DSM
(Dependency/Design Structure Matrix)

L T JayPrakash

Courtsey: DSMweb.org

Organization of the Talk

1. Introduction to DSM
   - Different DSM Types
   - How to Read a DSM
   - Building and Creating a DSM
   - Hands-on Exercises
   - Operations on DSM

Introduction

Introduction

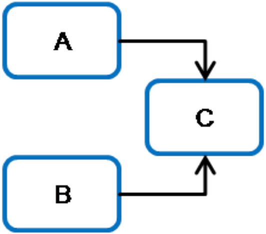
- Traced back 1960, Widespread attention 1990 onwards.
- Synonyms: Dependency Structure Matrix, Dependency Source Matrix, or Dependency Structure Method.
- Use: To represent and **analyze system models** in a variety of application areas.
- A DSM is a square matrix (i.e., # rows = # columns) that shows relationships between elements in a system.
- **Why Study?**
  - Since the behavior and value of many systems is largely determined by interactions between its constituent elements, DSMs have become increasingly useful in recent years.

Introduction ... cont'd

- Relative to other system modeling methods, a DSM has two main advantages:
  1. It provides a simple and concise way to represent a complex system.
  2. It is amenable to powerful analyses, such as clustering (to facilitate modularity) and sequencing (to minimize cost and schedule risk in system/processes).
- The DSM is related to other square-matrix-based methods such as,
  - dependency map, a precedence matrix, a contribution matrix, an adjacency matrix, a reachability matrix, and an N-square diagram.
- And, also related to non-matrix-based methods such as,
  - directed graphs, systems of equations, and **architecture diagrams** and other dependency models.

Introduction ... cont'd

- The use of graphs in managing complex structures is known.
- For example, consider a system that is composed of three elements (or sub-systems): element "A", "B", and "C".
- Resultant graph is called a directed graph or simply a digraph.



	A	B	C
A			X
B			X
C			

Figure: The journey of Digraph to DSM

Properties of Digraph

- The matrix representation of a digraph (i.e. directed graph) has the following properties:
  1. it is binary (i.e. a matrix populated with only zeros and ones)
  2. it is square (i.e. a matrix with equal number of rows and columns)
  3. it has n rows and columns (n is the number of nodes of the digraph)
  4. it has k non-zero elements, where (k is the number of edges in the digraph)

Relationships between Tasks

Three basic configurations to characterize a system

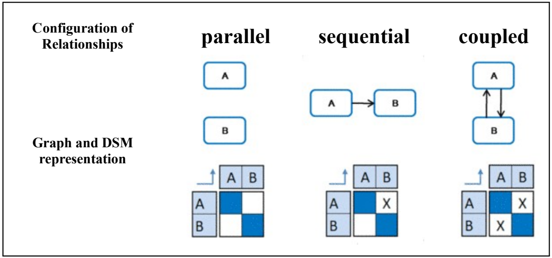


Figure: Relationships

Note: Synonomous usage: (i) Parallel, or Independent, (ii) Sequential, or Series, or Dependent, (iii) Coupled, or Inter-Dependent.

Different DSM Types

Remember about DSM

- There is no pre-defined DSM that is helpful for any problem that is to be structured.
- DSM needs to be adapted to the kinds of elements and relations that prevail in the system in focus.
- Type of the elements and dependencies needs to be defined as precisely as possible to obtain the information structure for the DSM.

Terms analogous in Graph Theory

Term in Systems Theory: entity, – relation
Term in Graph Theory: node, vertex – edge, arc
Terms common for the use of DSM:

- element: especially in matrix-based methodologies, the term element is used to refer to an entity that is entered into a row or a column
- dependency / interdependency: whereas relation refers to any kind of connection or association between two entities, may they be directed or not, a dependency often implies a direction as element A will depend on element B

Regrouped as: domain – relationship type

Idea of Domains in DSM

- From a structural point of view: System - network-like model of entities and their relations.
    - These entities can be of different kinds, e.g. documents, oral information, etc.
    - If many such kinds are mixed, the network is incoherent.
- Each kind of entities represents a SPECIFIC VIEW to the system, called domain.
- The purpose of a domain, which is comparable to a "class" of objects in OO paradigms, is to create homogeneous networks that allow to compare elements during structural analysis.
- The term "domain" can therefore be defined as one specific view (among several) on one complex system at a time.
- It comprises one type of entity that can be analyzed by the same algorithm, providing a meaningful result.
- A DSM always only contains elements from one domain.

Specifying the DSM you need

- To better specify the DSM type needed, first, the relevant elements should therefore be classified.
- Step 1: Similar elements should be regrouped and denominated respectively.
  - If, e.g. modularization is the goal, requirements, functions, and components are relevant.
  - To modularize the components better and to better understand their interfaces, the domain "components" can be selected, and a DSM can be set up that has the components of the system as elements in rows and columns.
- Step 2: Relationship type is defined.
  - It is important that a DSM contains only one relationship type, as only then the analyses will make sense.

Common classification of DSM

- Four different common types of data that can be represented in a DSM have been identified, however, any other type of DSM is possible, too.
- Details using: DSM data types - Representation - Applications
  - Component-based (Product) - Component relationships - System architecting, engineering and design
  - People-based (Organization), or Team-based - Organizational unit relationships - Organizational design, interface management, team integration
  - Activity-based (Process) - Activity input/output relationships - Process improvement, project scheduling, iteration management, information flow management
  - Parameter-based (low-level Process) - Design parameter relationships - Low level activity sequencing and process construction, sequencing design decisions

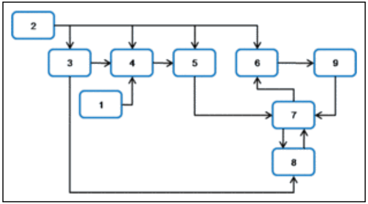
Few Example DSMs

		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Radiator	A		X														
Engine Fan	B	X				X											
Heater Core	C																X
Heater Hoses	D																
Condenser	E						X		X								
Compressor	F					X			X	X							
Evaporator Case	G																X
Evaporator Core	H					X	X			X							X
Accumulator	I						X		X								
Refrigeration Controls	J																
Air Controls	K																
Sensors	L																
Command Distribution	M																
Actuators	N																
Blower Controls	O																X
Blower Monitor	P			X				X	X							X	

		D	J	K	L	M	N	A	B	E	F	I	H	C	P	O	G
Heater Hoses	D																
Refrigeration Controls	J																
Air Controls	K																
Sensors	L																
Command Distribution	M																
Actuators	N																
Radiator	A								X								
Engine Fan	B							X		X							
Condenser	E								X		X		X				
Compressor	F									X		X	X				
Accumulator	I										X		X				
Evaporator Core	H									X	X	X			X		
Heater Core	C														X		
Blower Monitor	P												X	X		X	X
Blower Controls	O														X		
Evaporator Case	G														X		

Figure: Component-based DSM - Automobile Climate Control System

Few Example DSMs ... cont'd



	1	2	3	4	5	6	7	8	9
1				X					
2			X	X	X	X			
3				X				X	
4					X				
5							X		
6									X
7						X		X	
8							X		
9							X		

Figure: Activity-based DSM - Set of tasks in a process

Few Example DSMs ... cont'd

		1	2	3	4	5	6	7	8	9	10	11	12	13
Customer Requirements	1			X	X	X								
Wheel Torque	2					X	X		X		X			X
Pedal Mech. Advantage	3					X		X		X		X		X
System Level Parameters	4		X	X		X		X	X	X	X	X		X
Rotor Diameter	5			X				X				X		X
ABS Modular Display	6													
Front Lining Coef. of Friction	7					X								X
Piston-Rear Size	8			X		X		X			X	X		X
Caliper Compliance	9						X							
Piston- Front Size	10			X		X		X	X	X		X		X
Rear Lining Coef of Friction	11					X								X
Booster - Max. Stroke	12													
Booster Reaction Ratio	13			X		X		X		X		X	X	

		1	4	2	10	8	3	11	7	13	5	12	9	6
Customer Requirements	1		X				X				X			
System Level Parameters	4			X	X	X	X	X	X	X	X		X	
Wheel Torque	2				X	X				X	X			X
Piston- Front Size	10					X	X	X	X	X	X		X	
Piston-Rear Size	8				X		X	X	X	X	X			
Pedal Mech. Advantage	3							X	X	X	X		X	
Rear Lining Coef. of Friction	11									X	X			
Front Lining Coef. of Friction	7									X	X			
Booster Reaction Ratio	13						X	X	X		X	X	X	
Rotor Diameter	5						X	X	X	X				
Booster - Max. Stroke	12													
Caliper Compliance	9													X
ABS Modular Display	6													

Figure: Parameter-based DSM - Automobile brake system design

Understanding People-based (Team-based) DSM

- Used for organizational analysis and design based on information flow among various organizational entities.
  - Individuals and groups participating in a project are the elements being analyzed (rows and columns in the matrix).
  - A Team-based DSM is constructed by identifying the required communication flows and representing them as connections between organizational entities in the matrix.
- For the modeling exercise it is important to specify what is meant by information flow among teams. Possible ways information flow can be characterized as Flow Type - Possible Metrics.
  - Level of Detail - Sparse (documents, e-mail) to rich (models, face-to-face)
  - Frequency - Low (batch, on-time) to high (on-line, real)
  - Direction - One-way to two-way
  - Timing - Early (preliminary, incomplete, partial) to late (final)

Understanding People-based (Team-based) DSM ... cont'd

- Again, the matrix can be manipulated in order to obtain clusters of highly interacting teams and individuals while attempting to minimize inter-cluster interactions.
- The obtained groupings represent a useful framework for organizational design by focusing on the predicted communication needs of different players.

Reading a DSM

My First DSM

- Primary benefit of DSM: Graphical nature of the matrix display format.
- The matrix provides a highly compact, easily scalable, and intuitively readable representation of a system architecture.
- Rules to read the DSM
  - Cells along the matrix diagonal represent all system elements.
  - To keep the matrix diagram compact, full names of the elements are often listed to left of the rows (and sometimes also above the columns) rather than in the diagonal cells.
  - It is also easy to think of each diagonal cell as potentially having inputs entering from its top and bottom and outputs leaving from its left and right sides.
  - The sources and destinations of these input and output interactions are identified by marks in the off-diagonal cells.
  - Examining any row in the matrix reveals all of the outputs from the element in that row (which are inputs to other elements).

My First DSM ... cont'd

- Looking down any column of the matrix shows all of the inputs to the element in that column (which are outputs from other elements).
- My First DSM : See figure below.

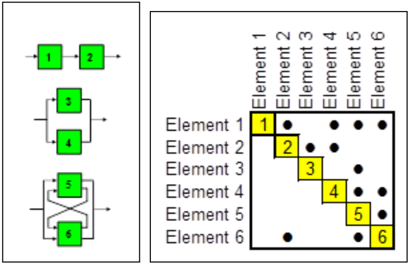


Figure: A simple DSM Model with six elements

Understanding our first DSM

For example, in DSM model (on previous slide),

- Reading across row 2, we see that element 2 provides outputs to elements 3 and 4.
- Reading down column 5, we see that element 5 receives inputs from elements 1, 3, and 4.
- Thus, a mark in an off-diagonal cell (e.g., cell 3,5) represents an interaction that is both an input and an output, depending on whether one takes the perspective of its provider (element 3) or its receiver (element 5).
- To the left of the DSM above are node-directed link diagram equivalents of portions of the DSM.
- Note that elements 1 and 2 form a linear chain or sequence, while elements 3 and 4 are independent, and elements 5 and 6 are interdependent or coupled.

Binary and Numerical DSM

- Example DSM also called a binary DSM because the off-diagonal marks indicate merely the presence or absence of an interaction.
- The binary DSM representation can be extended in many ways by including further attributes of the interactions, such as,
  - the number of interactions and/or
  - the importance, impact, or strength of each - which might be represented by using one or more numerical values, symbols, shadings, or colors instead of just the binary marks in each of the off-diagonal cells.

This extended form of DSM is called a numerical DSM.

What is additional in Numerical DSM?

- Additional attributes of the elements themselves may be included by adding more columns to the left to describe, for example, the type, owner, or status of each element.
- Additional attributes of the interactions, such as their names, requirements, etc. are usually kept in separate repositories but may be linked to the DSM cells by numerical identification numbers or indices.
- Many DSM resources use the opposite convention, the transpose of the matrix.
  - An elements inputs shown in its row and its outputs shown in its column.

DSM Conventions

Eppinger and Browning (2012) developed the following notation for these two conventions:

1. IR/FAD convention: DSM with inputs shown in rows, outputs in columns; hence, any feedback marks will appear above the diagonal.
2. IC/FBD convention: DSM with inputs shown in columns, outputs in rows; hence, any feedback marks will appear below the diagonal.

Both conventions convey same information; each is just the matrix transpose of the other.

Both are widely used - because of personal preferences.

Basic Steps in building and creating DSM

1. Interview stakeholders (for e.g., engineers and managers)
2. Check for possible sources of data that can be parsed or exported into a DSM
3. Determine list of system elements
4. Ask about inputs, outputs, strengths of interaction, etc between elements
5. Enter marks in matrix
6. Collect the comments that explain each element and each dependency (for later understanding and interpretation)
7. Check with staekholders to verify/comment on DSM
8. Refine the model over time by assimilating system / project / process / organizational learning

Exercise 1.

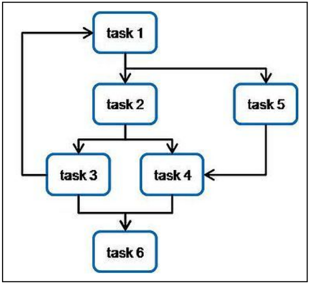


Figure: An Example Task Dependency Graph

Construct a DSM.

What basics should be verified after construction of DSM?

Incorrect Solution to Exercise 1.

	task 1	task 2	task 3	task 4	task 5	task 6
task 1		X			X	
task 2			X	X		
task 3	X			X		
task 4						X
task 5				X		
task 6						

Figure: The DSM[1]

[1] Move mark at (task3, task4) to (task3, task6)

Exercise 2.

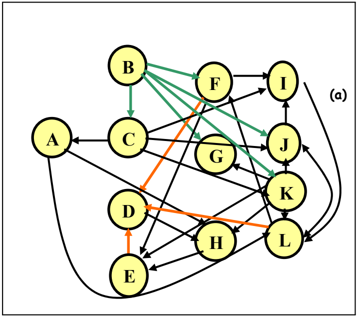


Figure: An Example Dependency Graph

Construct the base DSM for the dependency graph (a).

Solution to Exercise 2.

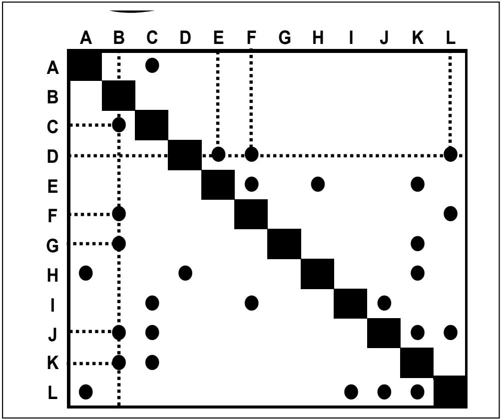


Figure: The Base DSM

Sample Exercise 3 using tools.

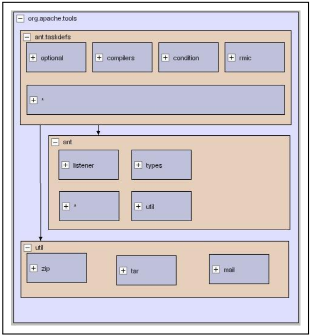
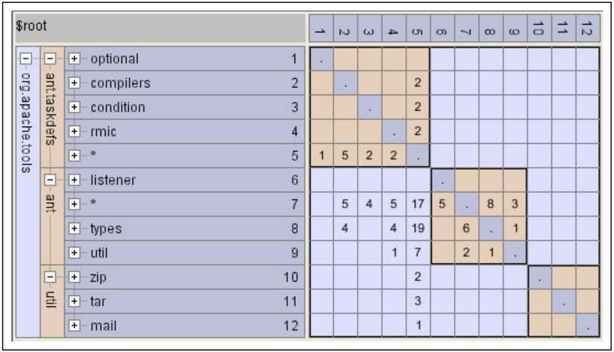


Figure: Conceptual Architecture for ANT Version 1.4.1

Construct the base DSM for the above architecture

Solution to Exercise 3 using Automated DSM Tools.



$root			1	2	3	4	5	6	7	8	9	10	11	12
org.apache.tools / ant.taskdefs	optional	1	.											
	compilers	2		.			2							
	condition	3			.		2							
	rmic	4				.	2							
	*	5	1	5	2	2	.							
org.apache.tools / ant	listener	6						.						
	*	7		5	4	5	17	5	.	8	3			
	types	8		4		4	19		6	.	1			
	util	9				1	7		2	1	.			
org.apache.tools / util	zip	10					2					.		
	tar	11					3						.	
	mail	12					1							.

Figure: Dependency Model for ANT Version 1.4.1 with Rules for Layering - The Base DSM

Key Lessons to Rememeber

- Concept: IR/FAD, IC/FBD.
- The significance of Boundary - The Diagonal.
- Division of DSM into two - Upper half and Lower half.
- What fits upper half? Sequential, Parallel.
  - More entries - No Issues. Rather preferable.
- What fits lower half? Coupled.
  - More entries - Reason for the RISK. Not preferred.

Applications of DSM

- Through analysis, one can prescribe a more modular system architecture or organization structure.
- Because the DSM highlights process feedbacks, it helps identify iteration and rework loops - key drivers of cost and schedule risk.
- The DSM can also show how delays in external inputs, such as requirements and equipment, trace directly to increased cost, schedule, and risk.

How to leverage the power of DSM?

- Sequencing a DSM
- Tearing a DSM
- Banding a DSM
- Clustering a DSM

Sequencing a DSM

1. Sequencing is reordering of rows and columns of the DSM such that,
   - the new DSM arrangement does not contain any feedback marks,
   - the transformed DSM is in upper triangular form (UTF).
2. For complex systems: Solution-[1] may not give DSM in UTF.
   - Therefore, the analyst's objective changes from eliminating feedback marks to moving them as close as possible to the matrix diagonal a.k.a. Block Triangular Matrix.
   - Equally, it is possible to learn about what elements of the system might possibly have to be reworked (e.g. split into two elements or perhaps removed) to achieve a better system / process architecture.

Approaches for DSM Sequencing

All approaches are similar - only difference in how they identify cycles/ loops/ circuits of coupled elements.
Major steps in sequencing algorithms:

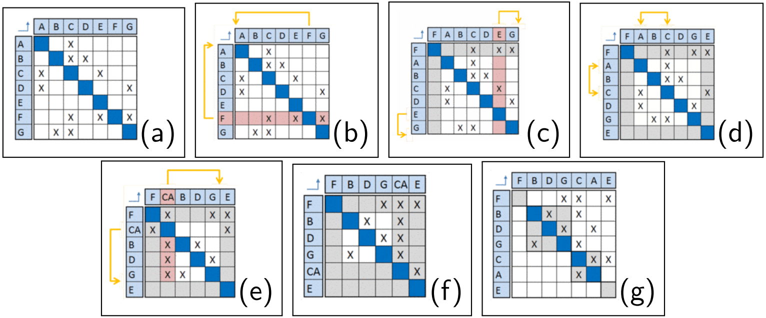
1. [No Input criteria] Identify system elements (or tasks) that can be determined (or executed) without input from rest of the elements in DSM.
   - To identify such elements - Observe an empty column in DSM.
   - Place those elements to the left of the DSM.
   - Once an element is rearranged, it is removed from the DSM (with all its corresponding marks).
   - Repeat Step 1 on the remaining DSM elements.
2. [No Output criteria] Identify system elements (or tasks) that deliver no information to other elements in DSM.
   - To identify such elements - Observe an empty row in DSM.
   - Place those elements to the right of the DSM.
   - Once an element is rearranged, it is removed from the DSM (with all its corresponding marks).
   - Repeat Step 2 on the remaining DSM elements.
3. If after Steps 1 and 2 there are no remaining elements in the DSM, then the matrix is completely partitioned; otherwise, the remaining elements contain information circuits (at least one).
4. Determine the circuits by one of the following methods:
   Path Searching, Powers of the Adjacency Matrix Method, The Reachability Matrix Method, Triangularization Algorithm, Tarjan's Depth First Search Algorithm.
5. Collapse the elements involved in a single circuit into one representative element and go to Step 1.

Identify loops by Path Searching

- In path searching, information flow is traced either backwards or forwards until a task is encountered twice.
- All tasks between the first and second occurrence of the task constitute a loop of information flow.
- When all loops have been identified, and all tasks have been scheduled, the sequencing is complete and the matrix is in block triangular form.

The figure, on next slide, is a simple example. The path searching partition proceeds as follows:

Identify loops by Path Searching ... cont'd



Is the above Transformation Correct?

Try calculating ratio of "X" movement, a value close to 1 indicates the proximity (% improvement) of DSM to optimum.

Powers of the Adjacency Matrix Method

- The Adjacency matrix is a binary DSM where an empty cell is replaced with a "zero" and a non-empty cell is replaced by "one".
- Raising the DSM to the n-th power shows which element can be reached from itself in *n* steps by observing a non-zero entry for that task along the diagonal of the matrix.
- For example, squaring the DSM (see next slide) shows that tasks A and C are involved in a two-step loop.
  - Note that in the resultant square matrix, cells with a value of greater than one were replaced by a value of one.
- Similarly, cubing the DSM, as shown in next slide, shows that tasks B, D and E are involved in a three-step loop. The higher powers of the DSM reveal no other loops in the system.

Powers of the Adjacency Matrix Method ... cont'd

	A	B	C	D	E
A		0	1	0	0
B	0		1	1	0
C	1	0		0	0
D	1	0	0		1
E	0	1	1	0	

(a)

	A	B	C	D	E
A	1	0	0	0	0
B	1	0	0	0	1
C	0	0	1	0	0
D	0	1	1	0	0
E	1	0	1	1	0

(b)

	A	B	C	D	E
A	0	0	1	0	0
B	0	1	1	0	0
C	1	0	0	0	0
D	1	0	1	1	0
E	1	0	1	0	1

(c)

	A	B	C	D	E
A	1	0	0	0	0
B	1	0	1	1	0
C	0	0	1	0	0
D	1	0	1	0	1
E	1	1	1	0	0

(d)

	A	B	C	D	E
A	0	0	1	0	0
B	0	0	1	1	0
C	1	0	0	0	0
D	1	0	0	0	1
E	0	1	1	0	0

(e)

Exercise 4.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	■						●							●	●
2		■			●	●					●				●
3			■		●	●					●				●
4				■	●								●		●
5					■			●	●				●	●	●
6					●	■		●	●		●	●	●	●	●
7					●	●	■	●					●		
8					●			■		●		●		●	
9									■						
10								●		■			●	●	●
11					●	●		●			■			●	
12								●				■			
13										●			■		
14														■	
15														●	■

Apply path searching algorithm on above DSM.

The Reachability Matrix Method (RMM)

- The reachability matrix is a binary DSM with diagonal elements equal to "1".
- The RMM calls for finding a multi-level hierarchical decomposition for the matrix.
- The top level in this hierarchy is composed of all elements that require no input, or are independent from all other elements in the matrix.
- Any two elements at same level of hierarchy are either not connected to each other, or are part of same circuit at that level.
- Once the top level set of elements is identified, the elements in the top level set and their corresponding from/to connections are removed from the matrix leaving us with a sub-matrix that has its own top level set.
- The top level set of this sub-matrix will be the second level set of the original matrix.

Major Steps in The Reachability Matrix Method ... cont'd

1. Construct a table with four columns.
   - In the first column, list all the elements in the matrix.
   - In the second column, list the set of all the input elements for each row in your table. This set can easily be identified by observing an entry of "ONE" in the corresponding column in the DSM. (Include the element itself as an input).
   - In the third column, list the set of all output elements for each row in your table. This set can easily be identified by observing an entry of "ONE" in the corresponding row in the DSM. (Include the element itself as an output).
   - In the fourth column, list the intersection of the input and output sets for each element in your table.
2. Identify top level elements and remove them from the table. An element is in the top level hierarchy of the matrix if its input set is equal to the intersection set.
3. Go to step 1.

Tearing a DSM

- Once a subset of coupled elements has been identified in a DSM (see Sequencing the DSM), tearing is one way to attempt to determine a sequence for elements in this subset.
- Tearing is the process of choosing the set of feedback marks that, if removed from the matrix (and then the matrix is re-partitioned), will render the matrix upper-triangular.
- The marks that we remove from the matrix are called "tears".
- Identifying those "tears" that result in an upper triangular matrix means that we have identified the set of assumptions that need to be made in order to start design process iterations when coupled tasks are encountered in the process.
- Having made these assumptions, no additional estimates need to be made.

Tearing a DSM ... cont'd

No optimal method exists for tearing, but the use of following two criteria can be used when making tearing decisions:

1. Minimal number of tears: the motivation behind this criterion is that tears represent an approximation or an initial guess to be used; we would rather reduce the number of these guesses used.
2. Confine tears to the smallest blocks along the diagonal: the motivation behind this criterion is that if there are to be iterations within iterations (i.e. blocks within blocks), these inner iterations are done more often. Therefore, it is desirable to confine the inner iterations to a small number of tasks.

Banding a DSM

- Banding is the addition of alternating light and dark bands to a DSM to show independent (i.e. parallel or concurrent) activities (or system elements).
- Banding is similar to partitioning the DSM using the Reachability Matrix Method (RMM) when the feedback marks are ignored (a band corresponds to a level).
- The collection of bands or levels within a DSM constitute the critical path of the system or project.
- Furthermore, one element or activity within each band is the critical or the bottleneck activity.
- Thus, fewer bands are preferred since they improve the concurrency of the system or project.

Clustering a DSM

- When the DSM elements represent design components (i.e. component-based DSM) or teams within a development project (i.e. people-based DSM), the goal of the matrix manipulation changes significantly from that of sequencing algorithms.
- The new goal becomes finding subsets of DSM elements (i.e. clusters or modules) that are mutually exclusive or minimally interacting subsets, i.e. clusters as groups of elements that are interconnected among themselves to an important extent while being little connected to the rest of the system. This process is referred to as "Clustering".
- In other words, clusters absorb most, if not all, of the interactions (i.e. DSM marks) internally and the interactions or links between separate clusters are eliminated or at least minimized.

Clustering a DSM ... cont'd

- As a simple example, consider a development process that includes seven participants as shown in the DSM, on next slide.
- Note that the interactions between different participants are also shown in the DSM.
- If we were to form several development teams within this project, what will be the number of teams required and the membership of each team?
- Clustering the DSM for this project will provide us with insights into optimal team formations based on the degree of interactions among participants.

Clustering a DSM ... cont'd

If the DSM in (a) was rearranged as in (b). One possible team assignment is:

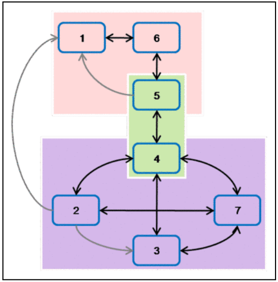
- Team 1 - participants 1, 5 and 6
- Team 2 - participants 4 and 5
- Team 3 - participants 2, 3, 4 and 7

	1	2	3	4	5	6	7
1						X	
2	X		X	X			X
3				X			X
4		X	X		X		X
5	X			X		X	
6	X				X		
7		X	X	X			

(a)

	1	6	5	4	2	3	7
1		X					
6	X		X				
5	X	X		X			
4			X		X	X	X
2	X			X		X	X
3				X			X
7				X	X	X	

(b)

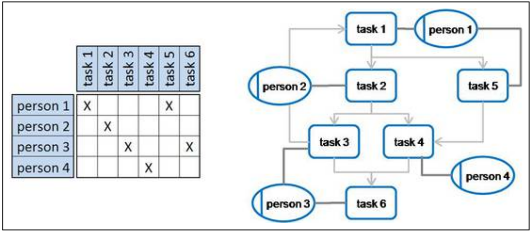


(c)

Note that by making participant 4 a member of both teams 2 and 3, we were able to absorb more interactions internally within a team without unnecessarily increasing the team size (by including participant 5 in team 3).

Domain Mapping Matrix (DMM)

- DSMs were later extended to Domain Mapping Matrices (DMM).
- Goal: To enable matrix methodology to include not just one domain at a time but to allow for the mapping between two domains.
- DMMs are thus rectangular matrices, and they can be binary or numerical. In the figure below, the process example shows who is responsible for which task. The DMM on the left hand side shows how these persons can be mapped to the tasks.



	task 1	task 2	task 3	task 4	task 5	task 6
person 1	X				X	
person 2		X				
person 3			X			X
person 4				X		

Figure: Domain Mapping Matrices

Multiple Domain Matrix (MDM)

- DSMs and DMMs can then be used to model whole systems consisting of multiple domains, each having multiple elements, connected by various relationship types. This is called a Multiple Domain Matrix (MDM).
- MDM allows analyzing a systems structure across multiple domains, condensing each single analysis into one DSM that represents multiple domains at a time.
- It is now well-known that matrix-based approaches integrating multiple views (domains) become more and more accepted to manage several perspectives onto a system, especially when it comes to large structures (e.g. > 1000 elements per DSM).
- MDM basically is a DSM with more detailed DSMs along its diagonal and DMMs outside the diagonal.
- MDM also depicts how multiple relationship types create several representations of a specific submatrix of the overall MDM.

Multiple Domain Matrix (MDM) ...cont'd

The organizational chart that is shown on the right hand side of the figure below shows how the people responsible for the different tasks are structured within the company. As the project team differs from the actual organizational setup, both representations are integrated into the MDM that now represents the whole system.

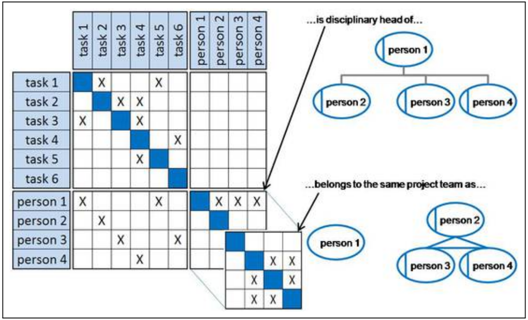


Figure: Multiple Domain Matrices

Brief Comparison between DSM and DMM

Table 1
Comparison between dependence matrix approaches

Dimensions	Design structure matrix (DSM)		Domain mapping matrix (DMM)
	Sequencing analysis	Clustering analysis	
Representation	Square matrix	Square matrix	Rectangular matrix
Analytical dimensions	Single domain	Single domain	Dual domain
Partitioning algorithm	Block diagonalization/triangularization	Clustering in blocks along the diagonal	Move items to clusters
Result, outcome of analysis	Flow of items, activities, i.e. sequencing	Clustering of items	Clustering of items
Visualization of dependencies	Parallelization of items Sequencing of items Feedback and circuits, loops of items	Clusters of items, Chunks Interfaces between clusters Hierarchical structures	Clusters of items Interfaces between clusters Hierarchical structures
Key words	Predecessors Successors Feedback and circuit loops Banding	Clustering Hierarchies Interfaces Linking pins	Clustering Hierarchies Interfaces Linking pins
Focus of analysis	Tasks Activities Information flow Deliverables flow	Parameters Components People Organizations Information flow	Components/organization Project/organizational structure Functionality/product architecture Information flow

Evaluation link

http://lattix.com/download-form

Practice Exercise

Design Structure Matrix

Test Example

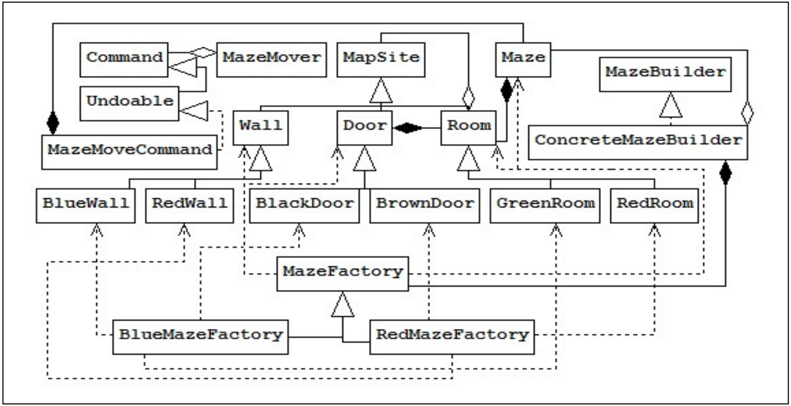


Figure: Class Diagram for the Intended Design of a System

Related Questions

Q.1 Construct Base DSM for the above class diagram.

Q.2 Carry out sequencing on the base DSM obtained in Q.1.

Q.3 Apply path searching algorithm to the base DSM.

Q.4 Consider a situation where a student submits a implementation of the system shown in the previous class diagram. However, it does not necessarily follow all the relationship restrictions. Based on this, answer the following.

(a) Given an intended design and a student submission implementing the design, is it possible to assess whether the student implemented the intended modular structure correctly?

(b) Is it possible to assess how well modularized a students submission is?

(c) For those students who have implemented the same assignment correctly but with different designs, is it possible to assess which design is better modularized?

Solution to Practice Exercise

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MapSite	1	.																			
Room	2	x	.																		
Wall	3	x		.																	
Door	4	x	x		.																
Maze	5		x			.															
MazeFactory	6		x	x	x	x	.														
Undoable	7							.	x												
Command	8								.												
MazeMover	9							x	x	.	x										
MazeMoveCommand	10	x	x		x	x		x			.										
ConcreteMazeBuilder	11		x		x	x	x					.	x								
MazeBuilder	12												.								
BrownDoor	13				x									.							
RedMazeFactory	14					x	x							x	.	x	x				
RedRoom	15		x													.					
RedWall	16			x													.				
BlackDoor	17				x													.			
BlueMazeFactory	18					x	x											x	.	x	x
GreenRoom	19		x																	.	
BlueWall	20			x																	.

Figure: DSM for the Class Diagram of a System