Design Process Sequencing in Product Development Process Using Design Structure Matrix and Particle Swarm Optimisation Techniques

Shajahan C A1, Subramanyan Embranthiri E2, Kumanan S3, Firoz N4.
1 Professor, Department of Mechanical Engineering, T K M College of Engineering, Kollam
2 Consulting Engineer, TCE, Bangalore
3 Professor, Production Engineering Department, NIT Tiruchirapalli
4 Assistant Professor, Department of Mechanical Engineering, T K M College of Engineering, Kollam

E-mail: cashaji@yahoo.com

Abstract. Product development (PD) is a complex business process of transforming new ideas into marketable new products, processes or services. PD process is characterized by information flow and iteration. Traditional project management techniques are limited in representing interdependent activities. Design Structure Matrix (DSM) is a powerful tool for identifying and managing information exchange and iterations. In this work the process is modeled with DSM and the design activities are optimally sequenced using a Particle Swarm Optimization (PSO) technique. The methodology is exemplified with a case example

Keywords: Product Development (PD), Design Structure Matrix (DSM), Product development iterative time (PDIT), Particle Swarm Optimization (PSO).

1. Introduction

A profitable and successful product is the decisive winning factor in fast changing and competitive market. The intense competition forces firms to develop better products at an increasingly rapid pace. Product development is a process that guides a new product from ideation through product launch. It is an interdisciplinary and complex activity that requires inputs from numerous knowledge domains [1]. Engineering products become increasingly complex and the markets of the future are expected to become even more competitive than today [2]. As a result, today’s large product development projects may involve a complex set of tasks that require the participation of many engineers, managers, technicians and other professionals over a period of several years. The rate of advance of any task in a product development project may strongly affect the progress of a number of other tasks in the same project. The real challenge for a product development project under such circumstances is to overcome the tremendous complexity involved in scheduling and executing a large number of interconnected and dynamic tasks [3]. Besides technical complexity, the managerial complexity that is necessary to manage the interactions between
various engineering disciplines allocated in different geographical locations, imposes additional challenges on the product development process. The dependency relationship among the activities in a product development project are of three types - independent, dependent, and interdependent [4]. Conventional tools like Program Evaluation and Review Technique (PERT) and Critical Path Method (CPM) are limited in application to represent interdependency. Researches have investigated DSM as a tool to identify and manage information exchange between activities [5] [6]. This work aims at bringing out an optimal activity execution sequence which will reduce the effect of iteration to a minimum level using DSM and PSO techniques.

1.1. Design Structure Matrix
Design structure matrix is a generic matrix based framework for information flow analysis and was introduced by Donald Steward. The basic representation of activity DSM is a square matrix containing a list of activities in the rows and columns in the same order in the matrix form. A DSM showing information dependency is shown in Figure 1(a) and 1(b). The order of activities in the rows or columns indicates the sequence of execution. The relationship between activities is represented with a ‘1’ or ‘X’ mark in the off diagonal cells. The activities have to be read along the column as “gives information to” and along the rows as “needs information from”. Marked cells above the diagonal represent iteration in the process.

Figure 1 (b) shows a typical DSM showing the information dependencies. This occurs when an activity is dependent on information from a task scheduled for a later execution. Such scenario often leads to rework and are undesirable. If any mark lies above diagonal, it implies that an assumption has to be made to execute the corresponding sequence.

A primary goal in basic DSM analysis is to minimize the number of feed backs and their scope by re-sequencing the activities to get the DSM into a lower-triangular form as possible. Partitioning and tearing are two main processes that can manipulate the matrix and transform the matrix into lower triangular form to the extent possible. It is the process of rearranging the order of activities by moving an entire row and column on either side in such a way that the resulting matrix has marks either below the diagonal or close to the diagonal. Through the manipulations, the task execution sequence will be reordered, the number of
iterations will be reduced, and fewer tasks will be involved in the iteration cycle and these will result in a faster development process.

There are different types of DSM like parameter based, component based, task based, numerical DSM etc. A numerical DSM can hold multi attributes like strength of dependence of task on specific data, sensitivity, evolution, variability accuracy of information etc. In this paper, a numerical DSM is used in which coupling marks are replaced with numbers (iteration factor). The strength of each coupling are quantified in the form of an iteration factor-representing the number of iterations required for convergence. Table 1 shows the available 7 coupling levels and their associated iteration factor default value.

| Coupling strength    | Iteration factor |
|----------------------|------------------|
| Extremely weak       | 2                |
| Very weak            | 3                |
| Weak                 | 4                |
| Nominal              | 5                |
| Strong               | 6                |
| Very strong          | 7                |
| Extremely strong     | 8                |

It is very useful in planning the activity sequences and also in identifying and managing information exchange. A major advantage of the matrix representation over other lies in its compactness. Moreover matrices are easy to manipulate and store in computer. The utility of a matrix representation lie in its ability to illuminate complex relationships between process elements in a compact, visual and analytically advantageous format (www.problems.com)[7].

1.2. Particle Swarm Optimization Algorithm

PSO is a population based optimization tool developed by Eberhart and Kennedy[8] and has been applied successfully in many areas like function optimization, artificial neural network, and fuzzy control system. Compared with other methods, the particle swarm concept was based on the premise of social behavior. A PSO algorithm mimics the behavior of flying birds and their means of information exchange to solve optimization problems. PSO has the advantage of being faster and cheaper. It can get the same quality results in significantly fewer fitness evaluations and constraint evaluations. The algorithm is intuitive and does not need specific domain knowledge to solve the problem. There is no transformation or any other manipulation needed to handle the constraints.

PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random feasible solutions and searches for optima by updating generations. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particle. Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. This value is called Pbest. When a particle takes into account of all the population it has already searched through, the best value is a global best and is called Gbst. At each time step, the velocity of (accelerating) each particle is changed towards its Pbest location. Acceleration is weighted by a random term, with separate random numbers being generated for
acceleration toward Pbest location. The velocity of each string can be calculated by using the equation ‘a’ and the velocity can be updated to the present population by using equation ‘b’.

\[
\begin{align*}
V_{id} &= V_{id} + C_1 \cdot \text{rand} \cdot (P_{id} - X_{id}) + C_2 \cdot \text{Rand} \cdot (P_{gd} - X_{id}) \\
X_{id} &= X_{id} + V_{id}
\end{align*}
\]  

Where, \(V_{id}\) is the velocity for the particle \(i\), represents the distance to be travelled by the particle from its current position.

\(X_{id}\) represents the particle position, \(P_{id}\) and \(P_{gd}\) is called the Pbest, the local best solution represents \(i\)th particle’s best previous position and \(P_{gd}\) is the Gbest, the global best solution.

\(C_1\) and \(C_2\) are two positive constants called accelerating constants, \(\text{rand}()\) and \(\text{Rand}()\) are two random functions in the range \([0,1]\). Equation (a) is used to calculate the particle’s new velocity according to its previous velocity and the distances of its current position from its own best experience (position) and the group’s best experience. Then the particle flies toward a new position according to equation (b). The performance of each particle is measured according to a pre-defined fitness function, which is related to the problem to be solved.

2. Proposed PSO Algorithm

The solution procedure for proposed PSO algorithm for Design process sequencing is developed in Turbo C. The flowchart of the process is given in Fig.2

2.1 Encoding scheme and initial swarm

Finding a suitable configuration between the problem solution and a PSO particle is a key factor in implementing the algorithm. This work encoded a feasible sequence of activity as a PSO particle. For a ‘\(n\)’ activity PD project, the PSO particle will be an array of ‘\(n\)’ segments, each segment containing different activity numbers in such a way that no activity could assume the same order. Or in other words activity repletion in the sequence is not permitted. The other hard constraints such as logical precedence relationship, fixation of first and last activities (if they are known) etc can be tailored according to the nature of the problem while generating feasible solutions. Hence in this problem, the initial swarm will be a set of \(N\) feasible sequences where \(N\) is the size of the swarm.

2.2 Fitness function

The circuits or feedback cycles require a finite number of iterations for its convergence and the associated time is called coupling iterative time (CIT). Product Development Iterative Time (PDIT) is nothing but the sum of all CITs. When the activities are changing its order of execution, the PDIT also changes accordingly. PDIT is the fitness function considered in this work. A heuristic is used to calculate the PDIT once the DSM is formed according to a particular sequence.
Figure 2. Proposed PSO algorithm for design process sequencing
2.3 Heuristic used for calculating PDIT

\[
\text{PDIT} = 0; \quad \{ \text{Initial project iterative time} \} \\
A = \{(X, T_x); 1 \leq X \leq N \} \\
C = \{(\text{Coup}(i, j), IF_{i,j})\}
\]

Repeat
Select_Coupling(coup(i, j));
\(CT_{i,j} = 0;\) \{ set initial coupling time \}
\(CIT_{i,j} = 0;\) \{ set initial coupling iterative time \}
\(k = 1.\) \{ set counter initial value \}
Repeat
Select activity(k);
If \(X_i \leq X_k \leq X_j\)
Then \(CT_{i,j} = CT_{i,j} + T_k\)
k = k + 1;
until \(k > N;\) \{ all activities in A are considered \}
\(CIT_{i,j} = CT_{i,j} \times IF_{i,j}\)
\(\text{PDIT} = \text{PDIT} + CIT_{i,j}\)
Until all couplings in C are considered;
End

A \quad \text{Set of activities in DSM and their associated times.}
N \quad \text{Number of activities}
C \quad \text{Set of feedback couplings and their associated iteration factors}
Coup(i, j) \quad \text{Coupling from activity i to j}
IF_{i,j} \quad \text{Iteration factor of coup(i, j)}
Select_coupling \quad \text{A function that selects a coupling from set C}
Select_activity \quad \text{A function that selects an activity from set A}
Tk \quad \text{Time associated with activity k.}
A hypothetical PD process consists of 5 activities and 2 couplings are considered for the illustration as shown in Fig 3.a. and 3.b. Let the initial activity sequence is 1-2-3-4-5 and the corresponding activity times are 40,10,30,20 and 50 respectively. It consists two couplings or loops namely C1,3 and C2,5 respectively. The iteration factors considered for couplings C1,3 and C2,5 are IF1,3 = 4 and IF2,5 = 6 respectively.

\[
\text{CIT for the first coupling C1,3} = (40+10+30) \times 4 = 320.
\]
\[
\text{Similarly CIT for C2,5} = 660
\]

Hence the associated PDIT = 980

By resequencing the activities as 1-3-2-4-5, the associated PDIT is calculated as 
\[
(40+30) \times 4 + (10+20+50) \times 6 = 760
\]

3. Implementation of the proposed PSO algorithm

The implementation of the algorithm involves the following steps.

Step 1: Input the number of activities, activity duration, iteration factors, constraints and the information dependencies.

Step 2: Generate the initial swarm which consists of N number of feasible solutions or particles. Each particle represents a feasible sequence which confines to the constraints. In the problem considered, first and last activities are fixed as hard constraints. Other hard constraints can be tailored according to the nature of the problem.

Step 3: For each feasible sequence (particle) corresponding DSM is modeled using the input data and the fitness of each particle (corresponding PDIT) is calculated. The Pbest and Gbest are to the identified. The sequence which is having minimum PDIT among the swarm of particles is selected as the Pbest. The Gbest corresponds to the sequence with minimum PDIT so far searched through.

Step 4: The velocity and position of each particle in the swarm is updated via Equation (a) and (b).

Step 5: Repeat step 3 and 4 till the termination criteria met.
3.1 Setting of Parameters for the proposed PSO algorithm

The acceleration constants C1 and C2 in Equation. (a) adjust the amount of tension in the system. High values cause rapid movement towards the targeted regions. According to the experiences of other researchers, the values of C1 and C2 are taken as 2 for all the examples figured in this work. Through the application of Equations (a) and (b) the particle may overshoot the problem space. That is absolute value of Vid and Xid may be great. So in this work the maximum value of velocity Vid, denoted by Vmax, is set to ‘n’ (number of activities). Vid is a value in the range [-n,n]. The maximum value of Xid, denoted by Xmax is set to ‘n’. Since Xid represents an activity number, it must be a positive integer. Thus Xid is an integer value in the range [1, n]. If the computational value of Xid is happened to be a real number, we round off it to the nearest integer. In this way we convert a continuous optimization problem to a discrete one.

The following are the values taken for different parameters of PSO algorithm.

Initial swarm size = 40; Cognition learning rate C1= 2; Social learning rate C2= 2 and Termination criteria is 20 iterations or no improvement in fitness function for 3 consecutive iterations, whichever occurs first.

4. Case Example

For the validation of the developed algorithm, a bench mark problem is taken from literature [9]. The project was taken from a larger conceptual design project. It consists of 22 activities and 39 couplings. The process is modeled using DSM and is shown in Figure 4. The iteration factor values are taken as shown in Table 1. By applying the proposed PSO algorithm, the activities are sequenced by keeping the logical constraint (that first and last activities are fixed.). The optimized sequence is obtained as 1-18-17-7-11-4-16-21-12-13-19-9-15-20-3-2-10-5-8-14-6-22. The DSM representation is shown in Figure 5 and the associated PDIT is 9560 units.
5. Conclusion

Iteration is inherent in PD process and has significant influence over development time and cost. Carrying out the activities in the optimum sequence will lead to considerable savings in terms of development time and cost since it minimizes iteration. This research work proposed a method which uses a DSM to model the design process and a PSO algorithm to obtain an optimized activity sequence. The method is capable of tolerating hard logical constraints in the activity sequence. The developed method is validated with a case example.

References

[1] Maoi, A. D., Roberto, Van Corso M. (1994), “Multi project management framework for new product development”, European journal of operational research, Volume 78, pp 178-191.

[2] Gohlich D, Hildebrand and Schellert D, D (2018) “Augmented DSM sequencing to support product development planning”, International Design Conference 2018. Pp 1139-1148.

[3] Chun-Hsien Chun, Shih Fu Ling, Wei chen, (2003), “Project Scheduling for Collaborative Product Development Using DSM”, International Journal of Project Management, Vol. 21, pp 291-299.

[4] Yassine A, and Braha D (2003), “Complex concurrent engineering and the design structure matrix method”, Concurrent. Engineering Management 11(3), pp 165-176.

[5] Steward DV (1981), “The Design Structure system: a method for managing the design complex system”, IEEE Transactions on Engineering Management, Vol. 28(3), pp 71-74.

[6] Eppinger S, D, Whitney, D E, Yassine A A and Romer T (2001) “The MIT design structure matrix – DSM homepage.” <http://web.mit.edu/dsm/>.

[7] Problematics Online ,http://www.problematics.com

[8] Kennedy, J., and Eberhart, R. C. (1995), Particle Swarm Optimization, Proc. IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, IV: 1942-1948.

[9] Rogers James L (1997), “Reducing design cycle time and cost through process resequencing” International conference on engineering design(ICED), Aug 19-21.