0	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Soft Foam Glove Box	GLVBX.soft	0	0	8	5	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0
Ashtray & Cup Holder (Lighter)	ASH/CUP.lighter	3	0	0	0	2	0	0	0	1	0	0	0	2	0	0	0	0	2	0	0
Steering Column Cover	SCC	1	0	6	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Airbag Door	A/BAG.door	0	0	7	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Airbag Canister	A/BAG.can	2	1	2	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Power Point	PW/PRT	2	0	0	0	1	2	0	0	3	0	0	0	0	0	0	0	0	0	0	0
Manufacturing	MFG	1	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Cluster	CLUST	1	0	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Airbag Module	ABMOD	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Assembly	ASSY	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Corporate Design	CORP DESIGN	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fastener	FASTEN	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Finish Panel (Grille)	F/P.grille	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

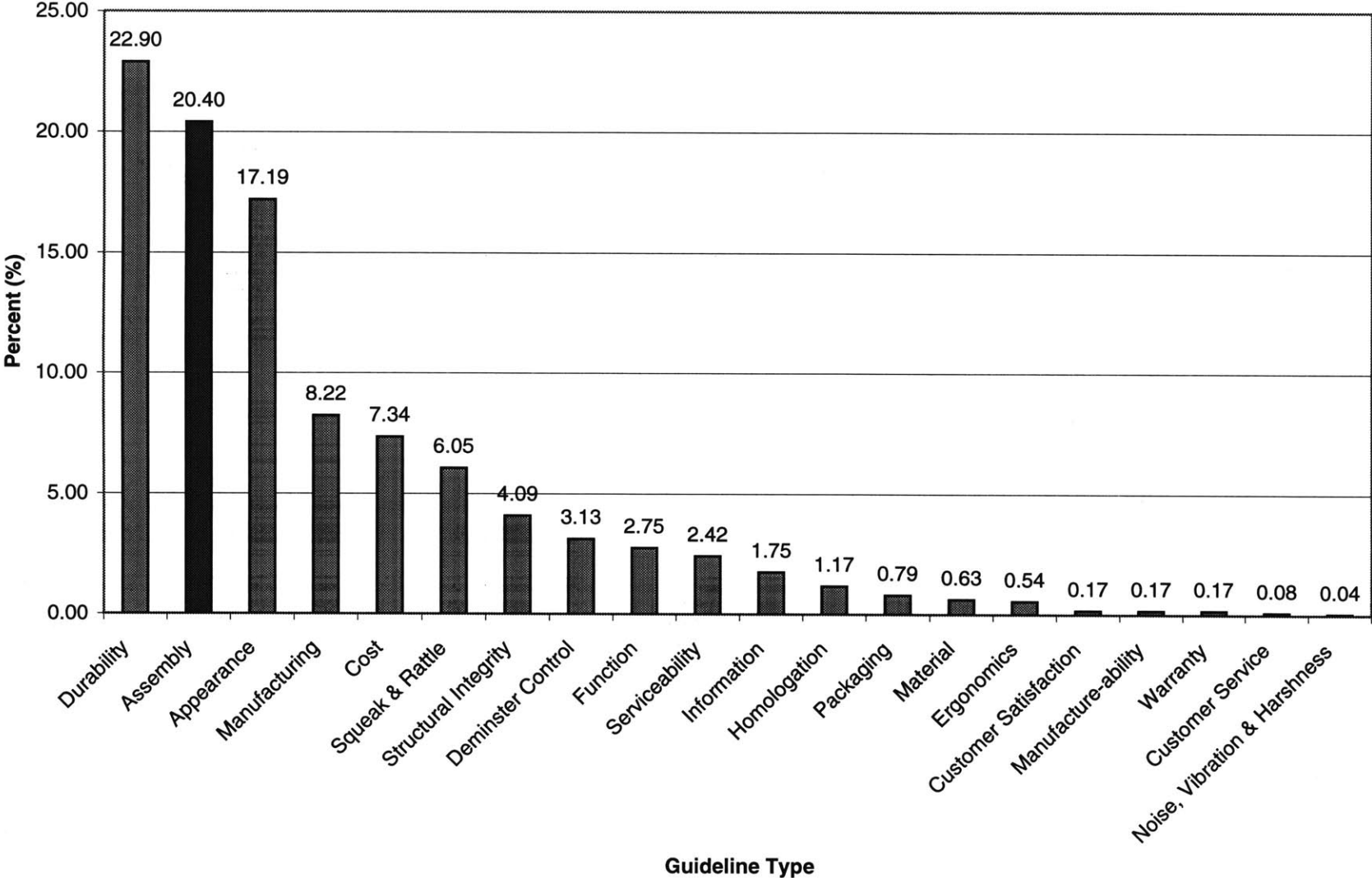
TOTAL	PERCENT	RANK
322	13.43	1
281	11.72	2
267	11.14	3
252	10.51	4
237	9.89	5
233	9.72	6
194	8.09	7
125	5.21	8
58	2.42	9
48	2.00	10
40	1.67	11
38	1.59	12
34	1.42	13
31	1.29	14
31	1.29	14
27	1.13	16
25	1.04	17
19	0.79	18
18	0.75	19
15	0.63	20
15	0.63	20
14	0.58	22
14	0.58	22
10	0.42	24
10	0.42	24
9	0.38	26
8	0.33	27
8	0.33	27
5	0.21	29
4	0.17	30
1	0.04	31
1	0.04	31
1	0.04	31
1	0.04	31
1	0.04	31

TOTAL	549	489	412	197	176	145	98	75	66	58	42	28	19	15	13	4	4	4	2	1
PERCENT	22.90	20.40	17.19	8.22	7.34	6.05	4.09	3.13	2.75	2.42	1.75	1.17	0.79	0.63	0.54	0.17	0.17	0.17	0.08	0.04
RANK	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	16	16	19	20





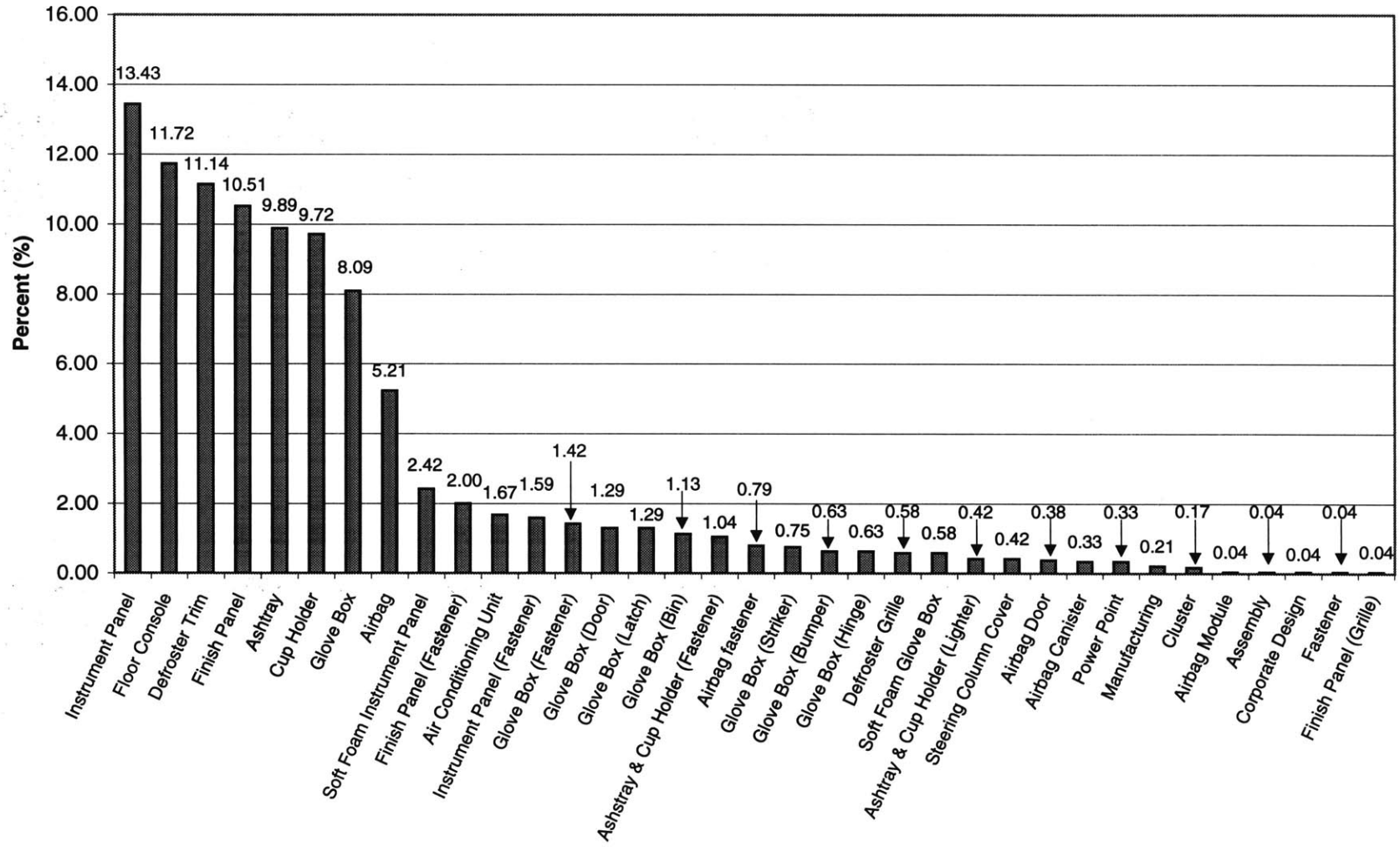
### Design Guidelines for Instrument Panels







## Guidelines for Instrument Panels



**Requirements for Specific Areas of Instrument Panel**



### Comments about the Correlation Matrices (for Instrument Panel Design Guides):

The figures (matrices) on the following pages show one way to express relationships either between different rule types (i.e., design drivers) or component types. The information for developing these correlation matrices came directly from the matrix titled “Number of Stated Rules that Apply to Specific Components and Rule Types.” It can be found near the beginning of the Appendix. (The author will refer to this matrix throughout the rest of this section as Matrix R.)

The cells in each matrix show the degree of association between a rule (or component) type in a column with a rule (or component) type in a row. The cell numbers range from 0 (weakest correlation) to 1 (strongest correlation). (Note: the numerical relationship between a rule or component and itself will always be 1; these values can be seen “chained together” along the main diagonal of the two arrays, starting with the top left cell (Cell [1,1]) and ending with the bottom right cell (Cell [n,n])

To illustrate the use of these matrices, four cells were selected, two from each array, and are described below. (Each selected cell is shown with a dark border around it.) In the rule correlation matrix, one can see that there is a strong connection between “Assembly” and “Squeak and Rattle.” (To some extent, this makes sense. Loose fitting parts can cause components to vibrate—certainly an undesirable effect when designing and manufacturing any vehicle.) Another example, from the same figure, shows that “Noise, Vibration and Harshness” has no association with “Warranty.” (In other words, the matrix “says” that issues governing noise, vibration and harshness are not warranty-related.) In the component correlation matrix, one can see that there is no relationship between “Airbag Module” and “Ashtray and Cup Holder.” (The two components are physically separate from one another.) On the other hand, however, there appears to be an extremely strong relationship between “Finish Panel” and “Airbag Module.” (The two components are physically tied together; it is at this “crossroad” where poor fit-and-finish conditions can make themselves known to the observer.)

The reader should note that all of these values might not be valid in an actual assessment; the numbers here are based purely on what is shown in the design manuals. As was mentioned in Chapter V, not all information regarding the design of a vehicle sub-system, like an instrument panel, can be easily represented in written form. For instance, system-level requirements are less likely to be found in design guidelines because they are very difficult to capture in a few sentences or with a simple figure. Designers and engineers, however, can use these matrices as a tool to see what types of rules are needed in current and future manuals.

### The Idea behind Developing a Correlation Matrix (Refer to Matrix R):

Consider, for now, developing a correlation matrix for rule types (i.e., design drivers). Each design driver should be pictured as a vector with n-dimensions, where each dimension is a designated component type. (In this case, there are 35 dimensions to

consider because there are 35 different component types that are associated with each design driver.) Determining the relationship between any two design drivers is the same as comparing the magnitude and direction of two vectors in “component” space. (This can be done simply by taking the dot product of any two unit vectors.)

Notation:

The following mathematical notation will be used to describe how one can develop a correlation matrix for rules and components.

- $R$  = Matrix with title “Number of Stated Rules that Apply to Specific Components and Rule Types.”
- $S_c$  = Matrix  $R$  normalized for component-type correlation
- $S_r$  = Matrix  $R$  normalized for rule-type correlation

Correlation Matrix for Component Types:

Calculate the magnitude of each row vector of  $R$ . To do this, multiply  $R^T$  by  $R$  to generate a new matrix  $RR^T$ . Take the square root of only those generated values which are located in Cells  $[i,i]$ . (The square root of Cell  $[1,1]$  represents the magnitude of row vector 1 of Matrix  $R$ ; the square root of Cell  $[2,2]$  represents the magnitude of row vector 2 of Matrix  $R$ ; etc.) Divide each vector component in each row of  $R$  by its designated vector magnitude. The result is a new array that is “normalized,” Matrix  $S_c$ .

Calculate the transpose of Matrix  $S_c$ . Multiply  $(S_c)^T$  by  $S_c$  to get the component correlation matrix,  $S_c(S_c)^T$ .

Correlation Matrix for Rule Types:

Calculate the magnitude of each column vector of  $R$ . To do this, multiply  $R$  by  $R^T$  to generate a new matrix  $R^TR$ . Take the square root of only those generated values which are located in Cells  $[i,i]$ . (The square root of Cell  $[1,1]$  represents the magnitude of column vector 1 of Matrix  $R$ ; the square root of Cell  $[2,2]$  represents the magnitude of column vector 2 of Matrix  $R$ ; etc.) Divide each vector component in each column of  $R$  by its designated vector magnitude. The result is a new array that is “normalized,” Matrix  $S_r$ .

Calculate the transpose of Matrix  $S_r$ . Multiple  $S_r$  by  $(S_r)^T$  to get the rule correlation matrix,  $(S_r)^TS_r$ .

Correlation Matrix for Component Types

Instrument Panel	Floor Console	Director's Tim	Finish Panel	Assembly	Cap Holder	Glove Box	Alibag	Soft Foam Instrument Panel	Finish Panel (Fastener)	Air Conditioning Unit	Instrument Panel (Fastener)	Glove Box (Fastener)	Glove Box (Door)	Glove Box (Latch)	Glove Box (Bin)	ASH/CUP Holder (Fastener)	Alibag Fastener	Glove Box (Striker)	Glove Box (Bumper)	Glove Box (Hinge)	GRILLE	Soft Foam Glove Box	ASH/CUP Holder (Lighter)	Steering Column Cover	Alibag Door	Alibag Canister	Power Point	Manufacturing	Cluster	Alibag Module	Assembly	Corporate Design	Fastener	Finish Panel (Grille)		
VP	CONS	DRTR	F/P	ASH	CUP	GLVBX	ABAG	VP.aot	F.P. fasten	A/C	IP. fasten	GLVBX fasten	GLVBX door	GLVBX latch	GLVBX bin	ASH/CUP. fasten	ABAG fasten	GLVBX strkr	GLVBX bumper	GLVBX hinge	GRILLE	GLVBX soft	ASH/CUP. lighter	SCC	ABAG door	ABAG can	PWPRT	MFG	CLUST	ABMOD	ASSY	CORP DESIGN	FASTEN	F.P. grille		
VP	0.9540	0.9540	0.9599	0.9668	0.9488	0.9379	0.9518	0.9786	0.9090	0.9015	0.8417	0.7980	0.8364	0.7895	0.5842	0.6615	0.8263	0.7745	0.5083	0.5436	0.5858	0.6207	0.6340	0.4096	0.6480	0.5547	0.8547	0.3424	0.7224	0.5250	0.5477	0.1111	0.5477	0.5180	0.5477	
CONS	0.9540	0.9540	0.9599	0.9668	0.9488	0.9379	0.9518	0.9786	0.9090	0.9015	0.8417	0.7980	0.8364	0.7895	0.5842	0.6615	0.8263	0.7745	0.5083	0.5436	0.5858	0.6207	0.6340	0.4096	0.6480	0.5547	0.8547	0.3424	0.7224	0.5250	0.5477	0.1111	0.5477	0.5180	0.5477	
DRTR	0.9599	0.9599	0.9668	0.9668	0.9488	0.9379	0.9518	0.9786	0.9090	0.9015	0.8417	0.7980	0.8364	0.7895	0.5842	0.6615	0.8263	0.7745	0.5083	0.5436	0.5858	0.6207	0.6340	0.4096	0.6480	0.5547	0.8547	0.3424	0.7224	0.5250	0.5477	0.1111	0.5477	0.5180	0.5477	
F/P	0.9668	0.9668	0.9668	0.9668	0.9488	0.9379	0.9518	0.9786	0.9090	0.9015	0.8417	0.7980	0.8364	0.7895	0.5842	0.6615	0.8263	0.7745	0.5083	0.5436	0.5858	0.6207	0.6340	0.4096	0.6480	0.5547	0.8547	0.3424	0.7224	0.5250	0.5477	0.1111	0.5477	0.5180	0.5477	
ASH	0.9488	0.9488	0.9488	0.9488	0.9488	0.9379	0.9518	0.9786	0.9090	0.9015	0.8417	0.7980	0.8364	0.7895	0.5842	0.6615	0.8263	0.7745	0.5083	0.5436	0.5858	0.6207	0.6340	0.4096	0.6480	0.5547	0.8547	0.3424	0.7224	0.5250	0.5477	0.1111	0.5477	0.5180	0.5477	
CUP	0.9379	0.9379	0.9379	0.9379	0.9379	0.9379	0.9518	0.9786	0.9090	0.9015	0.8417	0.7980	0.8364	0.7895	0.5842	0.6615	0.8263	0.7745	0.5083	0.5436	0.5858	0.6207	0.6340	0.4096	0.6480	0.5547	0.8547	0.3424	0.7224	0.5250	0.5477	0.1111	0.5477	0.5180	0.5477	
GLVBX	0.9518	0.9518	0.9518	0.9518	0.9518	0.9518	0.9518	0.9786	0.9090	0.9015	0.8417	0.7980	0.8364	0.7895	0.5842	0.6615	0.8263	0.7745	0.5083	0.5436	0.5858	0.6207	0.6340	0.4096	0.6480	0.5547	0.8547	0.3424	0.7224	0.5250	0.5477	0.1111	0.5477	0.5180	0.5477	
ABAG	0.9786	0.9786	0.9786	0.9786	0.9786	0.9786	0.9786	0.9786	0.9090	0.9015	0.8417	0.7980	0.8364	0.7895	0.5842	0.6615	0.8263	0.7745	0.5083	0.5436	0.5858	0.6207	0.6340	0.4096	0.6480	0.5547	0.8547	0.3424	0.7224	0.5250	0.5477	0.1111	0.5477	0.5180	0.5477	
VP.aot	0.5090	0.3563	0.3377	0.3362	0.3356	0.3781	0.3917	0.2866	0.2031	0.2122	0.2933	0.5222	0.1326	0.2453	0.2448	0.2605	0.1191	0.1502	0.1368	0.1710	0.7305	0.0901	0.1740	0.1615	0.1924	0.0665	0.7604	0.1360	0.1638	0.0513	0.1638	0.1262	0.1638	0.1262	0.1638	
F.P. fasten	0.9015	0.9513	0.9579	0.9354	0.9326	0.9351	0.9151	0.2988	0.3031	0.7470	0.9377	0.9125	0.5667	0.6201	0.6289	0.9036	0.3951	0.4726	0.4960	0.4522	0.3418	0.3920	0.8101	0.3565	0.8573	0.3889	0.7071	0.2357	0.3300	0.2357	0.3300	0.7071	0.3300	0.7071	0.3300	
A/C	0.8417	0.7407	0.7846	0.8036	0.7782	0.7564	0.7295	0.8913	0.2031	0.2122	0.2933	0.5222	0.1326	0.2453	0.2448	0.2605	0.1191	0.1502	0.1368	0.1710	0.7305	0.0901	0.1740	0.1615	0.1924	0.0665	0.7604	0.1360	0.1638	0.0513	0.1638	0.1262	0.1638	0.1262	0.1638	
IP. fasten	0.7980	0.9315	0.9289	0.8935	0.8113	0.9169	0.9246	0.7871	0.2122	0.2933	0.5222	0.1326	0.2453	0.2448	0.2605	0.1191	0.1502	0.1368	0.1710	0.7305	0.0901	0.1740	0.1615	0.1924	0.0665	0.7604	0.1360	0.1638	0.0513	0.1638	0.1262	0.1638	0.1262	0.1638		
GLVBX fasten	0.8364	0.9300	0.9191	0.9031	0.9018	0.9032	0.9199	0.8159	0.2933	0.5222	0.1326	0.2453	0.2448	0.2605	0.1191	0.1502	0.1368	0.1710	0.7305	0.0901	0.1740	0.1615	0.1924	0.0665	0.7604	0.1360	0.1638	0.0513	0.1638	0.1262	0.1638	0.1262	0.1638	0.1262	0.1638	
GLVBX door	0.7895	0.6116	0.6185	0.6902	0.6348	0.6069	0.6109	0.7834	0.5222	0.5667	0.7982	0.3676	0.5267	0.5941	0.7446	0.4065	0.3491	0.7155	0.8059	0.7374	0.8855	0.8728	0.2609	0.8997	0.8164	0.7688	0.3054	0.3599	0.6465	0.7918	0.2879	0.7918	0.0720	0.7918		
GLVBX latch	0.6615	0.5385	0.5234	0.5683	0.6305	0.6325	0.5263	0.6295	0.1338	0.5201	0.6453	0.4527	0.5210	0.5941	0.7307	0.4546	0.5251	0.6958	0.8907	0.8592	0.5092	0.3579	0.4309	0.5328	0.4363	0.6076	0.7344	0.2946	0.3782	0.4211	0.1984	0.4211	0.2526	0.4211		
GLVBX bin	0.6615	0.6327	0.6191	0.6572	0.5485	0.6393	0.6104	0.6668	0.2453	0.6289	0.6148	0.5613	0.7395	0.7446	0.7327	0.6021	0.6140	0.5986	0.7426	0.7076	0.5846	0.3988	0.5359	0.6265	0.4332	0.7814	0.5489	0.2900	0.5920	0.3967	0.3967	0.3967	0.1933	0.3967		
ASH/CUP fasten	0.8263	0.9439	0.9372	0.9032	0.9131	0.9182	0.9329	0.8146	0.3448	0.9435	0.5714	0.9911	0.9722	0.4065	0.4546	0.6021	0.3000	0.9781	0.2573	0.3135	0.4179	0.2251	0.1453	0.5312	0.2691	0.1047	0.7871	0.4502	0.7197	0.4407	0.0830	0.1681	0.0830	0.6844	0.0830	
ABAG fasten	0.7745	0.8994	0.8851	0.8453	0.8753	0.8870	0.8840	0.7702	0.2605	0.9036	0.5177	0.9622	0.9524	0.3491	0.5251	0.6140	0.8781	0.2000	0.2395	0.2706	0.3522	0.1324	0.1010	0.5416	0.1782	0.0162	0.7000	0.4899	0.7313	0.3300	0.0000	0.1155	0.0000	0.6928	0.0000	
GLVBX strkr	0.5083	0.4344	0.4185	0.4662	0.4836	0.4894	0.4305	0.5678	0.1191	0.3651	0.5384	0.2566	0.3607	0.7155	0.6956	0.5805	0.2573	0.2385	0.9069	0.7395	0.9056	0.6461	0.5184	0.2477	0.6874	0.5804	0.5390	0.4899	0.0430	0.5798	0.5184	0.3873	0.5184	0.0000	0.5184	
GLVBX bumper	0.5436	0.4392	0.4444	0.5203	0.5054	0.4837	0.4310	0.5753	0.1502	0.4726	0.6515	0.3220	0.4503	0.8059	0.6907	0.7426	0.3135	0.2706	0.7395	0.2000	0.8140	0.8551	0.5696	0.2498	0.8838	0.7292	0.7141	0.6137	0.0434	0.8504	0.6509	0.6509	0.6509	0.6509		
GLVBX hinge	0.5858	0.5455	0.5343	0.5957	0.5578	0.5389	0.5418	0.6078	0.1368	0.4960	0.5247	0.4181	0.5478	0.7574	0.5522	0.7076	0.4179	0.3822	0.9056	0.8140	0.8000	0.6897	0.4786	0.3251	0.7059	0.5339	0.7044	0.5032	0.1017	0.8093	0.4578	0.6100	0.4578	0.0000	0.4578	
GRILLE	0.6207	0.4410	0.4678	0.5455	0.4885	0.4537	0.4357	0.6610	0.1710	0.4522	0.8318	0.2074	0.3190	0.8855	0.5092	0.5846	0.2251	0.1324	0.1324	0.6461	0.8551	0.6837	0.2000	0.7426	0.1111	0.9974	0.9635	0.7223	0.2212	0.0348	0.6810	0.9383	0.3128	0.9383	0.0000	0.9383
GLVBX soft	0.6340	0.3886	0.4008	0.4506	0.4118	0.3878	0.4010	0.6066	0.7305	0.3418	0.6255	0.1075	0.2245	0.8728	0.3579	0.3988	0.1453	0.1010	0.5164	0.5696	0.4786	0.7426	0.2000	0.0000	0.7329	0.7527	0.4691	0.0589	0.4167	0.4082	0.7500	0.1250	0.7500	0.0000	0.7500	
ASH/CUP lighter	0.4096	0.5103	0.4577	0.5224	0.5325	0.5596	0.5150	0.3786	0.0901	0.3920	0.2860	0.5272	0.5956	0.2609	0.4308	0.5359	0.8312	0.5416	0.2477	0.2498	0.3251	0.1111	0.0000	0.2000	0.1845	0.0000	0.4924	0.5528	0.2132	0.2611	0.0000	0.0000	0.0000	0.0000	0.0000	
SCC	0.6480	0.4783	0.5031	0.5811	0.5235	0.4892	0.4733	0.6851	0.1740	0.4801	0.6446	0.2489	0.3695	0.8997	0.5328	0.6285	0.2691	0.1782	0.6574	0.8638	0.7059	0.9974	0.7329	0.1645	0.2000	0.9507	0.7572	0.2546	0.0514	0.8929	0.9258	0.3086	0.9258	0.0000	0.9258	
ABAG door	0.5547	0.3500	0.3834	0.4584	0.4038	0.3891	0.3480	0.6044	0.1615	0.3565	0.8178	0.1677	0.8164	0.4363	0.4332	0.1047	0.0162	0.5904	0.7292	0.5339	0.9635	0.7527	0.0000	0.9507	0.2000	0.8063	0.0680	0.0000	0.5145	0.9802	0.1400	0.9802	0.0000	0.9802		
ABAG can	0.8547	0.8464	0.8532	0.8929	0.8581	0.8397	0.8412	0.8783	0.1824	0.8573	0.8577	0.7804	0.8257	0.7688	0.6076	0.7814	0.7000	0.5590	0.7141	0.7044	0.7223	0.4691	0.4924	0.7572	0.6063	0.0000	0.4783	0.3849	0.7071							



Correlation Matrix for Guideline Types

	Durability	Assembly	Appearance	Manufacturing	Cost	Squeak & Rattle	Structural Integrity	Dimensional Control	Function	Serviceability	Information	Homologation	Packaging	Material	Ergonomics	Customer Satisfaction	Manufacturability	Warranty	Customer Service	Noise, Vibration & Harshness
	DUR	ASSY	APP	MFG	COST	SQUEK	STRUC	DIM CTL	FUNC	SERV	INFO	HOMOL	PKG	MATL	ERGO	CUST SATIS	MFGAB	WARR	CUST SERV	NVH
DUR	1.0000	0.9944	0.9350	0.7566	0.9678	0.9545	0.9817	0.9274	0.6277	0.9405	0.9688	0.9491	0.6100	0.8045	0.4100	0.6082	0.2894	0.2813	0.5666	0.3747
ASSY	0.9944	1.0000	0.9477	0.7806	0.9716	0.9494	0.9764	0.9325	0.6112	0.9463	0.9812	0.9266	0.5935	0.7996	0.4194	0.6165	0.3304	0.2340	0.6057	0.4218
APP	0.9350	0.9477	1.0000	0.8175	0.9548	0.9315	0.9457	0.9309	0.5595	0.8799	0.9477	0.8554	0.5291	0.7376	0.4861	0.6984	0.5027	0.2602	0.6328	0.5771
MFG	0.7566	0.7806	0.8175	1.0000	0.7736	0.7567	0.7900	0.7458	0.3842	0.6578	0.7734	0.7134	0.3970	0.5235	0.4861	0.5354	0.4101	0.1472	0.6100	0.5800
COST	0.9678	0.9716	0.9548	0.7736	1.0000	0.9695	0.9514	0.9379	0.6147	0.9162	0.9576	0.8766	0.5864	0.7769	0.5252	0.6551	0.3676	0.2794	0.6324	0.4575
SQUEK	0.9545	0.9494	0.9315	0.7567	0.9695	1.0000	0.9295	0.9174	0.6333	0.9068	0.9287	0.8871	0.5876	0.7557	0.4896	0.5813	0.2740	0.2740	0.5481	0.3875
STRUC	0.9817	0.9764	0.9457	0.7900	0.9514	0.9295	1.0000	0.9371	0.5865	0.8767	0.9517	0.9309	0.5934	0.7506	0.4143	0.6495	0.3395	0.2796	0.6390	0.4801
DIM CTL	0.9274	0.9325	0.9309	0.7458	0.9379	0.9174	0.9371	1.0000	0.6168	0.8497	0.9411	0.8676	0.5598	0.8216	0.4327	0.5826	0.3051	0.2136	0.6103	0.4315
FUNC	0.6277	0.6112	0.5595	0.3842	0.6147	0.6333	0.5865	0.6168	1.0000	0.5938	0.5804	0.6984	0.8291	0.5963	0.6165	0.2024	0.0859	0.1431	0.1145	0.0405
SERV	0.9405	0.9463	0.8799	0.6578	0.9162	0.9068	0.8767	0.8497	0.5938	1.0000	0.9198	0.8811	0.5325	0.8284	0.3624	0.5617	0.2927	0.2509	0.4181	0.2365
INFO	0.9688	0.9812	0.9477	0.7734	0.9578	0.9287	0.9517	0.9411	0.5804	0.9198	1.0000	0.8949	0.5559	0.8217	0.3965	0.5719	0.3235	0.2157	0.5931	0.4575
HOMOL	0.9491	0.9266	0.8554	0.7134	0.8766	0.8871	0.9309	0.8676	0.6984	0.8811	0.8949	1.0000	0.6565	0.8249	0.3990	0.4593	0.1443	0.2887	0.3608	0.2041
PKG	0.6100	0.5935	0.5291	0.3970	0.5864	0.5876	0.5834	0.5598	0.8291	0.5325	0.5559	0.8565	1.0000	0.4236	0.4380	0.2680	0.0758	0.3032	0.2274	0.1072
MATL	0.8045	0.7996	0.7376	0.5235	0.7769	0.7557	0.7506	0.8216	0.5963	0.8284	0.8217	0.8249	0.4236	1.0000	0.3002	0.3592	0.0000	0.2540	0.2540	0.0000
ERGO	0.4100	0.4194	0.4861	0.4861	0.5252	0.4896	0.4143	0.4327	0.6165	0.3624	0.3965	0.3990	0.4380	0.3002	1.0000	0.1857	0.2626	0.0000	0.2626	0.3714
CUST SATIS	0.6082	0.6165	0.6984	0.5354	0.6551	0.5813	0.6495	0.5826	0.2024	0.5617	0.5719	0.4593	0.2680	0.3592	0.1857	1.0000	0.7071	0.3536	0.7071	0.5000
MFGAB	0.2894	0.3304	0.5027	0.4101	0.3676	0.2740	0.3395	0.3051	0.0859	0.2927	0.3235	0.1443	0.0758	0.0000	0.2626	0.7071	1.0000	0.0000	0.5000	0.7071
WARR	0.2813	0.2340	0.2602	0.1472	0.2794	0.2740	0.2796	0.2136	0.1431	0.2509	0.2157	0.2887	0.3032	0.2540	0.0000	0.3536	0.0000	1.0000	0.0000	0.0000
CUST SERV	0.5666	0.6057	0.6328	0.6100	0.6324	0.5481	0.6390	0.6103	0.1145	0.4181	0.5931	0.3608	0.2274	0.2540	0.2626	0.7071	0.5000	0.0000	1.0000	0.7071
NVH	0.3747	0.4218	0.5771	0.5800	0.4575	0.3875	0.4801	0.4315	0.0405	0.2365	0.4575	0.2041	0.1072	0.0000	0.3714	0.5000	0.7071	0.0000	0.7071	1.0000