Proceedings of IDETC/CIE 2007
ASME 2006 International Design Engineering Technical Conferences
& Computers and Information in Engineering Conference
September 4-7, 2007, Las Vegas, Nevada, USA

DETC2007/DAC-34510

Three Dimensional Design Structure Matrix with Cross-Module and Cross-Interface Analyses

Fabrice Alizon
Bucknell University, Lewisburg, PA 17837, USA
Keyplatform Company, 91 rue Faubourg St. Honoré, 75008 Paris, France

Seung Ki Moon
The Pennsylvania State University
University Park, PA 16802, USA

Steven B. Shooter[1]
Bucknell University
Lewisburg, PA 17837, USA

Timothy W. Simpson
The Pennsylvania State University
University Park, PA 16802, USA

ABSTRACT

Many companies that struggle with product variety and configuration management issues turn to a module-based design approach. Although this approach is well-known to be efficient for managing variety of a product family, current methods do not enable designers to handle both modularity and variety within a product family. The Design Structure Matrix (DSM) has been widely used to identify modules within a product, but its use to identify modules across a family of products has been limited. In this context we propose two tools based on an extension of the basic DSM to manage variety of an entire product family. The Variety Design Structure Matrix, DSM^V, handles variety of the product family and 3D Design Structure Matrix, DSM^{3D}, enables visual analysis of across the entire product family. These two tools, combined into a single approach, enable analysis of the product family at many levels – family product, module, and interfaces – to better specify modules and interfaces across all of the products in the family. A case study involving a family of three single-use cameras is used to demonstrate the application of these new DSMs and accompanying cross-module and cross-interface analyses. This new approach can be applied during detailed studies as well as in the early stages of the design process.

Keywords: Design Structure Matrix (DSM), variety, product family, modularity, interface

1. INTRODUCTION

Many companies that struggle with product variety and configuration management issues turn to a module-based design approach. This approach – especially through the use of Design Structure Matrix (DSM) [1] – helps designers specify modules and their interfaces. This macroscopic tool (gathering all the components of a given product at once) specifies what should be the modules of the product based on the physical relationship (interfaces) between components. Most recent studies focus on improvement of the basic DSM with significant results [2]; however, extensions of this tool have been close to the status quo. New considerations such as lifecycle product management, technology management, or product family design need to be included in the decision model to be more practical. In this study, we focus on product family design and extend the basic DSM model to target a three-dimensional tool helping designers for managing variety within a product family.

The basic DSM is based on a single product approach where all the components of this product are put in a matrix, then a clustering algorithm analyzes the physical relationships between components and identifies appropriate "chunks". These "chunks" are potential modules where component interactions within a module are maximized and component interactions between modules are minimized. This approach currently does not support a multi-product problem.

[1] Please address all correspondences to shooter@bucknell.edu.

Consequently, in this paper, we (1) model variety of the product family in the basic DSM, (2) overlap resulting DSMs of each product in the family, and (3) analyze what we denote as "cross-modules" – modules that span across products in the family – to help manage variety and the interfaces within a product family. Thus, one of the goals is to integrate variety management within the basic DSM (through a specific definition and color coding) to obtain what we call the variety DSM (DSM^V). The proposed approach is product-driven; as a consequence, each DSM of each product is performed separately, and the resulting DSMs are studied as a whole to manage variety across the family to identify adequate modules and specify their interfaces. A visualization tool supports this approach to represent the resulting 3D model.

Section 2 provides an overview of the related research, and the Variety Design Structure Matrix (DSM^V) and the 3D Design Structure Matrix (DSM^{3D}) are introduced in Section 3. A case study involving a family of power tools is then presented and discussed in Section 4. Section 5 concludes the paper with closing remarks and future work.

2. RELATED LITERATURE

Product architecture is the arrangement of functional elements into physical "chunks" that constitute the building blocks for a product or family of products [3]. A modular system architecture facilitates product variety by enabling the additional, removal, or substitution of one or more modules to the platform to realize a family products [4,5]. However, modules are often hard to define and identify when part of existing products. To highlight the complexity of identifying modules, Hirschi and Frey [6] conducted an experiment with designers and showed that as the number of variables increases, the effect of coupling among variables has a drastic effect on the solution procedure and on task completion time. The design problem can rapidly become complex, and designers need assistance to identify these modules. As a result, several methods have been proposed to identify modules in a system [7,8]. Stone, et al. [9] introduced a heuristic method to specify modules. Otto and Wood [10] proposed a heuristic to help identify modules in a given product architecture. Another well-developed tool in the literature to capture the architecture of a product and ultimately identify the modules in a product is the Design Structure Matrix (DSM), originally introduced by Steward [1]. It is a square matrix with the same label on rows and columns (see Figure 1). The shaded diagonal is composed of identical elements; in this example, a "●" indicates a relation between two elements. Reading across a row indicates the other elements to which the given element provides input, and looking down a column reveals the elements on which the given element depends for output. Thus, in the example in Figure 1, element *B* provides something to *C* and *D*, yet depends on *A*.

Browning [11] provides an overview of the DSM and defines the goal of the DSM as a matrix "to illuminate the structure and aid in the design of products, processes and organizations." There are two main categories of DSMs: *Static* and *Time-based*. *Static* DSMs model system elements existing simultaneously: products, physical organizations (group), etc. This group is usually used for clustering analysis. In *Time-based* DSMs, a relation between two elements (rows or columns) indicates a flow through time for these two elements: upstream activities in a process precede downstream activities. Interfaces are then called *feed-forward* or *feedback*. In this case, the goal is to conduct a sequencing time analysis and cluster tasks together to minimize feed-forward and feedback loops. As an example, Kusiak and Wang [12] focused on using the DSM to structure tasks and parameters in a design project.

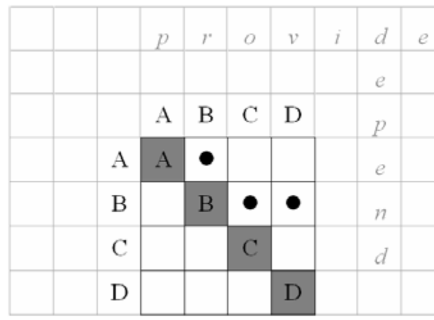

Figure 1. DSM Example

A DSM is usually constructed and used in three steps [13]:
1) Decompose the system into elements;
2) Document the physical interactions between elements;
3) Analyze potential clustering.

The model introduced by Pimmler and Eppinger [13] specifies design clustering. Their model considers four parameters: space, material, energy, and information. More recently a promising model was proposed by Yu, et al. [2] based on a Genetic Algorithm (GA) [14] that is able to identify the optimal number of modules in a given product. The Minimum Description Length (MDL) [15] used in their work has been shown to be effective in identifying modules as well as bus structures that interact with all modules.

However, these current approaches do not address the product family problem, i.e., how to create modules for not one product but for an entire family of products to enable sufficient variety for the marketplace. In fact, identification of modules requires analysis across multiple products in the family and consideration of common, variant, and unique components in addition to the basic interfaces (physical relationships) between components. As the products within the family represent variety, resulting modules must enable variety (e.g., taking some components, while another in the same family will not); hence, modules may be substantially different. The resulting interface becomes even more important as the goal is now to standardize interfaces as much as possible despite the presence of variant or unique modules in the family.

The proposed Variety Design Structure Matrix (DSM^V) and the 3D Design Structure Matrix (DSM^{3D}) address this need by studying modularity and interfaces across an entire product family. The DSM^V integrates variety in the product family in the DSM; hence, this DSM becomes a modularity tool for managing variety in the product family. The DSM^{3D} is still driven by the individual product because each product DSM is optimized before "stacking" them into three dimensions for the family. Hence, priority is given to each product DSM, and then modules and interfaces are studied across products. This extension enables designers to study module specifications and especially to identify variant modules and interfaces. Details for constructing and using the DSM^V and DSM^{3D} follow.

3. 3D DESIGN STRUCTURE MATRIX - DSM^{3D}

Constructing the DSM^{3D} is composed of three main steps:

1) Specify the DSM for each product in the family using generic components;
2) Analyze the interfaces across products to determine if they are common, variant, or unique and build the DSM^V; and
3) Stack all DSM^{V}s to form the DSM^{3D} and ultimately analyze cross-modules and cross-interfaces.

These three steps are described in the following sections.

3.1 Create Product DSMs Using Generic Components

The first step uses the basic DSM to specify modules in each individual product in the family, but at a higher-level of abstraction than traditionally used. In particular, generic variant component are defined to handle differences across products. For instance, each product in a family may have a battery, but the instantiation of that battery may be different in each product (e.g., AA, AAA). Thus, a generic component is defined for all products that have it, which in the previous example would simply be a "battery", and it is this generic component that is used when forming the basic DSM for each product.

So, let $\mathcal{F}$ be a family of products composed of P products: $\mathcal{F} = (P_1, \ldots, P_P)$. A product contains the set of components ensuring specific functionality, and every product performs one or more functions; so, each product P_i in the Family $\mathcal{F}$ is composed of specific components $P_i = \left\{C_i^1, \ldots, C_i^c\right\}$ where C_i^k denotes the instance of component k in product P_i. We call a *generic component* the invariant part of this component C_i^k. As a result, this first step will provide the basic DSMs with generic components so that cluster algorithms will not integrate variant components. In fact, clustering algorithms do not handle variety; so, product components (common, generic, and unique) are gathered in rows and columns, and then interfaces between these components must be documented. Finally, by switching corresponding rows and columns, clusters can be obtained. When the DSM is applied to the entire product family, the second step is performed – this is the real insight of the DSM^V and DSM^{3D} – addressing modularity and variety across all of the products in the family.

Figure 2 provides an example of this approach with the initial DSMs of component interfaces and the final DSMs clustered for each product. For simplicity this example consists of 4 products, each having 6 components, making the DSMs all the same size. As seen in the lower half of the figure, one group of components (C_1, C_2, and C_3) is identified as a potential module in Product 1. Using the same process, two modules are identified for Product 2 (C_1-C_2-C_3 and C_4-C_5) with two interfaces (C_2-C_6 and C_3-C_6); two modules appear in Product 3 (C_1-C_2-C_3 and C_4-C_5) with three interfaces (C_1-C_6, C_2-C_6, and C_3-C_6); and two modules (C_1-C_2-C_3 and C_4-C_5-C_6) and two interfaces (C_2-C_6 and C_3-C_6) are shown for Product 4. Note that components C_1 and C_2 form a bus that is connected to every other component.

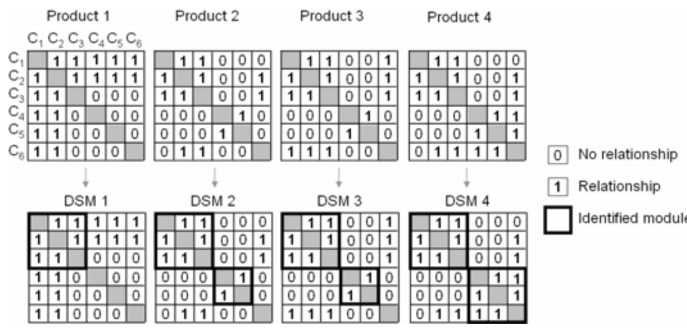

Figure 2. Resulting DSMs for a family of four products

3.2 Create Variety Design Structure Matrix - DSM^V

After clustering each DSM, the resulting modules and interfaces in each product need to be classified as common, variant or unique to formulate the DSM^V. In our model, variety is always considered to be the variety across products, and we use the terms cross-module and cross-interface defined as:

- A *common* cross-module (or cross-interface) is composed of similar components (interfaces for cross-interface).
- A *variant* cross-module is present in more than one product but is different among its instantiation. Two types of variant cross-modules can be identified:
    - Cross-modules having unique component(s): in this case the size of the module (matrix of the DSM) will differ across the product family; and
    - Cross-module implying variant components: in this case, the size of the cross-module is maintained across all products.
- A *variant* cross-interface is present in more than one product but is instantiated differently within the family.
- A *unique* cross-module (or cross-interface) is specific to a single product.

The cross-modules and cross-interfaces can be present in all or some of the products in the family. Because the literature is not clear on this point, we use "full" and "partial" to describe these two configurations. Thus, a "full cross-module" corresponds to a cross-module present in the entire family and can be common or variant, while a "partial cross-module" is present in only some of the products in the family, where it can also be either common or variant among those products. The resulting DSM^V is illustrated in Figure 3 for the four products in Figure 2. As the DSMs are symmetric, only the upper halves are illustrated to facilitate visualization.

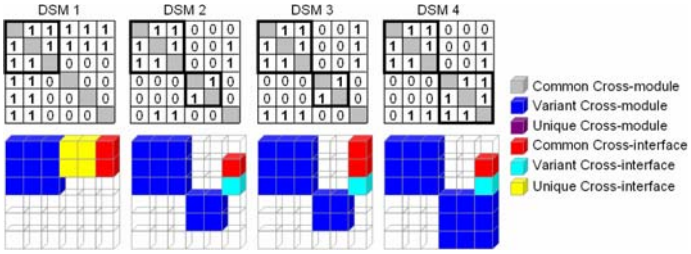

Figure 3. DSM^V representation for the four products

To model variety across products in the DSM^V, we use the color coding noted in Figure 3. Using previous definitions for our example, modules are defined as follows: the module (C_1, C_2, and C_3) is a variant cross-module across the family of products. This module is also a full cross-module being present in the entire product family. The second module composed of (C_4 and C_5) for Product 2 and 3 and (C_4, C_5, and C_6) for Product 4 is also a variant cross-module due to the different number of components involved in this cross-module (with a different size of matrix). This cross-module presents only in Product 2, 3, and 4 is a partial cross-module.

In the same vein, cross-interfaces (C_1-C_4, C_1-C_5, C_2-C_4, and C_2-C_5) are unique to Product 1. The interface (C_2-C_6) is a full cross-interface (present in all products of the family), while interfaces C_1-C_6 and C_3-C_6 are a partial cross-interface, as they are present in Products 1-3 and 2-3-4, respectively. The cross-interface C_1-C_6 is a common cross-interface (components are common across products and so are their interfaces). For the cross-interface C_3-C_6 since it implies the variant component 3, it will be necessary to check the nature of these cross-interfaces (common, variant, or unique) to build this DSM^V. Just to have a variant cross-interface in this example we consider that the component implies a variant interface too. As a result cross-interface C_3-C_6 is a variant cross-interface.

3.3 Create 3D Design Structure Matrix – DSM^{3D}

The resulting DSM^{V}s can now be "stacked" in 3D to get an overall picture of the product family and specifically to analyze modules across the product family. The third dimension in the DSM^{3D} represents the family, and the resulting DSM^{3D} for the family of 4 products is shown in Figure 4. As seen in the figure, the family has all possible configurations. This case (4 products with 6 components each) is simple, and such a DSM^{3D} can rapidly become complex to study as a whole when there are many products, each with many.

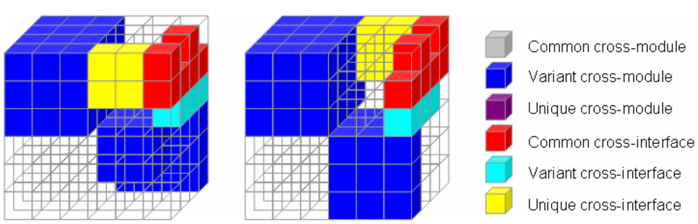

Figure 4. Two views of the DSM^{3D} for the product family

We can now study two aspects of the family dimension in the DSM^{3D} as illustrated in Figure 5:

- Cross-module analysis, to assess (and potentially improve) their integration within the family; and
- Cross-interface analysis, with the same goal but focusing instead on the interfaces.

In cross-module analysis, the goal is to study the identification of modules across the family and to answer the question of whether or not the variety is specified correctly. During this analysis we also check interfaces within each module, trying to make them common if possible. So, for our example, the variant cross-modules C_1-C_2-C_3 and C_4-C_5-C_6 need to be studied as shown in the figure. Concerning the cross-interface analysis, relationships between components and modules are assessed, i.e., everything "outside" the modules is considered. In this example, the partial cross-interface C_3-C_6 needs to be analyzed to determine whether this cross-interface can be changed to a common interface (see Figure 5). The goal of this analysis is to help designers manage interfaces across the family better by viewing this third dimension (the family).

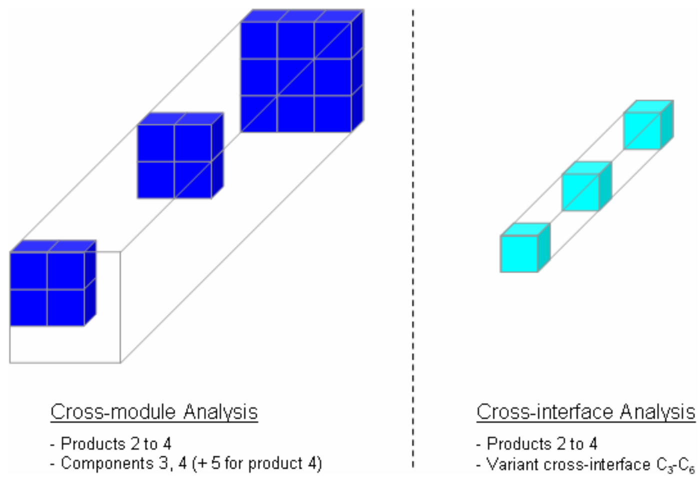

Figure 5. Cross-module and a cross-interface analyses

4. CASE STUDY

To demonstrate the potential of the DSM^V and DSM^{3D}, a case study involving three single-use cameras manufactured by Kodak (see Figure 6) is performed. Each step is applied to these three cameras to study their modular and interface specification.

Figure 6. Kodak® single-use camera (from left to right): Fun Saver, Plus Digital, Outdoor

4.1 Introduction to the Study

Each single-use camera was dissected to obtain the components and their interactions. Fun Saver, Plus Digital, and Outdoor models have 23, 26, and 21 components, respectively. As a result, the basic DSM of each single-use camera is built first (see Appendix A). As previously mentioned, each DSM is then clustered using Kusiak's on-line algorithm [17] to identify modules. Then, cross-analysis enables designers to identify common, variant, and unique modules and interfaces. The resulting DSM^V for the Fun Saver, Plus Digital, and Outdoor single-use cameras are presented in Figure 7, Figure 8, and Figure 9, respectively, after being translated back into an Excel spreadsheet for better viewing.

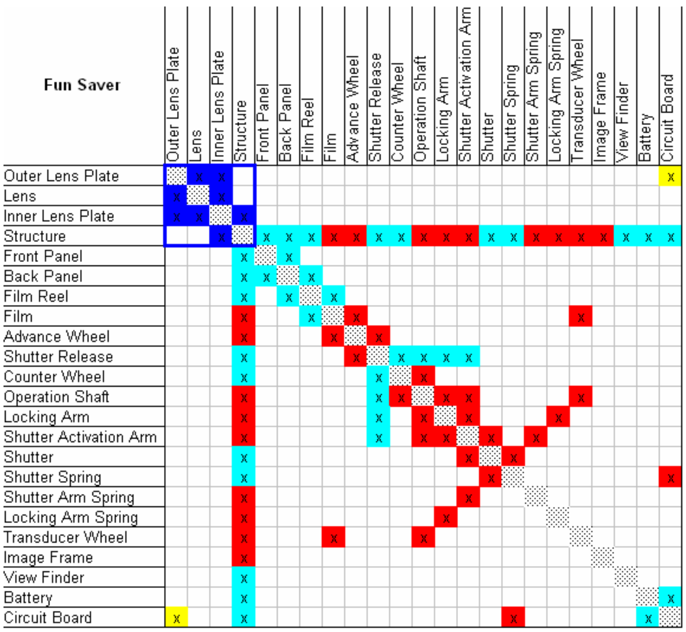

Figure 7. Fun Saver camera DSM^V after clustering

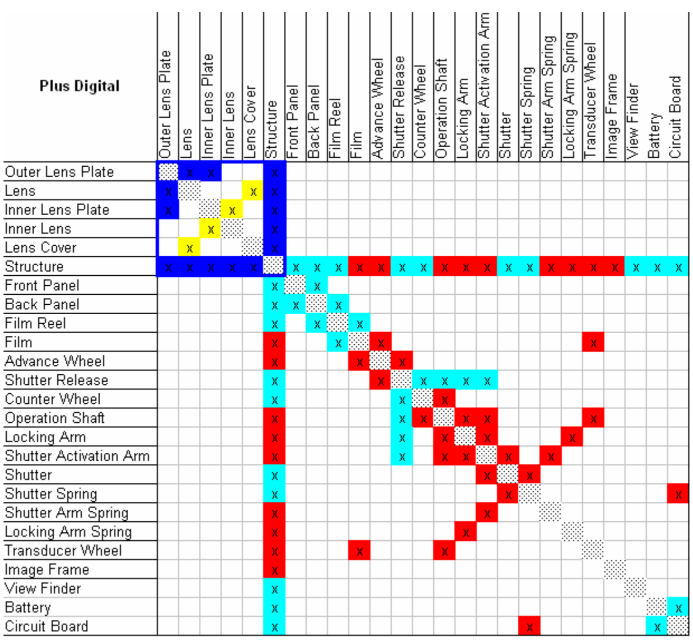

Figure 8. Plus Digital camera DSM^V after clustering

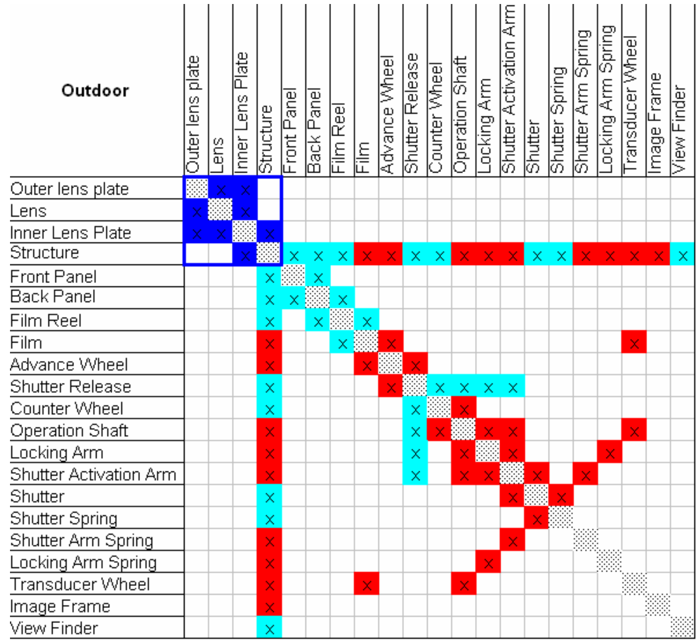

Figure 9. Outdoor camera DSM^V after clustering

These three single-use cameras present variant modules as well as unique, common, and variant interfaces. As mentioned before, the variety is first ensured via generic components like "Battery", which are differentiated as AA and AAA. The color coding in these figures is the same as before. One cross-module is identified for the Fun Saver, Plus Digital and Outdoor single-use cameras and is composed of the Outer Lens Plate, the Lens, the Inner Lens Plate, Inner Lens, Lens Cover, and the Structure. This module is a variant module and will be studied in more detail in Section 4.2. Some larger modules (involving 5 components) might be relevant for study like the one composed of "Shutter", "Counter wheel", "Operation shaft", "Locking arm", and "Shutter activation arm" but are specific to Kodak®. Interestingly enough this module was found to be implement by Fuji® as well. Other modules like Front panel and Back panel components cannot be modular for assembly reasons.

More than just the distribution of variant, unique, and common modules and interfaces (1 variant module; 43.5% variant, 55.5% common, and 0.1% unique interfaces), the DSM^V enables designers to study the architectural distribution of modules and interfaces. Thus, it is possible to study the overall architecture of the product integrating the nature of modules and interfaces. For instance, in this case study, the DSM^V highlights a variant cross-module, and designers can focus on this cross-module and elaborate a way to specify a common interface with the structure of the single-use cameras.

Figure 10 illustrates how the DSM^{V}s are "stacked" to obtain the DSM^{3D} – only the upper half is shown since it is symmetric. This DSM^{3D} provides visibility on the modules and interfaces. This visualization is performed using the ATSV [18], a powerful multi-dimensional visualization tool that has been developed separately from this work to support trade space exploration. Thus, designers can easily assess the variety in the product family; they can also identify interfaces that need particular attention. In this example, the variant module and the unique interfaces between the Circuit Board and Outer Lens Plate are identified. DSM^{3D} can also be used as a project review to check interface evolution converging toward definite design. This technique is used for digital mock-up review [19,20] and can be extended here to manage interfaces.

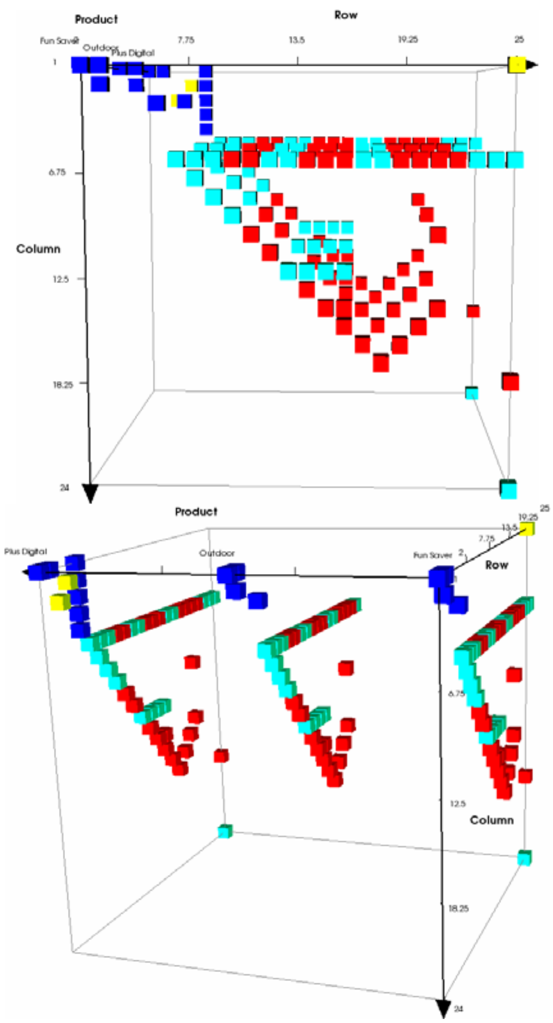

Figure 10. Two views of the camera family DSM^{3D}

4.2 Cross-Module and Cross-Interface Analyses

The next step is to study this DSM^{3D} in detail using cross-module and cross-interface analyses. In this example, we consider the full variant cross-module (Outer Lens Plate, the Lens, the Inner Lens Plate, Inner Lens, Lens Cover, and the Structure), to study the module and its interfaces, which are more relevant to the architecture of the product.

4.2.1 Cross-Module Analysis

The Lens/Structure module (Outer Lens Plate, the Lens, the Inner Lens Plate, Inner Lens, Lens Cover, and the Structure) is studied by analyzing DSM^{V}s. These DSM^{V}s are represented in Figure 11. This figure shows variant interfaces for the Outdoor and the Fun Saver models, with the same generic component (variant in detail). The Plus Digital has unique components (Inner Lens and Lens Cover); hence, their interface is unique. The rest of the interfaces are similar expect that components are more related to the structure. The DSM^{3D} (upper matrix) is illustrated separately in Figure 12.

Outdoor	Outer lens plate	Lens	Inner Lens Plate	Structure	
Outer lens plate		x	x		1
Lens	x		x		2
Inner Lens Plate	x	x		x	3
Structure			x		4

Fun Saver	Outer Lens Plate	Lens	Inner Lens Plate	Structure
Outer Lens Plate		x	x	
Lens	x		x	
Inner Lens Plate	x	x		x
Structure			x	

Plus Digital	Outer Lens Plate	Lens	Inner Lens Plate	Inner Lens	Lens Cover	Structure
Outer Lens Plate		x	x			x
Lens	x				x	x
Inner Lens Plate	x			x		x
Inner Lens			x			x
Lens Cover		x				x
Structure	x	x	x	x	x	

Figure 11. DSM^{V}s of Lens/Structure variant cross-modules

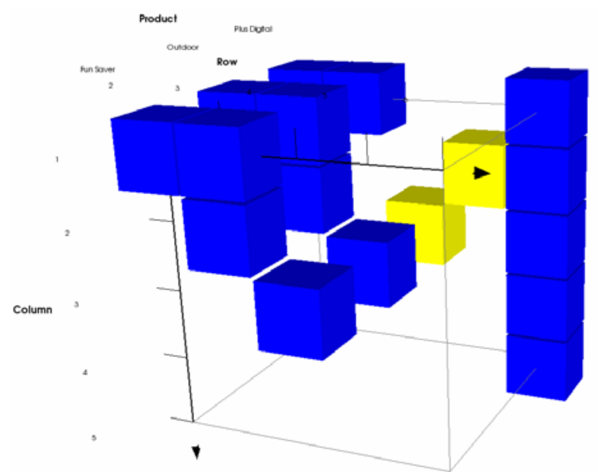

Figure 12. DSM^{3D} of variant Lens/Structure cross-module

By analyzing the nature of the interfaces it is possible to check the necessity for variety. Existing functional analysis of the necessity can be combined to this DSM^{3D} [21]. For instance, we can see that the Outdoor and Fun Saver cameras have another sub-module composed of the Outer Lens Plate, Lens, and Inner Lens Plate, which is not present in the Digital Plus. These components do not form a module for the Plus Digital; however, the Plus Digital Lens is the same, and the Outer Lens Plate and Inner Lens Plate can be made common to obtain a common sub-cross-module. In this case, components and their interfaces become common. The variety of Outer Lens Plate, Lens, and Inner Plate was the reason of the high relationship between these components and the structure (they were all related to the Structure). By defining a common module the interfaces are simplified and become common. The resulting DSM^{V}s are presented in Figure 13.

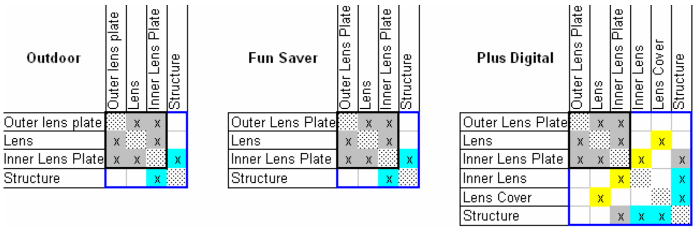

Figure 13. DSM^{V}s of Lens/Structure variant cross-modules after improvement

The DSM^{3D} modeling offers an added advantage by helping check variant interfaces and their related components. It is possible to study at the same time if variants can be achieved through scaled-based design [22] or merged-based design [23] and how their interfaces can be commonalized.

4.2.2 Cross-Interface Analysis

A cross-interfaces analysis enables designers to target a common interface even if components are variant. This can be managed using a merge based-design approach [23]. The first goal is to check all cross-interfaces for the entire product family. This analysis enables designers to determine why some interfaces are variant and to justify this variety. In other words, we check to see if the cross-interface variety is required for any particular reason (architecture, maintenance, etc.), and if not, then designers can converge on a more common interface.

We consider the variant cross-interface Shutter/Structure in our case study. If the actual component specification, the variety is warranted for architectural reasons: the shutter also activates the Flash, and the Outdoor camera does not have flash while architecture of both the Plus Digital and Fun Saver are different (due to the size/layout for the battery). So, this variety can be justified; however, an earlier consideration (in the project design) of this cross-interface may have lead to a common Shutter, a common architecture and a common interface with the Structure. Even if the constraints seem to be justified, it is relevant to study potential improvements. For instance, in this case study, variant cross-interfaces like Front Panel/Structure and Back Panel/Structure seem to be very strong and justified. A more in-depth study [23] identifies that these cross-interfaces can be actually made common. Hence, it is possible for designers to easily identify every variant interface and their relationship with other components to increase commonality in the product family and reduce production costs.

5. CLOSING REMARKS

The Variety Design Structure Matrix, DSM^V, and the 3D Design Structure Matrix, DSM^{3D}, have been introduced in this paper to extend the basic DSM to include variety within a product family. These two new matrices are combined into a single approach to better manage both modularity and variety. The DSM^V models common, variant and unique modules and interface across products and enables one to study in detail modules and interfaces across products. The DSM^{3D} permits a higher level of analysis for the entire family of products and captures a third dimension, i.e., the product family, to study cross-modules and cross-interfaces.

The case study investigates the implementation of the proposed approach using three single-use cameras from Kodak. The complete process has been described, and the different levels of analysis (entire product family or cross analyses) have been presented. This approach works on an existing product family as well as a new product family, and it can also bring new insight to designers during the project review process. Future work will continue to integrate a more realistic model, examining larger product families, and more complex products.

ACKNOWLEDGEMENT

We acknowledge the National Science Foundation through Grant Nos. IIS-0325321 and IIS-0325402. Any opinions, findings and conclusions or recommendations presented in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Steward, A. D., 1981, "The Design Structure System: A Method for Managing the Design of Complex Systems," *IEEE Transaction on Software Engineering*, **28**(3), pp. 71-74.
[2] Yu, T.-L., Yassine, A. and Goldberg, D. E., 2003, "A Genetic Algorithm for Developing Modular Product Architectures," *ASME International Design Engineering Technical Conferences*, ASME, Paper No. DETC2003/DTM-48657.
[3] Ulrich, K. T. and Eppinger, S. D., 2000, *Product Design and Development*, 2nd, McGraw-Hill, New York.
[4] Baldwin, C. Y. and Clark, K. B., 2000, *Design Rules: The Power of Modularity*, MIT Press, Cambridge, MA.
[5] Sanchez, R. and Mahoney, J. T., 1997, "Modularity, Flexibility, and Knowledge Management in Product and Organization Design," *IEEE Engineering Management*, **1**(1), pp. 50-61.
[6] Hirschi, N. W. and Frey, D. D., 2002, "Cognition and Complexity: An Experimentation on the Effect of Coupling in Parameter Design," *Research in Engineering Design*, **13**(3), pp. 123-131.
[7] Gershenson, J. K., Prasad, G. J. and Zhang, Y., 2003, "Product modularity: Definitions and Benefits," *Journal of Engineering Design*, **14**(3), pp. 295-313.
[8] Gershenson, J. K., Prasad, G. J. and Zhang, Y., 2004, "Product Modularity: Measures and Design Methods," *Journal of Engineering Design*, **15**(1), pp. 33-51.
[9] Stone, R. B., Wood, K. L. and Crawford, R. H., 2000, "A Heuristic Method for Identifying Modules for Product Architectures," *Design Studies*, **21**(1), pp. 5-31.
[10] Otto, K. and Wood, K. L., 2001, *Product Design - Techniques in Reverse Engineering and New Product Development*, Prentice Hall, Upper Saddle River, NJ.
[11] Browning, T. R., 2001, "Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and new Directions," *IEEE Transactions on Engineering Management*, **48**(3), pp. 292-306.
[12] Kusiak, A. and Wang, J., 1993, "Decomposition of the Design Process," *ASME Journal of Mechanical Design*, **115**(4), pp. 687-695.
[13] Pimmler, T. U. and Eppinger, S. D., 1994, "Integration Analysis of Product Decompositions," *ASME Design Engineering Technical Conferences - Design Theory and Methodology Conference*, Minneapolis, MN, **68**, pp. 343-351.
[14] Goldberg, D. E., 1989, *Genetic Algorithm in Search, Optimization and Machine Learning*, Addison-Wesley Publishing, Reading, PA.
[15] Rissinen, T., 1978, "Modeling by Shortest Data Description," *Automatica*, **14**, pp. 465-471.

[16] Kusiak, A., Larson, N. and Wang, J., 1994, "Reengineering of Design and Manufacturing Processes," *Computers and Industrial Engineering*, **26**(3), pp. 521-536.

[17] Kusiak, A., 2005, "Triangularization Algorithm," available online at: http://www.icaen.uiowa.edu/%7Eankusiak/algorithm.html.

[18] Stump, G. M., Yukish, M., Simpson, T. W. and Harris, E. N., 2003, 2003, "Design Space Visualization and its Application to a Design by Shopping Paradigm," *ASME Design Engineering Technical Conferences - Design Automation Conference*, Chicago, IL, ASME, Paper No. DETC2003/DAC-48785.

[19] Dassault Systems, 2006, "CATIA R17: Boots Innovation for Products Excellence," URL: http://www.3ds.com/products-solutions/plm-solutions/catia/overview/.

[20] PTC, 2006, "Pro/ENGINEER: The Industry Standard for 3D CAD," http://www.ptc.com/appserver/mkt/products/home.jsp?k=403.

[21] Alizon, F., Shooter, S. B. and Simpson, T. W., 2006, "Assessing and Improving Commonality and Diversity within a Product Family," *International Design Engineering Technical Conferences*, Philadelphia, PA, DETC2006/DAC-99499.

[22] Simpson, T. W., Siddique, Z. and Jiao, J., 2005, *Product Platform and Product Family Design: Methods and Applications*, Springer, New York, NY.

[23] Alizon, F., Shooter, S. B. and Simpson, T. W., 2007, " Merge-based Design: A Method Targeting a more Manageable Variety and an In-depth Mass Customization," *International Design Engineering Technical Conferences*, Las Vegas, NV, :Paper No. DETC2007/DAC-34541.

APPENDIX A: BASIC DSMs FOR THRE THREE SINGLE-USE CAMERAS

Fun Saver (2006)	Front Panel	Back Panel	Outer Lens Plate	Lens	Inner Lens Plate	Film	Film Reel	Image Frame	Shutter Spring	Shutter	Battery-AA	Circuit Board	View Finder	Shutter Release	Counter Wheel	Advance Wheel	Locking Arm	Locking Arm Spring	Shutter Activation Arm	Shutter Arm Spring	Operation Shaft	Transducer Wheel	Structure
Front Panel																							x
Back Panel							x																x
Outer Lens Plate				x	x																		x
Lens			x		x																		
Inner Lens Plate			x	x								x											x
Film							x															x	x
Film Reel		x				x										x							x
Image Frame																							x
Shutter Spring										x		x											x
Shutter									x										x				x
Battery-AA												x											x
Circuit Board					x				x		x												x
View Finder																							x
Shutter Release															x	x	x		x		x		x
Counter Wheel														x							x		x
Advance Wheel							x							x									x
Locking Arm														x				x	x		x		x
Locking Arm Spring																	x						x
Shutter Activation Arm										x				x			x			x	x		x
Shutter Arm Spring																			x				x
Operation Shaft														x	x		x		x			x	x
Transducer Wheel						x															x		x
Structure	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	

Outdoor (2006)	Back Panel	outer lens plate	Lens	Inner Lens Plate	Structure	Film	Film Reel	Image Frame	Shutter Spring	Shutter	View Finder	Shutter Release	Counter Wheel	Advance Wheel	Locking Arm	Shutter Activation Arm	Operation Shaft	Transducer Wheel	Locking Arm Spring	Shutter Arm Spring	Front Panel
Back Panel					x		x														
outer lens plate			x	x																	
Lens		x		x																	
Inner Lens Plate		x	x		x																
Structure	x			x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Film					x		x											x			
Film Reel	x				x	x								x							
Image Frame					x																
Shutter Spring					x					x											
Shutter					x				x							x					
View Finder					x																
Shutter Release					x								x	x	x	x	x				
Counter Wheel					x							x					x				
Advance Wheel					x		x					x									
Locking Arm					x							x				x	x		x		
Shutter Activation Arm					x					x		x			x		x			x	
Operation Shaft					x							x	x		x	x		x			
Transducer Wheel					x	x											x				
Locking Arm Spring					x										x						
Shutter Arm Spring					x											x					
Front Panel					x																

Plus Digital		Front Plate	Front Panel	Back Panel	Outer Lens Plate	Lens	Inner Lens Plate	Shutter Spring	Shutter	Lens Cover	Inner Lens	Film	Film Reel	Image Frame	Battery	Circuit Board	Shutter Release	View Finder	Counter Wheel	Advance Wheel	Locking Arm	Shutter Activation Arm	Operation Shaft	Transducer Wheel	Shutter Arm Spring	Locking Arm Spring	Structure
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Front Plate	1		x																								
Front Panel	2	x		x																							x
Back Panel	3		x										x														
Outer Lens Plate	4					x	x																				x
Lens	5				x					x																	x
Inner Lens Plate	6				x						x																x
Shutter Spring	7								x							x											
Shutter	8							x														x					x
Lens Cover	9					x																					x
Inner Lens	10						x																				x
Film	11												x							x				x			x
Film Reel	12			x								x															x
Image Frame	13																										x
Battery	14															x											x
Circuit Board	15							x							x												x
Shutter Release	16																		x	x	x	x	x				x
View Finder	17																										x
Counter Wheel	18																x						x				x
Advance Wheel	19											x					x										x
Locking Arm	20																x					x	x			x	x
Shutter Activation Arm	21								x								x				x		x		x		x
Operation Shaft	22																x		x		x	x		x			x
Transducer Wheel	23											x											x				x
Shutter Arm Spring	24																					x					x
Locking Arm Spring	25																				x						x
Structure	26		x		x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	