9^{TH} INTERNATIONAL DESIGN STRUCTURE MATRIX CONFERENCE, DSM'07
16 – 18 OCTOBER 2007, MUNICH, GERMANY

A SYSTEMATIC METHOD FOR MODELLING AND ANALYSING CONCEPTUAL DESIGN INFORMATION

Gregory M Mocko, Georges M Fadel, Joshua D Summers, Jonathan RA Maier, and Thulasiram Ezhilan

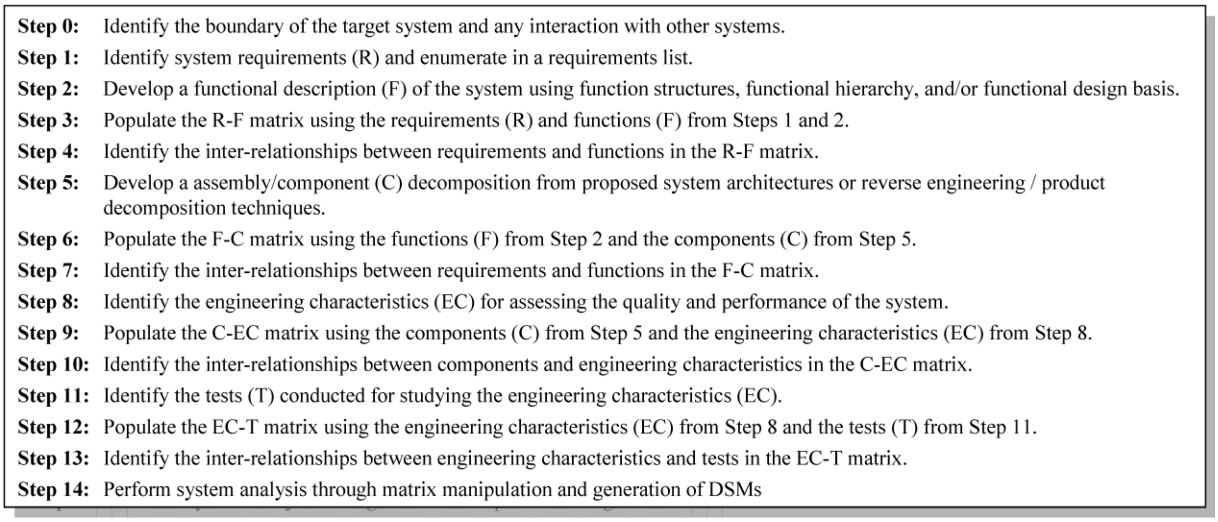
Department of Mechanical Engineering, Clemson University

Keywords: systematic method, constructing DSMs, requirements cascading, perspective driven DSM

1 INTRODUCTION

The steps for constructing Design Structure Matrices (DSMs) include (1) decompose a complex system into subsystem elements that are more easily understood, (2) identify relationships between the elements, and (3) analyze the system using algorithms such as clustering, partitioning, tearing, or banding. The steps do not prescribe what information domains and what types of relationships should be captured during the design process. However, the success or quality of the DSM and corresponding analysis is determined by the accuracy of information for the system being modelled [1]. Thus, it is important to correctly decompose a system and capture the relationships. Additionally, the use of DSMs is limited during design because they are often constructed using existing design documentation and interviews [2].

A systematic method and matrix-based modelling scheme are developed to support the design of complex systems through the identification of 1) repetitive or legacy requirements, 2) integration of functionality into a single component/assembly, 3) fulfilment of requirements and functionality by multiple systems, and 4) eliminating redundant and worst case system testing. The method and modelling scheme capture five information domains and the associated vertical and horizontal relationships. These information domains are system requirements, functions, components, engineering characteristics and tests. The primary contributions in this research are twofold. The first is the formalization of a systematic method for populating and deriving design structure matrices (DSM) of complex engineering systems. Second, a matrix-based modelling scheme that enables linkages between design information to be captured and analyzed is developed. This two part approach builds on existing matrix-based techniques including Axiomatic Design [3], the House of Quality [4] and the Design Structure Matrix (DSM) [5]. The work presented in this paper extends [6, 7] by leveraging DSM modelling techniques and analyses and applying it to automotive systems.



Step 0: Identify the boundary of the target system and any interaction with other systems.
Step 1: Identify system requirements (R) and enumerate in a requirements list.
Step 2: Develop a functional description (F) of the system using function structures, functional hierarchy, and/or functional design basis.
Step 3: Populate the R-F matrix using the requirements (R) and functions (F) from Steps 1 and 2.
Step 4: Identify the inter-relationships between requirements and functions in the R-F matrix.
Step 5: Develop a assembly/component (C) decomposition from proposed system architectures or reverse engineering / product decomposition techniques.
Step 6: Populate the F-C matrix using the functions (F) from Step 2 and the components (C) from Step 5.
Step 7: Identify the inter-relationships between requirements and functions in the F-C matrix.
Step 8: Identify the engineering characteristics (EC) for assessing the quality and performance of the system.
Step 9: Populate the C-EC matrix using the components (C) from Step 5 and the engineering characteristics (EC) from Step 8.
Step 10: Identify the inter-relationships between components and engineering characteristics in the C-EC matrix.
Step 11: Identify the tests (T) conducted for studying the engineering characteristics (EC).
Step 12: Populate the EC-T matrix using the engineering characteristics (EC) from Step 8 and the tests (T) from Step 11.
Step 13: Identify the inter-relationships between engineering characteristics and tests in the EC-T matrix.
Step 14: Perform system analysis through matrix manipulation and generation of DSMs

Figure 1. Steps for systematically constructing DSMs

2 SYSTEMATIC MODELLING METHOD

The systematic method comprises 15 steps for modelling system requirements, functions, components, engineering characteristics, and tests. It is based on several commonly accepted systematic engineering design processes (i.e., [8]). Consequently, the information domains and the sequence in which they are modelled correspond to these design processes (see Figure 1).

3 MATRIX-BASED MODELLING SCHEME

The aforementioned modelling and analysis are enabled through a matrix-based modelling scheme (see Figure 2).

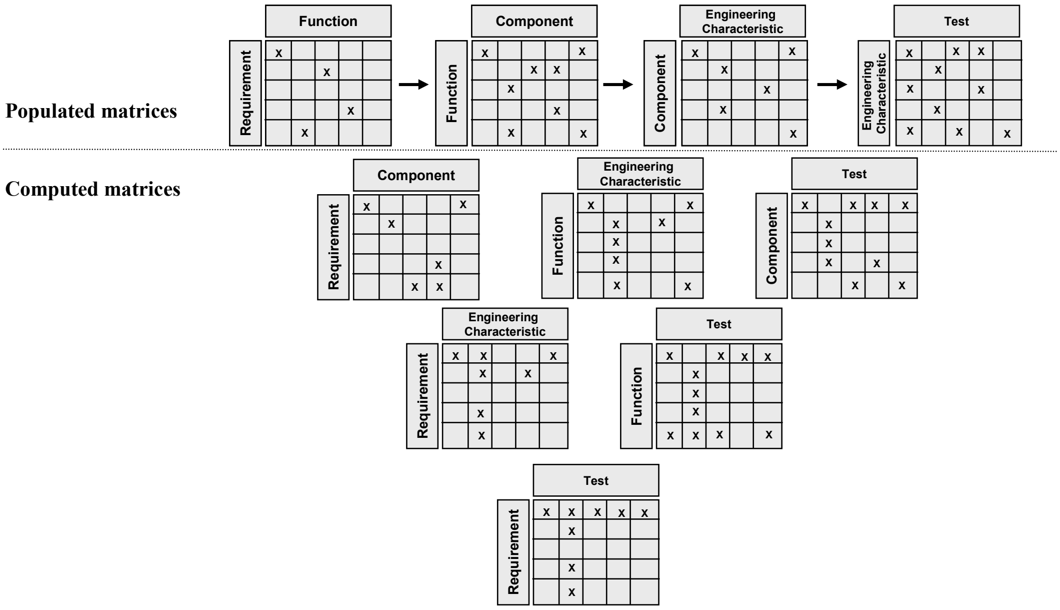


Figure 2. Matrix-based modeling approach

The modelling scheme consists of four populated matrices that enable system requirements, functions, components, engineering characteristics, and tests and associated binary inter-relationships to be modelled and analyzed. First, the mapping between system requirements and functions are captured in the requirements-function (R-F) matrix. Functional and non-functional requirements and requirements traceability can be identified through the R-F matrix. The function-component (F-C) matrix enables functionality to be ascribed to particular components/assembly in the system. The component-engineering characteristic (C-EC) matrix relates components to measurable engineering characteristics. Finally, the relationships between engineering characteristics and tests are described in the EC-T. From these populated matrices, six computed matrices can be derived using matrix multiplication to identify the inter-relationships between non-adjacent domains. In addition it is possible to generate 20 DSMs from each of the matrices through Equations 1 and 2.

$$DSM_i^j = M\left(i,j\right) \times M\left(i,j\right)^T \tag{1}$$

$$DSM_j^i = M\left(i,j\right)^T \times M\left(i,j\right) \tag{2}$$

where DSM_i^j is a DSM that describes the inter-relationships of type j in information domain i, DSM_j^i is a DSM that describes the inter-relationships of type i in information domain j and $M\left(i,j\right)$ is a matrix, either populated or completed between information domains i and j.

Inter-relationships between information domains can be captured from multiple design perspectives. For example, inter-relationships of system requirements can be modelled and studied from a functional perspective (where $M\left(i,j\right)$ is the populated R-F matrix through Equation 1) or from a component perspective (where $M\left(i,j\right)$ is the computed R-C matrix). Similarly, a function-based DSM that captures requirements inter-relationships can be computed using Equation 2 where $M\left(i,j\right)$ is the populated R-F matrix. Of the 20 DSMs that can be computed, some have been discussed in previous literature (e.g., components through function [9], design parameters through requirements [9]).

System analysis is performed by manipulating the matrices using approaches from linear algebra and DSM analyses including multiplication, transposition, summation of rows and columns, sorting, clustering, partitioning, and tearing to visualize and identify areas for design improvement. The analyses include:

- checking consistency between computed and manually populated DSMs
- deriving other matrix-based system representations (i.e., Axiomatic Design, House of Quality)
- discovering non-adjacent system relationships including requirement-to-components, requirements-to-tests, and components-to-tests.
- functional integration of components
- component integration into assemblies or modules
- identification of change propagation and cascading effects

4 ILLUSTRATIVE EXAMPLE

The method and modelling scheme are demonstrated through the characterization and analysis of a vehicle driver's seat subsystem. The example is chosen because it is a representative complex system (i.e., includes a number of components, requirements, and functions), it addresses physical systems and documentation is readily available.

5 CONCLUSIONS

While methods for building DSMs have been proposed [1, 9, 10], they only provide high-level guidance for studying complex systems and are not explicitly related to engineering design processes. The method and matrix based modelling scheme developed in this research and presented in this paper provides an essential linkage between systematic engineering design methods and methods for constructing DSMs. The results and observations for a representative automotive system is discussed and related to existing design literature to build confidence in the approach.

REFERENCES

[1] Yassine, A.A. An Introduction to Modeling and Analyzing Complex Product Development Processes Using the Design Structure Matrix (DSM) Method. *Quaderni di Management* (Italian Management Review), 2004, 9. (*English Translation)*

[2] Dong, Q. and Whitney, D. Designing a Requirement Driven Product Development Process. *Proceedings of the ASME Design Engineering Technical Conferences, Pittsburgh, PA*, 2001.

[3] Suh, N.P. Axiomatic Design Theory for Systems. *Research in Engineering Design*, 1998, 10(4), 189-209.

[4] Hauser, J.R. and D., C. The House of Quality. *Harvard Business Review*, 1988, 3, 63-73.

[5] Steward, D.V. *Systems Analysis and Management; Structure, Strategy and Design*. (Petrocelli Books, Inc., New York, 1981).

[6] Maier, J.R.A., Ezhilan, T., Fadel, G.M., Summers, J.D. and Mocko, G.M. A Hierarchical Requirements Modeling Scheme to Support Engineering Innovation. *International Conference for Engineering Design (ICED)* Paris, France, 2007.

[7] Mocko, G.M., Summers, J.D., Teegavarapu, S., Ezhilan, T., Maier, J.R.A. and Fadel, G.M. A Modelling Scheme for Capturing and Analyzing Multi-Domain Design Information: A Hair Dryer Design. *International Conference for Engineering Design (ICED)* Paris, France, 2007.

[8] Pahl, G. and Beitz, W. *Engineering Design: A Systematic Approach*. (Springer, New York, 1996).

[9] Pimmler, T.U. and Eppinger, S.D. *Integration Analysis of Product Decompositions*. (Alfred P. Sloan School of Management, Massachusetts Institute of Technology, 1994).

[10] Browning, T.R. Using the Design Structure Matrix (DSM) for Process Integration. Lockheed Martin Aeronautics Company, Fort Worth, Texas USA.

Contact: Gregory M. Mocko
Clemson University, Department of Mechanical Engineering
243 Fluor Daniel EIB
Clemson, SC 29634-0921, USA
(864) 656-1812 (office) / 864-656-4435 (fax)
gmocko@clemson.edu

9TH INTERNATIONAL DSM CONFERENCE

A Systematic Method for Modelling and Analysing Conceptual Design Information

Gregory M Mocko, Georges M Fadel,
Joshua D Summers, Jonathan RA Maier, and
Thulasiram Ezhilan

Clemson University
Department of Mechanical Engineering
Clemson, SC USA

Agenda

- Project Overview and Motivation
- DSM Overview and Limitations Addressed
- Overview of Proposed Method
- Matrix-based Modelling Scheme
- Detailed Discussion of Matrices
- System Analysis through Matrix Manipulation
- Generation of DSMs
- Demonstration and Application
- Experimental Scenario 1
- Driver‘s Seat DSMs
- Experimental Scenario 2
- Student Observations
- Summary and Future Work

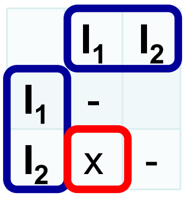
Project Overview & Motivation

- Develop a method to model and analyse cascading requirements from systems to components in vehicle design to
  - identify allocation of vehicle weight based on requirements and functionality
  - trace and capture changes in vehicle system requirements
  - reduce overall vehicle weight

- Existing methods / approaches
  - map non-adjacent domains (e.g., HoQ maps requirements to test measures)
  - capture complex inter-relationships within a domain (e.g., traditional DSMs)
  - lead to avoidable inconsistencies on system models

DSM – Overview

- Overview
  - A method for capturing the inter-relationships between an information domain through an additional information domain
  - Algorithms for analyzing the system model
    - Clustering, tearing, partitioning
  - Has been applied to:
    - Component / Assembly
    - Process / Information
    - Function & flow / Behavior
    - Workflow



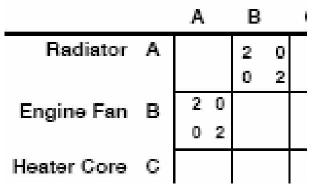
	I_1	I_2
I_1	-	
I_2	X	-

DSM – Discussion of Challenges

- What is the inter-relationship mapping domain?
- Will modelers remain consistent in the mapping domain?
- Is it possible for modelers to change the mapping domain for “large” DSMs
- It is better to take a bottom-up approach for mapping?
- Can “higher-level” inter-relationships be generated?

Current approach

Proposed approach



		A	B
Radiator	A		2 0 0 2
Engine Fan	B	2 0 0 2	
Heater Core	C		

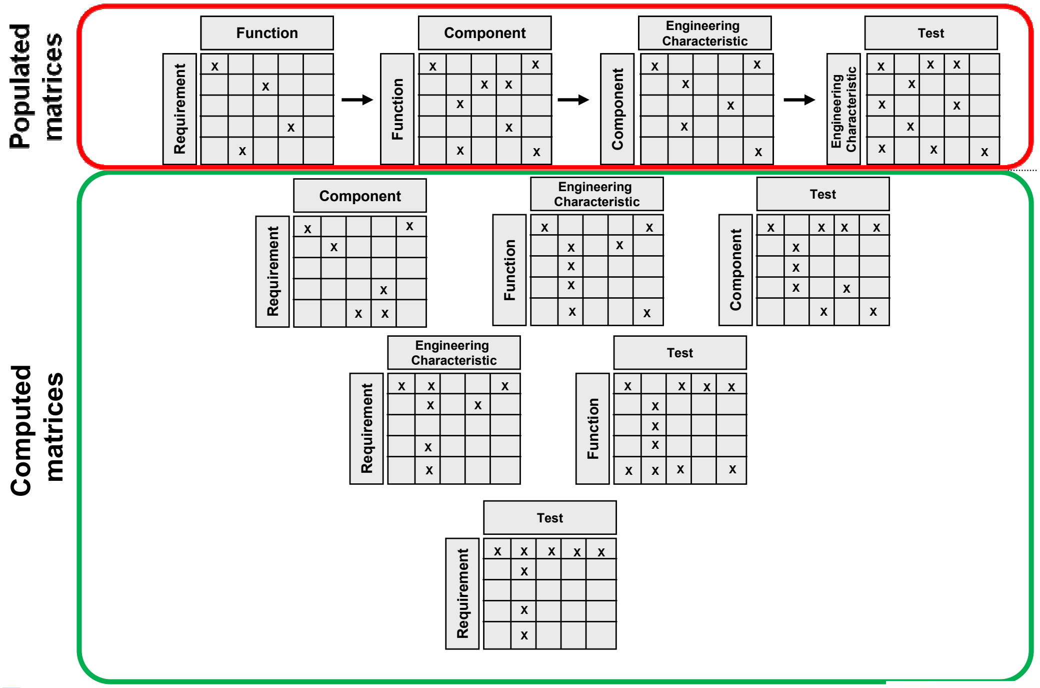
Overview of the Proposed Method

- Underlying assumptions
  - Information flow commonly accepted systematic design methods
  - Adjacent information domains
- Method consist of:
- 14 basic steps
- can be adapted according to nature of problem
  - Original design, reverse engineering & product decomposition
  - Structure and complexity of design team
  - Scope of system
- Information domains captured
- customer requirements,
- functions,
- components, and
- engineering characteristics.
- Utilizes a matrix-based modelling approach – (Multi-domain mapping)

Steps of the Proposed Method

Step 0: Identify system boundary
Step 1: Identify system requirements (R)
Step 2: Develop a functional description (F) of the system
Step 3: Populate the R-F matrix from Steps 1 and 2.
Step 4: Identify the inter-relationships in the R-F matrix.
Step 5: Identify the assembly (A) /component (C) system architecture
Step 6: Populate the F-C matrix from Steps 2 and 5.
Step 7: Identify the inter-relationships in the F-C matrix.
Step 8: Identify the engineering characteristics (EC) for assessing the quality and performance of the system.
Step 9: Populate the C-EC matrix from Steps 5 and 8.
Step 10: Identify the inter-relationships in the C-EC matrix.
Step 11: Identify the tests (T) for studying the engineering characteristics (EC).
Step 12: Populate the EC-T matrix from Steps 8 and 11.
Step 13: Identify the inter-relationships in the EC-T matrix.
Step 14: Perform system analysis

Matrix-Based Modelling Scheme



Populated matrices

	Function				
Requirement	x				
			x		
				x	
		x			

→

	Component				
Function	x				x
			x	x	
		x			
				x	
		x			x

→

	Engineering Characteristic				
Component	x				x
		x			
				x	
		x			
					x

→

	Test				
Engineering Characteristic	x		x	x	
		x			
	x			x	
		x			
	x		x		x

Computed matrices

	Component				
Requirement	x				x
		x			
				x	
			x	x	

	Engineering Characteristic				
Function	x				x
		x		x	
		x			
		x			
		x			x

	Test				
Component	x		x	x	x
		x			
		x			
		x		x	
			x		x

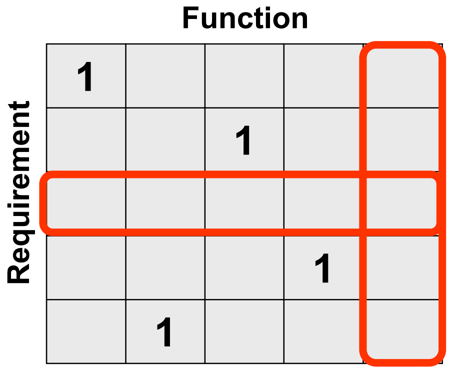
	Engineering Characteristic				
Requirement	x	x			x
		x		x	
		x			
		x			

	Test				
Function	x		x	x	x
		x			
		x			
		x			
	x	x	x		x

	Test				
Requirement	x	x	x	x	x
		x			
		x			
		x			

Requirement-Function (R-F) Model

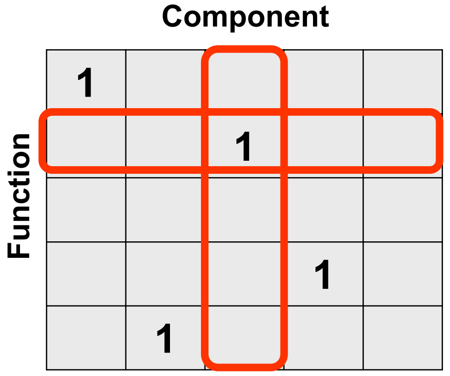
- Requirements and functions are generated through:
  - Product tear down & reverse engineering
  - Standards
  - Existing documentation
- Binary mapping (1/0)
  - Describe existence or non-existence of a relationship
  - Simplify the modelling process by requiring that designers identify if a relationship exists or not
  - *** May be augmented based on requirement weighting
- With the R-F matrix, one can:
  - Visualize how requirements affect functions
  - Identify non-functional requirements
  - Assess criticality of functions



1				
		1		
			1	
	1			

Function-Component (F-C) Model

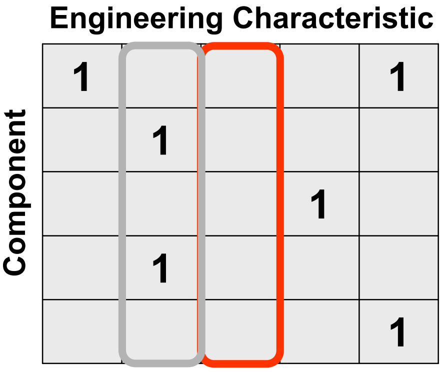
- Utilize the functions identified in R-F
- Product architecture generated through:
  - Product decomposition
  - Reverse engineering
  - Design documentation & model
- Binary mapping (1/0)
  - Functionality of components are described
  - Strength/allocation of functionality not captured
- With the F-C matrix, one can identify:
  - High/low functional components
  - Coupled functionality
  - Non functioning components



1				
		1		
			1	
	1			

Component-Engineering Characteristic (C-EC) Model

- ECs represent the measurable behaviour characteristics
- ECs are generated based on:
  - Test specifications
  - Anticipated component behaviour
  - Standardized testing and government regulations
- Binary mapping (1/0)
  - Performance characteristics related to specific components
- With the C-EC matrix, one can:
  - Summarize system performance
  - Identify quality measures
  - Find components that require extensive testing
  - Identify testing protocols
  - Eliminate unnecessary tests
- ECs are similar to HoQ measures



Engineering Characteristic

Component

1				1
	1			
			1	
	1			
				1

Engineering Characteristic-Test (EC-T) Model

- Tests can be completed through
  - Computational simulations
  - Track / road test
  - Controlled laboratory
- ECs are generated based on:
  - Test specifications
  - Anticipated component behaviour
  - Standardized testing and government regulations
- Binary mapping (1/0)
  - Performance characteristics are mapped to test
- With the EC-T matrix, one can:
  - Identify repeated tests
  - Couple testing protocols
  - Identify ECs that are not verified
  - Eliminate unnecessary tests

Test

Engineering Characteristic

1		1	1	
	1			
1			1	
	1			
1		1		1

System Analysis

- Intra-domain analysis
  - Operation on a single matrix
  - Summation and sorting of row and columns
- Inter-domain analysis
  - Manipulation of two or more matrices through linear algebra
  - Matrix multiplication, transpose
  - Derivation of computed matrices
- Hierarchical analysis
  - Traverse physical / assembly relationships using an additional Components to Assembly Matrix:

	Assembly			
Component	1			1
	1	1		
			1	1
	1	1		
	1		1	
	1	1		

- Derived matrices
  - Requirements-to-Components

$$[\mathbf{R\text{-}C}]=[\mathbf{R-F}]\times[\mathbf{F-C}]$$

  - Requirements-to-Engineering Characteristics

$$[\mathbf{R\text{-}EC}]=[\mathbf{R-F}]\times[\mathbf{F-C}]\times[\mathbf{C-EC}]$$

  - Requirements-to-Assemblies

$$[\mathbf{R\text{-}A}]=[\mathbf{R-C}]\times[\mathbf{C-A}]$$

  - Functions–to-Assemblies

$$[\mathbf{F\text{-}A}]=[\mathbf{F-C}]\times[\mathbf{C-A}]$$

Generation of DSMs & Analysis

- Model the system interaction from different perspectives
- Possible to generate 20 DSMs

$$DSM_i^j = M(i,j)\times M(i,j)^T$$

$$DSM_j^i = M(i,j)^T \times M(i,j)$$

- Example: Study ***component*** interaction through:
  - ***Function***

$$DSM_{Function}^{Component} = \mathbf{F\text{-}C}^T \times \mathbf{F\text{-}C}$$

  - ***Requirement***

$$DSM_{Requirement}^{Component} = \mathbf{R\text{-}C}^T \times \mathbf{R\text{-}C}$$

- Generation & Analyses enable:
  - finer granularity to be modelled
  - consistency checking of DSMs
  - derivation of other matrices (i.e., HoQ)
  - discovering non-adjacent system relationships
  - identification of information change effect
  - tracking between domains
- Observation
  - meaningful normalization of inter-relationships must be identified

Demonstration & Engineering Systems

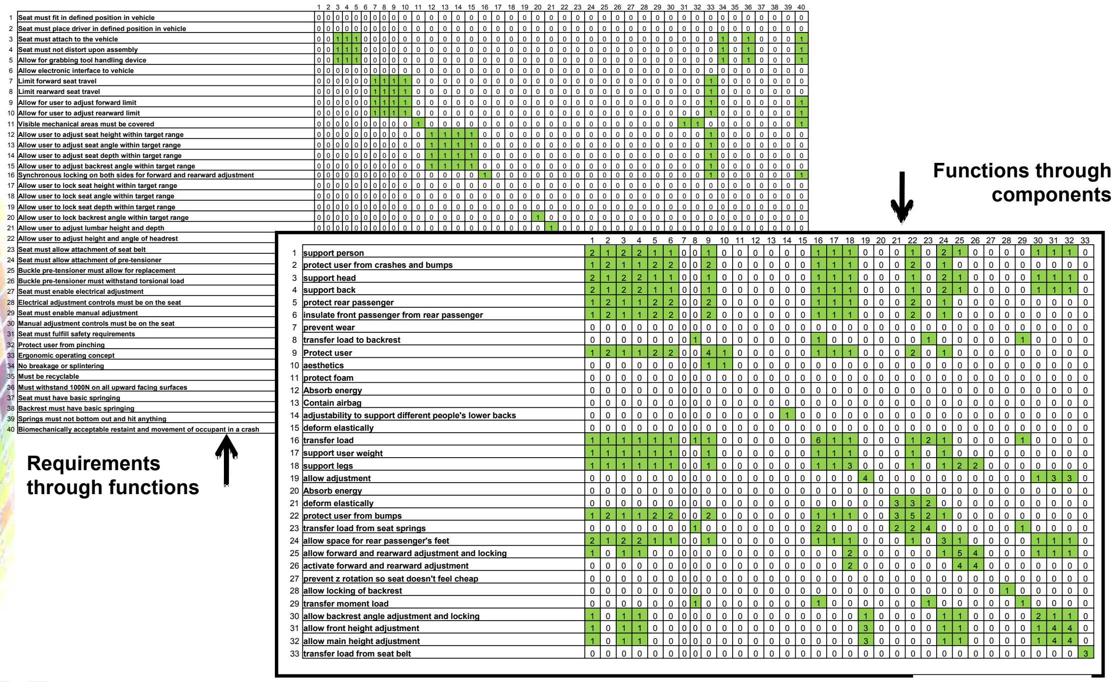
- Pilot Cases
  - Hair dryer
- Vehicle System Cases
  - Accelerator pedal module
  - Cooling system
  - Driver's seat

- System information extracted from OEM specifications
- 40 requirements
  - System and component level
- 33 functions
- 28 system elements
  - 1 assembly
  - 4 sub-assemblies
  - 23 components

Experimental Scenario 1

- 14 step method is followed
- Information domains are identified through existing literature and product tear-down
- Design team – 5 Clemson researchers, 1 industrial sponsor
- ~10 hours to collect and model system and generate solution ideas
- Matrices are manually populated
- DSMs are generated
- System solutions are identified based on matrix analysis

Driver's Seat DSMs



		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	Seat must fit in defined position in vehicle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	Seat must place driver in defined position in vehicle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	Seat must attach to the vehicle	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1
4	Seat must not distort upon assembly	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1
5	Allow for grabbing tool handling device	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1
6	Allow electronic interface to vehicle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	Limit forward seat travel	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
8	Limit rearward seat travel	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
9	Allow for user to adjust forward limit	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
10	Allow for user to adjust rearward limit	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
11	Visible mechanical areas must be covered	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1
12	Allow user to adjust seat height within target range	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
13	Allow user to adjust seat angle within target range	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
14	Allow user to adjust seat depth within target range	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
15	Allow user to adjust backrest angle within target range	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
16	Synchronous locking on both sides for forward and rearward adjustment	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
17	Allow user to lock seat height within target range	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	Allow user to lock seat angle within target range	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	Allow user to lock seat depth within target range	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	Allow user to lock backrest angle within target range	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	Allow user to adjust lumbar height and depth	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	Allow user to adjust height and angle of headrest																																								
23	Seat must allow attachment of seat belt																																								
24	Seat must allow attachment of pre-tensioner																																								
25	Buckle pre-tensioner must allow for replacement																																								
26	Buckle pre-tensioner must withstand torsional load																																								
27	Seat must enable electrical adjustment																																								
28	Electrical adjustment controls must be on the seat																																								
29	Seat must enable manual adjustment																																								
30	Manual adjustment controls must be on the seat																																								
31	Seat must fulfill safety requirements																																								
32	Protect user from pinching																																								
33	Ergonomic operating concept																																								
34	No breakage or splintering																																								
35	Must be recyclable																																								
36	Must withstand 1000N on all upward facing surfaces																																								
37	Seat must have basic springing																																								
38	Backrest must have basic springing																																								
39	Springs must not bottom out and hit anything																																								
40	Biomechanically acceptable restaint and movement of occupant in a crash																																								

Requirements through functions

Functions through components

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
1	support person	2	1	2	2	1	1	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	0	2	1	0	0	0	0	1	1	1	0
2	protect user from crashes and bumps	1	2	1	1	2	2	0	0	2	0	0	0	0	0	0	1	1	1	0	0	0	2	0	1	0	0	0	0	0	0	0	0	0
3	support head	2	1	2	2	1	1	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	0	2	1	0	0	0	0	1	1	1	0
4	support back	2	1	2	2	1	1	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	0	2	1	0	0	0	0	1	1	1	0
5	protect rear passenger	1	2	1	1	2	2	0	0	2	0	0	0	0	0	0	1	1	1	0	0	0	2	0	1	0	0	0	0	0	0	0	0	0
6	insulate front passenger from rear passenger	1	2	1	1	2	2	0	0	2	0	0	0	0	0	0	1	1	1	0	0	0	2	0	1	0	0	0	0	0	0	0	0	0
7	prevent wear	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	transfer load to backrest	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
9	Protect user	1	2	1	1	2	2	0	0	4	1	0	0	0	0	0	1	1	1	0	0	0	2	0	1	0	0	0	0	0	0	0	0	0
10	aesthetics	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	protect foam	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	Absorb energy	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	Contain airbag	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	adjustability to support different people's lower backs	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	deform elastically	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	transfer load	1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	6	1	1	0	0	0	1	2	1	0	0	0	0	1	0	0	0	0
17	support user weight	1	1	1	1	1	1	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
18	support legs	1	1	1	1	1	1	0	0	1	0	0	0	0	0	0	1	1	3	0	0	0	1	0	1	2	2	0	0	0	0	0	0	0
19	allow adjustment	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	1	3	3	0
20	Absorb energy	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	deform elastically	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	2	0	0	0	0	0	0	0	0	0	0
22	protect user from bumps	1	2	1	1	2	2	0	0	2	0	0	0	0	0	0	1	1	1	0	0	3	5	2	1	0	0	0	0	0	0	0	0	0
23	transfer load from seat springs	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	2	0	0	0	0	2	2	4	0	0	0	0	0	1	0	0	0	0
24	allow space for rear passenger's feet	2	1	2	2	1	1	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	0	3	1	0	0	0	0	1	1	1	0
25	allow forward and rearward adjustment and locking	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	1	5	4	0	0	0	1	1	1	0
26	activate forward and rearward adjustment	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	4	4	0	0	0	0	0	0	0
27	prevent z rotation so seat doesn't feel cheap	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	allow locking of backrest	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
29	transfer moment load	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
30	allow backrest angle adjustment and locking	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	2	1	1	0
31	allow front height adjustment	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	1	1	0	0	0	0	1	4	4	0
32	allow main height adjustment	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	1	1	0	0	0	0	1	4	4	0
33	transfer load from seat belt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3

Experimental Scenario 2

- 5 students working individually
- Novice understanding of DSM
- No previous seat design experience
- Domain information provided to students
- 4 DSMs (R→F; F→C; R→C; C→F)
  - Requirements through Functions
  - Requirements through Components
  - Functions through Components
  - Components through Functions
- Objectives
  - Identify the "quality" of the DSMs (Derived vs. Manual)
  - Discover differences in DSMs

Student Observations

- As [DSM]matrix expanded, usefulness decreased[1]
- I changed my answers [mappings] around a bit. … I would be thinking about the relationship between each of the axis's and not the what I should be basing them on.[2]
- Loosing focus of what the mappings represent[4]
- Since it was 40 by 40 matrix it was confusing for me as to which requirement I am relating the other requirements[3]

Difficulty in thinking in two domains at different levels of detail

A single cell in the requirements-to-requirements through function DSM is represented by collapsing 33 functions

---

Summary

- Contributions
  - Matrix based modeling approach
    - that links requirements, functions, assemblies, components, and engineering characteristics
    - follows systematic design methods
  - Systematic method for populating and analyzing/manipulating matrix models
  - Demonstration on several vehicle system & compared with existing literature
- Limitations & Future Research
  - Normalization scheme of matrices
    - Identify the significance of multiplied values
  - Conduct user / protocol study
    - Initial data collected, must formalize experiments for effectiveness
  - Thoroughly investigate hierarchical relationships (C-A matrix)
  - Additional usage applications