DSM AS A KNOWLEDGE CAPTURE TOOL IN CODE ENVIRONMENT

Syed Ahsan Sharif, Berman Kayis

School of Mechanical and Manufacturing Engineering, The University of New South Wales, UNSW, Sydney, NSW 2052, Australia. [1]Email address: sasharif_unsw@yahoo.com

Abstract: A design structure matrix (DSM) provides a simple, compact, and visual representation of a complex system/process. This paper shows how DSM, a System Engineering tool, is applied as a knowledge capture (acquisition) tool in a Generic NPD process. The acquired knowledge (identified in the DSM) is then validated in an Australian manufacturing company. This acquired knowledge helps NPD teams, managers and stakeholders to benchmark their NPD efforts and select areas to focus their improvement efforts.

Key words: Design Structure Matrix (DSM); New Product Development (NPD); Customer Order Driven Engineering (CODE); Knowledge Management (KM)

1. 1. INTRODUCTION

In modern times we have focused on new manufacturing methods, shifting from mass to lean production, and are now at the next wave of manufacturing innovations – Customer Order Driven Engineering (CODE). Customers are demanding products that feature the latest in style and technology; offer utility, value and price; and meet quality and reliability expectations. In order to meet these customer needs CODE is the key concept for the manufacturing industry (Anderson, 2004; Chandra and Kamrani, 2004; Cheng and et al., 2002). CODE attempts to provide customized products for individual customers without losing many benefits of mass production – high productivity, low costs, consistent quality and fast response. To be successful, the CODE requires a major combination of

Please use the following format when citing this chapter:
Sharif, Syed Ahsan, Kayis, Berman, 2006, in International Federation for Information Processing (IFIP), Volume 207, Knowledge Enterprise: Intelligent Strategies In Product Design, Manufacturing, and Management, eds. K. Wang, Kovacs G., Wozny M., Fang M., (Boston: Springer), pp. 95-102.

efforts in all areas of business. Among the enablers of CODE, Knowledge Management (KM) is considered to be one of the most fundamental and challenging one, because the integration of activities and resources heavily relies on the integration and sharing of Knowledge (i.e. Knowledge, Information or Data) (Biesner and Brüggen, 2005; Davis et al., 2005). This paper shows how DSM, a System Engineering tool, is applied as a knowledge capture (acquisition) tool in a Generic NPD process. The acquired knowledge (identified in the DSM) is then validated in an Australian manufacturing company.

2. DESIGN STRUCTURE MATRIX (DSM)

DSM (variously expanded as the design structure matrix, problem solving matrix (PSM), dependency structure matrix and design precedence matrix) is a representation and analysis tool for system modeling, especially for decomposition and integration (Browning, 1999, 2001) purposes. A DSM displays the relationships between components of a system in a compact, visual, and analytical format. The advantages of DSMs in respect to other system representation and analysis techniques have led to their increasing use in a variety of context, including product development, project planning, project management, system engineering, and organization design.

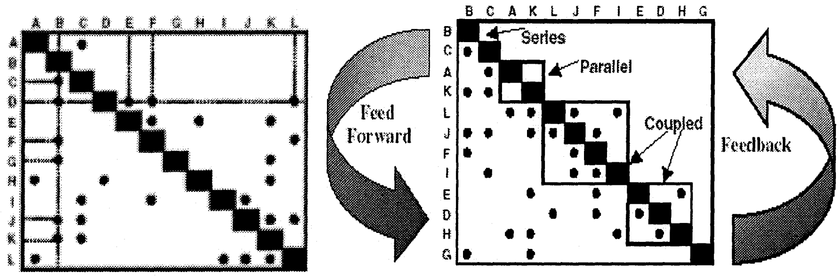


Figure 1. (a) Sample DSM, (b) Partitioned DSM matrix

DSM consists of an N-square matrix (Figure 1(a)) with one row and column per element and shows the interaction of each element with every other element in the model. DSMs elements are represented along the diagonal, usually represented by shaded area separate the upper and lower diagonal of the matrix. An off-diagonal mark signifies the dependency of one element on another. If the DSM elements represent tasks to be performed, then inspecting the row and column of a task reveals the inputs and outputs, respectively, for that task. For example, in Figure 1(a), B feeds

C, F, G, J, and K, while D is fed by E, F, and L. If there is a time sequence associated with the position of the elements in the matrix, then all marks above the diagonal are considered feedback marks. Feedback marks correspond to 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 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.

DSMs partitioning algorithms (Steward, 1981; Eppinger at el., 1990; Yassine et al., 1999) re-sequence the elements in the matrix, which makes it a lower triangular matrix. Lower triangular matrix means each task begins only after it receives all the required information from its predecessors. This shows that there are no interdependent/coupled tasks. They are either in serial or in parallel. However, this sequencing analysis sometimes produces a block triangular matrix – a coupled block. Figure 2.15, shows the same sample matrix after partitioning. In Figure 1(b), task B and C are serial, i.e. C can be done after B is completed. Task A and K are completely independent of each other, and can be done in parallel. Elements involved in a coupled block must be done concurrently, for instance task E, D and H should be performed simultaneously by using iterations and/or negotiations. Partitioned matrix provides the first step and the mathematical result of optimization. However, Partitioning algorithms do not eliminate coupled blocks, rather they provide iteration sequences. Further study can be done after partitioning, including assigning dependency values, tearing, de-coupling and add-coupling, and other non-binary DSM techniques (Dong, 1999) to deal with the coupled blocks (Optimization of the DSMs).

3. DSM REPRESENTATION OF NPD PHASES

The construction of DSM in this research was undertaken by using the following steps:

Step 1: Define the NPD phases

Step 2: List all the elements of the phases (Task breakdown structure)

Step 3: Construct the DSM matrix to represent the interactions between different tasks

Step 4: Partition and Optimize the DSM matrix

Step 1 and 2: Since the DSM is a system engineering tool, it is important to define the boundary of the system in order to focus the research work. A Generic NPD process (with six phases) is defined, and then detailed into 34 tasks (Task Level 1). These higher level tasks then decomposed and detailed into lower level tasks (Level 2, 3 or lower – depending on the nature of the tasks). Like wise Task Level 1, these Task breakdown structure is also generic in nature. This definition of NPD phase and identification of tasks, tries to cover the whole NPD process (i.e. for entire Product Life Cycle) in CODE environment and under the philosophy of Concurrent Engineering (CE), which emphasize on collaboration, co-ordination and better communication among the cross-functional teams inside the organization as well as active involvement of the extended enterprise (Customers and suppliers).

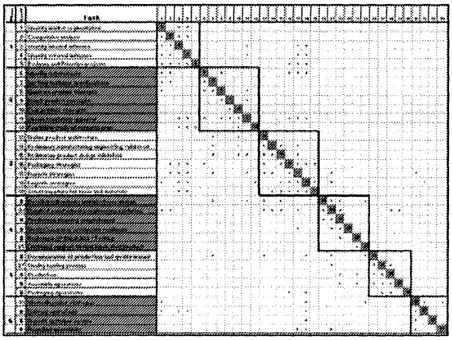


Figure 2(a). *Higher Order DSM for Generic NPD Process*

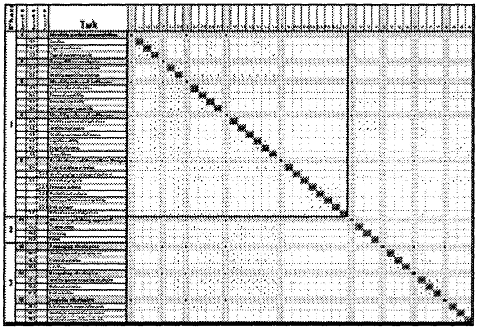


Figure 2(b). Decomposed DSM (Phase 1)

Step 3: The third step is to identify the interactions between different tasks and build the DSM matrix. The interactions between different tasks are identified by the "casual diagram" (Brian, 2006) and "surveying" the literatures related to NPD process. Figure 2(a) shows the DSM representation of a Generic Higher Order NPD process. Presenting very large models in a single matrix is challenging. When constructing models comprised of hundreds of tasks, the intuitiveness provided by the DSM representation diminishes. A very large DSM can be effectively structured into a hierarchy of smaller DSMs. This configuration avoids problems related to presenting extremely large matrices by shifting the focus to smaller ones, obtained through hierarchical decomposition. It also provides the flexibility to analyze the process at different levels of detail. This multi-tiered approach (developed by Dr. Grose (1994) at Boeing) is adopted to decompose the higher order DSM. Six more decomposed DSM (lower level) are developed from Figure 2(a) for the six phases of NPD. Figure 2(b) shows the decomposed DSM of Phase 1 only. Decomposed task breakdown structures are the source of these decomposed DSM.

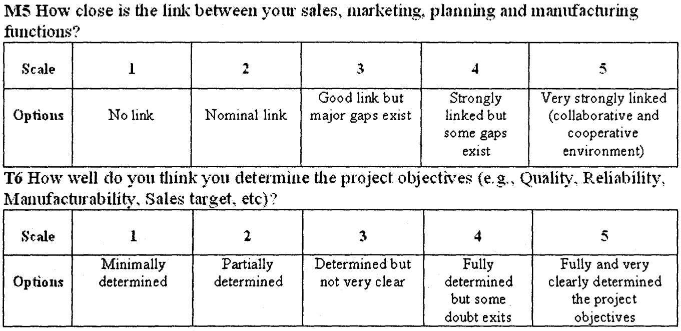
Step 4: After constructing the DSMs, they are partitioned and optimized as per requirement as discussed in section 2. The constructed DSM matrixes show the task interactions not only between the same phases but also between other phases too, and with different task levels. They also indicate the complexities (coupled blocks), important interactions and loops (feedback and iterative) exist in the different NPD phases.

4. KNOWLEDGE REPRESENATAION

Knowledge items identified in the DSM (e.g. important interactions, feedback and iterative loops etc) are provided in the form of Questionnaires. An Assessment Model is developed, consisting of five performance indicators of the organization namely 'Marketing', 'Technical', 'Financial', 'Resource Management', and 'Project Management'. Around 150 questions are developed (in total), which are organized into these five categories for each of the six NPD phases. More than 40 questions developed for 'Phase 1' only, to tackle the 'fuzzy front end' of NPD. These Questionnaires provide rich enterprise knowledge and designed to assist in the improvement of a company's product development performance. Figure 3 shows some sample questionnaires which are scaled from 1 to 5. Generally "Scale 1" represents poor and "Scale 5" represents excellent.

5. VALIDATION OF THE ASSESSMENT MODEL

The validation of the assessment model developed has two fold objectives. The first objective is to check the sensitivity of categories used by means of comparing successful versus less successful product from the same product group of the same company. And the second objective is to observe whether the model can display the overall performance of the company in the five categories of NPD phases.



M5 How close is the link between your sales, marketing, planning and manufacturing functions?

Scale	1	2	3	4	5
Options	No link	Nominal link	Good link but major gaps exist	Strongly linked but some gaps exist	Very strongly linked (collaborative and cooperative environment)

T6 How well do you think you determine the project objectives (e.g., Quality, Reliability, Manufacturability, Sales target, etc)?

Scale	1	2	3	4	5
Options	Minimally determined	Partially determined	Determined but not very clear	Fully determined but some doubt exits	Fully and very clearly determined the project objectives

Figure 3. Sample Questionnaires from Phase 1

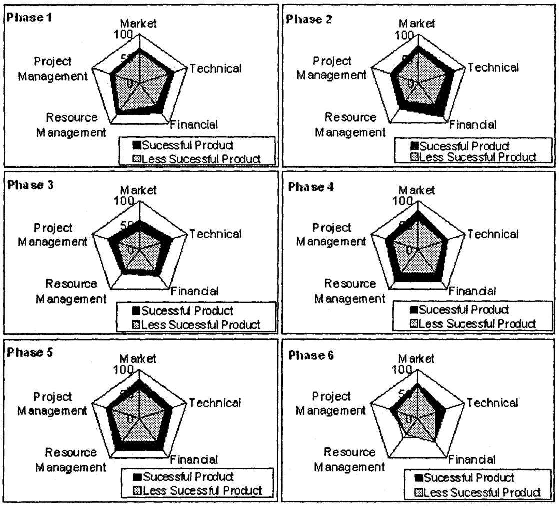


Figure 4. Sensitivity test

Figure 4 shows the sensitivity test of the Questionnaires, comparing Successful and Less Successful product. Both products are benchmarked against the acquired knowledge (100% score means the selected performance criteria are fully met). The results are consistent throughout the six NPD phases, i.e. less successful product have lesser percentage (or equal) than the successful product in each of five performance indicator axis. This sensitivity test (Figure 4) proves to be a powerful tool for the company to focus on areas that less successful product needs a remedial action.

The results of the successful product from previous "Sensitivity analysis" are considered again and the results are mapped (Figure 5) by "Performance Indicator" areas. This approach also assists with the benchmarking process where 100% score means the selected performance criteria are fully met. Figure 5 revealed the company's strong and weak areas in the product development process. For example, the company should give attention in 'Marketing' area of Phase 3, 'Technical' area of Phase 3, 4 and 6, and so on (as shown in circle in Figure 5). Although the data of the 'Project management' are steady for all of the six phases, it needs more consideration for all phases because of low scores which are in the order of 60%-70% only.

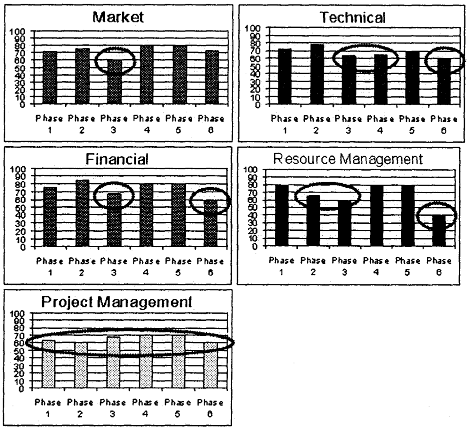


Figure 5. Total view

6. CONCLUSION

DSM is applied as a Knowledge capture (Knowledge acquisition) tool in the NPD process of CODE environment, which provides an essential knowledge to help NPD managers, teams and stakeholders in the complicated and uncertain nature of the NPD process. The industrial case used aims to validate the sensitivity of the acquired knowledge and more importantly shows the performance of the organization in five categories of each of NPD phases. It provides a means of assessing how well the company has met different product development performance criteria. Accordingly, companies would be able to identify the areas to focus for their improvement efforts.

7. REFERENCES

Anderson, D. M., 2004, *Build-to-Order and Mass Customization: The Ultimate Supply Chain Management and Lean Manufacturing Strategy for Low-Cost On-Demand Production without Forecasts or Inventory*, CIM Press, Cambria, California, pp. 1–9.

Biesner, J. and Brüggen, G., 2005, The Significance of Knowledge Management for a Larger Company, in: *Knowledge Management: Organizational and Technological Dimensions*, J. Davis, E. Subrahmanian, and A. Westerberg (Editors), Physica-Verlag Heidelberg, New York, pp. 159–167.

Brian, K.M.T., 2006, Ph.D. candidate, Research topic: *Implementing concurrent engineering with integration of intelligent agent based system in complex product*, School of Mechanical & Manufacturing Engineering, UNSW, Sydney, Australia.

Browning, T.R., 1999, The Design Structure Matrix, in: *Technology Management Handbook*, R.C. Dorf, Ed. Boca Raton, Fl: Chapman & Hall/CRCnet-BASE, pp. 103–111.

Browning, T.R., 2001, Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions, *IEEE Transactions on Engineering Management*, **48**(3):292–306.

Chandra, C. and Kamrani, A., 2004, *Mass Customization: A Supply Chain approach*, Kluwer Academic / Plenum Publishers, New York, pp. v–vii.

Cheng, F., Ettl, M., Lin, G. and Yao, D. D., 2002, Inventory-service optimization in configure-to-order systems, *Manufacturing and Service Operations Management*, **4**:114–132.

Davis, J., Subrahmanian, E. and Westerberg, A. (Editors), 2005, *Knowledge Management: Organizational and Technological Dimensions*, Physica-Verlag Heidelberg, New York, pp. v–vii.

Dong, Q., 1999, *Representing Information flow and Knowledge Management in Product Design using the Design Structure Matrix*. Masters Thesis, M.I.T., pp.61– 65.

Eppinger, S.D., Whitney, D.E., Smith, R.P., and Gebala, D., 1990, Organizing the Tasks in Complex Design Projects, *ASME Conference on Design Theory and Methodology*, Chicago, IL, September, pp. 39–46.

Grose, D. L., 1994, Reengineering the Aircraft Design Process, *Proceedings of the Fifth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Panama City Beach, FL, Sept. 7–9.

Steward, D.V., 1981, The Design Structure System: A Method for Managing the Design of Complex Systems, *IEEE Transactions on Engineering Management*, **28**(3):71–74.

Yassine, A., Falkenburg, D. and Chelst, K., 1999, Engineering Design Management: An Information Structure Approach, *International Journal of Production Research*, **37**(19):2957–2975.