Massachusetts Institute of Technology
Engineering Systems Division

Working Paper Series

ESD-WP-2002-03

DESIGNING A REQUIREMENT DRIVEN PRODUCT DEVELOPMENT PROCESS

Qi Dong
Massachusetts Institute of Technology
Department of Mechanical Engineering
Center for Innovation in Product Development

Daniel E. Whitney
Massachusetts Institute of Technology
Center for Technology, Policy, and Industrial evelopment

September, 2001

Proceedings of DETC 2001:
ASME 2001 International Design Engineering Technical Conferences
13th International Conference on Design Theory and Methodology
September 9-12, 2001, Pittsburgh, PA

DETC2001/DTM-21682

DESIGNING A REQUIREMENT DRIVEN PRODUCT DEVELOPMENT PROCESS

Qi Dong

Massachusetts Institute of Technology
Department of Mechanical Engineering
Center for Innovation in Product Development
77 Massachusetts Avenue Room 3-438
Cambridge, MA 02139
Telephone: (617) 253-2052
Fax: (617) 258-6427
e-mail: qdong@mit.edu

Daniel E. Whitney

Massachusetts Institute of Technology
Center for Technology, Policy, and Industrial Development
77 Massachusetts Avenue Room E40-243
Cambridge, MA 02139
Telephone: (617) 253-6045
Fax: (617) 258-6794
e-mail: dwhitney@mit.edu

ABSTRACT

This paper presents a technique to obtain a Design Structure Matrix (DSM) from a Design Matrix (DM). This technique enables us to obtain the design information flow pattern at early stage of the design, and apply the DSM system analysis and management techniques at the time when the most important decisions about the system and the design are made. The validity of this method is proven using a case study on the design integration process of an electrostatic chuck used in semiconductor wafer processing. The algorithm underlying this technique is also proven logically and mathematically to be valid.

Introduction

The product development process of complex products such as cars and airplanes takes the collaborative effort of hundreds even thousands of people over the course of several years. These products themselves usually consist of hundreds or thousands of parts and assemblies, all of which must be designed, tested, integrated, and tested again to meet customer requirements. Two large and interacting systems emerge in this picture—the system of the parts and assemblies forming the product, and the system of people and teams involved in the design process. In order to design a product development process that delivers quality products better, faster, and cheaper, we must address the following three issues. First, we must be able to capture, understand, and manage the interactions occurring in the system of the product and the system of the design teams. Second, we must also be able to capture and analyze the system interactions as early as possible in the product development process, when the most important decisions about the product design and the design process are made, and the cost of changes is at its minimum. Third, we must ensure the product design and the process are closely aligned with the product requirements.

The Design Structure Matrix (DSM) method provides answer to the first issue. Past research has proven DSM is a powerful analysis tool to capture complex system interactions, to suggest task sequences to minimize rework, to predict project length and cost, etc [9, 11-17, 19]. However, the existing DSM approach is limited in predicting system interactions before detailed design is carried out. The causes of this limitation is explained as follows.

The first step of any DSM analysis is to construct a DSM of the system of interest. Currently, DSM researchers obtain the DSM's through interviewing the engineers and mangers that are working on the system of interest, as well as reading design documentation. This method works well when the product development process is already well into its detailed design phase. The interviewees can tell the system interactions through their work experiences and the documentation of the system is already available. However, at this stage of the design, it is usually too late to make any significant changes to the design or the process. Therefore, the proposed changes for improvement by DSM analyses usually stay as lessons-learned for future reference, rather than taking immediate effect. Hence, the traditional DSM method lacks the mechanism to construct the DSM's at early stage of the design. A further implication is that we cannot obtain a DSM for a new product that has never been designed before.

In addition, the traditional DSM method captures the interaction among elements in a system, but does not explicitly capture the reasons for the interactions. Hence, how the design requirements are met through the interactions within the system is not explicitly addressed by the traditional DSM method.

This paper presents a technique to obtain a DSM from a Design Matrix. A Design Matrix (DM) is a matrix used in the Axiomatic Design theory to relate requirements to design parameters [7]. Constructing a DM is easy at the early stage of the design process because the relationship between the requirements and the design parameters is what engineers naturally think about at that stage of the design process, while the relationship between the design parameters is harder to predict at that time. The result of this conversion technique is a DSM derived from the product requirements for the specific design concept of choice. Using this DSM, the existing DSM techniques for system analysis and process planning can be carried out at the early stage of the design, and make maximum impact to the rest of the design process with minimum cost. Together with the existing DSM method, this technique provides a suite of tools and methods to address the three issues mentioned earlier—capturing system interactions, analyzing the system before carrying out detailed work, and using requirements to drive the product design and design process.

The rest of the paper consists of six sections. First, we visit some of the previous research work that inspired the authors of this paper. Second, the detailed procedures of this technique are presented. Third, we introduce a case study to test the validity of this technique. Fourth, the analysis and observation made from the case study is presented. Last, since this paper is a part of the results from an on-going PhD thesis research, we will give the summary and the next steps for this research.

Related Work

We can learn to appreciate the various technical, organizational, and business issues in developing simple and complex products through the reading of many existing product development literatures. One of the most classical product development books is from Ulrich and Eppinger [1]. To deal with the system issues in product development, Rechtin has written a book of heuristics in dealing with complex systems and product design [2]. To get a less technical feeling of the drama involved in developing a complex product, Walton's *Car* [3] is an excellent reading to see what it takes to design a car.

The product development process of large products is challenging because us human beings have limited capability when facing complex system interactions. The famous Miller's Law [5] states that average human minds can deal with only seven plus minus two things without the aid of external tools, which in turn stresses the importance of system analysis and management tools such as DSM.

Facing the challenges from product development, one branch of research work stresses the importance of good requirements management. For instance, Ivy Hooks' latest book with Farry [4] demonstrates the importance of requirement traceability and the decisions at early stage of the design process. The House of Quality technique [6] addresses the requirements flow-down issues. Suh's Axiomatic Design [7] suggests using the zigzagging process to relate the requirements and the design parameters in a systematic way. The zigzagging process is shown in Guindon's work [8] to be close to the natural thinking process of the design engineers, and therefore is a more practical design process to implement. The concept of functional requirements, design parameters, and design matrix later used in this paper are presented in Suh's book [7]. Although many of the Axiomatic Design concepts are used in this paper, the technique presented here does not completely agree with the axioms in Suh's book. Axiomatic Design urges that a good design must have a DM that is diagonal or lower triangular. In our research, we accept the fact that the DM will not always meet these conditions. Our results are most applicable to exactly those cases. More discussion of this point is in the case study analysis section of this paper.

Another branch of the research work in the field of product development stresses managing the system interactions in the development of complex products. Steward [9] and Warfield [10] are the founders of the DSM method. Introduction to DSM method and its applications in the product development process is provided in Eppinger et al. [11]. Pimmler and Eppinger [12] first used the DSM to decompose design teams. McCord and Eppinger used DSM technique to study the integration problem in systems [13]. Many more case studies are available on the MIT DSM web site [14]. Smith and Eppinger [15] first evaluated the convergence of the design iterations. Carrascosa [16] and Browning [17] developed models to predict the cost and length of the project.

Much of the mathematics involved in the technique introduced in this paper can be found in typical applied mathematics books, such as the one by Strang [18]. Steward [19] also has written about the output variables, a concept also used extensively in this paper.

The Method to Obtain DSM from DM

The technique of creating a DSM at early stage of the design process consists of three major steps:

Step 1: Construct a Design Matrix (DM).

Step 2: Choose an output variable in each row of the Design Matrix (later on in this paper, the section entitled "The Uniqueness of Output Variable Choice" will demonstrate that the only valid output variable choices are the elements on the diagonal of the DM).

Step 3: Permute the matrix by exchanging rows so that all the output variables are on the diagonal. Re-name each row according to the names of the columns, and then we get a DSM of the design parameters.

To demonstrate this process, we will take a hypothetical example involving a 3 by 3 design matrix. However, a few definitions may be necessary for readers that are less familiar with Axiomatic Design.

A Design Matrix (DM) is defined in axiomatic design as the matrix relating the functional requirements to the design parameters [7]. Each mark in the DM indicates that the design parameter in the column affects the functional requirement in the row. In many ways, DM's are very much like the second house of QFD [6]. Yet, the construction of DM leads to decomposing functional requirements along with the design parameters—the zigzagging process. The result is a well-documented design concept tightly connected to the design requirements.

An output variable is a concept used in solving systems of linear equations. It is the variable chosen to be solved using a particular equation. For example, we need to solve for x and y in the following system of equations:

X + Y = 5 (1)

X – 5 * Y = 2 (2)

We can solve for X using Eq. (1) and Y using Eq. (2):

(1) => X = 5 – Y

Substitute into Eq. (2),

(2) => 5 – Y + 5 * Y = 2

Y = -0.75

Therefore, X = 5.75. Here, X is called the output variable of Eq. (1) and Y is called the output variable of Eq. (2).

The hypothetical example used here to demonstrate the three steps in applying this technique has three functional requirements and three design parameters. The first step is to construct the design matrix shown below. Here we assume the functional requirements at the highest decomposition level are available.

	DP1	DP2	DP3
FR1	X		X
FR2	X	X	
FR3		X	X

The second step is to choose an output variable from each row. They are indicated by a “X0”.

	DP1	DP2	DP3
FR1	X		X0
FR2	X0	X	
FR3		X0	X

The significance of the output variable choice is:

DP3 = f (FR1, DP1)

DP1 = f (FR2, DP2)

DP2 = f (FR3, DP3)

The third step is to permute the rows so that the output variables (the X0’s) are on the diagonal. Then rename the rows according to the DP’s of the columns.

	DP1	DP2	DP3
DP1	X0	X	
DP2		X0	X
DP3	X		X0

Now we have a DSM of all the design parameters. We may also obtain a DSM of the FR’s by permuting the columns of the DM.

Case Study: An Electrostatic Chuck Design Integration Process

A. Description of the Component and Its System Context

The Electrostatic Chuck (ESC) (see Figure 1) in this case study is used in various semiconductor wafer process modules. When in use, ESC loads the chuck table and the wafer with opposite static charges. Consequently, the wafer is held down on the chuck table. After the wafer is processed, the chuck table and wafer are discharged, and the wafer is de-clamped. Contrary to a conventional mechanical clamp, ESC does not exert contact force on the processing side of the wafer. Hence, ESC is particularly suitable for processing wafers plated with brittle materials. ESC also contains backside gas channel and cooling system design to maintain wafer temperature during the process. In addition, an interface plate is designed so that it could be assembled in various wafer-processing modules, include Physical Vapor Deposition modules, Chemical Vapor Deposition modules, Ion Beam Etching modules, etc.

The ESC is a part of various wafer-processing modules. The process modules are assembled on the wafer processing cluster machines, and controlled by the central controller. A wafer handling system, which is also controlled by the central controller, transports wafers between the modules to complete a set of necessary deposition or etching processes. Details of the systems interactions are shown in Figure 2.

B. The Design of the Case Study

This case study was conducted in a company who produced semiconductor manufacturing equipment. When the case study started, the advanced product development group in the company had completed the technical feasibility study of the ESC as a stand-alone component. The next step facing the engineers in the company was to transfer the ESC technology to the product design group so that the chuck can be integrated with the existing wafer processing modules the company already had on the market. The purpose of this case study was to construct a DSM prior to the occurrence of the design integration, in order to use the resulting DSM to guide the system integration and testing phase.

The case study followed the three steps stated in the “Method” section in this paper. First, a DM was constructed based on the system integration design requirements. The design requirement documentation of ESC was used to construct the DM. The technical experts in the advanced product development team were consulted during the data collection process. Next, following the steps described in section III, a DSM concerning the interactions among the Design Parameters was constructed. During the conversion of from DM to DSM, the elements on the diagonal of the DM were chosen as the output variables. The reason for this is explained in the analysis section of this paper. The DSM was then modified and validated by technical experts involved in the integration process. Last, the DSM was partitioned to identify the sequence of tasks and the inevitable design iteration loops.

C. *Case Study Results*

The design requirements were decomposed using the zigzagging method [7]. Figure 3 shows how this process was carried out for the first Functional Requirement FR1. The rest of the FR's were decomposed in the same way, and a DM was constructed for the system integration requirements of the ESC (Figure 4). Due to its large size, only a part of the DM is shown in Figure 4 for legibility. Figure 5 shows the resulting DSM after partitioning. This DSM only contains the DP's at the leaf level of the decomposition tree in the DM (Figure 4), because the DP's at the leaf level of the tree are the parameters engineers actually work on during the integration process. The text in Figure 5 is not readable due to the size of the DSM. The authors would like to point out that the topology of the DSM rather than the text is what we based our conclusions upon. After partitioning the DSM, two iteration areas are identified—the design work to physically integrate the ESC with the existing process modules and the design of the control circuit (Figure 5).

Analysis of the Case Study

A. *The Validity of the Technique of Obtaining DSM from DM*

In order to verify the correctness of the DSM constructed from DM, the DSM was presented to 5 technical experts. These experts included one person from each of the functional groups that were preparing the integration process of the ESC—the advanced technology department, the mechanical engineering group, the electrical engineering group, the software and controls group, and the system engineering group. Each expert reviewed the interactions in the DSM and agreed that most of the interactions captured were correct and reasonable. A few modifications were proposed. However, these proposed modifications were caused by the missing information during the construction of the DM, not the technique of converting DM to DSM. In the end, all of the proposed modification on the DSM could be correctly incorporated into the original DM. Therefore the case study demonstrated that the matrix conversion method gives us a valid DSM.

One note on the validation of this technique is that the DSM was compared with the expert prediction of the system interactions, rather than what actually happened later on in the integration phase. The accuracy of the prediction using this technique is discussed in the next section.

B. *The Accuracy of the Prediction*

After the data was collected for this case study, the ESC integration work was carried out in the company. In observing engineers' work, many of the issues that came up during the integration phase were correctly predicted by the DSM constructed from DM. Therefore, the DSM obtained from DM can serve well as a prediction tool and help the planning and management of the design process.

Nevertheless, we do not know what we do not know. The DSM, being constructed at early stage of the design process, may not capture all of the subtle interactions among the system elements. Ulrich and Eppinger [1] identified two types of system interactions. The first type is called the fundamental interactions. These interactions correspond to the design intentions. The second type of interactions is called the incidental interactions. These are interactions that were not intended by the designers but arise from unexpected sources like electromagnetic interference, vibration, or other sources that could depend on detailed design choices made along the way. Examples include choice of materials or values of individual design parameters.

The completeness of the fundamental interactions captured increase as our knowledge of the design increases. Henderson and Clark [20] proposed four types of innovations in product design—incremental innovation, modular innovation, architectural innovation, and radical innovation, with increasing differences between generations of the same product. For products whose pace of change leans towards the incremental innovation side, the fundamental interactions mostly can be captured through one complete product program. Therefore, a large percentage of the DM and DSM built for the last generation of the product can be reused for the next generation. The accuracy of the prediction will increase as our experience with the product grows. For products whose pace of change leans towards the radical innovation side of the scale, or for products that have never been designed before, the DSM built at early stage of the design may miss some of the subtle the fundamental interactions. However, if we keep the DM and DSM as a live document through the design process, the DSM can still give us great guidance. Having this tool is better than purely relying on intuition. In the case the new product is successful and subsequent generations of the product will be built in years to come, the DM and DSM capture design histories and improve the prediction on the next generation of products.

However, the incidental interactions show a different picture. Due to their nature, we may never be able to capture and predict them all. Some of the incidental interactions learned from previous products can be kept on record and used as design constraints for the next product, but some will always come up as a surprise. The goal of the technique presented here is not to find an autonomic system prediction tool, but rather to help the documentation and reuse of expert experiences at the system level, and to conduct requirements-driven system analysis and management using the extensive DSM tool kit.

C. *The Significance of Obtaining a DSM from DM*

The significance of using this technique includes the following three aspects.

1. Enable the Use of DSM at Early Stage of the Design Process

This technique enables us to use the DSM system analysis tools at early stage of the product development process when the most important decisions about the system are made. To construct a DSM using this technique, we only need to know all the functional requirements and how the chosen design concept fulfills the requirements (the process of constructing a DM). Producing these two pieces of information is the task of design teams at early stage of the design process anyway. The DM captures these pieces of information in a systematic way without adding much work to what people are already doing. From the DM, this technique enables us to get a DSM that shows the system complexity of the chosen design concept. When several design concepts are available, we may compare their DSM's and choose one with less complex system interactions. Sometimes for various reasons such as market trend or technology maturity, we have to choose a design concept with complex system interactions. Then the DSM obtained at the early stage of the design process can serve as a process planning and management tool to help the design groups go through the design process with minimum amount of rework. In short, this technique enables us to compare the system complexity of design concepts, planning and managing the design process with knowledge about the system interactions before the resources are committed, so that we could avoid costly rework later on in the design process.

One question is that if the experts could review and validate the interactions in this case study, why couldn't we directly construct a DSM from the experts using the traditional DSM construction method? It is possible to get a DSM in the traditional way at early stage of the design. However, the traditional way has many shortcomings when used at early stage of the design. First, it constructs the DSM through interviewing design experts and asking them to list the important system parameters and the interactions among these parameters. At early stage of the design when the actual design work is not yet carried out, in our case, when the integration work had not yet started, it is rather difficult to generate parameters and their interactions without real examples to related to. The difficulty may be reduced for products that have been designed before, and the difficulty increases when the design is no precedence. Second, because we solely rely on the experts' subjective thinking process, the traditional DSM construction process may take a long time and we don't really know when we are getting a good-enough DSM. Most importantly, the process of constructing a DSM in the traditional manner is not an existing design practice in most companies. The interview and documentation processes add work to the already hectic phase in the design process. The engineering expert may be resentful to doing the DSM study because they are usually the busiest at this phase of the design. The technique presented here naturally follows the experts' thinking process at the early stage of the design process (as described in the last paragraph), and generates a DSM without adding a separate piece of work. Therefore, it is an easier and more systematic way of obtaining a DSM.

2. The Resulting Design Process is driven by the Design Requirements

The second significance is that the resulting DSM gives us a design process driven by the functional requirements, since the DSM is converted from the DM, which captures the requirements flow-down information. Therefore, the resulting process captures the underlying structure of the design problem rather than capturing the as-is process like the traditional DSM method does.

A simple example in this case study demonstrates this point. The ESC has a requirement to transfer the heat generated during wafer processing away from the wafer. Three DP's--the backside gas, the chuck material, and the chuck cooling system design--contribute to the fulfillment of this heat transfer requirement (see Figure 1). The DM captured this relationship. The DSM converted from the DM thus shows an iteration loop among the above three DP's. In other words, the resulting DSM shows that trade-off studies should be done among the three design parameters to fulfill the heat transfer requirements together (see Figure 5). However, when experts were exposed to the initial DSM, they pointed out that the design teams do not go through this trade-off iteration. Instead, each DP was independently optimized for the simplicity of inter-team coordination. However, the experts agreed that they could find potential cost savings and performance improvements if they considered heat transfer iteration as proposed in the DSM from DM. On the other hand, if we were to make a DSM using the traditional interview method, then we will get a DSM of the as-is process without the iterations among the three DP's.

The US Navy Admiral Grace Hopper once said "the most damaging phrase in the language is: it's always been done that way." The traditional way of obtaining DSM can only capture the as-is process--the way in which things have always been done. As shown in the heat transfer example, the DSM converted from the DM captures the essence of the design problem by considering how requirements are addressed with the chosen design concept. Consequently, the proposed improvements are based on the underlying structure of the design problem, and the design process is driven by the requirements.

3. Enable Requirements Management and System Level Design Knowledge Management from Early On

Ivy Hooks [4] has pointed out the importance of managing the requirements during the design process. The construction of DM is an excellent way of capturing the decomposition and flow-down of requirements. In addition, being able to obtain a DSM from the DM enables us to trace how requirements flow into the interactions among system elements starting at the early stage of the design process. Both DM and DSM can be updated throughout the design process of the product. Together, they provide a complete documentation of how requirements are met by the design, and how the elements within the design interact. In addition, we mentioned in the introduction part of this paper that the people and organization involved in the design process also form a system. In both the DM (see Figure 4) and DSM (see Figure 5), we can capture the team responsibility on the requirements and design parameters. Therefore, the DM and DSM also become guides to the team interactions.

D. The Uniqueness of the Output Variable Choice

According to the definition of output variables, each row and column in a DM can only contain one output variable [19]. It can easily be proved that the choice of output variable is unique where iteration does not exist in the DM. However, when coupled blocks exist in the DM, the choices of output variables are not unique. This situation is shown in the following example from the ESC study.

FR1 = Reduce Clamping Voltage
FR2 = Provide on/off and magnitude control for the voltage
FR3 = Prevent the chamber RF power from affecting the electric network
DP1 = Chuck dielectric Layer Thickness
DP2 = Electric Circuit for the Voltage Control
DP3 = DC choke circuit

Based on Axiomatic Design [7], each DP is the main solution to the FR with the same ID. For example, the DC choke circuit (DP3) is designed to prevent the chamber RF power from affecting the electric network (FR3). However, DP's with different ID may also have side effects on each FR. For instance, the choke circuit (DP3) has to be integrated with the electric circuit for the voltage control (DP 2), hence the design of DP2 constrains the design of DP3 and hence the ability of the product to fulfill FR3. This relationship causes an off-diagonal mark in the DM. The final DM for this example is:

	DP1	DP2	DP3
FR1	X	X	
FR2	X	X	X
FR3		X	X

This is a coupled design, and the choice of output variable may not be unique. Figure 6 compares two possible output variable choices and the resulting DSM. The conclusion from the case study data is that the only choices of output variables are those that are on the diagonal of the DM. This conclusion can be proven in two ways—the logic of the design process and the convergence of the design iteration (the mathematical proof).

1.The Logic Proof

At the bottom of Figure 6, we can read the interpretation of the difference choices for output variables. Choosing non-diagonal elements as the output variables leads to a non-executable design process. It is like designing a component not for its main purpose, but rather for its side effects. Choosing diagonal elements as the output variables gives us a feasible design process through the resulting DSM. Therefore, the diagonal elements are the correct choice for output variables in a DM.

2.The Mathematical Proof

Since different output variable choices give us different DSM results (Figure 6), we ask if one DSM provides an iteration process that converges better than the other does. First, we will assign sensitivity numbers to each pair of FR-DP relationships. In this example, we have

	DP1	DP2	DP3
FR1	0.75	0.2	
FR2	0.2	0.9	0.2
FR3		0.4	0.9

The values in the DM mean the percentage of change in fulfilling the corresponding FR caused by one unit of change occurs in the DP. These numbers are unitless. For instance, the value 0.75 means one unit change of DP1 will cause the FR1 to change 75%. The values on the diagonal should always be greater than the values off the diagonal. Otherwise, we have a design in which certain components have large side effects on certain FR's that overshadow the effect of the main functional components for those FR's.

Figure 7 shows the comparison between two different output variable choices. Choosing the diagonal elements in the DM as output variables gives a converging iteration loop, while choosing non-diagonal elements gives a diverging iteration loop. The relationship between Eigen values of DSM and the convergence can be found in Smith et al. [15] and Strang [18]. In practical sense, when the DSM has values greater than one, some design parameters have to change more than 100% to compensate the change in other design parameters, which is impossible.

In conclusion, the choice of the output variable is unique when applied to product development processes. Only by choosing diagonal elements as the output variables, could we have a converging and feasible design process.

E. Relationship with Axiomatic Design

The first Axiom in the Axiomatic Design method states that there always exists an uncoupled design or a decoupled design solution [7]. However, sometimes the uncoupled or decoupled design is not practical from business point of view. The risk of using the technology proposed by the Axiomatic Design solution may be too big. The cost of building new manufacturing facilities may be too large. The market trend may be different. On the other hand, DSM has been proven to be excellent in analyzing the existing design process and propose management improvement to deal with system interactions, especially when the DM is not uncoupled or decoupled. In fact, if the DM is uncoupled or decoupled, the resulting DSM will be lower triangular and there will be no iterations. The technique presented in this paper provides a bridge between the two methods. When Axiomatic Design fails to provide uncoupled or decoupled solutions that are feasible in the business context and design iterations are inevitable, DSM can be used subsequently to apply management leadership on the design process. As a result, the design teams can go through the inevitable design iterations smoothly.

In mathematically proving the uniqueness of the output variable, we have seen that when the values in the resulting DSM are less than one, the iteration always converges. This is also proven in applied mathematics [18]. In fact, the smaller the values in the DSM are, the faster the iteration converges. When the values in the DSM are approaching zero, we have almost no iteration. This happens when in the DM, there are only values on the diagonal—the uncoupled design in Axiomatic Design definition. Therefore, the principles of the Axiomatic Design and the technique presented here are well aligned.

Conclusions

This paper presents a technique to convert a Design Matrix into a Design Structure Matrix. The validity of the technique is demonstrated by the case study on the Electrostatic Chuck design integration process. The uniqueness of the algorithm is proven using logic as well as mathematics.

The technique presented here enables us to use DSM as a system analysis tool and a process management tool at early stage of the design when the most important decisions about the design are made. The DSM produced using this technique provides us a design process that is led by the design requirements. This technique can be used to manage requirements flow down, and capture system level design knowledge. This technique serves also as a bridge between the Axiomatic Design and Design Structure Matrix method, so that we can use both methods to our advantage according to the business situation we are facing.

.

Next Steps

This paper is the output of an ongoing doctoral thesis research. The following work will be done as next steps:

1. Capture the system interactions occurring during the actual system integration process of the ESC. Compare the actual interactions with the predicted interactions from the DSM built during the case study. Evaluate the degree of accuracy of the prediction using this method. Meanwhile, understand how an organization learns about the design of a product over time.

2. The Axiomatic Design theory does not have a structured way of addressing design constraints, such as reliability, MTBF, etc. More studies on different types of requirements and constraints will be done to see how this technique can capture all of the upstream inputs for the design and flow them downstream into the design process.

ACKNOWLEDGMENTS

Work supported by the MIT Center for Innovation in Product Development and NSF cooperative agreement EEC-9529140. Many thanks to Veeco-CVC for hosting the case study.

REFERENCES

[1] Ulrich, K. T., and Eppinger, S. D., 2000, *Product Design and Development,* 2nd Edition, Irwin McGraw-Hill, Boston.

[2] Rechtin, E., 1991, *System Architecting*, Prentice-Hall, Englewood Cliffs, NJ.

[3] Walton, M, 1997, *Car: A Drama of the American Workplace,* Norton, New York.

[4] Hooks, I. F., and Farry, K. A., 2000, *Customer Centered Products, Creating Successful Products Through Smart Requirements Management,* American Management Association, Boston.

[5] Miller, G. A., 1956, "The Magical Number Seven, Plus or Minus Two," *Psychology Review* (63) pp81-97.

[6] Clausing, D., 1993, *Total Quality Development : A Step-by-step Guide to World Class Concurrent Engineering,* ASME Press, New York.

[7] Suh, N. P., 2000, *Axiomatic Design*, Advances and Applications, Oxford University Press.

[8] Guindon, R., 1990, "Design the Design Process: Exploiting Opportunistic Thoughts," *Human-computer Interaction*, Volume 5, pp305-344.

[9] Steward, D. V., 1981, *Systems Analysis and Management. Structure, Strategy, and Design*, Petrocelli Books, Inc.

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

[11] Eppinger, S. D., Whitney, D. E., Smith, R. P., and Gebala, D. A., 1994. "A Model-based Method for Organizing Tasks in Product Development." Research in Engineering Design, vol. 6, pp1-13.

[12] Pimmler, T. U. and Eppinger, S. D., 1994, "Integration Analysis of Product Decomposition." ASME Design Theory and Methodology Conference. pp1-9.

[13] DSM web site at MIT. http://web.mit.edu/dsm.

[14] McCord, K. R., and Eppinger, S. D., August 1993, "Managing the Integration Problem in Concurrent Engineering," MIT Sloan School Working Paper #3594-93-MSA.

[15] Smith, R. P. and Eppinger, S. D., 1997. "Identifying Controlling Features of Engineering Design Iteration." Management Science. Vol. 43, No. 3, pp. 276-293.

[16] Carrascosa, M, Eppinger, S. D., and Whitney, D. E., 1998 "Using the Design Structure Matrix to Estimate Product Development Time." DAC98-6013, Proceedings of the 1998 ASME Design Engineering Technical Conference.

[17] Browning, T R., 1998, "Modeling and Analyzing Cost, Schedule, and Performance in Complex System Product Development," Ph.D. Thesis (TMP), Massachusetts Institute of Technology, Cambridge, MA.

[18] Strang, G., 1986, *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, Wellesley, MA.

[19] Steward, D. V., October 1962, "On an Approach to Techniques for the Analysis of the Structure of Large Systems of Equations," *SIAM Review*, Vol. 4, No. 4.

[20] Henderson, R. M., and Clark, K. B., 1990, "Architectural Innovation: The Reconfiguration of Existing Product

Technologies and the Failure of Established Firms," *Administrative Science Quarterly*, 35 (1990), pp9-30.

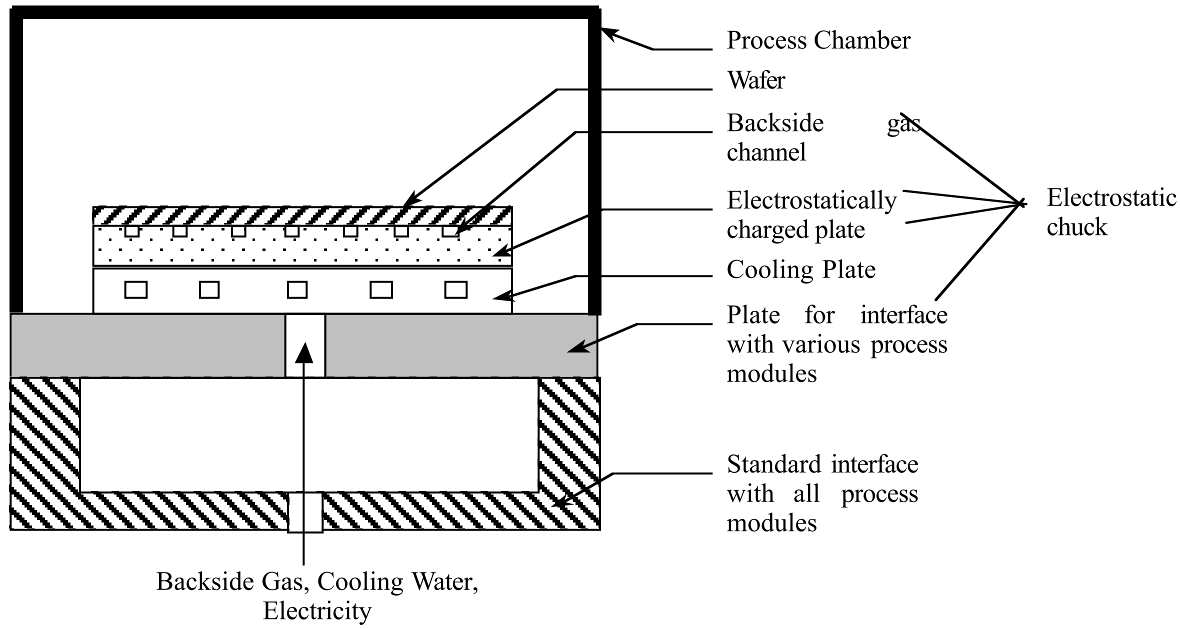

Figure 1: Schematic of a Electrostatic Chuck

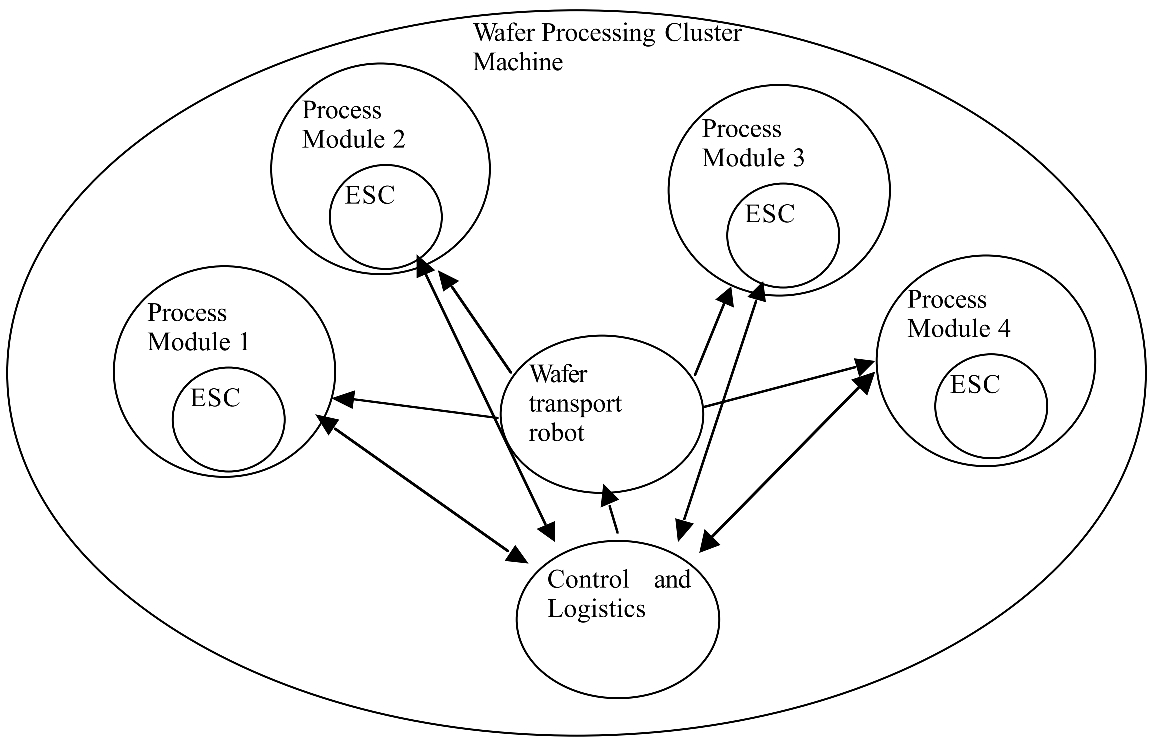

Figure 2: System View of the Electrostatic Chuck

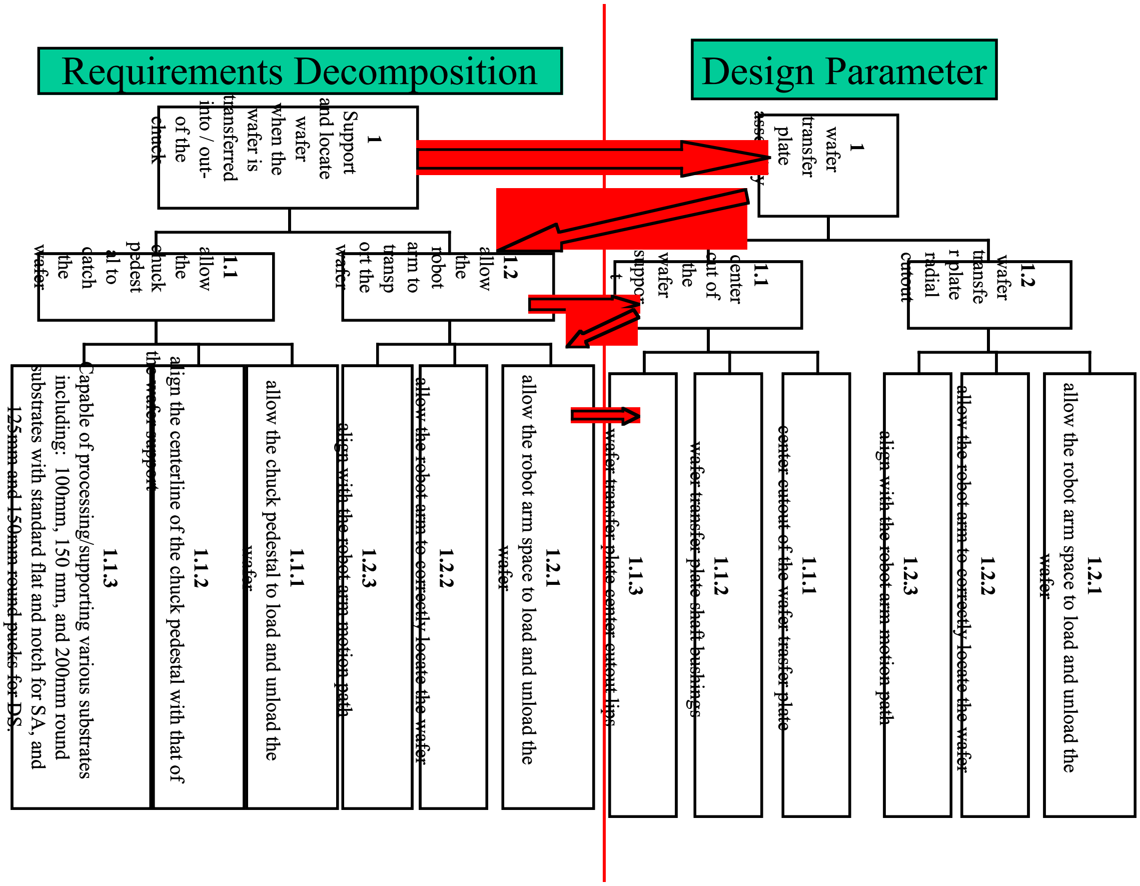

Figure 3: The Zig-zagging Process of Decomposing FR1 and DP1

			DP ID	1	1.1	1.1.1	1.1.2	1.1.3	1.2	1.2.1
			team			3	3	3		3
FR ID	FR / DP	team	source of FR	wafer transfer plate assembly	center cut of the wafer support	center cutout of the wafer trasfer plate	wafer transfer plate shaft bushings	wafer transfer plate center cutout lips	wafer transfer plate radial cutout	wafer transfer plate radial cutout
1	Support and locate wafer when the wafer is transferred into / out-of the chuck	1,3	Matt Coon from the drawings	x	x	x	x	x	x	x
1.1	allow the chuck pedestal to catch the wafer	1,3	Matt Coon from the drawings	x	x	x	x	x	x	x
1.1.1	allow the chuck pedestal to load and unload the wafer	1,3	Matt Coon from the drawings	x	x	x	x	x		
1.1.2	align the centerline of the chuck pedestal with that of the wafer support	1,3	Matt Coon from the drawings	x	x	x	x	x	x	
1.1.3	Capable of processing/supporting various substrates including: 100mm, 150 mm, and 200mm round substrates with standard flat and notch for SA, and 125mm and 150mm round pucks for DS.	1,3	White Paper Specifications (2)	x	x	x		x	x	x
1.2	allow the robot arm to transport the wafer	1,3	Matt Coon from the drawings	x	x	x	x		x	x
1.2.1	allow the robot arm space to load and unloac the wafer	3	Matt Coon from the drawings	x					x	x

Figure 4 The Design Matrix for ESC (only FR1 and DP1 are shown here)

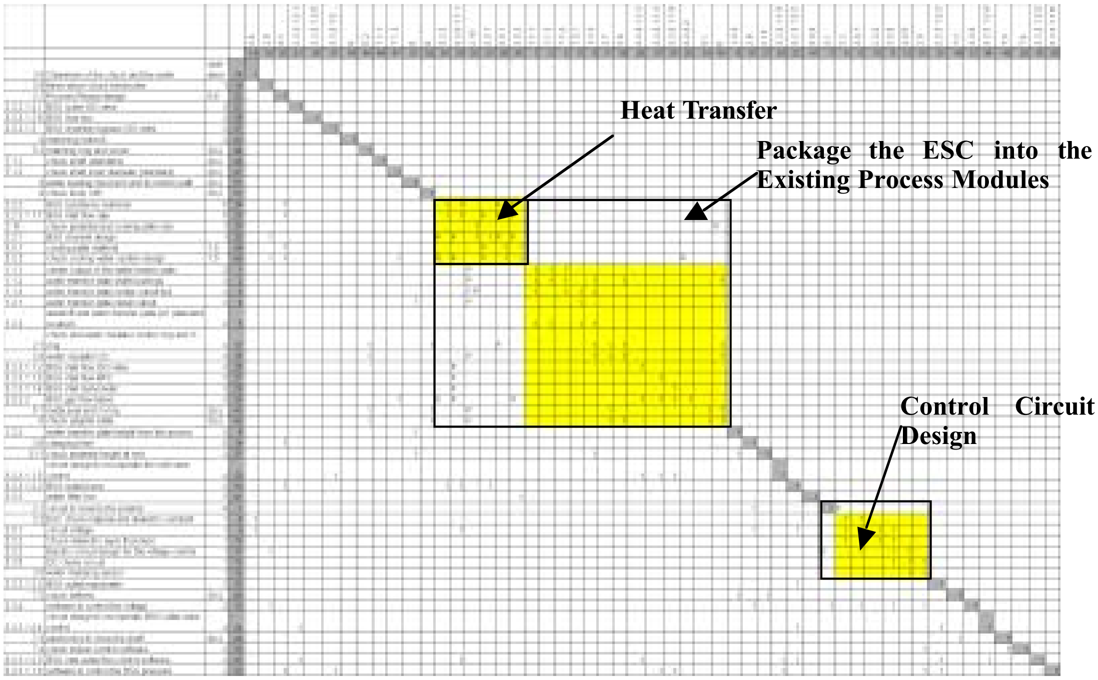

Figure 5: ESC System Integration DSM Derived from DM

DM

	DP1	DP2	DP3
FR1	1	1	
FR2	1	1	1
FR3		1	1

DP1 = f (FR1, DP2)
DP2 = f (FR2, DP1, DP3)
DP3 = f (FR3, DP2)

DSM

	DP1	DP2	DP3
DP1	1	1	
DP2	1	1	1
DP3		1	1

Interpretation

- *Given the information about the requirement FR1 (Reduce Clamping Voltage)and DP2 (Electric Circuit for the Voltage Control), design DP1 (Chuck dielectric Layer Thickness).*
- *Given the information about the requirement FR2 (Provide on/off and magnitude control for the voltage), DP1 (Chuck dielectric Layer Thickness), and DP3 (DC choke circuit), design DP2 (Electric Circuit for the Voltage Control).*
- *Given the information about the requirement FR3 (Prevent the chamber RF power from affecting the electric network) and DP2 (Electric Circuit for the Voltage Control), design DP3 (DC choke circuit).*

	DP1	DP2	DP3
FR1	1	1	
FR2	1	1	1
FR3		1	1

DP2 = f (FR1, DP1)
DP1 = f (FR2, DP2, DP3)
DP3 = f (FR3, DP2)

	DP1	DP2	DP3
DP1	1	1	1
DP2	1	1	
DP3		1	1

- *Given the information about the requirement FR1 (Reduce Clamping Voltage) and DP1 (Chuck dielectric Layer Thickness), design DP2 (Electric Circuit for the Voltage Control).*
- *the information about the requirement FR2 (Provide on/off and magnitude control for the voltage), DP1 (Chuck dielectric Layer Thickness), and DP3 (DC choke circuit), design DP2 (Electric Circuit for the Voltage Control).*
- *Given the information about the requirement FR3 (Prevent the chamber RF power from affecting the electric network) and DP2 (Electric Circuit for the Voltage Control), design DP3 (DC choke circuit).*

Figure 6: Logical Proof for the Various Output Variable Selections

DM:

	DP1	DP2	DP3
FR1	0.75	0.2	
FR2	0.2	0.9	0.2
FR3		0.4	0.9

$$\Delta DP1^* = \frac{1}{0.75} * \Delta FR1^* - \frac{0.2}{0.75} * \Delta DP2^*$$

$$\Delta DP2^* = \frac{1}{0.9} * \Delta FR1^* - \frac{0.2}{0.9} * \Delta DP1^* - \frac{0.2}{0.9} * \Delta DP3^*$$

$$\Delta DP3^* = \frac{1}{0.9} * \Delta FR1^* - \frac{0.4}{0.9} * \Delta DP2^*$$

DSM:

	DP1	DP2	DP3
DP1	0	0.2/0.75	
DP2	0.2/0.9	0	0.2/0.9
DP3		0.4/0.9	0

Maximum Eigen Value = 0.398
The iteration converges.

DM:

	DP1	DP2	DP3
FR1	0.75	0.2	
FR2	0.2	0.9	0.2
FR3		0.4	0.9

$$\Delta DP1^* = \frac{1}{0.75} * \Delta FR1^* - \frac{0.2}{0.75} * \Delta DP2^*$$

$$\Delta DP3^* = \frac{1}{0.2} * \Delta FR1^* - \frac{0.2}{0.2} * \Delta DP1^* - \frac{0.9}{0.2} * \Delta DP2^*$$

$$\Delta DP2^* = \frac{1}{0.4} * \Delta FR1^* - \frac{0.9}{0.4} * \Delta DP3^*$$

DSM:

	DP1	DP2	DP3
DP1	0	0.2/0.75	
DP2	0	0	0.9/0.4
DP3	0.2/0.2	0.9/0.2	0

Maximum Eigen Value = 3.211
The iteration does not converge.

Figure 7: Mathematical Proof for the Various Output Variable Selections
(ΔDpi* and ΔFri* denotes the percentage of change normalized with the nominal values of the particular variable)