24th INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2022
EINDHOVEN, THE NETHERLANDS, 11 – 13 October, 2022

Aggregation of multiple-level DSMs: Key challenges and some tentative solutions

Ali Yassine[1], Nicolay Worren[2], Tore Christiansen[3], Kim Soldal[3]

[1]Stevens Institute of Technology,
[2]Norwegian University of Life Sciences,
[3]Reconfig AS

Abstract: Team-based DSMs sometimes contain data representing three or more system levels (e.g., role/individual, team, department, organization). The question that then arises is whether and how lower-level units (e.g., teams) may be aggregated to higher level units (e.g., departments) in order to simplify clustering and analysis. In this paper, we describe the key challenges related to aggregation and outline some tentative solutions that may attempted.

Keywords: Design Structure Matrix, Aggregation, Organization design

1 Introduction

Modern engineering involves the development of large and complex systems (Sheppard et al., 2009). Traditionally, the technical complexity of these systems has been addressed through architectural decomposition (Yassine & Braha, 2003). Decomposition is a central concept in the analysis and management of complex systems. Once decomposed, a complex system can be visualized as a set of modules and components that interact together to provide some utility to customers.

Just like physical products, organizations can also be treated as complex systems, comprised of interacting elements such as people, teams and departments. When decomposition is utilized to manage complex systems, an important decision must be made regarding the level of granularity to be used. In a way, the level of granularity is the strength of the lens used for the analysis.

In the product development literature, a commonly used representation of a system is the Design Structure Matrix (DSM), which is a matrix representation of a network or graph. Although the DSM has been shown to be useful in displaying systems decompositions and preparing them for analysis, it has also been shown that these DSMs suffer from some shortcomings due to their limited ability to represent complex multi-dimensional relationships in a two- dimensional matrix (Sharman & Yassine, 2004). Also, it is uncertain how the level of granularity of the decomposition affects the results and conclusions of the architectural analysis (Chiriac et al., 2011). Figure 1 provides an illustration of how the DSM is a reflection of the real complex system and how zooming into and out of this reflection can provide different levels of granularity and detail.

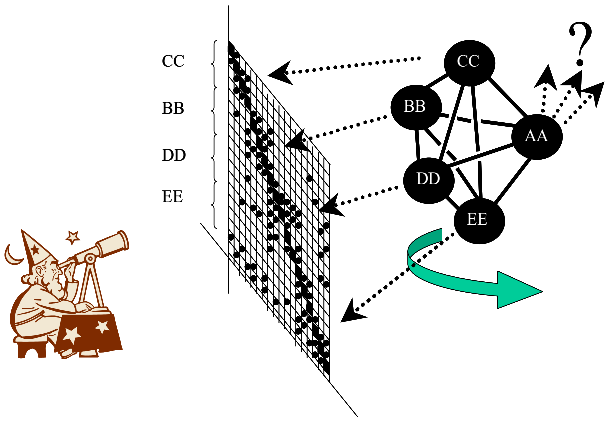


Figure 1. DSM as a reflection of a complex system.

To explore the notion of granularity, we need to recognize that DSMs sometimes contain data representing multiple levels in a system. Furthermore, that it is possible to aggregate lower level decompositions into higher level ones to arrive at the right granularity level. For example, a team-based DSM may contain data at the individual level (i.e., representing employees), the team level (i.e., employees grouped into teams), as well as the department level (i.e., teams grouped into departments). As an example, Figure 2 shows an organization that is functionally organized into two departments: Software engineering and Sales & Marketing (this is a hypothetical but realistic scenario for a software firm, where the interaction marks are carefully placed to represent typical communication patterns within such an organization). The DSM contains 24 elements (representing roles/individuals) that are grouped in 8 teams. This matrix may be condensed into a smaller one by aggregating the individual level information to the team level as shown in Figure 3.

Department	Product	Department	Team	Sub-team		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
A	A	Software engineering	Development A		1			X																					
A	A	Software engineering	Development A		2	X								X															
A	A	Software engineering	Development A		3		X						X																
A	B	Software engineering	Development B		4					X	X																		
A	B	Software engineering	Development B		5												X												
A	B	Software engineering	Development B		6					X						X													
A	A	Software engineering	Product management A		7								X																
A	A	Software engineering	Product management A		8									X						X									
A	A	Software engineering	Product management A		9	X						X							X										
A	B	Software engineering	Product management B		10				X							X	X							X					
A	B	Software engineering	Product management B		11					X					X		X								X				
A	B	Software engineering	Product management B		12						X				X	X													
B	A	Sales and Marketing	Sales A	Sales rep. 1	13														X			X					X		
B	A	Sales and Marketing	Sales A	Sales rep. 2	14								X							X		X				X			
B	A	Sales and Marketing	Sales A	Sales rep. 3	15							X							X										
B	A	Sales and Marketing	Sales A	Sales support 1	16													X				X							
B	A	Sales and Marketing	Sales A	Sales support 2	17														X		X								
B	B	Sales and Marketing	Sales B		18										X									X					X
B	B	Sales and Marketing	Sales B		19											X									X			X	
B	B	Sales and Marketing	Sales B		20																			X					
B	A	Sales and Marketing	Marketing A		21														X								X		
B	A	Sales and Marketing	Marketing A		22													X								X			
B	B	Sales and Marketing	Marketing B		23																		X						X
B	B	Sales and Marketing	Marketing B		24																			X				X	

Figure 2. Design Structure Matrix (DSM) showing interactions between individuals in an organization with two departments that are further developed into teams.

Department	Team	Product	A	B	A	B	A	A	B	A
Software engineering	Development A	A			X					
Software engineering	Development B	B				X				
Product	Product management A	A	X				X			
Product	Product management B	B		X				X		
Sales & Marketing	Sales A	A			X				X	
Sales & Marketing	Sales B	B				X				X
Sales & Marketing	Marketing A	A					X			
Sales & Marketing	Marketing B	B						X		

Figure 3. An aggregated version of the DSM shown in Figure 2.

There are three reasons why aggregation may be considered.

First, analysis of DSMs is highly processing intensive. As an example, we have implemented the Yu et al. (2007) algorithm in a software tool (Worren et al., 2020). Despite using 18 servers with parallel processing, it may take several hours for the tool to return a solution if the DSM contains 200 or more elements. By first aggregating such a DSM instead, one would drastically reduce the processing time (and cost, if server capacity is rented and charged based on utilization).

Secondly, although the general trend seems to be toward the use of complex models to analyze large data sets for analysis (i.e., analytics and Big Data), there is actually evidence that using less data and simpler models sometimes result in better and more robust results. As an example, by using a single data point (the infection rate from the previous week), Katsikopoulos et al. (2022) were able to outperform the predictions made by Google Flu Trends, which made use of approximately 160 different factors. Similar examples can be found in multiple fields including medical diagnostics, financial investment, and security (Gigerenzer, 2022)

Third, from a logical point of view, one can argue that use of an aggregated DSM for clustering is warranted in situations where the design decision relates to a higher rather than lower level of analysis. In general, clustering by means of a DSM implies that there is data one level below the level that is considered for re-design. For example, in the DSM of Figure 2, if the challenge facing decision makers is to determine how the teams should be grouped into departments (e.g., whether they should be grouped functionally, as in Figure 2, or by product or some other criterion), we must have information about the elements one level below the department level (i.e., at the team level). But we do not necessarily need information from the level further down (i.e., individuals).

The objective of this paper is to explore whether organizational decompositions using DSMs can be consistently analyzed at different levels of granularity through aggregating larger DSMs into smaller ones for ease of data collection and analysis. That is, will the conclusions / recommendations from a DSM study change at different levels of granularity for the organizational decomposition represented in the DSM? Additionally, we want to explore some universal rules for DSM aggregation.

2 Challenges

The overall question is whether one can condense a large matrix into a smaller one with fewer elements without a significant loss of information. But it has so far been unclear how one can make such an assessment. Hence there is a need to identify a procedure for testing and evaluating the impact on decision making of using full versus aggregated matrices.

One potential issue is that organizations may have an inconsistent hierarchical structure, as illustrated in Figure 4. When analyzing such organizations, it is challenging to deal with units that are not directly represented at the level of analysis. As an illustration, in Figure 4, the analysis is done at the department level, but Division 3 does not have any departments. Additionally, Department 2 (in Division 1) is divided into two teams.

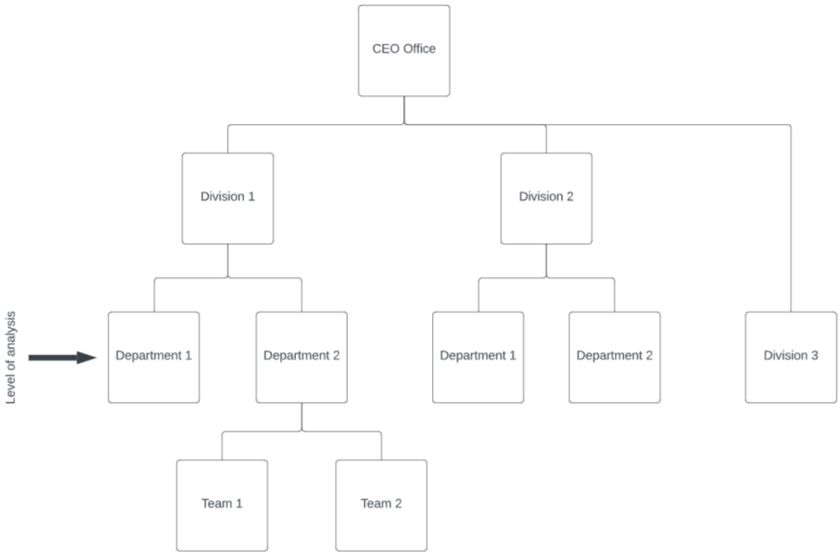


Figure 4. Inconsistencies in terms of the hierarchical decomposition of an organization.

This means that the “full sized” DSM will have units that differ in terms of the level of decomposition. If we follow a simple procedure where we aggregate each cluster "one level up", it means that the final, aggregated matrix will have elements representing units at different levels of the system, something which complicates analysis and interpretation. This problem is illustrated in Figure 5, which expands the example provided in Figure 2. We see that there are two alternative ways of representing the sub-teams within Sales & Marketing.

There are also challenges with regards to the technical procedure. In principle, aggregation is straightforward: Sets of elements are collapsed into single elements representing the set, as shown in Figure 3. However, in practice, it is not clear how this procedure should be carried out. First, in Figures 2 and 3, we simply represented any interaction within or across a unit as an “X” and did not differentiate between sparsely and heavily populated clusters when aggregating. However, as there will typically be an unequal number of marks within and across different sub-clusters, do we need to use weighted marks (e.g., a count of these marks) in the aggregated matrix? This problem is illustrated in Figure 6.

Finally, when a DSM with more than one level is analyzed, there is a need to consider whether sub-unit interactions would count as much as unit level interactions. That is, the question is whether we should penalize each type differently when clustering elements. Interactions that are “closer” (e.g., between employees across two teams, but inside the same department) should probably count less strongly than interactions that are “farther away” (e.g., between employees in two different departments) because the cost of coordinating across departments (or other major organizational sub-units) would normally be higher than that of coordinating inside departments. This is principle recognized in organizational theory (Thompson, 1967), which prescribes that the most highly interdependent roles are supposed to be placed in the same unit, while somewhat less strongly interdependent roles not necessarily be placed in the same unit, but close to each other organizationally. However, to our knowledge, this is assumption is not built into current clustering algorithms.

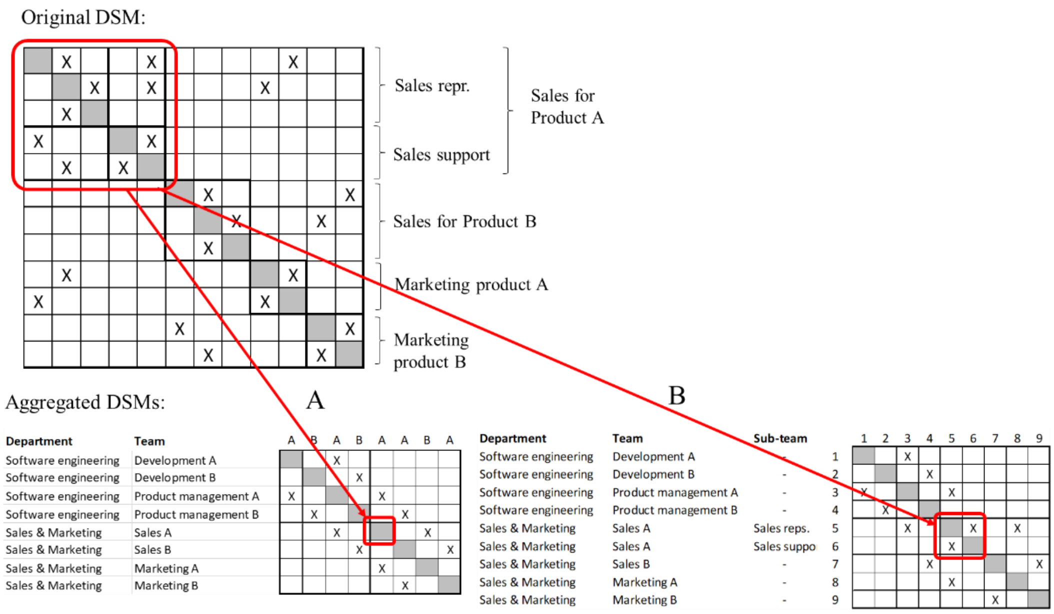


Figure 5. Two alternative ways of aggregating a unit (in this case, team) that is further decomposed into sub-units (in this case, sub-teams). Note that the upper matrix only shows the Sales & Marketing department, while the aggregated matrices at the bottom also includes software engineering.

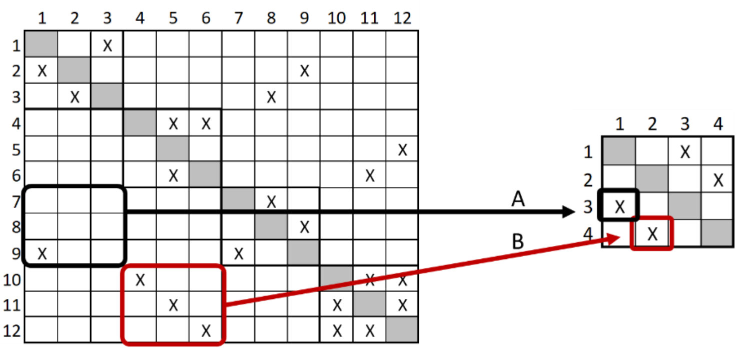


Figure 6. Example of a situation where an unequal number of marks (1 versus 3) are represented with the same value ("X") in the aggregated matrix.

3 Tentative solutions

How can we determine whether it is appropriate to use an aggregated DSM instead of one containing information representing several system levels? We propose a pragmatic test to find the answer to this question.

Consider again the organization depicted in Figure 2, which is currently functionally organized with two departments (Software engineering and Sales & Marketing). Let us assume that the leader of this organization wants to determine whether there is an alternative way of organizing the teams that would lead to lower coordination costs. For simplicity, let us only consider one alternative, namely a "divisionalized" model where the teams are organized according to product. As can be seen from Figure 3, the teams within each department are already specialized on either product A or B.

First, we use the full DSM to calculate the coordination costs for both alternatives. Then, we calculate the coordination cost using the aggregated DSM. Finally, we compare the coordination costs between the full DSM and the aggregated DSM.

The total coordination cost is the sum of the department coordination costs and the team coordination costs. Both the department coordination cost and the team coordination cost are calculated using Equation (1), but at two different levels. That is, Equation (1) is applied twice to the full DSM. As for treating the interactions within departments differently than interactions between departments (the concern we raised above, which is not addressed in current algorithms), we suggest to add the multiplier (γ) to the second part of Equation (1) to denote and highlight this imbalance in contribution to the overall coordination cost.

$C = \sum_{j=1}^{n_i} k_{ij}\, m_{ij} + \gamma Q_i\, N_i$ (1)

Where:

n_i = the number of teams in department i

k_{ij} = the number of marks inside team j of department i

m_{ij} = the size of team j in department i

Q_i = marks outside all teams in department i

N_i = size of department i (i.e. DSM size for that department)

γ = the relative importance of interactions between teams within the department (can be initially set to 1.5)

The department coordination cost is also calculated using Equation (1), but we exchange the teams by departments and the departments by the organization. Thus, the parameters of Equation (1) are redefined as follows:

n_i = the number of departments in the organization (i)

k_{ij} = the number of marks inside department j of the organization (i)

m_{ij} = the size of department j in organization (i) under investigation.

Q_i = marks between the departments of organization (i)

N_i = the size of organization (i) (i.e. DSM size for the whole organization)

γ = the relative importance of interactions between departments within the organization (can be initially set to 1.5)

Unlike clustering physical DSMs, where the interdependency types can vary significantly (such as "heat", "signal", "material" etc. transfers), the organizational DSMs that we use here only contain interdependencies that reflect work-process related communication and interaction between people. The magnitude of these interactions can vary between rare communication to daily and rich communication patterns. Equation (1) can be easily extended to account for such differences by using a numerical DSM instead of a binary DSM (The numerical DSM is used to replace the X marks by the sum of the marks at the lower level when aggregating.)

Equation (1) is first applied to the DSM in Figure 7; the number of interactions used in the calculations are shown in the lower part of the figure. Using Equation (1), the coordination cost for the DSM in Figure 7 is:

$C_{T1} = [(3*3) + (3*3) + (3*3) + (6*3)] + (1.5)(8)(12) = 189$

$C_{T2} = [(9*5) + (3*3) + (2*2) + (2*2)] + (1.5)(8)(12) = 206$

$CD = [(23*12) + (24*12)] + (1.5)(8)(24) = 852$

Total coordination cost equals $C_{T1} + C_{T2} + CD = 189 + 206 + 852 = 1247$

The coordination cost for the current model is then compared to the alternative model; namely, the product based or divisionalized structure (Figure 8).

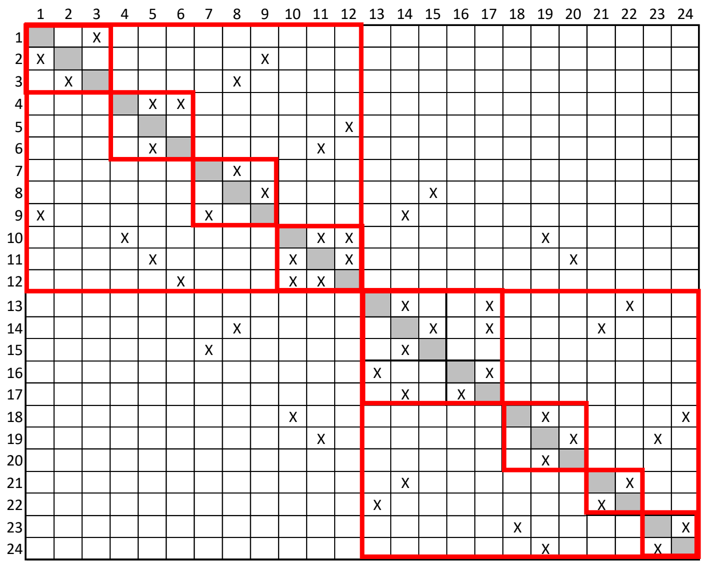
Using Equation (1), the coordination cost for the DSM in Figure 8 is:

$C_{T1} = [(3*3) + (3*3) + (9*5) + (2*2)] + (1.5)(11)(13) = 281.5$

$C_{T2} = [(3*3) + (6*3) + (3*3) + (2*2)] + (1.5)(13)(11) = 254.5$

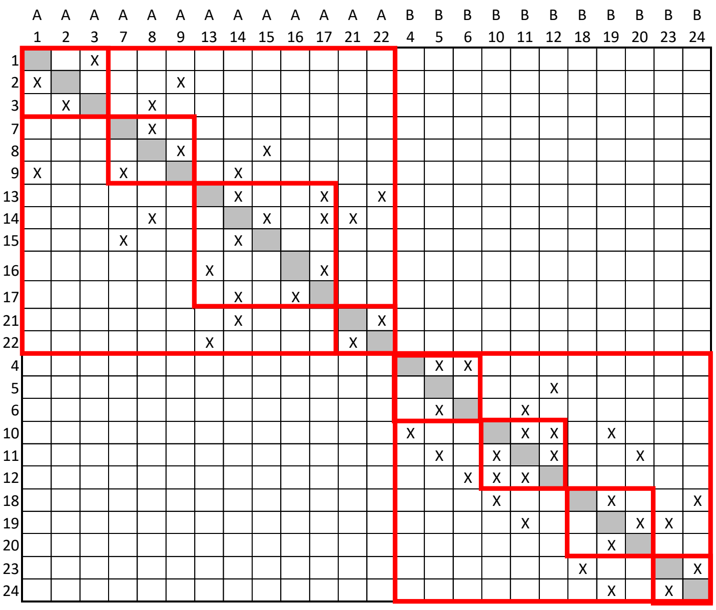
$CD = [(28*13) + (27*11)] + (1.5)(0)(24) = 661$

Total coordination cost equals $C_{T1} + C_{T2} + CD = 281.5 + 245.5 + 674 = 1197$



Number of marks within teams	3			3			3			6			9			3			2				2	
Sum												**15**												**16**
Number of marks outside teams, within department	1			3			0						2			2			0					
Number of marks outside teams, within department	0			2			2						0			2			2					
Sum												**8**												**8**
Number of marks between departments												**4**												**4**

Figure 7. Full DSM for the current organization (functional structure) with count of marks.



Number of marks within teams	3			3			9			2			3			6			3				2	
Sum												**17**												**14**
Number of marks outside teams, within department	1			2			2						3			2			2					
Number of marks outside teams, within department	2			2			2						2			2			2					
Sum												**11**												**13**
Number of marks between departments												**0**												**0**

Figure 8. Full DSM for the alternative organized by product with count of marks.

The conclusion from using the full DSM is that the coordination cost is lower for the product based alternative. This result can then be compared to the results of using the aggregated DSMs representing the same two alternatives (Figure 3 and 9).

Using Equation (1) to calculate the coordination cost for the DSM in Figure 3, we obtain:

Total coordination cost $= [(4 * 4) + (4 * 4)] + (1.5)(4)(8) = 80$

Similarly, we use Equation (1) to calculate the coordination cost for the DSM in Figure 9 and we obtain:

Total coordination cost $= [(6 * 4) + (6 * 4)] + (1.5)(0)(8) = 48$

Department	Team	Sub-team	Product	A	A	A	A	B	B	B	B
Product A	Development A	-	A		X						
Product A	Product management A	-	A	X		X					
Product A	Sales A	-	A		X		X				
Product A	Marketing A	-	A			X					
Product B	Development B	-	B						X		
Product B	Product management B	-	B					X		X	
Product B	Sales B	-	B						X		X
Product B	Marketing B	-	B							X	

Figure 9. Aggregated DSM for the alternative organized by product.

We see that the result is the same, namely that the coordination cost is lower for the product based alternative. Hence in this case one would reach the same conclusion using the aggregated DSMs as the full DSMs.

To be sure, we only used one potential scenario in this example, but one could use the same methodology to systematically test various scenarios that differ, for example, with regards to the density of the matrices and the extent to which the marks represent functionally-oriented or product-oriented interdependencies.

This analysis supports the “one level below” principle; that is, in this case it may be sufficient to use team-level data and unnecessary to collect and analyze data from levels further down in the organization (in this case, at the individual level). The assumption, though, is that one can treat the teams as intact units and simply re-shuffle them between departments. Individual-level data are required if there is a risk that one may need to split any of the existing teams in order to identify a satisfactory solution.

Finally, to resolve any inconsistencies resulting from an uneven organizational hierarchy, we suggest two principles.

The first principle is “stable hierarchy leaf nodes”. This principle suggests that units at a higher level of the hierarchy that do not have any sub-units are considered to be at the level of analysis that is being used. So in the case of Division 3 in Figure 4, we suggest that it should be considered a department when doing an analysis at the department level. In this manner, the elements belonging to Division 3 will be included in the analysis. This will also ensure that the clustering analysis will identify a need for sub-division into teams or departments in Division 3, and help the organization avoid an excessive span of control. Any leaf node should therefore appear as a reasonable organization unit at any level of the hierarchy.

The second principle is “aggregation to the level of analysis”. This principle suggests that units at a level below the level of analysis should be aggregated to the level of analysis using the methods suggested in this paper. So with regards to the example Figure 4, Department 2 would be represented as a combination Team 1 and Team 2.

We suggest that using the principles of stable hierarchy leaf nodes and aggregation to the level of analysis an analysis can be done at any level of an organization regardless of any inconsistencies in the levels of the organization hierarchy.

4 Summary and conclusion

In this paper, we described the key challenges related to DSM aggregation. The question was whether and how lower-level units (e.g., teams) may be aggregated to higher level units (e.g., departments) in order to simplify clustering and analysis. We proposed some tentative solutions as summarized in Table 1. In general, the choice of aggregation method will depend on the purpose of the analysis. The purpose will determine what information should remain intact after aggregation to perform the analysis in a way that would be consistent with the results of the same analysis done at a more detailed level.

Table 1. Summary of the proposed approaches for handling the challenges.

Challenge	**Description**
1. Assess information loss due to aggregation	Equation (1) provides a procedure for testing / evaluating the impact on decision making of using full versus aggregated matrices.
2. Uneven organizational hierarchy and handling units with different levels of decomposition	Apply the following two principles: 1. Stable hierarchy leaf nodes 2. Aggregation to the level of analysis
3. Differentiating between sparsely and heavily populated clusters when aggregating	When aggregating, replace the X marks by the sum of the marks at the lower level.
4. Different coordination cost of interdependencies within vs. between clusters	The multiplier (γ) in the second part of Equation (1) highlights this imbalance in contribution to the overall coordination cost

References

Chiriac, N., Hölttä-Otto, K., Lysy, D., & Suk Suh, E. (2011). Level of modularity and different levels of system granularity. Journal of Mechanical Design, Transactions of the ASME, 133(10).

Gigerenzer, G. (2022). How to stay smart in a smart world. MIT Press.

Katsikopoulos, K. V., Şimşek, Ö., Buckmann, M., & Gigerenzer, G. (2022). Transparent modeling of influenza incidence: Big data or a single data point from psychological theory? International Journal of Forecasting, 38(2), 613–619.

Sharman, D. M., & Yassine, A. A. (2004). Characterizing complex product architectures. Systems Engineering, 7(1), 35–60.

Sheppard, S., Macatangay, K., Colby, A., Sullivan, W. M., & Shulman, L. S. (2009). Educating engineers: Designing for the future of the field. Jossey-Bass.

Thompson, J. D. (1967). Organizations in action: Social science bases of administrative theory. McGraw-Hill.

Worren, N., Christiansen, T., & Soldal, K. (2020). Using an algorithmic approach for grouping roles and sub-units. Journal of Organization Design, 9(8), 1–19.

Yu, T.-L., Yassine, A. A., & Goldberg, D. E. (2007). An information theoretic method for developing modular architectures using genetic algorithms. Research in Engineering Design, 18(2), 91–109.

Contact: Ali Yassine, Stevens Institute of Technology, School of Systems & Enterprises, Babbio Center, 525 River St., Hoboken, NJ 07030, USA