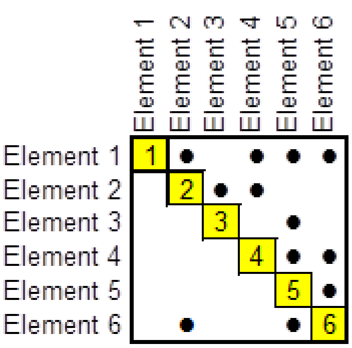
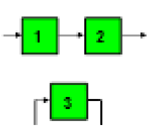
The Design Structure Matrix (DSM)

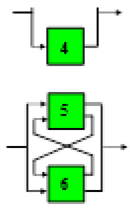
Reading a DSM

Note: Some material on this page has been adapted from Eppinger & Browning (2012), ***Design Structure Matrix Methods and Applications*** *(https://mitpress.mit.edu/books/design-structure-matrix-methods-and-applications)*, Cambridge, MA: MIT Press.

One primary benefit of DSM is the graphical nature of the matrix display format. The matrix provides a highly compact, easily scalable, and intuitively readable representation of a system architecture. The figure below shows a simple DSM model of a system with six elements. The cells along the diagonal of the matrix represent the system elements. To keep the matrix diagram compact, the full names of the elements are often listed to the left of the rows (and sometimes also above in 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). Looking down any column of the matrix shows all of the inputs to the element in that column (which are outputs from other elements). For example, in the figure below, 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 right 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.

This simple DSM example is 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.

Additional attributes of the elements themselves may also be included by adding more columns to the left of the square matrix 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, with an element's inputs shown in its row and its outputs shown in its column. Eppinger and Browning (2012) developed the following notation for these two conventions:

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

The two conventions convey the same information; each is just the matrix transpose of the other. Both are widely used because of the diverse roots of matrix-based tools for modeling systems (the IR convention originated from Steward's DSM work; the IC convention comes from N2 charts, IDEF0 diagrams, and Gantt charts), because of personal preferences, and because the IC/FBD convention provides a benefit with respect to the orientation of external input and output regions around the square DSM. For example, in the following figure (IC/FBD convention), one just continues to read through the external columns to see sources of external inputs, and one continues to read across the external rows to see destinations of external outputs.