SEAri Working Paper Series

Title: Analysis and Applications of Design Structure Matrix, Domain Mapping Matrix, and Engineering System Matrix Frameworks

Paper Number: **WP-2007-8-1**

Revision Date: July 19, 2007

Massachusetts Institute of Technology
Engineering Systems Division

ANALYSIS AND APPLICATIONS OF DESIGN STRUCTURE MATRIX, DOMAIN MAPPING MATRIX, AND ENGINEERING SYSTEM MATRIX FRAMEWORKS

Bartolomei, J.[1], Cokus, M.[2], Dahlgren, J.[3], de Neufville, R.[4], Maldonado, D.[5], Wilds, J.[6]

Abstract

The structure of this paper consists of three parts. Part 1 discusses three different, yet related, modeling frameworks: DSMs, DSM/DMMs, and ESMs. Part 2 examines the existing analysis techniques as applied to the different modeling frameworks. It also includes additional potential analysis techniques proposed for application to the three modeling frameworks. Part 3 provides a more detailed discussion of the application of the analysis techniques describe in Part 2 specifically to the DSM/DMM and ESM methodologies. Finally, conclusions and suggestions for future research are presented.

Part 1: Modeling Frameworks

Modeling frameworks are used to represent the knowledge about a system. Common modeling frameworks include axiomatic design, the Design System Matrix (DSM), system architectural frameworks, Quality Functional Deployment (QFD), and Unified Program Planning. Additionally, two newly conceptualized frameworks include Domain Mapping Matrix (DMM) developed by Browning and Danilovic (2007) and the Engineering System Matrix (ESM) developed by Bartolomei (2007).

The DSM, DMM, and ESM methodologies are closely related. It can be argued that the methods are somewhat evolutionary. Koo (2005) researched types of models engineers utilize to represent complex systems; this research included the Entity-Relationship Modeling (E-R Modeling). The E-R model applies graphical formalism to relations between abstract entities. DSM, DMM, and ESM are all forms of E-R models. The DSM is a single domain matrix of entities and relationships; the DMM is a two-domain matrix; and the ESM is a multi-domain matrix.

Design Structure Matrix (DSM)

The DSM methodology emerged in the early 1980s as scholars demonstrated how graph theory can be used to analyze complex engineering projects. (Steward 1981) Steward showed how the sequence of design tasks could be represented as a network of interactions. The DSM materialized as an nxn adjacency matrix of nodes and relations with identical row and column headings.

[1] Massachusetts Institute of Technology, Engineering Systems Division, jbart@mit.edu
[2] The MITRE Corporation, msc@mitre.org
[3] The MITRE Corporation, Dahlgren@mitre.org
[4] Massachusetts Institute of Technology, Engineering Systems Division, ardent@mit.edu
[5] The MITRE Corporation, dcmaldon@mitre.org
[6] Massachusetts Institute of Technology, Engineering Systems Division, wilds@mit.edu

A DSM can represent relations among components of a product, teams concurrently working on a project, activities or tasks of a process, and/or parameters within the system. In Steward's model, nodes represent individual design tasks, and relations represent information flows, thereby creating a DSM of the activities or process domain. DSMs have also been used to represent and analyze technical artifacts where nodes represent system components DSM (Pimmler and Eppinger 1994; Malmstrom and Malmquist 1998), design and analyze organizations with nodes representing individual members of the team (Eppinger 1997; Eppinger 2001), model the parametric relationships between technical parts (Smith and Eppinger 1997).

Pimmler and Eppinger (1994) suggest that all relations between nodes can be represented within four categories: Spatial, Energy, Information, and Material. A 'Spatial' DSM provides for adjacency or orientation between two elements. 'Energy' DSMs are used when there are needs for energy transfer/exchange between two elements. 'Information' DSMs define data or signal exchanges between two elements, and 'Material' DSMs represent material exchanges between two nodes within the matrix. By selecting the definition of nodes and the category of relations, the DSM can describe different contexts.

Types of DSMs
The nodes and relations differentiate the types of DSMs. According to Browning (2001), there are two main categories of DSMs: static-based and time-based. Each category contains two types of DSMs; component-based DSMs and organizational or team-based DSMs are static, while activity-based DSMs and parameter-based DSMs are time-based. Figure 1 depicts this proposed DSM hierarchy.

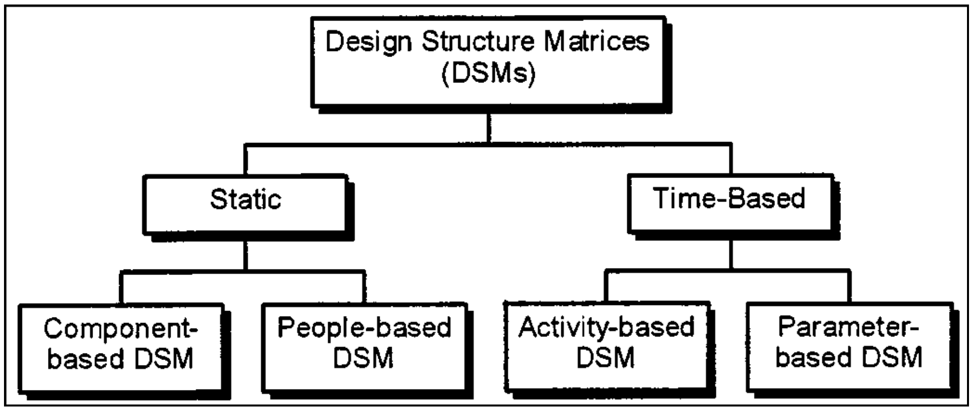


Figure 1 Hierarchy of DSMs (Source: Browning 2001)

Static-Based DSMs
A static-based DSM consists of nodes that are independent of time, ie all nodes exist simultaneously. Thus, the ordering of rows and columns reflects groupings, not time flow. The nodes in static-based DSMs are usually system components or people within organizations. System evolution or change over time may cause a time dependency of the existence or relations of a node within the system, but the nodes themselves remain static. For example, if a team member leaves an organization, he does not cease to exist,

but rather removes all or some relations to the organization. The organization has changed, but the departing team member remains static relative to the organization. This distinction of time-dependency is important when discussing analysis methods later in this paper.

Component-based DSMs are used for modeling system architectures based on components and/or subsystems and the relationships to each other. Pimmler and Eppinger (1994) used component-based DSMs to represent the product decomposition of an automotive climate control system. In this well known example, the nodes are components at the subsystem level, and the interactions are material relations. Other examples of component-based DSMs can be found throughout the literature in a multitude of industries, including building construction, semiconductor, photographic, aerospace, electronics, and telecom industries.

Organizational/Team-based DSMs are used for organizational analysis and design based on information flow among organizational entities. The nodes represent the individual team members or teams within the organization, while the relations represent the required communication flows between the nodes. Eppinger (1997, 2001) has used the team-based DSMs to attempt mapping the organizational structure to effective product architectures and efficient task allocations. In this example, the nodes are product development teams (PDTs) in a project corresponding to subsystems within a new product. The relations indicate the flow of information between the teams, emphasizing the capture of the frequency and direction of the information flow. McCord and Eppinger (1993) provide broad application examples from engine design to laptop development. Browning (1996) and Danilovic (1999) applied this research to the aerospace and automotive industries, and several other organizational DSMs can be found in the literature. Sosa, Eppinger, and Rowles (2004) defined a hierarchical organization DSM including individuals belonging to “Modular Design Teams” and “Integrative Design Teams” that were further decomposed into team groups.

Time-based DSMs
A time-based DSM consists of nodes that are time dependent. Ordering of rows and columns in a time-based DSM corresponds to sequencing or time flow. The interactions between nodes constitute feedforward and feedback interfaces. Time-based DSMs provide directed graph representations. The most common forms of time dependent nodes are activities or tasks within a process.

Activity or tasked-based DSMs are used to depict the dependency of one activity on another. The activities DSM can help identify activities that need to be completed in order for other activities to start. Kusiak and Wang (1993) applied the activities DSM to the automotive design industry. In the example, the nodes are design tasks for automobile design, and the relations represent the interaction (or activity information/material inputs/outputs) between the activities. Park and Cutowsky (1999) used a similar methodology to assess collaborative project management and to create process templates for rapid manufacturing.

A parameter-based DSM is constructed from a "bottom-up" approach to identify the low-level activities that influence the design parameters. The difference between an activities-based DSM and a parameter-based DSM is the level of analysis. The nodes of the parameter-based DSM represent system activities, rather than the process activities as in the activities DSM. For example, the activities-based DSM might include program reviews, documentation requirements, and system/subsystem tests, while the parameter-based DSM might include subroutines of a software algorithm or tasks describing "how" the physical system works. Cesiel (1993) provided an example of the application of a parameter-based DSM to the calibration development for automotive diagnostic systems. Rask and Sunnersjo (1998) defined a parameter-based DSM to represent design variables of a robotic arm and housing, in which the nodes were design variables of system components. For example, the housing was decomposed to outer radius, inner length, shaft radius, inner height, and wall thickness. Then, the relations represented the dependence of one design parameter on another. Parameter-based DSMs are the least documented in the DSM literature, however future applications may be extended upon the development of mathematical algorithms to analyze and incorporate DSMs with system dynamics modeling. This idea will be discussed later in this paper.

Completing a DSM

While completing a DSM appears to include merely inserting an "X" in a box showing a relationship between the row and column elements, the task is somewhat more complicated. The key first step is to determine the purpose of the DSM (i.e. what question(s) the DSM/ESM is intended to help answer) and define the relationship that the DSM represents. Failure to determine this information means that every person contributing to the DSM may be focusing on different purposes and completing the DSM by asking different questions. Context is the universal set of elements that will be considered for the model, as well as the expression of relations. Thus, several system models with different contexts can have same elements, but the expressions of the relations vary across the contexts. Same is true of the perspective—experience, knowledge, and bias contribute to the social element of perspective. For example, two key stakeholders asked to provide inputs regarding the relationships of the nodes may resolve very different models due to the differing perspectives. Dong (1999) suggests that engineers have different mental models of the design and no single actor had the complete picture of the technical system. Therefore, documenting the DSM's represented relationship and purpose, as well as the context, is essential to further analysis using the DSM.

Deciding the purpose of the DSM and determining the question that the DSM is meant to answer helps systems engineers decide if the elements should be completed in a binary manner, using an "X" or a 1 to show coupling, or if the elements should be completed using a numerical or relative ranking. Relative rankings can include H/M/L to indicate a High, Medium, or Low. Another method includes using positive and negative numerical values such as -2, -1, 1, and 2. (Browning 2001) The 2 is used to indicate a high degree of coupling, the 1 a much less degree of coupling, and empty box indicates no coupling. The negative numbers indicate where systems engineers want to ensure no coupling and the relative strength of limiting this coupling. Currently available analysis tools for

DSMs do not work well with negative indicators; therefore, systems engineers find other scales or alternative representations more useful.

The DSM is a flat matrix methodology, meaning that the methodology does not allow clear representation of multiple types of relationships or node attributes. For example, in an organization-based DSM, the nodes may represent stakeholders of a system. The stakeholders may relate to each other by means of communicating information, physical proximity, or financial interactions. If a DSM is constructed only defining the existence of a relationship (ie place an "X" or 1 to designate any existence of a relationship regardless of communication, physical, or financial), the analysis may not provide an appropriate result if the consumer of the information desires funding flows only. Likewise, even if a numeric DSM is constructed, the scales for each relationship may not be similar, and a question of aggregation of all relationships still presents a challenge. Therefore, to ensure quality and clarity of the analysis, multiple relationships may be more appropriately analyzed by constructing multiple DSMs consisting of the same elements or nodes, yet representing independent relationships.

Level of Detail within the DSM
The level of abstraction when defining the nodes will be context-specific depending on the desired output or analysis (Sabbaghian 1998). The DSM provides significant flexibility in the granularity of the entities represented within the matrix. An entity could be a "small", atomic component or a much more complicated component, made up of several sub-components. Highly detailed DSMs based on lower-level components would likely be large and complex, consisting of several rows/columns. Less detailed DSMs tracking higher-level entities would likely have fewer rows/columns, but sacrifice details potentially necessary for system analysis.

Due to the sheer size and complexity, highly detailed DSMs require automated analysis. Even with automation, analyzing very large DSM can represent significant challenges. In addition, gathering the data required to populate a large DSM may be impractical, and often the benefit of analysis results is not realized for the required cost of constructing the DSM. Highly detailed DSMs are best suited for large systems which require detailed analysis of the component interaction. Such a project is more likely to be longer term, requiring resources and cooperation from subject matter experts, and recognition of the benefits of analysis may be better perceived.

Alternatively, low-detail DSMs may be simple enough to allow analysis by hand/inspection, and not require significant costs in terms of time and resources. Although automation would be helpful, analysis is feasible without tools for simplified matrices. Low-detail DSMs are best suited for characterizing components which have known behavior apart from many sub-components, including sub-systems that have a specific functional purpose, such as "card reader".

Representation of organizations is a good example of how different levels of DSMs can be used. At the portfolio or even corporation level a less detailed DSM can help decision-makers determine major areas where groups should or should not collaborate in achieving functions. A medium-level DSM may focus more on the program level, and still a detailed DSM may focus on the project and task level. All of these DSMs can have tremendous value to making the total organization work well, yet as previously discussed, identifying the context of the DSM is critical to informing the assumptions of the resulting analysis.

Summary of DSMs

In summary, the DSM is a matrix consisting of nodes and relations of a single domain. The DSM can be utilized for both the social and technical domains, yet traditionally has not addressed the system interactions with the environment since the matrix is flat. DSM does not allow clear representation of multiple relations or time evolutions, but can be very useful in intra-domain analysis which will be discussed later in this paper.

Domain Mapping Matrix (DMM)

Eppinger, and many other researchers in product development and management research, recognized that the analysis of the single-domain interaction patterns leads to "learning about the particular product development situation and how to improve." Meanwhile, comparing patterns across the domains (multi-domain interactions) allows assessment of "effectiveness of the process and organization to develop the particular product." (Eppinger 2002) Although his framework only included three domains (Product, Process, and Organization), the realization of the importance of the multi-domain interactions has pushed the conceptualization of DSMs.

Building upon the DSM literature, Danilovic and Browning (2007) present a framework that distinguishes the single- and multi- domain interactions using DSM and Domain Mapping Matrices (DMM). The DMM examines the interactions across domains: the rows represent nodes of one domain, while the columns represent nodes of another domain. Unlike the DSM, the DMM is an mxn rectangular matrix since the rows and columns are not identical. By combining both DSM and DMM methodologies, the analysis results are enriched, providing an expanded view of the system.

The early research is largely focused on product development systems, identifying five domains important to the examination of product development projects. These domains include "the goals domain the product (or service, or result) system; the process system (and the work done to get the product system); the system organizing the people into departments, teams, groups, etc.; the system of tools, information technology solutions, and equipment they use to do the work; and the system of goals, objectives, requirements, and constraints pertaining to all the systems." (Danilovic and Browning 2007) Figure 2 depicts a generalized view of the DSM/DMM representation.

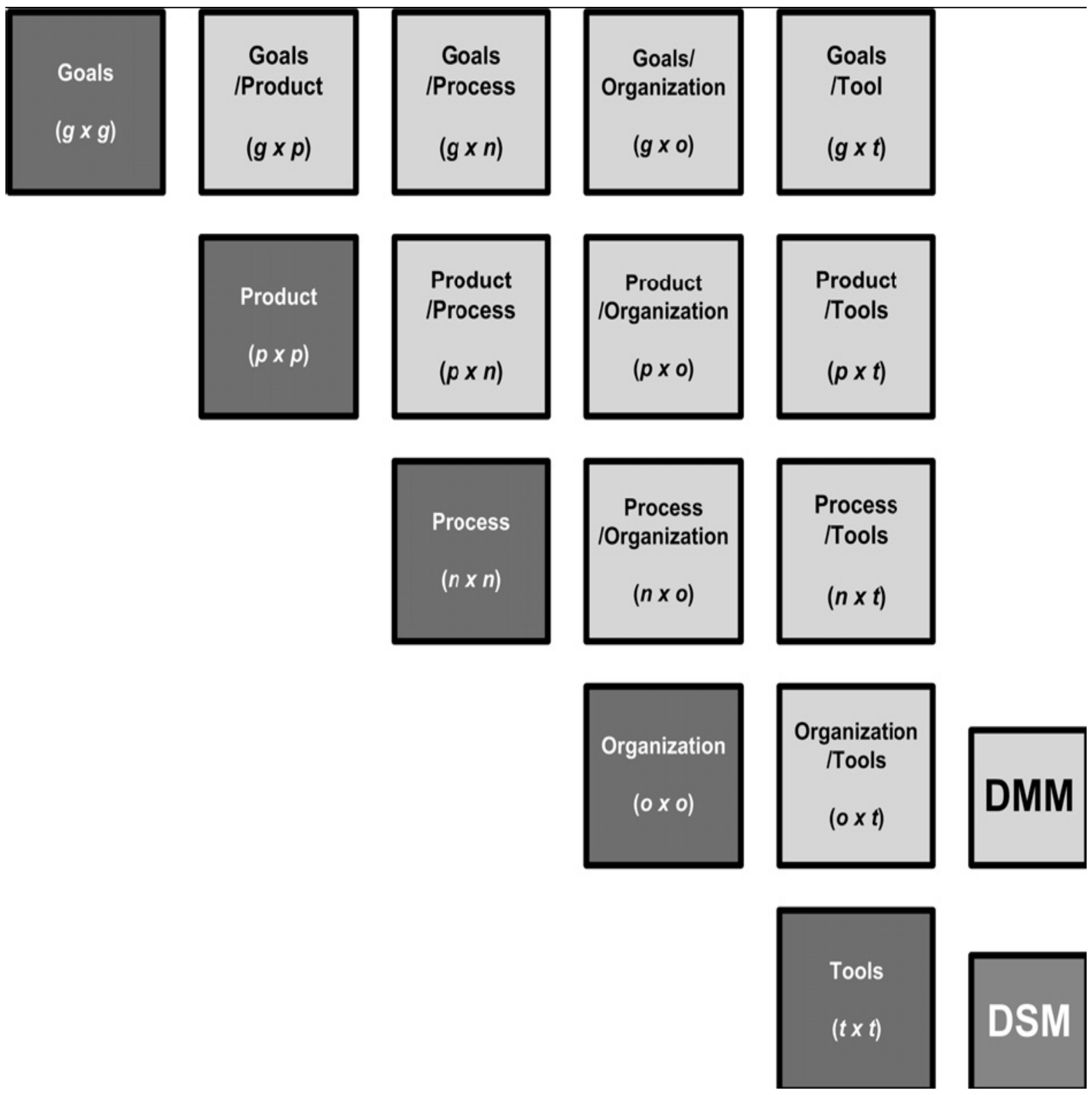


Figure 2 DSM/DMM Framework (Source: Danilovic and Browning 2007)

Each element along the diagonal represents a DSM representing the interactions within each of the five domains. The off-diagonal matrices represent the interactions between domains.

The DSM/DMM representation is similar to the flow-down matrices of QFD. (Danilovic and Browning 2007) However, because QFD representation is not easily manipulated for matrix analysis techniques, it is suggested that DSM/DMM methods help to focus the analysis results on interdependencies, interactions, and exchange of information within and across domains.

This modeling framework is relatively new, and thus the current literature is limited. Only a small sample of examples has been documented. These examples include a military aircraft product system mapping the physical products to functionalities and technologies, a manufacturing company assessment of the business portfolio mapping the physical product system and the organization, and a dynamic analysis of a multi-project structure across the product and organizational domains. (Danilovic and Browning 2007) While the DSM/DMM framework is newly conceptualized, researchers have been interested in multi-domain relations for many years. An example of multi-domain modeling not referring specifically to DSM/DMMs can be found in a study by Morelli, Eppinger, and Gulati (1995). The research attempted to improve the modeling and prediction of the technical communications (tasks) of the organization. Whereas most models are developed assuming that the tasks and personnel are mapped one-for-one, this case study was not constrained by this assumption.

As expected, the DMM is constructed in the same procedure as a DSM; after all a DMM is a variant of combining two DSMs. Similarly to the DSM methodology, the DSM/DMM framework lacks the capacity to analyze multiple relationships between single node pairs and express time. However, the DSM/DMM methodology provides significant benefits over the DSM framework by expanding the consideration beyond single domain information.

Engineering System Matrix (ESM)

The ESM methodology extends the DSM and DMM methods to include multiple domains, multiple relations, and changes over time. Bartolomei (2007) developed the ESM in response to the limitations of existing modeling frameworks to sufficiently represent the environmental interactions and influences of time to provide a more holistic representation of the system. The methodology reaches beyond the physical, social, and process domains to include the system drivers, node attributes, and system evolution.

Within the ESM methodology, there are six domains (environmental or system drivers, social or stakeholders, functional including objectives and functions, physical or objects, and process or activities) that are important to describe the engineering system. The ESM organizes this information using a matrix structure that can be used to facilitate network and graph theoretic analysis. The derived analysis consists of varying classes of nodes, relations, and attributes. Nodes represent different classes of objects, relations describe interactions between two nodes, and attributes generically describe the parameters and descriptions for both nodes and relations. The conceptualization is both a hyper graph and a multi graph. “A hyper graph implies the graph contains different classes of nodes and there are interactions between nodes of different types. A multi graph implies multiple edges can exist between nodes. For example, two human actors might have a financial relationship and communication relationship between them. In addition, the ESM is a designed to represent how the graph (nodes, relations, and attributes) changes over time.” (Bartolomei 2007)

The ESM is an adjacency matrix with identical row and column headings, where the diagonal cells represent the system elements and the off-diagonal cells represent the relationships between elements. The grey cell blocks along the diagonal represent a graph of a particular class of nodes. The off-diagonal blocks of cells represent a multi-partite graph that relates two classes of nodes. Figure 3 displays a generalized ESM.

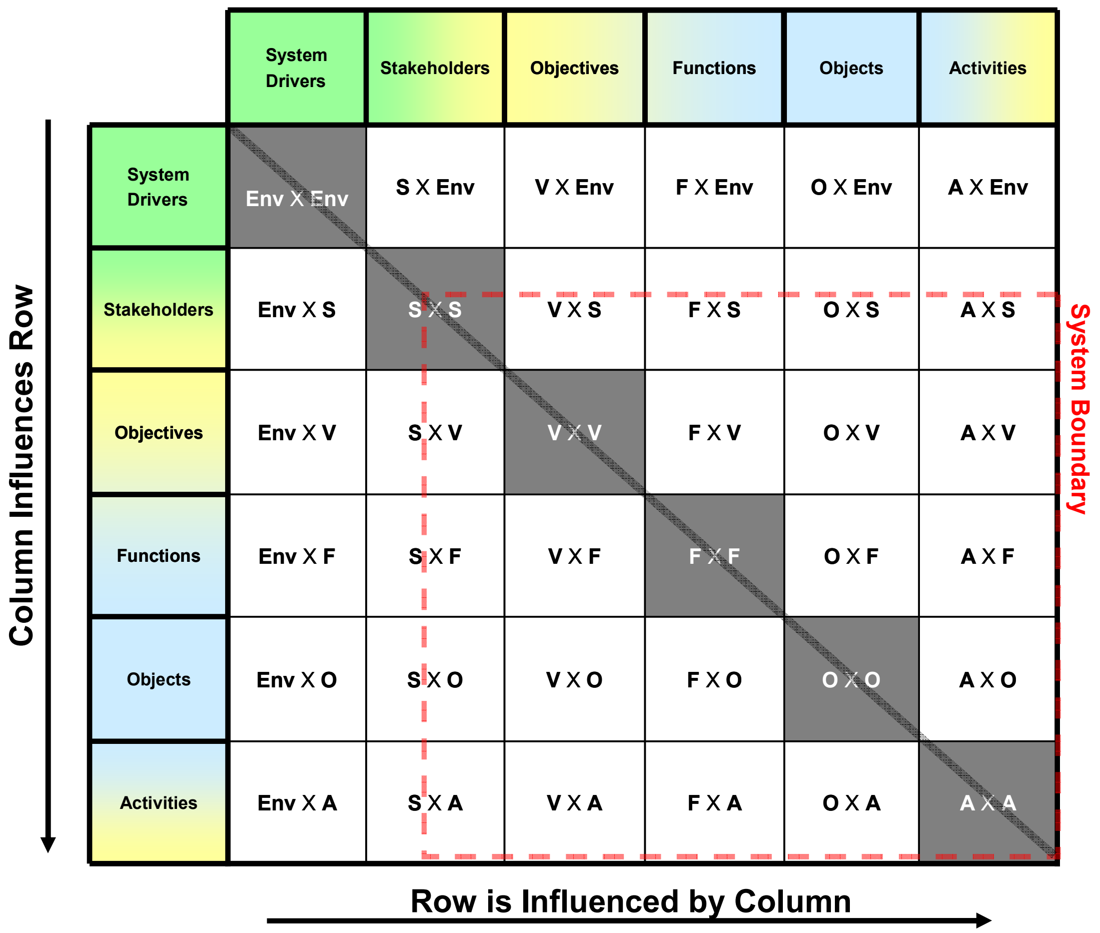


Figure 3 The Engineering System Matrix (Source: Bartolomei 2007)

As previously stated, the ESM includes six domains. These domains are represented in the ESM as System Drivers, Stakeholders, Objectives, Functions, Objects, and Activities. (Bartolomei 2007)

- **System Drivers** represent the non-human components that affect or are affected by the engineering system that are beyond the control of the system's human components.
- **Stakeholders** represent the social network of the system and consist of the human components that affect or are affected by the system (including organizations).
- **Objectives** represent the objectives, goals, and purposes of the engineering system.
- **Functions** represent the functional architecture of that system.
- **Objects** represent the physical, non-human components of the system that act or are acted upon.
- **Activities** represent the processes, sub-processes, and tasks performed by the system.

Each domain is populated with nodes and relations similar to the DSM and DMM methodologies. However, in the ESM nodes and relations in the system can be described with attributes. Attributes define the characteristics for each particular node or relation.

"Attributes can be binary, string, numeric or a mathematical function" as demonstrated by Bartolomei (2007) in the development of the Qualitative Knowledge Construction tool. Furthermore, the ESM can store multiple relations in the same matrix, i.e. the matrix is not flat like a DSM or DMM.

To incorporate the influences of time, each node and relation has at least one attribute known as existence. The representation of time in the system can be represented as a binary attribute for each node that defines whether a node or relation existed (1) or did not exist (0) for a particular time interval. For example, over the lifecycle of a system organizational changes result in the addition and removal of stakeholders over time. In some cases, stakeholders that may have existed in the beginning may have departed and later returned. Therefore, each node may have several existence attributes to describe periods of involvement. In addition, "particular attributes of nodes may change over time." (Bartolomei 2007) Component costs for particular products could be treated as independent variables that change periodically. Thus, discrete price changes for different time intervals can be captured in the system model. "Yet other attributes might be continuous, time dependent functions." (Bartolomei 2007) For example, uncertainty of stakeholder promotion, or rotation in or out of the system, may be a time-dependent function relating to the length of time he/she has served on the project. The ESM framework is designed to include these representations of time.

In summary, the ESM provides a more complete modeling framework as compared to other modeling frameworks. As expected, the ESM is constructed in a similar procedure as a DSM and DMM, except that the ESM allows the use of attributes to more clearly describe the system. The ESM allows the modeler the ability to represent each domain of the engineering system and maps interactions within and across these domains. Finally, the ESM allows the modeler the ability to manage time, parameterize the nodes and relations by defining multiple attributes, and enumerate values of uncertainty for the quality of information and/or the uncertainty of the elements of the system.

Part 2: Existing Analysis Techniques and Applications

This section reviews current analysis techniques that have been applied to DSMs and ESMs, including classical DSM techniques, sensitivity analysis techniques, network-based analysis techniques, coupling techniques, and real options techniques. Example applications are presented where available.

Classical DSM Techniques

A variety of techniques were developed to analyze information presented in DSMs. Table 1 shows the types of DSMs with the respective applications and classical DSM techniques that can be used for each.

Table 1 Classical DSM Techniques for Types of DSMs

DSM Data Types	Representation	Application	Analysis Method
Task-based	Task/Activity input/output relationships	Project scheduling, activity sequencing, cycle time reduction	Partitioning, Tearing, Banding
Parameter-based	Parameter decision points and necessary precedents	Low level activity sequencing and process construction	Partitioning, Tearing, Banding
Team-based	Multi-team interface characteristics	Organizational design, interface management, team integration	Clustering
Component-based	Multi-component relationships	System architecting, engineering and design	Clustering

Partitioning/Sequencing

This analytical technique is designed to reorder system components with time-based dependencies. The sequencing algorithm manipulates the rows and columns simultaneously to produce a lower triangular matrix. Reordering rows and columns such that the sub-diagonal ticks move closer to the diagonal maximizes the feed-forward flow of information and materials, while simultaneously minimizing possible inefficiencies caused by feedback and rework. The technique does not alter the relationships recorded in the DSM, but merely re-orders the rows and columns to eliminate apparent feedback loops.

In the results of sequencing, the ticks below the diagonal represent a precedence relationship between tasks, meaning the row node requires input in the form of information or material from the column node. Contrastingly, the ticks above the diagonal represent feedback relationships, meaning the column node requires input from the row node. This type of relation indicates where in the process assumptions about downstream activities must be made.

Minimizing feedback eliminates process iteration. Researchers applying DSMs to control systems historically have hypothesized that a system with feedback is inherently unstable. While this hypothesis is likely accurate for this specific application, many systems rely on process iteration. This leads to the question of whether automated tools that partition DSMs should only focus on eliminating feedback loops. Users must be very careful that the tools do not eliminate valuable information.

Managers and operations staffs desiring to improve process or streamline work tasks have applied the sequencing algorithm to examine strategies for process design. (Eppinger, 1990, Steward, 1981) Furthermore, Danilovic and Browning (2007) provide a sample analysis for DSM sequencing for a product development system in which the program milestones are ordered. The result emphasizes the need for assumptions required in task upstream from the information delivery. Because these assumptions require inefficient iterations, it is important for system engineers and managers to recognize where feedback is necessary to assess potential impacts and risks. (Danilovic and Browning 2007) This result is an advantage of the DSM technique over other program scheduling tools.

Clustering
Matrix clustering is a valuable technique for examining the structure of a system. Clustering is similar to sequencing; the technique applies graph theoretic cluster algorithms to reorder the rows and columns of the matrix by grouping highly related nodes, called clusters. By grouping nodes with high interaction into clusters, engineers and managers can more easily identify and examine interfaces between the clusters.

In the result of clustering analysis, the clusters contain most, if not all, of interactions (i.e. ticks) internally and the interactions between clusters is eliminated or minimized. (Fernandez 1998; Sharman and Yassine 2004; Yu et al. 2003) However, interactions between clusters may also be advantageous when considered in the system context. For example, engineers and program managers may desire interaction between two teams to share resources. Additionally, past discussions on clustering that discourage the overlapping of components into two clusters may not make sense for a tangible system or for an organization. This argument may have been based on a purely component based DSM, especially in binary DSM analysis. For example in an organizational DSM, the overlap may represent a person that participates in each group, ensuring the sharing of essential information.

One application of clustering analysis to the physical domain (i.e. system components) is to present subsets of components as candidates for modularization. The clustering can show which components can be segmented into larger modules or platforms, thus enabling future designers to merely adopt that module. Other research suggests that the interdependent components highlighted in a cluster may best be treated as a single, higher-level subsystem in the physical design. DSM clustering can also be applied to the social domain (i.e. stakeholders). Sosa, Eppinger and Rowles (2004) analyzed team organization, resulting in clusters of highly interacting teams and individuals with minimal inter-cluster interactions. They concluded that the groupings represent a useful framework for organizational design by focusing on the predicted communication needs of different players. MITRE Corporation recently used clustering to analyze the interdependence of several Air Force Missions. The result of this analysis shows that almost all missions are interdependent, however the analysis also indicated that one mission, Combat Search and Rescue (CSAR) did not depend on other missions for successful completion, yet was required for almost all missions.

Recently, a possible new application for clustering has presented itself. A prevalent item in systems today is called a Line Replaceable Unit (LRU). An LRU represents the lowest level of component that is worked on by maintainers in field organizations. In an effort to minimize the challenges of field maintenance, many LRUs are now major subsystems. While this might minimize the challenges of field maintenance, it can often force organizations to procure spare parts that take up much space and are very costly to ship back to the depot maintenance units. A clustered DSM can reveal the logical LRUs. Viewing some clustered DSMs also show that there can be subclusters within a major cluster. These subclusters can represent smaller LRUs that should be considered to develop an optimal grouping of LRUs. For example, the grouping may indicate that certain cards in an electronic system are a subcluster. Reliability and Maintainability experts can then analyze each subcluster to determine an optimal sparing concept, thereby minimizing the amount of equipment transferred to field units and also minimizing the equipment that must be returned to the depot for repair. Additionally, if the system is properly designed to facilitate the removal and replacement of the smaller LRUs, the major system should be operational a greater amount of time. These smaller LRUs may also facilitate more focused development of Built In Test (BIT) capabilities that enable the ready identification of the smaller LRU for removal and replacement.

Banding

Banding adds alternating light and dark shading to a DSM to show independent nodes or groups of nodes. (Grose 1994) Developed primarily for analyzing time-based DSMs, banding identifies activities that can be executed simultaneously or independent of one another. Bands represent the critical path to the project, where one node in each band is a potential bottleneck. (Browning 2001)

As in partitioning and clustering, banding manipulates the DSM without altering component relationships. The algorithm is similar to that of partitioning; however the feedback ticks are ignored in the process of determining the bands. The DSM Tutorial[7] provides an example of banding an activities-based DSM.

Banding, although typically associated with task-oriented DSMs, can also be employed in DSMs concerned with system components. Rather than identifying groups of interdependent components (as in clustering), banding identifies groups of independent components. Banding can be used to visually indicate groups of system components which do not affect one another. This can be very useful information in that designers can eliminate areas where they might have previously wasted time considering interfaces. This analysis may also prove useful in system testing. A component in a given band can be tested in isolation from other components within the same band. Particularly advantageous in testing large systems, banding may indicate opportunities to test components in parallel, resulting in significant time and cost savings for test phases.

Banding may also potentially be used on organizational DSMs containing data for information exchanges within an existing organization. The banded DSM may identify which people or subgroups do not interface. Management can then compare that information to a clustered DSM that shows which people or subgroups should exchange information and rectify the disparity.

[7] DSM Tutorial is a composition of many sources and can be found at http://www.dsmweb.org

Tearing

Tearing focuses on identifying various feedback marks that if removed from the matrix, and the matrix is repartitioned, will obtain a DSM with all marks in the lower triangle. Removing these marks is what is known as "tears". Determining the tears to make involves making assumptions about what items have the least impact to the design process. Tears are generally done in two ways: (Browning 2001)

- Minimal Number of Tears: this concept acknowledges that tears represent a guess/assumption, and as such designers should minimize their guesses.
- Confine Tears to the Smallest Blocks on Diagonal: this concept focuses on that if there are to be small iterations within larger iterations (blocks within blocks), then designers want to limit the inner iterations to a small number of tasks.

The concept of tearing is based on guessing and assumptions. While there was likely a good reason to develop this technique, these authors feel tearing should not be done unless absolutely necessary. Few examples exist in the literature. Tearing results in the loss of information. Possibly tearing may support a greater speed to the design process, but the cost is a loss of information. Engineers performing such tears should have a clear understanding of the cost/benefit ratio to this process.

Sensitivity Analysis Technique

Pairing classical DSM techniques with sensitivity calculations can potentially identify system elements that are sensitive to change; this is known as the Sensitivity DSM technique. (Kalligeros 2006) The technique uses a DSM containing information about the uncertainty of the impact due to the occurrence of a change event. Then, partitioning or clustering techniques, selected depending on the appropriate domain, is performed to determine groupings of interactions. "This analysis can result in platforming implications for groupings least sensitive to change impacts." (Bartolomei 2007)

For example, Kalligeros (2006) demonstrated this technique on a large engineered system, an off-shore oil drilling platform. He constructed a components-based DSM, where interactions represented whether a 100% change in one subsystem would cause at least a 20% change in the related subsystem. Then, clustering analysis was used to determine possible platforms for oil company standardizations, providing a significant benefit to energy companies that previously designed mobile oil and natural gas platforms for a single design point based on expectations for a specific product and flow rate. Kalligeros (2006) also analyzed a functional DSM (where the nodes were functional requirements), thereby showing what functional requirements might have driven that component-level interactions. More generally, engineers and managers can use the Sensitivity DSM technique to identify system elements that can be standardized across designs, allowing faster and more economical designs and savings in lifecycle costs.

Network-Based Analysis Techniques

Systems represented as a large network allow calculation of a variety of network metrics generated by the social network community, such as betweeness, path length, and centrality. (Bartolomei 2007) Betweeness is a measure of the number of times a vertex occurs on a geodesic (the short path connecting two vertices). Centrality is a measure of the connectedness of each node. And path length refers to the distance between pairs of nodes in the network.

Bartolomei (2007) provides an example of analyzing the betweeness of the stakeholders in the social domain. This measure is associated with the control of information. Thus, "stakeholders with higher betweeness have greater influence on a social network when compared with stakeholders with lower betweeness." (Bartolomei 2007) Furthermore, Bartolomei showed how redundancy of stakeholder interactions can be useful within an organization. "Two stakeholders maintaining the same interactions amongst other system stakeholders provide a shadowing effect. If one of the two stakeholders should leave the system, the remaining stakeholder maintains the continuity of interactions." (Bartolomei 2007) Additionally, an example of degree centrality analysis provided insights into identifying key system elements (stakeholders, components, and activities) that are highly connected within the system. Degree centrality is associated with power or importance. This information may assist the system engineers and managers in recognizing critical nodes to be carefully monitored.

Future research is necessary to develop theories for multi-domain network analysis, analytical methods for analyzing dynamic networks, new metrics for engineering systems, and methods for comparing networks. Engineers should look for patterns across systems in hopes of "developing new theories and better heuristics that describe and explain the structure and behavior of engineering systems," rather than just considering the social domain. (Bartolomei 2007)

Coupling Techniques

Coupling between system components can have a significant impact on the ability to make future changes to any single component since that may drive a requirement to make changes to other coupled components. (Dahlgren 2007) The requirement to make such changes is often strongly correlated to the tightness of coupling. At one time engineers worked to have systems tightly integrated, which became synonymous with tightly coupled. This may have been driven by the trend to design systems to a point solution, given specific requirement and expected funding. Research into some historical systems has shown that as subsystems become less tightly coupled to each other, the subsystems can evolve at independent rates. Systems engineers and managers can take advantage of this evolution if recognized by quantifying degrees of coupling.

The initial work on quantifying degrees of coupling was derived from Barabasi (2003) and Atkinson and Moffatt (2005). Here, a Coupling Coefficient (CC) is the ratio of tight connections divided by possible number of tight connections for a team. (Dahlgren 2007) A possible hardware analogy is to review the DSM and determine the number of tight and medium couplings between subsystems. Each coupling should be reviewed from a standpoint of every level or layer that the systems can be tightly coupled. For example, in a computer and networking situation, showing that two systems are tightly coupled is necessary but not sufficient. In this case the coupling should be reviewed at each layer of the Operational System Interconnect (OSI) stack. It is possible that each pair of tightly coupled subsystems could also be tightly coupled at a number of layers represented by the OSI, and then the degree of coupling is much larger, and probably problematic to designers, than what was originally anticipated by viewing the DSM.

Additionally, it is important to distinguish whether systems are built to standards or built to the prevailing Commercial Off-The-Shelf (COTS) system when determining degrees of coupling. (Dahlgren 2007) While the use of COTS in government systems was meant to provide an option (a right but not an obligation) to use commercially developed products produced under commercial Research & Development efforts, the unfortunate reality has become that as some COTS products have come to dominate the industry, then the use and upgrade of these products become an obligation and thus is no longer an option. Each coupling identified in the DSM should be evaluated according to whether COTS products are used and the degree of coupling.

In an effort to quantify degrees of coupling, and then to be able to quantify whether systems are tightly or loosely coupled, the following formula was developed. (Dahlgren 2007) These formulas attempt to take into account the couplings identified in the DSMs, the possible considerations for each level of the OSI stack, and the coupling to COTS. These formulas are not all inclusive. Systems not related to Information Technology may not need to be evaluated according to the OSI stack. Systems Engineers should attempt to determine if a modified factor needs to replace the OSI factors and the COTS factor.

$$\text{Coupling Coefficient} = \sum_{i=1}^{m} OSI_i \left[\sum_{\substack{j=1 \\ i=1}}^{n} S_i S_j \right] \Big/ \text{number of possible tight couplings}$$

$$\text{COTS Portion of Coupling Coefficient} = \sum_{\substack{j=1 \\ i=1}}^{n} S_{(COTS)i} S_{(COTS)j} / \text{Years to upgrade}$$

When subsystems are found to be tightly coupled, engineers need to determine if new interfaces should be inserted such that the subsystems are tightly coupled to the interfaces, yet can still evolve separately. In some cases, those interfaces may become system standards for future designs. Theoretically, if each system remains coupled to the standard, then they can be loosely coupled to each other and evolve separately at their own clock speed. Systems engineers and program managers should attempt to include standards and show that coupling in the DSM/ESM framework. A judicious use of standards and interfaces can greatly lower the degree of coupling on a major system. Unfortunately commercial vendors can attempt to "tweak" the implementation of standards to form what might be considered a "semi-proprietary" solution, leading to a developer-forced proprietary solution and tight coupling for most customers and applications.

Coupling can also relate to processes and organizations. For instance, DSM techniques can be applied to the original design of an organization or to the redesign of an organization (Dahlgren and Cokus 2007). The type of information provided in an organizational DSM can also aid designers to determine the degree of coupling that is required between individuals performing the functions. For instance, in a small world network an organization has a mix of tight and loose couples. Those people requiring tight couples may need to communicate or interact frequently. Those people may also need to be collocated together to improve efficiency and effectiveness. Furthermore, by looking across social and technical domains, a DSM that shows the linkage between people on a project and between tasks may help engineers and managers improve understanding of the organization evolution. The clustering coefficient (CC) is the organizational equivalent of the coupling coefficient for component DSMs: CC = # tight connections/# possible tight connections (Barabasi 2003). Too many tight connections will lead to a task or project that fails to utilize long-reach connections to leverage work done by other organizations (Dahlgren 2007).

MITRE recently applied coupling techniques to analyze a components-based DSM for VISA International. (Cokus and Dahlgren 2007) MITRE used high level DSMs in the analysis of historical changes of the four major subsystems of VISA International: credit card, card information reader, transmission system, and data base. The case study visually demonstrated the system change over time and how VISA International's subsystems became more loosely coupled. Interactions were ranked as High, Medium, or Low to show historical system changes. As the subsystems became more loosely coupled, the overall VISA International system became more efficient to operate and easier for the customers to use (Cokus and Dahlgren 2007). Transactions were completed much faster and likely with much greater accuracy, which helped make credit cards a useful tool.

Real Options

To better manage the uncertainties surrounding engineering systems, engineers are devising new methods to designing systems that are flexible. "One of the challenges for designers is to identify where in the system to lay in flexibility, or real options, that allow systems designers and managers to easily change the system in order to maximize benefit and minimize cost." (Bartolomei 2007) Strategies and methods for valuing flexibility are well documented in real options literature; however few have focused on how to screen a system to identify the best opportunities for options, or the "hot" spots in the design.

Real Options Analysis (ROA) can be applied "on" a system or "in" a system. (Wilds and Bartolomei, et al 2007) When analyzing options "on" a system, flexibility is external to the physical design. Alternatively, real options analysis "in" a system requires the flexible option be internal to the physical design. A real option "in" requires deep knowledge about the structure and behavior of the technical system.

ROA tools can be used by system designers, manufacturers, and consumers alike to make informed decisions about the value of adding flexibility to the system at various stages of development. This is most useful early in the development process when opportunities for flexibility are more available to developers. For example, Wilds and Bartolomei, et al (2007) value the predicted options for designing a Micro Air Vehicle. An option to alter the physical design to provide increased performance was valued from the perspective of a product developer. The research applied decision analysis and the lattice method of real options. The results showed that designing for the flexibility to modify the system to respond to changing performance requirements could be a valuable investment early in the design.

One of the challenges for designers is to identify where in the system to lay in flexibility, or real options, that allow systems designers and managers to easily change the system in order to maximize benefit and minimize cost. Bartolomei (2007) attempted to develop a technique to identify hot and cold spots in a system. Hot spots are those spots that are expected to frequently change, possibly due to technological innovation. Cold spots are those areas that are not expected to change. Below is a proposed approach using the ESM that incorporates and extends both the Sensitivity DSM and change propagation techniques. (Kalligeros 2006; Suh 2005)

Proposed Technique for Identifying Real Options
(Souce: Bartolomei 2007)

1. Construct ESM of a particular system
2. Identify sources of uncertainty driving change
3. Define change scenarios
4. Identify change modes for each scenario (E.g. Suh's change propagation method)
5. Calculate how change modes affect objectives for each scenario (e.g. Kallegeros' Sensitivity DSM.
6. Calculate the cost of change for each scenario (e.g. Suh's cost analysis)
7. Identify Hot/Cold Spots for each scenario
8. Examine Hot/Cold spots across scenarios
9. Value flexibility using Real Options Analysis

Most research has focused on hot spots; however customers have shown interest in identifying cold spots that must "support" the frequent change of hot spots. One possible strategy of support would be to over-design the cold spots to facilitate hot spot changes that might require a greater engineering margin. Bartolomei (2007) also brought out the possible use of ESMs to model the system changes due to changing Concepts of Operations, which is often common for a new system or new technology application.

Real Options relate not only to tangible hardware systems, but also to processes and organizations. "For 'hot/cold' spot analysis there are several advantages of representing each domain and the corresponding interactions between domains. For example, the ESM provides a richer picture for how changes propagate across domains (eg highlight how changes in the technical domain affect the process domain and social domains) and the identification of exogenous sources of uncertainties that might each of the domains by constructing the systems drivers matrix." (Bartolomei 2007)

Part 3: Applications of Analysis Techniques Across Multiple Domains

To review, the DSM is an nxn, square matrix containing nodes and relations within a single domain. The DMM is an mxn rectangular matrix containing nodes and relations across two domains, where the rows represent one domain and the columns represent another domain. (Browning and Danilovic 2007) Furthermore, the ESM is an mxn rectangular matrix consisting of nodes and relations, each having multiple attributes, across multiple domains. (Bartolomei 2007) Multi-domain analysis has many challenges, and the techniques described in Part 2 may not be applicable, or sufficient, to provide valid results. This section will consider the following questions:

- Can the DMM/ESM be organized or manipulated to apply the analysis techniques presented in Part 2 due to technique assumptions?

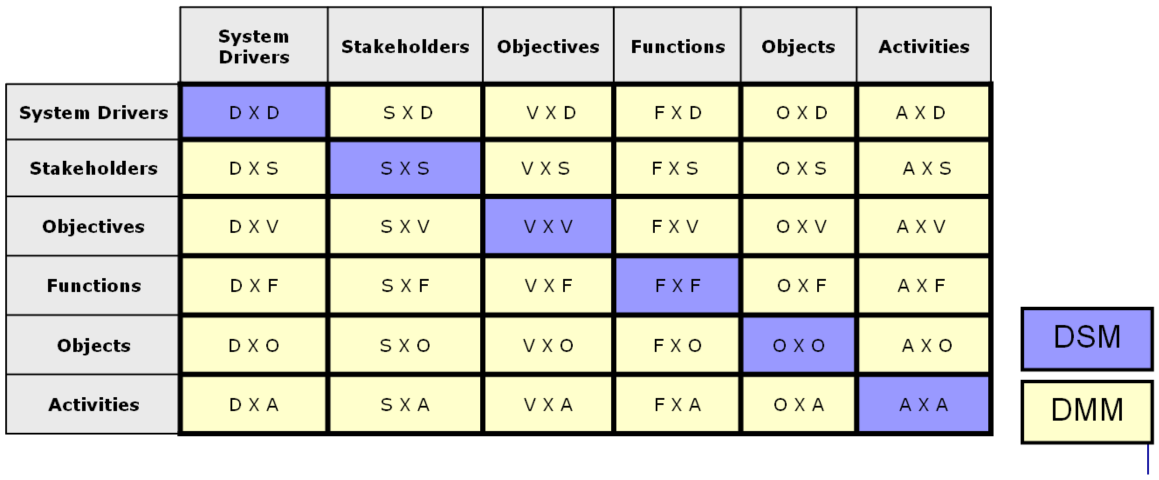
- Can the techniques presented in Part 2 be applied to DMMs and the ESM to provide useful analysis?

The order of these questions is intentional. First, it is important to understand how the ESM can be manipulated to accommodate the assumptions or constraints of the analysis techniques. Then, the practicality of applying those techniques to provide useful results can be considered.

Assumptions of Techniques

Classical DSM techniques assume square matrices and a single domain. The methods expect the rows and columns to be identical. Both the ESM and DMM violate both these conventions. While it is possible that a ESM or DMM may be square, this case would be the exception, not the rule. Attempts in past research to restrict the information matrix to square dimensions, forcing one-to-one mappings across domains, has produced incomplete or even misrepresented results, as expected.

The ESM can be viewed in two different perspectives: the matrix as a whole containing all the entered data or a collection of matrices comprising the matrix as a whole. Consider the following decomposition of the ESM, divided by the domain boundaries, to blocks of information as shown in Figure 4.



	System Drivers	Stakeholders	Objectives	Functions	Objects	Activities
System Drivers	D X D	S X D	V X D	F X D	O X D	A X D
Stakeholders	D X S	S X S	V X S	F X S	O X S	A X S
Objectives	D X V	S X V	V X V	F X V	O X V	A X V
Functions	D X F	S X F	V X F	F X F	O X F	A X F
Objects	D X O	S X O	V X O	F X O	O X O	A X O
Activities	D X A	S X A	V X A	F X A	O X A	A X A

Figure 4 ESM Decomposition to DSMs and DMMs

Notice that the blocks on the ESM diagonal are single domain DSMs. By further investigation, the off-diagonal blocks are DMMs. Therefore, the information residing in the individual DSMs now follow the assumptions required to classical DSM analysis techniques.

The DMM blocks may still violate the technique assumptions. The tools and algorithms currently employed for classical DSM analysis operate assuming a square matrix. Rather than forcing the use of insufficient tools, it is recommended that new algorithms be considered to accommodate rectangular matrices. Another approach might be to append "empty" rows or columns to create a square matrix; "empty" rows consist of a mock node that has no defined relationship to the true nodes in the matrix. However, this approach is not recommended. The influence of a mock node in the results may not be easily abstracted, thus invalidating any analysis.

Looking beyond the dimensions of the matrix, classical DSM techniques such as clustering and partitioning involve identifying patterns of relationships around the matrix diagonal. DMMs do not have this "diagonal," not only due to dimensions, but rather due to the rows and columns not being identical. The current algorithms for these techniques reorder the rows and columns together to form clusters or feedback loops. However, to apply clustering to DMMs, Danilovic and Browning (2007) required a new algorithm that moves the rows and columns individually in an attempt to find clusters of relations. Figure 5 displays an example of clustering of a DMM. (Danilovic and Browning 2007) Although no analysis has been accomplished to date, it is hypothesized that a similar algorithm, if not the same, could be applied to the ESM.

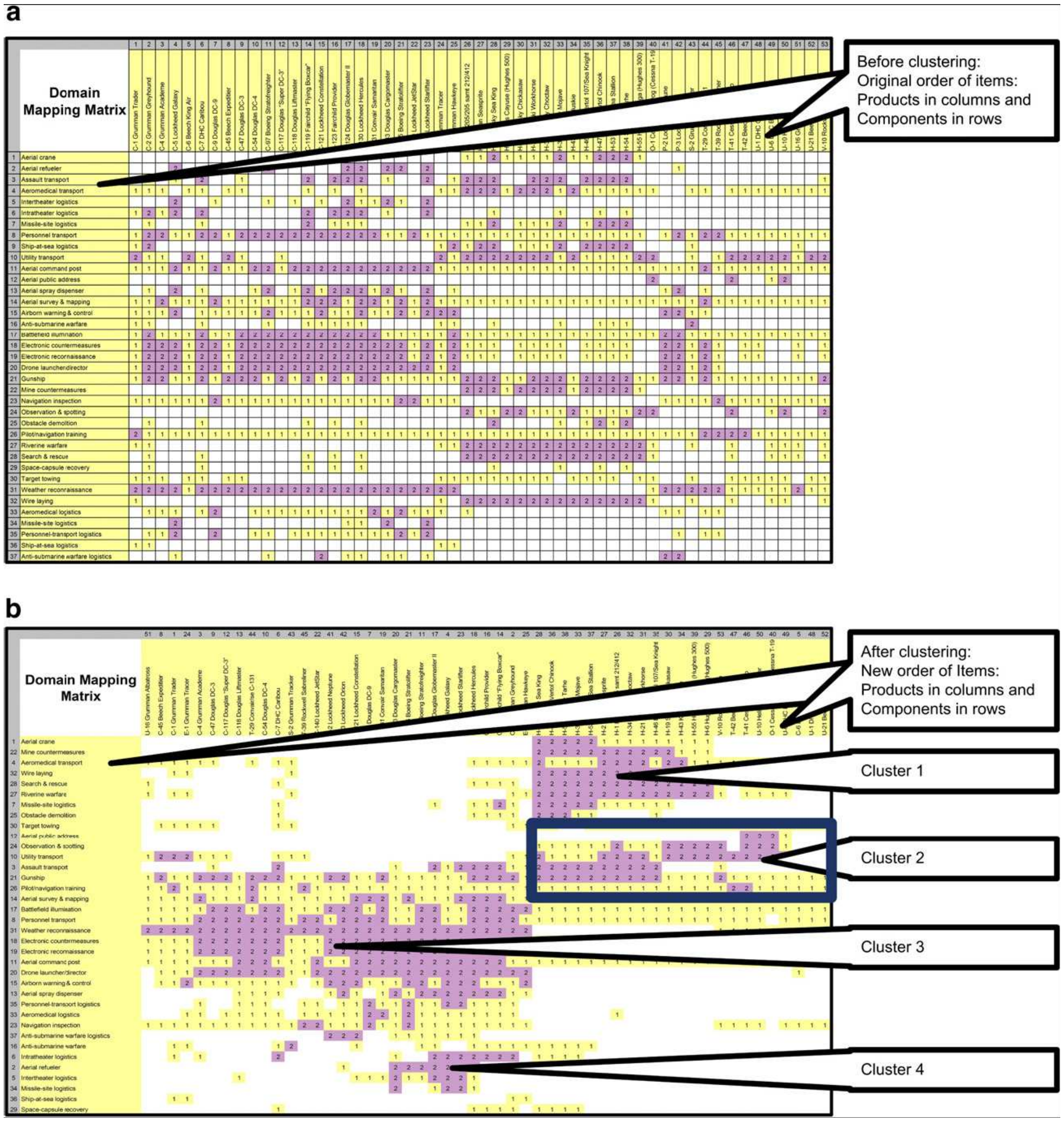

a
Domain Mapping Matrix
Before clustering: Original order of items: Products in columns and Components in rows
b
Domain Mapping Matrix
After clustering: New order of Items: Products in columns and Components in rows
Cluster 1
Cluster 2
Cluster 3
Cluster 4

Figure 5 Example of Clustering Applied to DMM (Source: Danilovic and Browning 2007)

Non-DSM techniques also include assumptions impacting the application to multi-domain analysis. However, most can be summarized by considering that typically only one domain is considered. Thus, analysis of the results must be carefully scrutinized to understand the implications. This is the emphasis of the remainder of Part 3.

<u>Producing Useful Results</u>

Now that it seems possible to extend the analysis techniques beyond the original purposes, consider whether it is appropriate to apply the techniques and the validity of the results. This section will provide examples of applying the analysis techniques to the ESM and DMM.

Application of Classical DSM Techniques

As previously emphasized, partitioning and sequencing apply to time-based DSMs, while clustering applies to static-based DSMs. Recall from Part 1 that time-based DSMs and static-based DSMs are distinguished by the domain represented by the nodes. Therefore, in the case of across domain analysis, as in a DMM or ESM, the application of clustering or partitioning is not clearly delineated. Classical DSM analysis techniques involve homogeneity of domain. However, Danilovic and Browning (2007) suggest that this assumption can be resolved with careful attention to interpreting the results. If the desired result is identifying interdependencies between system elements for instance, clustering would still identify groups of interdependent entities. The clusters derived from an ESM would likely be heterogeneous, with a potential for great variety of entities and relationships grouped in the clusters. Even though the implied meaning of an ESM cluster may not be immediately clear, clustering will uncover interdependencies across all the entities in the ESM. This result could be used to better understand how elements from other domains are driving or impairing the system design. Additionally, once the clusters are brought to attention, subsequent analysis can be applied to determine the potential implications. Table 2 compares and contrasts clustering results as applied to single-domain analysis (DSM) and multi-domain analysis (DMM).

Table 2 Outcome of Clustering Analysis (Source: Modified from Danilovic and Browning 2007)

	DSM		DMM
Technique	Sequencing	Clustering	Clustering
Partitioning Algorithm	Triangularization	Clustering of blocks along diagonal	Clustering of items
Outcome of Analysis	Sequencing, minimizing feedback loops	Clusters of items, hierarchical structures/interface identification	Clusters of items, dependencies/interface identification

The application of partitioning or sequencing implies time-based information, and not all domains in the ESM methodology have time implications. Recall that all elements in the ESM have a time attribute known as existence. However, this is not the same time description of time implied by sequencing. Sequencing involves a timed duration to achieve task completion or a timed event describing a schedule. Therefore, it is unclear if sequencing can be applied. One possible outcome is that all elements without time dependency would be "flushed out" to the bottom of the matrix. Although no examples have been accomplished to date, Danilovic and Browning (2007) suggest that sequencing may be appropriate if one or more of the domains is time dependent. Further research is required to better understand this hypothesis.

To better understand the implications of classical DSM techniques consider the following example. A two-domain DMM mapping stakeholders to activities may be constructed to represent the responsibility or tasking relationship between the organization and the process of the product development. What algorithm should be used to perform analysis?

The stakeholder domain implies a clustering algorithm, resulting in clusters of dependencies. The activities domain suggests a sequencing algorithm, resulting in a efficiently sequenced process minimizing iterative time cycles. However, when pairing these domains, perhaps it is most important to first understand why the interactions are being analyzed.

To carry the example through, consider a program manager that is interested in rationally organizing the team to maximize productivity (reduce delays for information flow) and allocation of expertise while diversifying to cover all fields of the project. Therefore, the manager is interested in understanding not only the stakeholder interactions (ie the stakeholder DSM) and the activities sequencing (ie activities DSM), but requires knowledge of how the stakeholders relate to the activities (ie the stakeholder-activities DMM). See Figure 6.

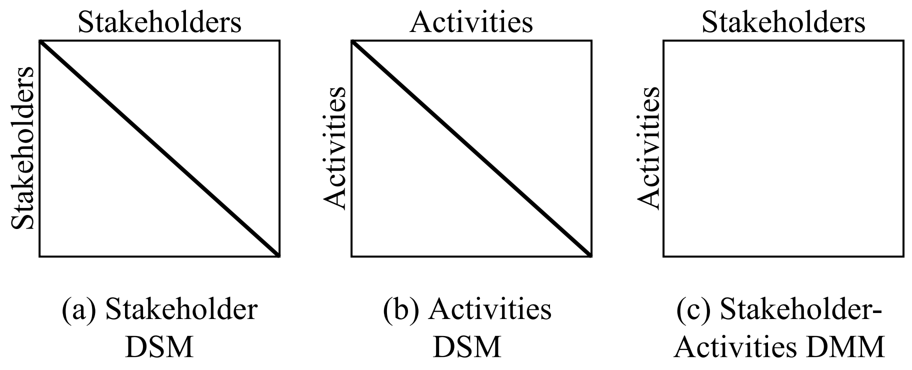


Figure 6 Individual Matrix Representation for (a) Stakeholders DSM, (b) Activities DSM, and (c) Stakeholders-Activities DMM

In this case, analyzing the two DSMs and the DMM separately may provide useful results. However, this compartmental strategy for analyzing the system may cause losses in true understanding of multi-domain interactions provided from a more holistic representation. Another strategy might be to apply techniques to a single combination matrix composed of the two-domains as shown in Figure 7.

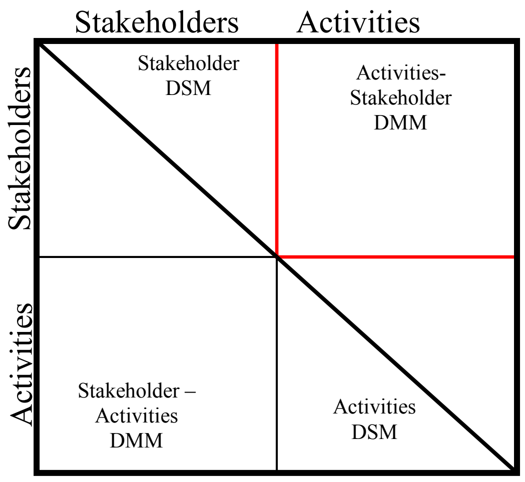


Figure 7 Stakeholder-Activities Combination Matrix Representation

Notice that the single matrix construction yields a different matrix diagonal. When analyzed separately, the stakeholders-activities DMM could only represent the existence of a relation between the stakeholder and the activity. Yet, when analyzed as a whole, the matrix can represent whether the stakeholder impacts the activity or the activity impacts the stakeholder, yielding feedforward/feedback information. This realization indicates that separate analysis, while sufficient in some cases, does produce losses of information or detail for clarity of information.

Furthermore, the analysis of the combination matrix still neglects the interactions with all the other domains of the ESM. Bartolomei concluded that managers often recognize the first-order relationships, but fail to understand the impacts of higher-order relations. For instance in this example, the program manager could most likely quickly interpret the analysis of the stakeholder-activities interactions from the individual DSMs and DMMs and the combination matrix with relative ease. However, answering why or how to affect the outcome requires additional information from other domains. Thus, expansion to multi-domain analysis and construction of an ESM might provide additional insights to consider, such as rotation requirements of stakeholders (system drivers), time delays of component delivery due to subsystem dependencies (objects), etc.

Application of Network Analysis Techniques

Social network analysis metrics, such as betweeness, path length, and centrality each have a particular meaning in the context of social networks, yet may be extensible to analysis of matrices including elements within the social domain. Bartolomei (2007) questions to what extent existing social network measures apply when analyzing a heterogeneous network with components from multiple domains. While his research does not attempt to answer this question, it does explore observations gleaned by applying these metrics to a product development case study. The metrics calculated included average degree (the average number of in- and out- relations per node), average path length, and clustering coefficient metrics for five different times in the product development life-cycle. (Newman 2003) Bartolomei (2007) concluded that the results differed over time, and thus warranted further investigation of the changing elements. He then analyzed the individual domains of the nodes appearing highest among the centrality measures, allowing him to better understand the product development dependency on a few critical system elements. Most interestingly to this discussion is that these critical elements were from multiple domains, results which could only be obtained by conducting multi-domain analysis.

Bartolomei (2007) also compared the betweeness analysis of the single domain stakeholders DSM to the analysis conducted over the entire ESM network. In this example, the betweeness for the top ten stakeholders in each result was predicted to be identical. However, the results proved that by restricting the analysis to the social domain only, the significance of stakeholders with strong interactions across other domains was not realized. Figure 8 displays the results of the betweeness analysis of both the DSM and the ESM.

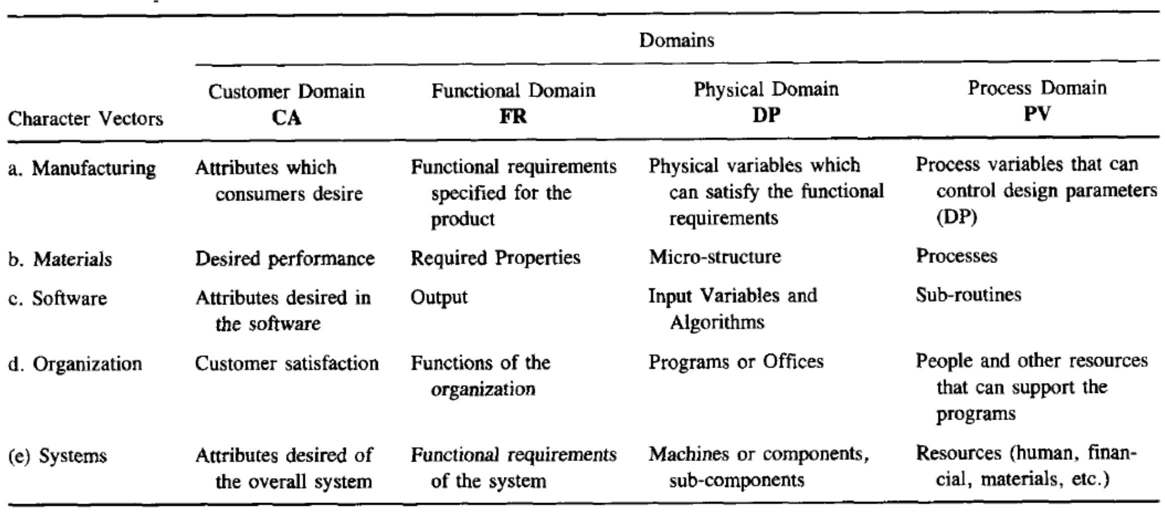
Rank	MAV-PD Social Network	Betweeness
1	PMWJ	500.199
2	STCC	199.471
3	**PMBI (MAV-PD PM 3)**	84.154
4	SPOMD	54.547
5	SPOKE	45.143
6	SPOGR	43.867
7	KTRDM	40.153
8	STYA	21.676
9	STSP	20.47
10	PMFC	15.23

Rank	MAV-PD Entire Network	Betweeness
1	PMWJ	10972.993
2	KTRDM	3680.017
3	KTRNM	1972.081
4	STCC	1556.707
5	PMBI (MAV-PD PM 3)	1372.588
6	KTRRC	1004.062
7	KTRTT	618.312
8	KTRBR	390.463
9	SPOMD	293.354
10	STYA	275.212

Figure 8 Stakeholder Betweeness Rankings Analyzed by (a) Single DSM and (b) the Entire ESM (Source: Bartolomei 2007)

These two examples suggest the importance of considering multi-domain interactions in complex systems involving both social and technical domains. Network analysis techniques may be able to give managers and engineers a richer understanding of how the social actors influence the system and/or provide insights into possible cause/effect relationships within a socio-technical system.

Network-based change propagation analysis been demonstrated as a tool for multi-domain analysis, although using Axiomatic Design rather than DMMs or ESM. Suh (2005) applies network-based change propagation analysis to the system components in an Axiomatic Design framework to identify the components classified as "change multipliers". (Bartolomei 2007) Axiomatic design consists of only four domains: customer domain, functional domain, physical domain, and process domain. (Suh 1995) Relating to ESMs, the customer domain can easily be correlated to the objectives matrix in terms of resident information. Likewise, the physical domain relates to the objects matrix, the process domain to the activities matrix, and the functional domain to the functions matrix. But the relationships above are not perfect and are incomplete. The ESM also contains information on the system drivers and the stakeholders. This information is indeed relevant and contained in axiomatic design, although not clearly observable. Suh (1995) suggests that the character vectors for the four domains changes depending on the context of the design as shown in Figure 9.



	Domains			
Character Vectors	Customer Domain **CA**	Functional Domain **FR**	Physical Domain **DP**	Process Domain **PV**
a. Manufacturing	Attributes which consumers desire	Functional requirements specified for the product	Physical variables which can satisfy the functional requirements	Process variables that can control design parameters (DP)
b. Materials	Desired performance	Required Properties	Micro-structure	Processes
c. Software	Attributes desired in the software	Output	Input Variables and Algorithms	Sub-routines
d. Organization	Customer satisfaction	Functions of the organization	Programs or Offices	People and other resources that can support the programs
(e) Systems	Attributes desired of the overall system	Functional requirements of the system	Machines or components, sub-components	Resources (human, financial, materials, etc.)

Figure 9 Characteristics for the Four Domains of the Design World [for various designs: manufacturing, materials, software, organizations, systems] (Source: Suh 1995)

Suh's character vector determines the type of information that will reside in the domain, and thus may also change how that domain corresponds to the ESM. For example, when examining the context of an organization the process variables are most likely people; however, the context of material design utilizes processes to populate the process variables. Therefore, it may be unwise to attempt to force the structure of axiomatic design onto the ESM approach. Instead, ESM may benefit from the general thinking of axiomatic design, specifically in thinking across two domains: function and form, form and process, or indirectly function and process.

Conclusions and Future Research

There are several promising analytical techniques and approaches that suggest that the DSM, DSM/DMM and ESM modeling frameworks can be useful to represent and improve understanding about engineering systems. The ESM extends beyond the knowledge of both DSM and DMM, including multi-domain information and the ability to include attributes of time and value.

This paper has provided many examples of each modeling framework and the application of analysis techniques. The literature contains numerous applications of classical DSM techniques applied to single domain matrices. However, the application of the DSM/DMM and ESM methodologies are limited at this time. Future work is required to determine the extent for which the methodologies can be efficiently used as a means to learn about the structure and behavior of engineering systems, with specific focus on multi-domain interactions. Additional consideration should be given to:

1. *How to apply appropriately the analysis techniques to the ESM Methodology?* The ESM can be analyzed holistically or as a decomposition of selected information for analysis. Further investigation of the impact of combining

information from multiple domains is required to understand the meaning of results from conventional analysis techniques. Additionally, research efforts for examining the integration of multi-design optimization, system dynamics models, and real options to the methodologies is critical for extending the applications of the frameworks.

2. *What are the applications of the ESM methodology?* Bartolomei (2007) suggested that the ESM methodology may be able to augment the development of system architectectures such as the Department of Defense Architecture Framework (DoDAF). MITRE has more recently expanded this effort and continues to correlate the information residing in the ESM to the individual views of the DoDAF. Furthermore, new research to determine the ability to scale engineering systems and identify real options for flexibility continues to mature.

Acknowledgements

The work presented in this paper was supported and funded in part by The MITRE Corporation, the US Air Force, and Massachusetts Institute of Technology Engineering Systems Division.

References

Atkinson, S. R. and Moffatt, J. (2005). "The Agile Organization: From Informal Networks to Complex Effects and Agility." Assistant Secretary of Defense (C3I/Command Control Research Program), DTIC: Washington, DC. NTIS Order Number: ADA457169. http://www.ntis.gov/search/product.asp?ABBR=ADA457169&starDB=GRAHIST

Barabasi, A. (2003). Linked: How Everything is Connected to Everything Else and What It Means. New York: Plume.

Bartolomei, J. E. (2007). Qualitative Knowledge Construction for Engineering Systems: Extending the Design Structure Matrix Methodology in Scope and Procedure. Cambridge, MA. Massachusetts Institute of Technology, Engineering Systems Division. Doctoral Dissertation.

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: 292-306. http://sbufaculty.tcu.edu/tbrowning/Publications/Browning%20(2001)--IEEE-TEM%204%20DSMs.pdf

Browning, T. R. (1996). Systematic IPT Integration in Lean Development Programs. Cambridge, MA. Massachusetts Institute of Technology, Technology and Policy Program. Master's Thesis.

Cesiel, D. S. (1993). A Structured Approach to Calibration Development for Automotive Diagnostic Systems. Cambridge, MA. Massachusetts Institute of Technology, Alfred P. Sloan School of Management/E.E. Master's Thesis.

Cokus, M. S. and Dahlgren, J. W. (2007). "System Evolution in the Intelligent Enterprise: A Historical Case Study of VISA's Transaction Processing Systems." San Diego, CA. International Conference on Systems Engineering (INCOSE) Proceedings, Session 5, Track 1: Intelligent Enterprises. http://www.incose.org/symp2007/

Dahlgren, J. W. (2007). "Developing a Framework for Exploring Clustering Coefficient to Evaluate System Coupling." Newport, RI. International Command and Control Research Technology Symposium (ICCRTS) Conference Proceedings, Technologies and Systems Track, No. 140. http://www.dodccrp.org/events/12th_ICCRTS/12th_ICCRTS_Detailed_Agenda.pdf

Dahlgren, J. W. and Cokus, M. S. (2007). "Real Options and Flexibility in Organizational Design." Honolulu, HI. First Annual Institute of Electrical and Electronics Engineers (IEEE) Systems Conference Proceedings, Track 2: Enterprise Systems Engineering. http://www.ieeesystemscouncil.org/conference-2007/sysconference2007.asp

Danilovic, M. and T. R. Browning (2007). "Managing Complex Product Development Projects with Design Structure Matrices and Domain Mapping Matrices." International Journal of Management 25: 300-314.

Danilovic, M. L. (1999). Leadership and Organization of Integration in Product Development. Linkoping, Sweden. Linkoping University. Doctoral Dissertation.

Dong, Q. (1999). Representing Information Flow and Knowledge Management in Product Design Using the Design Structure Matrix. Cambridge, MA. Massachusetts Institute of Technology, Department of Mechanical Engineering. Master's Thesis.

Eppinger, S. D. (2002). Patterns of Product Development Interactions. Cambridge, Mass. Massachusetts Institute of Technology: Working Paper No. ESD-WP-2003-01.05-ESD Internal Symposium. http://esd.mit.edu/WPS/author.htm#eppinger

Eppinger, S. D. (2001). "Innovation at the speed of information." Harvard Business Review 79(1): 149-+.

Eppinger, S. D. (1997). A Planning Method for Integration of Large-Scale Engineering Systems. International Conference on Engineering Design, Tampere. http://web.mit.edu/eppinger/www/publications.html

Fernandez, C. I. G. (1998). Integration Analysis of Product Architecture to Support Effective Team Co-Location. Cambridge, MA. Massachusetts Institute of Technology, Center for Innovation in Product Development. Master's Thesis. http://cipd.mit.edu/education/theses.htm

Grose, David L. (1994). "Reengineering the Aircraft Design Process." Panama City Beach, FL. Fifth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization. http://www.aiaa.org/content.cfm?pageid=406&gTable=mtgpaper&gID=78880

Kalligeros, K. (2006). Platforms and Real Options in Large-Scale Engineering Systems. Cambridge, MA. Massachusetts Institute of Technology, Engineering Systems Division. Doctoral Dissertation.

Koo, B. (2005). A Meta-Language for Systems Architecting. Engineering Systems Division. Cambridge MA. Massachusetts Institute of Technology, Engineering Systems Division. Doctoral Dissertation.

Kusiak, A. and Wang, J. (1993). " Efficient Organizing of Design Activities." International Journal of Production Research 31: 753-769.

Malmstrom, J. and J. Malmquist (1998). Trade-Off Analysis of Product Decompositions. Minneapolis MN. ASME Conference on Design Theory and Methodology. http://catalog.asme.org/ConferencePublications/CDROM/1998_Proceedings_Design.cfm

McCord, K. R. and Eppinger, S. D. (1993). Managing the Integration Problem in Concurrent Engineering. Cambridge, Mass. Alfred P. Sloan School of Management, Massachusetts Institute of Technology: Working Paper No. 3594. http://web.mit.edu/eppinger/www/publications.html
Morelli, M.D., Eppinger, S.D., and Gulati, R.K. (1995). "Predicting Technical Communication in Product Development Organizations." IEEE Transactions on Engineering Management 42: 215-222. http://web.mit.edu/eppinger/www/publications.html

Park, H. and M. R. Cutowsky (1999). "Framework for Modeling Dependencies in Collaborative Engineering Processes." Research in Engineering Design 6(1): 1-12.

Pimmler, T. U. and S. D. Eppinger (1994). Integration Analysis of Product Decompositions. Minneapolis, MN. ASME Conference on Design Theory and Methodology, pp343-351. http://web.mit.edu/eppinger/www/publications.html

Rask, I. and Sunnersjo, S. (1998). Design Structure Matrices for the Planning of Rule-Based Engineering Systems. Goteborg, Sweden. Proc. European Conference on Integration in Manufacturing.

Sabbaghian, N., Eppinger, S. D., and Murman, E. (1998). Product Development Process Capture and Display Using Web-Based Technologies. La Jolla, CA. IEEE Conference on Systems, Man, and Cybernetics 3: 2664-2669. http://web.mit.edu/eppinger/www/publications.html

Sharman, D. and Yassine, A. (2004). "Characterizing Complex Product Architectures." Systems Engineering 7(1): pp35-60.

Smith, R. P. and S. D. Eppinger (1997). "Identifying Controlling Features of Engineering Design Iteration." Management Science 43(3): 276-293. http://web.mit.edu/eppinger/www/publications.html

Sosa, M. E., Eppinger, S. D., and Rowles, C. M. (2004). "The Misalignment of Product Architecture and Organizational Structure in Complex Product Development." Management Science 50: 1674-1689. http://web.mit.edu/eppinger/www/publications.html

Steward, D. V. (1981). "The Design Structure System: A Method for Managing the Design of Complex Systems." IEEE Transactions on Engineering Management 28: 71-74.

Suh, E.S. (2005). Flexible Product Platforms. Cambridge, MA. Massachusetts Institute of Technology, Engineering Systems Division. Doctoral Dissertation.

Suh, N. P. (1995). “Designing-in of Quality Through Axiomatic Design.” IEEE Transactions on Reliability 44(2):256-264. http://dspace.mit.edu/handle/1721.1/25610

The Design Structure Matrix (DSM) Homepage. 10 July 2007. http://www.dsmweb.org/

Wilds, J. W., Bartolomei, J. E., de Neufville, R. and Hastings, D. (2007). “Real Options In a Mini-UAV System.” Hoboken, NJ. 5th Conference on Systems Engineering Research, Track 1.1: Doctoral Research in System Engineering. http://www.stevens.edu/engineering/cser/

Yu, T., Yassine, A., and Goldberg, D. (2003). “A Genetic Algorithm for Developing Modular Product Architectures.” Chicago, Illinois. ASME International Design Engineering and Technical Conference: Design Theory and Methodology. http://66.102.1.104/scholar?hl=en&lr=&q=cache:ATeENrYmPLMJ:www.ge.uiuc.edu/pdlab/Papers/DTM-48657.pdf+