14TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELLING CONFERENCE, DSM'12
KYOTO, JAPAN, SEPTEMBER 13 – 14, 2012

EFFECIVE & EFFICIENT DSM CONFIGURATION GUIDELINES FOR LOW-COST DEVELOPMENT OF COMPLEX SYSTEMS

Sadegh Mirshekarian B.[1]

[1]Sharif University of Technology, Tehran, Iran

ABSTRACT

With the proliferation of more complex systems has come the need to find better solutions in both technical and management domains. Such complex systems are usually larger in size, have more parallel operations and contain more complex interfaces (Eisner, 2005). The Design Structure Matrix is a very useful tool in handling such complexities, provided that the system designer can use it properly. This paper addresses how effectiveness & efficiency are defined for a DSM and how these two important characteristics can be achieved. The importance of understanding the solution space in constructing an effective & efficient DSM is discussed and general guidelines are given on configuring the DSM as such. Most discussions in this paper are valid for all system and DSM types, but the research is mainly built around a parameter-based DSM for use in low-cost space development.

Keywords: Design Planning, Complex Systems, Solution Space, Low-Cost Space Projects, Space Systems Engineering, Effective & Efficient DSM

2 INTRODUCTION

Even though the underlying concept of a Design Structure Matrix is relatively simple, effective and efficient use of it is not. It depends primarily on properly tackling two big problems in DSM configuration. First, choosing the best set of DSM elements, and second, orchestrating their interdependencies in the most suitable way. Both of these problems can be challenging, especially when the system being designed is complex, and if the system designer is not aware of their solutions' significance and contribution to the final result.

2.1 DSM elements set

Considering a parameter-based DSM, the elements of the matrix are design or system parameters, chosen from the system's *parameter pool*. The parameter pool is defined here as the set of all the parameters of all the elements of a system, at all available levels of abstraction. Figure 1 shows the system and its first two levels of abstraction along with its parameter pool.

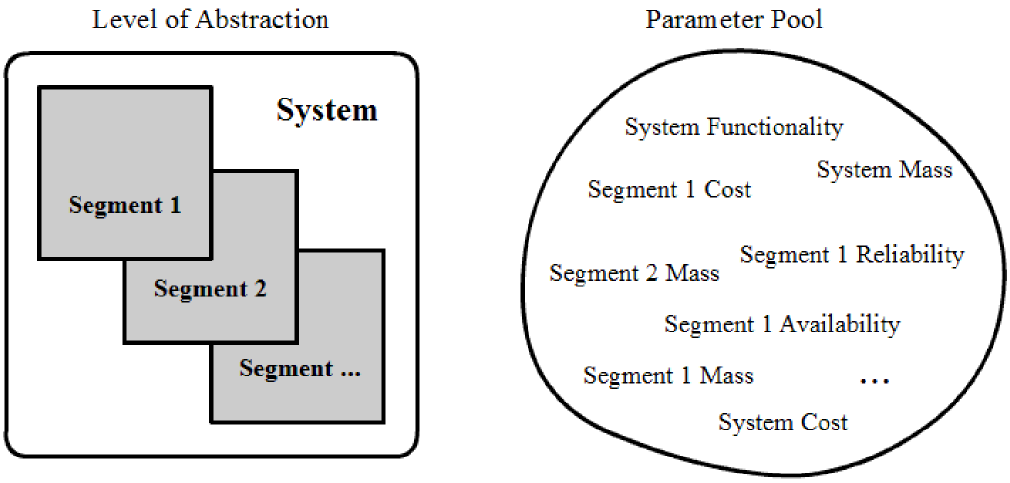

Figure 1. The parameter pool of a system at two levels of abstraction

Evidently the parameter pool of a complex system becomes enormous very quickly, especially as the design matures and more detailed levels of abstraction are introduced. An obvious question is then, how many of these parameters, and which ones, should be represented in the DSM. To answer this, one should note that:

- Using too many parameters and forming a detailed DSM can improve its effectiveness, but it will also reduce its efficiency. It's because handling a large DSM consumes lots of time and energy while interpreting the processed DSM (e.g. the partitioned DSM) may become really cumbersome.
- Using too few elements has the reverse result. Higher efficiency but lower effectiveness.

Therefore the first big problem of DSM configuration is choosing the right set of DSM elements from the pool of possible members (the system's parameter pool in case of a parameter-based DSM).

2.2 DSM interdependencies

When the DSM elements are chosen, another problem would be to arrange their interdependencies. Despite common perception, interdependencies between DSM elements (parameters, physical entities, tasks or whatever they are) are not intrinsic when the system is under design, but chosen by the system designer. During design, if it is decided that element A depends on element B for example, this means that the requirements of the element B have a higher priority than those of the element A and that the design of A should be adjusted based on the design of B (consider Figure 2).

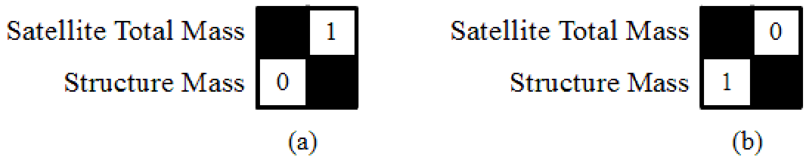

Figure 2. Example interdependency cases

In case 'a', the satellite structure mass depends on the satellite total mass. This means that the total mass governs the structure mass, in that the structure mass should be adjusted based on the value (or any other limitations) given to total mass. In case 'b' the dependency is reversed and the structure mass governs the total mass, so now the total mass should be adjusted based on the value given to the structure mass. The simple point here is that although the satellite's total mass seems to naturally depend on the structure mass, because the total mass is calculated from the mass of all physical subsystems, the dependency is a choice for the designer to make. In fact, what might be confused with the design interdependency is *intrinsic interrelationship*. The satellite total mass and the structure mass are intrinsically entwined concepts due to their definition, but their interdependency is a design decision.

3 THE SOLUTION SPACE

Facing up to the two big problems in DSM configuration is easy, only in the presence of a good understanding of the solution space. The solution space is defined as a bounded abstraction that represents a capability and level of performance that, when implemented, is intended to satisfy all or a portion of a higher level problem space (Wasson, 2006). In this view, each system entity is in fact a solution to its related problem (i.e. it satisfies its requirements), and therefore the solution space of each problem is in fact the set of all possible combinations of all possible values of that entity's parameters.

Understanding the solution space for each entity, gives the system designer an idea of that entity's design flexibility, significance and maximum suitability. He then can decide which elements and from what level of abstraction can remain in the DSM and how their interdependencies should be arranged.

4 CONFIGURATION GUIDELINES

From the point of view of this paper, the most efficient parameter-based DSM is a lower-triangle DSM with minimum elements and the most effective DSM is one that represents the design problem in maximum detail. Unfortunately, based on these definitions the requirement for a DSM to be both effective and efficient is a contradictory one, and trade-offs are inevitable.

However, there are guidelines as presented here, that can promote this trade-off process. They are built around the proper assessment of the solution space *flexibility* and *significance* of each system entity and its defining parameters. Please note that both the elements and their interdependencies should be chosen simultaneously because they are not independent features of the DSM.

4.1 Solution space flexibility

Solution space flexibility determines the degree to which an entity can be played with in the hands of the designer, or in fact the degree to which an element's design can be adapted to the rest of the system (flexibility enables adaptability). The flexibility rating is defined in four levels here, as shown in Table 1, and it is attributed to the solution space based by the three factors that follow.

Table 1. Flexibility rating definitions

Flexible	Most of the entity's decisive parameters are flexible or can have continuous values
Fairly Flexible	A few of the entity's decisive parameters are flexible or can have continuous values
Fairly Rigid	Some of the entity's decisive parameters have separate discrete options and the rest are rigid
Rigid	Only off-the-shelf options are available for the whole entity

4.1.1 Availability of the entity

Availability is defined as the number of solutions available in the solution space of an entity, regardless of their suitability and practicality. Note that by calling the possible parameter sets "solutions", we imply that they satisfy a problem, so the availability of an entity is bounded to its related problem space [2]. Consider the example in Table 2, which gives each entity a flexibility rating based on the definitions given in Table 1. Please also note that a proper judgement is necessary here, on the part of the designer, because the definitions given in Table 1 do not automatically fix the flexibility rating for all system entities.

Table 2. Example entities solution space flexibility assessment

Entity	Availability	Flexibility
X-Axis Reaction Wheel	Some off-the-shelf options RW1, RW2, RW3 and RW4, plus an option to buy off-the-shelf motors and design the wheel assembly	Fairly Flexible
Propulsion Subsystem	Only total mass limited to 40 kg	Flexible
UHF Antenna Mechanism	Only one off-the-shelf option	Rigid
Propellant	Anything more than 10 kg of the chosen propellant type	Fairly Flexible
Main Thruster	Few off-the-shelf options	Fairly Rigid

4.1.2 Practicality of the solutions

Not all available solutions are equally practical, and an assessment of each solution's practicality should be performed by evaluating its cost, performance and schedule characteristics. This will update the flexibility ratings determined by just assessing availability, into the ones shown in Table 3.

Table 3. Practicality updates for the flexibility ratings

Entity	Practicality assessment	Flexibility
X-Axis Reaction Wheel	RW1 and RW3 are the only practical options	Rigid
Propulsion Subsystem	Only cold gas propulsion practical, total mass limitation still applies	Flexible
UHF Antenna Mechanism	The only available option is practical	Rigid
Propellant	Under 20 kg and above 45 kg is unpractical	Fairly Flexible
Main Thruster	Only one option practical	Rigid

4.1.3 Intrinsic interrelationships

Consider the propellant entity in the example above. The availability and practicality assessments have given it a Fairly Flexible rating, but there are still other limitations on its solution space. Since the conceptual design constraints exist on the total mass of the propulsion subsystem, and the mass of the propellant is by definition part of this mass, an indirect constraint exists on the propellant mass as well (the constraint applies to the propulsion subsystem too because its minimum is limited by the propellant mass). This will change the propellant solution space into a mass range of 20kg to 25 kg for example, due to an intrinsic interrelationship between the two entities and based on the designer's judgement, this may reduce the entity's rating to Fairly Rigid.

It should be mentioned that an entity's solution space flexibility is in fact a net effect of the flexibility of the parameters that define that entity. A more detailed approach to determining the flexibility rating for an entity would be to first determine the flexibility of each of its parameters, and then to calculate that net effect. In the example above, this approach has been shortened, but as it will be discussed below, each parameter's flexibility matters separately when configuring the DSM.

4.2 How the flexibility rating affects DSM configuration

The flexibility rating helps tackle both of the big problems in DSM configuration, as discussed below.

4.2.1 Element selection

When the flexibility rating of an entity is Rigid (or Fairly Rigid with only off-the-shelf options), the whole entity should represent a single DSM element as shown in Figure 3. This way we save a lot of processing resources and lose very little effectiveness. In case there is only one option for the entity, it can be omitted from the DSM completely, unless (and it is recommended that) the system designer keeps it to increase DSM readability.

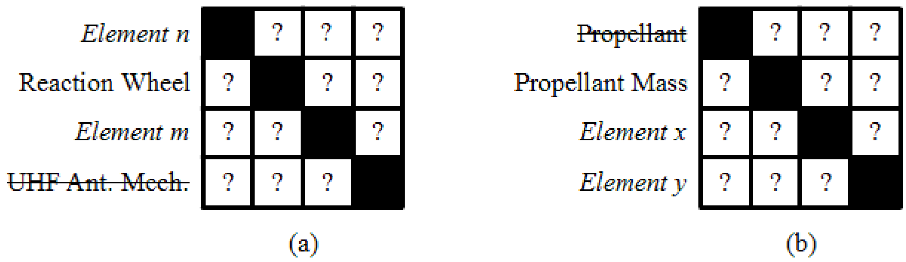

Figure 3. a) Off-the-shelf entities' DSM representation, b) Propellant entity's DSM representation

For other flexibility ratings a closer inspection of the parameters comprising the entity is required. The general rule is that (when there is at least one flexible parameter in the entity) the decisive parameters that are more flexible should be represented as individual DSM elements and those that are rigid or depend on the other more flexible ones should be grouped into a single element. Here again if the grouped elements have only one possible combination per design of the other elements, they can be omitted from the DSM altogether. Figure 3.b shows how to represent the propellant entity of our example.

4.2.2 Interdependency arrangement

A general rule to arrange interdependencies is to note that the more rigid a DSM element is, the less dependent it should be on other elements. In the extreme case of a rigid element with only one possibility, like the UHF Antenna Mechanism in the example, it should not depend on any other elements at all (that's why it can be omitted from the DSM).

Very often there are couples of DSM elements that are related intrinsically and a decision should be made as to which one should depend on the other or should there be a cyclic dependence between them. In such cases again, the one which is less rigid should depend on the one which is more rigid. If they are both equally flexible, a cyclic dependence would be best. Consider Figure 4 for example. All the shown elements are intrinsically interrelated and because the propulsion is the most flexible, it is decided that it should depend on the rest. Also the rigid element do not depend on any other elements at all.

Propellant		1	0	1
Propellant Mass	0		0	1
Main Thruster	0	0		1
Propulsion Subsystem	0	0	0	

Figure 4. Element interdependencies are arranged based on flexibility rating

4.3 Element significance

Besides solution space flexibility, which is the dominating factor in forming a DSM, the significance of an element relative to other elements can also be used to fine tune the DSM. This significance can be measured by looking at current element interdependencies. The more elements depend on element X, the less dependent X should be in general. This way the design flow would include less cycles, hence a more efficient DSM. Nevertheless it should be noted that changing the interdependencies based on significance should be done with care, since this process tends to make the DSM more efficient, while being rather blind to the other more important design considerations.

5 CONCLUSIONS

Applying the Design Structure Matrix to any design problem embodies two major challenges: how to choose the best set of DSM elements and how to arrange their interdependencies, to simultaneously achieve an efficient and effective DSM. Based on an understanding of the solution space, two concepts are introduced to serve as guidelines in tackling these challenges. First the solution space flexibility, which can be determined using the solution's availability, practicality and intrinsic interrelationships with other solutions, and second, the DSM elements' significance, which is assessed in a cyclic procedure, based on the element interdependencies. However, it should also be mentioned that the system designer's judgment still matters and these guidelines cannot fix the answers to the two big problems by themselves.

REFERENCES

Eisner, H. (2005). *Managing Complex Systems, Thinking Outside the Box*. New Jersey: John Wiley & Sons.

Wasson, Charles S. (2006). *System Analysis, Design and Development*. New Jersey: John Wiley & Sons.

Danilovic, Mike and Tyson R. Browning (2007), Managing Complex Product Development Projects with Design Structure Matrices and Domain Mapping Matrices. *International Journal of Project Management*, 25(3): 300-314.