SNAME Maritime Convention 2022
26-29 September, Houston, TX
Copyright © 2022 Society of Naval Architects and Marine Engineers (SNAME)
www.sname.org

Design Activity Modeling

Norbert Doerry[1] (FL), **Andrew Johnson**[2] (V), **Edward Ammeen**[3] (V), **Katherine Beaumont**[2] (M)
1. Naval Surface Warfare Center, Carderock Division
2. Gibbs & Cox, a Leidos Company
3. Cardinal Engineering

The design phase of a naval vessel is extraordinarily complex. The numerous interactions between design specialties have great potential to result in misunderstandings and confusion among engineers, which can then lead to deadline extensions or unnecessary rework. To overcome the difficulties coordinating ship design specialties, our effort seeks to simulate candidate design processes through the construction of a complete ship design planning tool. This planning tool lays out the effects of resources, requirements, and product quality on the overall design schedule, allowing ship design managers (SDMs) to more efficiently task and control work. The tool specifically includes the impacts of iterations on the overall convergence of the design. As a demonstration we focused on modeling the power and propulsion elements of the ship design.

KEY WORDS: surface combatants; ship design; modeling; simulation; program management; project management; scoping

NOMENCLATURE

BDD	Block Definition Diagram
BRAC	Base Realignment and Closure
CMMI	Capability Maturity Model Integration
COSYSMO	Constructive Systems Engineering Cost Model
CSM	Cameo Systems Modeler
DAM	Design Activity Model
DSM	Design Structure Matrix
GUI	Graphical User Interface
IBD	Internal Block Diagram
IDE	Integrated Data Environment
IDEF0	Integration Definition for Process Modeling
IPES	Integrated Power and Energy System
IPT	Integrated Product Team
MBSE	Model-Based Systems Engineering
MDD	Model Description Document
NAVSEA	Naval Sea Systems Command
NSWC	Naval Surface Warfare Center
PDE	Product Data Environment
SDM	Ship Design Manager
SOW	Statement of Work

INTRODUCTION

In this paper, we present a novel ship design model and tool for modeling and simulating the power and propulsion domain of the Ship Design Process. The model and tool provide a Ship Design Manager (SDM) with the ability to simulate many different design schedules with any number of varied parameters to determine the optimal sequence of power and propulsion design activity focus and support resource allocation to maximize efficiency in ship design. The ship design model, although applied to the power and propulsion system, is applicable to the entire ship design process.

To construct this ship design power and propulsion tool, we take advantage of MagicDraw®'s ability to integrate with MATLAB® in a digital environment. Through this integration, mathematically and logically complex MATLAB® functions written to simulate certain ship design activities can be contained in a single "*Call Behavior Action*" in an Activity Diagram in MagicDraw®. Thus, any number of these functions can be placed and called in a certain sequence to simulate the degree of completion, defined as quality in the model, of a ship design activity over several time steps, and the effect of the design activity's quality on the quality of other design activities - thereby simulating the necessary interaction between design specialties.

The SDM can fully customize the design activity environment and has control of dozens of parameters. These parameters include, but are not limited to, the numbers of engineers assigned to a design activity, the efficiency of junior engineers, the actual amount of work to be accomplished, the level of tool support, personnel experience, and even the level of documentation available. The tool also provides various metrics to the SDM during the simulation to indicate details. These details include what exactly is limiting any design activity's quality within its own domain, whether the quality of any design activity is hindering the convergence of other design activity qualities, what kind of work a design activity is engaged in (setup or recurring), and other metrics.

CURRENT STATE OF THE ART

Traditional Approaches

The design of a naval warship is complex in that it involves the interaction of a number of different design specialties. The experts of these design specialties may not fully understand the work carried out by the other design specialties, which may result in misinterpretation and misunderstandings leading to designs not meeting their requirements, or significant rework. This naturally occurring "friction" in the design process is a form of design complexity that should be addressed and controlled by the design management team (Suh 2005). Prior to acquisition reform in the mid-1990s, it was common to use co-located design teams working in either functional teams or later in Integrated Product Teams (IPTs) to facilitate communication and directly address this form of design complexity (See Keane et al. 2009). This was possible because the design expertise largely resided within commuting distance of Washington DC. Base Realignment and Closure (BRAC) of the 1990s however, resulted in ship design expertise being transferred to the Naval Warfare Centers, which were located throughout the United States. It was no longer possible to assemble a co-located team of the experts in all aspects of ship design.

In the 2007-2010 timeframe, as the Naval Sea Systems Command (NAVSEA) prepared for the preliminary design of a cruiser (CG(X)), a series of ship design process workshops was held. The purpose of the workshops was to baseline the ship design process as it was being conducted and then to modify it to reflect design approaches, such as Set-Based Design, which were better suited for a distributed work force (Singer, Doerry, and Buckley 2009 and Singer et al. 2017). The development of design tools and methods would also be facilitated from an understanding the design process. While much progress was achieved in defining the ship design process understood at that time, the lack of further funding precluded adapting that ship design process to one more suited for a distributed workforce. The cancelling of the CG(X) program eliminated the urgency for continuing this effort. See Helgerson, Billingsley, and Doerry (2009) and Cooper et al. (2011) for details of this effort.

This previous effort highlighted the iterative nature of preliminary design. A design activity may require the products of other design activities that in turn require the products of the original design activity. The classic design spiral shown in Fig. 1 (Evans 1959) highlights this dependency and the resulting need for iteration.

Current practice is to use traditional Gantt chart-based scheduling tools. These tools enable one to schedule activities, identify dependencies, identify the critical path, and predict resource loading. However, these tools do not handle iteration effectively. Each iteration becomes a new set of activities, and convergence is assumed after a given number of iterations. No attempt is made to explicitly identify iteration and rate of design convergence to determine if the assumed number of iterations is sufficient.

Gantt chart-based scheduling tools have another issue. Typically, many of the relationships between activities are not shown to simplify the presentation. If the relationships between all activities were shown, the Gantt chart would likely be covered with lines and arrows. Consequently, practitioners often choose to eliminate dependencies in an effort to enhance clarity.

Related methods, such as the Program Evaluation and Review Technique (PERT) (Honsinger 1968, Ballou 1966) share in this difficulty in modeling iteration.

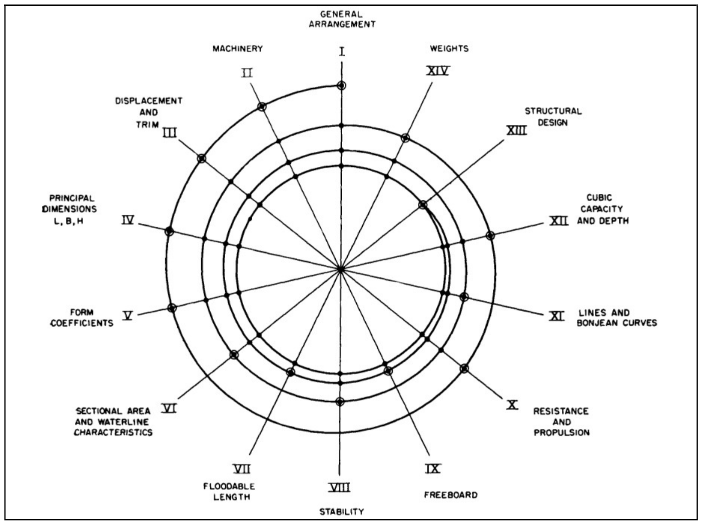


Fig. 1, Design Spiral (Evans 1959)

Design Structure Matrices (DSM) are ideal for highlighting relationships among design activities, identifying iteration cycles, and offering suggestions for optimal ordering of design activities (Doerry 2009, Eppinger and Browning 2012). However, DSMs do not inherently have ways to determine how many iterations of a cycle are required. Typically, iteration is handled by specifying a probability that an iteration cycle must be repeated (Lukas 2007, Browning and Eppinger 2002). Unfortunately, there is little guidance for determining these probabilities.

A metric for measuring design convergence is needed. There should be a means of calculating this metric based on properties of each design activity and the relationships among the design activities. This project employs such a metric based on the concepts of product quality. Using this metric, convergence can be predicted. Additional metrics are defined to identify what is limiting convergence at any time step. These additional metrics help identify how the schedule and resource loading should be adjusted to improve convergence.

MODELING APPROACH

The next destroyer (DDG(X)) will soon enter preliminary design. Other surface ship designs will follow in the future. An activity model of the ship preliminary design process can help manage the design effort, improve the quality of the design, and identify the prioritization of tools development efforts. The purpose of this project is to develop an activity-based design process model of a portion of the overall ship design process to include prime power generation, propulsion, prime electrical power distribution, and in-zone electrical power distribution.

Design Activity Model Use Case

The use case for employing the design activity model is depicted in Fig. 2. The design activity model is customized to estimate and optimize cost and schedule to execute the design process for a given project. Based on the results of the optimization, tasking statements are generated, as well as products (such as Gantt charts) to assist in managing the design effort. During the execution of the design, the design activity model can also be used to adjust the plan in terms of application of workload resources and schedule in response to unforeseen circumstances and thus speed convergence of the overall design effort.

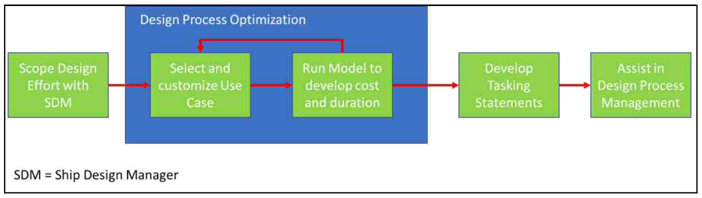

Design Process Optimization
Scope Design Effort with SDM
Select and customize Use Case
Run Model to develop cost and duration
Develop Tasking Statements
Assist in Design Process Management
SDM = Ship Design Manager

Fig. 2, Design Activity Model Use Case

Overview of Mathematical Model

While design activities can be modeled in many ways, this project employs an activity model effectively based on IDEF0 as depicted in Fig. 3. In this model, an activity interacts with other activities and control elements via inputs, outputs, controls, and mechanisms. An activity is work done by one organization to produce outputs based on the inputs and controls using the resources described in the mechanisms.

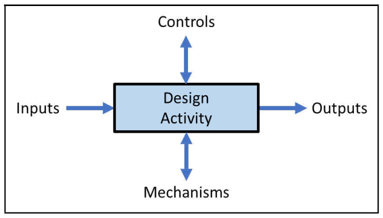


Fig. 3, IDEF0 Model of Design Activity

As depicted in Fig. 4, an activity model is distinct from the activity itself in that the activity model does not produce the outputs, but instead estimates the degree of completion of the activity outputs as a function of time. This estimation is then used in part in estimating the quality of the activity output as a function of time. The quality of the activity outputs is also based on the quality of the input variables. The degree of completion and quality of the activity outputs are estimated without actually producing the activity outputs.

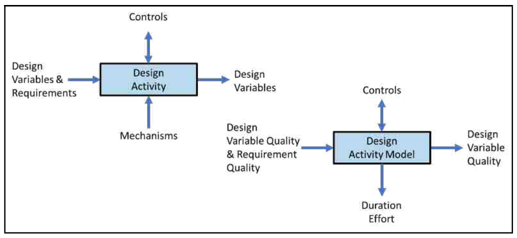


Fig. 4, Design Activity vs. Activity Model

The inputs of an activity consist of design data and requirements. Design data is produced by other activities within the design activity model to describe characteristics of the system being designed. Requirements are produced as part of the systems engineering process or in activities external to the design activity model to reflect the needs of the end users or the results of the external activities. Requirements development is considered external to the design effort being modeled. Table 1 lists the requirements.

Table 1. List of Requirements

REQUIREMENTS (inputs)	
Power System Requirements	R010
Ship Operating Conditions	R020
Margin and Service Life Allowance Policy	R030
Ambient condition Profile	R040
Machinery Arrangements	R050
General Arrangements	R060
Master Equipment List	R070
Combat Systems Design	R080
Other Distributed System Design	R090
Speed Power Curve	R100
Survivability Requirements	R110
QOS Requirements	R120
Endurance Requirements	R130
Operational Profiles	R140
System Safety Plan	R150
Security Controls (from Risk	R160
Product Support Analysis Plan	R170
Flexibility Requirements	R180
Build Plan	R190
Zone Boundaries	R200
SDM Guidance	R210

The outputs of an activity include design variables/data as the main products of the activity. A list of the design activities and their products (outputs) is presented in Table 2.

Controls consist of triggers and associated control data. Triggers are control signals to or from activities and activity models to indicate the following:

1. The activity or activity model should execute (i.e., start trigger). Start triggers are sent to an activity or activity model and include details of the type of work to perform in the form of control data and mechanisms. An activity or activity model executes when a start trigger is received.
2. The activity or activity model has completed execution (i.e., completion trigger). A completion trigger is sent by an activity to indicate it has completed and uploaded outputs, mechanisms, and possibly control data to a Product Data Environment (PDE). A completion trigger is implemented at the end of each week during the time period that an activity is being performed.

Table 2. List of Activities and Products

Activity	Product	Product ID
Power System Architecture	List and Discription of Power and Propulsion System Architectures	100
EPLA	EPLA-Power	200
	EPLA-Energy	210
	EPLA-Inrush	220
	EPLA-Pulse	230
Load List	Load List	300
Primary Power System Element Design	Primary Power System Element Design and Operating Methods	400
Zonal System Element Design	Zonal Power System Element Design and Operating Methods	500
Propulsion System Design	Propulsion System Design	600
Casualty Power System Design	Casualty Power System Design	700
EPS and Propulsion CONOPS	Electrical Power System Concept of Operation	800
	Propulsion Plant Concept of Operation	810
Electrical and Propulsion Control System Design	Electrical and Propulsion Control System Design	900
Endurance Fuel and Annual Fuel Calcs	Endurance and Annual Fuel Calculations	1000
Dynamic Simulation	Transient Analysis	1100
	Stability Analysis	1110
	Dynamic Response Analysis	1120
	Common Mode Current Analysis	1130
	Fault Current Analysis and Protective Device Coordination Study	1140
	Harmonic and Non-Fundamental Frequency Analysis	1150
	Thermal Analysis	1160
Reliability Analysis	Reliability Analysis Report	1200
QOS Analysis	QOS Analysis Report	1300
Vulnerability and Recoverability Analysis	Zonal Survivability Analysis Report	1400
	Compartment Survivabiilty Analysis Report	1410
Arc Flash Analysis	Arc Flash Analysis Report	1500
System Safety and Hazard Analysis	System Safety and Hazard Analysis Report	1600
Cybersecurity Analysis	Security Assessment Plan and Security Controls Assessment	1700
Product Support Analysis	Product support Analysis Report and Logistics Product Data	1800
Human Engineering Analysis	Human Engineering Analysis Report	1900
Develop Spec Sections	Specification Sections (see below)	2000
Power System Flexibility Strategy	Power System Flexibility Strategy	2100
Assess Power System Flexibility	Power System Flexibility Assessment	2200
Electrical System and Propulsion DT&E	DT&E Test Plan, Test Procedures, and Test Reports	2300
Mission System - Power System Control Interface	PPD: Mission System - Power System Control Interface	2400
Cost Engineering Analysis	Cost Engineering Analysis Report	2500
Develop Configuratrations	Configuration Descriptions	2600
Set Reduction	Design Space Classification Report	2700

Start and completion triggers enable design processes to be created via the following:

1. Directly linking the completion trigger of an upstream activity to the start trigger of a downstream activity, or
2. Interacting with a control element that centrally manages the creation of start triggers and control data based on the receipt of completion triggers and the quality of design variables and requirements.

This project employs a control element to manage triggers.

For this project, quality is defined as the degree to which a design variable or requirement has converged to the "final form." Therefore, quality is intended to be a measure of design convergence. Most activities can be characterized as definition activities, analysis activities, or decision activities. Quality for definition activities is a measure of the degree to which the outputs fully define a feasible product, and any remaining uncertainty in the definition is well understood and "small enough" to be covered by available margins. Quality for analysis activities is a measure of the confidence that the outcomes demonstrate feasibility, and any remaining uncertainty in either the definition or analysis method will not change the evaluation of feasibility. Quality for decision activities is a measure of the robustness of the decision; that is, the decision is expected to stand as the design evolves. In all cases, the evaluation of quality depends on both the quality of the input variables and the level of detail, precision, and accuracy of the definition, analysis, decision method.

Mechanisms describe the resources needed to accomplish the work associated with the activity or activity model. For activities, mechanisms include design tools, supporting data sets, design processes, duration (weeks) and effort (person-weeks by labor category). Each activity model includes a list of the activity mechanisms and produces an estimate of the total required workload and incremental accomplishment of that total required workload during a given week. The required workload (person-weeks) is calculated based on the capability of a senior engineer and ideal condition of high-quality inputs. The required workload is adjusted by factors to account for the size of the task and properties of the design implementation. Since the quality of the output is a function of both the quality of the inputs and progress towards completing the required workload, expending more effort beyond the workload required under ideal conditions may be necessary to achieve a desirable level of quality. Also, the effective person-weeks (work) applied to the workload is calculated as a function of the number and experience level of engineers, the relative effectiveness of each engineer based on experience level, and the loss or gain in efficiency due to working as a team rather than as an individual. This project assumes all design data, requirements, mechanisms, and controls are managed within a PDE. When a design activity receives its start trigger, the design activity performs the following:

1. Retrieves its input design data and controls from the PDE.
2. Applies its mechanisms to produce its output design data (based on input design data and controls).
3. Stores its output design data in the PDE.
4. Sends a completion trigger (typically to the control element) to indicate it has completed.

Design activity models perform the following when a start trigger is received:

1. At the start of each week between the start week and end week, retrieve input design data quality and control data (calculated as of the end of the previous week) from the PDE.

2. Produce estimates for total workload required and the fraction of the total workload accomplished at the end of each week between the start week and end week.
3. Calculate the quality of the output design data at the end of each week between the start week and end week.
4. At the end of each week, store the quality of output design data, control data, and fraction of the total workload accomplished in the PDE.
5. At the end of each week, send a completion trigger to the control element.

Generally, the analysis activities are applied to the results of multiple definition activities or are conducted by a different organization from that which performed a definition activity. A definition activity can include limited analysis, if the same organization performs both the definition and the analysis, and the analysis does not require the results of other definition activities.

Control Element

A control element performs the following:

1. Determines the sequence of executing activities and activity models. This sequence can be either fixed or adaptive. An adaptive control element examines the quality of design variables produced by activity models to guide the selection of the next design activity models to execute.
2. Develops the control data for each execution of the activity models, possibly based in part on the quality of requirements and design variables.
3. Sends start triggers to activities to initiate design activity models in accordance with the sequence from (1).
4. Responds to completion triggers as follows:
    a. Based on quality reported by the design activity models, determine if the design has converged.
    b. Based on schedule and effort reported by the design activity models, create a resource loaded Gantt Chart. Produce a cost estimate and overall schedules.
5. If a fixed sequence control element is employed, iterates steps (2) through (4) until the sequence is complete. Note if the design has converged at the completion of the sequence.
6. If an adaptive control element is employed, iterates steps (1) through (4) until the design has converged.

This control element reflects the design approach that the SDM will take to manage the project. This design approach will have a great influence on the total workload, rate of convergence, and total schedule needed to accomplish the design. One of the significant benefits of the activity-based design process model is that it enables the SDM to trade off different design approaches through simulation.

Model Description Documents

The design activity model does not depend on the selection of the modeling environment used to implement it. To facilitate implementation into an arbitrary modeling environment, the activity models and control element are defined in Model Description Documents (MDDs). The MDDs provides enough detail to enable the creation of activity models and control elements in whatever modeling environment is chosen. If the modeling environment evolves to use different modeling tools, then the MDDs will enable an easier migration to the new modeling tools.

For this project, the MDDs are initially configuration managed in Microsoft Word documents using a standard template. A future objective is to manage the MDDs within a Model Based Systems Engineering (MBSE) environment, such as the SysML® based Cameo®.

Standard Statement of Work

The value a standard statement of work provides is in ensuring that organizations performing the design activities clearly understand the expected scope of the work along with the associated cost and schedule.

Since the MDDs describe in some detail the work that should be accomplished within each activity, and each activity is performed by a single organization, the MDDs provide most, if not all, the details needed to define the scope for a standard statement of work. The results of exercising the activity-based design process model are the major source of the cost and schedule for the design activity.

Estimating Relationship for Project

As detailed by Valerdi (2005) and Fortune (2009), the Constructive Systems Engineering Cost Model (COSYSMO) estimates the time and effort associated with performing systems engineering tasks. While COSYSMO does not differentiate the activities making up a systems engineering effort, its structure for performing the estimate can be applied to estimating the duration and effort of design activities. The equation used to estimate performance is given by Fortune (2009):

Equation 1 $$PM_{NS} = A \cdot \left(\sum_k (w_{e,k}\Phi_{e,k} + w_{n,k}\Phi_{n,k} + w_{d,k}\Phi_{d,k}) \right)^E \cdot \prod_{j=1}^{14} EM_j$$

Where:

PM_{NS} = effort in Person Months (Nominal Schedule)

A = calibration constant derived from historical project data

k = {Requirements, Interfaces, Algorithms, Scenarios}

w_x = weight for "Easy", "Nominal", or "Difficult" size driver

Φ_x = quantity of "k" size driver

E = represents (dis)economies of scale

EM = effort multiplier for the j_{th} cost driver.

For this project, the form of the COSYSMO estimation equation is generalized as follows:

$$W_{setup\_x}(output) = A_{setup\_x}(output) \prod_{k-0}^{K-1} F_{setup\_k}$$

$$W_{recurring\ x}(output) - A_{1\ x}(output)\, f_{size}(N_0, N_1, N_2, \ldots, N_n) \prod_{k=0}^{K-1} F_k$$

where:

W_{setup_x} = setup workload for activity precision *x*
$W_{recurring\_x}$ = recurring workload for activity precision *x*
output = output variable
x = level of precision
1: imprecise
2: moderately precise
3: highly precise
A_{setup_x} = Nominal workload to setup the activity (employee-weeks)
K = number of adjustment factors
F_{setup} = adjustment factors for setup
A_{1_x} = Nominal workload to complete first item (recurring) for precision *x* (employee-weeks)
$N_0, N_1, N_2, \ldots, N_n$ = Size Measures
$f_{size}(\ )$ = Size Adjustment Function based on size metrics (e.g. learning curve)
F_k = adjustment factors for recurring

Not all values of level of precision (*x*) are required to be used. If only one level is used, then *x* = 3 should be used. Similarly, if only 2 levels are used, then *x* = 2 or *x* = 3 should be used. Generally, the levels of precision for an activity should increase over time. Some overlap is desirable such that the setup of activity workload for a higher level of precision occurs while the recurring activity workload for a lower level of precision is occurring.

The level of precision can be interpreted several different ways. For many activities, level of precision refers to the accuracy of the modeling used as part of the activity; that is, the "goodness" of the model output. A low level of precision may correspond to using parametric equations and analogy to other ships designs to estimate the outputs. A medium level of precision may incorporate physics-based modeling for a portion of the effort. A high level of precision may incorporate physics-based modeling extensively. The MDD should describe how the levels of precision are defined for each design activity.

For other activities, the level of precision refers to different phases of execution of the activity. For example, a testing activity could define a low level of precision to correspond to developing test plans, a medium level of precision to correspond to developing test procedures, and a high level of precision to conducting the tests and developing the test reports.

Because the level of precision impacts the evaluation of quality, precision should represent levels of value to other activities that use the output of the particular activity as an input.

Nominal Workloads are based on completion of an activity by a senior engineer at the specified level of precision and with all inputs of high quality. For activities with multiple outputs, a separate set of Nominal Workloads are defined for each output. Triggers may be defined to incrementally work on individual outputs. In the case of multiple outputs, the labor should be apportioned to work towards each output based on the trigger.

The setup workload is that work that is not directly valuable to other activities. This includes gathering of data, developing and integrating models, and other such work. The recurring workload provides direct benefit to other activities.

The adjustment factors (F_k and F_{setup_k}) are typically specified discretely based on an objective or subjective measure. The "normal" metric is assigned an adjustment factor of 1.0 and the other metrics are used to adjust "normal." Examples of adjustment factors include:

- Input novelty (degree to which available data applies) {Less than normal; Normal; More than normal}
- Personnel Experience {Inexperienced; Normal; Very Experienced}
- Tool Support {stand-alone tools -little data; stand-alone tools - moderate missing data; Normal – stand alone, missing data; integrated with IDE, some missing data; integrated with IDE – no missing data}
- Process Capability {perhaps CMMI}

The specific adjustment factors, nominal workloads, size adjustment function, and size measures are detailed in the MDDs.

The effective labor (L_1) that is applied during one week is a function of the total number of employees assigned and the seniority of the employees:

$$L_1 = \text{n\_senior} + L_{\text{factor}}\,(\text{n\_junior})$$

A junior engineer is assumed to be L_{factor} times as effective as a senior engineer. The default is 0.81 which assumes that a senior engineer has performed this task at least twice with a learning curve of 0.9. The specific expertise of the senior and junior engineers is specified as a labor category in the MDDs. Table 3 provides a list of the labor categories.

Table 3. Labor Categories

ID	Code	Weekly Rate	Description
0	eng_gen	$ 8,000.00	General Engineer
1	arch_sr	$ 8,000.00	Senior Power System Architect
2	arch_jr	$ 6,000.00	Junior Power System Architect
3	epla_sr	$ 8,000.00	Senior EPLA Engineer
4	epla_jr	$ 6,000.00	Junior EPLA Engineer
5	powr_sr	$ 8,000.00	Senior Power System Engineer
6	powr_jr	$ 6,000.00	Junior Power System Engineer
7	psim_sr	$ 8,000.00	Senior Power System Simulation Engineer
8	psim_jr	$ 6,000.00	Junior Power System Simulation Engineer
9	pwr_sim	$ 8,000.00	Power and Propulsion System Integration Manager
10	prop_sr	$ 8,000.00	Senior Propulsion System Engineer
11	prop_jr	$ 6,000.00	Junior Propulstion System Engineer
12	surv_sr	$ 8,000.00	Senior Survivabilty Engineer
13	surv_jr	$ 6,000.00	Junior Survivabiltiy Engineer
14	cost_sr	$ 8,000.00	Senior Cost Engineer
15	cost_jr	$ 6,000.00	Junior Cost Engineer
16	test_sr	$ 8,000.00	Senior Test Engineer
17	test_jr	$ 6,000.00	Junior Test Engineer
18	rma_sr	$ 8,000.00	Senior Reliability Engineer
19	rma_jr	$ 6,000.00	Junior Reliability Engineer
20	safe_sr	$ 8,000.00	Senior Safety Engineer
21	safe_jr	$ 6,000.00	Junior Safety Engineer
22	log_sr	$ 8,000.00	Senior Logistician
23	log_jr	$ 6,000.00	Junior Logistician
24	ctrl_sr	$ 8,000.00	Senior Control Engineer
25	ctrl_jr	$ 6,000.00	Junior Control Engineer

The effective work accomplished by a group of equally capable individuals is different than for a single individual. The effective work accomplished by a single individual in a group (due to inefficiency of working in a group) is given by:

For $n_{FTE} > 1$

$$f_{FTE}(n_{FTE}) = \frac{B_0}{n_{FTE}} + B_1 + B_2 n_{FTE} + B_3 n_{FTE}^2$$

For $n_{FTE} \leq 1$

$$f_{FTE}(n_{FTE}) = 1.0$$

Where:

n_{FTE} = n_senior + n_junior

Data from Mao et al., (2016) suggests the following values for the coefficients:

$$B_0 = 0.3631$$
$$B_1 = 0.6369$$
$$B_2 = 0.0$$
$$B_3 = 0.0$$

Data from this study indicates that the B_2 and B_3 terms are not needed. Since $f_{FTE}(n_{fte})$ is the effectiveness of a single worker, the effectiveness of the team of equally capable individuals is $f_{FTE}(n_{fte})$ multiplied by the team size. This results in B_0 being a constant term and B_1 a linear term for the team effectiveness. Other studies, without sufficient supporting data to establish values for the coefficients, state that small teams can have performance better than the linear relationship, and large teams can have performance worse than the linear relationship. If sufficient and more applicable data become available, the B_2 and B_3 terms could be useful.

The effective work accomplished by the team composed of both senior and junior engineers during week (W_{week}) for a given output is given by:

$$W_{week}(output) = L_1 f_{FTE}(n_{FTE}) t_f(output)$$

where $t_f(output)$ is the fraction of the work assigned to *output*. The values of $t_f(output)$ are constrained by

$$\sum_{i=0}^{i<nbr_{outputs}} t_f(i) = 1.0$$

$$0 \leq t_f(i) \leq 1.0$$

Note that $t_f(output)$ may be a function of the trigger.

If $w_{setup\_x} < 1$ (i.e., setup is not complete), then the work is applied to setup

$$w_{setup\_inc}(output) = \frac{W_{week}(output)}{W_{setup\_x}(output)}$$

If $w_{setup\_x} + w_{setup\_inc} > 1$, then the setup has been completed and some of the effective work can be applied to recurring work. The effective work to be applied to recurring work is given by:

$$W_{week}(output) = \left(W_{setup\_x}(output)\right)\left(w_{setup\_x}(output) + w_{setup\_inc}(output) - 1\right)$$

And

$$w_{setup\_x}(output) = 1$$

Otherwise

$$w_{setup\_x}(output) = w_{setup\_x}(output) + w_{setup\_inc}(output)$$

Similarly, if $w_{setup\_x} = 1$, then the work is applied to recurring work

$$w_{recurring\_inc}(output) = \frac{W_{week}(output)}{W_{recurring\_x}(output)}$$

$$w_{recurring\_x}(output) = min(w_{recurring\_x}(output) + w_{recurring\_inc}(output), 1)$$

Even if $w_{recurring\_x}(output) = 1$ at the beginning of a week, there may be benefits to continued execution of the activity to improve the output quality based on the input qualities improving. Output quality does not depend on $w_{setup\_x}(output)$. For $w_{recurring\_3}(output) = 1$ at the beginning of the week, the maximum increase in quality is given by

$$q_{inc\_max}(output) = 0.5 * w_{recurring\_inc}(output)$$

Estimating Relationship for Quality

For this model, Quality is represented by a real number q_o between 0.0 and 5.0. The quality of an output variable is a function of the level of precision and completion of recurring work of the activity model, the quality of the input variables (both design variables and requirements) for input variables of high impact (q_{hi_n}), and for input variables of low impact (q_{low_m}). The MDDs define the input variables, and whether they are low or high impact. The discrimination between low and high impact is somewhat arbitrary; the quality of low impact input variables should have less of an impact on output quality than high impact input variables. The quality is calculated on a maximum of Q_MAX_LOW (default 4) of the lowest of the low impact quality inputs, and a maximum of Q_MAX_HIGH (default 8) of the lowest of the high impact quality inputs. The number of low impact quality inputs included is M and the number of high impact inputs included is N.

Limiting the number of quality values considered in calculations magnifies the impact of a single low quality input as compared to the average of all inputs. With this formulation, adding one or more high quality inputs beyond the maximum number considered will not change the assessment of quality.

We define the following fractions of recurring workload ($w_{recurring\_x}$) completed at any given week:

$w_{recurring\_1}$ fraction of recurring workload completed at imprecise level [0 1]

$w_{recurring\_2}$ fraction of recurring workload completed at moderately precise level [0 1]

$w_{recurring\_3}$ fraction of recurring workload completed at highly precise level. [0 1]

While the following equations for quality are arbitrary and represent imprecise measures, absent any demonstrably better equations, they have proven to be useful for project purpose.

Product does not exist (All $w_{recurring\_x} = 0$):

$$q_O = 0$$

If product based on an imprecise level of precision:

$$q_O = w_{recurring\_1} + \frac{.5}{1.5}\left(\frac{1}{N}\right)\sum_{n=1}^{N} min(q_{hi\_n}, 1.5)$$

Note that the highest quality with a level of precision of imprecise is 1.5.

If the level of precision is moderate or higher, but one or more high impact inputs are of quality less than 1.5, then

$$q_O = 1 + 0.5max(w_{recurring\_2}, w_{recurring\_3}) + \frac{.5}{1.5}\left(\frac{1}{N}\right)\sum_{n-1}^{N} min(q_{hi\_n}, 1.5)$$

If the level of precision is moderate or better and all high impact inputs are of quality not less than 1.5 and either at least one high impact inputs is of quality less than 2.5 or at least one low impact inputs is of quality less than 1, then

$$q_O = 1.5 + 0.5max(w_{recurring\_2}, w_{recurring\_3}) + \frac{1.0}{2.5}\left(\frac{1}{N+M}\right)\sum_{n=1}^{N} min(q_{hi\_n}, 2.5) + \frac{1.0}{1.0}\left(\frac{1}{N+M}\right)\sum_{m=1}^{M} min(q_{low\_m}, 1.0)$$

If the level of precision is moderate, and all high impact inputs are of quality not less than 2.5 and all low impact inputs are of quality not less than 1.0, then

$$q_O = 2.5 + 0.5w_{recurring\_2} + \frac{0.5}{3.5}\left(\frac{1}{N+M}\right)\sum_{n=1}^{N} min(q_{hi\_n}, 3.5) + \frac{0.5}{2.0}\left(\frac{1}{N+M}\right)\sum_{m=1}^{M} min(q_{low\_m}, 2.0)$$

Note that the highest quality with a level of precision of moderate is 3.5.

If the level of precision is high, and all high impact inputs are of quality not less than 2.5 and all low impact inputs are of quality not less than 1.0 and either at least one high impact inputs is of quality less than 3.5 or at least one low impact inputs is of quality less than 2.0, then

$$q_O = 3.0 + 0.5w_{recurring\_3} + \frac{0.5}{3.5}\left(\frac{1}{N+M}\right)\sum_{n=1}^{N} min(q_{hi\_n}, 3.5) + \frac{0.5}{2.0}\left(\frac{1}{N+M}\right)\sum_{m=1}^{M} min(q_{low\_m}, 2.0)$$

If the level of precision is high, and all high impact inputs are of quality not less than 3.5 and all low impact inputs are of quality not less than 2.0, and either at least one high impact inputs is of quality less than 4.0 or at least one low impact inputs is of quality less than 3.0, then

$$q_O = 3.5 + 0.5w_{recurring\_3} + \frac{1.0}{4.0}\left(\frac{1}{N+M}\right)\sum_{n=1}^{N} min(q_{hi\_n}, 4.0) + \frac{1.0}{3.0}\left(\frac{1}{N+M}\right)\sum_{m=1}^{M} min(q_{low\_m}, 3.0)$$

If the level of precision is high, and all high impact inputs are of quality not less than 4.0 and all low impact inputs are of quality not less than 3.0, then

$$q_O = 4.5 + 0.5w_{recurring\_3}$$

If the quality of an output variable is calculated higher for a lower level of precision, the higher quality value is used for the output variable.

If $w_{recurring\_3}$ equals 1 at the start of a week, then the maximum increase of quality during that week is given by

$$q_{inc\ max}(output)$$

Software Selection

To determine the best platform to construct the Design Activity Model in, NSWC Philadelphia Division performed an assessment of viable software packages. They selected several programs that had the requisite capabilities and were available to the Navy community. The assessed software packages included:

- Cameo: Cameo Systems Modeler (CSM) is a cross-platform Model-Based Systems Engineering (MBSE) environment. CSM is design to allow for straightforward requirements management and traceability and contains a report generation tool that allows for the output of the diagrammatic elements to a set of documents. SysML parametric diagrams and constraint blocks can be used to defined mathematical expressions related to cost, risk, quality, schedule, etc. within the CSM GUI. Cameo Simulation Toolkit plugin extends Cameo Systems Modeler capabilities and allows for validating system behavior by executing, animating, and debugging system behavior. The Paramagic plugin allows for parametric trade studies by continuously modifying target variables in SysML instance models to capture design alternatives. Currently, Cameo is being used to perform systems engineering on an existing Navy program.
- Boxarr: Boxarr (previously known as Plexus) is a Systems Engineering tool designed specifically to elicit, model, and analyze very complex and often cyclic and/or ambiguous networks of dependency that exist in design processes for highly engineered products such as ships and aircraft.

Besides its unique visualization of these networks, Boxarr outputs optimized plans that resolve real world cyclic dependencies while respecting constrained resource scenarios. These plans can be exported to Project Management Tools such as Primavera for further detail planning, and especially task allocation and execution.

- Primavera: Primavera is a project portfolio management tool that is designed to execute portfolios of projects based on critical path technology, earned value analysis and reporting. It is well regarded for project management, cost control, and resource management purposes. The navy is considering its use on new programs.
- MATLAB/Simulink: MATLAB is a proprietary programming language and numerical computation software package. Simulink is an add-on to MATLAB that utilizes graphical modeling. MATLAB does not have document modelling functionality and does not have any intrinsic capability to generate and maintain project scheduling, so would need to be combined with another tool to be a viable option. MATLAB can interface with CSM, particularly with parametric simulations through the Paramagic or Simulation Toolkit plugins.

NSWC took these four programs and rated them against four primary modeling objectives:

1. The ability to manage MDDs as a system and exchange variables between MDDs automatically
2. The ability to automatically generate standard Statements of Work (SOWs) based on MDDs or other actively managed model artifacts
3. The ability to assess cost, schedule, and quality including performing trade studies of probabilistic cost, schedule, and quality metrics for scheduling design activities
4. The ability to provide collaborative engagement between multiple stakeholders

The results of the ranking are shown below in Fig. 5. Green indicates that the objective is fully satisfied, yellow indicates it is partially satisfied, and red indicates that the objective is not met.

Boxarr ranked highly in 3 of 4 objectives, but it is not a program that is commonly used within the surface ship design community. This would have made it challenging to integrate a Boxarr model with existing surface ship programs in the next phase of the project, therefore it was ruled out of consideration. Primavera was ruled out because it lacked the capability to manage the documents the way the team required.

Objective	Objective Description	Software
1	Manage Model Description Documents (MDDs) as system of MDDs to exchange variables between them automatically	🟩 CSM
		🟩 Boxarr
		🟥 Primavera
		🟨 MATLAB
2	Automatically generate standard Statements of Work (SOWs) based on MDD or other actively managed artifacts	🟨 CSM
		🟥 Boxarr
		🟥 Primavera
		🟨 MATLAB
3	Assess cost, schedule and quality (perform trade studies of probablistic cost, schedule and quality metrics for scheduling design activities)	🟨 CSM
		🟩 Boxarr
		🟨 Primavera
		🟨 MATLAB
4	Collaborative engagement between multiple stakeholders	🟩 CSM
		🟩 Boxarr
		🟩 Primavera
		🟨 MATLAB

Fig. 5, Software Performance Raking

The team selected a combination of CSM and MATLAB. Initially, CSM was assessed to be able to perform all the modeling objectives and the Navy is already utilizing it on some systems engineering problems. The ability to interface with current design efforts is beneficial – schedule and cost data generated by the Design Activity Model could be leveraged by the systems engineering management, especially in early phases of program management. Once modeling began, the team realized that the calculation features in CSM were not adequate and therefore decided to use MATLAB embedded in the CSM blocks to run the calculations.

MODEL DESCRIPTION

The model was constructed using MagicDraw 19.0 and utilizes its MATLAB integration capability as well as the Cameo Simulation Toolkit plug-in. The structure of the model can be seen completely in the Block Definition Diagrams (BDDs) nested under the "Domains" package in the containment tree. In the "Design Activity Model Domain" BDD, there are several blocks categorized as control element blocks. These blocks include the "banks" (Requirement and Product Quality, Workforce Metrics, Quality-In-Metrics, etc.) that store information as well as the "Product and Requirement Quality Input Distribution" block from which the most current product and requirement qualities are distributed to be used as inputs to the design activities in each iteration (week) of the simulation.

The remaining blocks in this BDD are categorized as the product blocks. They represent each individual design activity to be simulated and contain information derived from the Model Description Documents (MDDs) such as the level of precision, inputs to the activity (both high and low impact), adjustment factors, the number of engineers assigned to the activity, and other necessary values and parameters.

The single control element block that is the composition of all other blocks is aptly named the "DAM Control Element." It is through this block that all the information in the model can be accessed since it encompasses all other blocks and, therefore, it is this block that is the target of the simulation. Also significant is that the “DAM Control Element” block tracks the number of iterations of the simulation, and it contains most of the necessary Internal Block Diagrams (IBDs) wherein all other blocks are connected respectively via proxy ports to allow for the passage of signals and information.

The signals and information that are shared between blocks through these connected proxy ports are activated, sent, or controlled with activity diagrams. What activity diagrams are executed, and when, lies in the control of state machine diagrams that are nested respectively under most blocks in the containment tree. When the simulation is initialized, the first state of all state machine diagrams (with few exceptions) is a state called “Inactive” where the diagram then waits for a signal to continue. These states exist simply so that the model may be executed in a systematic way rather than all at once. The chief state machine diagram, nested under the “DAM Control Element” block, receives an initial signal directly from the user to begin the simulation. From that point, the user has the option to call a previously constructed and user-created activity diagram containing a list of activities to evaluate automatically by the tool for a set number of iterations, or they may simulate the activities manually, one at a time. Should the user desire to change the value of various parameters during simulation they need only hit the “pause” button in MagicDraw’s simulation dashboard, change the parameters under the appropriate block in the variable pane, and then hit the button again. The model will then continue the simulation using the new parameters.

When the model receives a signal to execute any given design activity (in isolation or as part of a sequence), the state of that product block transitions from the “Inactive” state to the “Call Quality Calculation” state. In this state, first, an activity diagram is executed that draws from the control blocks the current values of the requirement qualities and other product qualities that are used as inputs for the design activity being executed; according to the MDDs. Once complete, a second activity diagram is executed that automatically assigns an appropriate number of engineers to the activity to achieve a certain value threshold (increase in recurring work, increase in setup work, and maximum increase in quality) based on the level of precision and the fraction of work for that specific evaluation. These value thresholds are editable by the user as well. After that a third activity diagram then calculates necessary values beginning with setup work, then recurring work, and finally the quality of the product for any given week. These three key values (setup work, recurring work, and quality), among other values, are calculated using MATLAB functions that are dragged into the activity diagram and appear in the form of call behavior actions. So long as the inputs to the MATLAB function are defined either directly by input pins on the call behavior action itself or indirectly by existing as a defined value in the product block calling the activity diagram, the MATLAB function will be satisfied and will execute as any other “Opaque Expression.” The output values of these MATLAB functions are then communicated via output pins to another opaque expression where they are renamed and ultimately stored in the control blocks to be used in the next iteration of the model.

Following this third activity diagram execution in the “Call Quality Calculation” state, the product block’s state machine diagram transitions to a final state in which the cost of labor is calculated. This activity diagram is constructed such that it considers the type and number of engineers assigned to the activity for any given week and creates and maintains a compounding total labor cost based on a set table of labor categories. This activity-specific total is then added to the total labor costs of the other activities to create a single total cost for the design. Finally, the state machine diagram transitions back to the “Inactive” state waiting to be executed in the next iteration. Should the model be evaluating activities automatically, upon transitioning back to the “Inactive” state, a signal will be sent to the “DAM Control Element” to indicate that this specific design activity has finished. The sequence of events in this product block’s state machine diagram is common to all product blocks. When all the activities that the user has placed in the list of activities to be evaluated have returned their “completion” signals to the “DAM Control Element” block, the chief state machine diagram can transition the next state where requirement qualities are increased appropriately.

Using MATLAB in conjunction with MagicDraw in this fashion allows for, what would have been, extremely complex and lengthy opaque expressions or countless activity diagram actions to be contained in much more compact diagrams. Additionally, by importing MATLAB functions as call behavior actions, much of the logic pertaining to how the model deals with any combinations of product block parameters can be resolved in MagicDraw instead of MATLAB itself which would likely have resulted in a single, exceptionally complicated MATLAB function.

VERIFICATION AND VALIDATION EFFORT

Verification, Validation, and Accreditation (VV&A) is a priority for any model used in support of naval ship design programs.

The verification activities used for this effort provided confirmation that the model correctly matched the MDDs. This included functions that predict labor and duration and the dependency relationships that represent the integrated process. Additionally, metrics have been developed and included in the model to ensure that the software functions as planned and achieves the desired capability.

The validation activities are in progress and will rely on judgment that the results are logical, reasonable, and consistent with expectations. These subjective criteria are appropriate at this stage, since it is unlikely that traditional processes can provide match runs for the subset of machinery design activities. However, the general magnitude of results and the trends across the design space will be compared with expectations from a use case to confirm that the predictions are reasonable and that changes in behavior relative to changes in input parameters are reasonable. Additional validation approaches are targeted for future work.

Notional Design POA&M

The use case developed to verify the model is a notional schedule and manning plan for Power and Electric System with Energy Storage Capability developed by Gibbs & Cox, drawing from prior experience designing large ships with integrated power and electric systems. The plan covered the Preliminary and Contract Design phases. Preliminary Design covers five years, and Contract Design covers six. Each design task in the plan maps to either requirements or design activities in the model. They can encompass one single item, as in the case of something like the Reliability Analysis Report, or all items such as the Navy Review tasks. Resources corresponding to the resource types laid out in Table 3 are assigned by number of FTEs and seniority level. The activities are then scheduled in the Gantt chart based on the assumed design progression. Three sets of results were produced based on the notional POA&M, examining the capabilities and limits of the model.

First, the model was run reproducing the notional POA&M exactly as developed. The order of activities and the number of resources assigned are treated as inputs, and the quality is monitored over the timeline given in the Gantt chart. The results were then examined, and the schedule was updated based on the lessons learned. The model was then re-run following those updated parameters and cost and schedule impacts were examined.

Once the two initial cases were run against the notional POA&M, a series of test cases were developed with the requirements assumed to be at the full quality of 5.0 at the beginning of the analysis. The initial run held requirements at 5.0 and utilized the manpower assignments from the initial POA&M. The model was then run until product qualities were all at or above 4.0. Lessons learned from this test case were then used to develop an iterative schedule to more efficiently progress the product quality based on both cost and schedule.

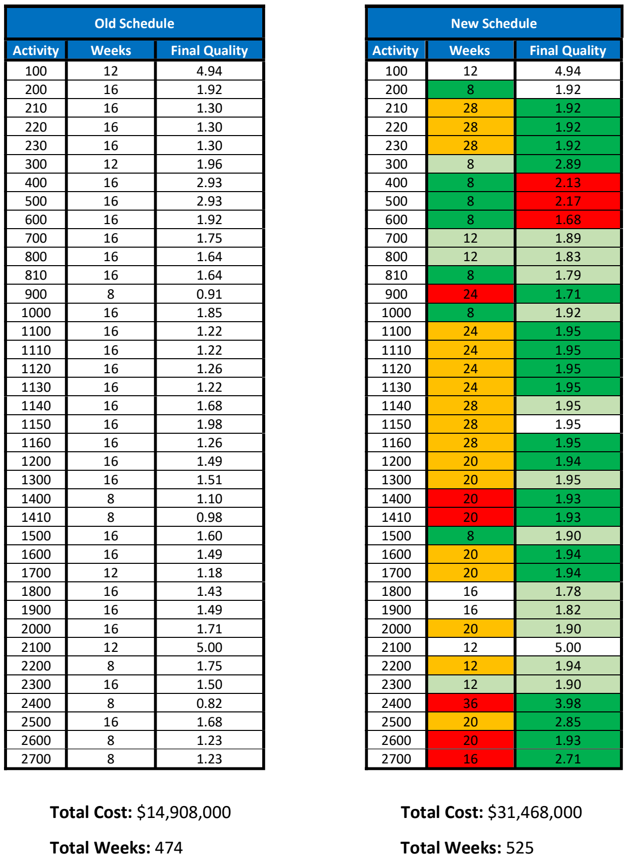
Finally, the model was run with a more realistic assumption of requirement quality. High level requirements were assumed to be at nearly full quality, while requirements that came from other aspects of the ship design were assumed to be increased gradually over time. Lessons learned from this initial run were again applied to the manning and sequencing of activities to produce a final, optimized schedule.

Results and Lessons Learned

The results of the first V&V effort showed that the model yielded all the information necessary for the user to interrogate the efficiency of their ship design plan. The result output is show in Fig. 6.

The initial run is on the left of the figure. The model output could easily be condensed by activity to show duration of each design activity, final quality, and project total duration and cost. The figure on the right shows the results of a second run, using lessons learned from the first run to create a resourcing plan that resulted in significant quality increases. The color code is as follows:

- Quality
  - DARK GREEN: quality increase of 0.5+
  - LIGHT GREEN: quality increase < 0.5
  - RED: quality decrease
- Schedule
  - DARK GREEN: 50% or greater improvement
  - LIGHT GREEN: Improvement < 50%
  - ORANGE: Negative impact < 50%
  - RED: Negative impact > 50%



Old Schedule

Activity	Weeks	Final Quality
100	12	4.94
200	16	1.92
210	16	1.30
220	16	1.30
230	16	1.30
300	12	1.96
400	16	2.93
500	16	2.93
600	16	1.92
700	16	1.75
800	16	1.64
810	16	1.64
900	8	0.91
1000	16	1.85
1100	16	1.22
1110	16	1.22
1120	16	1.26
1130	16	1.22
1140	16	1.68
1150	16	1.98
1160	16	1.26
1200	16	1.49
1300	16	1.51
1400	8	1.10
1410	8	0.98
1500	16	1.60
1600	16	1.49
1700	12	1.18
1800	16	1.43
1900	16	1.49
2000	16	1.71
2100	12	5.00
2200	8	1.75
2300	16	1.50
2400	8	0.82
2500	16	1.68
2600	8	1.23
2700	8	1.23

Total Cost: $14,908,000

Total Weeks: 474

New Schedule

Activity	Weeks	Final Quality
100	12	4.94
200	8	1.92
210	28	1.92
220	28	1.92
230	28	1.92
300	8	2.89
400	8	2.13
500	8	2.17
600	8	1.68
700	12	1.89
800	12	1.83
810	8	1.79
900	24	1.71
1000	8	1.92
1100	24	1.95
1110	24	1.95
1120	24	1.95
1130	24	1.95
1140	28	1.95
1150	28	1.95
1160	28	1.95
1200	20	1.94
1300	20	1.95
1400	20	1.93
1410	20	1.93
1500	8	1.90
1600	20	1.94
1700	20	1.94
1800	16	1.78
1900	16	1.82
2000	20	1.90
2100	12	5.00
2200	12	1.94
2300	12	1.90
2400	36	3.98
2500	20	2.85
2600	20	1.93
2700	16	2.71

Total Cost: $31,468,000

Total Weeks: 525

Fig. 6, Initial V&V Results Comparison

Overall, the output provided from the model was enough to allow the team to make quick changes that produced quality gains and compare the impacts to cost and schedule of those two attempts. Based on this finding, no additional parameters or output was added to the model and the team proceeded with the V&V effort.

The results of the second two V&V efforts are currently in development.

FUTURE WORK

The current work focused on the design process and model implementation for naval combatant integrated power and energy system design. Future work includes exercising the design process and model on a relevant program of interest for the following purposes: (1) Confirm the reasonableness of model output when compared to data from traditional design processes, (2) Identify the need for additional model functionality, and (3) Determine the need for new design process activities.

Expansion of the design activity model to cover the overall ship design process more completely, that is, beyond the power and propulsion system domain, or to apply the model to completely different activities that can benefit from prediction of labor and duration with reasonable fidelity are opportunities for future work.

Model validation can be approached by exercising the model alongside traditional processes to determine model performance based on traditional process output. Additionally, the traditional process data can be input back into the model to correlate model output against traditional process results.

CONCLUSIONS

The Design Activity Model provides an expedient and robust ability for SDMs to interrogate their design process and assumptions, examining a program's IMS and staffing to determine whether efficiencies could be gained. Additionally, it grants SDMs the ability to narrow down on what design activities are driving cost and schedule within a constrained funding or staffing environment, allowing the team to focus their resources on the most critical problems first.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the contribution by Naval Sea System Command's Program Executive Office – Ships, particularly the Electric Ships Office (PMS 320) and the Guided Missile Destroyer (DDG(X)) Office (PMS 460). Without their support this work would not have been possible.

REFERENCES

Ballou, L.D.,(1966) *Production Control in Shipbuilding*, Naval Engineers Journal, Volume 78, Number 5, October 1966, pp. 833-838.

Browning, Tyson R. and Steven D. Eppinger, *Modeling Impacts of Process Architecture on Cost and Schedule Risk in Product Development*, IEEE Transactions on Engineering Management, Vol 49, No. 4, November 2002.

Cooper, Seth, Gene Allen, Robert Smith, Dan Billingsley, and David Helgerson, *Ship Design Process Modeling: Capturing a Highly Complex Process*, 13th International Dependency and Structure Modelling Conference, DSM'11, Cambridge, MA, USA, Sept 14-15, 2011.

Doerry, Norbert (2009) *Using the Design Structure Matrix to Plan Complex Design Projects*, Presented at ASNE Intelligent Ships Symposium 2009, Philadelphia, PA, May 20-21, 2009 http://doerry.org/norbert/papers/DSMandComplexity-final.pdf

Eppinger, Steven D., and Tyson R. Browning, (2012) *Design Structure Matrix Methods and Applications*, The MIT Press, Cambridge, Massachusetts and London, England, ISBN 978-0-262-01752-7

Evans, J. Harvey, *Basic Design Concepts*, Journal of the American Society of Naval Engineers, Volume 71, Number 4, November 1959, pp. 671-678. https://doi.org/10.1111/j.1559-3584.1959.tb01836.x

Fortune, Jarod, *Estimating Systems Engineering Reuse with the Constructive Systems Engineering Cost Model (COSYSMO 2.0)*, PhD Dissertation, Industrial and Systems Engineering, University of Southern California, December 2009.

Helgerson, D.A., D. W. Billingsley, and N. H. Doerry, *Initial DSM Application to U.S. Navy Ship Design*, Proceedings of 11th International Design Structure Matrix Conference, DSM'09, Oct 12-13, 2009.

Honsinger, Vernon C. (1968) *Pert / CPM Management for Tough-Minded Shipyard Managers*, Naval Engineers Journal, Volume 80, Number 4, August 1968, pp. 637-641.

Keane, Robert G. Jr., John McIntire, Howard Fireman, and Daniel J. Maher, (2009) *The LPD 17 Ship Design: Leading a Sea Change Toward Collaborative Product Development*, Naval Engineers Journal, Volume 121, Number 2, June 2009 pp. 15-61.

Lukas, Michael, Thomas Gärtner, Norbert Rohlender, and Christopher M. Schlick,, *A Simulation Model to Predict Impacts of Alterations in Development Process*, 9th International Design Structure Matrix Conference, DSM'07, Munich, Germany, 16-18 October 2007.

Mao A, Mason W, Suri S, Watts DJ *An Experimental Study of Team Size and Performance on a Complex Task*, 2016,

Singer, David J., Norbert Doerry, and Michael E. Buckley, *What is Set-Based Design?* Presented at ASNE DAY 2009, National Harbor, MD., April 8-9, 2009. Also published in ASNE Naval Engineers Journal, 2009 Vol 121 No 4, pp. 31-43.

Singer, David, Jason Strickland, Norbert Doerry, Thomas McKenney, and Cliff Whitcomb, *Set-Based Design*, SNAME T&R 7-12, 2017.

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

Valerdi, Ricardo, *The Constructive Systems Engineering Cost Model (COSYSMO)*, PhD Dissertation, Industrial and Systems Engineering, University of Southern California, August 2005.