Analyzing the Systems Underlying an Enterprise

Tyson R. Browning, *tyson@alum.mit.edu*

Enterprises are complex systems. As such, they should lend themselves to improvement via the application of systems engineering principles and approaches. To do so, we must first identify the system or systems underlying an enterprise.

Four key systems that tend to receive attention are the product, process, organization, and information systems. *Product* systems are the valued outputs or results produced by the enterprise, such as aircraft, buildings, computers, consulting, etc. Today, the design of complex product systems is the focus of most systems engineering applications. *Process* systems are the networks of work packages (activities) and decisions that produce a product system. Process systems are also referred to as value streams, and when the frame of reference becomes multiple companies, the terms supply chain and value chain are often heard. *Organization* systems are the structures of companies, teams, departments, and people that execute process systems. *Information* systems provide the infrastructure in which much of the data regarding the product and process systems are managed. Surely there are other important systems or systems views within an enterprise, but most enterprises do not have a satisfactory understanding of these four, basic, critical ones.

Enterprises are faced with the problems of describing, connecting, organizing, visualizing, understanding, managing, and improving these four systems. Usually, an enterprise will have multiple instances of each of these systems, compounding the challenge. Furthermore, the elements of each of these systems are related to elements of the others. An intelligent enterprise will address, learn about, understand, and leverage these systems and their relationships for competitive advantage.

Of the four systems, organizations and processes each provide a view of the system underlying an enterprise, because the structure of an enterprise can be described based on its organizations or its processes. The organizational view is more traditional, relying upon organization charts, descriptions of roles and responsibilities, team structures, boards and committees, etc. However, these descriptions do not adequately describe the organization system architecture, since relationships between organizational elements are often ignored.

Recently, many enterprises are discovering the advantages of the process view, which tends to align better with the product system, and thus with customers. The process view, since it describes work packages that must be done and work products that must be produced, also ties to the product, organization, and information systems in a relatively straightforward way. Thus, the process view provides a point of integration for the four systems: activities generate results while requiring people and tools. In accounting circles, the process view is akin to the activity-based view (e.g., activity-based costing). Activities (process elements) and their results seem to be powerful kernels for modeling and integrating a baseline system view.

The concept of a system *architecture* has proved helpful in describing systems. A system architecture includes the elements of the system and their relationships. Architecture modeling (where a model can be thought of as a systematic and disciplined, albeit abstract, description) has helped with visualizing and understanding systems. A classic approach to modeling a system architecture includes:

(1) decomposing the system into subsystems,
(2) defining the attributes of each subsystem,
(3) noting and verifying the relationships between (the integration of) the subsystems that give rise to the system's behavior, and
(4) noting and verifying the relationships with external systems and their impact on the system.

The resulting model improves understanding of the system and provides the basis for insightful analyses. However, the size and complexity of enterprise systems makes modeling them especially challenging.

This article presents a technique, the *design structure matrix* (DSM[1]), for modeling, representing, and analyzing at least three of the aforementioned enterprise systems—products, processes, and organizations. This article will focus on applying the DSM to modeling and analyzing a process system architecture, referring the reader to the references for information on other applications.

Process System Architecture Modeling with the DSM

The DSM is advantageous for representing and analyzing models of system architectures. Donald Steward codified the DSM in 1981 (Steward, 1981), based on matrix algebra and precedence diagram work in the 1960s.

[1] a.k.a. dependency structure matrix

The DSM is similar to an N-square diagram (Lano, 1979)—a familiar systems engineering tool used to represent system elements and their interfaces—with the addition of a time basis.

As shown in Figure 1, a DSM is a square matrix with corresponding rows and columns. The diagonal cells represent the systems elements (in this case, process activities), which are listed from upper left to lower right in a roughly temporal order. Off-diagonal cells indicate the dependency of one element (activity) on another. In processes, dependencies are often needs for work products or information. Reading down a column shows work product sources; reading across a row shows work product sinks. For example, row 1 indicates that Activity 1 provides information to Activities 2, 4, 5, and 6. Column 2 shows that Activity 2 depends on information from Activities 1 and 6.

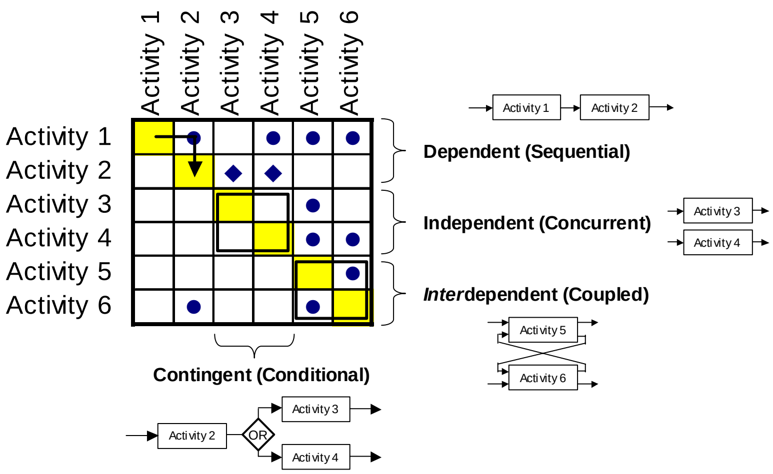

Figure 1: Example DSM

Figure 1 also shows how the DSM displays dependent, independent, and interdependent activity relationships. Since Activity 2 depends on information from Activity 1, these two activities will probably be executed sequentially in the workflow. Activities 3 and 4 do not depend on each other for information, so they may safely proceed in parallel (barring other resource constraints). Activities 5 and 6 both depend on each other's outputs. These activities are said to be interdependent or coupled and are discussed below. Note also that decisions are a kind of activity, one which produces information upon which other activities depend. Their sequence in a process will have a great bearing on its efficiency and effectiveness.

Of particular interest are the cases where marks appear in the lower triangular region of the DSM. Such marks indicate the dependence of an upstream activity on information created downstream. If project planners decide to execute the activities in the given order, Activity 2 will have to make an assumption about the information it needs from Activity 6. After Activity 6 finishes, Activity 2 may have rework if the assumption was incorrect. The DSM conveniently highlights iteration and rework, especially when it stems from activities working with potentially flawed information.

When we see a mark in the lower left corner of the DSM, we know that there is a chance of having to return to the beginning of the process, which could have a catastrophic impact on cost and schedule. The marks in the lower left corner of the DSM may represent key drivers of cost and schedule risk. Rearranging the activity sequence (by rearranging the rows and columns in the DSM) can bring some subdiagonal marks above or closer to the diagonal, thereby reducing their impact. Simple algorithms automate this exercise. Adding quantitative information to the DSM and using simulation can quantify the impacts of process configuration changes on cost and schedule risk.

Sometimes a subdiagonal mark cannot be brought above the diagonal without pushing another mark below the diagonal. This is a case of interdependent activities, such as Activities 5 and 6. Each activity depends on the other. They must work together to resolve a "chicken and egg" problem. Typically, coupled activities work concurrently, exchanging preliminary information frequently. If a subset of coupled activities must begin before the rest, the more robust (less volatile and/or sensitive) information items should be the ones appearing below the diagonal in the DSM. If coupled activities are functionally-based, an opportunity may exist to fold the activities into a single activity assigned to a cross-functional team.

Integration, test, and design review activities typically have marks in their rows to the left of the diagonal. These activities create information (including results of decisions) that may cause changes to (and rework for) previously executed activities. Unfortunately, most process planners “plan to succeed” and their process models fail to account for these possibilities. Fortunately, the DSM provides an easy way to document potential “process failure modes” and their effects on other activities. The simple marks in the DSM can be replaced by numbers indicating the relative probability of information change, iteration, etc. This enables an analysis of process failure modes and their effects on cost, schedule, and risk. Process improvement investments can then target mitigation of the biggest risk drivers.

By accounting for contingent activities and feedback loops, the DSM can provide a basis for exploring adaptive processes. While the DSM itself is a static view of a process, the DSM can be updated over time to reflect a current situation. The remaining activities in such a situation could then be quickly resequenced in an advantageous way, providing rapid project replanning.

As a real-life example, Figure 2 displays a DSM of the Conceptual and Preliminary Design phases for an uninhabited combat aerial vehicle (UCAV).[2] The first dozen activities comprise the Conceptual Design phase. In this phase, design requirements and objectives (DR&O) are prepared, a configuration concept is proposed, it is analyzed by a variety of discipline perspectives, and then these results are assessed. The assessment may reveal a need to alter the DR&O, to create a new configuration concept, and/or to alter the current configuration concept. This cycle repeats until the design space is sufficiently understood and/or time and money are exhausted. The design process then moves into the Preliminary Design phase, where the configuration is developed and analyzed in more detail and the objective is to prepare a proposal to acquire funding for additional phases. Figure 2 shows the process “as is,” without any attempt to resequence the process to eliminate feedback. This basic model served as the basis for additional process analysis, evaluation, discussion, and improvement. It was also augmented with additional regions above and to the right, which represented external inputs and outputs, respectively.

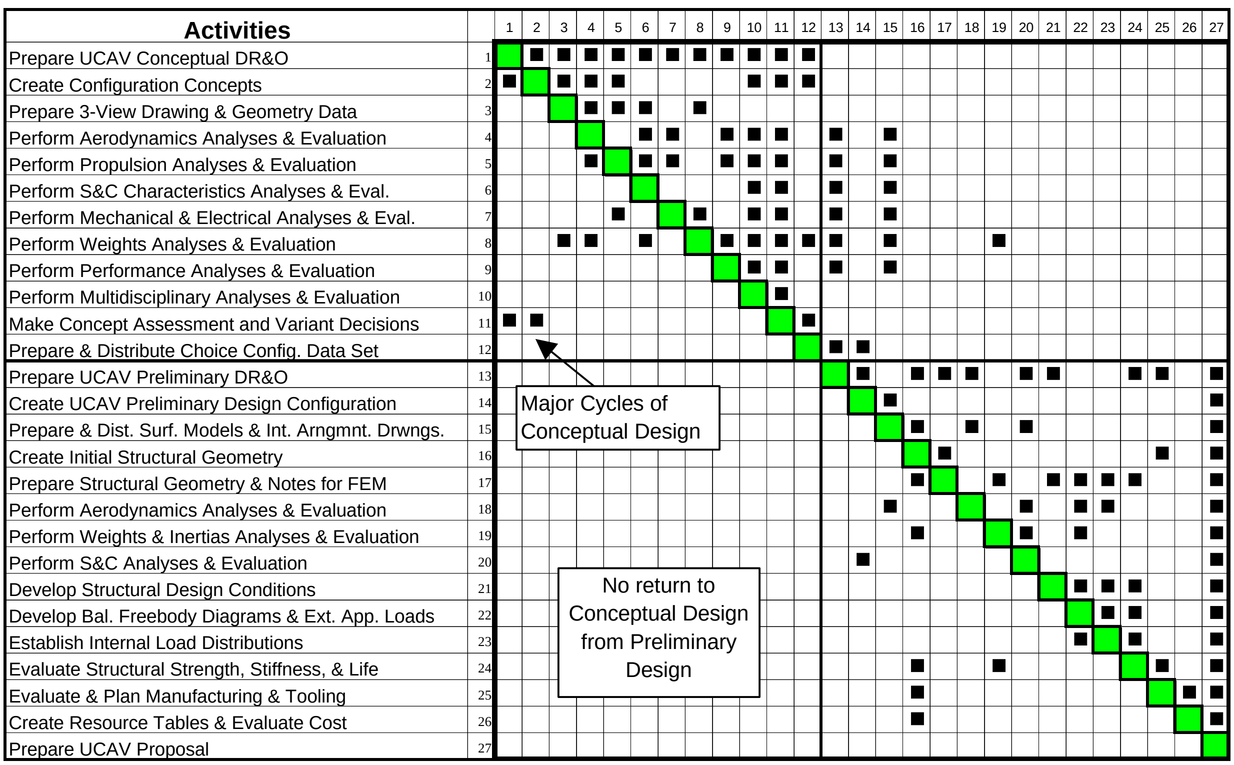

Figure 2: DSM of Conceptual and Preliminary Design Phases for a UCAV

The DSM provides a concise, visual format for representing processes. A process flowchart consuming an entire conference room wall can be reduced to a single page DSM. After a quick orientation, everyone can see how his or her activity affects a large process. People can see where information comes from and where it goes. They can see why delaying the activities they depend on forces them to make assumptions, which may trigger rework later. It becomes apparent that certain information changes tend to cause rework. Such situation visibility and awareness leads to improved process design and coordination. The DSM can provide a portal to a process

[2] The UCAV example is fully documented in (Browning 1998).

knowledge base from which the foundations of process plans and risk assessments can be drawn. Moreover, the DSM is amenable to some simple yet powerful analyses.

Conclusion

Thinking of and modeling an enterprise as a group of constituent *processes* and activities, executed by *organizational* elements, using *information* system elements (and other tools), and resulting in *product* elements, is a helpful paradigm. The DSM facilitates representing, analyzing, and understanding system architectures.

An intelligent enterprise will seek to understand its underlying systems and to ensure they provide sufficient and sustainable value to stakeholders, such as customers, shareholders, employees, suppliers, and society. If an enterprise plans and manages its work based on adequate knowledge of the work elements and their relationships, it will in effect understand how its "genome" drives behaviors and results (Browning, 2002). The network of activities and results provides the basis for making commitments and assigning accountabilities throughout the enterprise. Thus, it provides a "nervous system" for the enterprise, enabling it to sense and respond with greater agility (Haeckel, 1999; Pall, 1999).

About the Author

Tyson Browning is a Sr. Project Manager in Integrated Company Operations (an enterprise-level staff organization) at Lockheed Martin Aeronautics Company in Fort Worth, Texas, USA. He previously worked with the Lean Aerospace Initiative at MIT, where he earned two Master's degrees and a Ph.D. in Technology Management and Policy.

References

Browning, Tyson R. (1999) "Process Modeling with Design Structure Matrices (DSMs)," *INCOSE INSIGHT*, **2**(3): 15-17.

Browning, Tyson R. (2001) "Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions," *IEEE Transactions on Engineering Management,* **48**(3): 292-306.

Browning, Tyson R. (2002) "Process Integration Using the Design Structure Matrix," *Systems Engineering,* **5**(3): 180-193.

Browning, Tyson R. (1998) *Modeling and Analyzing Cost, Schedule, and Performance in Complex System Product Development,* Ph.D. Thesis (TMP), Massachusetts Institute of Technology, Cambridge, MA.

Haeckel, Stephan H. (1999) *Adaptive Enterprise: Creating and Leading Sense-and-Respond Organizations*, Boston, MA: Harvard Business School Press.

Lano, R.J. (1979) *A Technique for Software and Systems Design*, New York: North-Holland.

Pall, Gabriel A. (1999) *The Process-Centered Enterprise: The Power of Commitments*, New York: St. Lucie Press.

Steward, Donald V. (1981) *Systems Analysis and Management: Structure, Strategy, and Design*, New York: PBI.