INTEGRATION ANALYSIS OF PRODUCT DECOMPOSITIONS

Thomas U. Pimmler and Steven D. Eppinger
Massachusetts Institute of Technology
Cambridge, Massachusetts

ABSTRACT

This paper describes a methodology for the analysis of product design decompositions. The technique is useful for developing an understanding of the "system engineering" needs which arise because of complex interactions between components of a design. This information can be used to define the product architecture and to organize the development teams. The method involves three steps: 1) decomposition of the system into elements, 2) documentation of the interactions between the elements, and 3) clustering the elements into architectural and team chunks. By using this approach, development teams can better understand the complex interactions within the system, thus simplifying the development process for large and complex projects.

INTRODUCTION

It is common practice for engineers to solve a complex problem by first breaking it into a set of smaller problems that are more easily handled. In the development of any complex engineered system or product, the function of the device is decomposed into multiple sub-functions so that the team can research solutions for each of the smaller pieces. There are two reasons why this approach is attractive: 1) simplification— the smaller problems are generally simpler to solve than the large system; and 2) speed— solutions to the sub-problems can be derived in parallel. There are also two challenges associated with this problem-solving strategy: 1) decomposition— finding the most suitable set of sub-problems may be difficult; and 2) integration— combining the separate sub-systems into an overall solution may also be difficult.

This paper addresses the integration problem as applied to product development. For a complex product, such as an automobile, a computer, or an airplane, there are thousands of possible decompostions which may be considered. Each of these alternative decompositions defines a different set of integration challenges. Ulrich and Eppinger (1994) define the architecture of a product as the scheme by which the decomposed elements are arranged in chunks. The choice of product architecture has broad implications for product performance, product change, product variety, and manufacturability. Product architecture is also strongly coupled to the firm's development capability, manufacturing specialties, and product strategy. Selecting the proper architecture of the product is an extremely influential decision which must be made during the concept development and system-level design phases of the project; the architecture defines the sub-systems upon which the team will work for the bulk of the development effort.

In product development, analysis of the product decomposition provides valuable insight into the structure of the problem and the choice of architecture. The integration analysis presented in this paper considers the interactions which occur between the elements of the decomposition. The building blocks (called chunks) which result from integration analysis can be used to define the product architecture and to structure the development teams.

Examples of architecture and team structure can be found in any highly engineered product. In the automobile industry, development programs include hundreds or thousands of team members. It would be impractical to design the entire vehicle at once (too complex); nor would it be possible to develop the thousands of components one at a time (too slow). The vehicle is decomposed into a few major systems: body, powertrain, chassis, interior, climate control, electrical, and trim. Each of these major systems is in turn decomposed into a large number of sub-systems, resulting in hundreds of interconnected pieces with names like: passenger restraint system, fuel delivery system, remote entry system, etc. Finally, these sub-systems are decomposed into component parts which are designed and tested individually and together.

The decomposition of the vehicle into sub-systems and components facilitates the rapid development of the individual pieces, yet this strategy does not address the needs for integration of the components' functions during the development process.

ASME Design Theory and Methodology Conference
Minneapolis, MN
September 1994

This is a significant concern because alternative architectures will have different interactions between sub-systems. For example, if the seat-belt restraint system is contained entirely within the seat system, then the restraint system design team needs to interact quite closely with the seat system developers in order to ensure that both systems function properly. On the other hand, if the seat belt restraints attach to the body pillars, doors, and center console, such an architecture would require much more complex interactions among several design teams, now including body and interior systems. Since these types of interactions involve tremendous coordination efforts, we propose that architecture and team structure decisions should be made with coordination complexity in mind.

In industrial practice, product architecture and team structures do not change very often, even over several product generations. This implies that either the “optimal” solution structure has been found, or that the companies have so much invested in a given structure that they have difficulty in considering alternatives. This paper provides a methodology for generating and selecting improved product architectures and development team structures.

There is some related research and literature regarding decomposition and architecture at the system definition stage of product design. The core research begins with Alexander (1964), who describes a design process which decomposes (or partitions) designs into minimally coupled groups. Simon (1981) continues by suggesting that complex design problems can be described in terms of hierarchical structures consisting of "nearly decomposable systems" organized such that the strongest interactions occur within groups and only weaker interactions occur among groups. Pahl and Beitz (1991) and Suh (1990) build upon these concepts by modeling the functional requirements of product design in terms of exchanges of energy, materials, and signals between function elements organized in hierarchical function structures. Ulrich□ (1994) defines several types of product architecture in terms of how the functional elements are mapped onto physical components and relates the strategic importance of architecture choice to firm performance. Henderson and Clark (1990) also relate the importance of architecture by noting that established firms frequently fail when confronted by a novel architecture. Ulrich and Eppinger (1994) provide a methodology for developing a product architecture, however interactions are only considered after the architecture is chosen.

Other researchers utilize these concepts to focus on other product development challenges. Smith and Browne (1993) describe decomposition as a fundamental approach to handling complexity in engineering design. Warfield and Hill (1972) suggest that an approach to solving complex problems lies in understanding and controlling the interactions between the elements. Verho and Salminen (1993) propose that interactions between elements in a design vary in strength which relates to the speed of the development process. Kusiak and Szczerbicki (1993) use binary interactions represented in a digraph to develop physical design layouts. Organization researchers, such as Galbraith (1973) relay the importance of having a good fit between the structure of the organization and the structure of the problem. Lovejoy (1992) relates design decomposition to the organizational issues of complex product design processes. McCord and Eppinger (1993) describe a methodology using interactions between components to structure system teams in a development project.

Researchers using the design structure matrix technique, including Steward (1981) and Eppinger, et al. (1994) have analyzed parameter-level interactions to create design parameter groupings that must be solved iteratively. Kusiak and Wang (1993) focus on using decomposition to structure tasks and parameters in the detail design stage. In optimization research, Wagner and Papalambros (1993) and others use decomposition to structure mathematical programming problems applied to detail design.

Although the above research gives considerable insight into product architecture and decomposition, no structured methodologies exist for generating and analyzing alternative architectures and decompositions. This paper focuses on finding alternative architectures in order to improve the quality of the resulting product design and to ease the substantial coordination demands that are required when sub-systems interact. We present a development methodology that can be used to evaluate product decompositions based upon various criteria.

The next section of the paper introduces an example problem that we will use to illustrate the methodology. We then explain the three steps to our approach: decomposing the system into elements, documenting the interactions between the elements, and clustering the elements into chunks. We close the paper with a discussion of the strengths, weaknesses, and implications of such an approach. Finally, we recognize that this methodology, while useful in practice, leaves room for additional research and improvement.

AN EXAMPLE

To illustrate our integration methodology, we will refer to an industrial example throughout the paper. We have been developing this application in our research with Ford Motor Company. Figure 1 shows some of the typical components of an automotive climate control system. This is an example of decomposition applied at the system-to-components level, rather than at the product-to-systems level. However, we intend the methodology to apply to both types of problems.

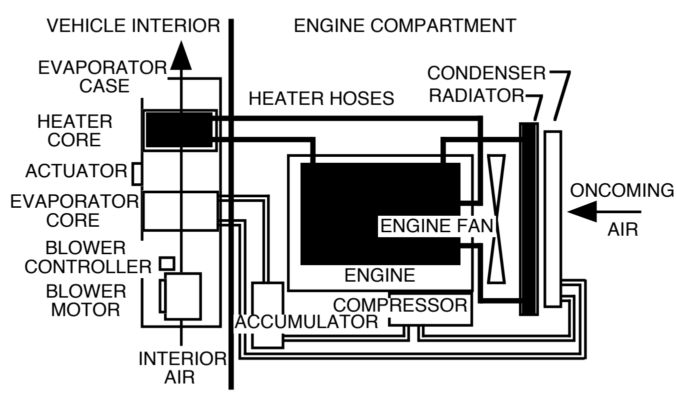

FIGURE 1: AUTOMOTIVE CLIMATE CONTROL SYSTEM COMPONENT SCHEMATIC.

An automotive climate control system performs two basic functions: passenger compartment heating and cooling. Heating is achieved by circulating hot engine coolant via the heater hoses through the heater core (a heat exchanger). Passenger compartment air is forced across the heater core to transfer energy. Engine coolant is also circulated through the radiator (another heat exchanger) to dissipate additional energy to the outside air. Passenger compartment cooling is achieved using a refrigeration loop comprised of five main components: compressor, condenser, evaporator, expansion valve (not shown), and accumulator. The compressor receives low pressure refrigerant gas and delivers high pressure gas to the condenser (also a heat exchanger), where it condenses, dissipating energy to the outside air. The high pressure liquid then flows to the evaporator through an expansion valve (not shown), which maintains the pressure difference in the loop. As the refrigerant expands, heat energy is absorbed through the evaporator, which is another heat exchanger over which warm passenger compartment air is passed. The refrigerant, now a mixture of liquid and gas, then moves to the accumulator, which traps the liquid droplets, allowing only gas to enter the compressor.

The schematic identifies several components in the system, but their interactions are not depicted here. (We will use an interaction matrix for this purpose later.) Such interactions come about due to a number of effects, including energy and material transfers, among others. For example, the radiator and engine fan are functionally coupled via the air stream between them. The amount of energy the radiator can dissipate is strongly related to the volume of air flow provided by the engine fan. Both the radiator and the engine fan are also closely coupled with the condenser via the same air stream. The energy dissipation rate of the condenser is also dependent on the air flow provided by the fan. The radiator and condenser are also directly coupled because the amount by which the air stream temperature is raised by the first heat exchanger limits the heat exchange capability of the second. These interactions demand a high degree of coordination between the engineers or teams developing the related components. Our development methodology must consider such interactions when defining the product architecture and organizing the development teams.

OUR APPROACH

In this paper, we present a three-step integration analysis methodology which begins with the product concept and results in the definition of architectural chunks and/or team chunks ready for resource allocation and detail design.

Step 1) **Decompose the System into Elements:** Describe the product concept in terms of functional and/or physical elements which achieve the product's functions.

Step 2) **Document the Interactions between Elements:** Identify the interactions which may occur between the functional and physical elements.

Step 3) **Cluster the Elements into Chunks:** Cluster the elements into chunks based on criteria set by the overall product design strategy of the team. These chunks then define the product architecture and system team structure.

Figure 2 illustrates this integration analysis methodology. Note that this approach involves breaking the overall function into smaller elements than the final level desired. This allows us to consider interactions among the elements when clustering them into chunks.

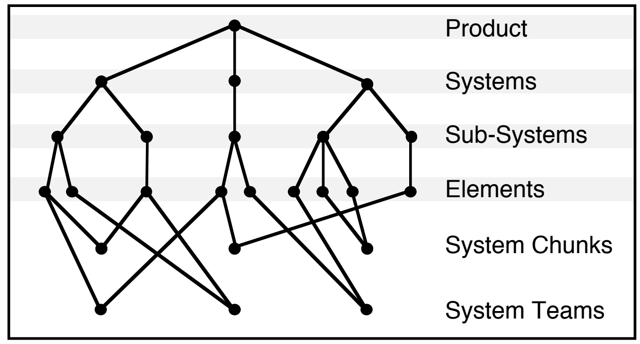

FIGURE 2: INTEGRATION METHODOLOGY

The next three sections detail the above three steps and apply them to the climate control system example.

STEP 1: DECOMPOSE THE SYSTEM INTO ELEMENTS

The first step in the analysis requires specification of the overall product concept in terms of functional and/or physical elements. This step is usually straightforward. Indeed, complex problems are quite commonly broken down into simpler sub-problems. The challenge, however, is in determining how finely the elements should be divided. As depicted above, we suggest specifying the elements to one level of detail further than that desired for the product architecture. For example, if a sub-system level product architecture is desired, then one would decompose elements to the component level so that they can be clustered at the sub-system level.

Both functional and physical elements may be used in the decomposition. In more novel design situations, one would primarily utilize functional elements, organized in a function diagram, because most of the physical elements have not yet been identified. In more incremental design, physical elements or components would be used. However, in many situations, a mixture of both would be used. For example, in a novel design, perhaps only one form of technology exists or is feasible to fulfill particular functions. Physical elements would then be used to appropriately describe that technology. Similarly, in incremental design, functional elements may replace certain physical elements in order to explore more innovative design alternatives.

For our climate control system example, the system is well understood and the goal is to consider alternative architectures utilizing the existing technologies and to develop system teams based on the needs of those technologies. We therefore choose to represent the system decomposition as a set of physical elements, or components, as listed in Figure 3.

At this point, we have a design decomposed into unit-level elements. We next document the interactions between these elements so we can cluster them into architectural and team chunks.

Radiator	Accumulator
Engine Fan	Refrigeration Controls
Heater Core	Air Controls
Heater Hoses	Sensors
Condenser	Command Distribution
Compressor	Actuators
Evaporator Case	Blower Controller
Evaporator Core	Blower Motor

FIGURE 3: DECOMPOSED ELEMENTS FOR CLIMATE CONTROL SYSTEM.

STEP 2: DOCUMENT THE INTERACTIONS BETWEEN ELEMENTS

With the product concept appropriately described in terms of functional and physical elements, the next step is to determine how these elements might interact. Documenting interactions between elements is important because it allows us to understand the needs for coordination in the later development stages, and how the coordination requirements depend upon the clustering of elements.

The description of the interactions between the elements essentially captures the current level of knowledge about the design. Depending upon the balance between functional and physical elements, and other factors such as engineering and manufacturing strategies, the current level of knowledge includes understanding of both the functionality and imposed constraints on the chosen elements. Further, as knowledge of element relationships progresses, the interactions must be revised and updated.

To develop a scheme for systematically identifying and describing interactions, one could begin by considering a taxonomy of interactions. An example of a climate control system interaction is the transfer of energy via the refrigerant between the compressor and condenser.

We propose consideration of four important types: 1) associations of physical space and alignment, 2) associations of energy exchange, 3) associations of information exchange, and 4) associations of materials exchange. We define these four generic interaction types as follows:

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

Energy: An energy-type interaction identifies needs for energy transfer between two elements.

Information: An information-type interaction identifies needs for information or signal exchange between two elements.

Material: A material-type interaction identifies needs for materials exchange between two elements.

The number and definitions of the interaction types is dependent upon the context of the given design problem. For example, in a design primarily involving many different flows of materials, such as a chemical processing system, defining two or more material type interactions may be appropriate. In this paper, we propose a general case which we then apply to a problem relatively balanced in the number of interaction types present.

Spatial		
Required:	(+2)	Physical adjacency is necessary for functionality.
Desired:	(+1)	Physical adjacency is beneficial, but not absolutely necessary for functionality.
Indifferent:	(0)	Physical adjacency does not affect functionality.
Undesired:	(-1)	Physical adjacency causes negative effects but does not prevent functionality.
Detrimental:	(-2)	Physical adjacency must be prevented to achieve functionality.
Energy		
Required:	(+2)	Energy transfer is necessary for functionality.
Desired:	(+1)	Energy transfer is beneficial, but not absolutely necessary for functionality.
Indifferent:	(0)	Energy transfer does not affect functionality.
Undesired:	(-1)	Energy transfer causes negative effects but does not prevent functionality.
Detrimental:	(-2)	Energy transfer must be prevented to achieve functionality.
Information		
Required:	(+2)	Information exchange is necessary for functionality.
Desired:	(+1)	Information exchange is beneficial, but not absolutely necessary for functionality.
Indifferent:	(0)	Information exchange does not affect functionality.
Undesired:	(-1)	Information exchange causes negative effects but does not prevent functionality.
Detrimental:	(-2)	Information exchange must be prevented to achieve functionality.
Materials		
Required:	(+2)	Materials exchange is necessary for functionality.
Desired:	(+1)	Materials exchange is beneficial, but not absolutely necessary for functionality.
Indifferent:	(0)	Materials exchange does not affect functionality.
Undesired:	(-1)	Materials exchange causes negative effects but does not prevent functionality.
Detrimental:	(-2)	Materials exchange must be prevented to achieve functionality.

FIGURE 4: GENERAL INTERACTION QUANTIFICATION SCHEME.

Some interactions between elements are more important than others. Moreover, some interactions are described as desirable, while others are detrimental. Interactions can be quantified on a five-point scale based on the relative need for each interaction type. Figure 4 describes this interaction quantification scheme.

For the climate control system example presented in this paper, the scoring of the spatial type interaction is modified in the following manner: Elements performing the function of a conduit, or flow pathway have a spatial interaction score of (+1). Examples of such elements include hoses and electrical wiring. This modification is necessary for the following reason: Although hoses and electric wires are required to achieve product functionality, generally they are spatially flexible and can be routed around other elements in the design. Nevertheless, it is desirable for the elements which the conduits connect to be placed close to one another in order to minimize conduit length.

Two steps are therefore required to document an interaction: 1) identify the interaction types which describe the interaction, and 2) score each interaction type identified for relative strength.

1) <u>Identify Interaction Types:</u> Based on the definition of each of the four interaction types described above, determine the types needed to capture the essence of the interaction.

2) <u>Score Each Interaction Type:</u> Using the quantification scheme outlined in Figure 4, assign a score to each interaction type. The default score for unneeded types is zero (Indifferent).

An interaction between two elements can thus be represented by a vector of four scores (spatial, energy, information, and materials). Figure 5 shows how this rating scheme is applied to an interaction between two components (radiator and engine fan) in the climate control system example. (Please refer to Pimmler (1994) for interaction documentation data for the remaining interactions for this climate control system example.)

Elements:	**Radiator and Engine Fan**		
Function: (Radiator)	The radiator dissipates excess engine heat, via forced convection, to the outside surroundings.		
Function: (Engine Fan)	The engine fan draws outside air into the engine compartment.		
Relationship:	The engine fan **provides airflow** across the radiator. They are **located in close proximity** for design efficiency and due to space management constraints.		
Score:	Spatial: +2 Information: 0	Energy: 0 Materials: +2	

FIGURE 5: QUANTIFICATION OF THE INTERACTION BETWEEN THE RADIATOR AND ENGINE FAN IN THE CLIMATE CONTROL SYSTEM EXAMPLE.

Interaction information can be obtained in a variety of ways. We have had success with interviewing and surveying engineering team members and other domain experts. Engineering models and hardware can also provide useful insight into element-level interactions.

Figure 6 shows an interaction matrix displaying the results of quantifying the interactions for the sixteen elements of the climate control system described above. These interactions were found by interviewing climate control system engineers.

STEP 3: CLUSTER THE ELEMENTS INTO CHUNKS

After the interactions have been quantified, the next step is to cluster the elements into chunks. Clustering can be used to define not only the physical architecture of the product, but also the product development team structure. This may be done merely on the basis of the interactions alone, or other architectural and resource criteria may be considered such as product line strategy, anticipated technological changes, manufacturability, as well as team capabilities, core competencies, etc. In this paper, we illustrate clustering based only upon interaction complexity.

Clustering can significantly impact the coordination complexity of the design process resulting from this analysis. The interactions documented in the previous step describe, at the system level, the design issues which the engineering teams must resolve. Therefore, coordination complexity can be reduced if the elements are clustered such that the interactions predominately occur within chunks, rather than between chunks.

Even though the interactions have been quantified for each type, some types of interactions are generally more important than others. For example, in product architecture clustering, spatial adjacency requirements may be given a high priority because it is often difficult to overcome adjacency restrictions that are necessary or detrimental for the product's function. Information signals, on the other hand, may be more easily carried across chunks and can be specified through coordination across chunks. However, relative importance of the interaction types is highly dependent upon the nature of the product being designed.

For the climate control system, we propose clustering the decomposed elements based on individual interaction types.

The independent perspectives gained by considering each type individually can then be used to build the product architecture and to organize the development teams. Another approach to clustering might involve combining the individual types to form a weighted composite. However, we believe that this approach muddles the perspectives gained by clustering interaction types singly. Nevertheless, the methodology we have developed for clustering allows either single interaction types or composite weightings to be considered easily.

There are several algorithms which can be used to cluster interaction matrices. The appropriate clustering algorithms must reorder the rows and columns in the matrix in order to cluster the positive elements closer to the diagonal. A blocked matrix results, with the blocks on the diagonal corresponding to the resulting architectural clusters. For the climate control system example given in this paper, we have used a heuristic swapping algorithm facilitated by computer spreadsheet software and specialized macros. The algorithm reordered the matrix based on a distance penalty computed for each interaction. Researchers such as Kusiak (1990) have developed other clustering algorithms.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Radiator A		2 0 0 2			2 -2 0 0											
Engine Fan B	2 0 0 2				2 0 0 2								1 0 0 0			
Heater Core C				1 0 0 0			2 0 0 0	-1 0 0 0								0 0 0 2
Heater Hoses D			1 0 0 0						-1 0 0 0							
Condenser E	2 -2 0 0	2 0 0 2				0 2 0 2		-2 2 0 2								
Compressor F					0 2 0 2			0 2 0 2	1 0 0 2	0 0 2 0	0 0 2 0		1 0 0 0			
Evaporator Case G			2 0 0 0					2 0 0 0						2 0 0 0	2 0 0 0	2 0 0 2
Evaporator Core H			-1 0 0 0		-2 2 0 2	0 2 0 2	2 0 0 0		1 0 0 2							0 0 0 2
Accumulator I				-1 0 0 0		1 0 0 2		1 0 0 2		1 0 0 0						
Refrigeration Controls J						0 0 2 0			1 0 0 0		0 0 2 0		1 0 0 0			
Air Controls K						0 0 2 0				0 0 2 0		0 0 2 0	1 0 0 0	0 0 2 0	0 0 2 0	
Sensors L											0 0 2 0		1 0 0 0			
Command Distribution M		1 0 0 0				1 0 0 0				1 0 0 0	1 0 0 0	1 0 0 0		1 0 0 0	1 0 0 0	1 0 0 0
Actuators N							2 0 0 0				0 0 2 0		1 0 0 0			
Blower Controller O							2 0 0 0				0 0 2 0		1 0 0 0			2 0 0 2
Blower Motor P			0 0 0 2				2 0 0 2	0 0 0 2					1 0 0 0		2 0 0 2	

NOTE: BLANK MATRIX ELEMENTS INDICATE NO INTERACTION (FOUR ZERO SCORES).

Legend:

Spatial: S E :Energy
Information: I M :Materials

FIGURE 6: CLIMATE CONTROL SYSTEM INTERACTION MATRIX.

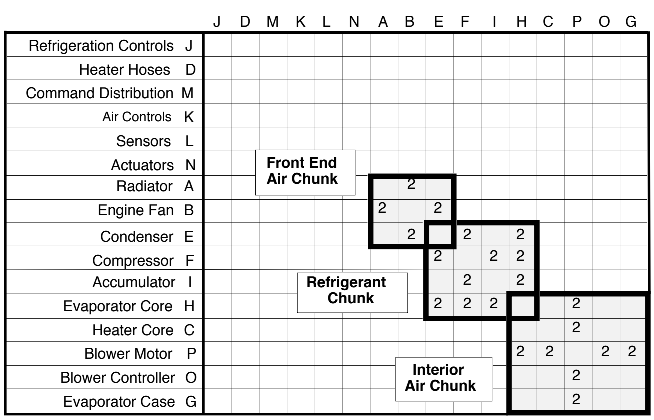

FIGURE 7: INDEPENDENT CLUSTERING OF MATERIALS INTERACTIONS FOR CLIMATE CONTROL SYSTEM.

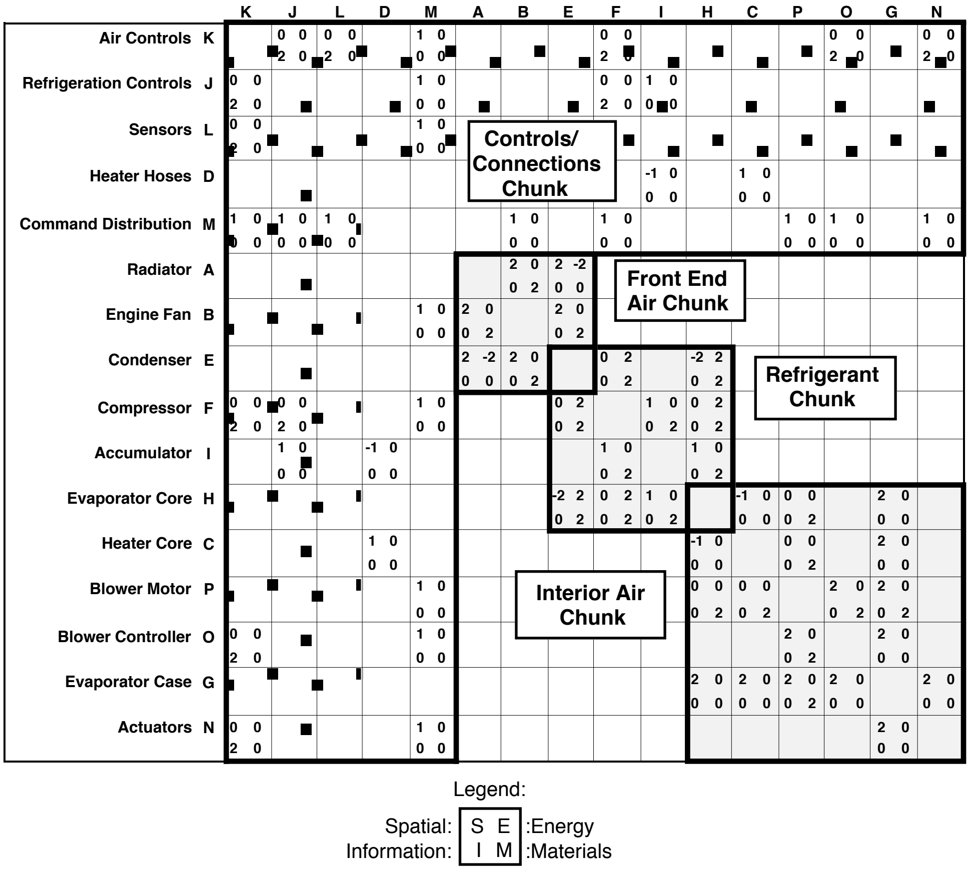

FIGURE 8: CLUSTERED INTERACTION MATRIX FOR THE CLIMATE CONTROL SYSTEM EXAMPLE.

Figure 7 shows the materials perspective resulting from clustering the climate control system interaction types independently. Figure 8 builds upon that structure by adding the spatial, energy and information perspectives to the materials clustering. The four prominent interacting chunks identified by this clustering are interpreted and utilized as described below. (Please refer to Pimmler (1994) for the independent clustering of the spatial, energy, and information type interactions.)

The results obtained from our climate control system example highlight a number of significant issues regarding the formation of product architecture (system chunks) and team structure (team chunks).

Flow and exchange of materials between the elements is critical to achieve the overall functionality of the climate control system. Therefore, we begin to develop system and team chunks by considering only the materials interactions. This initial consideration suggests the following chunks, as shown in Figure 7:

Initial Chunks

Front End Air Chunk: This chunk comprises the radiator, engine fan, and condenser components. It represents the set of components located at the front of the engine compartment that are involved with heat transfer to the exterior air.

Refrigerant Chunk: This chunk comprises the condenser, compressor, accumulator, and evaporator core. These are the four traditional air conditioning system components, however, they are not located in one space.

Interior Air Chunk: This chunk comprises the evaporator core, heater core, blower motor, blower controller, and evaporator case elements. It represents the set of components located at the front of the passenger compartment that are involved in changing the interior air temperature.

This clustering identifies strong coupling of elements across the usual (geographic) system boundaries. In this case, a chunk is identified for the air conditioning components, but the heating system components (radiator and heater core) are spread across different chunks due to their spatial separation. It is also significant to note that two of the elements (evaporator core and condenser) are each assigned to two chunks. This forces the chunks to overlap, which identifies the areas requiring integration across chunks.

We then build upon this initial materials-based clustering by considering the perspectives of the spatial, energy, and information interactions. This final clustering is shown in Figure 8 and suggests modifying the above chunks in the following manner:

Final Chunks

Front End Air Chunk (Unchanged): This chunk comprises the radiator, engine fan, and condenser, as identified above in the Front End Air Chunk.

Refrigerant Chunk (Unchanged): This chunk comprises the condenser, compressor, accumulator, and evaporator core, also as identified in the above Refrigerant Chunk.

Interior Air Chunk (Revised): This chunk comprises the evaporator core, heater core, blower motor, blower controller, evaporator case, as identified in the above Interior Air Chunk, with the addition of the actuators due to their spatial interaction with the evaporator case.

Controls/Connections Chunk (Added): This new chunk comprises the air controls, refrigeration controls, heater hoses, command distribution, and sensors. Additionally, this chunk overlaps with the other three chunks.

The clustering resulting from consideration of all the interactions verifies not only the importance of the initial three chunks, but also identifies significant needs for integration across those chunks. This integrative role is played by the Controls/Connections Chunk.

DISCUSSION

Although perhaps not surprising, the clustering result is interesting because it suggests a departure from existing industrial practice. Ford's Climate Control Division had normally structured the system and team chunks to mirror the existing organizational structure without necessarily considering component function. It is common for development engineers and managers to consider only the familiar architecture used in previous designs. Organizations skilled at incremental innovation have difficulty in considering more radical alternatives. As Henderson and Clark (1990) showed, established firms frequently fail when confronted with novel architectures brought about by new technologies. A methodology which makes it more normal practice to consider alternatives may offer some promise in overcoming such a handicap.

Clustering Negative (Detrimental) Interactions

In our example, clustering for system and team chunks began by closely grouping positive (i.e. required) materials interactions into chunks. We then expanded the clustering to include the perspectives of the other interaction types, which resulted in mixing negative interactions within those positive interaction base chunks. This result may be desirable because the important design challenges are described by both positive and negative interactions. Therefore, both must be considered in order to structure system and team chunks. For example, assume that two elements, one sensitive to heat and the other producing heat are characterized as having a detrimental (–2) spatial-type interaction between them. If the system chunks are then defined using only positive interactions, these elements would be placed in different system chunks. However, if these elements are inadvertently placed next to one another in the product layout, the detrimental interaction could reappear. This situation can be avoided by clustering the two elements together in order to bring their interaction to the attention of the development team.

Out-Of-Chunk Interactions

For very dense or complex interaction matrices, it is not possible to cluster all interactions into small chunks. Rather, several out-of-chunk interactions will remain. These out-of-chunk interactions, which could be either positive or negative, can be viewed as system engineering "flags" which indicate that special attention is required. Indeed, some "flags" can be resolved independently, without unnecessarily complicating the system and team chunks. For example, the evaporator core and condenser are described as having a detrimental (-2) spatial interaction between them because they must operate in completely different environments. Since the environmental separation in this case is obvious, they should not be placed in the same chunk for only that reason. (They would, however, be placed in the same chunk due to energy and materials exchange interactions.)

Benefits and Other Considerations

We believe that the system engineering insight provided by this methodology is significant because "system engineering" is not consistently defined or applied in industrial practice. Allowing product systems and team structures to be defined by the interactions between the actual elements of the design eliminates the need to attempt assignment based on other less relevant criteria.

We believe that this analysis provides the following core benefits for product development:

1) The methodology offers potentially many insights into the structure of a product development problem. Conducting the integration analysis using various different objective functions provides alternate perspectives of the problem: a) clustering based on individual interaction types (spatial, energy, information, and materials), and b) identifying system engineering needs by considering both positive and negative interactions and all out-of-chunk interactions.

2) This analysis offers a methodology for developing product architectural and team chunks based on the design challenges involved. Ulrich and Eppinger (1994) consider interactions only after the architecture is chosen; here the interactions drive the process of developing architecture and team chunks.

3) The chunks developed here suggest a departure from industrial practice. Product development chunks used in industry are usually formed of mutually exclusive sets. (See for example, McCord and Eppinger (1993).) The interactions identified here suggest that overlapping of chunks is required to properly resolve issues common to two or more chunks.

This methodology has not yet been evaluated since it only applies to complex and costly design projects. Perhaps it could be tested with the following three-step evaluation: 1) Interview team members of such a project and ask them to describe the system and team chunks. 2) Perform the analysis. 3) Review the analysis with the team to see if the results are insightful.

In this paper, we motivate the analysis utilizing an example of incremental design. Therefore, physical elements, with well-known interactions, were used to describe the design. In a more novel design situation, functional elements would instead be used to describe the product concept. Since the information regarding interactions between functional elements would be less complete, it is unclear that this analysis would offer significant insights to structure that problem. Perhaps this analysis is only useful after a mature understanding of interaction information has been attained.

The integration analysis methodology requires key extensions before it can be applied more effectively in practice. 1) More structured guidelines for the initial system decomposition would be useful to provide consistent identification of functional and physical elements. 2) The proposed method for clustering needs to be structured in specialized algorithms. 3) Other, more strategic, architectural issues should be considered in the clustering process.

CONCLUSIONS

The methodology presented in this paper provides a useful tool for analyzing given design decompositions. This technique is especially suited to highly engineered products where the need for this type of analysis is greatest but has been ignored because of the complexity involved and the lack of suitable methods. The method can potentially reduce the complexity of the product development process by reducing the need for difficult coordination across large development teams.

The three steps to integration analysis are: 1) decomposing the system into elements, 2) documenting the interactions between the elements, and 3) clustering the elements into architectural chunks. Integration analysis is a process which can translate a given product decomposition into a product architecture and development team structure. This method reduces design complexity by considering interactions between product elements. Architectural and team chunks are structured so as to cluster essential interactions within the chunks while yet coordinating detrimental interactions so as to receive the proper attention during the development process. Our emphasis is on generating alternative architectures and team structures from which the most attractive choice can be made. Although not directly addressed in this paper, it is important to consider integration solutions based not only on interactions, but also on other strategic and product architecture concerns.

ACKNOWLEDGMENTS

Funding for this research has been provided by the MIT Leaders for Manufacturing Program. We appreciate the assistance of the Ford Motor Company Climate Control Division in developing the examples presented in this paper.

REFERENCES

Alexander, Christopher, *Notes on the Synthesis of Form*, Harvard University Press, Cambridge, 1964.

Eppinger, Steven D., et al., "A Model-Based Method for Organizing Tasks in Product Development," *Journal of Engineering Design* (Forthcoming), Vol. 1994.

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

Henderson, Rebecca M. and Kim B. Clark, "Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms," *Administrative Science Quarterly*, Vol. 35, No. 1, pp. 9-30, 1990.

Kusiak, Andrew, *Intelligent Manufacturing Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1990.

Kusiak, Andrew and Edward Szczerbicki, "Transformation from Conceptual to Embodiment Design," *IIE Transactions*, Vol. 25, No. 4, 1993.

Kusiak, A. and J. Wang, "Decomposition of the Design Process," *Journal of Mechanical Design*, Vol. 115, December, 1993.

Lovejoy, William S., "Rationalizing the Design Process," presented at the Conference on Design Management, Anderson Graduate School of Management, UCLA, September 17-18, 1992.

McCord, Kent R. and Steven D. Eppinger, "Managing the Integration Problem in Concurrent Engineering," Massachusetts Institute of Technology Sloan School of Management Working Paper 3594-93-MSA, 1993, 1993.

Pahl, Gerhard and Wolfgang Beitz, (K. Wallace, ed.), *Engineering Design, A Systematic Approach*, Springer-Verlag, New York, 1991.

Pimmler, Thomas U., "A Development Methodology for Product Decomposition and Integration," Massachusetts Institute of Technology Master's Thesis, 1994.

Simon, Herbert A., *The Sciences of the Artificial*, 2nd edition, MIT Press, Cambridge, MA, 1981.

Smith, Gerald F. and Glenn J. Browne, "Conceptual Foundations of Design Problem Solving," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, No. 5, September/October, 1993.

Steward, Donald V., *Systems Analysis and Management: Structure, Strategy and Design*, Petrocelli Books, New York, 1981.

Suh, Nam P., *The Principles of Design*, Oxford University Press, New York, 1990.

Ulrich, Karl T. and Steven D. Eppinger, *Product Design and Development*, McGraw-Hill, New York, 1994.

Ulrich, Karl T., "The Role of Product Architecture in the Manufacturing Firm," *Research Policy* (Forthcoming), 1994

Verho, Arto and Vesa Salminen, "Systematic Shortening of the Product Development Cycle," presented at the International Conference on Engineering Design, The Hague, Netherlands, 1993.

Wagner, Terrance C. and Panos Y. Papalambros, "A General Framework for Decomposition Analysis in Optimal Design," presented at the 23rd ASME Design Automation Conference, Albuquerque, New Mexico, September, 1993.

Warfield, John and J. Douglas Hill, (Edited by Benjamin B. Gordon), "A Unified Systems Engineering Concept," Battelle Monograph, No. 1, June, 1972.