Managing complex product development projects with design structure matrices and domain mapping matrices [☆]

Mike Danilovic ^{a,*}, Tyson R. Browning ^{b,1}

^a Jönköping International Business School, Jönköping University, Box 1026, SE-551 11 Jönköping, Sweden

^b M.J. Neeley School of Business, Texas Christian University (TCU), Box 298530, Fort Worth, TX 76129, USA

Received 4 April 2006; received in revised form 18 August 2006; accepted 2 November 2006

Abstract

Complexity in product development (PD) projects can emanate from the product design, the development process, the development organization, the tools and technologies applied, the requirements to be met, and other domains. In each of these domains, complexity arises from the numerous elements and their multitude of relationships, such as between the components of the product being developed, between the activities to develop them, and among the people doing the activities. One approach to handling this complexity is to represent and analyze these domains' *design structures* or *architectures*. The design structure matrix (DSM) has proved to be a very helpful tool for representing and analyzing the architecture of an individual system such as a product, process, or organization. Like many tools, the DSM has been applied in a variety of areas outside its original domain, as researchers and practitioners have sought to leverage its advantages. Along the way, however, its fundamental rules (such as being a square matrix) have been challenged. In this paper, we formalize an approach to using a domain mapping matrix (DMM) to compare two DSMs of different project domains. A DMM is a rectangular ($m \times n$) matrix relating two DSMs, where m is the size of DSM_1 and n is the size of DSM_2 . DMM analysis augments traditional DSM analyses. Our comparison of DSM and DMM approaches shows that DMM analysis offers several benefits. For example, it can help (1) capture the dynamics of PD, (2) show traceability of constraints across domains, (3) provide transparency between domains, (4) synchronize decisions across domains, (5) cross-verify domain models, (6) integrate a domain with the rest of a project or program, and (7) improve decision making among engineers and managers by providing a basis for communication and learning across domains.

© 2006 Elsevier Ltd and IPMA. All rights reserved.

Keywords: Project management; Design structure matrix; Dependency structure matrix; Domain mapping matrix; Product development; Management of complexity

1. Introduction

1.1. Background

Complexity in product development (PD) projects stems from many sources. The product or service to be developed

(the deliverable) may be complex in its function, form, integration, technology, etc. The work required to develop it is often complex in its number of activities, people, teams, and organizations involved and their relationships. These areas are interwoven, creating a number of complexities and uncertainties for managers. In our view, managers should focus on identifying, understanding, and reducing these product, process, and organizational uncertainties, among others, to add value.

Complexity can be identified and handled, and uncertainty reduced, by using a systematic approach to gathering, organizing, integrating, and analyzing the best information about a project. Models and tools that enable this also provide a basis for planning and learning [2,3].

[☆] The authors are thankful for numerous conversations with participants in the 6th International Design Structure Matrix Workshop, where this work was originally presented [1].

* Corresponding author. Tel.: +46 36 157588; fax: +46 708 157588.

E-mail addresses: mike.danilovic@jibs.hj.se (M. Danilovic), t.browning@tcu.edu (T.R. Browning).

¹ Tel.: +1 817 257 5069.

However, models must be based upon the latest and most accurate input information if they are to provide helpful output. Hence, trans-disciplinary or cross-functional teams are advocated to provide the collective expertise, information, and resources for effective model building and problem solving. However, such intensive interaction between people often causes conflicts because of variations in experience, knowledge, organizational or professional loyalty, understanding of the purpose and goals, and/or contradictory purposes and goals. Each project stakeholder has a different mental model of the project, assumptions about it, interpretations of realities, expectations, etc. A shared, codified model can test and align participants' mental models through discussions and lead to joint understanding of the reality in PD projects. To coordinate agents effectively into teams in the dynamic environment of PD, within and across different domains as indicated above, our research explores the following idea: we must lay bare the assumptions about the nature of the desired result, the activities to get to it, and the organization that will do the work, and the logic by which these domains have been decomposed and integrated.

PD projects are dynamic ones in which different domains are interwoven and effective management requires understanding how they interrelate and influence each other. Fig. 1 illustrates some critical aspects of PD that dynamically relate to each other. Product specifications are a consequence of customer requirements as well as logistic and manufacturing system capabilities. It is an illusion that PD starts solely with customer requirements and ends in design of the product structure. In reality this process is like a web. Domains are interrelated and information is flowing back and forth between them. The crucial aspect here is to

Fig. 1. Managing PD requires coordinating across many domains that enable and constrain each other.

Fig. 2. Five domains of complexity in PD projects (adapted from [4]).

understand and explore dependencies and the need for information exchange between different domains of product architectures, organizations and processes.

The problem for managers is to find the appropriate way to organize people and assign work over time, enable communication, and synchronize actions. The implication of such a dynamic approach is that managers and engineers must understand and take into account interdependencies and relations, and the information that needs to be exchanged, not only within each domain but also across domains.

PD projects contain at least five different domains (Fig. 2): 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. Each of these five systems is composed of elements with relationships and thus can be discussed in terms of its structure, network, and architecture—where architecture is defined, for example, as: “the structure of components, their relationships, and the principles and guidelines governing their design and evolution over time” (IEEE STD 610.12). Moreover, each of the five systems is related to the others. Each system both enables and constrains the others.

1.2. Motivation for our research

An enterprise typically has multiple projects going on at once (represented by the layers in the Fig. 2), and there are strong incentives to achieve commonality in these five systems across projects. In multi-project situations, each project usually does not have full control over its organizational structure, product architecture, process structure, etc., since companies usually want some commonality in these across projects to provide economies of scale and scope and easy project comparison. For example, they want organizational commonality so that the employee evaluation and promotion process is similar, product commonality so that all of the company's products have similar

themes, process commonality so that standard processes can be used and people can be assigned to work on multiple projects without having to learn an entirely different process, tool (especially information technology) commonality to provide economies of scale in purchasing software and training employees, etc. In spite of all of these limitations, projects have to be coordinated in order to produce a complete result—a product, a service, etc. However, when a project fails, it is often because someone forgot to account for the effects of some of these systems and their implications. And this should not be surprising, because there is a tremendous amount of information to consider and manage. In multi-project situations it is crucial to examine interfaces between projects and their needs for information exchange [5–10]. However, this is necessary in and across all of the domains. It is therefore essential to have a technique for examining the relationships between the elements of these systems.

1.3. Outline of the paper

In this paper, we compare two complementary, matrix-based approaches for representing, analyzing, and managing the crucial information regarding project domains and their interactions. The first, the design structure matrix (DSM), is a square matrix that has been used in a variety of product, process, and organization modelling applications over the last twenty years. We formalize the domain mapping matrix (DMM) as a second type of matrix-based approach, used to map between two different project domains. A DMM is a rectangular ($m \times n$) matrix relating two DSMs, where m is the size of DSM_1 and n is the size of DSM_2 . DMM analysis augments and adds insights to traditional DSM analyses. Our comparison of DSM and DMM approaches shows that DMM and DSM–DMM analyses offer a number of managerial insights and benefits.

The rest of this paper is organized as follows. Section 2 presents the DSM and DMM as two matrix-based tools for organizing the information that drives complexity within and across the five domains in PD. Section 3 compares and contrasts the DSM and DMM. Section 4 presents three example applications, Section 5 provides further discussion and insights, and Section 6 concludes the paper.

2. Matrix-based tools for managing PD complexity

2.1. Design structure matrix (DSM)

The methodology that is used to handle dependences and relations between items is widely known as the *design structure matrix* (DSM) [11].² As illustrated in Fig. 3, a DSM is a square matrix representing the elements in a system (the shaded cells along the diagonal) and their interac-

	A	B	C	D
A	A	1	1	
B		B	1	
C	1		C	1
D			1	D

Fig. 3. A DSM showing four elements of a system and their relationships.

tions (the off-diagonal marks). One reads across an element’s row to see its inputs and down its columns to see its outputs (although the opposite convention, the transpose of the matrix, is also used). For example, the DSM in Fig. 3 shows element A receiving inputs from elements B and C and providing an output to element C.

Over the years the DSM approach has been extended and compared with similar approaches (such as N^2 diagrams and precedence matrices), leading to a variety of extensions, approaches, and applications. Browning [12] provides a taxonomy of these approaches and identifies two discriminating dimensions related to whether the DSM represents static or time-based dependences. Integration analysis in each dimension requires a different approach: for static DSMs, the goal is typically to cluster the elements into modules with relatively high internal interaction and relatively low external interaction; for time-based DSMs, the goal is typically to lower-triangularize the matrix, if possible, and otherwise get all sub-diagonal marks as close to the diagonal as possible, thereby minimizing the number and scope of feedbacks in a temporal process. Algorithms for these two approaches are generally called *clustering* and *sequencing*, respectively. Browning’s taxonomy identified four types of DSM applications in these two dimensions and hypothesized additional applications and relationships between the domains represented by the individual square matrices.

The DSM in Fig. 4a shows an original ordering of activities, while Fig. 4b shows the reordered rows and columns of the matrix after implementing a sequencing algorithm. Sequencing orders the activities based on their dependencies, attempting to maximize the feed-forward flow of provided information and materials. Potential feedback loops stand out. The alternating light and dark bands³ identify sets of independent activities that can run concurrently. In Fig. 4, the item 1 is the first one to be done, followed by items 2–6 that can be done in parallel (although there is a potential feed-back loop in those items). Items 7–10 can be done concurrently after items 2–6. In this analysis it is possible to identify each activity’s predecessors and successors. Each set of activities that is prescribed to occur in parallel is supposed to be shown within a band. Sub-diagonal marks imply precedence relationships, while super-diagonal marks show the assumptions that must be made by upstream activities in lieu of information not yet

² The term “dependency structure matrix” is also widely used.

³ The innovation of banding was introduced by David Grose [13].

a

b

Fig. 4. Example of a DSM sequencing analysis: (a) before sequencing, (b) after sequencing.

produced by downstream activities. These assumptions are often the drivers of rework in projects, so it is important to expose them clearly and early and account for their potential impacts (risks) during project planning. Once the assumptions and couplings that drive iteration and rework in the PD project are identified and “unwound”, then traditional, linear project management tools and techniques like Gantt charts, project evaluation and review technique (PERT), critical path method (CPM), and critical chain [14] can be applied.

Other DSM applications focus on the product and organizational domains in a static sense. Here, sequencing is not the issue; rather, grouping highly related elements is the goal. A clustering algorithm is used to group elements along the diagonal. The interpretation of the analysis becomes very different than with sequencing.

Fig. 5 shows an example of clustering analysis. The DSM in Fig. 5a shows product components in their original order and their interdependencies as they were estimated on three levels: 1—low, 2—medium, and 3—high. Levels 2 and 3 interactions are shaded to highlight their importance. In Fig. 5b, the rows and columns have been reordered to group components with a high level of interaction. Clusters of items are identified and interfaces between clusters stand out as requiring special attention. Thus, a hierarchical approach to product and organizational architectures can be identified. Some items such as components or people can be part of at least two clusters and thereby function as “linking pins” [15–17].

The DSM has several close relatives—other techniques based on square matrices such as precedence matrices, adjacency matrices, reachability matrices, N^2 diagrams, etc. Examples of research using these methods are cited in [12]. In addition, Martin and Ishii [18] measure couplings between elements in square matrices to identify how each element contributes to variation in the entire product architecture. However, this and most other square-matrix approaches have not been used for integration analysis—i.e., they do not seek to reorder the matrix elements in sequences or clusters.

2.2. Domain mapping matrix (DMM)

Non-square matrices have also been used widely for product and project planning and analysis. One need look no further than quality function deployment (QFD), for example, which presents a series of five matrices mapping customer wants to product functional solutions, etc. e.g., [19]. Kusiak and Wang [20] use an incidence matrix to map activities to design parameters. Suh [21] similarly advocates matrices for mappings between the customer, functional, physical, and process domains in the design of mechanical systems. O'Donovan [22] demonstrates the use of rectangular matrices mapping activities to deliverables. In a mixture of rectangular and square matrices, Krackhardt and Carley [23,24] propose six modelling “primitives” with the acronym PCANSS:

- Precedence (represented by a square matrix, $P = T \times T$, mapping tasks to tasks)—which corresponds to an activity-based DSM.
- Commitment of resources (represented by a rectangular matrix, $C = T \times R$, mapping tasks to resources).
- Assignment of personnel to tasks (a matrix $A = I \times T$ mapping individuals to tasks).
- Network (a matrix $N = I \times I$ mapping individuals' direct relationships)—which corresponds to an organizational DSM.
- Skill (a matrix $S = I \times R$ mapping individuals to resources).
- Substitutes (a matrix $S = R \times R$ indicative of possible resource substitutes).

These primitives can be explored through possible matrix multiplications. The authors mention possible hypotheses to test regarding correlations between these primitives and find that the rectangular matrices can provide superior predictors of overall system performance. The best predictors of all stem from using several of the square and non-square matrices together.⁴

While there are strong relationships between the product, process, and organizational domains [12,28–32], DSMs have limited direct utility for inter-domain analyses. They have only been applied with this intent with the assumption that two domains did and should contain an equal number of the same elements (e.g., any product component would have a corresponding organizational unit responsible for its development). To move beyond this limitation, a rectangular matrix, mapping from one domain to another, is required. Only with such a rectangular matrix can the dynamics between different domains can be captured, relations and dependencies identified, and information that needs to be exchanged pointed out.

Danilovic presented studies of dependencies between domains in PD. Two papers explored product architecture versus organization structure [33] and task structure versus organization structure [34]. Danilovic and Sigemyr [35] presented dual-domain analyses of product requirements, product specifications, functional requirements, and product components. These papers referred to the dual-domain analyses as “rectangular DSM analysis”. Maurer et al. [36] showed a rectangular matrix relating product architecture and customer requirements.

To make a clear distinction between the square matrices that provide a self-mapping of the relationships among the elements of a system in a single domain and the rectangular matrices that map the elements of one domain to another, we developed the term domain mapping matrix (DMM) for the latter set to provide contrast with the term DSM, which has traditionally applied to the former set. Thus, while a DSM is always a square matrix, a DMM will usually be rectangular, although it can be square in cases

⁴ See also [25–27].

a

b

Fig. 5. Example of a DSM clustering analysis: (a) before clustering, (b) after clustering.

where two domains contain an equal number of elements in their respective systems.

As we stressed in the beginning of this article, PD projects are quite dynamic. The relationships between the domains in a project call for their consistency and synchronization, and the dynamic nature of PD calls for this to recur as the project unfolds and the domains evolve. It is critical to keep track of information and ensure that we do not lose important information or add superfluous information. Information traceability is also important.

Fig. 6 shows a DMM analysis of customer requirements mapped against product specifications, akin to a top-level QFD matrix. The upper matrix shows the original mapping, while the lower one shows the same data after clustering. Note that there is no diagonal in the matrix around which to cluster items: items can be clustered anywhere in the matrix, using an algorithm by McCormick et al. [37]. The outcome of such DMM analysis is that we can see how a particular set of customer requirements is met by a set of product specifications—or, vice versa, how a particular set of product specifications satisfies customer requirements. (In traditional QFD analysis, this tends to happen only in

terms of individual elements, not sets thereof.) Where the two dimensions correspond to each other indicates where a high level of interdependencies requires intense coordination, and what products and subsystems can be integrated in team settings. Fig. 6 shows four major and one minor cluster. Also, in the Fig. 6 we can identify an area without interdependencies between customer requirements and product specifications, called cluster 6. Here we can identify an area where two items in the product specification do not correspond to any identified customer requirement. This implies that we might have lost some important information when the product specifications were designed or that specifications were added without corresponding to any customer requirements. An incongruence between requirements and specifications results in both cases.

3. Comparing, contrasting and relating DSMs and DMMs

As mentioned above in Fig. 2, at least five major domains impact PD projects. Each is a system and can be represented by a DSM, and its relationship to each of the other systems can be shown by a DMM.

Fig. 6. Example of a DMM clustering analysis: (a) before clustering, (b) after clustering.

Fig. 7. A “periodic table” of DSMs and DMMs for the five project systems/domains.

Fig. 8. DSMs and DMMs specifically for the product system.

Fig. 7 arrays these five domains and their relationships with each of the other four. Three of these domains (product, process, and organization) have been investigated through traditional DSM analysis; see [12] for a review. The other two systems, tools and goals, are prime candidates for DSM analysis. Between these domains we indicate possible intermediate DMM investigations. The current order of these five DSMs in Fig. 7 is nearly arbitrary; future research might enable prescription of a preferred order. By synchronizing information about these five domains and their interactions in a PD project, we have an improved basis for representing and analyzing the sources of complexity, uncertainty, and dynamism in PD projects. Since each of these DSMs and DMMs represents an important “element” of a project, we refer to Fig. 7 as a kind of “periodic table”, since it reminds us of the classical periodic system of physical atomic elements. Much as the periodic classification of the elements spurred the investigation and discovery of previously unknown elements to fill in the gaps, Fig. 7 enables us to hypothesize the presence of potentially enlightening inter-domain representations and analyses in projects.

Fig. 8 shows some other traditional domains of interest to PD projects: customer requirements, product functionality, design parameters, product specifications, and product architecture [38]. The product domain is shown in both Figs. 7 and 8. Fig. 8 reminds us of the flow-down of matrices for QFD or the “House of Quality” e.g., [39], where some “rooms” might be single-domain representations while others might map between domains. However, QFD does not include an analyst’s “tool kit” for clustering or sequencing within and between domains. In this regard we suggest that the DSM and DMM representations can help to put focus on interdependencies and relations and the need for exchanging information within and between domains.

Table 1 summarizes some major aspects of the three types of matrices in terms of representation, focus, analysis, visualization, and key words. While the possible and appropriate analyses of DMMs remain an important area for future research, our preliminary work suggests that clustering analysis is insightful. We have applied McCormick et al.’s [37] algorithm so far. We note some major differences between the three matrices in terms of integration analysis:

- DSM sequencing shows dependencies in terms of how they have to be carried out. In this kind of analysis we can see which items can be carried out in parallel, in sequence, or in some combination (in the case of coupled blocks). This analysis is time-dependent and therefore the outcome is sequencing of activities.
- DSM clustering is time-independent. The visualization of dependencies regards where and how dependencies can be clustered in hierarchical structures. Between the identified clusters, interface areas are identified. In organizational analysis this can also contribute to identifying and designing hierarchical structures.
- DMM is in this regard similar to DSM clustering. The only difference is the analytical dimensions and that clustering is not concentrating along the diagonal.

The three approaches use different words to identify the focus of analysis and the reordered matrix:

- In DSM sequencing we identify relations between items in terms of the level of sequencing or parallelization. The major outcome of this analysis is identification of feedback loops and circuits.
- In DSM clustering we identify groups of items. Here we can identify interface areas and also see how different clusters relate to each other in the hierarchical aspect.

Table 1
Comparison between dependence matrix approaches

Dimensions	Design structure matrix (DSM)		Domain mapping matrix (DMM)
	Sequencing analysis	Clustering analysis	
Representation	Square matrix	Square matrix	Rectangular matrix
Analytical dimensions	Single domain	Single domain	Dual domain
Partitioning algorithm	Block diagonalization/triangularization	Clustering in blocks along the diagonal	Move items to clusters
Result, outcome of analysis	Flow of items, activities, i.e. sequencing	Clustering of items	Clustering of items
Visualization of dependencies	Parallelization of items	Clusters of items, Chunks	Clusters of items
	Sequencing of items	Interfaces between clusters	Interfaces between clusters
	Feedback and circuits, loops of items	Hierarchical structures	Hierarchical structures
Key words	Predecessors	Clustering	Clustering
	Successors	Hierarchies	Hierarchies
	Feedback and circuit loops	Interfaces	Interfaces
	Banding	Linking pins	Linking pins
Focus of analysis	Tasks	Parameters	Components/organization
	Activities	Components	Project/organizational structure
	Information flow	People	Functionality/product architecture
	Deliverables flow	Organizations	Information flow
		Information flow	

- In the DMM analysis the same words are used as for DSM clustering, but the analysis is performed across two domains.

Finally, the three analyses involve different foci:

- DSM sequencing is preferably used to analyze time-dependent items such as activities based on the analysis of information flow and dependencies among them. However, Steward's [40] recent work extends the original DSM approach to focus on general problem solving. In such an approach the same algorithm is applied and the analysis supports identifying the structure of a problem, without relation to time dimensions.
- DSM clustering is preferable for analyzing time-independent systems or single-domain analyses such as product architecture or project organization.
- DMM is preferable for analyzing relations and dependencies between domains and combinations of different domains.

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

From a PD project manager's perspective, complexity and uncertainty must be managed. Complexity cannot be "solved". What we can do is understand the aspects of complexity driving uncertainty in order to reduce it when making decisions such as how to design the product structure (modularization), how to design the project flow in terms of sequencing activities, and how to design the organization for efficient project execution and coordination. DSMs and DMMs are both useful contributors in this process of uncertainty reduction and management.

4. Example applications

In this section we provide three illustrations of DMM analysis:

1. An example of a DMM-analysis between a product system (military aircraft used in the US Air Force) and functionalities/technologies (technical subsystems such as aerial crane, aerial refuel, and electronic countermeasures). Often these technical subsystems contain specific technologies and are developed and delivered by a number of suppliers.
2. An example of integrated DSM and DMM analysis of the business portfolio, project system and organization system.

3. An example of a dynamic analysis where two DSMs, representing the organization and product systems, are combined with a dual-domain DMM analysis of product architecture and multi-project structure.

We intend for these analyses to demonstrate even further applications than the ones classified in Figs. 7 and 8.

4.1. Example 1

The managerial issue in this illustration is the problem of coordinating development of different aircraft with the development of various technologies to be used in them. When technologies are developed to enable certain functionalities, a high level of coordination is required between the system integrators applying those functionalities in their aircraft designs—both new models under development and old models being upgraded over their 35–50 year service lives.

Fig. 9 shows an example of a DMM analysis reflecting the issues raised above. The upper DMM shows a mapping of dependencies between the functions/technologies (in rows) used across different aircraft and helicopters ("products", in columns) in the US military air fleet. In the matrix numbers 1 and 2 are used and coloured in different ways to highlight the pattern of interdependencies between the domains. The lower DMM shows a matrix sorted by clustering, which moves rows and columns independently, searching for clusters of items in rows and columns. This analysis identifies four major clusters. Each contains a set of products (aircraft and helicopters) that uses a certain set of technologies/functions. One possible interpretation is that these three clusters could be seen as three joint development teams containing systems integrators and suppliers of different technologies.

4.2. Example 2

Fig. 10 shows one DSM and one DMM analysis based on data collected at a military aircraft manufacturing company by Danilovic and Börjesson [34]. One aircraft was undergoing a large number of technical upgrades, but these new functionalities were based on many separate business deals with the customer. The large number of business deals made it difficult to identify how the project should be structured to ensure on-time deliveries of finished aircraft. A high level of coordination is needed between the portfolio of projects and the overall organization. The company traditionally designed projects according to this product structure. Could the organization be structured in a more effective way, focusing on delivery of the complete product rather than its subsystems, through insights from the product architecture?

Analyzing a combination of one DSM and one DMM provided several insights. The columns and rows in the DSM in Fig. 10 show the business portfolio elements and their interdependencies (i.e., related business deals), after

Fig. 9. DMM analysis Example 1: (a) before clustering, (b) after clustering.

clustering the original matrix. Analyzing this DSM indicated four clusters, each of which was then defined as a development project—i.e., project teams were assigned to carry out the engineering work for each of the sub-systems represented by a cluster. Thus, four major projects were defined based on the pattern of interdependencies among business deals.

The next issue was how to coordinate each project with the functional organization. This was important because some people were working in several projects that were interdependent in the functional organization that was responsible for final system approval and certification. Effective communication was important both within each

project and between the projects and the functional organization. In the DMM in Fig. 10b, the four projects identified from Fig. 10a are mapped against the elements of the functional organization. Interdependencies of varying strengths are identified across projects and departments, as indicated by numbers and shading in the DMM. By clustering this DMM, we identify areas between projects and functional organizations that require a high level of coordination and integration. Interfaces between each project and functional organization indicate the people who must communicate to transfer information. The outcome of this DSM and DMM investigation was two-fold: DSM investigation ended in a more purposeful multi-project structure, and

Fig. 10. (a) DSM Clustering—business portfolio into multi-project portfolio. (b) DMM Clustering—multi-project portfolio vs. organization.

the DMM investigation ended in a more elaborated communication and coordination plan between the multi-project structure and the functional organization. Two years later the final product was delivered on time.

4.3. Example 3

We now turn to a dynamic DSM and DMM study. The original data were collected in one large study containing

three separate, simultaneous DSM and DMM studies of interdependencies between (1) departments in the functional organization, (2) subsystems of the product architecture, and (3) projects in the multi-project environment [6]. The company, which manufactures trucks, faced problems running many different development projects simultaneously, especially the huge need for coordination to secure the development and manufacturing of customized trucks. People were organized in many different projects, and some were in functional departments. The crucial issue was to ensure timely communication between people to meet the schedule of the entire development program.

The DSM and the DMM can be combined to depict inter-domain dynamics. Fig. 11 shows a combination of two DSMs and one DMM.

Fig. 11a shows an organizational DSM with departmental interdependencies, already sorted into three major clus-

ters. The clusters indicate possibilities for cross-functional, integrated teams to facilitate the most intensive interdepartmental coordination. Fig. 11b shows a clustered product architecture DSM for one of the engine development projects. It indicates two clusters of highly related components. These clusters indicate two modules, which portend cross-functional integration teams for the project. However, many other projects were going on at the same time in the organization, and all of these projects needed to communicate with each other to deliver complete trucks to the customers on time.

The DMM in Fig. 11c maps the interdependencies between the product analyzed in Fig. 11b and eleven simultaneous development projects on gearboxes, chassis, driving lines, cabins, etc. All of these projects required extensive coordination. The DMM reflects dependencies between one engine project's product-oriented architecture

Fig. 11. Dynamics in product development—combination of DSM and DMM: (a) DSM—organizational system; (b) DSM—product architecture; (c) DMM—product architecture vs. multi-project system.

and all other on-going projects. Each of the identified points of interaction must be handled through some kind of information exchange. From a managerial perspective, some major areas were identified that could guide managers to give special attention to the dense areas between particular project elements and certain other projects. This gave managers a reasonable ground for prioritization of their action. The outcome was coordinated engineering work within a project and between this project and all other projects. The final delivery of the major project was on time.

5. Managerial implications

In this paper, we have pointed out that complexity in PD is a consequence of elements and their relationships and dynamic variations in both. Management must deal with the resulting uncertainty in PD projects that affects the product, process, and organization designs, among other areas. Uncertainty stems from (often flawed) assumptions about dependences and the need for information exchange within and between domains and people that are needed to solve problems and define/design/manufacture the final product. Handling all of these aspects of complex PD requires approaches to aid understanding multiple domains simultaneously and in an integrated way. From a managerial perspective organizational design requires that temporal teams are designed to run PD projects, as a complement to the basic organization. How many teams need to be set up? Who is supposed to be a member of one team or another? How can managers be sure that relevant information is exchanged between people in different teams? If managers put focus on dependencies and the need for information exchange between people in the organizational design, purposeful team structures might evolve. Similar arguments can be put forward when it comes to product architectural design and process design, especially in the multi-project settings that characterize complex systems. The suggested approaches using DSMs and DMMs and combinations of them can lead to practical tools that support managers in analyzing dependencies within and between domains, and in different phases of PD.

- In order to handle the dynamics of PD, different teams are formed and may perform the work in different phases. There is a risk that actions and information in one team, phase and domain will be lost or unconsidered when work is carried out in another team, phase and/or domain. DMM analysis enables the synchronization of activities by mapping information from one domain into another and helping analyze its implications.
- Analysis of DSMs and DMMs within and across several domains enables integration of the individual systems into a cohesive system—a truly integrated view of the technical and programmatic aspects enabling managers and engineers to have a holistic approach to problem solving in PD projects.
- A combination of DSM and DMM analyses provides managers with highly improved decision support and visibility into the total project system and at the same time provides engineers with information that they depend upon.
- When DMM analyses are performed and when information in different domains is depicted, compared to other domains, and transformed from one domain to another, this process shapes an arena for communication between people doing different work.
- Traditionally, people in corporations, engineers as well as managers, are specialized in different departments or specific activities. DMM analysis provides important information on relations and the need for information exchange between domains. It thereby stimulates people to learn from each other and about each other's jobs and working conditions through reflection, understanding, and action. The final outcome is situational visibility that enables more efficient problem solving.

6. Conclusions

In this paper, we have provided information and comparisons between traditional DSM approaches and the cross-domain DMM approach that show their complementary nature and mutual advantages.

We harness dependency matrix approaches to represent and analyze the structure of elements and their interactions both within and across PD project domains. Since the traditional DSM focuses on a single domain, its analysis yields insights deemed optimal *from the point of view of that domain only*. However, the DMM provides way to represent interactions and relations between domains. We summarize the following points in comparing DSMs and DMMs:

- Traditional DSM analysis focuses on dependencies and flows of information in one domain. These single-domain analyses can be carried out in different domains simultaneously, but these analysis do not reflect the dynamics of PD and the need for transformation and comparison of information in one domain to another.
- Dual-domain DMM analysis reflects the dynamics of PD and enables transformation and traceability of information between domains. Performing DMM analysis between domains can ensure that information captured in one domain is not forgotten, added or dropped in another.
- Cross-domain verification of system models and project assumptions is possible when DMM analysis is performed in different domains.
- DMM analysis enables information from different domains to be communicated and understood by a variety of project participants, including engineers, marketers, manufacturers, financiers, and managers. During

domain comparisons, information updates in one domain are transparent to everyone and almost automatically updated in other domains.

- While all DSMs are square matrices, not all rectangular matrices are DMMs, and some DMMs may be square matrices. The most important characteristic of a DMM is that one domain is mapped against another.

Used together, DSMs and DMMs increase our knowledge and understanding of approaches to analyzing dependencies in complex systems and thereby facilitate the reduction of uncertainty and ambiguity in PD projects.

References

- [1] Danilovic M, Browning T. A formal approach for domain mapping matrices (DMM) to complement design structuring matrices (DSM). In: Proceedings of the 6th international design structure matrix (DSM) workshop, Cambridge, UK, September 12–14, 2004.
- [2] de Geus AP. Planning as learning. *Harvard Business Review* 1988;66(2):70–4.
- [3] De Meyer A, Loch CH, Pich MT. Managing project uncertainty: from variation to chaos. *MIT Sloan Manage Review* 2002;43(2):60–7.
- [4] Browning TR, Fricke E, Negele H. Key concepts in modeling product development processes. *Systems Eng* 2006;9(2):104–28.
- [5] Cusumano MA, Nobeoka K. Thinking beyond lean: how multi-project management is transforming product development at Toyota and other companies. New York: The Free Press; 1998.
- [6] Danilovic M, Sandkull B. The use of dependence structure matrix and domain mapping matrix in managing uncertainty in multiple project situations. *Int J Project Manage* 2005;23(3):193–203.
- [7] Hendriks MHA, Voeten B, Kroep L. Human capacity allocation and project portfolio planning in practice. *Int J Project Manage* 1998;17(3):181–8.
- [8] Grey RJ. Alternative approach to programme management. *Int J Project Manage* 1997;15(1):5–19.
- [9] Payne JH. Management of multiple simultaneous projects: a state-of-the-art review. *Int J Project Manage* 1995;13(3):163–8.
- [10] van der Merwe AP. Multi-project management: organizational structure and control. *Int J Project Manage* 1997;15(4):223–33.
- [11] Steward DV. The design structure system: a method for managing the design of complex systems. *IEEE Trans Eng Manage* 1981;28(3):71–4.
- [12] Browning TR. Applying the design structure matrix to system decomposition and integration problems: a review and new directions. *IEEE Trans Eng Manage* 2001;48(3):292–306.
- [13] Grose DL. Reengineering the aircraft design process. In: Proceedings of the 5th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization, Panama City Beach, FL, September 7–9, 1994.
- [14] Goldratt EM. Critical chain. Great Barrington, MA: The North River Press; 1997.
- [15] Sharman DM, Yassine AA. Characterizing complex product architectures. *Systems Eng* 2004;7(1):35–60.
- [16] Likert R. The human organization: its management and value. New York: McGraw-Hill; 1967.
- [17] Danilovic M. Loop: leadership and organization of integration in product development. Ph.D., Linköpings Universitet (Management and Economics); 1999.
- [18] Martin MV, Ishii K. Design for variety: developing standardized and modularized product platform architectures. *Research Eng Design* 2002;13(4):213–35.
- [19] Akao Y, editor. Quality function deployment: integrating customer requirements into product design. Cambridge, MA: Productivity Press; 1990.
- [20] Kusiak A, Wang J. Decomposition of the design process. *J Mech Design* 1993;115(4):687–95.
- [21] Suh NP. The principles of design. New York: Oxford University Press; 1990.
- [22] O'Donovan BD. The input/output matrix method. In: Proceedings of the 4th international design structure matrix workshop, Cambridge, MA, October 7–8, 2002.
- [23] Krackhardt D, Carley KM. A PCANS model of structure in organizations. In: Proceedings of the 1998 international symposium on command and control research and technology, Monterey, CA, June, 1998.
- [24] Carley KM, Krackhardt D. A typology for C^2 measures. In: Proceedings of the 1999 international symposium on command and control research and technology, Newport, RI, June, 1999.
- [25] Carley KM, Ren Y, Krackhardt D. Measuring and modeling change in C^3I architectures. In: Proceedings of the 2000 command and control research and technology symposium, Monterey, CA, June 26–28, 2000.
- [26] Carley KM, Ren Y. Tradeoffs between performance and adaptability for C^3I architectures. In: Proceedings of the 2001 command and control research and technology symposium, Annapolis, MD, June, 2001.
- [27] Butts CT, Carley KM, Krackhardt D, Ren Y. Analyzing organizational structure with metamatrix. Carnegie-Mellon University; 2001. Tutorial.
- [28] Eppinger SD, Salminen V. Patterns of product development interactions. In: Proceedings of the international conference on engineering design (ICED), Glasgow, August 21–23, 2001.
- [29] Morelli MD, Eppinger SD, Gulati RK. Predicting technical communication in product development organizations. *IEEE Trans Eng Manage* 1995;42(3):215–22.
- [30] Gulati RK, Eppinger SD. The coupling of product architecture and organizational structure decisions. MIT International Center for Research on the Management of Technology, Working Paper no.151; 1996.
- [31] Sosa ME. Analyzing the effects of product architecture on technical communication in product development organizations. Ph.D., MIT (M.E.); 2000.
- [32] Sosa ME, Eppinger SD, Rowles CM. Identifying modular and integrative systems and their impact on design team interactions. *J Mech Design* 2003;125(2):240–52.
- [33] Danilovic M, Börjesson H. Managing the multi-project environment. In: Proceedings of the 3rd international design structure matrix workshop, Cambridge, MA, October 29–30, 2001.
- [34] Danilovic M, Börjesson H. Participatory dependence structure matrix approach. In: Proceedings of the 3rd international design structure matrix workshop, Cambridge, MA, October 29–30, 2001.
- [35] Danilovic M, Sigemyr T. DSM approach in early product development phases. In: Proceedings of the 5th international design structure matrix (DSM) workshop, Cambridge, UK October 22–23, 2003.
- [36] Maurer M, Pulm U, Lindeman U. Tendencies towards more and more flexibility. In: Proceedings of the 5th international design structure matrix (DSM) workshop, Cambridge, UK, October 22–23, 2003.
- [37] McCormick WT, Schweitzer PJ, White TW. Problem decomposition and data reorganization by a clustering technique. *Operat Research* 1972;20(5):993–1009.
- [38] Ulrich KT, Eppinger SD. Product design and development. third ed. New York: McGraw-Hill, Inc.; 2004.
- [39] Clausing D. Total quality development: a step-by-step guide to world-class concurrent engineering. New York: ASME Press; 1994.
- [40] Steward DV. It's all about problem solving. In: Proceedings of the 5th international design structure matrix (DSM) workshop, Cambridge, UK, October 22–23, 2003.