Capt. Norbert Doerry, USN

Using the Design Structure Matrix to Plan Complex Design Projects

ABSTRACT

Ship Design and other related endeavors are characterized by the complexity of the interactions of the design team that often results in inferior designs, cost over-runs, and late delivery. This paper discusses the complexity of designing ships and the use of the Design Structure Matrix (DSM) to understand and reduce the complexity. Three design approaches are discussed and related to the different stages of ship acquisition: Synthesis Model Based Design Optimization, Set Based Design, and the classic Design Spiral. The Design Structure Matrix is introduced and related to IDEF0 activity modeling and Design Process Modeling. DSM enables the identification of coupled design activity "clusters" that suggest application of Integrated Product Teams, automated data interchange, and the exploration of alternate design methods (such as response surface methods) to decouple the design activities. A complexity metric based on the DSM is proposed. Finally, using DSM and other techniques in planning and executing complex design projects is detailed.

INTRODUCTION

The design of defense systems is complex and difficult. The Department of Defense's (DOD) experience in acquiring weapons systems is not good with respect to delivering products on time and within budget. The Government Accountability Office (Sullivan 2008) made the following observation:

"DOD is not receiving expected returns on its large investment in weapon systems. The total acquisition cost of DOD's 2007 portfolio of major programs under development or in production has grown by nearly $300 billion over initial estimates. While DOD is committing substantially more investment dollars to develop and procure new weapon systems, our analysis shows that the 2007 portfolio is experiencing greater cost growth and schedule delays than the fiscal years 2000 and 2005 portfolios. Total acquisition costs for programs in DOD's fiscal year 2007 portfolio have increased 26 percent from first estimates—compared to a 6-percent increase for programs in its fiscal year 2000 portfolio. Total RDT&E costs for programs in 2007 have increased by 40 percent from first estimates, compared to 27 percent for programs in 2000. The story is no better when expressed in unit costs. Schedule delays also continue to impact programs. On average, the current portfolio of programs has experienced a 21-month delay in delivering initial operational capability to the warfighter, and 14 percent are more than 4 years late."

Hence there is tremendous interest in improving both the design of systems to enable affordable production and to improve the design process itself to enable delivery of products to the customer on time and within budget.

Reinertsen (1997) recognizes that while manufacturing processes are generally repetitive in nature, the product development process generally must be designed specifically for each project. A complex project however, can be composed of standard and common design activities; customization of a design process is achieved through the selection of the appropriate design activities.

Choosing the proper design approach is also critical to success. For ship design, three general approaches are typically used: The classic Design Spiral, Synthesis Model based Design Optimization, and Set-Based Design. Each of these design approaches is a tool appropriate for different stages of design. The trick is using the right tool for the right problem; no one tool is universally optimal.

One of the leading barriers to successful execution of a design process is complexity. Suh (2005) recognizes four different dimensions of complexity: real, imaginary, combinatorial, and periodic. These different aspects of complexity should be addressed when developing a design process.

The Design Structure Matrix (DSM) is a useful tool for developing a Process Model. DSM methods can be used to identify the optimal ordering of design activities as well as mitigate and identify sources of complexity.

This paper also presents a number of other activities that are useful for preparing for and planning a complex design project. Following this guidance should help a design manager successfully lead a complex system design.

DESIGN APPROACHES

Historically, naval architecture and ship design has been taught using the Classic Design Spiral where an initial concept is iterated until the design has converged. More recently a host of Synthesis Model Based Design Optimization techniques such as response surface methodologies, design of experiments, genetic algorithms, and multi-domain optimization have used many point designs, typically generated by a computer based synthesis program, to characterize the design space for the purpose of identifying the optimal characterizations of a solution to a given set of requirements. More recently, Set-Based Designs have been employed to establish a point design based on an initial identification of the feasible design space. None of these methods is universally better than the others; each is a tool that is appropriate for different stages of the acquisition process and different acquisition strategies.

Classic Design Spiral – Point Based Design

The design spiral approach is a Point Based Design technique. As shown in Figure 1, design activities are accomplished in a specific order. At the end of each cycle around the spiral, design convergence is tested. If not converged, then another cycle at the same fidelity is repeated. If converged, then the next stage of design is entered where the steps are repeated at higher levels of fidelity.

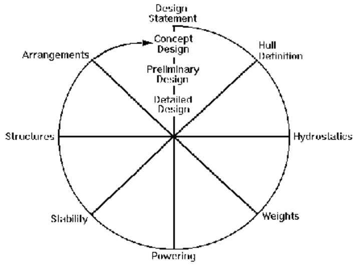


Figure 1: Classic Design Spiral

Figure 2 presents an alternate view of the Classic Design Spiral. Since each design iteration for a complex ship takes between 8 to 12 weeks, relatively few design iterations are possible within the 40 to 50 weeks typically allocated to a given stage of design. The design is "done" when you run of time, not necessarily when the design is converged or optimal. For this reason, the design spiral is most appropriate for refining an existing solution, rather than as a method for achieving the initial, almost-optimal converged starting concept.

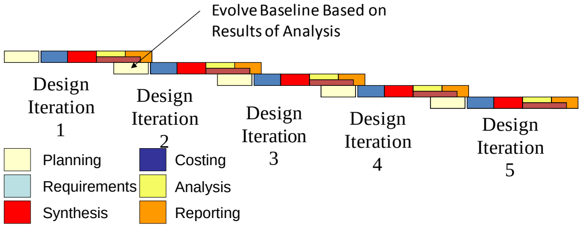


Figure 2: Alternate View of the Classic Design Spiral

Synthesis Model Based Design Optimization

Synthesis Model Based Design Optimization techniques are used extensively in the early stages of design to gain insight on the cost – performance trade-offs between requirements and the feasible material solutions. These methods include response surface methodologies, design of experiments (Doerry et al. 2002)(Drake et al. 2008), genetic algorithms (Neti 2005)(Brown and Salcedo 2003), and other multi-objective optimization techniques. These methods are generally characterized by the use of many point designs, typically generated by a computer based synthesis program. Because of the need to generate large number of ship concepts, the fidelity of the designs is generally only at the concept level. As the design matures and the required design fidelity increases, the ability to create large number of synthesized ship designs becomes too difficult to employ this method. The use of high performance computing environments with high fidelity synthesis programs and physics based analysis may extend the use of these optimization methods into pre-preliminary design.

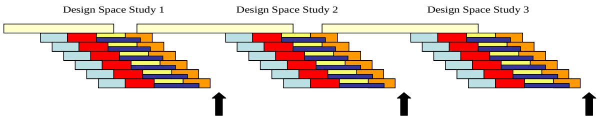


Figure 3: Synthesis Model Based Design Optimization

Set Based Design

Set-Based Design as described by Bernstein (1998) preserves design flexibility through three basic tenets:

> *“Understand the design space*
>   Define feasible regions
>   Explore tradeoffs by designing multiple alternatives
>   Communicate sets of possibilities
> *Integrate by intersection*
>   Look for intersection of feasible sets
>   Impose minimum (maximum) constraint
>   Seek conceptual robustness
> *Establish feasibility before commitment*
>   Narrow sets gradually while increasing detail
>   Stay within set once committed
>   Control by managing uncertainty at process gates”

In a set-based design process, engineers of different systems (i.e. electrical systems, combat systems, hull design, etc.) communicate ranges of solutions with associated derived requirements on other systems and levels of performance. As shown in Figure 4 regions of feasibility are determined by the intersections of the different ranges of solutions offered by the different engineering disciplines. Initially, the ranges of discipline solutions may need to grow to enable a sufficiently large region of feasibility at the intersection of independent solutions. The range of solutions for each engineering discipline is then reduced at the process gates to eliminate sub-system solutions that are not likely to contribute to a total system solution. Following the reduction in design space, engineers produce additional levels of details of the subsystems to refine the solution, improve cost estimates, and reduce risk. The design space is only reduced at a process gate if the design has sufficiently reduced the variability of design metrics to ensure with high probability that the eliminated portions of the design space are Pareto dominated by other regions. A solution is Pareto dominated when there are other solutions which perform better at lower cost. In this sense, Set-Based design is about eliminating solutions that are likely not optimum rather than picking one and modifying it to become an optimum. See Singer et al. (2009) for more details on Set-Based Design.

A marine engineering example of set based design would be the interaction of hull shape, propeller selection, and propulsion motor selection. For a range of required displacements and deck area, the hull designer would provide the range of speed – Effective Horsepower (EHP) curves and propeller size limitations. For this range, the propeller designer would provide the marine engineer with achievable propeller efficiencies, associated shaft speed – shaft power – ship speed curves along with maximum shaft speeds to preclude cavitation. The propulsion engineer would look at the range of powers and shaft speed required, and identify a motor architecture that could cover that region. The cost engineer would identify the cost and cost uncertainty that would apply to the different design spaces.

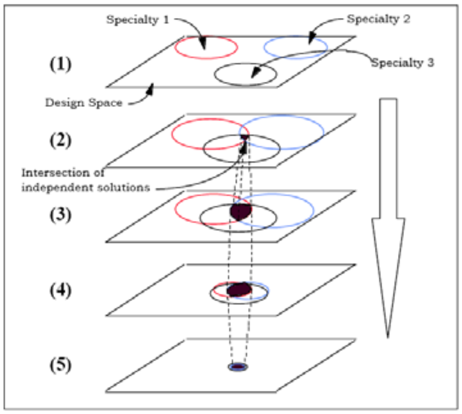


Figure 4: Set Based Design (Bernstein 1998)

ACQUISITION AND DESIGN

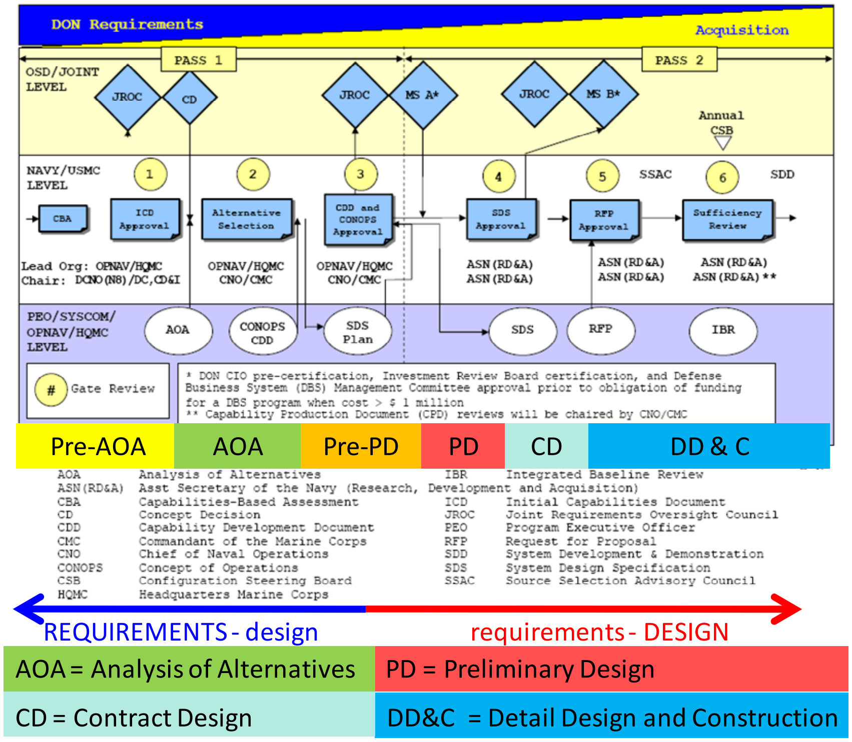
During 2008, the Secretary of the Navy (SECNAV) implemented a modified acquisition process as shown in Figure 5. This “2 Pass – 6 Gate” process ensures that the appropriate stakeholders are involved in acquisition decisions from the development of the Initial Capabilities Document (ICD) through Detail Design and construction. (SECNAV 2008 and 20008a) Figure 5 also shows the mapping of the traditional ship design stages onto the new process.

During the Pre-AOA and AOA phases, low fidelity automated models are typically used to systematically explore the design space in order to trade-off cost and performance. The synthesis model optimization techniques are appropriate to identify the region of the design space where the optimal solution is likely to reside. This region forms the basis of the Initial Capabilities Document and the selection of a broadly defined alternative from the analysis of alternatives.

Pre-Preliminary Design is a unique opportunity to perform trade-offs among individual system performance, total ship performance / requirements, the Concept of Operation (CONOPS) and cost. Because these activities are typically performed by many geographically dispersed organizations, Set-Based Design techniques are ideally suited for communicating individual design solution opportunities and requirements to systematically neck down the design space while improving design fidelity. By the end of Pre-Preliminary Design, the requirements are fixed in a Capability Development Document (CDD) and the Concept of Operation formalized in a CONOPS document. The ship design is developed to the level of detail necessary to produce a budget quality cost estimate. The Ship-to-Shore Connector (SSC) design is a good recent example of using Set-Based Design.

At the start of Preliminary Design following a Milestone A decision, the requirements and CONOPS for the ship are largely fixed. While some change is still possible, large changes are generally avoided. Set-Based Design can still be desirable to further refine system designs and integrate them into a total ship design. At some point, the design will “converge” and the point design based Classic Design Spiral is typically used to modify the design in response to detailed analysis, obsolescence management, and optimization efforts.

Use of the Design Spiral will typically continue through Contract Design, and Detail Design & Construction.



* DON CIO pre-certification, Investment Review Board certification, and Defense Business System (DBS) Management Committee approval prior to obligation of funding for a DBS program when cost > $ 1 million

** Capability Production Document (CPD) reviews will be chaired by CNO/CMC

AOA	Analysis of Alternatives	IBR	Integrated Baseline Review
ASN(RD&A)	Asst Secretary of the Navy (Research, Development and Acquisition)		
CBA	Capabilities-Based Assessment	ICD	Initial Capabilities Document
CD	Concept Decision	JROC	Joint Requirements Oversight Council
CDD	Capability Development Document	PEO	Program Executive Officer
CMC	Commandant of the Marine Corps	RFP	Request for Proposal
CNO	Chief of Naval Operations	SDD	System Development & Demonstration
CONOPS	Concept of Operations	SDS	System Design Specification
CSB	Configuration Steering Board	SSAC	Source Selection Advisory Council
HQMC	Headquarters Marine Corps		

AOA = Analysis of Alternatives
PD = Preliminary Design
CD = Contract Design
DD&C = Detail Design and Construction

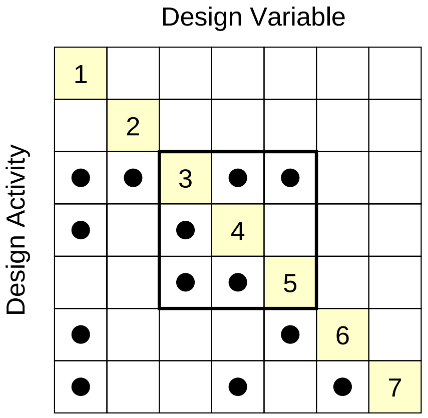
Figure 5: Navy 2 Pass 6 Gate Acquisition Process

DESIGN STRUCTURE MATRIX

DSM Overview

A Design Structure Matrix (DSM) compactly represents the relationships between design activities. Figure 6 shows an example of a DSM. In this representation, each of the rows corresponds to a Design Activity, and each of the columns a Design Variable. The numbered diagonal represents that Design activity for row *n* produces as output variable the design variable in column *n*. A dot in a cell indicates that the associated design activity for the row takes as input the design variable corresponding to the column of the dot. By sequencing the design activities within the DSM in the order of execution, much can be learned. Dots below the diagonal indicate variables that have been produced by previous design activities. Dots above the diagonal indicate variables that are needed by a design activity, but are not scheduled to be produced until the future. The value of the variable must be assumed, a "cluster" of activities as shown in Figure 7 must be solved simultaneously, or the design activities must be re-sequenced. Determining the optimal ordering of design activities is relatively easily accomplished with a DSM using well known matrix operations. One such method is describe in Appendix A.

Another insight that can be easily observed from a DSM is shown by variables 1 and 2 of Figure 7. These two variables do not depend on each other in any way and could be solved in parallel.



1						
	2					
●	●	3	●	●		
●		●	4			
		●	●	5		
●				●	6	
●			●		●	7

Figure 6: Design Structure Matrix Example

As the level of fidelity of design activities increase with time, the number of relationships between design activities as well as the total number of design activities is expected to increase. The design process should not be expected to be constant over the evolution of a design. The DSM provides valuable insight on how the design process must evolve as fidelity increases.

While Figure 6 shows the relationships between design activities and design variables as "dots", these dots can represent data structures defining the fidelity and data format used in the data transaction. Likewise, the diagonal "numbered boxes" could represent the data structure defining the characteristics of the design activity.

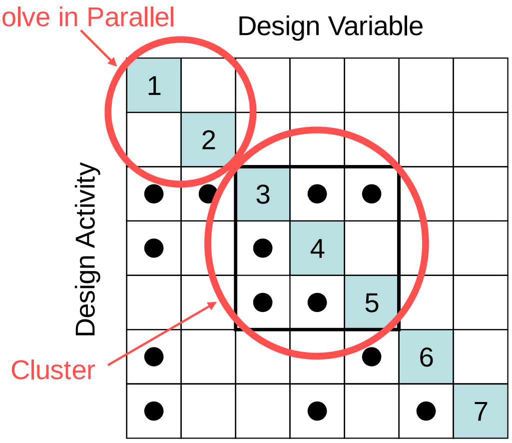


Figure 7: DSM Insights

Activity Modeling

There are many ways to model a ship design activity for integration into a DSM. A modeling technique that has proven useful over time is based on the IDEF0 definition of a function. As shown in Figure 8, a Design Activity interacts with external activities via Inputs, Outputs, Controls, and Mechanisms. Inputs are those data elements needed to perform the design activity. Outputs are those data elements that are produced by the design activity. Controls impact the manner in which the design activity is performed, and Mechanisms are the resources needed to perform the design activity.

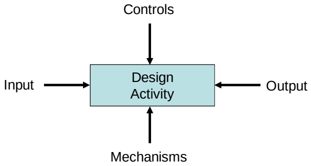


Figure 8: IDEF0 Activity Model

In executing a process defined by a DSM, much of the focus is on the interaction of the design activities via the Input and Output variables. In constructing the DSM from a set of design activities however, the controls become equally important; the controls govern the list of input variables, the properties of the input variables, and the properties of the output variables associated with the design activity.

The primary Control variable used in design activity modeling is the requisite fidelity of the Input and Output Variables. The level of fidelity of the Output variable may govern which design tool is used for that part of the design process; it may also require a different set of input variables of varying levels of fidelity.

Other Control Variables include input variable data formats, output variable format, type of hull, major hull material, and mission type.

Defining a design activity in this manner can result in multiple sets of design tools being employed depending on the Control Variables.

This dynamic nature of the number and type of input variables based on the value of a control variable differentiates ship design processes from classical IDEF0 process modeling. Consequently, a different technique for interconnecting design activity models is needed. Instead of IDEF0 process modeling, using a DSM for describing the inter-relationships of design activities is more appropriate.

More than one design activity can share the same output variable. For example, one design activity that fulfills the "Hull Resistance Analysis" function may be based on model testing while another may be based on detailed computational fluid dynamics. The two design activities could differ in the required input variables and would likely result in a differing set of Mechanisms.

Design Process Modeling

Figure 9 shows a simplified DSM based process model. In addition to the core DSM shown in Figure 6, it also shows dependencies on assumptions, as well as the dependency of output activities on the design variables and assumptions. Note that all the Outputs are independent of each other and all of the assumptions are independent of each other. Also note that while the DSM is inherently square, the number of assumptions does not have to equal the number of outputs, hence the Design Process Model is not required to be "square."

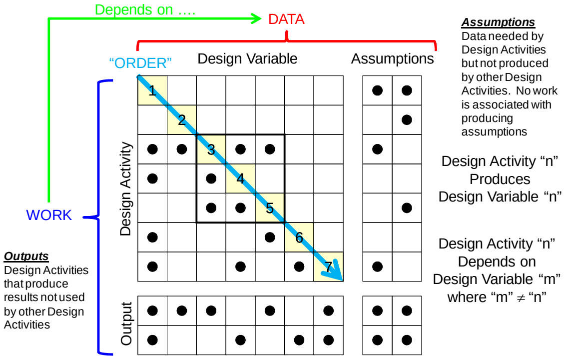


Figure 9: DSM Based Design Process Model

DSM Analysis

The DSM is a powerful tool for identifying integration issues within a given design process.

When using the design spiral, common practice is to ignore any dependencies above the diagonal of the DSM. Instead the results from the previous iteration are assumed for the design activity. In this way, the design spiral ensures a lower triangular DSM for a given design iteration. This will generally work if the dependencies above the diagonal are weak compared to other dependencies. In fact, Appendix 1 implements a method for classifying the dependencies as Weak or Strong to help ensure that the Strong dependencies are below the diagonal.

In certain cases, it will be impossible to eliminate a strong dependency from above the diagonal. In this case, convergence may be faster if this internal "cluster" of design activities is solved as a spiral within the overall ship design spiral.

Alternatively, the design activities and the variables they produce may be redefined to eliminate dependencies above the diagonal. One way is to redefine each design activity within a "cluster" to produce a response surface instead of a point solution. After all of the response surfaces from the design activities in the cluster are produced, an additional design activity is employed to find the intersections of the response surfaces which would constitute the design point for that iteration.

Another way to reduce the interdependency of design activities is through the development and use of standard open architectures with associated open interfaces and implementation of modularity. Design activities on either side of the interface can proceed largely independently of each other. (Smith and Reinertsen 1998) (Reinertsen 1997) In ship design, zonal design methods can reduce the complexity resulting from the inter-relationship of arrangements and survivability. (Doerry 2006)

Clusters of design activities should also be the focal point of efforts to automate data exchange, to conduct process improvement efforts in the conduct of the design activities, as well as to establish the boundaries for Integrated Product Teams (IPTs).

Because the Synthesis Model Based Design Optimization methods heavily rely upon computer based synthesis models, dependencies above the diagonal are generally addressed through an internal solver of the impacted design activities or by redefining the design activities as response surfaces as described above. An additional technique is also available where the dependency is initially ignored, but a check for "convergence" is made at the end. Solutions that do not converge are simply discarded.

In Set-Based-Design, the DSM can be effectively used to define the required dimensions of the "sets" or response surfaces provided by each discipline. At each process gate however, the level of fidelity of the design must improve. This implies that additional design activities and dependencies may be required. In planning a Set Based Design iteration, one has to understand these ever-increasing dependencies to ensure the domain specific design and analysis produces the requisite response surfaces (design variables) of the right dimensions (dependencies). The end result is that in execution a Set-Based Design DSM may have additional integration activities, but the DSM will be lower triangular.

COMPLEXITY

Complexity deals with functions and the way they interact and interfere with each other to prevent achieving the overall objectives. With this definition, complexity is a function of process, not product. It can also exist in multiple dimensions such as:

Design Complexity
Acquisition Complexity
Production Complexity
Testing Complexity
Operations Complexity
Maintenance Complexity
Modernization Complexity

While this paper concentrates on Design Complexity, many of the methods can also be used for the other dimensions of complexity. One should also note that the design process itself has a great influence on the other dimensions of complexity. Hence when we speak of "Design for Production" we are generally addressing ways to reduce Production Complexity. In fact we may elect to accept increased Design Complexity to reduce the other dimensions of complexity in search of the lowest Total Ownership Cost.

Design Complexity is hard to define, but its impact is well known. Bob Colwell claims complexity leads to fragile designs that are very sensitive to small perturbations. (Colwell 2005) It also complicates design management because few engineers understand the whole design. This can lead to sub-optimal design or different design teams working to cross-purposes. Colwell does not attempt to quantify complexity, but states it is a function of:

- "Number of ideas you must hold in your head simultaneously;

- Duration of each of those ideas; and

- Cross product of those two things, times the severity of the interactions between them."

Nam Suh (2005) defines complexity as:

"A measure of the uncertainty in understanding what it is we want to know or in achieving a functional requirement (FR). Functional requirements (FR) are defined, as in axiomatic design, as a minimum set of independent requirements that completely characterize the functional needs of the product in the functional domain."

Based on this definition, Suh further categorizes complexity into Real Complexity, Imaginary Complexity and Combinatorial Complexity. He also highlights the importance of functional periodicity for achieving stability over long periods of time.

Real Complexity

As defined by Suh, Real Complexity is time-independent and depends on the ability of the design activities to produce the requisite fidelity. That is, the probability that the design activity results are inaccurate. In DSM based process modeling, this can be addressed by having a good understanding of the Controls and Mechanisms to ensure the output variable has the requisite level of fidelity. The Controls can influence the number and required fidelity of the input variables.

Imaginary Complexity

Imaginary Complexity is a result of not being able to produce the desired results, not because of the inherent inaccuracies of the design activities, but because we don't know the optimal order of conducting the design activities. Ideally, the systematic use of DSM in planning design iterations should eliminate much of the Imaginary Complexity.

Combinatorial Complexity

Combinatorial Complexity results from having many dependencies between the design activities, especially those above the diagonal. In a design process with combinatorial complexity, it becomes difficult to determine how to adjust individual variables to ensure the design converges.

Functional Periodicity

Suh observes that systems that are long-lived and stable tend to have functional periodicity. Within the design processes described above, each method has distinct iteration boundaries or gates: each spiral of the design spiral, each generation of Synthesis Model Based Design Optimization, and each gate in Set-Based Design. These serve to "reset" the instabilities caused by Combinatorial Complexity.

Complexity Metric

A metric is a measure of something of interest. To be useful, one must be able to calculate or measure the metric and be able to place a value on the metric. Ideally an "improvement" in the metric should reliably result in an "improvement" in the desired outcome. There are many theoretical metrics for complexity, but most fail the test of being readily calculable.

In a previous paper, the author (Doerry 2006) proposed a complexity metric based on a Space Complexity Factor that in turn is a function of the number of systems and functional requirements that impact that space. This complexity metric recognized that many of the design activities in later stages of design are focused on the arrangement and design of individual spaces on a ship.

DSM Based Complexity Metric

The DSM however, offers the opportunity to develop a more generalized metric of Combinatorial Complexity. Combinatorial Complexity is singled out because it should have a strong influence on the planning for a design process. As shown in equation [1], the proposed metric is the sum of the square of the sizes of the clusters.

$$Complexity = \sum_{i=1}^{n} C_i^2 \qquad [1]$$

Where:

n *Number of Clusters*
C_i *Size of Cluster "i"*

For example, Figure 10 shows a DSM with complexity equal to 1+1+9+1+1=13. Eliminating the cluster of size 3 by redefining design activities 3, 4, and 5 and inserting a new integration activity 6, the complexity becomes 1+1+1+1+1+1+1+1=8. In this manner, beneficial changes to the design process can be measured and articulated to senior management as a reduction in the complexity metric.

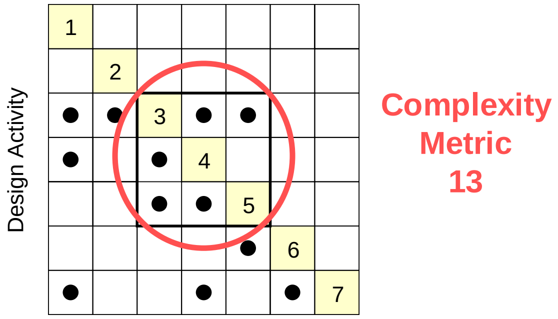


Figure 10: Initial Complex DSM

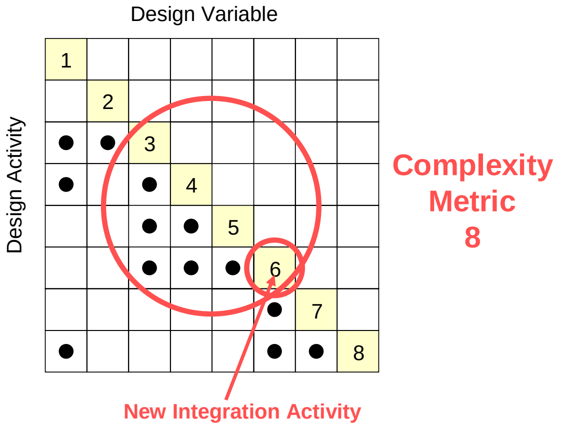


Figure 11: Less Complex DSM by redefining Activities

PLANNING COMPLEX DESIGN PROJECTS

Complex design projects are often characterized by design teams comprised from multiple organizations that may be geographically dispersed. Understanding the entire design process and the interaction of the different design activities is very difficult. Often, there is no one who has detailed knowledge of every design activity; at best the design integrators understand their interactions and view at least some of the design activities as "black boxes."

Because each design is somewhat unique, success is not assured by replicating the design process from previous projects. Rather, specific planning activities should be undertaken to increase the probability of project success.

Develop Study Guides

The primary purpose of a study guide is to document the design problem, the desired outcome, assumptions, metrics for success, and the design process. All important stakeholders (including the customer or customer representative) should concur to the contents of the study guide. If there are problems with the study planning and execution, it is best to identify them as early as possible. If all stakeholders agree to the assumptions and the process, then they are also more likely to concur to the design outcome. Without going through the Study Guide process, projects often expend an inordinate amount of time at the end trying to "sell" the resulting design. The author's experience is that a good study guide can considerably reduce the amount of time to bring a project to a successful conclusion.

Work can start before the study guide is finalized, but the design manager must plan for the possibility that changes may be necessary as the study guide is finalized. Any areas of contention should be highlighted and simple, inexpensive tests devised and executed to resolve them.

Execution planning usually occurs simultaneously with development of the Study Guide. To keep on schedule however, work orders and tasking may have to be issued before all the details of the Study Guide are worked out. Still, it's better to resolve disagreements early, when the project still has funds available to implement the resolution, rather than at the end when all the funds have been expended.

Develop Design Process Model

A DSM based Design Process Model is incredibly useful for gaining an understanding of the overall design process and for identifying sources of complexity. Ideally, the design organization will have deployed a Design Process Modeling System similar to that shown in Figure 12. A Design Activity Library is continually updated with IDEF0 models and Standard Statements of Work for all of the design activities. The Design Manager selects design activities from the library to form a DSM based process model. A DSM Optimizer identifies the optimal ordering of design activities and highlights the "clusters." The Design Manager then has the option to edit the design activities to eliminate the "clusters" or can elect to deal with the complexity in another way. The DSM based process model can then be used to generate a traditional Gantt chart based schedule to help in managing project execution.

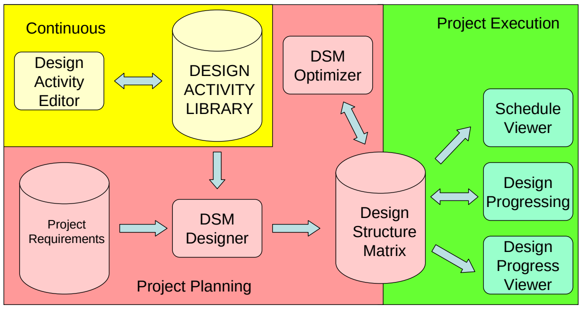


Figure 12: Design Process Modeling System

Understanding and eliminating the "clusters" is an effective way of reducing the combinatorial complexity of the design process. Whitcomb and Szatkowski (2000) demonstrated reducing the complexity of an early stage ship synthesis model by analyzing a DSM and eliminating the "clusters." Alternate design process models can be evaluated and ranked using the complexity metric defined in equation [1].

Understand and Control Variance

Managing the real complexity requires an understanding of variances in the design variables throughout the design process. The variance of the design variables produced by a design activity is a function of the variance of the input variables as well as the variance of the design algorithms. In other words, a design activity that perfectly calculates its output variables will still result in the output having variance due to variance in the input. Likewise a simplified, parametric based design activity will likely have variance in its output even if provided perfect inputs.

One of the controls that should be common to all design activities is the desired variability (or fidelity) of the output variable. This variability may be achieved by either using more accurate design tools/methods with inputs having higher levels of variance, or by using design tools/methods with more variance, but requiring more accurate inputs. The process designer has the ability to trade these off. Developing good design activity models is key to enabling this trade-off.

Design and Task the Workforce

The design activities included in the DSM Process Model provide a natural work-breakdown structure for organizing and tasking the workforce. Consideration should be given to co-locating the workforce of tightly coupled design activities. Alternatively, one can apply IPTs or automated digital data exchange to tightly coupled design activities.

Train the Workforce

Imaginary Complexity is reduced significantly when the workforce is properly tasked and trained.

The workforce must have an understanding of the design activities they participate in and how their design activities integrate into the overall process. They must understand how inputs and constraints are provided to them, the variance in the input data, and how to estimate the variance of the output data. They must know how to provide the output data in the form needed for integration. The workforce must know and be able to articulate assumptions that their design activities inherently rely upon and be able to communicate to the design integrators when those assumptions are violated.

Use DSM to Track Execution

A DSM can also be a good visualization tool for tracking progress of a design. Figure 13 shows an example where one can immediately see which activities are complete. For the activities that have not been started, identifying the status of the predecessor activities is easy.

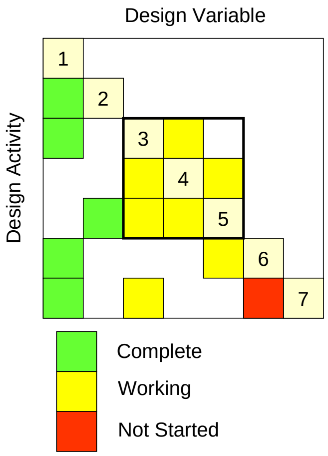


Figure 13: Using a DSM to track execution

Use Peer Reviews

The purpose of Peer Reviews is to identify and correct mistakes as early as possible. A Peer Review differs from a Design Review in that a Peer Review concentrates on one or more design activities and their interaction with other design activities while Design Reviews are more concerned with the design itself. Peer Reviews should also examine the appropriateness of any assumptions made as well as ensuring that the design methods and tools are appropriate for the design project.

Peer Reviews should be conducted with experts in the field of the design activities and the interfacing design activities. Typically Peer Reviews should be done both at the design activity level and at the total system integration level.

The observations and recommendations of the Peer Review should be recorded in minutes that are formally managed. The design team should formally respond to each item in a formal document.

The timing of when to hold the peer review depends on the maturity of each design activity. Ideally, a peer review should be held no earlier than after the first complete iteration of the design activity – the peer reviewer should be reviewing real products, not just a plan. Towards the end of a project, the peer review should be conducted early enough such that any mistakes or problems identified can be corrected in time without impacting the critical path.

CONCLUSIONS

This paper has provided recommended practices for planning complex design projects. Complexity has been presented in terms of four dimensions:

- Real
- Imaginary
- Combinatorial
- Functional Periodicity

The Design Structure Matrix has been presented as a means of developing a Design Process Model and identifying sources of Combinatorial Complexity. Having and using the Design Process Model also helps control the Imaginary Complexity. Functional Periodicity is accomplished through the use of one of the three standard design approaches:

- Classic Design Spiral
- Synthesis Model Based Design Optimization
- Set Based Design

Each of these design approaches is appropriate for different stages of design. In ship design, Synthesis Model Based Design Optimization is done in the earliest stages, followed by Set-Based Design during Pre-Preliminary Design and perhaps during the first few iterations of Preliminary Design. The Classic Design Spiral is typically used starting in Preliminary Design and continuing through Detail Design.

Real Complexity is addressed by understanding the variability of design variables and the design activities that produce them. Good design activity modeling will help control Real Complexity.

A number of activities have also been suggested for helping ensure a successful design of a complex system.

REFERENCES

Bernstein, Joshua, “Design Methods in the Aerospace Industry: Looking for Evidence of Set-Based Practices,” Thesis for the degree of Master of Science at the Massachusetts Institute of Technology, Technology and Policy Program, May, 1998.

Brown, Alan, and Juan Salcedo, “Multiple-Objective Optimization in Naval Ship Design,” Naval Engineers Journal, Vol 115, Issue 4, Oct 2003, pp 49-62.

Colwell, Bob, “Complexity in Design,” *Computer*, IEEE Computer Society, Vol 38. Issue 10, October 2005

Doerry , CDR Norbert, William H. Austin III, Erik Strasel, "JCC(X) Concept Exploration", presented at the ASNE Engineering the Total Ship Symposium, ETS 2002, February 26-27, 2002.

Doerry, CAPT Norbert H., "Systems Engineering and Zonal Ship Design," presented at the ASNE Engineering the Total Ship Symposium, ETS 2006, May 1-2, 2006

Drake, Kimberly J., Michael Robinson, Patrick Violante, Sara E. Wallace, and David Woodward, “Intelligent Tools Used in the Design of Navy Ships,” ASNE NEJ, Vol 120, Issue 1, March 2008, pp. 41-49.

Neti, Swaroop N., *Ship Design Optimization using ASSET*, Master of Science in Ocean Engineering Thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, Feb 20, 2005.

Reinertsen, Donald G., *Managing the Design Factory*, The Free Press, 1997.

Secretary of the Navy (SECNAV 2008), “Department of the Navy (DON) Requirements and Acquisition Process Improvement,” SECNAVNOTE 5000 of Feb 26, 2008.

Secretary of the Navy (SECNAV 2008a), “Implementation and Operation of the Defense Acquisition System and the Joint Capabilities Integration and Development System,” SECNAVINST 5000.2D of October 16, 2008.

Singer, David J., Norbert Doerry, and Michael Buckley, “What is Set-Based Design?,” ASNE Day 2009, April 8-9, 2009, Washington, DC.

Smith, Preston G., and Donald G. Reinertsen, *Developing Products in Half the Time*, 2nd Edition, John Wiley & Sons, Inc., 1998.

Suh, Nam P., *Complexity*, *Theory and Applications*, Oxford University Press, 2005.

Sullivan, Michael J., "Fundamental Changes are Needed to Improve Weapon Program Outcomes," GAO-08-1159T, September 25, 2008.

Whitcomb, Clifford A., and John J. Szatkowski, "Concept Level Naval Surface Combatant Design in the Axiomatic Approach to Design Framework," Proceedings of ICAD2000, First International Conference on Axiomatic Design, Cambridge, MA, June 21-23, 2000.

Capt. Norbert Doerry (Ph.D. Naval Electrical Power Systems – MIT '91, SMEECS, NE – MIT '89, BSEE USNA '83) is an Engineering Duty Officer currently assigned as the Technical Director for Future Concepts and Surface Ship Design Group (SEA 05D) in the Naval Sea Systems Command. Previous tours at NAVSEA include Technical Director for IPS and Ship Design Manager for JCC(X). He additionally served as an Assistant Project Officer for Aircraft Carrier repair and new construction at SUPSHIP Newport News and as the Assistant Acquisition Manager for LHD 8 within PMS 377. Prior to becoming an Engineering Duty Officer, he served as gunnery and fire control officer on USS *Deyo*.

APPENDIX A: A Method for Ordering Design Activities

One of the advantages of using a Design Structure Matrix is that one can use matrix manipulation algorithms to determine the order that one should conduct design activities within a design iteration. Design activities in general can be defined as the effort needed to produce a design variable. Design activities can incorporate the use of computer programs, manual calculations and defining assumptions. Each design activity can either strongly or weakly depend on the results, called design variables, of other design activities. The goal of determining the order is to ensure that any design activity upon which another design activity strongly depends on is performed first. Unfortunately, some sets of design activities strongly depend on each other and must be solved as a block or “cluster”. Weak dependencies can be treated less strictly; if a variable that a design activity has a weak dependency on has not been calculated in a previous block of the design iteration, then the value of the variable from the previous design iteration is used.

Definitions:

$x = f(x)$

x = a vector of vectors. Each element of x, or x_i, is a vector describing the results, called the design variable, for a design activity.

$f(x)$ = a vector of design activities that generate x.
$f_i(x)$ = a design activity or tool that generate x_i.

P = the Design Structure Matrix.

Each row of P, or P_i, corresponds to a design activity to produce the i element of x, or $f_i(x)$.

Each element of P, or P_{ij} is an element of the set {0,W,S}. where

0 means that the design activity $f_i(x)$ does not depend on the design variable x_j
W means that the design activity $f_i(x)$ only weakly depends on the design variable x_j
S means that the design activity $f_i(x)$ strongly depends on the design variable x_j

The goal of the process is the development of the vector d. Each element of d consists of two elements:

d_{i0} = the block number – design activities are performed in the order of their block number. All the design activities within a “cluster” will have the same block number. Design activities that are not part of a cluster and are only weakly coupled with each other also share the same block number.

d_{i1} = the sub-block number – if 0, then all the design activities of the sub-block are all strongly dependent on each other, form a “cluster” and must be performed as a set (generally as a sub-iteration within the larger design iteration). If non-zero, then while design activities can be performed independently of each other, the preferred order, taking weak dependencies into account, is to perform them in the order of the sub-block number.

Methodology:

1. Initialize
Set the block number index counter to 1
Set all elements of d to 0

2. Find order 1 blocks

Search rows of P, only look at rows and columns for which $d_{i0} = 0$, but ignore the diagonal.

Set the subblock index counter to 0
For each row, count the number of S are in columns where $d_{i0} = 0$

```
For any row without any S's,
    set $d_{i0}$ to the block number index counter
    increment the subblock index counter
    set $d_{i1}$ to the subblock number index counter
```

```
If any rows were found without any S's
    Increment the block number index counter
    Repeat step 2.
```

3. Find order 2 blocks

Search rows of *P*, only look at rows and columns for which $d_{i0} = 0$, but ignore the diagonal.

Keep a list of rows that only have one S in any columns for which $d_{i0} = 0$

If the list has fewer than 2 rows, then there are no order 2 blocks, go on to step 4.

Check to see if the list is self contained. Eliminate any row that has an S in a column that corresponds to a row that is not in the list. Each time you eliminate a row, all the previous rows have to be rechecked.

If the list has fewer than 2 rows, then there are no order 2 blocks, go on to step 4.

If the list has exactly 2 rows, then you have a solution, mark them as the next block and go back to step 1 to start looking for order 1 blocks again.

3a. If the list has more than 2 rows, then have to start trial solutions.

Start at the first row in the list, mark that row, and all the rows it depends on.

For each marked row, ensure all the rows that it depends on are also marked.

If you have more than 2 marked rows, then the first row can not be part of the solution.

` Mark it for deletion and all the rows that depend on it for deletion.

For each marked rows, ensure all rows that depend on it are also marked for deletion.

Delete from the list all the rows marked for deletion.

If there are fewer than 2 rows remaining, then there are no order 2 blocks, go to step 4.

If there are exactly 2 rows remaining, you have a solution, mark them as the next block and go back to step 1 to start looking for order 1 blocks again

If there are still more than 2 rows remaining, Go back to step 3a.

4. Continue looking for blocks of increasing orders (using the general algorithm of step 3) until all elements of d_{i0} are non-zero. Once a block of larger order is found, must go back and look for blocks of size 1 and progress incrementally to larger numbers.

Once the above algorithm is completed, one can use the Weak dependencies to reorder the clusters that can be solved in parallel. Essentially, the process is to identify the clusters that can be solved in parallel (they do not strongly depend on each other), then examine the weak dependencies. It is straight forward to identify the predecessor design activity for each cluster (see below). Clusters can be accomplished in parallel if all of the predecessor activities of the two cluster are earlier than all of the design activities comprising the two clusters. For two clusters that can be solved in parallel, if one cluster weakly depends on the other, but not vice versa, then it should be solved after the cluster it depends on.

For clusters of size 1, reordering clusters involves swapping values for d_{i1}

For clusters of size greater than 1, reordering clusters involves swapping values for d_{i0}

Notes:

Under certain conditions, blocks with a higher block index may be started earlier in parallel with blocks with lower block indexes. For the rows of *P* corresponding to the block, the variables corresponding to columns that contain an S must have been previously defined or part of the block. To find the actual predecessor block, look in all the rows of *P*. Search all the columns not in the block for an "S". Find the column with an S and the largest block number. This block number corresponds to the predecessor activity for the block.

The final step is to reorder *P* so the rows and columns are in the order of the block number / sub-block number. Doing so will result in the "S" elements being predominately in the lower left triangle of the reordered *P* matrix. The only "S" elements in the upper right triangle of the reordered *P* matrix will be part of a cluster.