Product development process improvement: A review of Design Structure Matrix methods

Toufik Boudouh^{1*}, and Larbi Bendada²

¹ICB UMR 6303, CNRS, Univ. Bourgogne Franche-Comté, UTBM, Belfort, France

²Université Larbi Ben M'hidi, Oum El Boughi, Algeria

Abstract. Iteration is an important characteristic of the design process. The Design Structure Matrix (DSM) is a powerful method allowing modelling, analysis, and optimisation of iterations in the design process. Boosted by the concepts of concurrent engineering, it has been widely developed during the decades by researchers from both academia and industry. The objective of this paper is to provide a review of different DSM based techniques used to improve the performance of the product development process.

1 Introduction

Engineering design is crucial for the competitiveness of manufacturing companies. An optimal timing and scheduling of the design process is a critical factor in the successful time-to-market completion of the product development process. Design is a multidisciplinary process and then requires a large variety of approaches to be understood. Therefore, the existence of a great number of design process models should not be surprising, and we have to consider them as complementary rather than competing. Design is also a dynamic process because of to the continuous changes in the flow of information among its tasks.

In this work we will focus on the iterative aspect of the design process. Iteration is defined as the process by which a solution is approximated step by step [1]. It is also defined to be the repetition of design tasks due to the arrival or discovery of new information [2]. Iteration has been used as an important element in several research works for developing models of the design process. These models are used to make important managerial decisions in planning design processes.

The design structure matrix (DSM) is one of the methods used to model and analyse iterations in the design process. DSM is a network modelling tool used to model the constituting elements (tasks, components, people...) of a system (design project, product, organization...) and their interactions. In the last decades, an important research work has been undertaken to develop and apply DSM to a wide range of engineering management problems, including product design, systems engineering, project planning, and organization design. This led to the emergence of different types of DSMs. Browning [3] proposes a classification of DSMs into four categories: time-based DSMs with (1) activity-based DSMs and (2) parameter-based DSMs; and static DSMs with (3) component-based DSMs and (4) people-based DSMs. We propose to provide a review of the main DSM models found in the

* Corresponding author: toufik.boudouh@utbm.fr

literature. We present also some techniques used to improve the design process by manipulating the DSM.

The remainder of the paper is organized as follows. We provide in the next section a brief description of iterations in the design process and a review of models accounting for iterations. We present then the basic DSM model and its extensions. Some methods used to optimize the DSM are then presented.

2 Iterations in the design process

Design is an information processing activity. Because of the complex information dependencies that exist between design tasks, we cannot achieve the design process using a straightforward procedure. Therefore, iteration is fundamental to resolve design problems. We can observe two types of iterations in the design process: planned iterations, caused by coupled tasks or uncertainty; and unplanned iterations, caused by design errors or the introduction of new information. Understanding and controlling iterations can improve the design process, and reducing them would have positive effects for the product development cycle time. In many papers it was reported that iterations represent up to 70% of total development effort in a design process [4]. It is then important to consider the iteration aspect of design tasks when developing models of the design process. The introduction of concurrent engineering concepts, in order to reduce the product development time, has drawn attention to consider thoroughly the iterative aspect of the design process.

Several models of the design process that account for iterations are derived from the DSM method. One of these models is the sequential iteration model developed by Smith and Eppinger [2], in which they use linear systems theory to identify controlling features of iteration in a coupled development process. The authors developed also the work transformation matrix model [5], which has a fully parallel structure. In this model, the tasks are re-sequenced and an expected duration for each sequence is determined. While both of these models are useful in characterizing the two extreme cases of product development (parallel and sequential iteration) for any number of tasks, they do not model intermediate scenarios where overlapping might be more appropriate.

Carrascosa [6] developed a mathematical model based on characterizing the information exchanged between tasks in terms of probability of change and impact. DSM has been used in this model as an interface to graphically represent the information flow. The model provides a basis for determining the appropriate task sequence and degree of concurrency to minimize development time and cost.

Belhe and Kusiak [7] proposed a different matrix based methodology to define and sequence tasks by mapping their technical relationships. The model developed includes probabilistic OR and XOR (exclusive OR) relationships between tasks. In this model, the probabilities of executing one or more of the OR/XOR paths are dependent on the iteration number, and these probabilities are fixed in advance. In this model, the length of each task is fixed and deterministic and does not change as the iteration process progresses.

In addition to these models, other models of the iteration process in design do not use the DSM approach. The model developed by Boudouh et al. [8] presents the effects of parallel scheduling. Doing tasks in parallel allows the completion of the design process more quickly. On the other hand, doing tasks in series leads to less tasks requiring repetition, and therefore the development cost is lessened. The objective of the model is to determine how much parallelism is desirable, and whether minimizing development time justifies an increase in development cost.

Krishnan et al. [9] developed a model for overlapping nominally sequential tasks in order to reduce development lead time. In their model, the downstream activity begins with preliminary upstream information and incorporates subsequent upstream design changes in

future iterations. They present a framework to determine how to disaggregate design information and overlap consecutive stages based on the evolution and sensitivity properties of the information exchanged.

3 DSM models

3.1 Basic model

The original DSM was first developed in the 1970’s by Steward [10] to analyse information flow. The objective was to apply the square matrix to represent a network of design variable interactions in order to solve large systems of equations. The order in which the variables are solved is determinant to minimize the number of iterations needed to solve the problem. DSM was then picked up in the early 1990’s at MIT by Eppinger [11] and applied to a number of industry projects in the manufacturing industry.

The DSM is a matrix representation of the elements of a system and their interactions. If the considered system is the design process, the corresponding DSM will be a square (N * N) matrix, mapping the interactions between the set of N design tasks. In the DSM method, we assume that each design task can be modelled as an information processing task, using information supplied by other tasks, and creating information for other ones. The original DSM matrix is square with one row and one column per task. The tasks are listed in a roughly chronological sequence of execution. In a simple DSM, diagonal elements of the matrix are not used. Off-diagonal elements indicate information flows between design tasks. DSM was used to model different types of systems and not only the tasks of a design process. Hence, DSM can be used to model the product’s architecture. In this case, the elements of the matrix would be the components of the product and the interactions would be the interfaces between the components.

- Using DSM, we can represent four types of interactions between design tasks:
- Serial (or independent): Output of the upstream tasks is necessary to execute the downstream tasks, so they are executed sequentially.
 - Parallel (or independent): Tasks can be executed simultaneously. Notice that in case where the tasks use the same resource, even if they are independent, they should interact to share the same resource.
 - Coupled (or interdependent): Each task needs the information supplied by the other task. In this case, the tasks must be iterated until they converge to an acceptable solution.
 - Conditional (or contingent): The execution of downstream tasks depends on the result of upstream tasks. The DSM representation of conditional processes needs the adoption of specific notations.

Fig. 1. Relationships types in a DSM.

Figure 1 shows the four relationship types and their DSM representations. Reading across a row indicates all of the tasks whose output is required to process the task corresponding to the row, whilst reading down a column indicates all of the tasks which receive output from the task corresponding to the column. Upper diagonal elements of a DSM indicate the existence of cyclic information flows.

3.2 DSM models

The original DSM method proposed by Steward [10] does not contain quantitative information about the strength of interaction between the matrix elements. Several extensions have been therefore developed to allow quantitative analysis. Over recent years, DSM was used in different domains related to the design process, including product design, systems engineering, project planning, and organization design. This led to the emergence of different types of DSMs. A classification of the different DSM models was proposed by Browning [3].

3.2.1 Activity-based DSM

In this type, DSM is used to model the activities (or tasks) of a process and their interactions. Each task transforms one or more inputs into one or more outputs. This type is well suited to model the activities of the design process. The main objective of the model is to analyse the process architecture and determine the best schedule of the design tasks that will minimise iterations. The DSM elements are sequenced by using manipulation techniques called partitioning.

3.2.2 Parameter-based DSM

This type of DSM is used to model low level design tasks. The elements of the DSM are the design parameters that should be determined in a design task. For complex products, the determination of the value of one parameter depends on several other ones. So, the modification of one parameter will influence many others. The use of DSM will help in determining the sequence of parameter computations that allow minimum need for iterations. As in activity-based DSM, we use partitioning techniques to sequence the elements of the DSM.

3.2.3 Component-based DSM

This type of DSM is used to model the interactions between the different components of a product. Many interactions can be depicted through interfaces between the components. The analysis of component-based DSM results in defining clusters which are sets of components grouped on the basis of certain relationships. The manipulation techniques of the DSM used in this case are called clustering. The results of the analysis could be for example the determination of modular designs.

3.2.4 Team-based DSM

This type of DSM is used to model the interactions among units in an organisation. Organisations represent a network of people sharing a common objective such as the development of a new product. Analysing a team-based DSM highlights interteam relationships which leads to a better understanding of interfaces between teams and improve

coordination and communication. Clustering is the manipulation technique used to analyse team-based DSMs.

4 Reducing iterations in the design process

In this section, we present an example of the DSM analysis. We focus on time-based DSMs and especially task-based DSMs. After modelling the design process using the DSM technique, we obtain the as-is model in which the iterations are highlighted and visible in the matrix. It is then possible to reduce them by applying a technique called partitioning or sequencing. This is an operation that consists in reordering the DSM rows and columns such that the feedback marks are eliminated or minimized in the new DSM. Design activities or tasks are then rescheduled accordingly. Figure 2 represents an initial DSM of a design process with 11 activities.

Fig. 2. Example of an initial DSM.

Upper diagonal elements of the matrix represent the feedbacks or the iterations. In this example, we have 9 elements in the upper diagonal part of the matrix. They should be eliminated or minimized in order to improve the performance of the design process. By applying a partitioning algorithm, we obtain a new DSM (Figure 3 left) in which the number of upper diagonal elements is reduced to only 3.

There are several heuristics used to partition a DSM. They generally differ just in the way they identify the loops in the DSM. The heuristics attempt then to find the sequence of the design tasks so that each task do not begin until it receives all the input information needed to its execution from its predecessors. We can find in the following references the details of different techniques used to partition a DSM [12] [13].

Fig. 3. Partitioned DSM.

The use of partitioning heuristics does not always yield an optimal partitioned DSM. For our example of Figure 2, we can find another solution (Figure 3 right) that seems better than the one presented in Figure 3 left. When there is no quantification of the strengths of the relationships between tasks in a DSM, we should then try to have the remaining upper diagonal elements of the DSM closer to the diagonal. In this case the iteration process will be reduced. We have developed in [14] a new approach to partition a DSM. It is formulated as an integer problem and an optimal solution is possible to obtain.

5 Conclusion

We presented in this paper a quick review of DSM models. DSM is a powerful method and tool to capture the main interactions among the elements of a system. In the field of product development, DSM has been identified to be useful because it can generate and represent design process architecture, and, by manipulating operations (partitioning and clustering) it can improve the process, reduce cost, duration, and risk.

References

1. G. Pahl, W. Beitz, *Engineering Design: A systematic approach*, (Springer-Verlag, 1996)
2. R.P. Smith, S.D. Eppinger, *Manage. Sci.*, **43** (1997)
3. T.R. Browning, *IEEE Trans. Eng. Manage.*, **48** (2001)
4. T. Boudouh, C. Vernier, N. Lebaal, S. Gomes, *14th International DSM Conference*, Kyoto, Japan, (2012)
5. R.P. Smith, S.D. Eppinger, *CERA*, **6** (1998)
6. M. Carrascosa, S.D. Eppinger, D.E. Whitney, *ASME Design Automation Conference*, Atlanta, (1998)
7. U. Belhe, A. Kusiak., *J. Mech. Des.*, **118** (1996)
8. T. Boudouh, D. Noyes, *Int. J. Mech. Prod. Syst. Eng.*, **6** (2002)
9. V. Krishnan, S.D. Eppinger, D.E. Whitney, *Manage. Sci.*, **43** (1997)
10. D. Steward, *IEEE Trans. Eng. Manage.*, **28** (1981)
11. S.D. Eppinger, T.R. Browning, *Design Structure Matrix Methods and Applications*, (MIT Press, Cambridge, 2012)
12. S.H. Cho, S.D. Eppinger, *Proceedings of DETC2001/DTM-21691 conferences*, ASME, (2001)
13. S.H. Cho. *An integrated method for managing complex engineering projects using the design structure matrix and advanced simulation*, MS thesis, (MIT, 2001)
14. T. Boudouh, O. Grunder, *App. Mech. Mater.*, **760** (2015)