Analyzing the Relationships in Product Architecture using DSM, DMM, and MDM

MASTER OF TECHNOLOGY
DISSERTATION REPORT

BY

PRITAM KUMAR CHANDEL

(2015PIE5243)

DEPARTMENT OF MECHANICAL ENGINEERING
MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR

JUNE 2017

A

DISSERTATION REPORT

ON

Analyzing the relationships in Product Architecture using DSM, DMM, and MDM

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

MASTER OF TECHNOLOGY
IN
INDUSTRIAL ENGINEERING

BY

PRITAM KUMAR CHANDEL
(2015PIE5243)

UNDER THE GUIDANCE OF
Prof. (Dr.) A.P.S. RATHORE

DEPARTMENT OF MECHANICAL ENGINEERING
MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR

JUNE 2017

MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR
JAIPUR – 302017 (RAJASTHAN), INDIA

CERTIFICATE

This is to certify that the dissertation entitled **“Analysing The Relationships In Product Architecture Using DSM, DMM And MDM”** being submitted by **Pritam Kumar Chandel (2015PIE5243)** is a bonafide work carried out by him under my supervision and guidance, and hence approved for submission to the **Department of Mechanical Engineering, Malaviya National Institute of Technology Jaipur** in partial fulfillment of the requirements for the award of the degree of **Master of Technology (M.Tech.)** in **Industrial Engineering**. The matter embodied in this dissertation report has not been submitted anywhere else for the award of any other degree or diploma.

Prof. A.P.S. Rathore
Professor,
Department of Mechanical Engineering,
MNIT Jaipur
Place: Jaipur
Dated: 30th June 2017

MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR
JAIPUR – 302017 (RAJASTHAN), INDIA

CANDIDATE'S DECLARATION

I hereby declare that the work which is being presented in this dissertation entitled **"Analysing The Relationships In Product Architecture Using DSM, DMM And MDM"** in partial fulfilment of the requirements for the award of the degree of **Master of Technology (M.Tech.)** in **Industrial Engineering**, and submitted to the **Department of Mechanical Engineering, Malaviya National Institute of Technology Jaipur** is an authentic record of my own work carried out by me during a period of one year from July 2015 to June 2017 under the guidance and supervision of **Prof.** (**Dr.) A.P.S. Rathore** of the Department of Mechanical Engineering, Malaviya National Institute of Technology Jaipur.

The matter presented in this dissertation embodies the results of my own work and has not been submitted anywhere else for the award of any other degree or diploma.

Pritam Kumar Chandel
(2015PIE5243)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Place: Jaipur **Prof. A.P.S. Rathore**
Dated: 30 June, 2017 Supervisor

ACKNOWLEDGEMENTS

With great delight, I acknowledge my indebted thanks to my guide and mentor **Prof. Ajay Pal Singh Rathore** who has always been a source of inspiration and encouragement for me. His stimulated guidance and unwavering support always motivated me to reach out for, and achieve higher levels of excellence. This dissertation could not have attained its present form, both in content and presentation without his active interest, direction, and help. I am grateful to him for keeping trust in me in all circumstances. I thank him for being big-hearted with any amateurish mistakes of mine.

I express my sincere gratitude to **Prof. Rakesh Jain, Prof. Gopal Agrawal, Prof. G.S. Dangayach, Prof. Awadhesh Bhardwaj, Dr. M.L. Mittal** and **Dr. M.L. Meena** for their support and guidance throughout the course of study at MNIT Jaipur.

I am extremely thankful to **Dr. Gunjan Soni** for providing me with his valuable time and inputs since the inception of my studies at MNIT Jaipur. I extend my deep sense of gratitude to him for his cooperation and sharing his immense pool of knowledge.

I highly acknowledge and duly appreciate the support extended by Sandeep Kumar Sisodia, Manager at Hero Cycles, for assistance with the forming DSMs and DMMs for my research work.

I want to express my thanks to my senior research scholars at MNIT Jaipur**, e**specially **Mr. Avanish Singh (Ph.D. scholar)** for their support and guidance at various stages of the project. I extend my heartiest thanks to my colleagues-friends for their help & support in the accomplishment of this work during my stay at MNIT Jaipur.

Final thanks go to almighty God, my parents **Sh. Devi Lal Khatik** and **Smt. Sita Devi** for their continuous support, encouragement, and blessings.

- Pritam Kumar Chandel

ABSTRACT

Product development (PD) is amongst the most critical and complex process for any organization in this globalized world. It requires innovation and advancement which requires complex learning (feedback) loops. Increasing worldwide rivalry among competitors forces the firms to invest in PD for developing new products at an inexorably quicker pace. This directive places a generous weight on engineering groups to *develop better products* and at the same moment *develop products faster*. The success of any new product development (NPD) projects is the need of the hour for the survival of any organization in the global market.

Every organization tries to fulfill maximum needs of the target customers to increase their market share. This puts enormous pressure on PD team to strike balance between product risks and modularity in a product architecture based on the diversity of consumer needs. Designing a product is an iterative process because most of the elements are interlinked with some other elements. Changes in one element force the product developers to make changes in another element. This complexity requires a tool to capture and show the PD team all conceivable relationships among the various components of a product or process beforehand so that it becomes easy to plan the PD stages with fewer iterations.

The most acknowledged tool in this regard is the Design Structure Matrix (DSM). It is an exceptionally supportive instrument for representing and breaking down the design of an individual framework, for example, a product, a process, or an organization. By capturing the relationships among the various elements of the same domain, DSM of any development process provides valuable awakening. Primarily, it clearly uncovers which information exchanges will cause design iterations and which don't.

This study has discussed and clearly explained the construction, reading and working with DSM and other modified tools based on DSM’s philosophy like Domain Mapping Matrix (DMM), Multi-Domain Matrix (MDM) and Connectivity Maps. It also explained one of the widely used tools known as Quality Function Deployment (QFD) or House of Quality. Just like many other tools, these tools have been implemented in diverse areas outside its original space, as scholars and researchers have tried to use

their benefits. This research focuses on PD and audits the major work done in implementation and utilization of DSM and other tools in PD to manage complex relationships.

This study has distinctively discussed that all the current methods have some sort of drawbacks in capturing complex relationships that are critical to the PD and that there is strong need to improve the existing methods or develop a new method.

The *connective MDM* developed in this research is based on the concept of connectivity maps and MDM. It shows, with an example, the benefits accrued from it as it gives a clear picture of the cause of a relationship helping in the identification of key relationships or emerging factors and the level of relationships. At the end of this research, it further provides the directions for future research dedicated to augmenting PD process' effectiveness. The implications of this research will help companies to easily identify the critical relationship combinations of different elements of a product or activities while designing product architecture, which requires more focus for the success of any project.

Keywords: Product Development, Product Architecture, Design Structure Matrix, Domain Mapping Matrix, Multi-Domain Matrix, Connectivity Map.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

DMM	Domain Mapping Matrix
DSM	Design Structure Matrix
GPD	Global Product Development
MDM	Multi-Domain Matrix
NPD	New Product Development
PD	Product Development
PA	Product Architecture
QFD	Quality Function Deployment

Chapter 1

INTRODUCTION

Product Development (PD) processes have an extremely significant role in achieving aggressive competitive edge and success of any product or services. PD is significant in light of the fact that any new product is specifically connected with competition for many organizations' success. (Clark and Fujimoto, 1991; Jensen and Harmsen, 2001). New product development (NPD) is seen as a champion among the most critical scope of company's ability identified with business accomplishment (Guo, 2008). Any reasonable person would agree that if any firm is losing its focused position, it can be credited to products or services that are not aggressive (Yadav et.al., 2007).

With increasing development and income, market has begun to felt the requirements of new products every now and then to satisfy their needs. Thus there is a lot of space for companies to develop and improve their products or services to deliver what is actually needed by the customers to occupy a large market share and increase their revenue.

For this companies needs to develop a new product or improve the existing product. Engineering and building complex products pose both specialized technical and administrative difficulties. With adding the geographical, cultural, and age diversity it becomes difficult to contain the needs of all the customers which may vary due to such diversity. Thus to fulfill these needs and to save the cost of developing a very new product, companies prefer modularity to enable the postponement of the changes. Modularity increases the number of components and this further increases the complexity as it becomes difficult for the PD team to capture and visualize the relationships between different components of a product.

The blend of the distinctive components (or subsystems) into the conclusive product (or structures) requires the identification and comprehension of the relationships among the diverse components. To address such technical complexities PD teams generally aim towards decomposing the development procedure thoroughly into smaller and easier components (i.e. activities, components, and subsystems) which are dealt with by different PD groups. Decomposition is very useful in containing the specialized, technical, complex and multi-sided design and helps to address administrative complexity reducing the number of iterations.

A few research scholars have proposed matrix-based models for system modeling and study aimed at capturing the relationships, dependencies or linkages among various elements of the products or activities. (Steward 1962, 1981, Warfield 1973, Kehat and Shacham 1973). In general, matrix based methods are preferred for capturing the relationships because they are simple and compact approach that has proved successful in representing a system and capturing the relationship between different system elements. (Steward et. all, 1962)

A matrix model was introduced by Steward (1962, 1981) who called it 'design structure matrix' (DSM) and elaborated few algorithms for manipulating the matrix as a tool for systems design. Thereafter, DSM usage has led to the development of *domain mapping matrices* (DMMs) and *multidomain matrices* (MDMs). DMM is a rectangular two-dimensional matrix tool used to represent and analyze dependencies and relationships between two different domains. Using DSM and DMM a new matrix came into a picture called Multi-Domain Matrix (MDM) which captures the relationships between multiple domains. MDM have broadened the capabilities and applications of matrix-based models in complex systems and provided further insights. Such capabilities have become increasingly beneficial and important in this age of ever-increasing complex projects, products, processes, organizations, and other systems. Finally, the quality function deployment (QFD) method given by Hauser and Clausing (1988) is another matrix-based model for systems design and analysis in an engineering context. QFD is an example of a relationship map to make an interpretation of customer needs into design and engineering requirements through a matrix known as the 'house of quality'. Using symbols and numerical rankings, the matrix compactly yet exhaustively speaks to the connections between customer needs, specifications, competitive products, and engineering metrics.

The methods mentioned above lacks in one or another way. Most importantly, they fail in capturing the logic or cause of the dependencies or relationships. To overcome this a new method was developed by Yassine et.all, 2003 called Connectivity Map (C-Map), a more complex matrix-based tool, combining two relationship maps into a single matrix. Still cannot cater the needs when a project have to deal with more than 3 domains.

Therefore, the purpose of this research is to explore existing matrix based methods capturing the relationships in PD process and to develop a structured model that can easily capture various relations among multiple factors or components of PD and capable of finding out the root or cause of that relationship through another factor in a multi-domain environment.

1.1 Motivation for Research:

Relationships & connections can exist among the components of the same domain and between the components of different domains. Thus making the product and its development process very complex and capturing connections between subsidiary components vital to take care of product architecture if the product has to be robust and fruitful. This multi-faceted nature makes the essential connections between the elements difficult to visualize and communicate. The main source of difficulty is a lack of models capable of capturing all of these elements and their dependencies in a single, but simple, way (Yassine, Ali, et al., 2003). All the mentioned matrix-based tools provide somewhat insight into the dependency structure of a complex system or process (i.e. a dependency map). But only C-Map captures capture the logic behind these dependencies. That is if a component of domain A is shown to relate to another component of domain B, it is not clear why and how this relationship exists and how can this relationship can affect the components of domain C. It is highly beneficial to know that element of A is related to element of B through some components of C to reduce the number of iterations.
However, C-Maps still remains a two-dimensional matrix capturing relationships among two domains via third domain. But even for only three domains one may have to refer three different C-Maps to visualize different relations for different reasons. The number of C-Maps needed to refer increases with increase in number of domains. Therefore, there is a scope to develop a comprehensible method having single matrix capable of capturing all possible interactions like MDM and the mutual influences they have on each other.

1.2 Structure of the Dissertation:

This dissertation report is organized into six chapters as shown in Figure 1.1.

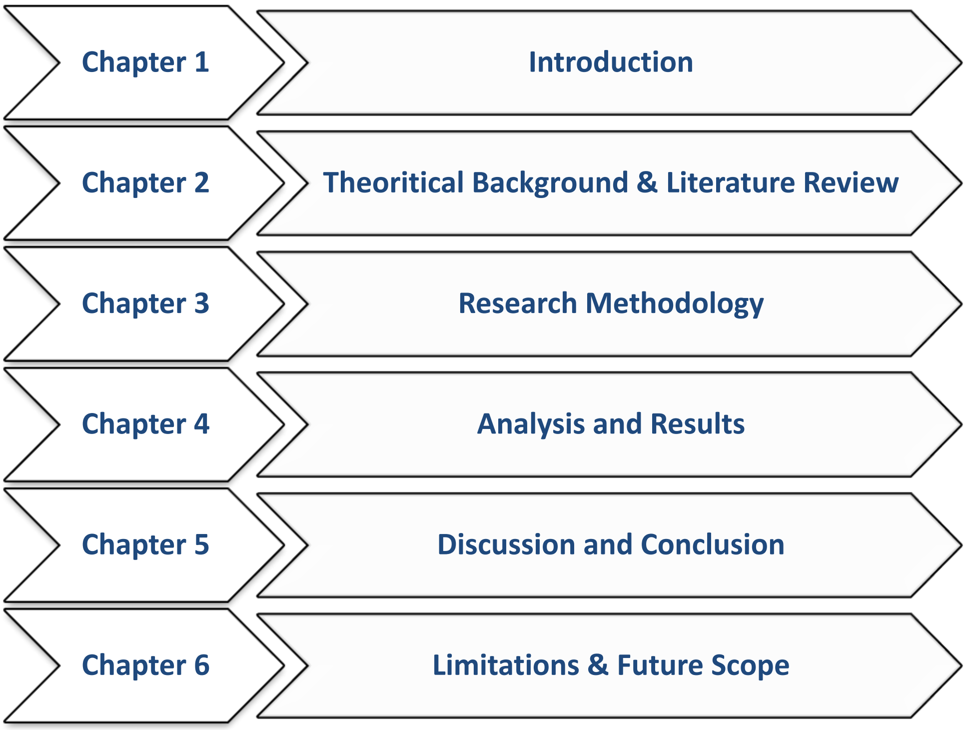


Figure 1.1 – Outline of dissertation

Chapter 1: This chapter discusses the topic of the study, its motivation, and need of study. It outlays the objectives of the research. Finally, the layout and content of the chapters are described.

Chapter 2 provides theory and literature review of PD, PA and currently used methods for capturing relationships in PD like DSM, DMM, MDM, and C-Maps. The gathered in-depth information is summarized in a tabular form clearly differentiating between DSM, DMM, and MDM. It also explains the working of each method.

Chapter 3 consists of the description of the research methodology. Each step of research is clearly demarcated and explained here maintaining the actual flow of research giving the clear insight of what is done and how.

In *Chapter 4*, the results of the current method are compared with the new method. The analysis of interactions among the various factors and key attributes emerging and the cause of their emergence using the new method is discussed. The data analysis and results of the whole study are documented in this chapter.

Chapter 5 discusses the consequences of the study and reports the conclusions drawn from the study. Normative advice is provided for the PD team of any industry to use this new method.

Chapter 6 lists the limitations of this research study, and suggestions are made to show the path for future research scope.

Chapter 2

THEORETICAL BACKGROUND & LITERATURE REVIEW

Literature Review is of huge significance to figure out response to any research and exploration question to accumulate in-depth information on the subject so that a superior comprehension of the issue can be accomplished. The literature review is the basis for a study and contributes to the formulation of an answer to the proposed research question (Cooper and Schindler, 2008).

There are various matrix-based methods for system modeling and analysis in PD helping in finding relevant linkages among intra-domain and inter-domains. In this chapter, the literature review portion is organized into four sections, each providing a theoretical background and addressing the research and literature available on the following four areas- DSM, DMM, MDM, and C-Maps.

2.1. Product Development

PD is an arrangement of steps that incorporates the conceptualization, design, development and promoting of recently made products or services. The target of PD is to prepare, maintain and raise an organization's overall market share by fulfilling a buyer's request. Eppinger defined PD as the steps of activities beginning with the perception of a market opportunity and ending in the production, sale and delivery of a product (Ulrich and Eppinger, 2000).

Adaptability of the PD procedure alludes to the capacity to design reliable product in the light of an evolving situation. However, the adaptability and adequacy of PD processes rely on various other variables, for example, design exercises and tools utilized, their planning and scheduling, basic leadership approaches, information flow structure, and quality and accessibility of information.

2.1.1. Product development process:

According to Karl. T. Ulrich and Steven D. Eppinger, PDP can be divided into six phases as follow (Ulrich et.al., 2009):

Phase 0: Planning is frequently alluded as 'phase zero' as it precedes the project approval and launch of the genuine PD process. This phase starts with corporate strategy and incorporates evaluation of technology development and market goals. The output of this stage is the 'Mission Statement', which indicates the target market, business objectives, constraints, and key assumptions.

Phase 1: Concept development phase consists of generation and evaluation of alternative product concepts addressing the needs of the target market. A concept is a description of form, function, and features of a product accompanied by a set of specifications, and competitive and financial analysis.

Phase 2: System-level design phase includes the definition of the product architecture and the decomposition of product into sub-systems and components. The final assembly scheme for the production system is usually defined in this phase.

Phase 3: Detail design phase includes the complete specifications of the shape, size, geometry, and tolerances of all the unique parts of the product and the identification of standard parts to be purchased from suppliers. The critical issues of production cost and robust performance are addressed in this phase.

Phase 4: Testing and refinement phase involves the construction and testing of multiple prototypes of the product. The prototypes are tested and refinements are done to make better quality product best suited for consumer needs.

Phase 5: Production ramp-up phase is the final phase of product manufacturing where the actual product is made using the intended production system. The purpose of the ramp-up is to train the workforce and to work out any remaining problems in the production process.

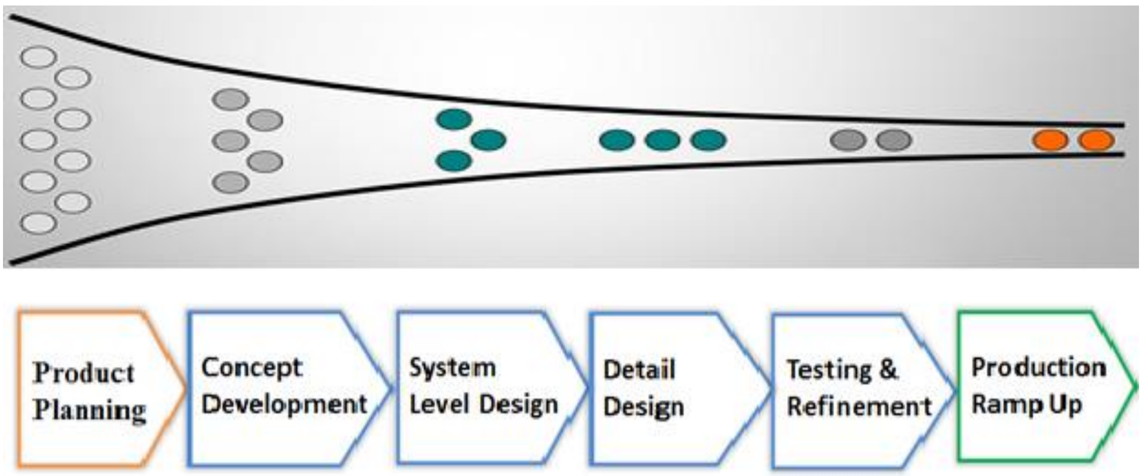


Figure 2.1. Phases of Product Design (Adapted from Ulrich et.al., 2009)

2.2. Product Architecture

Product architecture (PA) is the scheme by which the functional elements of the product are arranged into physical chunks and by which the chunks interact (Eppinger et.al., 1994). In phase 2 i.e. System level design PA comes into the picture. PA has very profound implications for how the product is planned, designed and made, even how they are sold, utilised and repaired, and so forth. The influence of PA is felt during assembly.

Strictly speaking, there are two types of PA exists:

a) Modular: In this, each functional element or function is implemented by exactly one single physical chunk

b) Integrated: In this a single chunk implements many functions i.e. one chunk implements more than one function.

2.2.1. Considerations at product architecting (Ulrich et.al., 2009):

- By what means will it influence the capacity to offer product variety?
- By what means will it influence the product cost?
- By what means will it influence the design lead time?
- By what means will it influence the development process management?

2.2.2. Influences of Product Architecture:

- **During Product Development**
  - How families and platforms are structured?
  - How functions are realized?
  - How reuse and standardization are accomplished?
  - How development work is divided up?
  - Where subassembly and module boundaries are?
  - Where DFCs go?
- **During Production System Design**
  - Assembly sequences
  - Reuse of facilities and knowledge
  - Planning for flexibility
  - Sharing of facilities to match capacity to demand

- **During manufacturing and assembly**
  - Where production happens
  - How customer orders are fulfilled
  - How unpredictable demand patterns are met
- **During Use**
  - How service is delivered
  - How the product is updated
  - How the product is recycled

Table 2.1 - Difference Between Modular and Integral
*Adapted from (*MacDuffie, 2000)

Modular	**Integral**
Chunks are independent of each other functionally and physically	Chunks are inter-dependent among each other
Standard, pre-designed interfaces can be used that can remain the same even if internal characteristics change	Interfaces are tailored to the chunks and are dependent on functional behaviour
Modules can be specialized to their individual contributions to overall function and can be used interchangeably	Chunks are tailored to their application and cannot be interchanged without requiring changes to other chunks
Standard interfaces are physically separate from the module and thus waste other design resources such as space or weight; interfaces are "weak"	Interfaces can be integral to the chunk, saving space or weight; interfaces are "strong"
Interface management, if planned properly, can provide flexibility during production	Interface management occurs entirely during design and is frozen; it is not aimed at flexibility after design
Business performance may be favoured	Technical performance may be favoured

2.2.3. Factors affecting architecture modularity or degree of Modularity (Ulrich et.al., 2009):

- Product changes

- Product variety
- Component standardization
- Product performance
- Manufacturability
- Product development management

Seeing at these factors, in present world, most of the product are neither strictly 'modular' nor strictly 'integral'. Companies are choosing actually mixture of these two strategies according to their capability and need. Hence degree of modularity varies, from product to product and company to company. More the degree of modularity more difficult it is to manufacture and integrate that product. So as the modularity increases complexity in product development increases and no. of tasks increases per product. And as the task increases the confusion arises as when and which task to be done as many tasks are interrelated to each other and may need some information from one or more tasks. More importantly, their order has to be maintained for each product to reduce variation and error. So the need for the tool arises here which can tell us when to do a task.

Hence DSM comes into the picture to resolve all these problems which made product development much easier in concurrent time than before.

2.3. Design Structure Matrix (DSM)

Complexity is usually present in any business and is a big challenge. Managing complex systems is therefore a core competency to successfully run any business ("Technical DSM Tutorial"). An engineering team who responsible in developing different module of PA should work together in order to obtain product architecture as a blueprint of the project. It is common to breakdown system or product into smaller elements as follows: subsystems, modules and component and define the interactions between components and subsystems. In order to achieve the performance of the system as a whole, these elements must be integrated to work together. One of the method to develop product architecture is Design Structure Matrix (DSM).

The Design Structure Matrix (DSM) is a simple tool that perform both the analysis and the management of complex systems. It enables the user to model, visualize, and analyse the dependencies among the entities of any system and derive suggestions for the improvement or synthesis of a system ("Technical DSM Tutorial").

The Design Structure Matrix (DSM) is also known as:

- Dependency Structure Matrix
- Dependency Source Matrix
- Dependency Structure Method
- Problem Solving Matrix (PSM)
- Design Precedence Matrix

A DSM is a square matrix, i.e. it has an equal number of rows and columns, which shows relationships between elements in a system. Since the behaviour of many systems is largely determined by interactions between their constituent elements, DSMs have become very useful and important in recent years ("Technical DSM Tutorial").

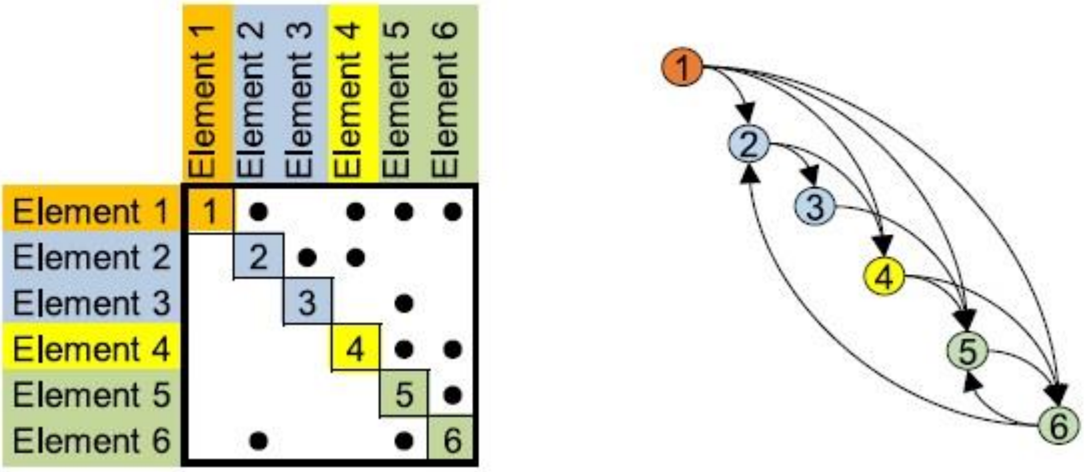


Figure 2.2 – Example of DSM (IC Convention) and its equivalent node-link diagram (Adapted from Browning, 2016)

Compared to other system modeling methods, a DSM has two main advantages:

- It provides a very simple and concise method to represent a complex system.
- It is amenable to few powerful analyses, like clustering and sequencing ("Technical DSM Tutorial").

Steward’s work on systems of equations in the early 1960s led to the first papers on DSM (Steward, 1967) as internal reports for general electric, but it was not until 1981 that his book (Steward, 1967a) and paper (Steward, 1967b) were published.

Aside from some citations by Warfield (e.g., Warfield, 1974), few references to Steward’s DSM works can be found until the late 1980s, when researchers at Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, and NASA began to apply and extend the method. The 1990s saw several developments, including the broadening of DSM applications beyond Steward’s temporal models to include static models of organizations and products. The new millennium brought an explosion of DSM research and applications across multiple industries and contexts. Browning’s 2001 review of the DSM literature cited about 100 DSM papers; there have been over 1000 since. These developments are of great interest to researchers and practitioners; therefore, it is valuable to provide an organized account of the evolving landscape to consolidate progress and provide a foundation for further advancement. Primarily targeting practitioners, Eppinger and Browning’s recent book (Eppinger and Browning, 2012) provided an introduction to DSM methods along with 44 industrial application examples. Primarily targeting researchers, a survey was conducted on recent DSM extensions and innovations in the scholarly literature and illuminates areas with a plethora of publications as well as areas offering excellent research opportunities. (Browning, 2016)

Further, the work on DSM in PD was categorized by Browning in 2016, as:

a) Building Product DSMs
   - Increasing model consistency and inter-rater reliability
   - Distinguishing types and strengths of interfaces/relationships
   - Constructing software architecture DSMs automatically from source code
   - Constructing a product DSM automatically from other models
   - Building function-to-function, concept-to-concept, and other types of product DSMs

b) Displaying Product DSMs
   - Showing nested module/subsystem structures with hierarchical DSMs
   - Showing varied types and strengths of interfaces/relationships
   - Showing change probability and impact as mini-graphs in the DSM

- Using DSM appendages to show external relationships
- Showing multiple product variants with a three-dimensional DSM

c) Analyzing Product DSMs

- Determining product modules
- Determining product modules
- Clustering via evolutionary algorithms
- Clustering with the criterion of component volatility and option value
- Sequencing to determine architectural levels
- Analyzing change propagation

2.3.1. Reading a DSM

One major benefit of DSM is that it's graphical nature of the matrix display format. This matrix provides a very compact, easily scalable, and readable representation of a system architecture. The figure given below shows a simple DSM model of a system with six different elements. The diagonal cells of the matrix represent the system elements. To keep the matrix compact, the elements' full names are listed to the left of the rows and can also be written above in the columns rather than in the diagonal cells. Each diagonal cell possesses potential to have inputs entering from its top and the bottom and the outputs leaving from matrix' left and right sides. The marks in the off-diagonal cells identify the sources and the destinations of these input and output interactions. Examining any row in the matrix reveals all of the outputs from the element in that row (which are inputs to other elements). Looking down any column of the matrix shows all of the inputs to the element in that column (which are outputs from other elements).

Many DSM resources use the opposite convention, the transpose of the matrix, with an element's inputs shown in its row and its outputs shown in its column. Following notation for these two conventions have been developed as follow ("Technical DSM Tutorial"):

- IR/FAD (Input in Rows/Feedback Above the Diagonal) convention: DSM with inputs shown in rows, outputs in columns; hence, any feedback marks will appear above the diagonal.

- IC/FBD (Input in Columns/Feedback Below the Diagonal) convention: DSM with inputs shown in columns, outputs in rows; hence, any feedback marks will appear below the diagonal.

The two conventions convey the same information; each is just the matrix transpose of the other. Both are widely used because of the diverse roots of matrix-based tools for modelling systems ("Technical DSM Tutorial").

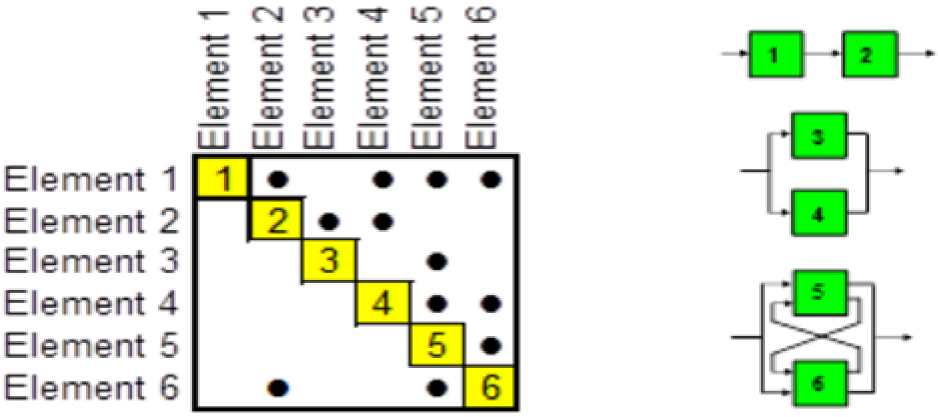


Figure 2.3 – Example of DSM (Adapted from "Technical DSM Tutorial")

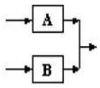
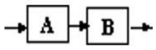
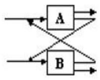
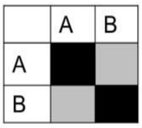
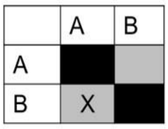
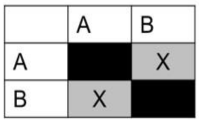
For example, as per IC/FBD convention, in the figure 2.3, reading across row 2, we see that element 2 provides outputs to elements 3 and 4. Reading down column 5, we see that element 5 receives inputs from elements 1, 3, and 4. Thus, a mark in an off-diagonal cell (e.g., cell 3,5) represents an interaction that is both an input and an output, depending on whether one takes the perspective of its provider (element 3) or its receiver (element 5).

<u>Types of Relationship in DSM</u>

From the table 2.2 one can make following conclusions:

a) Parallel – Tasks A and task B are mutually independent of each other and can be started together.
b) Sequential – Task A and Task B form a linear chain or sequence and B can be started after A is finished.
c) Coupled – Both tasks A and B are mutually interdependent and have to be done together

Table 2.2 – Types of Relationship in DSM (Adapted from Jayprakash, 2015)

Relationship	Parallel	Sequential	Coupled
Graph Representation			
DSM Representation			

To the right of the DSM in above figure are node-directed link diagram equivalents of portions of the DSM. Note that elements 1 and 2 form a linear chain or sequence, while elements 3 and 4 are independent, and elements 5 and 6 are interdependent or coupled.

This simple DSM example is called a binary DSM because the off-diagonal marks indicate merely the presence or absence of an interaction. The binary DSM representation can be extended in many ways by including further attributes of the interactions, such as the number of interactions and/or the importance, impact, or strength of each which might be represented by using one or more numerical values, symbols, shadings, or colours instead of just the binary marks in each of the off-diagonal cells. This extended form of DSM and is called a numerical DSM (“Technical DSM Tutorial”). Below given is an example of Numerical DSM:

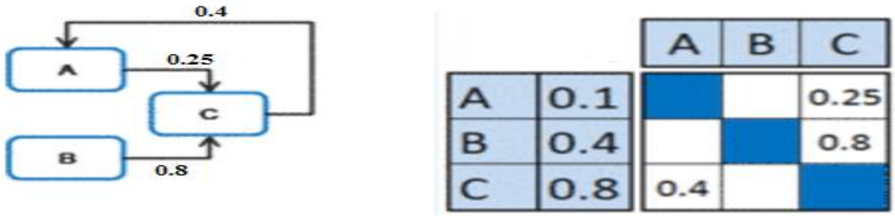


Figure 2.4 – Example of Numerical DSM (Adapted from “Technical DSM Tutorial”)

Additional attributes of the elements themselves may also be included by adding more columns to the left of the square matrix to describe, for example, the type, owner, or status of each element. Additional attributes of the interactions, such as their names, requirements, etc. are usually kept in separate repositories but may be linked to the DSM cells by numerical identification numbers or indices ("Technical DSM Tutorial").

2.3.2. Types of DSM

DSM can be divided into four types depending on the data types. It can be of components based in which all components of a product are taken as tasks and mapped on DSM. Similarly for team/people based, activity based or parameter based. The following table explains in brief about types of DSM:

Table 2.3 – Types of DSM (Adapted from "Technical DSM Tutorial")

DSM Data Types	Representation	Application	Analysis Method
Component based	Multi-component relationships	System architecting, engineering and design	Clustering
Team/People based	Multi-team or Organizational unit relationship	Organizational design, interface management, team integration	Clustering
Activity based	Activity input/output relationships	Process improvement, Project scheduling, activity sequencing, cycle time reduction	Sequencing & Partitioning
Parameter based	Design parameter Relationship	Sequencing Design Decision, Low level activity sequencing and process construction	Sequencing & Partitioning

2.3.3. Sequencing and Clustering

After drawing DSM we do some iteration known as Sequencing. It is the reordering of the DSM rows and columns such that the new DSM arrangement does not contain any feedback marks, thus transforming the DSM into an upper triangular form or lower triangular form in IC/FBD and IR/FAD convention respectively. For complex engineering systems, it is highly unlikely that simple row and column manipulation will result in an upper triangular form. Therefore, the analyst's objective changes from eliminating the feedback marks to moving them as close as possible to the diagonal (this form of the matrix is known as block triangular) ("Technical DSM Tutorial").

After Sequencing, our new goal becomes finding subsets of DSM elements (i.e. clusters or modules) that are mutually inter-dependent. Types of DSM exclusive or minimally interacting subsets, i.e. clusters as groups of elements that are interconnected among themselves to an important extent while being little connected to the rest of the system. This process is referred as "Clustering". In other words, clusters absorb most, if not all, of the interactions (i.e. DSM marks) internally and the interactions or links between separate clusters are eliminated or at least minimized.

Here is an example of DSM before sequencing and after Clustering.

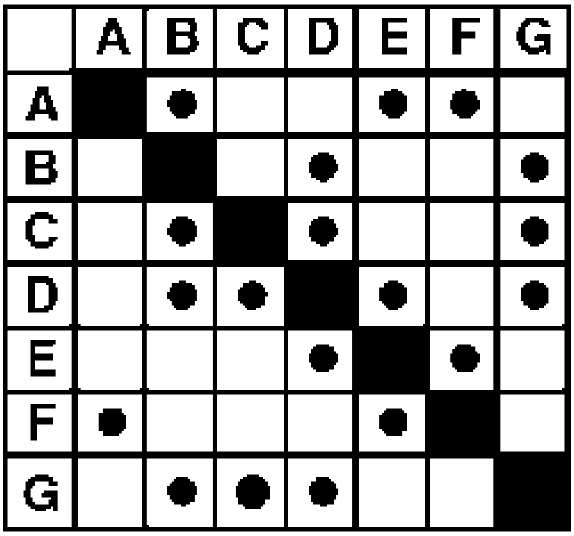


Figure 2.5 - Example of DSM before Sequencing
(Adapted from Yassine and Braha, 2003)

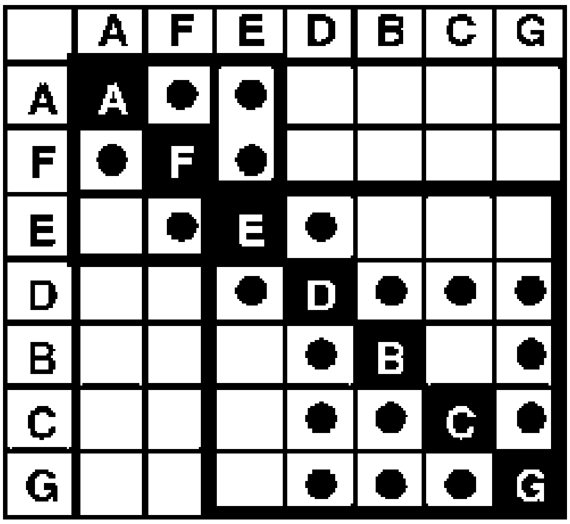


Figure 2.6. Example of DSM after Sequencing and Clustering
(Adapted from Yassine and Braha, 2003)

2.4. Domain Mapping Matrix

So far we have focused on DSMs within individual domains, but many applications, such as a need to show the organizational unit responsible for each activity in a process, transcend a single domain. For richer models across domains, a single DSM usually will not suffice. Whereas a DSM is always a square matrix, rectangular matrices have long been used to map relationships across domains. The product domain contains at least two prominent sub-domains, functions and components. Most of the product DSM applications discussed are for components, although some model functions. Both sub-domains matter, as does their relationship. The appropriate allocation of functions to components is a salient aspect of effective design. (Browning, 2016).

Complexity arises from the relationships and dependencies among items such as product development-related tasks and activities, product functionality, components in a product architecture, and people involved in the process. Variation among and the number of dependencies and relations determines the level of complexity (Danilovic and sandkull, 2005).
In 2001, Danilovic introduced 2 –D matrix' studies on product architecture vs. organization and in another paper the same year a study on Systems vs. Organization (Danilovic and Sandkull, 2005). He presented studies of dependencies between dual domains in product development. These dual domain and matrix-based analyses are called DMM (Danilovic and Sandkull, 2005). The DSM/DMM approaches are complementary to each other. While the first focus on one domain the other one focus on interactions between domains.

- N × N approach is named DSM,
- N × P approach is named DMM.

DMM is a rectangular two-dimensional matrix tool used to represent and analyze dependencies and relationships between two different domains. Domains can be

- components in the product architecture
- tasks in the processes
- people in the cross-functional teams
- Tools used in process
- Metrics used to measure needs and so on.

People Vs. Component	Fabricate the Cap	Manufacture hanging clip	Molding upper body	Mold & thread lower body	Manufacture of Spring	Molding Ink's Pipe	Fabricate the Tip
David	X						
Steward			X				
Mark						X	
Maria		X					
Michel				X			
Kirsi							X
John					X		

Figure 2.7 – Example of DMM (adapted from Shamsuzzoha, 2009)

There is no diagonal in the matrix around which to cluster items: items can be clustered anywhere in the matrix, using an algorithms by Mccormick et.all (Danilovic and Sandkull, 2005).

While a DSM is always a square matrix, a DMM will usually be rectangular, although it can be square in cases where two domains contain an equal number of elements in their respective systems.

In 2003, in a another DMM, some DMM analysis was introduced, Product requirements vs. Functional requirement, Functional requirement vs. Product architecture, Product requirement vs. Product specifications, and Functional requirement vs. Product specifications, and Product specifications vs. Product architecture (Danilovik and Sandkull, 2005).

In 2004 Danilovic and Browning dubbed such matrices *domain mapping matrices* (DMMs) and proposed a “periodic table” of then existing and potential DMM models across five project domains: product, process, organization, tools, and goals and also across five product domains: product, specification, parameters, and functionality (Browning, 2016) as shown in figure 2.8 and 2.9.

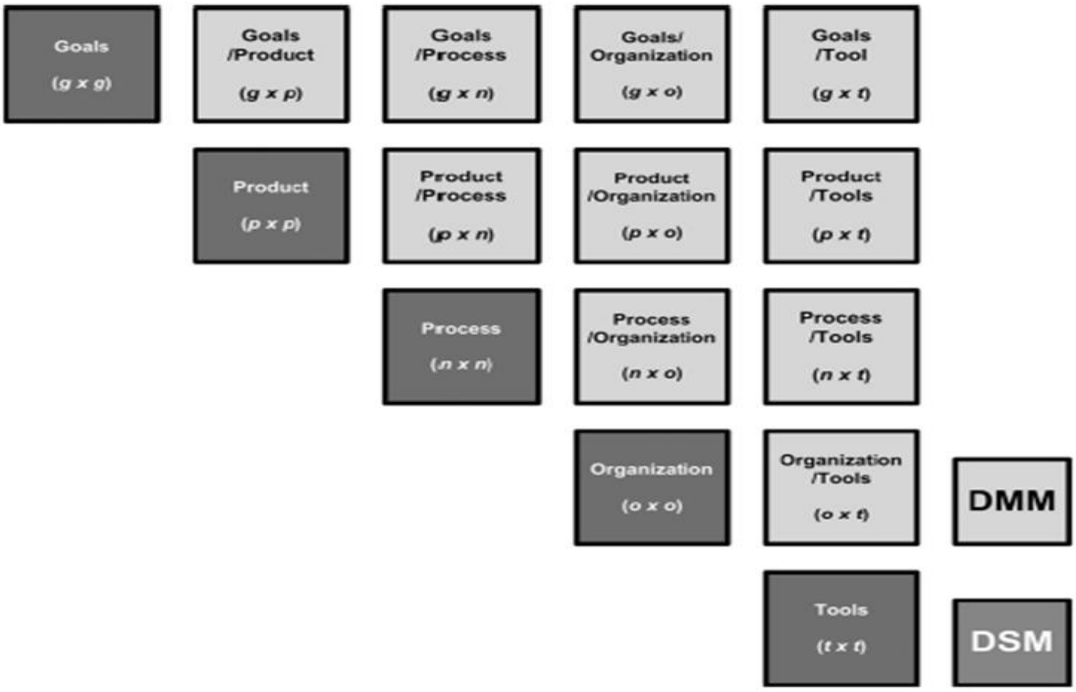


Figure 2.8 - Periodic Table of DSMs and DMMs for Project System
(Adapted from Danilovik and Browning, 2007)

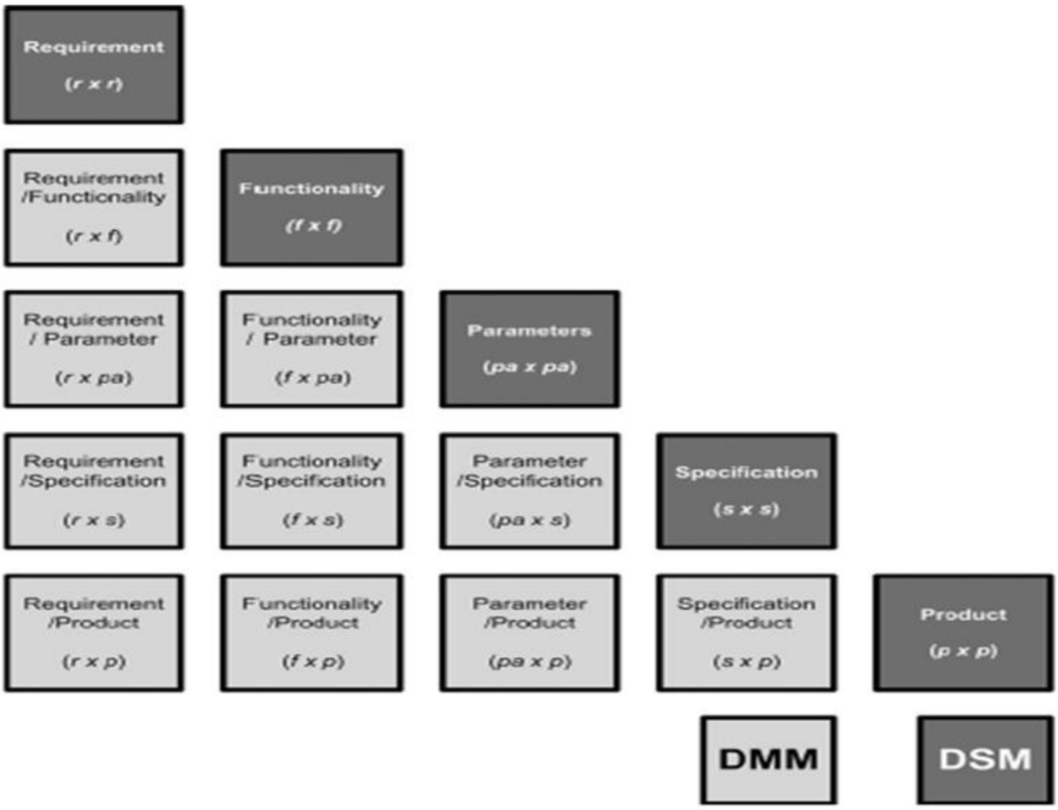


Figure 2.9 - Periodic Table of DSMs and DMMs for Product System
(Adapted from Danilovik and Browning, 2007)

Several researchers have used the DMM to model and explore function-to-component relationships. As with many DMMs, the function-component DMM can be used to generate both DSMs: multiplying this DMM by its transpose yields either the function-function or component-component DSM, depending on the order of operations. Bonjour et al. used this approach to derive a component DSM, which they then compared to a component DSM built through traditional methods. Danilovic and Browning (Danilovik and Browning, 2007) proposed additional product sub-domain DMMs, and further research is still needed to ground these in the engineering design literature and in relation to each other. (Browning, 2016)

Finally, the three analyses involve different foci:

- DSM sequencing is preferably used to analyze time dependent items such as activities based on the analysis of information flow and dependencies among them. However, Steward's recent work extends the original DSM approach to focus on general problem solving. In such an approach the same algorithm is applied and the analysis supports identifying the structure of a problem, without relation to time dimensions (Browning, 2016).
- DSM clustering is preferable for analyzing time-independent systems or single-domain analyses such as product architecture or project organization.
- DMM is preferable for analyzing relations and dependencies between domains and combinations of different domains.

With this comparison, we see that DSM and DMM differ substantially in points of departure, objective of analysis, and presentation of dependencies. While DSM employs both sequencing and clustering, depending on the domain, we have so far explored DMMs only through clustering, although sequencing may also be possible if one or more of the domains contains a time basis. Generally, all of the approaches are useful and complementary (Danilovik and Browning, 2007).

2.5. Multiple Domain Matrix

As we have discussed that DMM is a two domain matrix. But problem arises when one have to deal with more than two domains. The importance of modeling both inter- and intra- domain relationships simultaneously led to the advent of MDMs. An MDM could take the form of Danilovic and Browning's (Danilovik and Browning, 2007) "periodic table," an integration of various DSMs and their intervening DMMs.

A MDM is an extended version of DSM and DMM methods that includes three or more different domains and multiple relationships which is formed by combining DSMs and DMMs. It is a square matrix just like DSM. Skeleton of a MDM is shown in figure 2.10.

MDM	Component	Need	Metric
Component	DSM	DMM	DMM
Need	DMM	DSM	DMM
Metric	DMM	DMM	DSM

Figure 2.10 – Skeleton of MDM

Figure 2.11 shows a general MDM structure, consisting of a symmetric alignment of elements on both axes and element groups of different domains. This formation causes sub-matrices of DSM and DMM types.

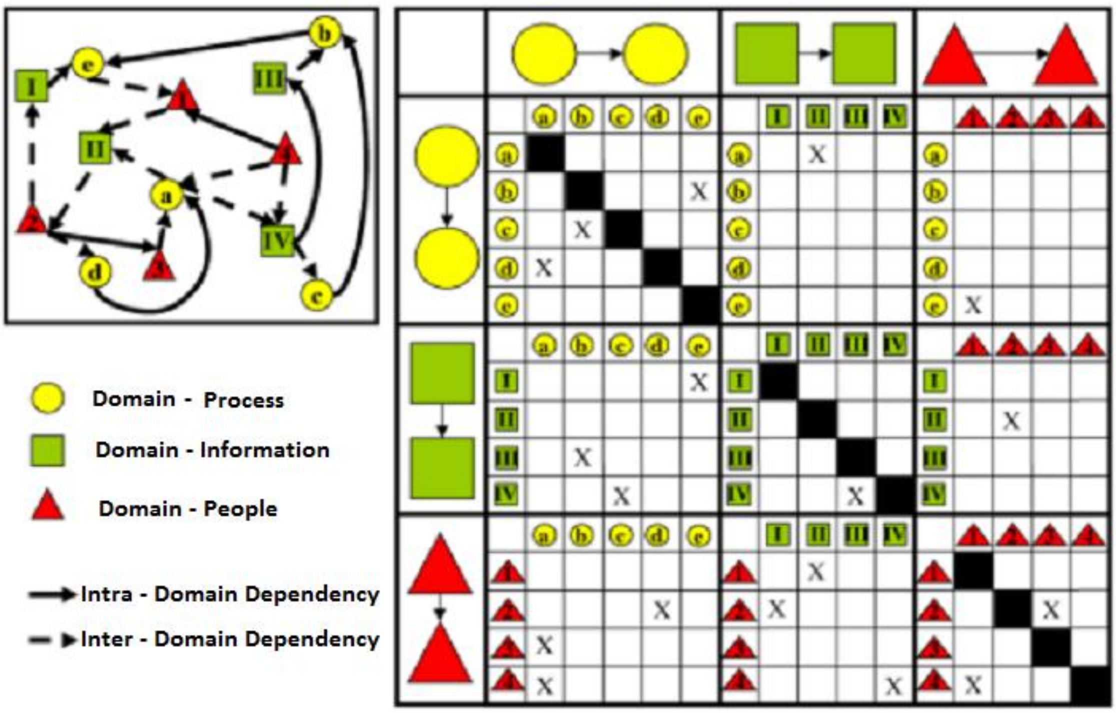


Figure 2.11 – Example of MDM creation

Figure 2.11 shows a general MDM structure, consisting of a symmetric alignment of elements on both axes and element groups of different domains. This formation causes sub-matrices of DSM and DMM types. The sub-matrices aligned along the MDM diagonal are DSMs, the sub-matrices in the upper and lower triangular of the MDM are consequentially DMMs as shown in figure 2.11. MDM help to focus on the result analysis on interdependencies, interactions and exchange information within and across the domains (Eichinger et.al. 2006).

In a MDM, bidirectional relationships are modeled, that is, the matrices in the upper and lower triangular of the MDM do not necessarily contain the same information. The MDM contains all intra- and inter-domain relations between the included domains. For example, the mutual relations between components and functions are modeled in the components-functions (directed impact from components to functions) matrix and the functions-components (directed impact from functions to components) matrix. For each domain combination two matrices exist that represent the directed relation between these domains. The MDM relations (the content of the MDM) are not necessarily symmetrical to the diagonal of the matrix, even if some sub-matrices may be symmetrical (a matrix that describes physical relations between components, for example, is always a symmetrical matrix due to the principle that action equals reaction). The aggregation of the product domains in a MDM offers new possibilities of interdependency analysis between the domains (Eichinger et.al. 2006).

From the outset, MDM models have been used to help build and verify DSMs and DMMs. For example, Sosa used a product (component) DSM and a product-org (component-to-person) DMM to derive an org DSM of potential interactions for comparison with an org DSM built through traditional means—thus enabling a comparison of predicted and actual communications in software development. Senthil Kumar and Varghese used product and org DSMs to derive a process DSM in the construction industry. Other MDM applications have explored and supported change propagation, knowledge management, engineering design, and manufacturing systems. Because the implications of design or engineering changes reach across the product, process, and organizational domains, several have used MDM models to investigate change propagation in various industries. (Browning, 2016).

MDM research is still in its infancy with many researchers trying a variety of applications. Much recent work in the DSM community has focused on MDM models, yet many opportunities exist to further codify and standardize MDM terminology and methods, categorize application areas, and develop analysis techniques. Although clustering and sequencing have been used with DSMs, and clustering with DMMs, it remains unclear how best to analyze an MDM containing a mix of static and temporal DSMs. MDMs also hold great promise for the emerging fields of “big data,” data science, and analytics. For example, huge DSMs can capture relationships among large groups of people, and DMMs can map those people onto other domains, such as organizational memberships, product and service preferences, and purchasing habits. Analyzing all of this information in tandem reveals patterns, clusters, cycles, segments, associations, “hot spots,” and so on. (Browning, 2016).

2.6. Quality Function Deployment

Quality function deployment (QFD) is ‘‘an overall concept that provides a means of translating customer requirements into the appropriate technical requirements for each stage of product development and production (i.e., marketing strategies, planning, product design and engineering, prototype evaluation, production process development, production, sales)’’.

Generally in past, Japanese industries started to formalize the QFD ideas when Mr. Oshiumi of the Kurume Mant plant of Bridgestone Tyre delivered a processing confirmation chart containing some of QFD's primary qualities in 1966 and K. Ishihara built up the thoughts of "functional deployment of business" like those of QFD and connected them to Matsushita in the late 1960s (Chan and Wu, 2002a).

However, it was Akao who initially understood the value of this approach in 1969 and needed to use its power amid the product design stage so that the product design attributes could be changed over into exact quality control focuses in the manufacturing quality control chart. After a few industrial trials, Akao composed a paper on this new approach in 1972 and called it hinshitsu tenkai (quality deployment). This paper and Nishimura (1972) were the initial two papers encouraging the then new idea of QFD known toward the West. Then it spread across the world. And more research was dedicated to QFD (Chan and Wu, 2002a).

2.6.1. Theory of QFD

Quality function deployment (QFD) is “a system to assure that customer needs drive the product design and production process” (Sullivan, 1986). QFD uses symbols and numerical rankings which thoroughly represents the connections between customer's needs, competitive and focused product, engineering metrics, and specifications.

Ordinarily, a QFD framework can be decomposed into four inter-connected stages to completely deploy the customer's needs stage by stage(Chan and Wu, 2002b). In QFD, each stage's vital output (HOWs), created from the stage's inputs or information sources (WHATs), are changed over into the following stage as its data sources (new WHATs). So each stage can be portrayed by a network of "WHATs" and "HOWs", which is simple and advantageous to manage in practice. The four QFD stages include: Stage I to make an interpretation of customer's needs into product configuration traits which will be called as technical measures; Stage II to make an interpretation of vital technical measures into parts qualities; Stage III to make an interpretation of vital parts attributes into process operations; and Stage IV to make an interpretation of key process operations into everyday production necessities. The principal stge of QFD, more often called house of quality (HOQ), is of strategic and fundamental significance in the QFD framework, since it is in this stage the customer needs for the product are identified and afterward, consolidate the manufacturing organization's focused needs, changed over into suitable specialized measures to fulfill the necessities.

At the end of the day, HOQ joins the "voice of the customer" to the "voice of the technician" through which process and manufacturing arrangements can be produced in other stages of the QFD framework. A house of quality (HOQ) includes list and examination of the "voice of the customer" which incorporates the customer requirements for an product, customer's perceptions on the relative significance of these necessities and the relative execution of the manufacturing organization and its principle rivals on the requirements. It additionally requires the generation and examination of the "voice of the technician" which incorporates the technical measures changed over from the customer needs, experts' assessments on the connection between every customer need and every technical measure, and the execution of the significant organizations as far as these technical measures.

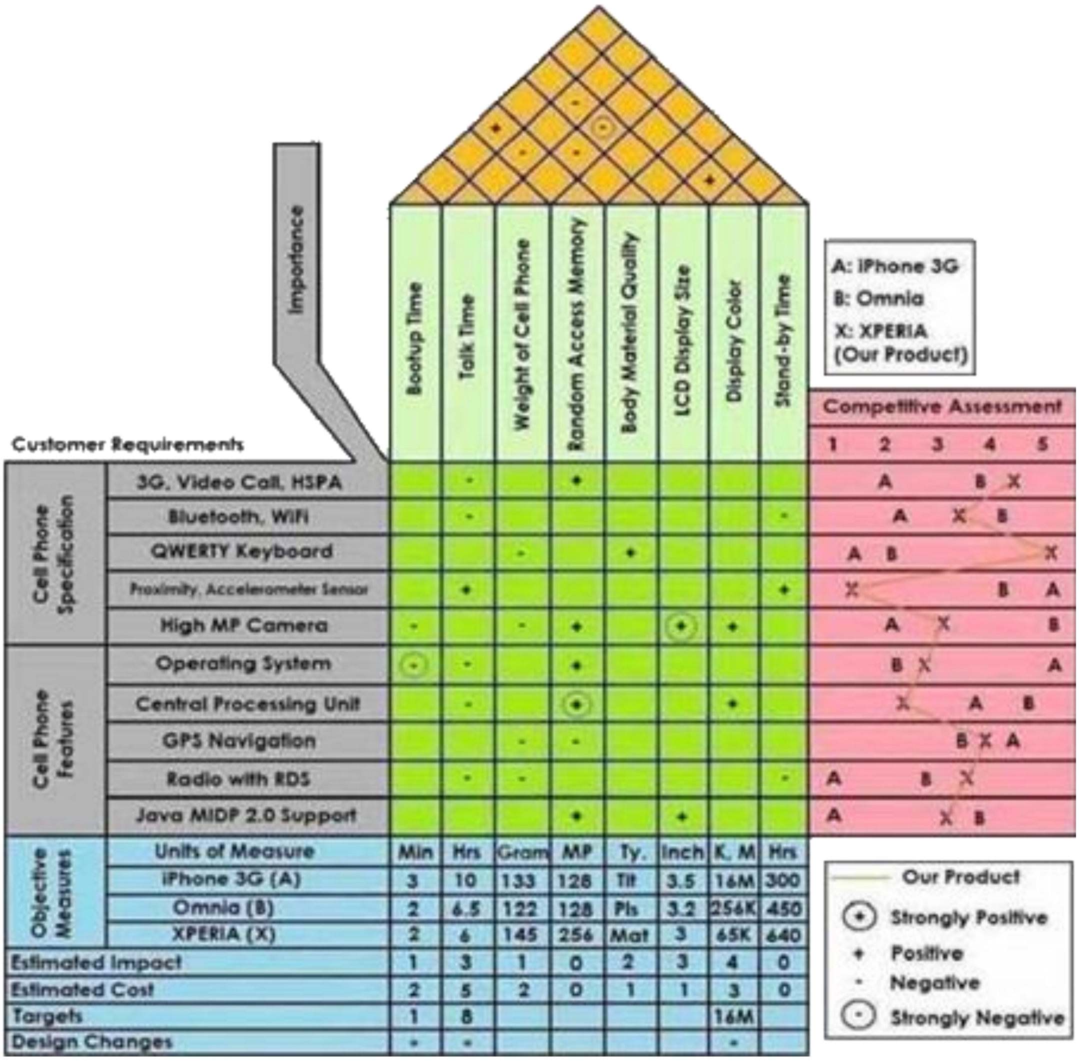


Figure No. 2.12 – Example of QFD (Adapted from QFD and House of Quality, n.d.)

With such a lot of data to be gathered and prepared, fabricating a HOQ might be too intricate to ever be finished and tantamount.

2.7. Connectivity Maps

Generally, it is possible to set up a matrix of indirectly connected system elements and to note the linking causes in the matrix cells. Hereby, the causes mean the system elements that are sited on the path connecting the indirectly linked elements (Maurer & Braun, 2008). One researcher applied such a notation for "connectivity maps", which indicate indirect dependencies in Domain-Mapping Matrices (DMMs) (Yassine et. al., 2003). Figure 2.13 shows the exemplary creation of a C-Map.

If two DMMs are apparent that provide the direct links between elements from domain B to domain A and from domain A to domain C, the approach on C-Maps derives indirect links from elements of domain B to domain C.

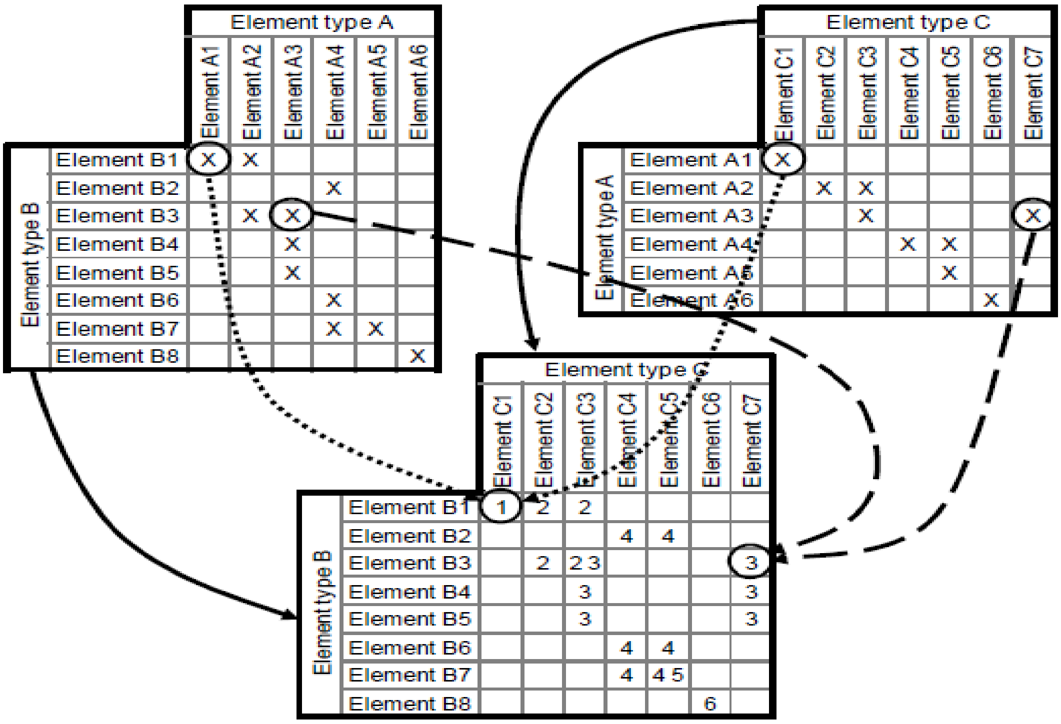


Figure 2.13 – Example of creating Connectivity Map (Adapted from Yassine et. al. 2003)

The figure depicts these elements from domain A in the matrix cells of the resulting DMM that cause the indirect links. In practice, limits of applicability exist for this notation of indirect dependencies. Complex systems often possess a high quantity of indirect dependencies. Thus, matrices representing all indirect dependencies can becme difficult to read. As well, indirect dependencies do probably not pass by one further system element only. In fact, many indirect dependencies result from dependency chains spanning several system elements. There are six general possibilities to define indirect dependencies. If these are considered simultaneously, the quantity of indirect dependencies further increases.

Figure 2.13 provides a simple example of a connectivity map and presents a schematic of how it might be constructed. As shown in the figure, the roots of the C-Map can be found in relationship maps. The element types B and C are put on the axes, while their respective relationships with element type A are used to map their connection.

In this example, the numbers in the cells refer to the particular subelements of A that connect particular subelements of B and C. A couple of specific examples highlighted in Figure 2.3 are (Yassine et. al., 2003):

- subelement A1 connects subelement B1 and C1; and
- subelement A3 connects subelements B3 and C7.

Note that some subelements of B and C might be connected by multiple subelements of A. For instance, subelements B7 and C5 are connected by both sub-elements A4 and A5. This means that both A4 and A5 have relationships with B7 and C5.

If the 'X' marks in matrix A and B are replaced by a numerical scheme (such as using '1' to indicate the existence of a relationship and '0' to indicate the absence), then a numerical C-Map can be generated. A numerical C-Map can serve as an accounting ledger for identifying what connections are complex. For instance, if the '1' or '0' scheme is used, then those cells that have numbers greater than 1 are clearly indicating the existence of multiple relationships (Eppinger, 2001). Because it is a matrix, the connectivity map is compact and easily constructed and modified, which are important attributes in any project management tool. However, the complexity in the C-Map is introduced by the method or code used to capture the three-way relationships (Yassine et. al., 2003).
The key step in the development of the connectivity map is representing the connections between the three types of subelements in such a way that the map remains compact yet is still able to communicate information and support design or management analysis. The marks in the cells, which represent the relationships (e.g. influence, dependence and association) between sub-elements must be carefully designed.

The challenge is for the coded marks in the cells to not just indicate the existence of a relationship, but also to inform on an aspect of that relation by illuminating the nature of the connection between the three types of elements. The specific coding or legend chosen is highly dependent on the particular elements being studied and the complexity of their inter-relationship. Numerical rankings, symbols, or colours can be used to indicate the relative strengths, direction, or significance of the relation.

Table: 2.4 Difference between DSM, DMM and MDM (Adapted from Shamsuzzoha, 2009)

SPECIFICATION	DSM	DMM	MDM
	Single Domains Square Matrix	Two Domains Square And rectangle	Three or more Domains Square
Directional Dependency	Single Directinal	Bi-Directional	Tri-directional and more clustering
Partitioning algorithm	Sequencing, Clustering Triangulisation Clustering of blocks along diagnol	Clustering	Clustering of items
Integration Analysis	Sequencing, minimizing feedback loops, cluster of items	Clustering of items, dependencies or interface identification	Clusters of items dependencies or interface identification
Repesentaion	Information flow between components, organization vs operations	Information flow between components vs specifications product architecture vs operational processes	Information flow among components vs architecture vs operational process .etc

However, unless the A–C (or A–B) relation is very simple, an alphanumeric code (or symbol or colour) will probably have to be developed for the subelements of A to keep the connectivity map relatively compact and intuitive. If the relationships are extremely complex, the C-Map can become overwhelmed or equivocal. There is only few research found dedicated to or involving "Connectivity Maps" in PD.

2.8. Drawbacks of current methods and research gaps

All the aforementioned matrix-based tools provide some insight into the dependency structure of a complex system or process (i.e. a dependency map). However, they fail to expose and explore the logic behind these dependencies. (Yassine et all., 2003) For example, if an element B is shown to relate to another element C, it is not clear why and how this relation exists and where it stems from.

If it is done so, the analyst will be capable of understanding the complexity of a system better and in turn has more information to manage these dependencies and leverage them to the improvement of system performance. (Shoval et all, 2016)

Even C-Map fails to accommodate multiple dimensions in more than one way. For example, one needs to develop three Connective Maps to ensure that a designer may need to study any relationship between any domain i.e. linkage between A and B via C or between A and C via B or between Band C via A. So a need is felt to develop a new system of capturing relationships among domains and both inter- and intra- domains with the cause of that linkage. This is the very objective of this research.

2.9 Objectives of the Dissertation:

The key objectives of this dissertation are as follows:

- To analyze the current matrix based methods used in PD for capturing the relationships within a factor and among factors.
- To develop a matrix based method capable of capturing, visualizing, and drawing conclusions about a product's architecture, its complexity, and modularity.
- To illustrate the constructions, working, and superiority of the new method by comparing it with the present methods by using one example.
- To communicate, in the example, what attributes are emerging, what combinations of factors are interacting to cause or create particular attributes.

Chapter 3

RESEARCH METHODOLOGY

The methodology followed during this research is clearly explained in this chapter. Each step of research is fragmented into a number of steps and is clearly demarcated and elaborated. The outline of the steps is shown in figure 3.1.

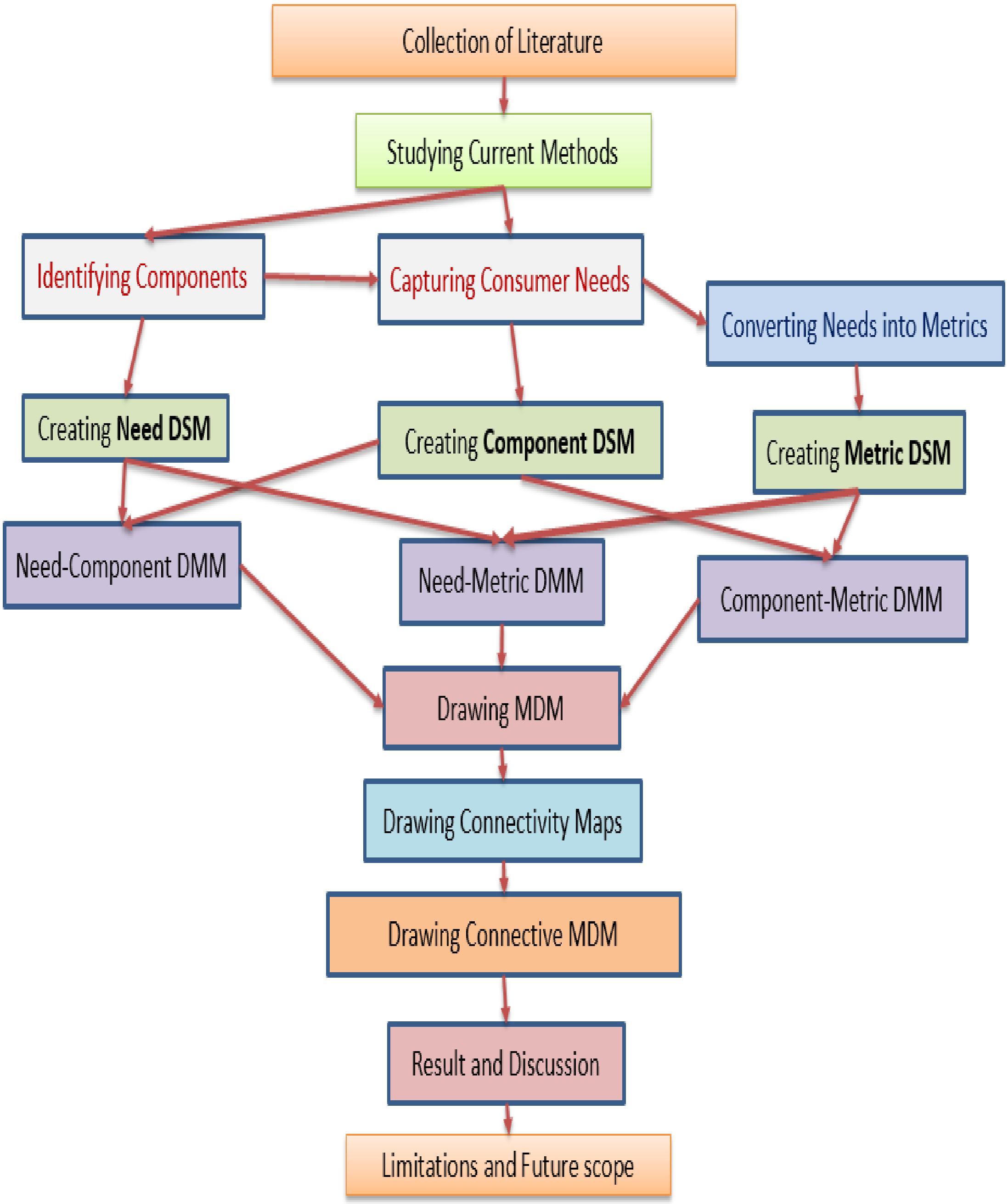


Figure 3.1 – Outline of Research Methodology

3.1. Collection of Literature

In order to find relevant articles concerning the research objectives, a systematic literature review was conducted. Using a broad range of sources resulted in the sufficient literature to conduct this study. Online databases of technical & management publishers were used, viz. Science Direct, Emerald Insight, Inderscience, Taylor and Francis, IEEE online, Google Scholar, etc.

With the purpose of finding relevant literature for this study keywords were used based on preliminary readings and logical thinking. As the research context was described, the main concepts are product development, relationship matrix, product architecture, modularity, information management and clustering. Using these as a guideline a list of the keywords that are related to the main concepts was drawn. The keywords used are presented below. The articles found in the preliminary search based on the keywords were filtered by grounding on the relevance in title and abstract of the articles. In the second phase, another selection took place by using another criterion so that the most relevant articles would remain for further review. The keywords used for finding relevant articles were:

- Product development
- Product design
- Design Structure Matrix
- Domain Mapping Matrix
- Multi Domain Matrix
- Connectivity Maps
- Use of DSM in Product Design/Development

3.2. Planning scope of research

After studying the number of papers on the use of DSM, DMM, MDM and C-Maps a gap in research is identified. A plan to develop a new method capable of capturing multiple domains and the cause of the relationship through other variable and test its utility and viability was formed.

So the scope of the research is limited to that only. The product chosen is bicycle because it is quite common and it is easy to find the customers to understand their needs w.r.t. bicycle. Also, it is easy to visualize, identify the components and their interactions.

Thereafter, three domains were chosen to capture their relationships because it is easy to show the interaction of three domain MDM here in the report as complexity and size increase with a number of domains. Three domains chosen were Customer's Needs, Components of the bicycle (mentioned as 'components' in report hereafter) and Metrics to measure the needs (mentioned as 'metrics' in the report hereafter).

The success of the DSM method is determined by an appropriate system decomposition and by the accuracy of the dependence relationships. Therefore, it is vital to decompose the system under study carefully into a comprehensive set of meaningful system elements. So each domain is decomposed into sub-elements as given in next points.

3.3. Identifying Customer Needs

Needs are consumer's desires from specific product or problems that customers intend to solve with the purchase of a good or service. These are critical sensible perceptions that customers use to evaluate various products/services. However great your product or service is, the straightforward truth is that nobody will purchase it in the event that they don't need it. Also, you won't convince anybody that they need or need to purchase what you're putting forth unless you obviously comprehend what it is your customer truly need.

So to get the real need of the customers a survey was done in Jaipur city at many bicycle shops at various location of Jaipur city of Rajasthan, India. Customer who visited these shops to purchase bicycle were directly contacted and asked about their desires and expectations from the bicycle. The target customers were the youth of age range 15 years to 30 years of age and that too only those who opted for ranger style of bicycle. A small range and a particular style of the product are chosen so as to get accurate data with least possible deviations. Because different age groups have different needs and different style of product fulfills different needs.

After getting the needs, they were refined, cut shorts into short and simple language and some clubbed to other need looking similar to each other to avoid the duplication.

Then each Need is given a number to represent in matrices of DSM, DMM, MDM, and C-Maps. The list of final needs identified is given in Table 3.1.

Table no. 3.1 – List of Needs

Need Number	Need
N1	Economical
N2	Good Looking
N3	Less Shocks
N4	Less Force required
N5	Seating Comfort
N6	Light Weight
N7	Strong Built
N8	Maneuverability
N9	Easy Brakes
N10	High Speed
N11	All Terrain

3.4. Converting Needs into Metrics

There is a wise old saying, *you can only manage what you can measure,* and this applies to PD also. Metrics are the parameters that can be measured used by engineers to measure the needs. Metrics are needed to understand the level of process performance, project performance, and product performance. They are needed to set goals and measure the trend and rate of improvement.

Proper metrics need to be selected. Improper metrics can optimize the performance of a product development sub-process at the expense of global sub-optimization. Improper metrics can require significant effort to collect data and develop without providing meaningful information of any real benefit. Criteria for effective metrics are:

- Keep them simple and minimum
- Base them on business objectives and the business process – avoid those that cause dysfunctional behavior
- Keep them practical – avoid metrics that require significant additional data collection and effort

List of needs was presented to the expert in the field of the bicycle to get convert them into appropriate metrics. Metrics of each need is given in Table 3.2.

Table No. 3.2 – List of Needs and their metrics

Need No.	Need	Metric
N1	Economical	Cost per unit
N2	Good Looking	Appearance Ratings (AR)
N3	Less Shocks	Suspension Spring Stiffness,Vibration Coefficient
N4	Less Force required	Mechanical Advantage
N5	Seating Comfort	Height, Length
N6	Light Weight	Kerb Weight
N7	Strong Built	Frame Toughness
N8	Maneuverability	Speed, Kerb Weight, Height, Length
N9	Easy Brakes	Brake Friction
N10	High Speed	Speed, Sprocket-Teeth Ratio, Lubricant Viscosity
N11	All Terrain	Height, Frame Toughness, Tyre Width

Just like needs, each Metric is given a particular number to be represented in matrices as shown in Table 3.3.

Table 3.3 – List of Metrics

Metric No.	Metrics
M1	Frame Toughness
M2	Mechanical Advantage
M3	Appearance Ratings (AR)
M4	Kerb Weight
M5	Suspension Spring Stiffness
M6	Sprocket-Teeth Ratio
M7	Height
M8	Length
M9	Lubricant Viscosity
M10	Tyre Width
M11	Brake Friction
M12	Vibration Coefficient
M13	Cost
M14	Speed

3.5. Identifying components

Components are physical assemblies or parts of a product that are decomposed to a defined level of detail. The component can be a part or a subassembly has multiple parts fulfilling any particular need. In short components are subsections of the product. Components of the bicycle are identified with the help of a subject expert in bicycle manufacturing. List of components and their number is given in Table 3.4.

Table 3.4 – List of Components

Component Number	Component
C1	Pedal & Crank
C2	Chain Ring
C3	Chain Set
C4	Hub
C5	Wheel
C6	Frame
C7	Brake Set
C8	Brake Lever
C9	Seat
C10	Suspensions
C11	Handle
C12	Carrier
C13	Fender
C14	Tyres

3.6. Creating DSM of each domain

Once the appropriate system elements or set of activities that comprise a project have been identified, they are listed in the DSM as row and column labels in the same order. The elements within the matrix are then identified by asking the appropriate managers or expert in the group for the minimum set of parameters that influence their own sub-system and contribute to its behavior.

Approach and steps to create a DSM are explained below:

This section is taken and adapted from Qi *Dong's Ph.D. Thesis, MIT, 2002*.

3.6.1. Define the system and its scope

Since the DSM is a tool that studies the design process as a system with many interacting elements, it is important to define the boundary of the system in order to focus the research work. Different system definition results in the different output of the DSM.

3.6.2. List all the system elements

Initially, the system elements can be chosen based on the existing project plans, engineers' suggestions, etc. The author of the DSM usually defines the initial set of system elements based on the reading of design documentation. However, experience shows that the initially defined system elements often need to be modified in the process of assigning interactions to them. A critical review of the list of elements in collaboration with engineering staff or other relevant experts is therefore necessary.

3.6.3. Study the information flow between system elements

The third step is to study the information flow between system elements. Reading the design documents as well as interviewing experienced engineers who were working on the particular product is a good source of knowledge. Since the DSM is a tool to analyze the design project and to seek improvements, it is important that the data is accurate. Although talking to engineers in person is time-consuming, the interviewer can usually gather accurate information and gain a very good insight into the system. However, when necessary, one may have to trade the speed of data collection with the quality of the data.

Step 2 and Step 3 are highly iterative. A very deep understanding of the system usually

results in modification of the initial system elements.The system elements in this thesis research were modified many times during the interviews and documentation readings in order to represent the system accurately.

3.6.4. Complete the matrix to represent the information flow

Having collected the elements and the dependencies, initially, a binary DSM can be built to represent the basic dependency structure and information flows between various system elements. A binary DSM serves as a good start for preliminary analysis; however, a better understanding of the system (or project) might require the use of a numerical DSM that will provide better system understanding and allow for more detailed analysis.

3.6.5. Give the matrix to the engineers and managers to comment on and use

DSM provides aid to design engineers and engineering managers to understand the design process better and approach the communication more systematically. Hence, the constructed DSM's are usually provided to the engineers and manager who participated in their building to receive comments. This creates, on the one hand, transparency about the benefits of building a DSM, as seeing the entire picture of the design process like never before makes many engineers rethink their current practice, and seek improvements. On the other hand, the collection of comments can further aid the refinement of the structure of the DSM.

3.7. Creating DMMs

In order to demonstrate the information exchange among needs, components and metrics, three DMMs are created with the help of subject expert naming need-component DMM, component-metric DMM, and need Metric DMM.

3.8. Creating MDM

As explained above, an MDM is a matrix capable of capturing relationships both intra- and inter- domains. It is formed by combining DSMs and DMMs. As DSMs and DMMs have already been created this task becomes so easy. The only point is to take care of the order of domains. One has to put domains in the same order in rows as that in a column otherwise resulting matrix will be of no use.

It is to be noted that in DSMs, DMMs, and MDM, there is no use of clustering, tearing or partitioning done because it is out of the scope of this research. The aim of the research is to develop a new method not to solve the DSM, DMM or MDM. These have been created for the very same reason as mentioned below:

- DSMs will be used in connective MDM, a new matrix as the outcome of this research.
- All the three DMMs will be used for developing C-Maps which in turn used to develop connective MDM.
- MDM will be used to comparing with the new method showing the superiority of new matrix.

3.9. Developing Connectivity Maps

A Connectivity Map is formed by using two DMMs having one common domain and one different domain. Formation process is already explained in Chapter 1, ‘Introduction’. Three C-Maps have been created from three DMMs i.e. Components vs Needs via Metrics, Needs vs Metrics via Components, and Components vs Metrics via Needs. These will also be used for Connective MDM.

3.10. Connective MDM

Now the last step is to create a matrix that is able to explain all the possible relationships. Creating a Connective MDM matrix is similar to MDM, the only difference is that instead of DMMs Connective Maps formed in section 3.9 are used.

Chapter 4

ANALYSIS AND RESULTS

4.1. DSMs

The DSMs created with the help of subject expert are shown in the the figure 4.1, 4.2 and 4.3 each DSM show the relationship among the components in the very same domain i.e. Need, Metric and Components.

A relationship between any two elements is shown by 'X' in the matrix and blank cells shows that there is no relationship between the respective elements of the matrix. For example 'less shocks is connected to economical so a 'X' is marked in the corresponding cell. It is to be kept in mind that only significant relations are marked. There may be components related slightly and those relationships are not considered to ease the process of this research.

		N 1	N 2	N 3	N 4	N 5	N 6	N 7	N 8	N 9	N 10	N 11
		Economical	Good Looking	Less Shocks	Less Force required	Seating Comfort	Light Weight	Strong Built	Maneuverability	Easy Brakes	High Speed	All Terrain
N 1	Economical			X			X	X		X		X
N 2	Good Looking											
N 3	Less Shocks	X				X		X				X
N 4	Less Force required						X		X		X	
N 5	Seating Comfort			X								
N 6	Light Weight	X			X			X	X		X	X
N 7	Strong Built	X		X			X					X
N 8	Maneuverability				X		X					X
N 9	Easy Brakes	X										X
N 10	High Speed				X		X					
N 11	All Terrain	X		X			X	X	X	X		

Figure 4.1 - DSM of Needs

The standard approach to DSMs, like sequencing, partitioning, clustering and tearing, can be applied here to get a better result. But it is kept out of the scope of this study. These DSMs are directly used as it is in MDM and Connective MDM.

		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14
		Frame Toughness	Mechanical Advantage	AR	Kerb Weight	Spring Stiffness	Sprocket-Teeth Ratio	Height	Length	Lubricant Viscosity	Tyre Width	Brake Friction	Vibration Coefficient	Cost	Speed
M1	Frame Strength				X									X	
M2	Mechanical Advantage						X			X	X				X
M3	Appearancec Rating (AR)							X	X		X			X	
M4	Kerb Weight	X						X	X		X			X	X
M5	Spring Stiffness												X		
M6	Sprocket-Teeth Ratio		X											X	X
M7	Height			X	X									X	X
M8	Length			X	X									X	
M9	Lubricant Viscosity		X												X
M10	Tyre Width		X	X	X							X		X	X
M11	Brake Friction										X				
M12	Vibration Coefficient					X									X
M13	Cost	X		X	X		X	X	X		X				
M14	Speed		X		X		X	X		X	X		X		

Figure 4.2 - DSM of Metrics

		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14
		Pedal-Crank	Sprocket	Chain Set	Hub	Wheel	Frame	Brakes Set	Brake Lever	Seat	Suspensions	Handle	Carrier	Fender	Tyres
C1	Pedal-Crank		X				X								
C2	Sprocket	X		X		X	X								
C3	Chain Set		X		X	X									
C4	Hub			X		X	X								
C5	Wheel		X	X	X		X	X						X	X
C6	Frame	X	X		X	X		X		X	X	X	X	X	
C7	Brakes Set					X	X		X						
C8	Brake Lever							X				X			
C9	Seat						X								
C10	Suspensions						X						X		
C11	Handle						X		X						
C12	Carrier						X				X				
C13	Fender					X	X								X
C14	Tyres					X								X	

Figure 4.3 - DSM of Components

4.2. DMM

DMMs were created to capture the relationships among the components of two different domains. In this stage also all the DMMs were created with subject expert's advice. Here also only significant level of relationships are taken into account. 'X' in a matrix shows that a relation exists between the respective elements of the domains.

		N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11
		Economical	Good Looking	Less Shocks	Less Force required	Seating Comfort	Light Weight	Strong Built	Maneuverability	Easy Brakes	High Speed	All Terrain
C1	Pedal-Crank				X		X	X	X		X	
C2	Sprocket				X						X	
C3	Chain							X			X	
C4	Hub							X			X	
C5	Wheel	X	X				X	X		X	X	X
C6	Frame	X	X			X	X	X				X
C7	Brakes Set	X								X		
C8	Brake Lever		X					X		X		
C9	Seat		X	X		X			X			
C10	Suspensions	X		X		X	X	X	X			X
C11	Handle	X	X			X	X	X	X			
C12	Carrier	X	X				X	X				
C13	Fender						X					
C14	Tyres	X			X		X		X	X	X	X

Figure 4.4 - Components-Needs DMM

		M 1	M 2	M 3	M 4	M 5	M 6	M 7	M 8	M 9	M 10	M 11	M 12	M 13	M 14
		Frame Strength	Mechanical Advantage	Colour	Kerb Weight	Spring Stiffness	Sprocket-Teeth Ratio	Height	Length	Lubricant Viscosity	Tyre Width	Brake Friction	Vibration Coefficient	Cost	Speed
N 1	Economical	X			X			X	X		X		X	X	
N 2	Good Looking			X											
N 3	Less Shocks	X				X						X	X	X	
N 4	Less Force required		X		X		X			X	X				
N 5	Seating Comfort							X	X						
N 6	Light Weight	X			X			X	X					X	X
N 7	Strong Built	X			X	X							X	X	
N 8	Maneuverability				X			X	X		X	X			X
N 9	Easy Brakes											X			
N 10	High Speed		X				X			X	X				X
N 11	All Terrain	X	X		X	X	X	X			X	X	X	X	

Figure 4.5 - Needs-Metrics DMM

		M 1	M 2	M 3	M 4	M 5	M 6	M 7	M 8	M 9	M 10	M 11	M 12	M 13	M 14
		Frame Strength	Mechanical Advantage	AR	Kerb Weight	Spring Stiffness	Sprocket-Teeth Ratio	Height	Length	Lubricant Viscosity	Tyre Width	Brake Friction	Vibration Coefficient	Cost	Speed
C 1	Pedal-Crank	x			x										x
C 2	Sprocket		x				x			x				x	x
C 3	Chain		x				x			x				x	
C 4	Hub									x			x		x
C 5	Wheel	x			x			x			x	x		x	x
C 6	Frame	x		x	x	x		x	x				x	x	
C 7	Brakes Set			x								x		x	
C 8	Brake Lever														
C 9	Seat			x									x	x	
C 10	Suspensions				x	x							x	x	
C 11	Handle	x		x	x			x					x	x	
C 12	Carrier	x		x	x				x						
C 13	Fender			x	x						x				
C 14	Tyres				x						x			x	x

Figure 4.6 - Components-Metrics DMM

4.3. MDM

A MDM is created by combining all three DSMs and three DMMs. One also need to take transpose of three DMMs to fill the entire matrix of MDM. This MDM is very basic one and shows relationships among intra- and inter- domains' components. This is very useful but it do not show the cause of any relationship and also the level of the relationship. This MDM is used to be compared with the Connective MDM, final matrix of the new method developed out of this study.

		C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10	C 11	C 12	C 13	C 14	N 1	N 2	N 3	N 4	N 5	N 6	N 7	N 8	N 9	N 10	N 11	M 1	M 2	M 3	M 4	M 5	M 6	M 7	M 8	M 9	M 10	M 11	M 12	M 13	M 14
C 1	Pedal+Crank		X				X												X		X	X	X		X		X			X										X
C 2	Sprocket	X		X		X	X												X						X			X				X			X				X	X
C 3	Chain Set		X		X	X																X			X			X				X			X				X	
C 4	Hub			X		X	X															X			X										X			X		X
C 5	Wheel		X	X	X		X	X						X	X	X	X				X	X		X	X	X	X			X			X			X	X		X	X
C 6	Frame	X	X		X	X		X		X	X	X	X	X		X	X			X	X	X				X	X			X	X		X	X				X	X	
C 7	Brakes Set					X	X		X							X								X													X		X	
C 8	Brake Lever							X				X					X					X		X																
C 9	Seat						X										X	X		X			X															X	X	
C 10	Suspensions						X						X			X		X		X	X	X	X			X				X	X							X	X	
C 11	Handle						X		X							X	X			X	X	X	X				X			X			X					X	X	
C 12	Carrier						X				X					X	X				X	X					X			X				X						
C 13	Fender					X	X								X						X									X						X				
C 14	Tyres					X								X		X			X		X		X	X	X	X				X						X			X	X
N 1	Economical					X	X	X			X	X	X		X			X			X	X		X		X	X			X			X	X		X		X	X	
N 2	Good Looking					X	X		X	X		X	X																X											
N 3	Less Shocks									X	X					X				X		X				X	X				X						X	X	X	
N 4	Less Force required	X	X												X						X		X		X			X		X		X			X	X				
N 5	Seating Comfort						X			X	X	X						X															X	X						
N 6	Light Weight	X				X	X				X	X	X	X	X	X			X			X	X		X	X	X			X			X	X					X	X
N 7	Strong Built	X		X	X	X	X		X		X	X	X			X		X			X					X	X			X	X							X	X	
N 8	Manoeuvrability	X								X	X	X			X				X		X					X				X			X	X		X	X			X
N 9	Easy Brakes					X		X	X						X	X										X											X			
N 10	High Speed	X	X	X	X	X									X				X		X							X				X			X	X				X
N 11	All Terrain					X	X				X				X	X		X			X	X	X	X			X	X		X	X	X	X			X	X	X	X	
M 1	Frame Strength	X				X	X						X			X		X			X	X				X				X									X	
M 2	Mechanical Advantage		X	X															X						X	X						X			X	X				X
M 3	AR																X																							
M 4	Kerb Weight	X				X	X				X	X	X	X	X	X			X		X	X	X			X	X						X	X		X			X	X
M 5	Spring Stiffness						X				X							X				X				X												X		
M 6	Sprocket-Teeth Ratio		X	X															X						X	X		X											X	X
M 7	Height					X	X					X				X				X	X		X			X				X									X	X
M 8	Length						X						X			X				X	X		X							X									X	
M 9	Lubricant Viscosity		X	X	X														X						X			X												X
M 10	Tyre Width					X								X	X	X			X				X		X	X		X		X							X		X	X
M 11	Brake Friction					X		X										X					X	X		X										X				
M 12	Vibration Coefficient				X		X			X	X	X				X		X				X				X					X									X
M 13	Cost		X	X		X	X	X		X	X	X			X	X		X			X	X				X	X			X		X	X	X		X				
M 14	Speed	X	X		X	X									X						X		X		X			X		X		X	X		X	X		X		

Figure 4.7 - Needs-Metrics-Components MDM

4.4. Connectivity Maps

Following the steps mentioned in chapter 3, three C-maps were created to gain the insights into the relationships among components of two domains and the cause of those relationships i.e. connectivity elements. Three maps created are shown in figure 4.8, 4.9, 4.10

		M 1	M 2	M 3	M 4	M 5	M 6	M 7	M 8	M 9	M 10	M 11	M 12	M 13	M 14
		Frame Strength	Mechanical Advantage	AR	Kerb Weight	Spring Stiffness	Sprocket-Teeth Ratio	Height	Length	Lubricant Viscosity	Tyre Width	Brake Friction	Vibration Coefficient	Cost	Speed
N 1	Economical	5,6,11,12		6,7,11,12	5,6,10,11,12,14	6,10		5,6,11	6,12		5,14	5,7	6,10,11	5,6,7,10,11,14	5,6,14
N 2	Good Looking	5,6,11,12		6,9,11,12	5,6,11,12	6		5,6,11	6,12		5	5	6,9,11	5,6,9,11	5,6
N 3	Less Shocks			9	10	10							9,10	9,10	
N 4	Less Force required	1	2		1,14		2			2	14			1,2,14	2,14
N 5	Seating Comfort	6,11		6,9,11	6,10,11	6,10		5,6,11	6				6,9,10,11	6,9,10,11	6
N 6	Light Weight	1,5,6,11,12		6,11,12,13	1,5,6,10,11,12,13,14	6,10		5,6,11	6,12		5,13,14	5	6,10,11	1,5,6,10,11,14	5,6,14
N 7	Strong Built	1,5,6,11,12	2,3	6,11,12	1,5,6,10,11,12	6,10	2,3	5,6,11	6,12	2,3,4	5	5	4,6,11	1,2,3,5,6,11	2,4,5,6
N 8	Maneuverability	1,11		9,11	1,10,11,14	10		11			14		9,10,11	1,9,10,11,14	14
N 9	Easy Brakes	5		7	5,14			5			5,14	5,7		5,7,14	5,14
N 10	High Speed	1,5	2,3		1,5,14		2,3	5		2,3,4	5,14	5	4	1,2,3,5,14	2,4,5,14
N 11	All Terrain	5,6		6	5,6,10,14	6,10		5,6	6		5,14	5	6,10	5,6,10,14	5,6,14

Figure 4.8 - Need-Metric Connectivity Map via Component

In this C-map, one can see the relationship and also the connectivity element. For example, need ‘Easy Brakes’ (N9) is linked with metric ‘Appearance Ratings (AR)’ (M3) via C7 i.e. Brake sets. Hence these matrices are very helpful in understanding the complex relationships in an easy way.

The color code, relationship have five or more connective elements is given Red and yellow to four elements is given to the important relationships in all C-maps as they are affecting the most number of the connective element. Rest of the relationships are treated at a normal level. This will benefits the reader or PD team to read the level of relationships and identify the critical ones.

		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14
		Frame Strength	Mechanical Advantage	AR	Kerb Weight	Spring Stiffness	Sprocket-Teeth Ratio	Height	Length	Lubricant Viscosity	Tyre Width	Brake Friction	Vibration Coefficient	Cost	Speed
C1	Pedal-Crank	6,7	4,10		4,6,7,8	7	4,10	6,8	6,8	4,10	4,8,10	8	7	6,7	6,8,10
C2	Sprocket		4,10		4		4,10			4,10	4,10				10
C3	Chain	7	10		7	7	10			10	10		7	7	10
C4	Hub	7	10		7	7	10			10	10		7	7	10
C5	Wheel	1,6,7,11	10,11	2	1,6,7,11	7,11	10,11	1,6,11	1,6	10	1,10,11	9,11	1,7,11	1,6,7,11	6,10
C6	Frame	1,6,7,11	11	2	1,6,7,11	7,11	11	1,5,6,11	1,5,6		1,11	11	1,7,11	1,6,7,11	6
C7	Brakes Set	1			1			1	1		1	9	1	1	
C8	Brake Lever	7		2	7	7						9	7	7	
C9	Seat	3		2	8	3		5,8	5,8		8	3,8	3	3	8
C10	Suspensions	1,3,6,7,11	11		1,6,7,8,11	3,7,11	12	1,5,6,8,11	1,5,6,8		1,8,11	3,8,11	1,3,7,11	1,3,6,7,11	6,8
C11	Handle	1,6,7,		2	1,6,7,8	7		1,5,6,8	1,5,6,8		1,8	8	1,7	1,6,7	6,8
C12	Carrier	1,6,7		2	1,6,7	7		1,6	1,6		1		1,7	6,7	6
C13	Fender	6,7			6,7	7		6	6				7	6,7	6
C14	Tyres	1,6,11	4,10,11		1,4,6,8,11	11	4,10,11	1,6,8,11	1,6,8	4,10	1,4,8,10,11	9,8,11	1,11	1,6,11	6,8,10

Figure 4.9 - Component-Metric Connectivity Map via Need

		C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10	C 11	C 12	C 13	C 14
		Pedal+Crank	Sprocket	Chain Set	Hub	Wheel	Frame	Brakes Set	Brake Lever	Seat	Suspensions	Handle	Carrier	Fender	Tyres
N 1	Economical	1,4	13	13	12	1,4,7,10,13	1,4,7,8,12,13	13		12,13	4,12,13	1,4,7,12,13	1,4,8	4,10	4,10,13
N 2	Good Looking						3	3		3		3	3	3	
N 3	Less Shocks	1	13	13	12	1,11,13	1,5,12,13	11,13		12,13	5,12,13	1,12,13	1		13
N 4	Less Force required	4	2,6,9	2,6,9	9	4,10	4				4	4	4	4,10	4,10
N 5	Seating Comfort					7	7,8					7	8		
N 6	Light Weight	1,4,14	13,14	13	14	1,4,7,13,14	1,4,7,,8,13	13		13	4,13	1,4,7,13	1,4,8	4	4,13,14
N 7	Strong Built	1,4	13	13	12	1,4,13	1,4,5,12,13	13		12,13	4,5,12,13	1,4,12,13	1,4	4	4,13
N 8	Maneuverability	4,14	14		14	4,7,10,11,14	4,7,8	11			4	4,7	4,8	4,10	4,10,14
N 9	Easy Brakes					11		11							
N 10	High Speed	14	2,6,9,14	2,6,9	9,14	10,14	5							10	10,14
N 11	All Terrain	1,4	2,6,13	2,6,13	12	1,4,7,10,11,13	1,4,7,12,13	11,13		12,13	4,5,12,13	1,4,7,12,13	1,4	4,10	4,10,13

Figure 4.10 - Component-Need Connectivity Map via Metric

4.5. Connective MDM

The goal of the example given in this study is to capture, envision, and reach conclusions about a product's architecture, its perplexing complexity, and its capacity to help effective, sensible changes that can adjust the customer-perceived traits. The final result of this study is shown in the matrix called ‘Connective MDM’ (in figure 4.11). As stated in chapter 3, it is formed by joining the three DSMs shown in figure 4.1, 4.2, and 4.3, three C-Maps are shown in figure 4.8, 4.9, and 4.10 and the transpose of these C-Maps. Just like conventional MDM, Connective MDM’s diagonal matrices are DSMs showing the relationships across the elements of the same domain, like components.

In figure 4.11, one can see that it captures the relationships in three domains’ components with the connective components like M4 and C5 have some relationship or dependencies via connectivity elements N1, N7, N8, and N9.

It shows which combinations of relationships are most important or critical for the product architecture or to satisfy the customer needs. The most important relationships are given red colour, slightly less important is given yellow colour and normal linkages are given white colour. For example in the connective MDM shown in figure 4.11, the strongest level of relationship exists between M4 and N6 as it has six connective elements, most in the matrix. Whereas those cells which do not have any entry i.e. blank cells represents that there are no relationships like the cell across C2 and M12.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14
C1		X				X									1,4		1	4		1,4,14	1,4	4,14		14	1,4	6,7	4,10		4,6,7,8	7	4,10	6,8	6,8	4,10	4,8,10	8	7	6,7	6,8,10
C2	X		X		X	X									13		13	2,6,9		13,14	13	14		2,6,9,14	2,6,13		4,10		4		4,10			4,10	4,10				10
C3		X		X	X										13		13	2,6,9		13	13			2,6,9	2,6,13	7	10		7	7	10			10	10		7	7	10
C4			X		X	X									12		12	9		14	12	14		9,14	12	7	10		7	7	10			10	10		7	7	10
C5		X	X	X		X	X						X	X	1,4,7,10,13		1,11,13	4,10	7	1,4,7,13,14	1,4,13	4,7,10,11,14	11	10,14	1,4,7,10,11,13	1,6,7,11	10,11	2	1,6,7,11	7,11	10,11	1,6,11	1,6	10	1,10,11	9,11	1,7,11	1,6,7,11	6,10
C6	X	X		X	X		X		X	X	X	X	X		1,4,7,8,12,13	3	1,5,12,13	4	7,8	1,4,7,,8,13	1,4,5,12,13	4,7,8		5	1,4,7,12,13	1,6,7,11	11	2	1,6,7,11	7,11	11	1,5,6,11	1,5,6		1,11	11	1,7,11	1,6,7,11	6
C7					X	X		X							13	3	11,13			13	13	11	11		11,13	1			1			1	1		1	9	1	1	
C8							X				X															7		2	7	7						9	7	7	
C9						X									12,13	3	12,13			13	12,13				12,13	3		2	8	3		5,8	5,8		8	3,8	3	3	8
C10						X						X			4,12,13		5,12,13	4		4,13	4,5,12,13	4			4,5,12,13	1,3,6,7,11	11		1,6,7,8,11	3,7,11	12	1,5,6,8,11	1,5,6,8		1,8,11	3,8,11	1,3,7,11	1,3,6,7,11	6,8
C11						X		X							1,4,7,12,13	3	1,12,13	4	7	1,4,7,13	1,4,12,13	4,7			1,4,7,12,13	1,6,7,		2	1,6,7,8	7		1,5,6,8	1,5,6,8		1,8	8	1,7	1,6,7	6,8
C12						X				X					1,4,8	3	1	4	8	1,4,8	1,4	4,8			1,4	1,6,7		2	1,6,7	7		1,6	1,6		1		1,7	6,7	6
C13					X	X								X	4,10	3		4,10		4	4	4,10		10	4,10	6,7			6,7	7		6	6				7	6,7	6
C14					X								X		4,10,13		13	4,10		4,13,14	4,13	4,10,14		10,14	4,10,13	1,6,11	4,10,11		1,4,6,8,11	11	4,10,11	1,6,8,11	1,6,8	4,10	1,4,8,10,11	9,8,11	1,11	1,6,11	6,8,10
H1	1,4	13	13	12	1,4,7,10,13	1,4,7,8,12,13	13		12,13	4,12,13	1,4,7,12,13	1,4,8	4,10	4,10,13			X			X	X		X		X	5,6,11,12		6,7,11,12	5,6,10,11,12,14	6,10		5,6,11	6,12		5,14	5,7	6,10,11	5,6,7,10,11,14	5,6,14
H2						3	3		3		3	3	3													5,6,11,12	9	6,9,11,12	5,6,11,12	6,9		5,6,11	6,9,12		5	5	6,11	5,6,11	5,6
H3	1	13	13	12	1,11,13	1,5,12,13	11,13		12,13	5,12,13	1,12,13	1		13	X				X		X				X				10	10							10	10	
H4	4	2,6,9	2,6,9	9	4,10	4				4	4	4	4,10	4,10						X		X		X		1	2		1,14		2			2	14			1,2,14	2,14
H5					7	7,8					7	8					X									6,11		6,11	6,10,11	6,10		5,6,11	6				6,10,11	6,10,11	6
H6	1,4,14	13,14	13	14	1,4,7,13,14	1,4,7,,8,13	13		13	4,13	1,4,7,13	1,4,8	4	4,13,14	X			X			X	X		X	X	1,5,6,11,12		6,11,12,13	1,5,6,10,11,12,13,14	6,10		5,6,11	6,12		5,13,14	5	6,10,11	1,5,6,10,11,14	5,6,14
H7	1,4	13	13	12	1,4,13	1,4,5,12,13	13		12,13	4,5,12,13	1,4,12,13	1,4	4	4,13	X		X			X					X	1,5,6,11,12	2,3,9	6,9,11,12	1,5,6,10,11,12	6,9,10	2,3	5,6,11	6,9,12	2,3,4	5	5	4,6,11	1,2,3,5,6,11	2,4,5,6
H8	4,14	14		14	4,7,10,11,14	4,7,8	11			4	4,7	4,8	4,10	4,10,14				X		X					X	1,11		11	1,10,11,14	10		11			14		10,11	1,10,11,14	14
H9					11		11								X										X	5	9	7,9	5,14	9		5	9		5,14	5,7		5,7,14	5,14
H10	14	2,6,9,14	2,6,9	9,14	10,14	5							10	10,14				X		X						1,5	2,3		1,5,14		2,3	5		2,3,4	5,14	5	4	1,2,3,5,14	2,4,5,14
H11	1,4	2,6,13	2,6,13	12	1,4,7,10,11,13	1,4,7,12,13	11,13		12,13	4,5,12,13	1,4,7,12,13	1,4	4,10	4,10,13	X		X			X	X	X	X			5,6		6	5,6,10,14	6,10		5,6	6		5,14	5	6,10	5,6,10,14	5,6,14
H1	6,7		7	7	1,6,7,11	1,6,7,11	1	7	3	1,3,6,7,11	1,6,7,	1,6,7	6,7	1,6,11	5,6,11,12	5,6,11,12		1	6,11	1,5,6,11,12	1,5,6,11,12	1,11	5	1,5	5,6				X									X	
H2	4,10	4,10	10	10	10,11	11				11				4,10,11		9		2			2,3,9		9	2,3							X			X	X				X
H3					2	2		2	2		2	2			6,7,11,12	6,9,11,12			6,11	6,11,12,13	6,9,11,12	11	7,9		6														
H4	4,6,7,8	4	7	7	1,6,7,11	1,6,7,11	1	7	8	1,6,7,8,11	1,6,7,8	1,6,7	6,7	1,4,6,8,11	5,6,10,11,12,14	5,6,11,12	10	1,14	6,10,11	1,5,6,10,11,12,13,14	1,5,6,10,11,12	1,10,11,14	5,14	1,5,14	5,6,10,14	X						X	X		X			X	X
H5	7		7	7	7,11	7,11		7	3	3,7,11	7	7	7	11	6,10	6,9	10		6,10	6,10	6,9,10	10	9		6,10												X		
H6	4,10	4,10	10	10	10,11	11				12				4,10,11				2			2,3			2,3			X											X	X
H7	6,8				1,6,11	1,5,6,11	1		5,8	1,5,6,8,11	1,5,6,8	1,6	6	1,6,8,11	5,6,11	5,6,11			5,6,11	5,6,11	5,6,11	11	5	5	5,6				X									X	X
H8	6,8				1,6	1,5,6	1		5,8	1,5,6,8	1,5,6,8	1,6	6	1,6,8	6,12	6,9,12			6	6,12	6,9,12		9		6				X									X	
H9	4,10	4,10	10	10	10									4,10				2			2,3,4			2,3,4			X												X
H10	4,8,10	4,10	10	10	1,10,11	1,11	1		8	1,8,11	1,8	1		1,4,8,10,11	5,14	5		14		5,13,14	5	14	5,14	5,14	5,14		X		X							X		X	X
H11	8				9,11	11	9	9	3,8	3,8,11	8			9,8,11	5,7	5				5	5		5,7	5	5										X				
H12	7		7	7	1,7,11	1,7,11	1	7	3	1,3,7,11	1,7	1,7	7	1,11	6,10,11	6,11	10		6,10,11	6,10,11	4,6,11	10,11		4	6,10					X									X
H13	6,7		7	7	1,6,7,11	1,6,7,11	1	7	3	1,3,6,7,11	1,6,7	6,7	6,7	1,6,11	5,6,7,10,11,14	5,6,11	10	1,2,14	6,10,11	1,5,6,10,11,14	1,2,3,5,6,11	1,10,11,14	5,7,14	1,2,3,5,14	5,6,10,14	X			X		X	X	X		X				
H14	6,8,10	10	10	10	6,10	6			8	6,8	6,8	6	6	6,8,10	5,6,14	5,6		2,14	6	5,6,14	2,4,5,6	14	5,14	2,4,5,14	5,6,14		X		X		X	X		X	X		X		

Figure 4.11 – Connective MDM

Chapter 5

DISCUSSION AND CONCLUSION

This study has presented the construction and utility of a DSM, DMM, MDM, C-Map, and Connective MDM. A compact, simple, and matrix-based framework for recognizing, capturing and surveying connections between various sorts of components is presented. It can be utilized with physical components of a product, processes or activities of PD process, consumer needs, and product attributes, raw materials requirements and also to assess products or processes. A procedure for creating the C-Map was displayed, alongside basic example to show the construction and analysis of that matrix. An example of the bicycle was also presented which clears the approach of this new method.

From the discussion in chapter 3 and 5, it can be specified that a C-Map has three components. These diverse component groups can be marked as row components, column components, and the connectivity components. The connectivity component ought to be objective, subject to quantification, and should bolster the improvement or development of the column component from the row component. The connectivity components should state what needs the column components ought to fulfill with a specific end goal to create or bolster the coveted state represented by the row components. From the column components' perspective, the column components speak to how the connectivity components will be supported and satisfied. Following the very same logic, in general, the relationship model of the components in any C-Map can be shown as:

$$\text{What} \longleftrightarrow \text{What} \longleftrightarrow \text{How}$$

That is, the row components depict "What" is desired. The connectivity components depict impartially "What" the row components appeal from the column components. The column components portray "How" the target requirement of the connectivity components will be met.

In a first look, connective MDM looks similar to conventional MDM but if one pay close attention then the real difference between both will emerge. Each relationship or dependencies are clearly stated in this matrix and that too showing the linkage to third domain's components. One can get why and how a relationship exists and which the third factor will be affected by this relationship. This shows the superiority of this matrix over conventional MDM.

A major benefit of Connective MDM is that it not only establishes the relationship through connectivity element but also determines the level of relationship or dependency i.e. strong or weak linkage. The level can be established by simply looking at the number of connective elements, more the number of connective elements stronger is the relationship.

Connective MDM can be used in any industry or any project. It can be used in Project management where the need is to capture the dependencies among various activities that too of different departments maintaining their hierarchy, sequence or precedence. Similarly, it can be used in a multi-project scenario where one needs to keep in mind the dependencies of activities in different projects.

Further, the systemic structure proposed in this research has a wide range of application and can be utilized to enhance execution, managerial capacities, and adequacy of any industry.

Chapter 6

LIMITATIONS AND FUTURE SCOPE

This research is very limited and conducted just to develop a new method of capturing complex relationships which are easy to understand. The main limitation of this study is that this method is developed keeping the only PD in mind. Its utility is also checked by only one example in PD only and that too from a single city. Generalization can be expanded by gathering information from different nations and from diverse industries. This method can be used to conduct more case studies to enrich it, confirm its robustness and effectiveness in all industries. Its utility can be checked in project management and any other areas where DSM has proved very helpful.

Another limitation of this research is that sequencing, partitioning, tearing and/or clustering of DSM, DMM, and MDM is not done. This is because the main aim of this research is to compare Connective MDM and conventional MDM and to prove the superiority of new method over MDM. This can be done without these mentioned steps. But one can try to compare these two after clustering or sequencing. This comparison may give a better result.

The example in this research is based on the expert advice of a single subject expert to gain insights into components of bicycle and capturing relationships in three domains. This concerns legitimacy and reliability of the study. It presumes that the respondent knows about PD, DSM, and DMM and have the required level learning and experience. So in future, this method can be tested by getting data from multiple respondents to increase the reliability.

This research is conducted by using only three domains and not account the level of relationship or dependencies i.e. strong or weak or moderate. It assumes that all linkages are of the same level. This is not true in all cases in this dynamic and complex world. Therefore in a follow-up research can be dedicated to checking the effectiveness of connective MDM with taking into account the different level of dependencies.
There is very less research done for developing the algorithm for clustering the MDM which can be served as the topic of research.

REFERENCES

[1] Batallas, D. A., & Yassine, A. A. (2006). Information Leaders In Product Development Organizational Networks: Social Network Analysis Of The Design Structure Matrix. Ieee Transactions On Engineering Management, 53(4), 570-582.

[2] Black, T. A., Fine, C. H., & Sachs, E. M. (1990). A method for systems design using precedence relationships: An application to automotive brake systems.

[3] Bonjour, E., Deniaud, S., Dulmet, M., & Harmel, G. (2009). A fuzzy method for propagating functional architecture constraints to physical architecture. Journal of Mechanical Design, 131(6), 061002.

[4] 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), 292-306.

[5] Browning, T. R. (2016). Design structure matrix extensions and innovations: a survey and new opportunities. IEEE Transactions on Engineering Management, 63(1), 27-52.

[6] Chan, L. K., & Wu, M. L. (2002). Quality function deployment: A literature review. European journal of operational research, 143(3), 463-497.

[7] Chan, L. K., & Wu, M. L. (2002). A systematic approach to quality function deployment with a full illustrative example.

[8] Clark, K. B., & Fujimoto, T. (1991). Product development performance: Strategy, organization, and management in the world auto industry. Harvard Business Press.

[9] Cooper, R. G., & Kleinschmidt, E. J. (1995). Benchmarking the firm's critical success factors in new product development. Journal of product innovation management, 12(5), 374-391.

[10] Danilovic M, Browning T. (2004) A formal approach for domain mapping matrices (DMM) to complement design structure matrices (DSM). In: Proceedings of the sixth design structure matrix (DSM) international workshop. September 12–14th, 2004. Cambridge, UK: Trinity Hall College, University of Cambridge, Cambridge Engineering Design Centre.

[11] Danilovic, M., & Sandkull, B. (2005). The use of dependence structure matrix and domain mapping matrix in managing uncertainty in multiple project situations. International journal of project management, 23(3), 193-203.

[12] Danilovic, M., & Browning, T. R. (2007). Managing complex product development projects with design structure matrices and domain mapping matrices. International journal of project management, 25(3), 300-314.

[13] Eichinger, M., Maurer, M., & Lindemann, U. (2006). Using multiple design structure matrices. In DS 36: Proceedings DESIGN 2006, the 9th International Design Conference, Dubrovnik, Croatia.

[14] Eppinger, S. D., Whitney, D. E., Smith, R. P., & Gebala, D. A. (1994). A model-based method for organizing tasks in product development. Research in engineering design, 6(1), 1-13.

[15] Eppinger, S., 1991, Model-based approaches to managing concurrent engineering, Journal of Engineering Design, 2, 283–290.

[16] Eppinger, S. D., & Browning, T. R. (2012). Design structure matrix methods and applications. MIT press.

[17] Gulati, R. K. (1996). The coupling of product architecture and organizational structure decisions (Doctoral dissertation, Massachusetts Institute of Technology).

[18] Guo, L. (2008). Perspective: An analysis of 22 years of research in JPIM. Journal of Product Innovation Management, 25(3), 249-260.

[19] Hauser, J. R., & Clausing, D. (1988). The house of quality, 66, 63–73.

[20] Jayprakash,T.L. DSM (Dependency/Design Structure Matrix). 1st ed. 2015. Retrieved on 8 Sept. 2016 from http://www.dsmweb.org/

[21] Jensen, B., & Harmsen, H. (2001). Implementation of success factors in new product development–the missing links?. European Journal of Innovation Management, 4(1), 37-52.

[22] Kehat, E., & Shacham, M. (1973). Chemical process simulation programs-2. Process Technology International, 18(3), 115-118.

[23] Ko, Y. T. (2013). Optimizing product architecture for complex design. Concurrent Engineering, 21(2), 87-102.

[24] Li, S., & Mirhosseini, M. (2012). A matrix-based modularization approach for supporting secure collaboration in parametric design. Computers in Industry, 63(6), 619-631.

[25] Lin, J. S., Chang, J. H., & Chang, Y. C. (2014). A structural process-oriented mechanism for supplier selection involved in the early stage of new product development. International Journal of Business and Systems Research, 8(3), 228-255.

[26] Luh, D. B., Ko, Y. T., & Ma, C. H. (2009). A dynamic planning approach for new product development. Concurrent Engineering, 17(1), 43-59.

[27] MacDuffie, J. P. (2000). Automotive" Build to Order": The modularity–E-business link. Communication to the. In IMVP Annual Sponsor Meeting (pp. 27-29).

[28] Mccormick Wt, Schweitzer Pj, White Tw. Problem Decomposition And Data Reorganization By A Clustering Technique. Operat Research 1972;20(5):993–1009.

[29] QFD and House of Quality, n.d., retrieved on 05.04.2017 from http://www.transtutors.com/homework-help/industrial-management/product-development/quality-function-deployment.aspx.

[30] Senthilkumar, V., & Varghese, K. (2009). Structured methodology to formulate drawing dependency structure matrix for construction design. Architectural Engineering and Design Management, 5(4), 225-248.

[31] Shamsuzzoha, A. H. M. (2009). Restructuring design processes for better information exchange. International Journal of Management and Enterprise Development, 7(3), 299-313.

[32] Sosa, M. E. (2008). A structured approach to predicting and managing technical interactions in software development. Research in Engineering Design, 19(1), 47-70.

[33] Steward, D. V. (1962). On an approach to techniques for the analysis of the structure of large systems of equations. Siam Review, 4(4), 321-342.

[34] Steward, D. V. (1965). Partitioning and tearing systems of equations. Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis, 2(2), 345-365.

[35] Steward, D. V. (1967). The design structure system. General Electric, CA 67APE6.

[36] Steward, D. V. (1981), Systems Analysis and Management. New York, NY, USA: Petrocelli Books, 1981.

[37] Steward, D. V. (1981). The design structure system: A method for managing the design of complex systems. IEEE transactions on Engineering Management, (3), 71-74.

[38] Steward, D. V. (2003, October). It’s all about problem solving. In Proceedings of the 5th international design structure matrix (DSM) workshop, Cambridge, UK.

[39] Sullivan LP. Quality function deployment. Quality Progress (1986).19(6),39–50.

[40] Technical DSM Tutorial. (n.d.). Retrieved September 08, 2016, from http://www.dsmweb.org/en/understand-dsm/technical-dsm-tutorial.html

[41] Ulrich, K. T., & Eppinger, S. D. (2000). Product design and manufacturing.

[42] Ulrich, K. T., Eppinger, S. D. and Goyal, A. (2009), “Product Design & Development”, 4th edition, McGraw-Hill, New York.

[43] Warfield, J. N. (1974). Developing subsystem matrices in structural modeling. IEEE Transactions on Systems, Man, and Cybernetics, (1), 74-80.

[44] Yadav, O. P., Nepal, B. P., & Jain, R. (2007). Managing product development process complexity and challenges: a state-of-the art review. Journal of design research, 6(4), 487-508.

[45] Yang, Z., & Zhou, J. (2010, December). Engineering portal with coordination mechanism for collaborative product development. In Industrial Engineering and Engineering Management (IEEM), 2010 IEEE International Conference on (pp. 2263-2267). IEEE.

[46] Yassine, A., Falkenburg, D., & Chelst, K. (1999). Engineering design management: an information structure approach. International Journal of production research, 37(13), 2957-2975.

[47] Yassine, A., Whitney, D., Daleiden, S., & Lavine, J. (2003). Connectivity maps: modeling and analysing relationships in product development processes. Journal of Engineering Design, 14(3), 377-394.

[48] Zhong, P., Song, J., & Liu, M. (2008, April). DSM based process analysis for concurrent product development and its implementation. In Computer Supported Cooperative Work in Design, 2008. CSCWD 2008. IEEE, 12th International Conference, 619-624.

Analysing the Relationships in Product Architecture using DSM, DMM, and MDM

ORIGINALITY REPORT

%18	%14	%10	%6
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.iese.illinois.edu Internet Source	%5
2	m.designsociety.org Internet Source	%3
3	www.redteam.se Internet Source	%1
4	www.softwarecatalog.info Internet Source	%1
5	Submitted to University of Salford Student Paper	%1
6	Submitted to Indian Institute of Management Student Paper	%1
7	Danilovic, M.. "Managing complex product development projects with design structure matrices and domain mapping matrices", International Journal of Project Management, 200704 Publication	%1
8	Submitted to Malaviya National Institute of Technology	%1