Managing the Integration Problem in Concurrent Engineering

Kent R. McCord
and
Steven D. Eppinger

WP# 3594-93-MSA August, 1993

Acknowledgment

This research was funded by General Motors and the Leaders for Manufacturing Program, a partnership involving thirteen major U.S. manufacturing firms and M.I.T.'s schools of engineering and management.

Managing the Integration Problem in Concurrent Engineering

by

Kent R. McCord
and
Steven D. Eppinger
M.I.T. Sloan School of Management

Abstract

Concurrent engineering in large-scale product development generally involves multiple cross-functional teams working simultaneously on separate aspects of the overall development effort. The often complex technical coupling among such teams makes integrating their activities an essential yet difficult task for project management. We refer to this challenge of integrating teams as the *integration problem* in concurrent engineering. This paper presents a methodology for determining the needs for integration and coordination by studying the underlying technical structure of a project. We use a project modeling tool known as the Design Structure Matrix (DSM) to depict the patterns of required information flow in a project. This matrix representation allows us to identify where coordination is most essential and then to design integration mechanisms based on the specific technical information needs of the project. The utility of this methodology in an industrial setting is demonstrated by an application to the development of a new automobile engine at General Motors.

1. Introduction

In striving to bring new products to market faster and with higher quality, manufacturing firms are focusing more than ever on improving the *process* through which they engineer new products and manufacturing technologies. This effort has led to the concept of concurrent engineering, heralded during the past decade as a sure-fire approach to developing better products faster [1, 2, 12, 16]. For most firms this has meant the use of multi-functional development teams designed to break down traditional communication barriers between functional groups such as product design, marketing, manufacturing engineering, and purchasing.

For relatively simple products and processes, where only a handful of people are required to bring the product from concept to market, a multi-functional team approach is regarded to be an effective means of reducing development time [3, 15]. It works because the close communication and mutual understanding developed among the members of the small team allows for the aggressive overlapping of and iteration among coupled development tasks. Challenging inter-functional technical issues are exposed early and resolved rapidly.

When faced with product development on a very large and complex scale, such as in the automotive or computer industries, implementing concurrent engineering is much less straightforward. In such cases, hundreds of people are generally organized into many small teams, each working simultaneously on separate portions of the overall development effort. Though the teams are independent, they generally cannot work in isolation of one another due to the often complex technical interfaces between them. Communication must occur across team boundaries. Managing concurrent engineering for complex product development therefore entails more than simply making each team multi-functional -- it also requires facilitating communication between teams to integrate their separate efforts into a well-designed system [10].

The examples in Figure 1.1 illustrate these challenges in managing concurrent engineering. The many deficiencies with the traditional "over-the-wall" approach to product development, as shown in Figure 1a, have been well documented [1, 17]. Marketing precedes design, design precedes manufacturing, and all communication between the functional groups is through batch information transfer at the end of each phase. Figure 1b shows concurrent engineering in a small project where the greatest challenge is to facilitate frequent information transfer between the separate functions. An example of such a project is the development of a new soda bottle, in which the marketing, design, and manufacturing people can all work closely together on one team. Management's main concern in such a project is *concurrently executing activities which are nominally sequential.* We refer to this as the *overlapping problem* in concurrent engineering [8].

The added complexity of managing a larger project, such as the development of a new laptop computer, is shown in Figure 1c. In this case, project management must also focus on coordinating the activities of the separate sub-system development teams. The drive system team must interface with the team responsible for the main board, which in turn must work closely with the LCD screen team. The packaging issue might be addressed by yet another team, requiring coordination across the entire project. The challenge of integrating several teams is typical of large engineering efforts in which management is attempting to implement concurrent engineering [10]. We refer to this as the *integration problem* in concurrent engineering - *integrating activities which are nominally concurrent.* Our goal in this paper is to expose this issue and to present management tools and strategies for tackling the integration problem in industrial projects.

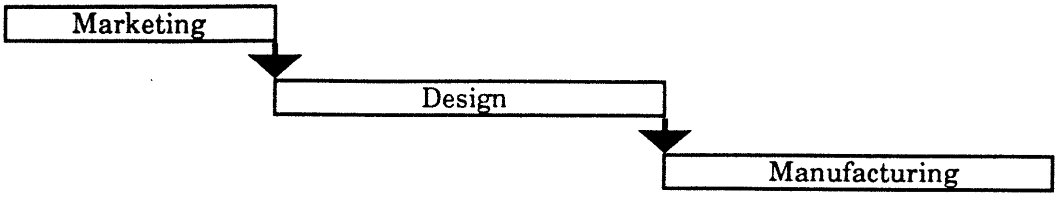


Figure 1a. Traditional "over-the-wall" approach

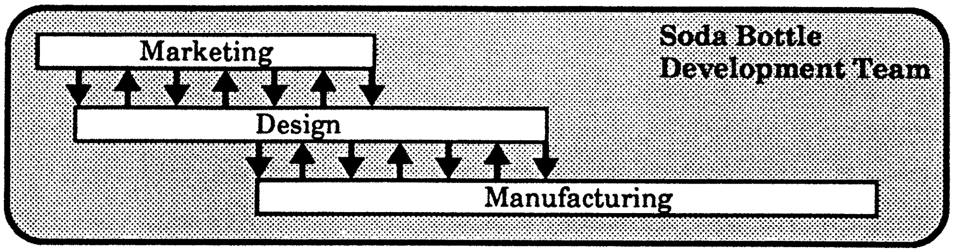


Figure 1b. Overlapping in concurrent engineering

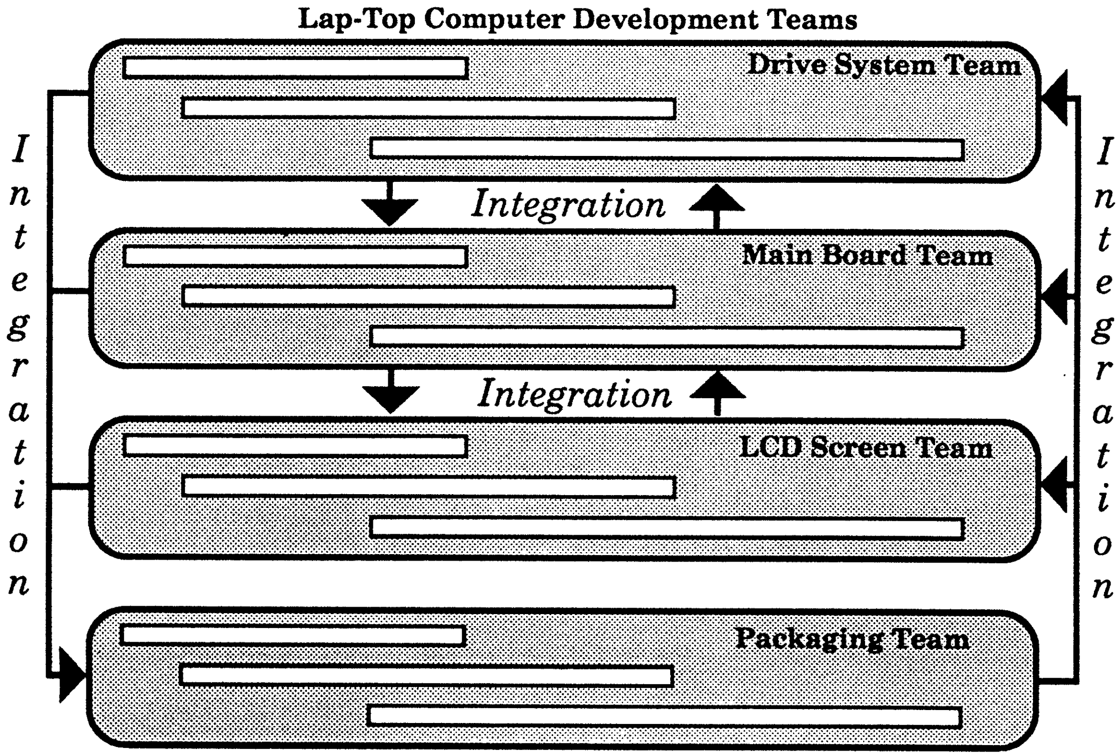


Figure 1c. Integration in concurrent engineering

Figure 1.1 Problems in Concurrent Engineering

Several aspects of this research distinguish it from traditional research on the product development process. First, we are striving to improve the efficiency of the entire development effort whereas most engineering design research focuses on improving the efficiency of isolated engineering tasks.

Second, as discussed earlier, we focus on the development of complex products where several hundred people are involved and the organizational solution is not as simple as forming a multi-functional team. Lastly, as will be shown, we draw a link between the technical information transfer needs of a project and organization design, whereas most research ignores this relationship.

We will present our methodology for addressing the integration problem in the next section. Following that will be an overview of several traditional mechanisms for integrating teams as well as a presentation of some new concepts for more effective integration. We will lastly describe our involvement with the development of new automobile engine at General Motors and draw conclusions from this study on the utility of our methodology in the management of the integration problem.

2. Our Approach to the Integration Problem

Our approach to tackling the integration problem is based on two premises. The first is that the most effective means to integrate separate teams is to facilitate and ensure strong and direct communication links across the project organization. The second is that in order to do this in the most effective and efficient manner possible, one must have an accurate understanding of the necessary information transfer. In other words, before trying to get specific people to communicate with one another, we must first understand who needs to talk to whom. It is this driving need which motivates us to use a novel project modeling tool known as the Design Structure Matrix (DSM). Modeling information flow in a project using the DSM is our first step in developing project organizations and mechanisms to facilitate necessary information flow within an organization.

2.1 The Design Structure Matrix

The Design Structure Matrix is a project modeling tool which represents the relationships among project tasks in a matrix form. The information captured in a DSM is similar to that in a directed graph or a PERT chart, yet the matrix representation allows for a more complete model of information flow and is a tool more capable of describing and analyzing complex projects. This section presents a brief introduction to the DSM and we will later describe how it can be used to aid in the management of the integration problem. (For a more complete overview of the DSM and its utility in addressing other important problems in product development, the reader is referred to Steward [14] and Eppinger *et al.* [4]).

An example DSM of a simple project, in this case the development of a new soda bottle, is shown in Figure 2.1. In this matrix form, an individual task is represented as a row and the corresponding column of a square matrix while the relationships between tasks are represented as marks (**X**s) in the matrix. Each mark indicates the need for information to flow between two tasks. Reading across

		A	B	C	D	E	F	G	H	I
Perform Market Research	A	A								
Select Bottle Material	B	X	B	X	X		X		X	X
Design Bottle Shape	C	X	X	C		X	X		X	X
Select Cap Material	D	X	X		D	X		X	X	X
Detail Cap Geometry	E	X		X	X	E		X	X	X
Develop Bottle Mold	F		X	X			F			
Design Cap Mfg. Process	G				X	X		G		
Layout Assembly Process	H		X	X	X	X			H	
Test Cap Sealing	I	X		X		X				I

Figure 2.1 Example Design Structure Matrix for the Development of a New Soda Bottle

a task's row indicates from which other tasks information must be received in order to complete the given task. For instance, in the example, the marks in row **F** are in columns **B** and **C**, indicating that the design of the bottle mold requires input from the design of the bottle material and shape. Such information could be the complete output of a task or just a piece of the information generated therein. Conversely, the marks in a task's column indicate which tasks require its output. The example DSM indicates that tasks **C**, **D**, **F**, and **H** all depend on task **B** for information. Diagonal elements do not convey any meaning at this point since a task cannot depend on itself.

The DSM representation has many advantages over traditional project modeling tools. Its compact size makes it practical for modeling complex projects and the matrix form allows for computer manipulation and analysis. The most important advantage, however, lies in the ability to accurately describe complex relationships that are so common among product development tasks. Unlike the PERT technique, the DSM allows for tasks to be coupled, or interdependent (tasks **B** and **C** are coupled in the example DSM). This task coupling often represents the most significant challenges to improving development processes. For instance, in the soda bottle example, the model shows that not only does the mold design depend on the design of the bottle, but also that the bottle itself depends on the design of the mold. The need for obtaining input from manufacturing early in the design stage is clearly identified.

Because of these unique capabilities, the design structure matrix is the basis for much research on improving product development processes. Smith and Eppinger [13] use the DSM as the foundation for their model of engineering design iteration. Morelli [11] uses the tool to compare information needs of a project with actual communication patterns in an organization.

2.2 The DSM Methodology

As previously mentioned, our approach to resolving the integration problem begins by using the DSM to generate a complete map of the required information flow within a project. We feel that each mark in the matrix should be regarded as information which must be exchanged and that an important role of project management is to ensure that this information transfer is occurring in the most efficient, timely, and effective way possible. If communication is not naturally occurring where it is needed, then formal mechanisms should be designed into the organization to ensure this communication. Resolving the integration problem in complex product development is therefore a two-step process. The first step is to identify the information flow needs of the project to determine where communication must occur. The second is to design mechanisms into the organization to ensure and facilitate this communication. We will show how the design structure matrix can be used to help with the first step and then suggest several mechanisms and strategies for facilitating the transfer of information as identified in the DSM.

To illustrate our proposed methodology, we revisit the example of the development of a new lap-top computer. The DSM in Figure 2.2 is a model of what the information flow requirements for this project might be. (We have left out the task names as they are unimportant for the purposes of this example.) It should be observed that the tasks are tightly coupled - there is a high level of information transfer required among them. Ensuring the communication of this information among the people responsible for these tasks is an essential role of management. The DSM model can help with this task.

There are numerous means of facilitating information within such a project, but our first inclination (and one commonly recommended in the literature) might be to form one team of people responsible for all tasks. Yet, as is often the case in complex projects, such a team would be too large to be effective at facilitating communication among all the people involved in the development of a laptop computer. A more practical step, however, is to establish several smaller teams around sub-problems where close communication is most essential. Designing these teams and their responsibilities can be made easier by using the DSM to identify groups of highly coupled tasks. By swapping rows and columns of the matrix, dense "blocks" of marks can be formed along the diagonal. These blocks indicate where high levels of information flow are required and where teams are most appropriate.

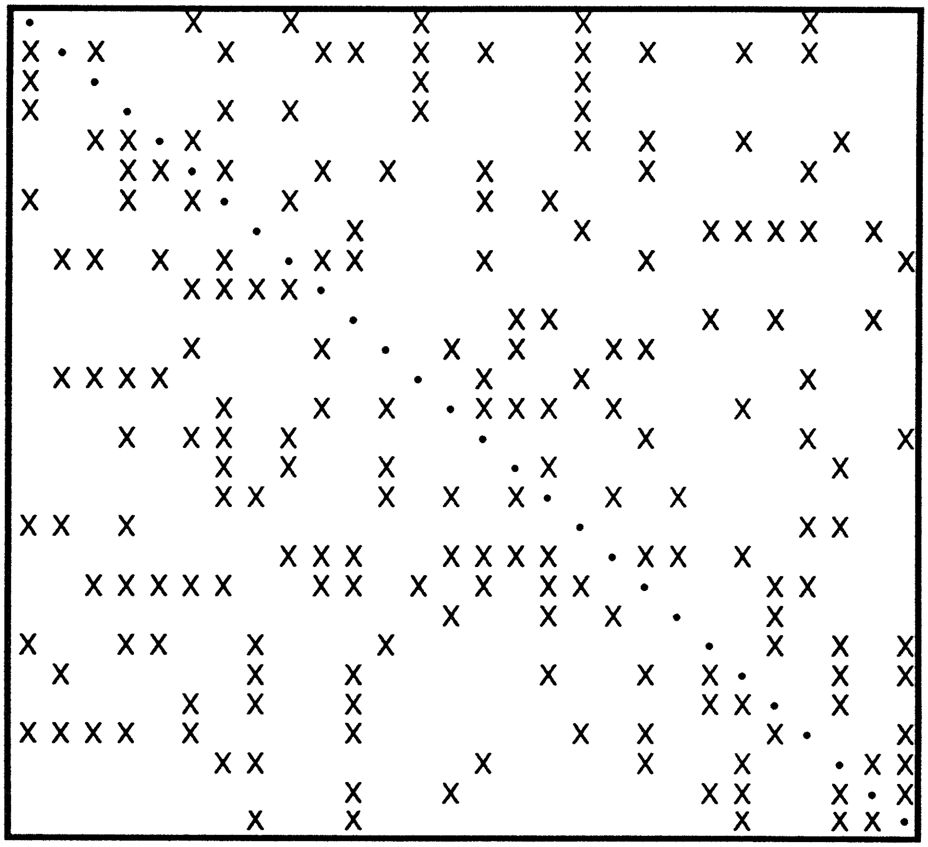


Figure 2.2 Example DSM of Laptop Computer Development

Figure 2.3 shows the results of this process for the laptop example. As can be seen, reconfiguring the DSM identifies four blocks of coupled tasks, each block corresponding to a major component or sub-system of a laptop computer. Establishing a separate team around each of these four blocks is therefore an effective first step toward facilitating information transfer for this project. The teams ensure a high level of communication where it is needed most.

These four teams are not sufficient, however, to facilitate all of the information flow in the project; the DSM clearly shows that communication is also required across team boundaries. The next step, therefore, is to design mechanisms into the organization to facilitate this inter-team communication. We will present several means of doing this in the next section, but we must emphasize that the DSM model should first be used to determine where close communication between teams is most needed. It is by providing a map to direct these efforts that the DSM can be most helpful in tackling the integration problem. In the example, the matrix model indicates that a strong communication link should be established between the drive system team and the main board team whereas the interface between the drive system team and the LCD screen team can be largely ignored. Furthermore, it is evident that the packaging team must establish close links with each of the other three teams. Once all of these needs have been identified, appropriate steps can be taken to meet them. We will now focus on methods of forming strong communication links between development teams.

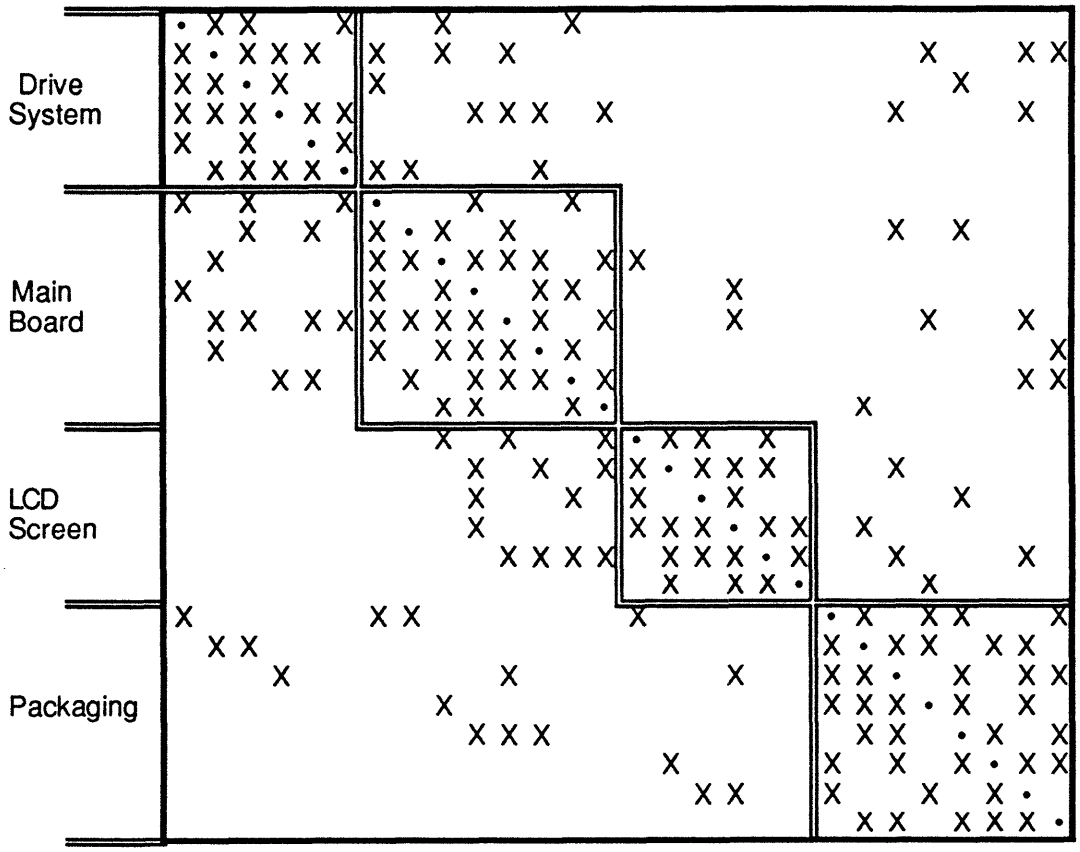


Figure 2.3 Reconfigured DSM Showing Main Blocks of Tasks

3. Mechanisms for Integrating Teams

In this section we provide an overview of several mechanisms that can be used to effectively coordinate actions across separate development teams of a project. Each mechanism may not be appropriate for every development effort. Methods of integration should be carefully chosen based on the technical information needs of a project and the organizational environment of the firm.

Information and Communication Technologies

Improved information and communication technologies (such as linked CAD tools, shared databases of engineering information, e-mail, and voice mail) can serve to break down common barriers to communication and to increase the capacity of an organization to transfer information [6, 7]. Whitney [18] points to many examples where innovative CAD tools are being successfully used to facilitate concurrent engineering in complex development projects. Though this approach may increase information transfer, it might not be sufficient for coordinating team activities since the transfer of the most essential and difficult information is not assured.

Management Hierarchy

Challenging technical conflicts and interface issues between development teams are sometimes resolved by referring them upward to a level of management common to both teams. Communication between teams is achieved through management rather than directly between the working engineers. This traditional "up-over-down" approach is somewhat effective for relatively stable and certain development efforts. The management hierarchy quickly becomes overloaded and ineffective as the degree of uncertainty or the level of complex coupling between teams rises [5].

Heavy Weight Project Manager

Clark and Fujimoto [1] have popularized the concept of a heavy weight project manager as an effective means of integration. They define this role as a manager who has "direct access to the working-level engineers" and who exercises "strong direct and indirect influence across all functions and activities in the project." (This role is also described by Lawrence and Lorsch [9] as an "integrator".) A clear vision of the overall project needs and a devotion to play an active, central role in coordinating the development effort make heavy weight project managers effective integrators.

Conflict Resolution Engineers and Liaison Roles

Conflict resolution engineers are individuals dedicated to act as arbitrators between particular development teams. They typically do not hold responsibilities for any development tasks other than to facilitate the resolution of technical conflicts that arise between the teams. An example of this mechanism can be found in the platform team organization at a division of a U.S. automaker where thirty-five product development teams are involved in car development (such as doors & hardware, chassis, and body structures). To help resolve technical conflicts between teams, the car has been broken down into five major zones -- front end, instrument panel, interior trim, doors, and rear end. One "zone engineer" is assigned to each of the five zones and is responsible for arbitrating technical conflicts between teams that concern their zone of the vehicle.

Liaisons are similar to conflict resolution engineers yet play a more proactive role in resolving inter-team technical issues. Whereas conflict resolution engineers address conflicts that have been brought to their attention by the teams, liaisons are charged with facilitating continuous and intensive information exchange between particular teams. Technical conflicts are hopefully discovered earlier and resolved faster.

System Engineering

The concept behind system engineering is to ensure the integrity of the overall product by imposing system-level technical specifications on all teams and actively monitoring the development of the product as a system. A group of "system engineers" is typically established to help refine voice-of-the-customer data to system-level specs and to work directly with the component development teams to ensure that these specs are met. System engineers are also often responsible for maintaining and updating central files of component technical specifications and design decisions. [10]

Task Forces

Task forces are a common and effective method for tackling project-wide technical conflicts of a particularly complex and critical nature. Engineers and other personnel are drawn from across the project organization to form a special group to focus exclusively on a single technical issue that concerns several teams. Task forces are formed on an as-needed basis as challenging issues arise and they exist only as long as the issue remains unresolved. A task force was effectively used in the GM engine development project that serves as the focus of this study to address the issue of engine balance, an issue that spanned several product development teams.

Town Meetings

In this mechanism, all project personnel, from all teams, meet occasionally to review the project's progress. Town meetings are useful for improving morale and camaraderie, but are generally ineffective for large amounts of technical information transfer.

Technically Independent Teams

The task of coordinating and integrating teams can be eased if teams are formed with the goal of minimizing necessary technical interactions among them. A common way to do this is to form teams around separate components of a product rather than by functional discipline. The more modular a product's architecture, the more independent the teams. Other means of reducing necessary inter-team interaction may be found for some projects. In this paper, we will demonstrate how the design structure matrix can be used to identify independent groups of teams for the engine development project at General Motors.

Engineering Liaisons

Engineering liaisons are individuals that are formally made members of two or more development teams in order to facilitate information transfer between the teams. They differ from traditional liaison roles in that they are responsible not only for establishing and maintaining a firm communication link between the teams, but also for performing specific technical tasks on at least one of the teams. They are working development engineers. This special dual relationship allows for more direct and rapid response to technical inter-team conflicts through the development of a mutual understanding between the teams.

The key to effectively utilizing these integration mechanisms in development projects is to understand where they are most needed. The DSM-based methodology presented in this paper allows managers to identify the needs for integration based on the technical structure of the project. The rest of the paper presents a study of an engine development project at General Motors for which the DSM methodology was useful for designing mechanisms to integrate several teams.

4. Engine Development Project

The complete redesign of a small block V-8 automotive engine at General Motors served as a test-bed for this research. Our goal was to study the many issues of concurrent engineering in a large industrial project and to test our proposed methodology for addressing the integration problem using the design structure matrix. After describing important elements of the GM development effort, we will present a DSM of the project and use it to make recommendations to management for addressing the integration problem.

4.1 Project Scope and Organization

The development of a new automobile engine is typical of large and complex engineering projects. Large numbers of parts and sophisticated product and manufacturing technologies mandate the involvement of several hundred people. Furthermore, a highly integral architecture and tightly coupled systems demand close coordination across the entire development organization.

For the project at hand, which at the time of this study was in the earliest stages of development, a large product engineering group is responsible for completely redesigning nearly all major components of the engine. (An estimated 90% of the parts are to be redesigned.) This group of experienced design release engineers and CAD designers are members of a larger engineering organization responsible for engine design for all of GM. All reside on one floor of a large building and they are generally quite familiar with one another as most have been working for this engine design organization for several years.

An even larger manufacturing organization is involved. In addition to all new tooling required by new part designs, several unfamiliar processes must be developed and validated and entire production lines must be built from the ground up. (An estimated 80% new manufacturing equipment will be used.) Most of the manufacturing engineers work at one of the component manufacturing or final assembly plants, which are spread out throughout Canada and the United States. A core manufacturing team, however, is set up down the hall from the product engineers. This team is comprised of at least one engineer from each of the plants and each of the separate production lines to be built. They generally split their time between their plants and the core team office. Management has dubbed this team the simultaneous engineering team since its main purpose is to interact with the product engineers, thereby facilitating concurrent engineering and design for manufacture. Add purchasing, financial, a variety of engineering specialist groups, and a host of component, tooling, and equipment suppliers and it is clear that this project entails ample scale and complexity for the purposes of this research.

4.2 Product Development Teams

To execute this complex project, 22 product development teams (PDTs) have been established around the major components and sub-systems of the engine. Figure 4.1 identifies these teams and gives an example of their composition. Though termed "teams", PDTs are teams only in that they have been identified as such and hold regular meetings (typically once a week during the detailed stages of design). Team members do not sit together and do not share responsibilities as would members of true teams.

It should be observed that PDTs have multi-functional membership and that they are primarily intended to generate and improve communication between the various people working on the development of the same system, from CAD designers to purchasing representatives. This has been verified with both the teams and management as most tell us that PDT meetings generally focus on design for manufacture (DFM) issues and interfaces with purchasing and financial.

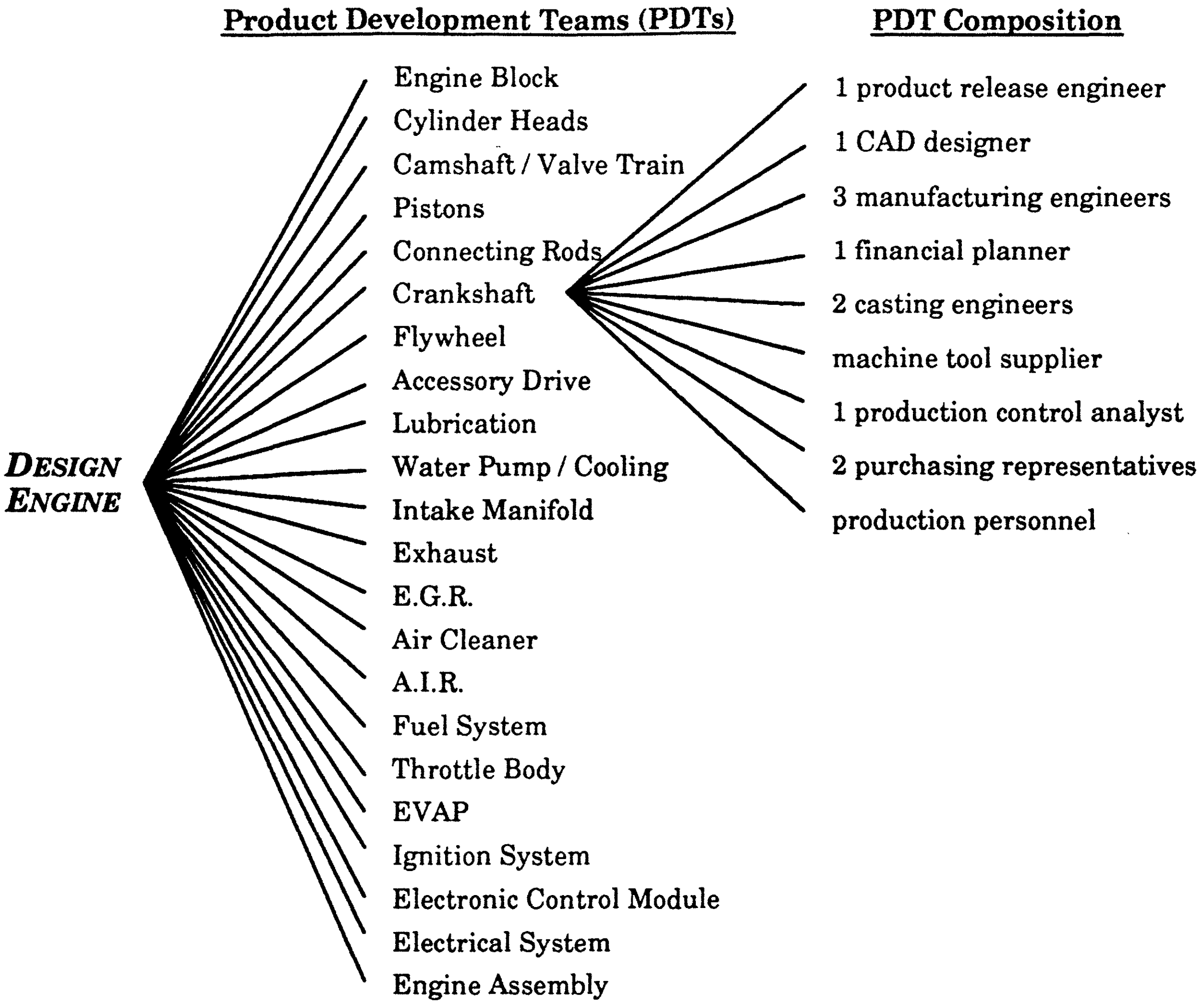


Figure 4.1 Product Development Teams for the Engine Development Project

Product development teams in this project are therefore an organizational approach for facilitating the overlapping of development tasks. We do not question the existence or the design of these teams. In fact they appear to be well chosen and an effective means of addressing the overlapping problem. Instead we focus on the need to ensure and facilitate information exchange *between* PDTs. The issue of integrating teams is one that project managers recognize to be important and have taken steps to address, yet find quite difficult to completely resolve.

4.3 Integration Efforts

The systems and components of an engine are highly coupled and cannot be developed independently of one another. PDTs must interact. Communication between teams is most essential within the product engineering group where countless individual part designs must be integrated into a well-designed system. Interaction is also required between manufacturing engineers on different PDTs who must work together to standardize equipment and tooling within production plants. Other functions, such as purchasing and financial, have a lesser need for integration.

Informal communication plays an important integrating role in this project as engineers are aware of the need for integration and will naturally communicate across PDT boundaries when performing their tasks. For instance, the product engineer designing the engine block will consult with the engineer designing the crankshaft to determine the diameters and locations of the journal bearings for the crankshaft. Such informal communication is especially common within the product engineering group where not only are interactions most essential, but also communication is facilitated by the engineers' close proximity and established relationships.

Relying solely on such an informal communication network for integration, however, means to depend on the engineers to comprehend and initiate all of the necessary interactions between PDTs. Unfortunately, engineers are rarely sensitive to all inter-PDT relationships, especially concerning how their work affects the work of other PDTs. Furthermore, many unforeseen conflicts between PDTs arise throughout the course of the project which are too slowly resolved through informal integration. For instance, the design of a crankshaft speed sensor by the electrical system team may significantly increase the cost of the engine block manufacturing line. If there is not a strong link between the electrical sensor designer and the engine block manufacturing engineer, this issue could remain hidden until significant rework is required. More formal, planned integration mechanisms must be designed into the organization to ensure necessary information exchange between PDTs and to expose and resolve inter-PDT issues as early and as quickly as possible. The project managers understand this need and have taken steps to better integrate the PDTs.

4.4 System Teams

The project managers have grouped the twenty-two PDTs into four *system teams* as shown in Figure 4.2. System teams are nothing more than groups of PDTs that are required to meet together with management every other week. The main purpose of these meetings is to serve as a forum for engineers and managers to raise and discuss technical conflicts between the PDTs of a system team. In every meeting, all PDTs are required to report on their activities. This not only allows management to review the progress of the project, but also ensures that the PDTs develop a common understanding of each others' desires and intents. It is hoped that developing such a common understanding will help to expose conflicts between teams before significant rework is required to resolve them.

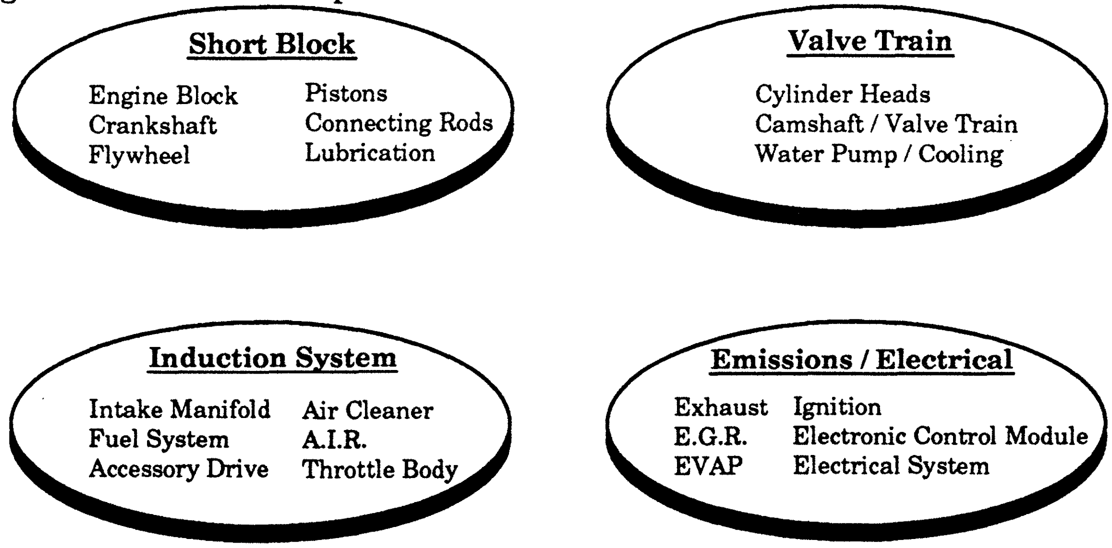


Figure 4.2 System Teams

Though these system team meetings can be very large (up to fifty people) and inefficient, management claims that the meetings have been effective at surfacing challenging inter-PDT issues that otherwise would not have been discovered nearly as early. They are especially encouraged when they compare the progress of this program to previous engine development programs in which system teams were not used. The managers realize, however, that these meetings cannot address all conflicts between PDTs; issues will certainly arise that concern PDTs on different system teams. They are now searching for an effective means to integrate the system teams.

The challenge of integrating these twenty-two product development teams poses a good test for our proposed methodology. Our first step is to investigate the interdependence of these PDTs and to map the required information flow between them with the design structure matrix. After presenting a DSM of this project, we will demonstrate how such a representation can lead to better ways to integrate the PDTs.

4.5 DSM Model

To focus on the coupling of PDTs, we have constructed a DSM in which each team is represented as a single development task. For example, all of the activities of the intake manifold team, from material selection to vendor relations, are represented as one row in the matrix labeled "Intake Manifold". The information transfer from one team to another is therefore captured as a single mark in the matrix. To characterize PDT interdependence further, the necessary (or anticipated) frequency of information exchange is identified as either high, average, or low. High frequency is defined as several times per week, while average means at most weekly and low indicates infrequent yet essential information transfer. The DSM is shown in Figure 4.3.

The proper relationships between PDTs were identified by directly questioning the engineers and their managers, both from the product and manufacturing engineering groups. At least one engineer from each team (generally the product release engineer) responded to a questionnaire that asked them to rate as high, average, low, or zero the frequency at which they need to get information from each of the other PDTs in order to complete the technical tasks of their own PDT (equivalent to one row in the matrix). Several managers were also separately asked to define how they believed the teams should ideally interact. Gathering both sets of data was essential as engineers were inclined to disregard significant interactions that were not currently occurring and the managers sometimes missed important technical coupling. Significant cases of disparity were resolved by further consultations with the managers. (A complete description of the process used to construct the DSM is presented in Appendix A.)

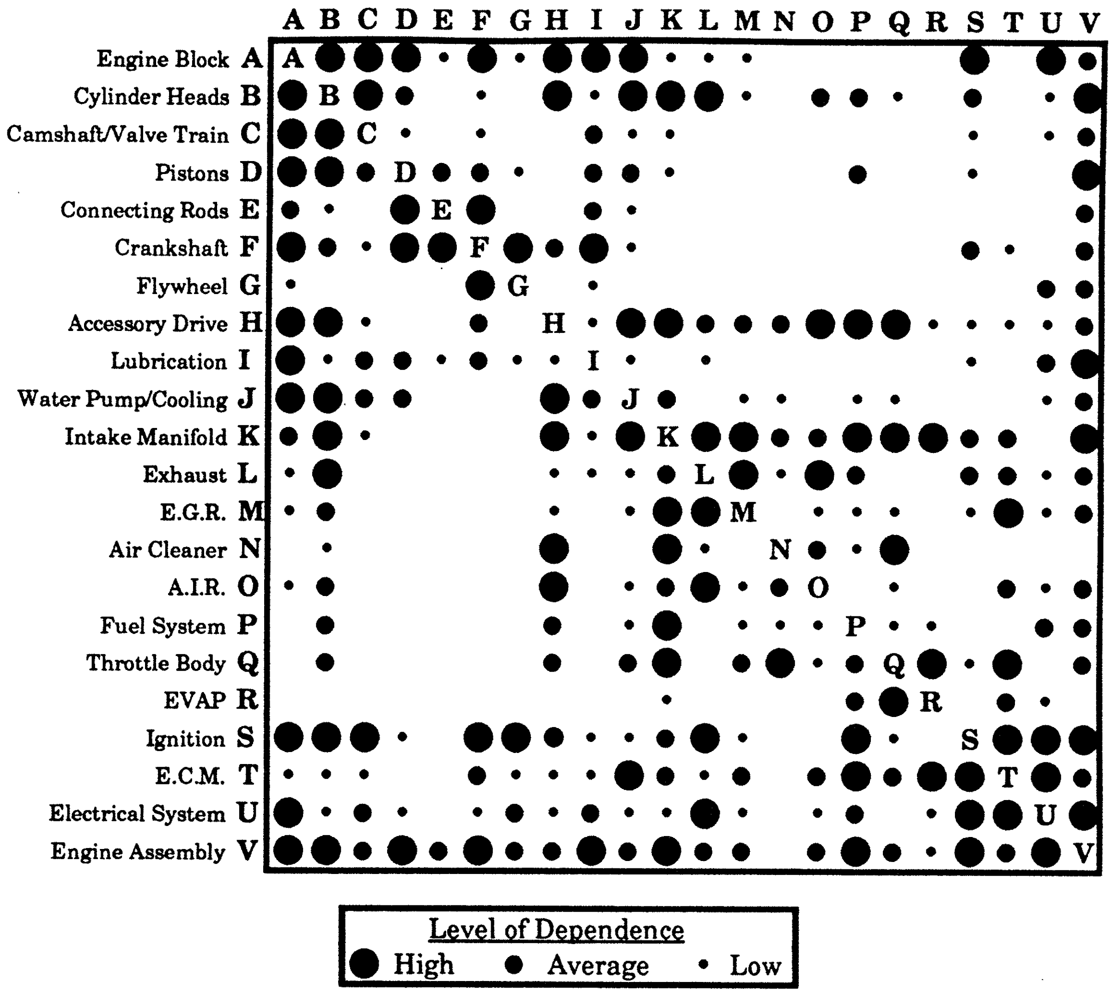


Figure 4.3 Design Structure Matrix of the Engine Development Project

The DSM clearly shows why integrating the product development teams is a difficult task - the teams are highly coupled in many complex ways. Our search for better integration mechanisms begins by using the model to examine the design of system teams.

4.6 Existing System Teams

Management's effort to integrate the PDTs through system teams is captured in the DSM shown in Figure 4.4. We have clustered the PDTs in the matrix to show the information transfer required both within and between system teams. This mapping graphically shows both the benefits and the shortcomings of the system teams defined by the managers. It can be seen in the matrix that although system team meetings ensure interaction among many of the coupled PDTs, significant information exchange is still required between PDTs on different system teams. For example, note the strong interactions required between the cylinder heads PDT (**B**) and members of the induction system team. Such inter-system team information flow has been difficult for the managers to facilitate. The next section shows how the DSM can be used to explore better ways to group the PDTs into system teams.

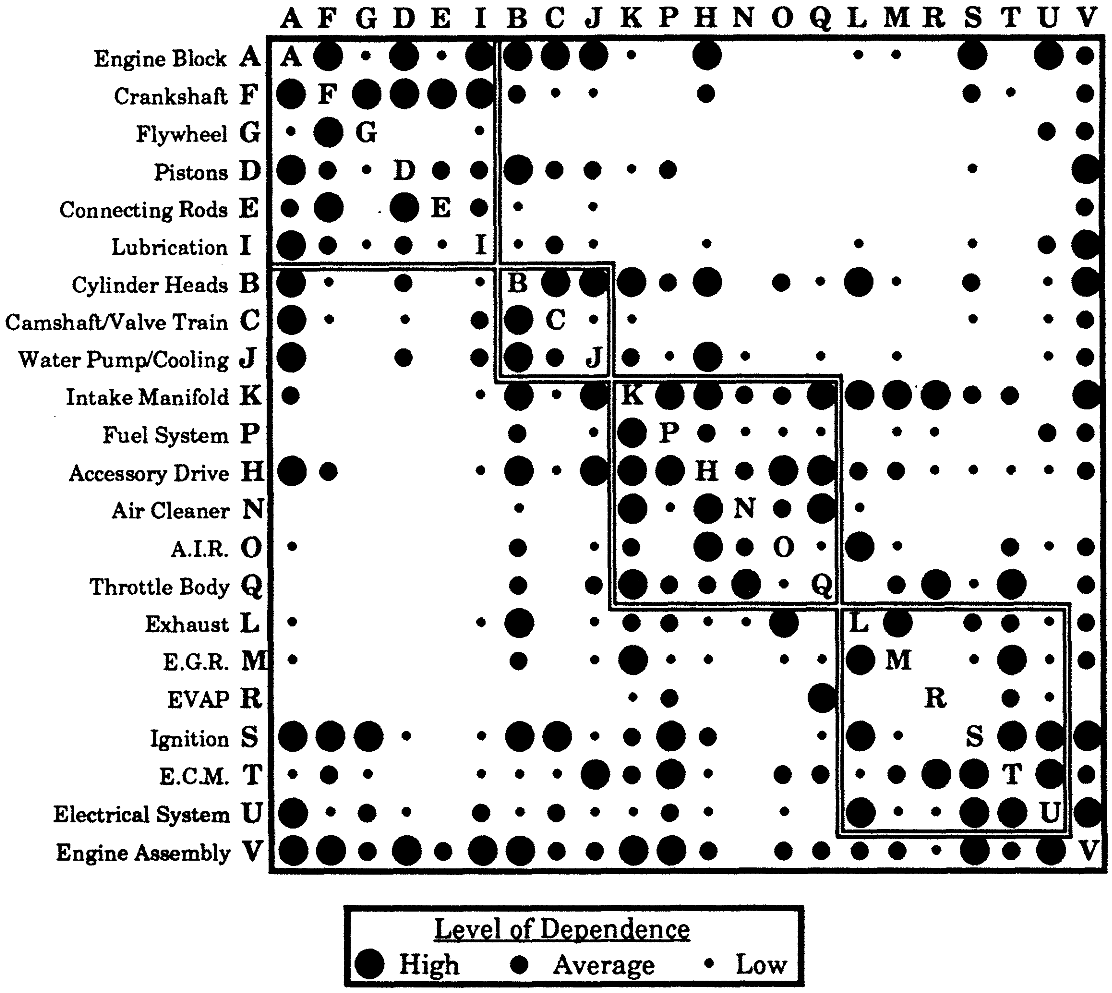


Figure 4.4 Ordered DSM Showing Existing System Team Structure

4.7 Reorganization of System Teams

The matrix in Figure 4.5 shows a new structure for the system teams which reduces interactions between the system teams and improves integration across all the PDTs. The primary mechanism used to integrate the teams is to overlap them by making some PDTs formal members of more than one system team. Furthermore, five PDTs - accessory drive, ignition, electronic control module, electrical system, and engine assembly (**H**, **S**, **T**, **U**, and **V**) - are not included in the system teams. We will explain each mechanism in turn.

The layout of the proposed system teams is clearly illustrated in Figure 4.6. The first two system teams are centered around the same PDTs as the original system team structure (refer to Figure 4.2). The third and fourth system teams have been restructured so that the third team focuses more closely on the fuel injection and air intake systems while the fourth team is primarily concerned with the exhaust side of the engine. The most significant change, however, is the overlapping. The pistons, lubrication, and engine block PDTs (**D**, **I**, and **A**) each actively participate in the meetings of both system teams 1 and 2. The water pump/cooling PDT (**J**) is active on both system teams 2 and 3. The cylinder heads and intake manifold PDTs (**B** and **K**) each have responsibilities on the last three system teams. (Note that in order to represent one PDT on three system teams we split its interactions into two separate rows in the DSM (**B1/B2** and **K1/K2**). Only the interactions with teams in the immediate system team, or teams, are represented in each row and column.)

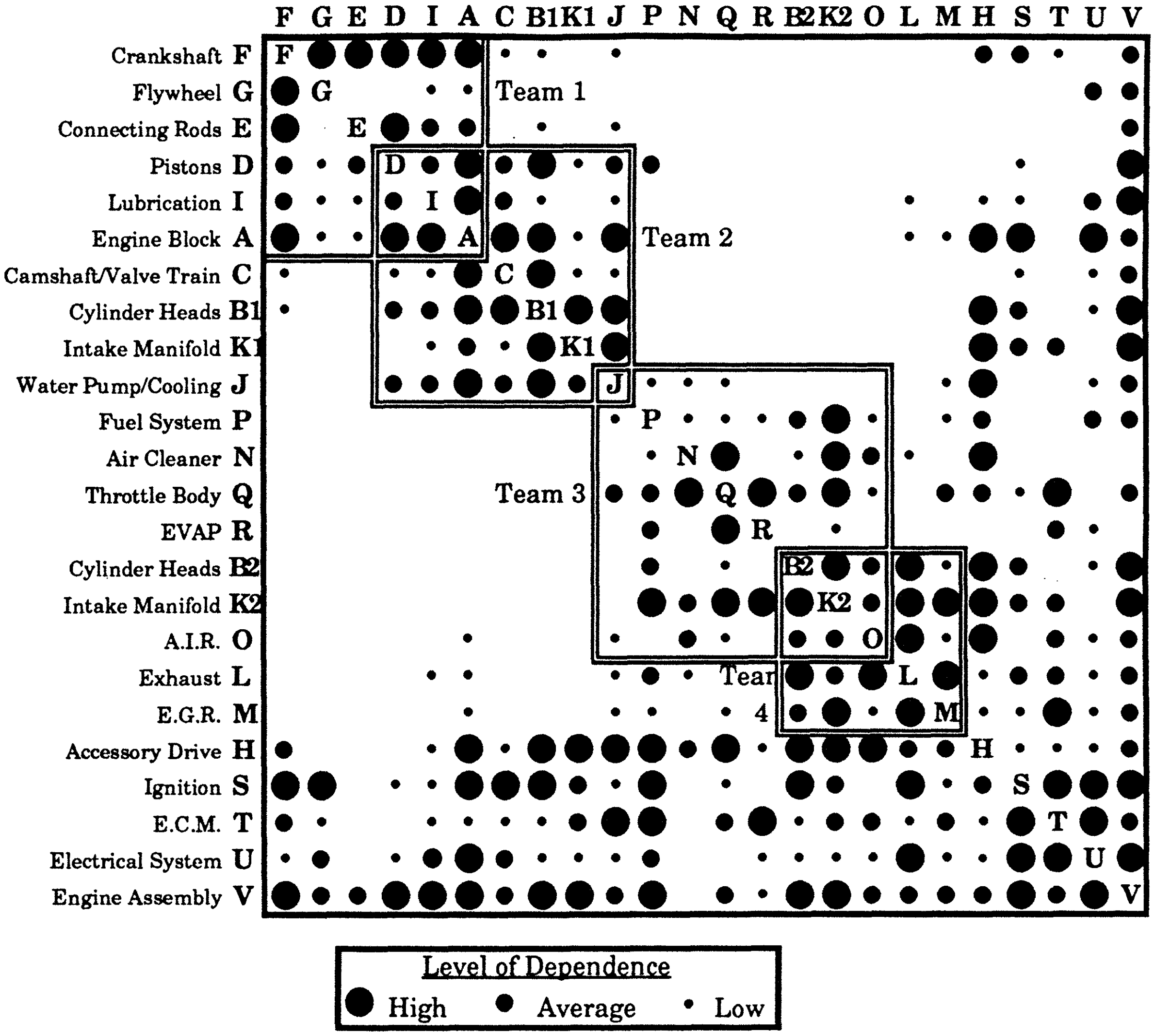


Figure 4.5 DSM Showing Proposed Design for System Teams

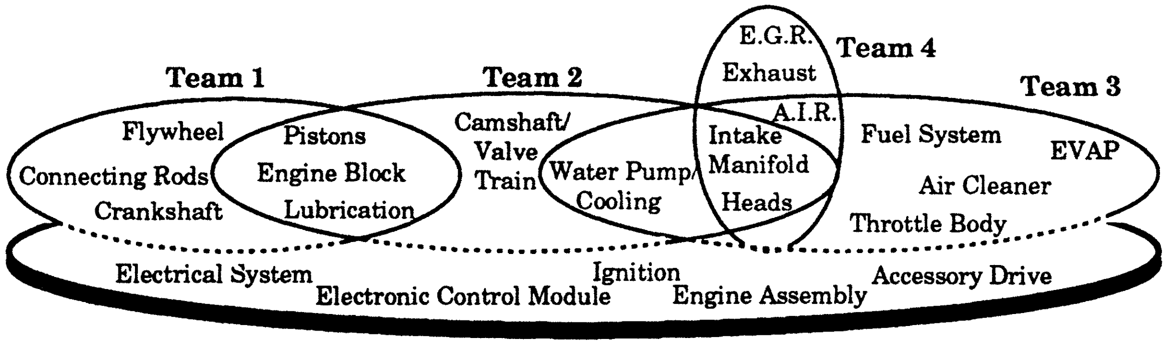


Figure 4.6 Proposed Design for System Teams

The objective of reconfiguring and overlapping teams was to envelope more of the highest inter-PDT dependencies in the project into the system teams structure while still maintaining relatively small system teams. We have been successful in this regard as there are no identifiable high dependencies between system teams and the size of teams has been kept to under eight PDTs. There remain some required interactions between system teams, yet they are mostly of low dependence. We do not attempt to encompass these minor interactions through more overlapping as there are more efficient means to facilitate this information transfer. Engineering liaisons might be effective, for instance. Exposing the working engineers to the DSM mapping might also result in more informal communication between system teams by making engineers aware of the need for information transfer.

The five PDTs at the bottom of the matrix (**H**, **S**, **T**, **U**, and **V**) have been set apart to show that they do not fit into any one system team but rather require interaction with nearly all of the other PDTs. They play an important integrating role in the project. One effective way to ensure this integration is to formally incorporate these PDTs into each of the system teams (as is shown in Figure 4.6). This effort is facilitated by the fact that each of these PDTs has several product release engineers responsible for different elements of each system. Assigning a different engineer to each system team may be a more efficient way to integrate these PDTs across the project than requiring all members to attend all system team meetings.

Another possibility is to create an "engine integration" system team which would include the five highly integrative PDTs plus major representatives from each of the other four system teams, such as block, heads, and intake manifold. This new team would focus solely on global engine issues.

4.8 Matrix Analysis

Two separate techniques were found to be effective for analyzing the DSM model in search of a better design for the system teams. The first is a heuristic method which relies on visual inspection and logical reasoning while the second method employs a mathematical algorithm. The objective of both methods is to identify sets of tasks PDTs that are highly coupled and should be grouped into system teams.

Studying the interactions for each PDT and swapping the rows and columns in a trial-and-error fashion is a reasonably effective means for identifying highly coupled blocks in a matrix this size. Representing the DSM in a computer spreadsheet greatly facilitates this matrix manipulation. The first step is to identify and remove from the matrix the PDTs that are highly coupled across the entire project, such as engine assembly and accessory drive. This serves to isolate the interactions among the rest of the

PDTs. The interactions for each PDT are then separately examined to determine which teams should be grouped together according to their needs for information transfer. This heuristic process was successfully used to identify the system teams as defined in Figure 4.5.

The second technique is based on the Work Transformation Model developed by Smith and Eppinger [13]. In this model, the marks in the DSM are replaced by numerical values which are scaled to represent the strength of dependency between tasks. For example, in the engine redesign DSM, 0.1 was used for high dependencies, 0.03 for average, and 0.01 for low. As before, the highly integrative tasks must also be removed prior to the analysis. The eigenvalues and eigenvectors of the resulting matrix are then analyzed to reveal the highly coupled sets of tasks (or PDTs). The largest elements in each eigenvector characterize a separate *design mode,* defined by Smith and Eppinger as a "group of design tasks which are very closely related." The dominant design modes (the most highly coupled sets of tasks) correspond to the eigenvectors with the largest eigenvalues. (For a complete description of this model, the reader is referred to Smith and Eppinger [13].)

Applying this technique to the engine redesign DSM revealed major design modes which roughly match the groupings identified by the heuristic method. (The complete analysis is presented in Appendix B.) Though more research needs to be done to verify this method, the eigenstructure analysis is a promising means of identifying logical groupings for teams. This technique would be especially valuable for very large projects and matrices for which heuristic manual manipulation is impractical.

5. Discussion

The results of our involvement with this engine development project clearly demonstrate the benefits to be gained by using a DSM mapping to plan coordination across parallel development activities. Whereas the project managers relied on intuition and prior development experiences to design integration mechanisms, we based our recommendations on a clear and concise documentation of the necessary information transfer in the project. The result was an improved layout for the system teams which serves to address more inter-PDT issues without unnecessarily burdening the PDTs.

5.1 Overlapping System Teams

The DSM mapping motivated the concept of overlapping the system teams as a way to facilitate more of the important inter-team interactions. It was also instrumental in determining the most effective ways to overlap the system teams. The matrix in Figure 4.5 clearly shows, for example, that the engine block PDT not only has a need for significant interaction with the PDTs of the short block system team, but also with the valve train system team. This graphic mapping led us to the concept of formally incorporating the engine block PDT into both of the first two system teams. The other overlaps were identified in a similar manner. Though the managers had naturally selected a mutually exclusive set of system teams, they may have considered overlapping the teams if they had initially developed a complete model of inter-PDT interactions.

Identifying the PDTs which require strong links across the entire project organization was another insightful result of modeling the project with the design structure matrix. These highly integrative PDTs require an extreme form of overlapping across the four system teams.

5.2 Application to Smaller Project Organizations

In our study of the engine development project, we limited our focus to the interactions required among product development teams and to the design of system-level teams. It should be clear, however, that the basic DSM methodology presented here can help to ensure and facilitate information transfer at more detailed levels of information flow. It may be especially valuable for the design of smaller project organizations and the formation of working teams of individuals (such as PDTs). Consider, for example, if the DSM of the engine redesign project captured relationships between engineers rather than between teams. Groupings identified as in Figure 4.5 would then depict an efficient layout for the first level of engineering teams.

5.3 Robustness to Inaccuracy in the DSM Model

Because it is difficult to accurately predict all future information exchange, the DSM model will undoubtedly misrepresent some interactions between PDTs. The work of Morelli [11] suggests, however, that most of the *essential and frequent* communication links that develop over the course of a project can indeed be predicted beforehand. The matrix model will in most cases differ only slightly from the actual information flow; rarely will significant interactions be left out. For this reason, we believe that the results for the engine redesign project are robust to small inaccuracies in the DSM model. That is, slight changes in the dependencies between PDTs will not affect the overall design of the system teams. This is because the goal is to incorporate all of the information transfer within a system team, whether high or low. Unpredictable information transfer must be accommodated through other integration mechanisms which foster less formal interactions, such as situating engineers near one another or establishing engineering liaisons.

6. Conclusions and Future Research

We have shown how the design structure matrix can be effectively used to direct the process of facilitating the essential information transfer in complex engineering projects. The DSM is a concise, purely descriptive representation of information flow in a project. It allows mechanisms for promoting and ensuring communication to be designed based on the specific technical information needs of the project rather than on managers' intuitive best guesses. The DSM mapping does not prescribe specific integration mechanisms; it indicates where such integration efforts must be focused.

The practicality of using the DSM for this purpose in an industrial setting was demonstrated by modeling an engine development project and recommending significant changes to the project organization based on the DSM model. Our involvement with the engine development project at General Motors also confirmed that the issue of integrating several teams is important and worth addressing in complex development projects.

Although we have demonstrated the advantages of constructing a DSM model in order to address the integration problem in concurrent engineering, the methodology must be further refined. One important next step is to develop more robust mathematical algorithms to identify the optimal groupings of tasks in a matrix. Such algorithms must allow for the groups to be overlapped. Another direction is to develop more effective and efficient mechanisms for integrating individuals and teams. Of most importance, however, is to test the practicality and usefulness of this approach by continuing to model and study complex industrial projects.

References

[1] Clark, Kim B. and Takahiro Fujimoto. *Product Development Performance: Strategy, Organization, and Management in the World Auto Industry*. Harvard Business School Press, Boston, MA, 1991.

[2] Clausing, Don. "Concurrent Engineering", Paper presented at the Winter Annual Meeting, *American Society of Mechanical Engineers.* San Francisco, CA, December 13, 1989.

[3] Dean, James W., and Gerald I. Susman. "Organizing for Manufacturable Design", *Harvard Business Review* . January-February 1989.

[4] Eppinger, Steven D., Daniel E. Whitney, Robert P. Smith, and David A. Gebala. "Organizing the Tasks in Complex Design Projects", *ASME Conference on Design Theory and Methodology,* Chicago, IL, September, 1990, pp. 39-46.

[5] Galbraith, Jay R. *Designing Complex Organizations.* Addison-Wesley, Reading, MA, 1973.

[6] Hauptman, Oscar, and Thomas J. Allen. "The Influence of Communication Technologies on Organizational Structure: A Conceptual Model for Future Research", *Working Paper, MIT Sloan School of Management.* 1987.

[7] Jakiela, Mark J., and Wanda J. Orlikowski. "Back to drawing board? Computer-mediated communication tools for engineers," *Working Paper, MIT International Center for Research on the Management of Technology.* 1990.

[8] Krishnan, Viswanathan, Steven D. Eppinger, and Daniel E. Whitney. "Overlapping Product Development Activities by Modeling Information Exchange", *Working Paper, MIT Sloan School of Management.* forthcoming.

[9] Lawrence, Paul R., and Jay W. Lorsch. "New Management Job: The Integrator", *Harvard Business Review.* November-December 1967.

[10] Mattis, David P. "Case Study on the Strategies of Systems Engineering", *Master's Thesis.* MIT Sloan School of Management, 1992.

[11] Morelli, Mark D. "Evaluating Information Transfer in Product Development", *Master's Thesis.* MIT Sloan School of Management, 1993.

[12] Nevins, James L., Daniel E. Whitney, and Thomas De Fazio. *Concurrent Design of Products and Processes: A Strategy for the Next Generation in Manufacturing.* McGraw Hill, New York, 1989.

[13] Smith, Robert P., and Steven D. Eppinger. "Identifying Controlling Features of Engineering Design Iteration", *Working Paper, MIT Sloan School of Management.* 1987.

[14] Steward, Don. "The Design Structure System: A Method for Managing the Design of Complex Systems", *IEEE Transactions on Engineering Management.* August 1981.

[15] Takeuchi, Hirotaka and Ikujiro Nonaka. "The New New Product Development Game", *Harvard Business Review.* January-February 1986.

[16] Wheelwright, Steven C., and Kim B. Clark. *Revolutionizing Product Development, Quantum Leaps in Speed, Efficiency, and Quality.* The Free Press, A Division of Macmillan, Inc., 1992.

[17] Whitney, Daniel E. "Manufacturing by Design", *Harvard Business Review.* July-August 1988, pp. 83-91.

[18] Whitney, Daniel E. "Electro-Mechanical Design in Europe: University Research and Industrial Practice", ESNIB. vol. 93-01, pp 1-52.

Appendix A. Constructing the DSM for the Engine Development Project

The DSM of the engine development project was constructed from the input of both engineers and their managers concerning the necessary level of interaction between product development teams. The primary mechanism for collecting these data was the survey shown in Figure A.1. The surveys ask for a characterization of the dependence of each PDT on information from each of the other PDTs in the project. Each individual survey captures the information in one PDT's row of the DSM. The level of dependence must be identified as either high, average, or low based upon the required frequency of information exchange. High dependence indicates the need to interact several times a week, whereas average means weekly or bi-weekly interaction and low is infrequent yet important information transfer.

The data collection process began by directly questioning two high-level managers about all PDT interactions. In a three hour interview, the two managers together completed one survey for each of the twenty-two PDTs in the project, enough to fill the DSM. One manager is a part of the product engineering group, two steps above the working engineers. The other is the manager of the core group of manufacturing engineers (dubbed the "simultaneous engineering team") that is located down the hall from the product engineers. Together they have a great deal of experience with engine development, especially with the base engine. (The base engine is the engine block, the cylinder heads and all the components within them, such as the camshaft, valve train, pistons, connecting rods, crankshaft, and flywheel.)

The next step was to distribute surveys to the engineers on the PDTs. The response was good as at least one product release engineer from each team filled out a survey. Because product release engineers are responsible for the entire design of their PDT's component or system, they are generally quite aware of the necessary inter-PDT information transfer. Manufacturing engineers also responded from some of the base engine PDTs (cylinder heads, pistons, and crankshaft).

The survey "answers" from the engineers often differed with those from the managers. That is, the engineers often characterized the relationships between PDTs differently than did the managers. Furthermore, the responses from two or more engineers on the same PDT were often significantly different. Because our goal was to construct one complete and accurate DSM, we were forced to resolve these cases of disparity. Figure A.2 shows the complete set of data from the engineers and managers as well as our final determination of the required PDT interactions. (The data matrix reads just like a DSM; columns represent output from a team whereas rows represent input to a team.) We will proceed to explain some possible causes for the disparity in the data and to describe the process used to determine the most accurate relationships.

Potential Causes of Disparity and Inaccuracy in the Data

Responses from the managers can be expected to differ with those from the engineers due to the difference in each group's level of focus in the project. The managers might be too removed from the working details to completely understand all of the technical relationships between PDTs. On the other hand, the engineers might be too absorbed in the everyday details to consider the broader, longer-term requirements for information transfer. For these reasons, input was purposefully sought from both managers and engineers in the hope that obtaining both viewpoints might allow PDT interactions to be determined more accurately.

There are several other discernible causes for the disparity in the data. As previously mentioned, the experiences of the two managers interviewed make them especially familiar with the responsibilities and interactions of the base engine PDTs. Unfortunately, however, they are much less familiar with the PDTs developing the control elements of the engine. (Control PDTs are fuel system, accessory drive, air cleaner, A.I.R., throttle body, exhaust, E.G.R., EVAP, ignition, E.C.M., and electrical system.) In fact, they were not directly involved in establishing these teams and are rather unsure of the exact responsibilities of each team. Therefore, though the manager's input was important to obtain, their answers for the inputs to and the outputs from control PDTs should be regarded with some uncertainty.

An additional cause of disparity in the data for the control PDTs is that two significantly different engine control systems are being developed in this project, one for car engines and one for truck engines. (The base engine is the same for both car and truck engines.) The controls PDTs each have one product release engineer responsible for car engine design and one responsible for truck engine design. Survey responses from these engineers should be expected to differ since car and truck control systems entail completely different component technologies and designs. Nevertheless, survey responses were sought from all release engineers since they are all defined to be on the PDTs.

In general, the responses of the engineers appeared to be carefully and deliberately thought-out. However, it is likely that some surveys were filled out in haste and are therefore somewhat inaccurate. Furthermore, one engineer's personal definition of an average dependence might not coincide with another engineer's definition. Calibration across the engineers' responses was not completely assured by the definitions of high, average, low, and zero on the survey. In fact, a few engineers (such as the E.G.R. product release engineer) never marked down a zero dependence, apparently unwilling to suggest that they didn't have at least some need to communication with every other PDT. This lack of calibration was in part a fault of the survey.

Another source of error in the data concerns the electrical system PDT. This PDT is named the "platform" PDT by the project management and was defined as such in the surveys distributed to the engineers. Unfortunately, most engineers were unfamiliar with this name for this PDT and interpreted "platform" team to mean the large platform teams that are responsible for the design and development of an entire car line. Consequently, the responses from the engineers concerning information required *from* the electrical system team (forming the electrical system team column) are not meaningful. Members of the true platform PDT (renamed "electrical system" PDT for this research) were surveyed to accurately determine the input required *to* this team (the electrical system team row).

Determining the Most Accurate Dependencies

If engineers and managers identified the same level of information dependence between two PDTs, their mutual response was taken to be the correct dependency. If there were a range of responses for a specific inter-PDT relationship, more careful examination was required to determine the most accurate dependency.

Each case of disparity in the data was considered on an individual basis. In general, the response indicating the highest level of dependence for a given interaction, whether from the managers or from an engineer, was taken to be the most accurate. It was believed that if an engineer identified a higher level of dependence than did the managers, it was because they had a better understanding of the technical nature of their work than did the managers. Conversely, if the managers indicated a higher dependence than did the engineers, their response was believed because they most likely possessed a more global and progressive outlook on the ideal relationships in the project.

The most prevalent exception to this rule was in the case when it was deemed that an engineer's responses were not calibrated with the rest of the responses. An example of this is the throttle body and E.G.R. engineer's responses, none of which are zero. Examining the rows for these two PDTs reveals that we believed the managers' responses of "zero" even when the engineer's response indicated "low" or "average".

Another helpful method for determining the most accurate PDT dependencies was to compare the data for information transfer between two PDTs in both directions. For example, consider the relationship between the engine block PDT and the cylinder heads PDT. There is disagreement concerning the level of dependence of the block team on information from the heads team. The managers feel this dependence should be high whereas the engine block release engineer believes it to be only average. To resolve this disparity we looked at the data for information transfer in the opposite direction, from block to heads. Both the cylinder heads product release engineer and the high-level managers agree that the heads PDT has a high dependence on information from the block PDT. This led us to believe that the engine block release engineer is downplaying or under representing the block PDT's dependence on information from the heads team. We therefore chose "high" to be the most accurate dependency in this case. Examining data on the opposite flow of information in this way was often helpful for determining the most likely interactions.

Input from the two high-level managers concerning information transfer to and from the controls PDTs was largely ignored due to their unfamiliarity with the control systems of the engine (though it should be observed in data matrix that these managers were generally quite accurate with their assessment of these interactions). The controls engineers' responses were generally taken to be accurate. When the responses from two or more engineers differed significantly, however, the immediate managers of the engineers were directly consulted to help identify the best answer. The input from these managers is represented as a 4 in the data matrix and was generally believed to be the best characterization of the interaction.

Responding to this 5 minute questionnaire will help support ongoing research between M.I.T. and General Motors which is focusing on the challenge of integrating the efforts of several teams working on the same project.

Please characterize the degree to which your PDT is dependent upon technical information obtained from each of the other PDTs in the program.

Scale of Dependence

How often do you (and will you) need to get technical information from each of the other PDTs in order to complete the technical tasks of your PDT (over the next 6 months)?

Regularly (perhaps daily) --> **HIGH** dependence
Frequently (weekly or bi-weekly) --> **AVE** dependence
Infrequently (yet sometimes) --> **LOW** dependence
Never --> **ZERO** dependence

Please place a check beside your PDT.

	Level of Dependence on Technical Information			
	HIGH	**AVE**	**LOW**	**ZERO**
Engine Block	☐	☐	☐	☐
Pistons	☐	☐	☐	☐
Connecting Rods	☐	☐	☐	☐
Crankshaft	☐	☐	☐	☐
Flywheel	☐	☐	☐	☐
Lubrication	☐	☐	☐	☐
Heads	☐	☐	☐	☐
Camshaft/Valve Train	☐	☐	☐	☐
Cooling	☐	☐	☐	☐
Intake Manifold	☐	☐	☐	☐
Fuel System	☐	☐	☐	☐
Accessory Drive	☐	☐	☐	☐
Air Cleaner	☐	☐	☐	☐
A.I.R.	☐	☐	☐	☐
Throttle Body	☐	☐	☐	☐
Exhaust	☐	☐	☐	☐
EGR	☐	☐	☐	☐
EVAP	☐	☐	☐	☐
Ignition	☐	☐	☐	☐
PCM	☐	☐	☐	☐
Engine Electrical	☐	☐	☐	☐
Engine Assembly	☐	☐	☐	☐

Please return to Kent McCord at M.I.T. in the envelope provided.

Figure A.1. Survey of Inter-PDT Interaction

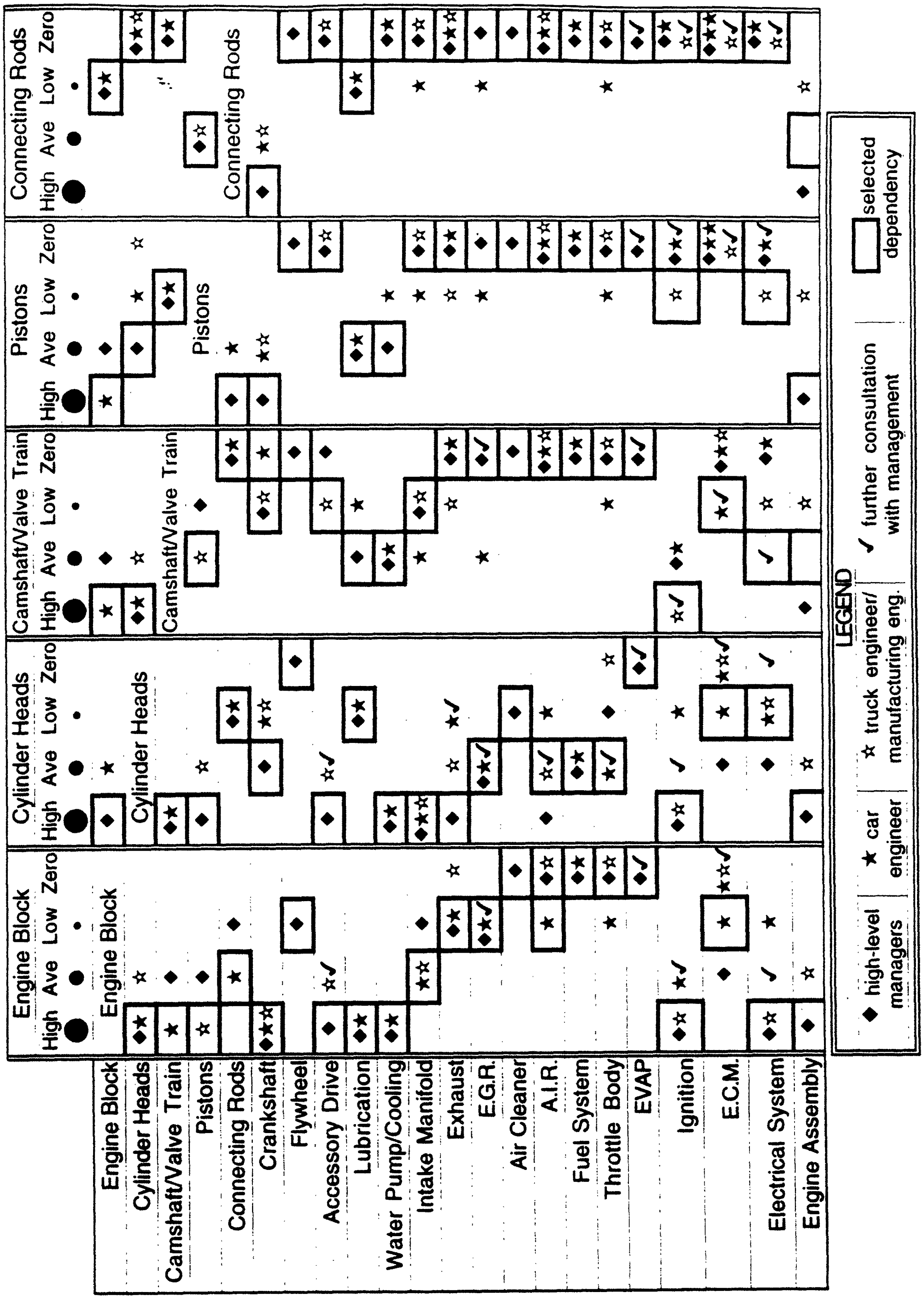
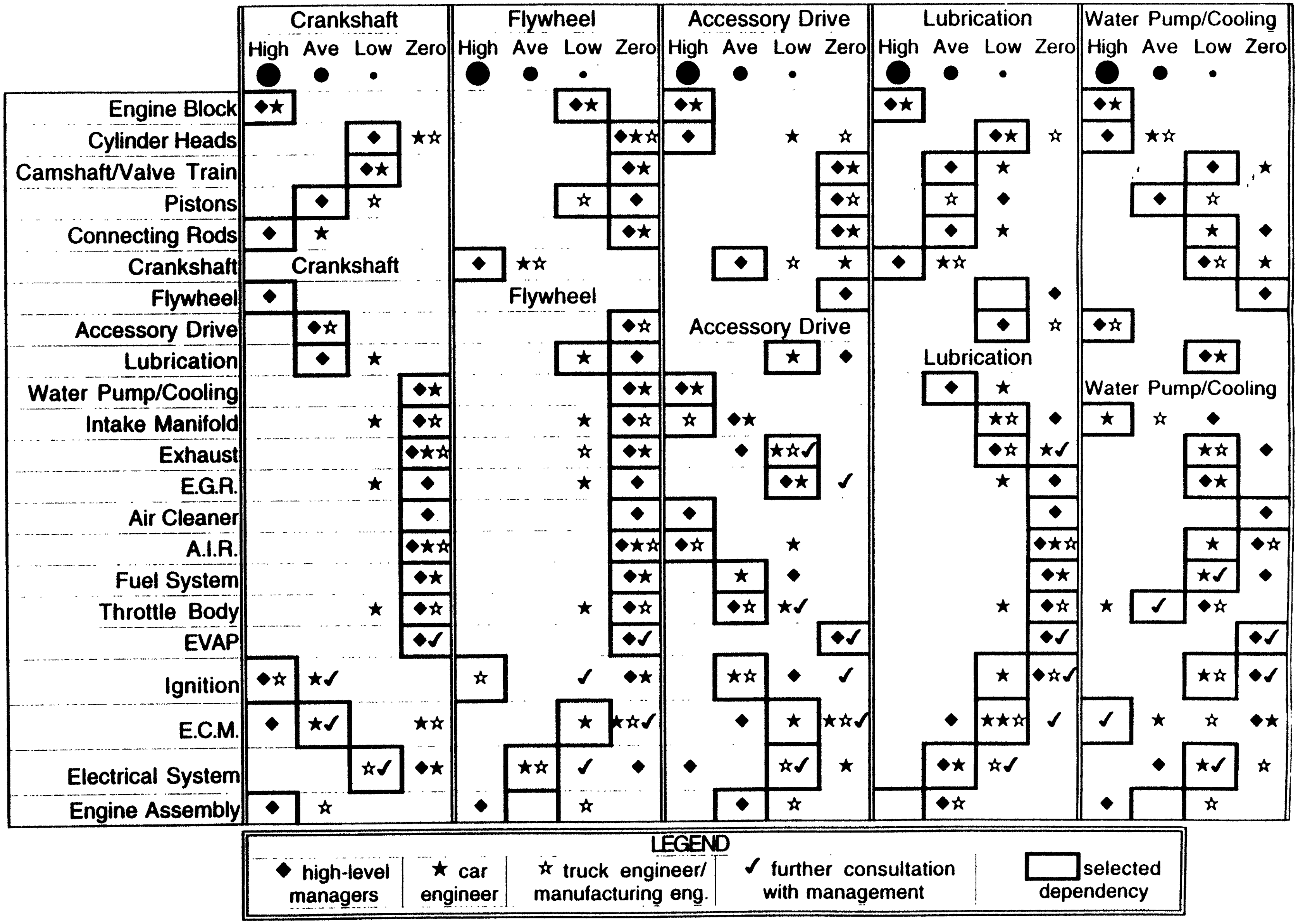


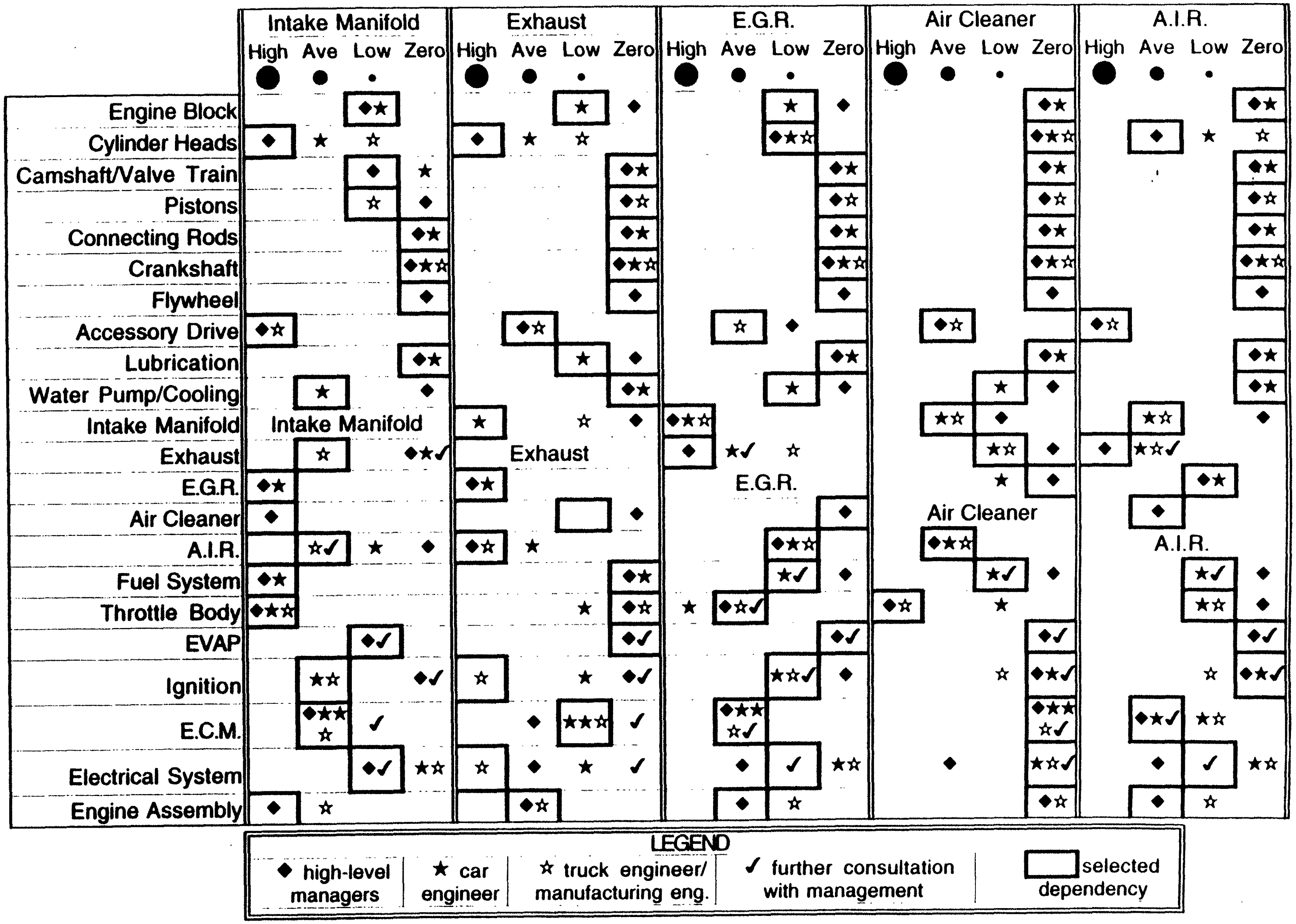
Figure A.2. Complete Set of Data for Constructing DSM



	Crankshaft				Flywheel				Accessory Drive				Lubrication				Water Pump/Cooling			
	High	Ave	Low	Zero	High	Ave	Low	Zero	High	Ave	Low	Zero	High	Ave	Low	Zero	High	Ave	Low	Zero
	●	●	•		●	●	•		●	●	•		●	●	•		●	●	•	
Engine Block	◆★						◆★		◆★				◆★				◆★			
Cylinder Heads			◆	★☆				◆★☆	◆		★	☆			◆★	☆	◆	★☆		
Camshaft/Valve Train			◆★					◆★				◆★		◆	★				◆	★
Pistons		◆	☆				☆	◆				◆☆		☆	◆			◆	☆	
Connecting Rods	◆	★						◆★				◆★		◆	★				★	◆
Crankshaft	Crankshaft				◆	★☆				◆	☆	★	◆	★☆					◆☆	★
Flywheel	◆				Flywheel							◆				◆				◆
Accessory Drive		◆☆						◆☆	Accessory Drive						◆	☆	◆☆			
Lubrication		◆	★				★	◆			★	◆	Lubrication						◆★	
Water Pump/Cooling				◆★				◆★	◆★					◆	★		Water Pump/Cooling			
Intake Manifold			★	◆☆			★	◆☆	☆	◆★					★☆	◆	★	☆	◆	
Exhaust				◆★☆			☆	◆★		◆	★☆✓				◆☆	★✓			★☆	◆
E.G.R.			★	◆			★	◆			◆★	✓			★	◆			◆★	
Air Cleaner				◆				◆	◆							◆				◆
A.I.R.				◆★☆				◆★☆	◆☆		★					◆★☆			★	◆☆
Fuel System				◆★				◆★		★	◆					◆★			★✓	◆
Throttle Body			★	◆☆			★	◆☆		◆☆	★✓				★	◆☆	★	✓	◆☆	
EVAP				◆✓				◆✓				◆✓				◆✓				◆✓
Ignition	◆☆	★✓			☆		✓	◆★		★☆	◆	✓			★	◆☆✓			★☆	◆✓
E.C.M.	◆	★✓		★☆			★	★☆✓		◆	★	★☆✓		◆	★★☆	✓	✓	★	☆	◆★
Electrical System			☆✓	◆★		★☆	✓	◆	◆		☆✓	★		◆★	☆✓			◆	★✓	☆
Engine Assembly	◆	☆			◆		☆			◆	☆			◆☆			◆		☆	

LEGEND

◆ high-level managers
★ car engineer
☆ truck engineer/ manufacturing eng.
✓ further consultation with management
☐ selected dependency



	Intake Manifold				Exhaust				E.G.R.				Air Cleaner				A.I.R.			
	High	Ave	Low	Zero	High	Ave	Low	Zero	High	Ave	Low	Zero	High	Ave	Low	Zero	High	Ave	Low	Zero
	●	●	•		●	●	•		●	●	•		●	●	•		●	●	•	
Engine Block			◆★				★	◆			★	◆				◆★				◆★
Cylinder Heads	◆	★	☆		◆	★	☆				◆★☆					◆★☆		◆	★	☆
Camshaft/Valve Train			◆	★				◆★				◆★				◆★				◆★
Pistons			☆	◆				◆☆				◆☆				◆☆				◆☆
Connecting Rods				◆★				◆★				◆★				◆★				◆★
Crankshaft				◆★☆				◆★☆				◆★☆				◆★☆				◆★☆
Flywheel				◆				◆				◆				◆				◆
Accessory Drive	◆☆					◆☆				☆	◆			◆☆			◆☆			
Lubrication				◆★			★	◆				◆★				◆★				◆★
Water Pump/Cooling		★		◆				◆★			★	◆			★	◆				◆★
Intake Manifold	Intake Manifold				★		☆	◆	◆★☆					★☆	◆			★☆		◆
Exhaust		☆		◆★✓	Exhaust				◆	★✓	☆				★☆	◆	◆	★☆✓		
E.G.R.	◆★				◆★				E.G.R.						★	◆			◆★	
Air Cleaner	◆							◆				◆	Air Cleaner					◆		
A.I.R.		☆✓	★	◆	◆☆	★					◆★☆			◆★☆			A.I.R.			
Fuel System	◆★							◆★			★✓	◆			★✓	◆			★✓	◆
Throttle Body	◆★☆						★	◆☆	★	◆☆✓			◆☆		★				★☆	◆
EVAP			◆✓					◆✓				◆✓				◆✓				◆✓
Ignition		★☆		◆✓	☆		★	◆✓			★☆✓	◆			☆	◆★✓			☆	◆★✓
E.C.M.		◆★★☆	✓			◆	★★☆	✓		◆★★☆✓						◆★★☆✓		◆★✓	★☆	
Electrical System			◆✓	★☆	☆	◆	★	✓		◆	✓	★☆		◆		★☆✓		◆	✓	★☆
Engine Assembly	◆	☆				◆☆				◆	☆					◆☆		◆	☆	

LEGEND

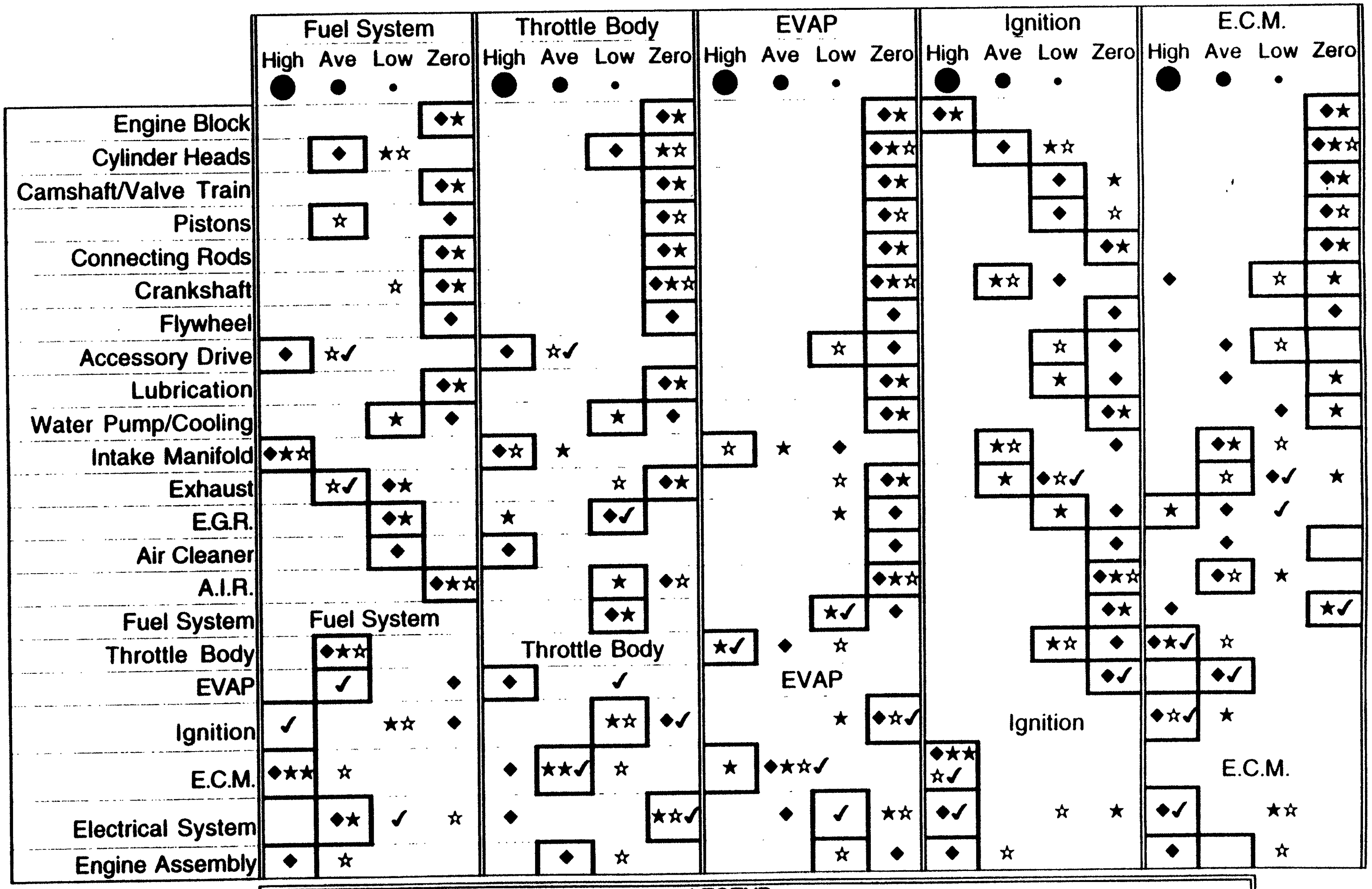
◆ high-level managers

★ car engineer

☆ truck engineer/ manufacturing eng.

✓ further consultation with management

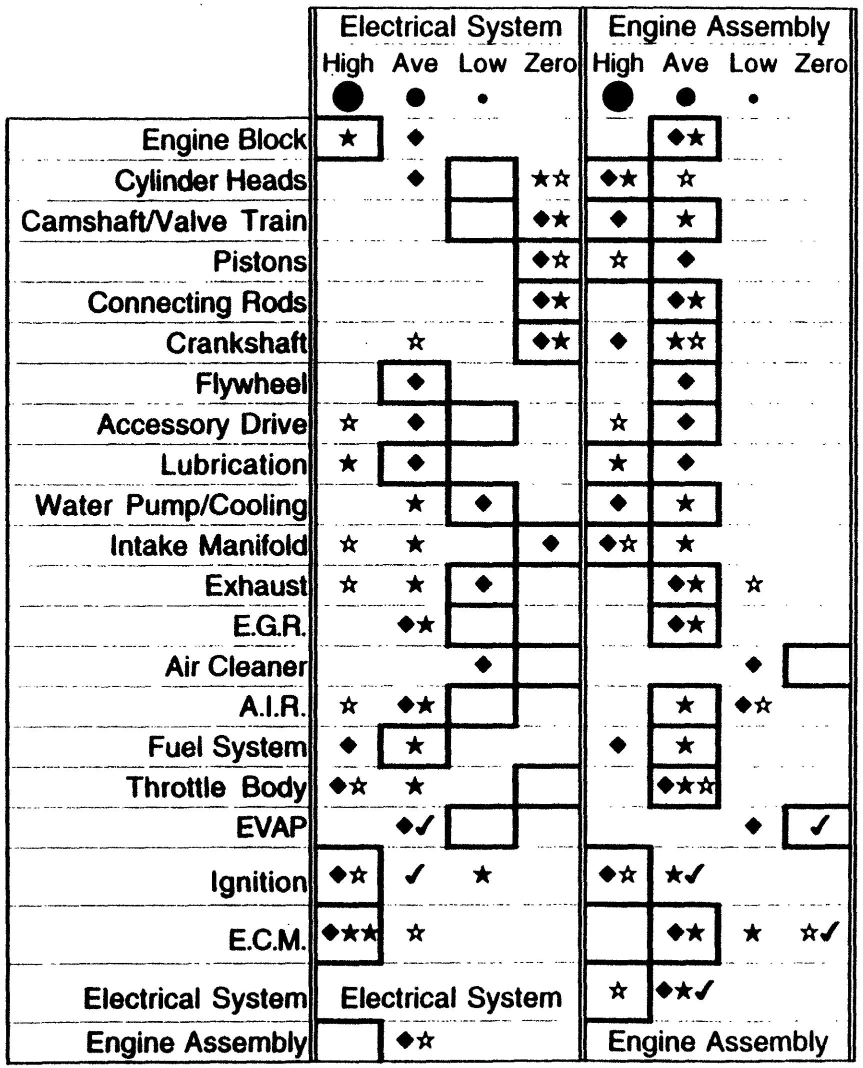
▭ selected dependency



	Fuel System				Throttle Body				EVAP				Ignition				E.C.M.			
	High ●	Ave ●	Low •	Zero	High ●	Ave ●	Low •	Zero	High ●	Ave ●	Low •	Zero	High ●	Ave ●	Low •	Zero	High ●	Ave ●	Low •	Zero
Engine Block				◆★				◆★				◆★	◆★							◆★
Cylinder Heads		◆	★☆				◆	★☆				◆★☆		◆	★☆					◆★☆
Camshaft/Valve Train				◆★				◆★				◆★			◆	★				◆★
Pistons		☆		◆				◆☆				◆☆			◆	☆				◆☆
Connecting Rods				◆★				◆★				◆★				◆★				◆★
Crankshaft			☆	◆★				◆★☆				◆★☆		★☆	◆		◆		☆	★
Flywheel				◆				◆				◆				◆				◆
Accessory Drive	◆	☆✓			◆	☆✓					☆	◆			☆	◆		◆	☆	
Lubrication				◆★				◆★				◆★			★	◆		◆		★
Water Pump/Cooling			★	◆			★	◆				◆★				◆★			◆	★
Intake Manifold	◆★☆				◆☆	★			☆	★	◆			★☆		◆		◆★	☆	
Exhaust		☆✓	◆★				☆	◆★			☆	◆★		★	◆☆✓			☆	◆✓	★
E.G.R.			◆★		★		◆✓				★	◆			★	◆	★	◆	✓	
Air Cleaner			◆		◆							◆				◆		◆		
A.I.R.				◆★☆			★	◆☆				◆★☆				◆★☆		◆☆	★	
Fuel System		Fuel System					◆★				★✓	◆				◆★	◆			★✓
Throttle Body		◆★☆				Throttle Body			★✓	◆	☆				★☆	◆	◆★✓	☆		
EVAP		✓		◆	◆		✓			EVAP						◆✓		◆✓		
Ignition	✓		★☆	◆			★☆	◆✓			★	◆☆✓		Ignition			◆☆✓	★		
E.C.M.	◆★★	☆			◆	★★✓	☆		★	◆★☆✓			◆★★ ☆✓					E.C.M.		
Electrical System		◆★	✓	☆	◆			★☆✓		◆	✓	★☆	◆✓		☆	★	◆✓		★☆	
Engine Assembly	◆	☆				◆	☆				☆	◆	◆	☆			◆		☆	

LEGEND

◆ high-level managers	★ car engineer	☆ truck engineer/ manufacturing eng.	✓ further consultation with management	☐ selected dependency



	Electrical System				Engine Assembly			
	High	Ave	Low	Zero	High	Ave	Low	Zero
	●	●	•		●	●	•	
Engine Block	★	◆				◆★		
Cylinder Heads		◆		★☆	◆★	☆		
Camshaft/Valve Train				◆★	◆	★		
Pistons				◆☆	☆	◆		
Connecting Rods				◆★		◆★		
Crankshaft		☆		◆★	◆	★☆		
Flywheel		◆				◆		
Accessory Drive	☆	◆			☆	◆		
Lubrication	★	◆			★	◆		
Water Pump/Cooling		★	◆		◆	★		
Intake Manifold	☆	★		◆	◆☆	★		
Exhaust	☆	★	◆			◆★	☆	
E.G.R.		◆★				◆★		
Air Cleaner			◆				◆	
A.I.R.	☆	◆★				★	◆☆	
Fuel System	◆	★			◆	★		
Throttle Body	◆☆	★				◆★☆		
EVAP		◆✓					◆	✓
Ignition	◆☆	✓	★		◆☆	★✓		
E.C.M.	◆★★	☆				◆★	★	☆✓
Electrical System	Electrical System				☆	◆★✓		
Engine Assembly		◆☆			Engine Assembly			

LEGEND

◆ high-level managers	★ car engineer	☆ truck engineer/ manufacturing eng.	✓ further consultation with management	☐ selected dependency

Appendix B. Eigenstructure Analysis: Methods and Results

After describing the Work Transformation Model and eigenstructure analysis developed by Smith and Eppinger [13], we will present our efforts to use this analysis for identifying more effective system teams for the engine development project.

Work Transformation Model

The Work Transformation Model (WTM) is an analytical extension to the design structure matrix framework. The WTM analysis is used to predict elements of a design problem which will require many iterations to reach a technical solution. It is based on a transformed version of the DSM in which the off-diagonal marks are replaced by numerical values that represent the strength of dependencies between tasks. All other elements in the matrix, including the diagonal marks, are replaced with zeros. Task dependencies are defined in terms of the amount of rework the execution of a task will create for other tasks in the project. In the transformed matrix, identified as A, the entries a_{ij} indicate that doing one unit of work on design task j creates a_{ij} units of rework on design task i.

Smith and Eppinger use this model to derive the following expression for the total amount of rework created for each task in a project:

$$U = S(I-\Lambda)^{-1}S^{-1}u_0$$

where U is a vector for which each element corresponds to the total amount of work done on one of the tasks in the model, Λ and S are the eigenvalue and eigenvector matrices for A, respectively, and u_0 is a vector of ones. (For the sake of brevity, we do not present a complete derivation of this expression and refer the reader to Smith and Eppinger [13] for details.)

Analyzing the Eigenstructure

The groups of tasks that are most responsible for rework and iteration in a project can be identified by analyzing the eigenstructure of the matrix A. Inspecting the above expression reveals that the last several terms:

$$(I-\Lambda)^{-1}S^{-1}u_0$$

result in a vector that is a combined weight on the eigenvector matrix S. This weighting vector describes the impact of each individual eigenvector on the total work vector U. The eigenvectors corresponding to the largest weights (in absolute value) characterize the most significant *design modes* of the project. A design mode is defined by Smith and Eppinger as a "group of design tasks which are very closely related"; they are the primary iteration loops in a project. The individual elements of the eigenvectors identify which tasks play the largest role in each design mode. Large positive values are most important if the weight is positive, large negative values if the weight is negative.

In this way the eigenstructure of a DSM can be analyzed to identify the groups of tasks (or teams) in a project which are strongly dependent on each other for information. Smith and Eppinger successfully used this model to describe the primary iteration drivers in the development of an automotive brake system. We attempt to apply the eigenstructure analysis to a different sort of project management problem.

Application to Engine Development Project

The management challenge associated with the engine development project differs slightly from the objective of the Work Transformation Model. Though the aim in both cases is to identify tightly coupled groups of tasks, task coupling is defined in two different ways. The WTM characterizes task coupling in terms of resulting rework whereas the engine development project DSM captures team coupling in terms of the required frequency of information exchange between teams. We believe, however, that the eigenstructure analysis outlined above can be effectively used to identify highly coupled entities regardless of how the coupling is defined. We now present our effort to apply this technique to the engine development project.

The first step in the analysis was to represent the inter-PDT coupling as numerical values in the matrix. We chose to use 0.1 to represent a high frequency of information exchange, 0.03 for average, and 0.01 for low. The eigenstructure of the resulting matrix was then determined and analyzed as outlined above. The complete eigenstructure is shown in Figure B.1.

The eigenvalues and the combined weight factors indicate only one dominant eigenvector -- the first eigenvector which has a combined weight on it of 14.66. The remaining eigenvectors have significantly smaller weights and therefore do not represent important groups of PDTs. (Though a few eigenvectors have relatively large weights (e.g. 5.06), they are ignored as their elements do not identify meaningful groups (e.g. lubrication, A.I.R., and EVAP).) The largest elements in the dominant eigenvector are highlighted in Figure B.2. These terms identify a group of eight highly interdependent PDTs -- engine block, cylinder heads, accessory drive, intake manifold, ignition, E.C.M., electrical system, and engine assembly.

The results of the eigenstructure analysis for the complete engine development matrix fell short of our expectations. We anticipated that several groups of tightly coupled teams would be identified and that every PDT would be associated with at least one group. We attribute this relative

EIGENVALUES	0.70	0.31	0.18	-0.24	0.14	-0.18	-0.18	-0.17	-0.17	0.08	0.08	-0.14	-0.10	-0.10	0.01	0.01	-0.05	0.00	0.00	-0.08	-0.01	0.09
TOTAL WEIGHTS	14.66	0.53	1.01	-0.40	-0.71	2.44	2.44	0.23	0.23	-1.00	-1.00	1.13	-0.77	-0.77	-0.46	-0.46	-0.72	2.21	2.21	0.81	5.06	0.27
											EIGENVECTOR MATRIX											
Engine Block	0.32	-0.27	-0.06	0.65	0.01	0.24	0.24	-0.14	-0.14	0.08	0.08	0.36	0.2	0.2	-0.12	-0.12	0.04	0.09	0.09	0.13	-0.11	0.29
Cylinder Heads	0.3	0.07	-0.27	-0.19	0.13	0.13	0.13	0.01	0.01	0.05	0.05	0.02	-0.04	-0.04	0.08	0.08	0.08	0.08	0.08	-0.05	-0.06	-0.31
Camshaft/Valve Train	0.13	-0.12	-0.11	-0.15	0.09	-0.18	-0.18	0.1	0.1	0.2	0.2	-0.23	-0.15	-0.15	-0.11	-0.11	-0.21	-0.07	-0.07	-0.04	0.05	0.14
Pistons	0.19	-0.19	-0.12	-0.24	-0.07	-0.09	-0.09	-0.04	-0.04	0.09	0.09	-0.22	-0.1	-0.1	0.16	0.16	0.46	0.03	0.03	0.25	-0.05	0.25
Connecting Rods	0.1	-0.23	-0.07	0.05	-0.32	0.14	0.14	-0.12	-0.12	-0.19	-0.19	-0.19	-0.13	-0.13	-0.03	-0.03	-0.41	0.24	0.24	-0.03	-0.3	-0.34
Crankshaft	0.18	-0.34	-0.02	-0.06	-0.39	-0.21	-0.21	0.26	0.26	-0.3	-0.3	0.43	0.29	0.29	-0.1	-0.1	-0.22	0.15	0.15	-0.2	-0.08	0.12
Flywheel	0.06	-0.15	0.11	0.02	-0.21	0.13	0.13	-0.17	-0.17	-0.3	-0.3	-0.31	-0.21	-0.21	0.04	0.04	0.38	0.16	0.16	0.15	-0.07	-0.04
Accessory Drive	0.28	0.27	-0.3	-0.12	-0.17	-0.16	-0.16	0.1	0.1	-0.2	-0.2	-0.06	0.17	0.17	0.05	0.05	-0.2	0.04	0.04	-0.28	0.03	0.27
Lubrication	0.15	-0.22	0.1	-0.26	-0.02	0.02	0.02	-0.08	-0.08	0.08	0.08	-0.23	-0.15	-0.15	-0.25	-0.25	-0.3	-0.62	-0.62	-0.08	0.61	-0.27
Water Pump/Cooling	0.19	0.02	-0.32	-0.07	-0.04	-0.07	-0.07	0.01	0.01	0.08	0.08	-0.09	-0.2	-0.2	-0.08	-0.08	0.13	0.22	0.22	0.17	-0.13	-0.3
Intake Manifold	0.31	0.36	-0.06	-0.01	-0.06	0.22	0.22	-0.33	-0.33	0.03	0.03	0.08	0.08	0.08	-0.01	-0.01	0.03	0.07	0.07	-0.04	-0.04	-0.02
Exhaust	0.16	0.16	0.03	0.11	0.35	-0.15	-0.15	0.04	0.04	-0.1	-0.1	-0.03	-0.04	-0.04	-0.03	-0.03	0.14	-0.1	-0.1	0.33	0.03	-0.22
E.G.R.	0.15	0.18	0.2	-0.14	0.27	-0.03	-0.03	0.11	0.11	0.15	0.15	0.01	-0.01	-0.01	-0.24	-0.24	0.02	0.28	0.28	0.08	-0.32	0.36
Air Cleaner	0.12	0.33	-0.18	0.11	-0.35	0.02	0.02	0.04	0.04	-0.32	-0.32	0.29	-0.15	-0.15	0.26	0.26	0.19	0	0	0.4	-0.21	-0.08
A.I.R.	0.13	0.2	-0.11	-0.04	0.09	0.14	0.14	-0.11	-0.11	-0.32	-0.32	-0.01	-0.04	-0.04	0	0	-0.05	-0.37	-0.37	-0.11	0.36	0.11
Fuel System	0.11	0.15	-0.02	0.06	-0.02	-0.08	-0.08	0.13	0.13	0	0	-0.05	-0.06	-0.06	0.24	0.24	-0.23	-0.08	-0.08	0.01	0.11	0.13
Throttle Body	0.17	0.32	0.09	-0.11	-0.35	-0.12	-0.12	0.15	0.15	0.01	0.01	-0.42	-0.1	-0.1	-0.05	-0.05	0.1	0.01	0.01	0.01	-0.08	-0.17
EVAP	0.05	0.13	0.14	0.03	-0.22	0.07	0.07	-0.12	-0.12	0.11	0.11	0.32	0.16	0.16	-0.22	-0.22	0.01	-0.23	-0.23	0.04	0.28	0.18
Ignition	0.34	-0.18	0.36	-0.19	0.16	0.18	0.18	-0.31	-0.31	-0.04	-0.04	-0.01	0.34	0.34	0.27	0.27	0	0.28	0.28	-0.33	-0.3	0.15
E.C.M.	0.21	0.03	0.39	0.23	0.06	-0.01	-0.01	0.07	0.07	0.19	0.19	-0.05	0	0	-0.12	-0.12	-0.21	0.06	0.06	-0.28	0.04	0.01
Electrical System	0.25	-0.15	0.49	-0.37	0.33	-0.04	-0.04	0.07	0.07	0.14	0.14	-0.09	-0.3	-0.3	-0.03	-0.03	0.24	-0.14	-0.14	0.46	-0.01	-0.26
Engine Assembly	0.38	-0.17	0.22	0.28	-0.02	-0.15	-0.15	0.22	0.22	0.01	0.01	0.07	-0.02	-0.02	0.15	0.15	0.01	-0.18	-0.18	-0.21	0.14	-0.05

Figure B.1. Eigenstructure for the Complete Engine Development Matrix

EIGENVALUE	0.70
TOTAL WEIGHT	**14.66**
EIGENVECTOR	
Engine Block	**0.32**
Cylinder Heads	**0.3**
Camshaft/Valve Train	0.13
Pistons	0.19
Connecting Rods	0.1
Crankshaft	0.18
Flywheel	0.06
Accessory Drive	**0.28**
Lubrication	0.15
Water Pump/Cooling	0.19
Intake Manifold	**0.31**
Exhaust	0.16
E.G.R.	0.15
Air Cleaner	0.12
A.I.R.	0.13
Fuel System	0.11
Throttle Body	0.17
EVAP	0.05
Ignition	**0.34**
E.C.M.	**0.21**
Electrical System	**0.25**
Engine Assembly	**0.38**

Figure B.2. Dominant Eigenvector for the Complete Engine Development Matrix

failure of the analysis to the fact that several PDTs are highly coupled across the entire project and consequently serve to directly or indirectly integrate all PDTs together. In fact, the eight PDTs identified in the dominant eigenvector are the major integrative teams in the project. Examining the DSM confirms that each of these eight teams requires input from and gives output to most of the other teams in the project.

The complete matrix of all twenty-two PDTs is therefore perhaps too coupled (too dense) for the eigenstructure analysis to reveal PDT coupling at a level beneath the high-level coupling created by the eight integrative tasks. Based on this conjecture, we repeated the eigenstructure analysis with the eight integrative teams removed from the matrix. Our goal was to identify logical groupings for the non-integrative teams.

The complete eigenstructure for the matrix of the remaining fourteen PDTs is shown in Figure B.3 and the results of the analysis are summarized in Figure B.4. The analysis shows two significant eigenvectors which characterize two major groupings for the PDTs. One grouping consists of the camshaft/valve train, pistons, connecting rods, crankshaft, flywheel, lubrication, and water pump/cooling PDTs. The second grouping is of the exhaust, E.G.R., air cleaner, A.I.R., fuel system, throttle body, and EVAP teams.

It should be observed that the results of this analysis roughly correspond with the system teams proposed from the visual heuristic method. The first group identified above is comprised of the same PDTs as the first two proposed system teams, excluding the engine block, cylinder heads, and intake manifold PDTs which were removed from the eigenstructure analysis. Likewise, the second grouping above matches the combination of the third and fourth system teams, excluding the cylinder heads and intake manifold PDTs which were removed.

Discussion

Though the results from the visual heuristic method are more detailed and applicable, the eigenstructure analysis successfully identified two basic groupings of PDTs based only on the technical structure of the project as captured in the DSM. Therefore, in the absence of a good baseline project organization or without a good experiential and intuitive understanding of the project, the eigenstructure analysis may be useful in identifying basic groupings of tasks or teams. It may provide a good starting point from which to continue with the visual heuristic method.

Though the eigenstructure analysis is more mathematical than the visual inspection method, it is still somewhat heuristic. The first attempt at applying the method to a complete DSM might only highlight the most integrative tasks or teams in the project, in which case the analysis must be done at least one more time without the integrative teams in order to determine more meaningful, detailed groupings. The method is not a robust algorithm. Furthermore, the eigenstructure analysis is suggestive rather than definitive. It may provide a good starting point for a project organization but more refinement will likely be necessary to identify the best possible solution.

Though the eigenstructure method might be effectively used for some projects, there is clearly a need for more algorithmic and robust methods for identifying groups of coupled tasks or teams in a DSM.

EIGENVALUES	0.20	0.17	0.12	-0.15	-0.15	-0.13	0.04	-0.05	0.01	0.01	-0.04	-0.01	-0.02	0.00
TOTAL WEIGHTS	4.39	2.92	0.00	0.86	-0.21	-0.18	-2.26	-0.49	1.61	0.58	-0.30	-0.49	0.38	0.79
						EIGENVECTOR MATRIX								
Camshaft/Valve Train	0.08	-0.02	0	0.03	0.03	0.01	-0.18	0.07	0.11	-0.03	-0.46	-0.12	0.14	-0.05
Pistons	0.26	-0.05	0.02	0	0	-0.01	-0.29	0.58	-0.2	0.05	-0.38	-0.06	-0.04	0.04
Connecting Rods	0.51	-0.16	0.01	0.41	0.47	0.11	-0.18	-0.64	-0.62	0.18	0.19	0.17	0.18	-0.55
Crankshaft	0.68	-0.22	0	-0.62	-0.7	-0.13	0.3	-0.16	0.09	-0.02	0.1	0.02	0.04	-0.02
Flywheel	0.36	-0.13	0	0.42	0.48	0.1	0.74	0.33	0.74	-0.22	-0.43	-0.04	-0.2	0.56
Lubrication	0.21	-0.04	-0.03	0.08	0.07	-0.03	-0.09	-0.17	0.09	-0.01	0.6	-0.12	0.1	-0.05
Water Pump/Cooling	0.09	0.06	0.04	-0.02	-0.02	0.03	-0.42	-0.29	0	-0.04	0.21	0.58	-0.64	0.16
Exhaust	0.07	0.43	-0.47	-0.14	0.01	0.49	0.03	0.02	0	0.08	0	-0.06	0.01	-0.05
E.G.R.	0.05	0.3	-0.35	0.12	-0.01	-0.31	-0.01	0.02	-0.01	0.45	-0.04	0.45	0.27	-0.3
Air Cleaner	0.05	0.38	0.32	0.23	-0.12	0.39	0.12	0.05	0	-0.56	-0.02	-0.05	0.36	0.2
A.I.R.	0.05	0.36	-0.28	0.07	0.01	-0.4	0.07	0	0	-0.38	-0.03	-0.33	-0.33	0.17
Fuel System	0.02	0.11	0.06	-0.02	0.01	0.03	-0.05	0.04	0	0	-0.03	-0.49	0.28	0.33
Throttle Body	0.07	0.5	0.53	-0.34	0.17	-0.45	0.02	-0.03	0	0.05	0.02	0.16	-0.01	-0.09
EVAP	0.04	0.31	0.44	0.24	-0.12	0.33	0.03	0.04	0	0.48	-0.03	-0.1	-0.3	-0.27

Figure B.3. Eigenstructure for
Engine Development Matrix without Integrative PDTs

EIGENVALUES	0.20	0.17
TOTAL WEIGHTS	**4.39**	**2.92**
	EIGENVECTORS	
Camshaft/Valve Train	**0.08**	-0.02
Pistons	**0.26**	-0.05
Connecting Rods	**0.51**	-0.16
Crankshaft	**0.68**	-0.22
Flywheel	**0.36**	-0.13
Lubrication	**0.21**	-0.04
Water Pump/Cooling	**0.09**	**0.06**
Exhaust	0.07	**0.43**
E.G.R.	0.05	**0.30**
Air Cleaner	0.05	**0.38**
A.I.R.	0.05	**0.36**
Fuel System	0.02	**0.11**
Throttle Body	0.07	**0.50**
EVAP	0.04	**0.31**

Figure B.4. Two Significant Eigenvectors for Engine Development Matrix without Integrative PDTs