项目进度管理 2018

Dr. Huang Dan
huangdan@sjtu.edu.cn

Homework Assignment 2
Task-based Design Structure Matrix

Out: Oct 6, 2018
Due: Oct 27, 2018

Learning objectives

In this homework you will apply the Design Structure Matrix (DSM) method to a hypothetical design project. You will first learn how to translate the project graph from the previous assignment to a DSM representation. Next you will add iterations to the project and analyze their effect on the previous task sequence. You will consider partitioning the DSM to reveal meta-tasks. Finally you will estimate the effect of these changes on the critical path and estimated project completion time.

Resources

[1] Steven D. Eppinger, Daniel E. Whitney, Robert P. Smith, and David A. Gebala. "A Model-Based Method for Organizing Tasks in Product Development", Research in Engineering Design. Vol. 6, No. 1, pp. 1-13, 1994.
[2] Steven D. Eppinger, Murthy V. Nukala, and Daniel E. Whitney. "Generalized Models of Design Iteration Using Signal Flow Graphs", Research in Engineering Design. Vol. 9, No. 2, pp. 112-123, 1997.
[3] DSM website: http://www.dsmweb.org/
[4] DSM software: http://www.problematics.com (PSM32)
[5]draft chapter (Design Structure Matrix Method)

Situation

You are still Project Manager for the NMA-X1 UAV development project described in HW1. You have successfully completed the requirements definition (b, 10) step, effectively the first task in the process. You realize that your previous CPM plan was useful, but did not capture a number of important design iterations.

Assignment

1. Construct a Design Structure Matrix (DSM), with dimension 24x24, including all NMA-X1 tasks *a* through *x* (including dummy tasks "start" and "finish"). This DSM should capture the same task dependencies as the project graph from HW1. See Appendix I for an initial DSM template that you could use to get started. Comment briefly on the structure of the completed matrix.

An example of a binary DSM is shown in the DSM Chapter, Fig. 10.9.

2. During the requirements review (b) you discussed the upcoming design project with your functional team managers and your suppliers. You realize that your previous CPM plan is useful, but does not capture some important task interdependencies:

- Avionics design (f) requires information from software development (h) in terms of properly sizing the avionics bus and controllers for processing speed, memory and communications bandwidth
- In order to decide what the power budget between the engine and the payload should be, there must be some concurrent development. This means that UAV engine development (i) and payload development (j) require data from each other.
- In previous vehicle projects there were structural misalignment problems between the fuselage, wing and empennage. In order to avoid this on the NMA-X1 project, tasks (k) and (l) must be coupled together (i.e. they feed information to each other).
- Software specification (g) and software development (h) are really iterative.
- Experience from other programs has shown that significant errors and non-conformances are often discovered during avionics/software integration (q). This can require substantial rework from task (q) to software development (h).
- Integrating the vehicle (s) is never a single step process and data from past projects indicates that adjustments must be made to power system integration (o) due to problems discovered during (s).
- Finally, despite early efforts to avoid collisions between fuel lines, electric wire harnesses, air ducts and trim panels it is a frequent occurrence that internal fittings (m) need to be adjusted based on interferences discovered during final vehicle assembly (t).

Modify the DSM to reflect these iterations and briefly comment on the changes.

3. Convert the DSM to a binary matrix using only {1,0} and multiply the matrix by itself a number of times. Describe the result of each sequential multiplication and draw conclusions for your vehicle development project. The purpose of this step is to identify the loops in the project of length 2 (two tasks coupled) and/or larger…. How many iteration loops are there in the project? What tasks are involved in each loop?
   HINT:
   Use MATLAB to solve this problem. Do the operation of square, cube, power of 4, etc.

4. Apply the manual partitioning algorithm we discussed in class by swapping columns and rows of the DSM in a deliberate fashion. Attempt to improve the task sequence of the project such that feedback loops (entries above the main diagonal of the DSM) are minimized and appropriate meta-tasks emerge. Identify which tasks can be done sequentially, in parallel and which ones must be worked on iteratively. Highlight the coupled meta-tasks by enclosing them in a highlighted box. How many iterations remain in the upper right triangular part of the DSM after you have re-sequenced? Discuss briefly how this new way of modeling the project might change your view and focus relative to the original CPM project plan in HW1. Note: you may compare your

results with an automated algorithm such as the one available in PSM32 but you should be aware of what is happening and manually check the solution.

HINT:

The partitioning algorithm works as follows:

1. Move all tasks that don't require inputs from other tasks to the top
2. Move all tasks that don't produce outputs for other tasks to the bottom.
3. Group coupled tasks together until a group has either no input or no output requirement
4. Repeat steps 1 or 2 for each group as appropriate
5. Finish when tasks are clustered such that the number of off-diagonal terms above the diagonal is minimized

5. Compute the Visibility Matrix V (see Lecture slide) for the DSM using as many terms as you feel are appropriate and determine which is the most influential task in the project (in terms of impacting most other tasks). Explain why this is and if you agree with this finding.

6. Estimate a new expected finish date for the project (after the January 2, 2012 start) with your new, reordered DSM by assuming that typically each iteration loop has to be done three times before convergence has been achieved. However, you can also assume that every time a task has to be repeated it will only take half the time it took the previous time that task was executed. Answering this question will probably require you to write and execute a discrete event simulation of the project, but you might be able to find another, more approximate way to obtain a reasonable result.

   How does the expected project duration (with iterations) compare against the total project completion time you had predicted in HW1? Why is there such a large difference if there is one? Which meta-tasks identified in question 4 are driving the overall project duration?

- **Turn in your answers with the rubric attached as the cover sheet.**

Design Structure Matrix (DSM) **Appendix I**

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x
a	a																							
b		b																						
c			c																					
d				d																				
e					e																			
f						f																		
g							g																	
h								h																
i									i															
j										j														
k											k													
l												l												
m													m											
n														n										
o															o									
p																p								
q																	q							
r																		r						
s																			s					
t																				t				
u																					u			
v																						v		
w																							w	
x																								x