DESIGN STRCUTURE MATRIX (DSM): NEW DIRECTIONS

*A.H.M. Shamsuzzoha[1] and Nadia Bhuiyan[2]

[1]Department of Industrial and Production Engineering, Shah Jalal University of Science and Technology, Sylhet-3114, Bangladesh, and [2]Department of Mechanical and Industrial Engineering, Concordia University, Montreal, Quebec, H3G 1M8, Canada

Received August 2005, accepted November 2005

Communicated by Prof. Dr. Abdul Raouf

Abstract: The Design Structure Matrix (DSM) is a compact representation of the information structure of a design process. It is a powerful tool for representing and analyzing task dependencies of a design project. This method provides a major need in engineering design management through documenting information that is exchanged. Analyzing the structure of a design process can identify many opportunities to improve it. Building a DSM model of a project/system, improves the visibility and understanding of project/system complexity through information flows. With the help of a DSM model it can easily convey the process to others in a single snapshot. In this research work, an improvement of the existing DSM tool is proposed which permits design managers to find an optimum way of restructuring complex design tasks, exposing problems, and creating unique solutions that could not be found simply by manually inspecting the design matrix. The improved model/algorithm follows the information-based approach of the design structure matrix (DSM) method, and uses transformed matrix techniques to reduce product development time and cost through optimal task ordering, while maintaining a high level of quality.

Keywords: Product design, special operator, triangular matrix, design iteration, concurrent engineering

Introduction

Advanced technology, fierce market competition and changing demand are forcing companies to design better quality and less expensive products at a rapid time pace. A product is something sold by an enterprise to its customers. Product development (PD) is the set of activities beginning with the perception of a market opportunity and ending in the production, sale and delivery of a product. PD process is the sequence of steps or activities which an enterprise employs to conceive, design and commercialize a product [1]. The PD process in an organization can be a source of competitive advantage in many industries. PD teams today are facing a growing number of concerns, such as production complexity, resource consumption, future upgrades, maintenance, and recycling [2]. A complex PD project involves a large number of activities that may require coordinating the work of hundreds or thousands of people from various disciplines. The work of any one design task can affect many other development decisions throughout the organization. As complexity increases, it becomes very difficult to manage the interactions among tasks and people. It may be even impossible to predict the impact of a single design change throughout the development process [3]. Coordinating design decisions has therefore become a crucial responsibility of engineering management.

*E-mail: zohaipe@yahoo.com

Product development process is generally a complex procedure involving information exchange across many tasks in order to execute the work [4]. It requires innovation and innovation requires feedback loops. Product development performance is generally measured by the lead time to develop the product, the cost of the development effort, the manufacturing cost of the product, and the product's quality or attractiveness in the market [5]. Analysis of product development (PD) processes allows us to study product development efficiency and to suggest process improvements.

Iteration is a fundamental characteristic of any product development processes [3,6]. It is assumed that the iteration of a task occurs for the following reasons: (1) new information is obtained from overlapped tasks after starting to work with preliminary inputs, (2) inputs change when other tasks are reworked, and (3) outputs fail to meet established criteria. Many traditional project management tools such as CPM [7], Gantt, and PERT [8], models do not represent iterative task relationships very well. Although, these tools allow the modeling of sequential and parallel processes, they fail to address interdependency (feed back and iteration) which is very common in PD projects. To address this issue, a matrix-based tool called the Design Structure Matrix (DSM) has evolved. Steward [9] developed the design structure matrix (DSM) to model the information flow of design tasks and to identify their iterative loops. It differs from conventional project-management tools such as PERT, Gantt charts and CPM network diagrams in that it focuses on representing information flows of a design project rather than on the work flows.

The aim of this research was to examine the existing DSM tool and to develop a mathematical model or algorithm to restructure the complex PD projects in order to develop quality products more quickly and economically. Such improved design procedures offer opportunities to speed up development progress by enhancing inter-task coordination. This model/algorithm follows the information-based approach of the design structure matrix (DSM) method, and uses transformed matrix techniques to reduce product development time and cost through optimal task ordering, while maintaining a high level of quality.

Design structure matrix: an overview

A matrix-based tool called the Design Structure Matrix (DSM) introduced by Donald Steward [9] provides generic framework for information flow in a simple and elegant manner. Both the sequences and technical relationships are performed by using a matrix representation. These relationships define the technical structure of a project, which is then analyzed in order to find alternative sequences of the tasks. A DSM is a compact matrix representation of a project network. The matrix contains a list of all constituent activities and the corresponding information exchange patterns. That is, what information pieces (parameters) are required to start a certain activity and where does the information generated by that activity feed into. The DSM provides insights about how to manage a complex project, and highlights issues of information needs and requirements, task sequencing and iterations.

It is relatively straight forward to construct a DSM of any company's existing or future product development process. The first step is to identify the tasks involved, which is easy and often available as part of the project management documentation. The next step is to correctly identify the information needed of the various tasks. Once all of the task information is ready, the next step is to draw the projects DSM. First, all tasks are listed in the order in which they are presently carried out. These tasks are then arranged in the same order horizontally and vertically to form a matrix of rows and columns. The other tasks that supply the necessary information are marked off across each row corresponding to a task. In other words, looking across a row shows all the information inputs needed to complete a task and looking down a column shows all the information outputs that will be provided to other tasks.

Figure 1 (adopted from [3]) shown below is an example of DSM construction, where task B supplies input information to tasks C, F, G, J and K, while task D receives output information from tasks E, F and L. All marks above the diagonal are feedback marks. Feedback marks correspond to the required inputs that are not available at the time of executing a task. In this case, the execution of the dependent task will be based on assumptions regarding the status of the input tasks. As the project unfolds these assumptions are revised in the light of new information, and the dependent task is re-executed if needed. It is worth noting how easy it is to determine feedback relationships in the DSM compared to the graph, which makes the DSM a powerful, but simple, graphic representation of a complex system or project. The matrix can be manipulated in order to eliminate or reduce the feedback marks. This process is called partitioning [9,10]. When this is done, a transparent structure for the network starts to emerge, which allows better planning of the PD project. In Fig. 2, it is seen which tasks are sequential, which ones can be done in parallel, and which ones are coupled or iterative.

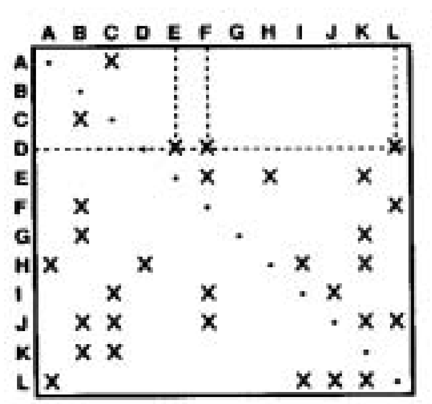

Fig. 1. A binary DSM (partitioned).

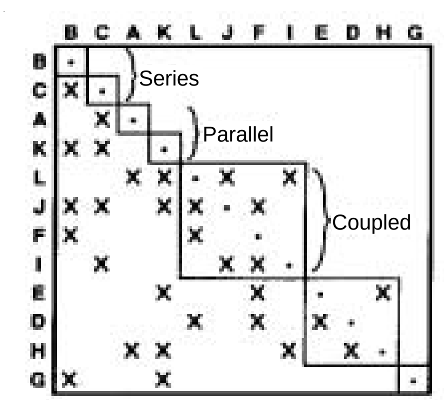

Fig. 2. A binary DSM (unpartitioned)

After portioning the DSM, the tasks in series are identified and executed sequentially. Parallel tasks are also exposed and can be executed concurrently. For the coupled ones, upfront planning is necessary. For example, we would be able to develop an iteration plan by determining what tasks should start the iteration process based on an initial guess or estimate of a missing piece of information. In Fig. 2, block E-D-H can be executed as follows: task E starts with an initial guess on H's output, E's output is fed to task D, then D's output is fed to task H, and finally H output is fed to task E. At this point, task E compares H's output to the initial guess made, and decides if an extra iteration is required or not depending on how far the initial estimate deviated from the latest information received from H. This iterative process proceeds until convergence occurs.

Methodology

The analytical methodology of the design process thus starts with building a structural model using a DSM. The DSM is then sorted out and tasks are rearranged in an attempt to eliminate feedback marks. The DSM is then partitioned into blocks containing task subsets involved in a cyclic information flow. Finally, feedback marks are torn from the DSM to break the cycles to eliminate one or more feedback marks within a coupled block in such a way that the rearrangement of the tasks within the block converts it in lower triangular form.

Results and Discussion

Research Problem: Transformed DSM

In the literature review, it has been observed that after partitioning, some coupled tasks still remain which cause lengthy lead-time and cost. The present research attempted to find a mathematical model or algorithm, which removes all coupled blocks or reduces coupling to a minimum level. To perform this operation, all upper diagonal feedback marks of a DSM had to be brought back to a more appropriate lower triangular form.

In mathematics, we know that if we have a matrix A and if |A| ‘“ 0, then there certainly exists another matrix H, which can make A become diagonalized, if the following calculation is applied:

H^{-1}. A. H = D (is a diagonalised matrix) …...… (i)

As in this example, we desired to find the transformation that can convert a DSM into one, which has the desired features in order to optimize the organization of the design activities. The problem can be solved in other ways, using an ‘operator’ to express the matrix form. Let’s consider the above example again; if A = a given DSM and B = the transformed DSM of A then its operator form becomes:

H^{+}. A. = B ……..……………………….... (ii)

Similarly, if we can find the expression of H^{+}, it will tell us how to coordinate the design activities to obtain optimal orderings of the DSM.

Here H^{+} is a ‘special operator matrix’, which can be defined as:

H^{+}.A. A^{-1} = B. A^{-1},

⇨ H^{+} = B. A^{-1} ……..................................… (iii)
[A. A^{-1} = I (identity matrix)]

In other way, the transformed matrix can be found out which is shown below:

H^{+}.A.H = B ………………………….. (iv)

Here, H^{+} and H are two different operators, which convert A to transformed matrix B. It is therefore, necessary to define these two operators for the DSM transformation.

H^{+}. A. H = B (is a diagonalised matrix)

Example of Matrix Transformation

Consider matrix A as a work transformation matrix, where the diagonal elements are zero and off diagonal marks represents dependency strengths between tasks. Now the columns of matrix A can be interchanged to bring the higher dependency marks into a lower triangle. This transformation occurs by multiplying matrix A by unit matrix H and different values of H may be used to observe the actual transformation-taking place. Therefore it can be written as:

A × H = B……………………………… (1)

where A is the original matrix and B is new matrix after changing the column or row. The matrix H is the unit matrix, which transforms matrix A to matrix B. Transformation of a sample matrix **A** is shown below.

Let:

A=	0.0	0.1	0.2	0.3
	0.3	0.0	0.4	0.2
	0.1	0.3	0.0	0.5
	0.1	0.1	0.2	0.0

B1=	0.1	0.0	0.2	0.3
	0.0	0.3	0.4	0.2
	0.3	0.1	0.0	0.5
	0.1	0.1	0.2	0.0

(We obtained ‘B1’after interchange of columns 1& 2 of ‘A’)

Therefore

H1= B1/A = ……………(2)

0	1	0	0
1	0	0	0
0	0	1	0
0	0	0	1

B2 =

0.2	0.1	0.0	0.3
0.4	0.0	0.3	0.2
0.0	0.3	0.1	0.5
0.2	0.1	0.1	0.0

(We obtained 'B2' after interchange of columns 1& 3 of 'A')

H2= B2/A = ……………(3)

0	0	1	0
0	1	0	0
1	0	0	0
0	0	0	1

B3 =

0.3	0.1	0.2	0.0
0.2	0.0	0.4	0.3
0.5	0.3	0.0	0.1
0.0	0.1	0.2	0.1

(We obtained 'B3' after interchange of columns 1& 4 of 'A')

H3= B3/A = ……………(4)

0	0	0	1
0	1	0	0
0	0	1	0
1	0	0	0

B4 =

0.0	0.2	0.1	0.3
0.3	0.4	0.0	0.2
0.1	0.0	0.3	0.5
0.1	0.2	0.1	0.0

(We obtained 'B4' after interchange of columns 2& 3 of 'A')

H4= B4/A = ……………(5)

1	0	0	0
0	0	1	0
0	1	0	0
0	0	0	1

and so on .....

From the above transformations, it can be observed that the *columns* of matrix A can be interchanged by multiplying matrix H1, H2, H3, H4 and so on, which convert easily to the higher dependency marks in the lower triangular form as necessary.

In another trial, carried out by interchanging the rows of matrix A, the following results were obtained:

A=

0.0	0.1	0.2	0.3
0.3	0.0	0.4	0.2
0.1	0.3	0.0	0.5
0.1	0.1	0.2	0.0

B'1=

0.3	0.0	0.4	0.2
0.0	0.1	0.2	0.3
0.1	0.3	0.0	0.5
0.1	0.1	0.2	0.0

(We obtained 'B1' after interchange of rows 1& 2 of 'A')

H'1= B'1/A = ……..(6)

-1.42	0.81	-1.6	0.80
0.48	0.84	0.32	-0.16
0.96	-0.32	1.6	-0.32
0.19	-0.06	0.12	0.93

B'2 =

0.1	0.3	0.0	0.5
0.3	0.0	0.4	0.2
0.0	0.1	0.2	0.3
0.1	0.1	0.2	0.0

(We obtained 'B'2' after interchange of rows 1& 3 of 'A')

H'2= B'2/A = ……..(7)

0.29	-1.42	1.42	-1.42
-0.25	0.48	0.52	-0.52
0.48	0.96	0.03	0.96
0.09	0.19	-1.19	1.19

B'3 =

0.1	0.1	0.1	0.0
0.3	0.0	0.4	0.2
0.1	0.3	0.0	0.5
0.0	0.1	0.2	0.3

(We obtained 'B'3' after interchange of rows 1& 4 of 'A')

H'3= B'3/A = 0.54   0   0   1…........(8)
-0.71   1   0   2
0.08   0   1   -0.24
0.52   0   0   -0.55

B'4 = 0.0   0.1   0.2   0.3
0.1   0.3   0.0   0.5
0.3   0.0   0.4   0.2
0.1   0.1   0.2   0.0

(We obtained 'B'4' after interchange of rows 2& 3 of 'A')

H'4=B'4/A=
0.80645   0.29032   -0.3871   0.29032 ..(9)
0.83871   -0.25806   1.6774   -1.2581
-0.32258   0.48387   0.35484   0.48387
-0.06451   0.096774 -0.12903   1.0968

and so on ....

From the above transformations, it is evident that rows of matrix A can be interchanged by multiplying matrix H'1, H'2, H'3, H'4, and so on, but it seems that a very complex form of H's is required to convert the higher dependency marks in the lower triangular form than the column interchange of matrix A. In this way, we hoped that after changing each row of any DSM, a generalized mathematical model could be developed for optimal task ordering.

Steps to find the 'Special Operator Matrix' H+:

(i) Analyzing published examples of product design using the DSM method and its transformed matrix to find out the relation with the operator matrix.

(ii) Using Mat Lab software to find out the internal relationship between two square matrices and also to find a way how they are transformed into a more convenient coupled or lower triangular form.

The following is an example which shows (Figs. 3 to 8) the way transformation occurs (using equation (i))

	A	B	C	D	E	F	G
A						1	
B				1			1
C					1		
D		1					1
E			1				
F	1						
G		1		1			

Fig. 3. A = Original DSM.

	A	F	G	B	D	C	E
A		1					
F	1						
G				1	1		
B			1		1		
D			1	1			
C							1
E						1	

Fig. 4. B1 = Transformed DSM.

$$A := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 5. A = Original DSM.

$$B := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & \frac{-1}{2} & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{-1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & \frac{-1}{2} \end{bmatrix}$$

Fig. 6. B = Inverse of Matrix A.

$$B1 := \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Fig. 7. B1 = Transformed Matrix (Optimized DSM).

H1=transformed matrix (B1) x Inverse of original matrix 'A' (B)

$$H1 = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -½ & 0 & ½ & 0 & ½ & ½ & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ ½ & 0 & -½ & 0 & -½ & ½ & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ ½ & 0 & ½ & 0 & ½ & -½ & 0 \end{vmatrix}$$

Fig. 8. H1= Transformed matrix (B1) x Inverse of original matrix'A' (B).

In this way a generalized mathematical model/ algorithm could be developed for the optimized DSM.

Significance of this research

To increase competitiveness, every firm has to develop its products with the importance of improving the efficiency and predictability of their design processes. Since, any process improvement requires process understanding, researchers and practitioners put effort into observing product design and development processes, looking for their important characteristics and developing models that account for those features. Most of the advances in this area assume that the design process has an underlying structure. An important characteristic of product development (PD) processes is that, unlike most business and production processes, they are described by terms like “creative,” “innovative,” and “iterative.” At an interesting level of detail, PD processes do not proceed in a purely sequential fashion. The activities in a PD process interact by exchanging information which is iterative.

Product development is considered to be a process of input information about customer needs and market opportunities into output information, which correspond manufacturability designs and functional tooling for volume production. In practice, the information exchanged between activities takes various forms such as customer specifications, parts dimension, and prototypes. Information exchanged in the engineering stages of product development can often be represented as a collection of parameters [3]. In real life, it is rare that a company will be able to design a process in which all interdependent or coupled tasks can be carried out together. In coupled blocks, a significant number of potential unplanned iterations can occur when errors are discovered during the project development process. This rework would also require the company to redo some intervening tasks. The company then decides what to do about them. The coupled tasks may be so far apart that a delay caused by incorporating late information effectively means starting the whole process again. These situations usually arise because some fundamental mistake in assumptions was made at the beginning of the project [11].

In this research, an improvement of the DSM tool is proposed that permits managers to find optimum ways of restructuring complex design tasks, exposing problems, and creating unique solutions that could not be found simply by manually inspecting the design matrix. This work will be able to reduce the lead-time of any development project. It can be done through resequencing/reordering the coupled task by using the proposed mathematical model/algorithm. If the model is developed it will definitely help designers/engineers to organize their works in more efficient ways than ever.

References

1. **Ulrich, K.T. and Eppinger, S.D.** 2004. *Product Design and Development.* ISBN 0-07-247146-8, 3rd Edition, McGraw-Hill, New York.
2. **Smith, R.P. and Eppinger, S.D.** 1997. A predictive model of sequential iteration in engineering design. *Management Science* 43:1104-1120.
3. **Eppinger, S., Whitney, D., Smith, R. and Gebala, D.** 1994. A model-based method for organizing tasks in product development. *Research in Engineering Design* 6:1-13.
4. **Krishnan, V., Eppinger, S.D. and Whitney, D.E.** 1995. Accelerating product development by the exchange of preliminary product design information. *J. Mechanical Design* 117:491-498.
5. **Clark, K.B., Chew, B. and Fujimoto, T.** 1987. *Product development in the world auto industry: performance: strategy, organization and performance*. Brookings Papers on Economic Activity, vol. 3, pp.729-771.
6. **Browning, R.T. and Eppinger, S.** 2002. Modeling the impact of process architecture on cost and schedule risk in product development. *IEEE Trans. Engineering Management* 49:428-442.
7. **Kelley, J. and Walker, M.** 1959. *Critical-path planning and scheduling*. Proceedings of Easter Joint Computer Conference, pp. 160-173.
8. **Malcolm, D., Roseboom, J., Clark, C. and Fazar, W.** 1959. Application of a technique for research and development program evaluation. *Operations Research* 7:646-669.
9. **Steward, D.V.** 1981. The design structure system: a method for managing the design of complex systems. *IEEE Trans. Engineering Management* 28:71-74.
10. **Yassine, A., Falkenburg, D. and Chelst, K.** 1999. Engineering design management: an Information structure approach. *International Journal of Production Research* 37:2957-2975.
11. **Kong, L.** 2003. Determining the value of processes and information structures for new product development. Ph.D. thesis, McGill University, Montreal, Canada.