SYM-AM-17-066

Proceedings of the Fourteenth Annual Acquisition Research Symposium

Wednesday Sessions Volume I

Acquisition Research: Creating Synergy for Informed Change

April 26–27, 2017

Published March 31, 2017

Prepared for the Naval Postgraduate School, Monterey, CA 93943.

Acquisition Research Program
Graduate School of Business & Public Policy
Naval Postgraduate School

Optimal Selection of Organizational Structuring for Complex Systems Development and Acquisitions

Alexandra Dukes—is a Graduate Student in the School of Aeronautics and Astronautics at Purdue University. She is currently conducting research in the Center for Integrated Systems in Aerospace (CISA) led by Dr. Daniel DeLaurentis. [dukes@purdue.edu]

Scott Parrigon—is a Graduate Student in the Department of Psychological Sciences at Purdue University. He is currently part of Dr. Sang Eun Woo's research group. [spariggo@purdue.edu]

Navindran Davendralingam—is a Research Scientist in the School of Aeronautics and Astronautics at Purdue University. He is currently conducting research in the Center for Integrated Systems in Aerospace (CISA) led by Dr. Daniel DeLaurentis. [davendra@purdue.edu]

Sang Eun Woo—is an Associate Professor of Industrial and Organizational Psychology in the Department of Psychological Sciences at Purdue University. Her research focuses on how people's personality and motivation can help explain various psychological phenomena in the workplace. [sewoo@purdue.edu]

Daniel DeLaurentis—is a Professor in the School of Aeronautics and Astronautics, Purdue University. His research and teaching interests focus on design and optimization of aerospace vehicles and systems of systems. [ddelaure@purdue.edu]

Abstract

Research suggests that product designs tend to reflect the structure of the organization in which they are conceived (i.e., Conway's Law). Prior works on this topic, especially in the context of acquisitions, have been largely descriptive without prescribing tangible ways to reduce the inefficiencies resulting from possible misalignments between a product's structure and the structure of the organization that builds the product. We present a mathematical modeling framework that enables the optimal selection of an organization's structure (here, the different ways that various types of program managers are allocated) and its product structure (here, a modular, complex system structure). We leverage quantitative and qualitative methods from areas of organizational sciences, systems engineering, and operations research in a unified manner. We demonstrate application to a defense acquisition concept problem that seeks to maximize overall performance of a complex system (the "product") being developed, while minimizing risks associated with mismatches between program manager competencies and system development ("the organizational structure").

Introduction

A product's structure is strongly affected by organizational structure, communication mechanisms, and resource channels between organizational units that work together to realize an intended product. Inefficient setup in an organization's structure often results in poor requirements being set, poor understanding of interfaces between elements of the product, and ultimately, a poor return on investment due to a consequently subpar product being realized. Prior research conducted in software engineering analyzes this relationship and *concludes that product designs tend to reflect the structure of an organization in which they are conceived*, also known as Conway's Law (Conway, 1968). Work by Ulrich (1995) and Sinha (2012) explored the question of how the degree of a new product's novelty affects the structure of an organization. In more recent literature, Honda performed a comparison of information passing strategies in system-level modeling and found that the structure of information coordination, for the case of an example satellite design problem, directly impacts the drive towards an optimal design configuration (Honda, Ciucci, Lewis, & Yang, 2015). A recent article, published in *Harvard Business Review*, presents a case study of how Juniper networks, a company that provides IT routing and network solutions, utilized

HR strategies to improve business processes across its complex organizational structure (Boudreau, 2015). The strategies reduced the number of decision chains involved in product development and sought to identify “clusters” of employees with the most diverse experience in promoting healthy innovation.

While these prior literatures allude to the coupled nature between a product structure and the structure of the organization that builds it, they are mainly descriptive in nature. These literatures do not provide a framework to improve decision-making processes related to the product structure (e.g., what collection of systems to acquire and connect) and to the organizational structure (e.g., how to allocate human resources such as program managers to constituent systems). Such decision-making processes have significant implications for improving the performance of the product. It is the couplings between organization structure and product architecture, in the context of acquisition, which forms the heart of our research goal.

Motivation

Our research is motivated by a need to enable better decision-making on how to objectively select systems that comprise a complex system and allocate program managers to each of these selected systems in a manner that maximizes complex system performance, while minimizing risks associated with mismatches between program manager competencies and system development. More specifically, we refer to organizational structures based on the allocation of program manager types (based on a spectrum of program management competencies) to manage each of the selected systems in the complex system. We follow Simon’s definition of a complex system as being a hierarchical collection of systems and subsystems that are interconnected to provide some desired capability (Simon, 1962). We consider multiple collaborating systems within this definition too since complex systems are typically developed within a collaborative construct of units within and/or across an organization.

Currently, there is a lack of systematic and quantitative modeling framework to assist decision-makers in forming organizational structures that best fit the desired complex systems development and vice versa (Honda et al., 2015; MacCormack, Baldwin, & Rausnak, 2012). This lack is driven in part by difficulties associated with underlying problem of simultaneously selecting a product structure and an organizational structure in an optimal fashion. From a *product* perspective, the task of maximizing a product’s (here, complex system) performance may result in a product structure that cannot be well managed, given the population and distribution of program manager types. From an organizational perspective, on the other hand, fixing the selection of an organization’s distribution of program managers will limit the types of products that can be effectively developed. Therefore, there needs to be an objective means of selecting systems in a complex system and allocating managers in a quantitative manner.

Methodology

We first define a scope for the “product” and “organizational” components of our mathematical framework. For the organizational structure, we focus on the program manager competencies and how various skillsets and variability can impact product development. On the product side, we adopt a modular perspective on the complex system architecture where the complex system consists of a hierarchical tree of constituent systems that connect via defined interfaces and standards. We illustrate our methodology in the context of defense acquisition; here, the organizational structure is reflected by the distribution of Department of Defense (DoD) program manager types, and the complex system architecture is reflected by modular systems that are yet-to-be acquired and connected to form a complex defense system.

Our research employs a cross-discipline strategy that seeks to allocate different organizational program manager types, based on program management competency ratings, to the system acquisition life cycle architecture for optimal performance through its phases. For the organizational elements of our framework, we adopt methods from organizational psychology to translate qualitative insights from literature into a quantitative assessment of program manager competency requirements and clarify how they may relate to the execution of the defense acquisition life cycle. For the complex system architecture, we adopt the mathematical modeling techniques and abstractions as used by Davendralingam (Davendralingam, Mane, & DeLaurentis, 2012) and an optimization perspective to enable objective selection of both the complex system architecture and organizational structure.

Problem Description

We seek to address the problem of how to optimally select systems, from a candidate pool of modular systems that constitute a complex system and allocate program managers to each system in a manner that maximizes overall performance of the complex system (the “product”) while minimizing risks associated with mismatches between program manager competencies and system development (“the organizational structure”). Our problem is based on a defense acquisitions and is motivated by availability of data and inputs. We first establish a model for the organizational component and a model of the complex system components of our work. The organizational model reflects the relationship of program manager competencies to defense acquisition processes that need to be executed in developing a constituent system. The product model, on the other hand, reflects how selection of different collections of constituent systems, when combined, provide a desired overarching military capability. In the following sections, we explain our modeling perspective of the organizational and product portions of our framework. We then present an optimization based approach that unifies both models within a decision-making framework. The data available for this study is derived from studies conducted on program manager competencies by Roy Wood (2010, 2014), and prior case study reports on various defense acquisition programs.

Modeling Organizational Structures (i.e., Program Manager Competency Mapping)

In modeling the organizational component of our mathematical framework, we first need to understand the context by which the organizational units (here, the program managers) perform. In the case of our defense acquisition problem, the program manager performs a series of required programmatic tasks throughout an acquisition process life cycle. The ability of the program manager to execute each of the required tasks in the life cycle, is based on a list of program manager competencies; this naturally has an impact on the end development of each system and the complex system as a whole. First, we need to identify/create a life cycle model that allows us to readily map program manager competencies. Second, we need to identify a list of program manager competencies that are relevant to our life cycle model. Lastly, we need to effectively map these program manager competencies onto the life cycle model by relating relevant subsets of these competencies to each phase of the life cycle model. In the following sections, we articulate each of the steps in the development of our organizational structure model, beginning with the identification of our life cycle model.

Life Cycle Model Identification

The first step in our organizational structures modeling process was to identify a useful model of the acquisition life cycle. For this purpose, we chose to use a swim lane process model. The decision to create a swim lane model stemmed from a qualitative analysis of life cycle models provided by the Department of Defense and the Defense Acquisition University. There are two prominent models used to describe the system acquisition life cycle of the DoD. Figure 1 is titled "Generic Acquisition Phases and Decision Points" within the literature and is presented in multiple variations throughout DoD Instruction (DoDI) 5000.02 (DoD, 2015).

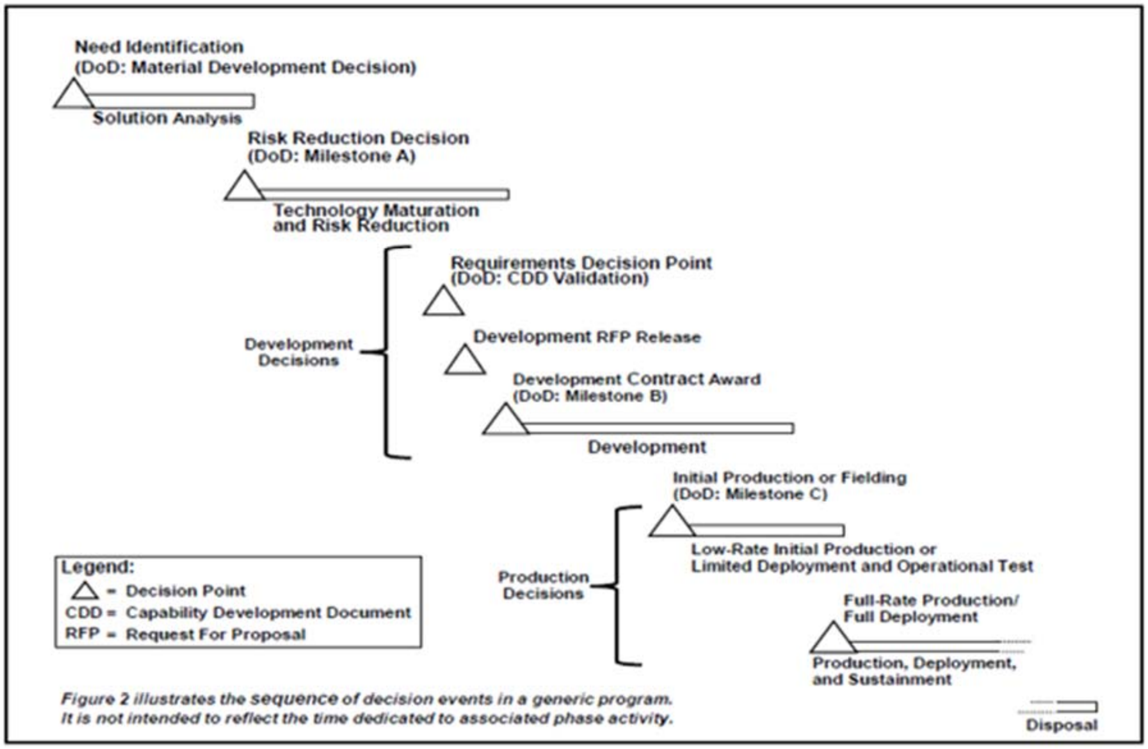

Figure 1. Generic Acquisition Phases and Decision Points
(DoD, 2015, p. 6)

For our purposes, this model does not provide enough detail to properly distinguish where the competency data would be utilized through the different phases. A significant contribution of the 5000.02 documentation is the descriptions of the phases given with Figure 1 and its ability to provide insight into the DoD program manger's role throughout each step within the life cycle. The second model, provided by the Defense Acquisition University and presented in Figure 2, provided significantly more visual detail in the processes occurring within each phase.

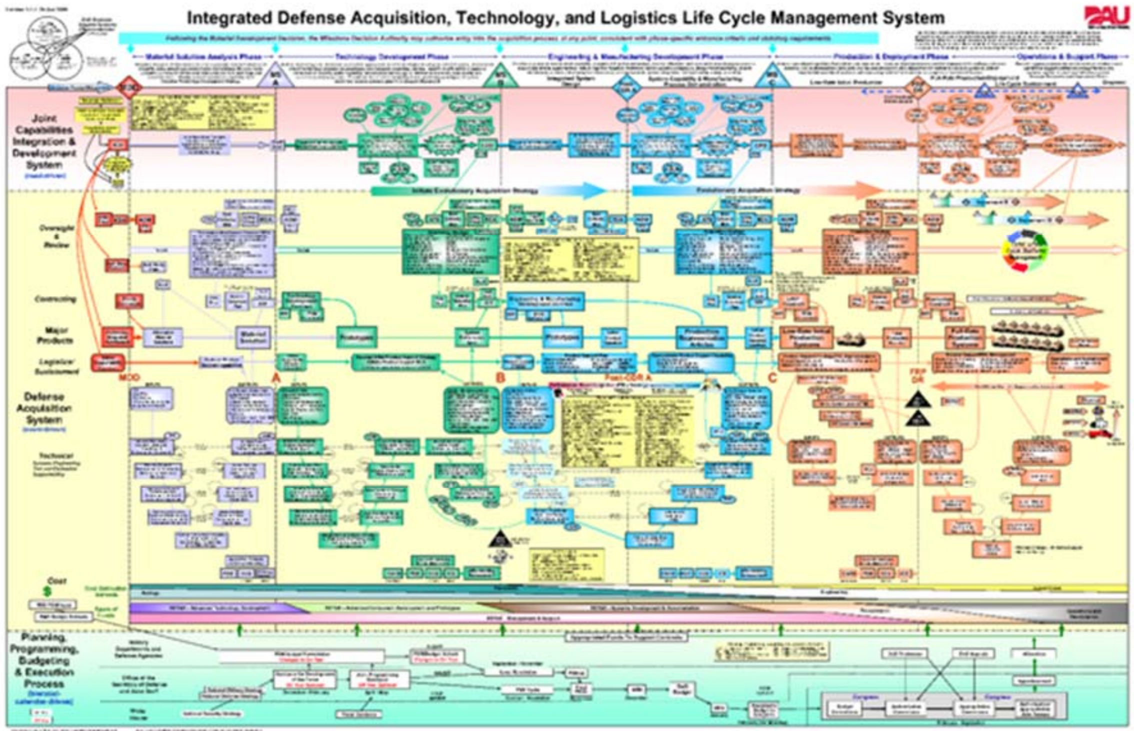

Figure 2. Integrated Defense Acquisition, Technology, and Logistics Life Cycle Management System
(Defense Acquisition University, 2009)

Due to the scope of this research, this diagram was not ideal for the time frame given to perform our analysis. Thus, we synthesized the information from both existing models forming a new model (swim lane model, Figure 3) that was executable within our given time frame.

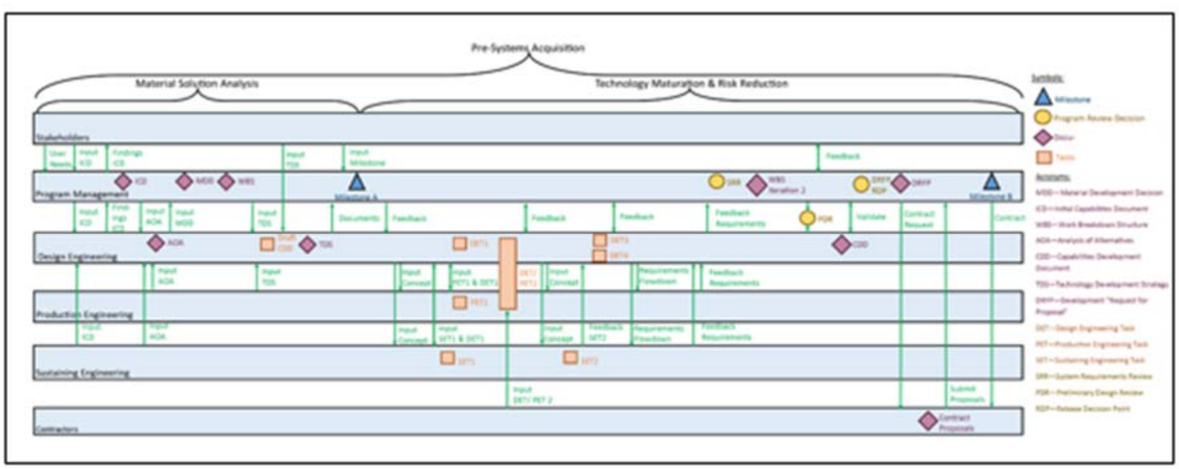

Figure 3. Swim Lane Model Depicting Processes Within the DoD System Acquisition Life Cycle

The swim lane model encompasses DoD System acquisition processes from the inception of acquisition process to Milestone B. To reduce scope for demonstration, the model was furthered reduced to processes between Milestone A and Milestone B for evaluation in the optimization problem of this paper. The swim lanes represent the tasks and interactions between the Stakeholders, Program Management, Design Engineering, Production Engineering, Sustaining Engineering, and Contractors. Each swim lane contains several actors within the DoD that were grouped within these categories based on the functions they are described to perform by DoDI 5000.02, Defense Acquisition University's Integrated Defense Life Cycle Management System visualization, and the DoD Product Support Implementation Roadmap. For example, the "Product Support Management" as stated in the DoD Integrated Product Support Implementation Roadmap diagram would fall into the "Product Management" swim lane (DoD, 2012). The elements within the swim lanes are grouped within four major categories: Milestones, Program Review Decisions, Documents, and Tasks. The Milestones, Program Review Decisions, and Documents are referenced in the instructional and GAO literature. We created the Tasks to capture steps within the life cycle that must be accomplished but are not given a formal title within the DoD literature. A description of each of the tasks are provided in Table 1.

Table 1. Swim Lane Model Task Descriptions

Task Label	Task Description
DET1	Evaluate program integration and potential risks based on Milestone A results
PET1	Evaluate potential production needs based on Milestone A results
SET1	Evaluate potential support and maintenance needs based on Milestone A results
DET/PET2	Perform competitive prototyping
SET2	Define support objectives based on competitive prototyping results
DET3	Develop system architecture
DET4	Develop technical architecture

In this study, we focus on the Technology Maturation & Risk Reduction phase. The Technology Maturation & Risk Reduction phase aims to mitigate potential risks and develop a program plan, budget, and schedule. After this phase, a contractor has been selected to pursue the program and the DoD commits its resources to the development, manufacturing, and fielding of the selected solution. The Technology Maturation & Risk Reduction was partitioned into four phases for evaluation within the optimization problem. Phase 1 begins at the conclusion of Milestone A and ends at the start of DET/PET2. Phase 2 begins at DET/PET2 and ends at the start of DET3 and DET4. Phase 3 begins at DET3 and DET4 and ends at the start of SRR. Phase 4 begins at the start of SRR and ends at the conclusion of Milestone B. The competencies addressed in Wood and the availability of qualitative data describing the program manager's role within the life cycle motivated the selection and partitioning of this phase as well as the time frame of this pilot study.

With the components of the swim lane model articulated, we can now move onto the second major phase of our organizational structure modeling—identifying the program manager competencies that can be effectively mapped onto the swim lane model. In the following sections, we articulate the competencies used, as well as the process we used to map them onto the swim lane model.

Identifying Program Manager Competencies

To map program manager competencies onto this swim-lane model, we needed to first obtain a relatively comprehensive initial list of relevant program manager competencies. For this, we utilized data collected by Wood (2010, 2014) that used a set of 35 program manager competencies indicative of the major capabilities that influenced how successful a program manager would be. Specifically, these were designed to assess the program manager competencies that "can be used in drafting project management interviewing questions, developing appraisal models to select the most qualified project managers for promotion, and designing job descriptions for project managers that can be tailored by an organization to clearly outline the roles, duties, and responsibilities of a project manager" (Golob, 2002, p. 7). These competencies were developed based upon a literature review, subject-matter expert reviews, and two surveys of program managers and the managers of program managers. A more detailed explication of these and this process can be found in Golob (2002).

These 35 competencies that resulted from this process were posited to measure 20 technical (or "hard" skills), and 15 behavioral (or "soft" skills). However, as has been posited recently in the program manager literature (Nijhuis, Vrijhoef, & Kessels, 2015) these individual program manager competencies likely are subcomponents that are attributable to more general, higher-order taxonomies of competencies from the general management/organizational psychology literatures. For example, Nijhuis et al. (2015) found that these higher-order taxonomies were effectively able to integrate the diversity of program manager competencies that had been identified in the extant literature. For example, the two soft skill competencies of project leadership (i.e., the ability to set a vision, identify the action steps, motivate others to maintain their commitment to program success and the ability to influence a team to willingly work toward predetermined program objectives) and facilitation (i.e., the ability to facilitate or guide team members through a process that helps them discover answers and overcome barriers to successful program completion) likely map onto the higher-order managerial competency of Leading and Deciding that has been well-validated within the general managerial/organizational psychology literatures (Bartram, 2005; Kurz & Bartram, 2002). Thus, while these 35 competencies are a great start, to make them practically useful for our optimization problem, as well as more theoretically parsimonious, it is important for us to map them onto these higher-order managerial competencies.

For this higher-order managerial competency mapping, we used the Great Eight model of managerial competencies (Bartram, 2005; Kurz & Bartram, 2002). These researchers defined competencies as "sets of behaviors that are instrumental in the delivery of desired results or outcomes (Bartram, Robertson, & Callinan, 2002, p. 7). The Great Eight competencies represent a parsimonious representation of the domain of managerial competencies that exist in the extant literature. The Great Eight structure has been extensively validated and refined. This refinement has created not only the broad Great Eight, but 112 component competencies that underlie the eight core dimensions. The eight core dimensions are Leading and Deciding, Supporting and Cooperating, Interacting and Presenting, Analyzing and Interpreting, Creating and Conceptualizing, Organizing and Executing, Adapting and Coping, and Enterprising and Performing.

Due to the high degree of conceptual overlap between our 35 program manager competencies and the Great Eight dimensions, we used the Great Eight as the basis for our higher-order managerial competencies. To link our 35 competencies to the Great Eight dimensions, we engaged in an iterative process of mapping the individual competencies onto the broad Great Eight. Once complete agreement of the mapping was established between all members of the research team, this mapping was finalized. With this mapping in hand, we can parsimoniously integrate these program manager competencies into our swim lane.

Deriving Baseline Great Eight Ratings From Qualitative Data

In this part, we derive a set quantitative ratings for each of Great Eight dimensions where each rating represents the degree to which each Great Eight dimension is important towards accomplishing the acquisition tasks in the swim-lane model; these ratings are considered to be baseline as they each represent an aggregate, required rating for each Great Eight dimension, based on the qualitative data from the GAO reports. To accomplish the task of generating these baseline values, it becomes necessary to properly map the program manager competencies from Wood (2010) onto the swim-lane model, through integrating the qualitative data available from the GAO reports and instructional documentation with the Wood competencies. Specifically, we utilized information regarding the tasks and competencies required at each stage of the swim-lane model to determine the importance of each competency for successful performance of the program manager at that stage in the life cycle. As articulated previously, rather than mapping each of the 35 specific competencies used within the Dr. Wood's research, we use the higher-order Great Eight dimensions that these 35 specific competencies correspond to as depicted in Table 2. This reduces our mapping from 140 ratings (i.e., 35 competencies x 4 phases) to 32 (i.e., 8 competencies x 4 phases) that is more theoretically and empirically parsimonious due to the aggregation of theoretically-redundant competencies.

Table 2. Placement of the Roy Wood Competencies to the Great Eight Dimensions

Great Eight Competencies	Roy Wood Competencies
Leading and Deciding	Document program assumptions; Implement corrective action; Project leadership; Facilitation
Supporting and Cooperating	Trustworthiness; Issue and conflict resolution; Coaching
Interacting and Presenting	Communicated program status; Negotiations; Setting and managing expectations; Communication style; Listening skills; Team building
Analyzing and Interpreting	Document program constraints; Measure program performance; Implement change control; Conduct administrative closure; Problem solving
Creating and Conceptualizing	Define program strategy; Decision making
Organizing and Executing	Determine program goals; Determine program deliverables; Quality assurance; Identify resources requirements; Develop a budget; Create a work breakdown structure (WBS); Develop a resource management plan; Establish program controls; Develop program plan; Organizational Skills
Adapting and Coping	Respond to risk; Flexibility
Enterprising and Performing	Technical ability; Sound business judgement

The process of mapping the Great Eight dimensions onto the swim-lane model was done via a systematic coding process. First, aggregated qualitative data from the GAO reports and instructional documentation were reviewed by a two-person cross-discipline team (an example of this aggregated data can be found in Table 3).

Table 3. Example of the Qualitative Aggregated Data Used to Map the Competencies to the Life Cycle Phases

Phase 3: Requirements Development Qualitative Data	
Instructional Documentation Summary	The Requirement Development effort involves tasks DET3 (develop system architecture) and DET4 (develop technical architecture), as well as, the program review decision SRR (System Requirement Review). The goal is to reduce risk and create a set of requirements which will create a baseline for the program to be presented at the PDR. The resulting requirements are additionally used in the CDD, RFP, and Milestone B.
Instructional Documentation Sources	Department of Defense, DoDI 5000.02, 2015; Department of Defense, DoD Integrated Product Support Implementation Roadmap, 2012; Defense Acquisition University, Integrated Defense Life Cycle Management System, 2004
Instructional Documentation Representative Quotes	"The point at which the major cost and performance trades have been completed and enough risk reduction has been completed to support a decision to commit to the set of requirements that will be used for preliminary design activities, development, and production (subject to reconsideration and refinement as knowledge increases)" (DoD, 2015). "Capability requirements are not expected to be static during the product life cycle. As knowledge and circumstances change, consideration of adjustments or changes may be requested by acquisition, budgeting, or requirements officials" (DoD, 2015).
GAO Reports Summary	The Requirements Development phase is hindered by the continual changing of key requirements throughout the acquisition life cycle and the lack of proper requirements development before Milestone B.
GAO Sources	GAO-08-674T, 2008; GAO-06-110, 2005; GAO-16-489T, 2016
GAO Representative Quotes	"We found four factors that have the potential to impact acquisition outcomes on individual programs: (1) unsettled requirements in acquisition programs can create significant turbulence including increased cost growth" (GAO, 2008). "Second, they (users/contractors) cannot veto new requirements. Faced with long development life cycles and promising technology advances, users often ask for new or better capabilities as a program proceeds forward. Program managers themselves are not always empowered to say "no" to demands that may overly stretch their programs, and few senior leaders above them have been willing to" (GAO, 2005). "Because DoD does not yet have approved requirements and is not planning to hold a Milestone B review, its approach for Block 4 modernization will not require the program to have such important cost, schedule, and performance reporting and oversight mechanisms in place" (GAO, 2016).

The two-person coding team consisted of one engineering graduate student with expertise in the intricacies of the program management/engineering life cycle a doctoral student in organizational psychology with expertise in leadership competencies and job performance. During the review of the aggregated GAO reports/instructional documentation, this team discussed each stage of the project life cycle, the tasks involved, how each phase fed into those which followed, and the metrics for successful performance at each phase. Once a similar frame-of-reference was created, the team discussed each of the Great Eight dimensions (considering both the general dimension, as well as the specific Roy Wood competencies underlying it) and its relevance to each phase. After the general relevance was thoroughly articulated by both members of the team, a consensus as to a numeric rating of importance (ranging from 1 to 10) for each Great Eight dimension was mapped onto each phase of the swim-lane model, for a total of 32 ratings. The team had 100%

consensus as to the final ratings. These final ratings were then used as a baseline in the development and execution of the optimization model and are presented in Table 4.

Table 4. Great Eight Mapping to Life Cycle Phases and PM Archetypes

	Acquisition Life Cycle Phase				Program Manager Archetype			
	Loop 1: Post A Concept Dev	Loop 2: Prototyping	Loop 3: Req. Dev.	Loop 4: SRR to DRFP	PM Type I	PM Type II	PM Type III	PM Type IV
Leading & Deciding	8.0	7.5	9.5	3	9	7	6	6
Supporting & Cooperating	4.5	6.5	6.5	6.5	9	6	7	3
Interacting & Presenting	9.0	9	5.5	10	7	5	4	3
Analyzing & Interpreting	2.5	5	3.5	4	5	6	3	3
Creating & Conceptualizing	2.0	8	8	6.5	5	9	9	2
Organizing & Executing	2.0	4.5	3	7.5	6	9	9	1
Adapting & Coping	2.0	4.5	2	5	3	5	5	4
Enterprising & Performing	7.0	7	8	7.5	5	5	7	3

Table 4 shows both the Great Eight Mapping assessment scores that were ascertained for each of the four studied phases of the total defense acquisition life cycle; columns 1–4 provide estimated numerical values of required level of competence, in each of the Great Eight dimensions, for the corresponding life cycle phase. Table 4 also shows a set of notional Great Eight Mapping scores for four classifications (columns 5–8) of program managers. In this example used to generate the product architecture, we *assume* that there exist four archetypes of program managers, each with a different distribution of Great Eight Mapping strengths. While the values and number of program manager archetypes in this example problem are for illustrative purposes only, we note that there are well-known quantitative methods that can be used to solicit such values in real world situations. For example, clustering algorithms such as *hierarchical clustering* can be used to quantitatively determine the number of clusters and the values of Great Eight dimensions for program managers in each cluster, given a large survey pool and survey instrument that is executed to extract relevant information.

Modeling Complex System Structures

The complex system (product) architecture portion is modelled as an interconnected set of nodes, each having a finite set of inputs and outputs. The interconnections characterize how node capabilities (outputs) feed and consequently fulfill requirements (inputs) of any connected compatible node.

Figure 4 (a) and (b) show a generalized representation of a complex system which has interdependencies between constituent systems, across multiple layers of the hierarchical structure. Each node (system) is connected to other nodes on the network, in accordance with the set of requirements needed for them to interdependently operate. The connections between nodes are also governed by a set of interaction rules. Interactions between systems are modeled as relatively simple nodal behaviors that are applicable to a wide variety of types of inter-system connections. While not exhaustive, the combinations of these nodal behaviors as modeling rules can cover a large set of real world inter-system interactions.

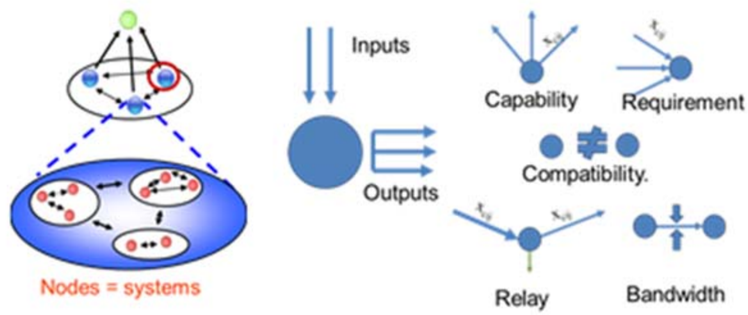

Figure 4. (a) Complex System Hierarchy (b) Nodal (System) Behaviors

Figure 4 (b) shows the five most intuitive system (node) interactions:

- Capability: systems have finite supply of capabilities that limit the number of connections they may form.
- Requirements: System requirements are fulfilled by receiving connections from other nodes that possess a capability to fulfill said requirements.
- Relay: Systems can relay capabilities between adjacent system. This can include excess input of capabilities that are used to fulfill node requirements.
- Bandwidth: Total amount of capabilities and number of connections between systems are bounded by the bandwidth of the connection linkages between systems.
- Compatibility: Systems can only connect to other systems based on a pre-established set of connection rules.

The performance of the complex system is related to the ability of the connected network of individual systems to fulfill overarching core objectives. System-wide performance is quantified by the capability of nodes that most directly contribute to the core objectives.

An Optimization Approach to Selecting Optimal Organizational and Complex System Structure

We pose the task of selecting the optimal organizational architecture and product architecture as a mathematical programming (optimization) problem involving two main segments. The first segment of an optimization problem involves an *objective function* equation that is either maximized or minimized, depending on the metric that is being used. The second segment involves a set of equations called *constraints* that reflect rules as in Figure 4 (b). A simple example of a mathematical program is the maximization of expected stock investment returns, subject to constraints on availability of funds to invest, where the decision variables are which stocks to buy, and how much to buy of each stock.

The problem of selecting an optimal complex system architecture and its organizational architecture is more specifically posed as a multi-objective optimization problem that addresses both an index that describes the level of performance for a chosen product architecture and the uncertainty in program manager performance allocated across the selected architecture. (In the simple case of the stock problem, the notion is tradeoff between expected portfolio returns and risk). The decision variables involve which systems to select in the product architecture and which program manager types to be assigned to systems that need to be developed (we explain types in a subsequent section).

Concept Application: Naval Warfare Scenario Acquisitions

Our naval warfare scenario concept application problem is based on developing a complex military system, through selection of constituent systems (from a candidate set of systems), and allocating DoD program managers in a way that maximizes the complex system performance, while minimizing risks associated with mismatches between program manager competencies and system development. The performance of the complex system is based on an aggregated performance index of its constituent systems, and risks of mismatches between program manager competencies and system development are reflected in each program manager's competencies in executing the Technology Maturation & Risk Reduction phase of the defense acquisition life cycle.

Table 5. Candidate Systems or Naval Warfare Scenario

		SoS Capabilities (Outputs)			Capabilities (Outputs)				Cost	Num Power	Num Comm	TRL
No.	System Name	SoS CAP1	SoS CAP2	SoS CAP3	Power.	Comm.	Power Req.	Comm Req.	[$]	Links	Links	
1	Control Station 1	150	0	0	150	0	0	0	$10,000.00	3	3	9
2	Control Station 2	300	0	0	300	0	0	0	$20,000.00	3	3	9
3	Control Station 3	450	0	0	450	0	0	0	$300,000.00	3	3	9
4	Control Station 4	600	0	0	600	0	0	0	$400,000.00	3	3	6
5	Control Station 5	750	0	0	750	0	0	0	$500,000.00	3	3	4
6	First Satellite 1	0	0	100	0	0	75	95	$500,000.00	3	3	9
7	First Satellite 2	0	0	200	0	0	125	150	$650,000.00	3	3	9
8	First Satellite 3	0	0	300	0	0	150	250	$750,000.00	3	3	7
9	First Satellite 4	0	0	400	0	0	175	350	$850,000.00	3	3	3
10	First Satellite 5	0	0	500	0	0	185	450	$900,000.00	3	3	4
11	UAV-1	20	0	0	0	0	100	0	$200,000.00	3	3	9
12	UAV-2	30	0	0	0	0	200	0	$300,000.00	3	3	9
13	UAV-3	40	0	0	0	0	300	0	$400,000.00	3	3	4
14	UAV-4	50	0	0	0	0	120	0	$450,000.00	3	3	3
15	UAV-5	60	0	0	0	0	300	0	$500,000.00	3	3	2
16	Carrier Ship -1	0	5	0	0	0	50	0	$500,000.00	3	3	9
17	Carrier Ship -2	0	10	0	0	0	150	0	$600,000.00	3	3	9
18	Carrier Ship -3	0	20	0	0	0	200	0	$700,000.00	3	3	2
19	Second Satellite 1	0	0	100	0	100	0	0	$50,000.00	3	3	9
20	Second Satellite 2	0	0	200	0	200	0	0	$60,000.00	3	3	9
21	Second Satellite 3	0	0	300	0	300	0	0	$70,000.00	3	3	7
22	Second Satellite 4	0	0	400	0	400	0	0	$80,000.00	3	3	3
23	Second Satellite 5	0	0	500	0	500	0	0	$90,000.00	3	3	3

Table 5 lists a catalogue of systems and their hypothetical characteristics. The table shows 23 available systems that can be acquired towards development of an overarching capability, across five classes of systems (Control Station, First Satellite, UAV, Carrier Ship, Second Satellite). The first three columns (SoS CAP1, SoS CAP 2, SoS CAP 3) list outputs of system level capabilities that directly contribute to the top-level performance of the overall complex systems. For example, Control Station 1's SoS Cap1 contribution of 150 refers to a capability of 150Mbps of communication bandwidth that contributes directly to the overall performance index of the complex system in general. Columns three and four are capabilities that do not contribute directly to the top-level performance index, but contribute to satisfying constraints at a lower level of abstraction; for example, the same Control Station 1 generates 150 units of power that can be distributed to other systems that connect to it. While power is an output of Control Station 1, it is not a capability that directly contributes to the top-level capabilities of the overall complex system. Columns 5–6 are the requirements of each system. Column 7 reflects acquisition costs. Columns 8 and 9 reflect the number of other systems can link to each system; this constraint, in the case of Control Station 1, is to be able to provide power to up to three other systems that connect to it.

The last column is the Technology Readiness Level (TRL) of each system. We assume that high TRL numbers denote a commercial off-the-shelf type of system that has relatively straightforward acquisition processes in place, where as a lower TRL level system will require the assignment of a program manager to develop and mature the system towards final acquisition. We assume a finite number of each type of program managers that are available to be assigned to each system listed in Table 5. For simplicity, the measure of performance of each program manager type, in executing acquisition tasks listed in Table 1, is defined as the Euclidean norm of program managers dimensional scores (columns 5–8) that are less than the estimated required values (columns 1–4). The overall performance of the program manager in executing acquisition tasks is taken as simply the average Euclidean norm values across the four loops—here, we term this as an average risk. Values of the average risk and population of program managers for each type are tabulated in Table 6.

Table 6. Concept Problem Program Manager Population per Type

PM Type	Population	Average Risk
I	2	4.1
II	2	5.3
III	2	4.7
IV	2	10.1

Mathematical Formulation: Mixed Integer Programming (MIP)

We formulate our concept problem of maximizing a complex system's performance while minimizing program manager competency related risks as a multi-objective optimization problem. We adopt a modified version of a prior optimization model by Davendralingam that views a complex systems architecture as a collection of nodes with interdependency rules that govern their connectivity. The resulting mathematical program is as follows:

$$\max\left(\frac{\sum_i S_{ic}\cdot w\cdot x_i^B - R_c}{R_c}\right) \qquad (1)$$

subject to:

$$\sum_i x_{cij} \geq x_j^B S_{rj} \qquad (2)$$

$$\sum_j x_{cij} \leq x_i^B C_{ci} \qquad (3)$$

$$x_1^B + \ldots + x_n^B = T \qquad (4)$$

$$\sum_c x_{cij} - x_{ij}^{c_{bin}} M \leq 0 \qquad (5)$$

$$M\sum_c x_{cij} - x_{ij}^{c_{bin}} \geq 0 \qquad (6)$$

$$\sum_i x_{ij}^{C_{bin}} \le n_{max} \tag{7}$$

$$\sum_i x_{cij} - \sum_j x_{cij} - x_j^B S_{rj} = 0 \tag{8}$$

$$\sum_i x_{iq}^{PM} - x_{q \in Q} M \le 0 \quad \forall q \in Q_{TRL<8} \tag{9}$$

$$M \sum_i x_{iq}^{PM} - x_{q \in Q'} \le 0 \quad \forall q \in Q_{TRL<8} \tag{10}$$

$$\sum_t x_{tq}^{PM} \le 1 \qquad \forall q \in Q \quad t=1 \ldots 4 \tag{11}$$

$$C_{req}^{PM} x_{iq}^{PM} \le C_{cap}^{PM} x_{iq}^{PM} \quad \forall q \in Q_{TRL<4} \tag{12}$$

$$C_{req}^{PM} x_{iq}^{PM} \le E_{max} \tag{13}$$

$$x_{cij} \in \text{real, integer}, x_q^B \in \text{binary}, x_{iq}^{PM} \in \text{binary}, x_{ij} \in \text{binary}$$

where:

S_{ic}	capability (c) of system (i)
w,	weighting factor vector of SoS capabilities (constant)
x_{ib}	binary decision variable for selecting system (i)
R_c	base SoS capability for normalization
x_{cij}	quantity of capability (c) between system (i) and (j)
x_{ij}	adjacency matrix (binary) that indicates connection between systems (i) and (j)
S_{rj}	requirement (r) of system (j)
M	Big-M constant value
Q	-set of all possible system choices (q = 1 … 23)

The mathematical model as represented by Equations 1–13 represent the formulation of a mixed integer linear programming model. The "mixed" term denotes the existence of both integer and continuous decision variables. Equation 1 is the objective function that represents the maximization of the overall complex system capability index. Here, the capability index is the normalized sum of capabilities of the complex system level capabilities (columns 1–3 in Table 5), where the normalization is done with respect to some lowest common denominator, R. Equation 2 ensures that for each system type (j) selected, there is sufficient capability type (C) being received from other connecting systems (i) that can satisfy the requirement type (R). Equation 3 ensures that the amount of capability provided by each system, type (i) for each capability type (c) does not exceed the maximum capability of type (c) for the system. Equation 4 generically defines mutual exclusivity rules for systems—for example, if selection of system 1 (x1) and system 2 (x2) is a mutually exclusive condition, then the constraint would be (X1+X2 ==1) where x1 and x2 are binary variables and constant T denotes the condition that the sum of the can only result in one system. Equations 5 and 6, more specifically, follow a "Big-M formulation" that facilitates the calculation of the number of connections that can be made to individual nodes. Equation 7 constrains the number of connections that can exist for each system type (i) and for each capability type (c) for the system. Equation 8 enforces that the total of some capability (q) that is supplied to a node (e.g., power flow or communications bandwidth), combined with its inherent capability (q) is not exceeded by demand for the capability from connected nodes.

Equations 9 and 10 jointly enforce that if a system type (q) is selected from the set of systems that have a TRL level less than 9, then a program manager must be assigned to the system. Equations 9 and 10, like Equations 5 and 6, employ the use of a Big-M formulation where the pairs of constraints act as logical conditions. Equation 11 sets the condition that only up to one program manager from the four types (t) can be assigned to each system. Equation 12 imposes the condition that for each system type (q) that belongs to the set of systems with a TRL of level 5 or below, the program manager assigned to the system needs to have a Great Eight competency score that at least meets the score for the requirements of a critical subset of the Great Eight in columns 1–3 of Table 4; these critical subsets are for the top three highest scores for the loops (1–2). Equation 13 limits the total performance error, accumulated due to assigning program managers across different systems, to a maximum value of Emax; this value is varied to generate an efficiency frontier that trades off the overall complex system performance against the uncertainty in overall program manager performance. It must be noted that while Equation 13 is a linear equation and is reflective of the relatively simple model used for our concept problem, it does not detract from more complex forms of modeling for program manager performance. With a richer collection of data, approaches that account for more explicit interdependencies between program manager interactions, when allocated to systems, can be modeled in quadratic forms (Davendralingam et al., 2012) that can be efficiently included in the current modeling framework, even under conditions of data uncertainty. Furthermore, there are a range of robust optimization techniques that can be applied to address data uncertainty as well (Davendralingam et al., 2012).

Results

The resulting optimization model as represented by Equations 1–13 is solved in MATLAB 2016b using the YALMIP toolbox with the GNU Linear Programming Kit (GLPK) solver. The problem is solved for a bounded range of values of E_{max} in Equation 13 ($5 \leq E_{max} \leq 50$) to generate the Pareto frontier that trades off the overall complex system capability index (optimal values of the objective function) against overall program manager performance; this includes Pareto filtering to only include non-dominated solutions on the efficiency frontier. Figure 5 shows the filtered Pareto frontier generated by solving the optimization model for each range value of E_{max} . Table 7 provides the breakdown of selected systems that comprise the portfolio of systems within the overall complex system, and program manager allocations across each portfolio point on the efficiency frontier.

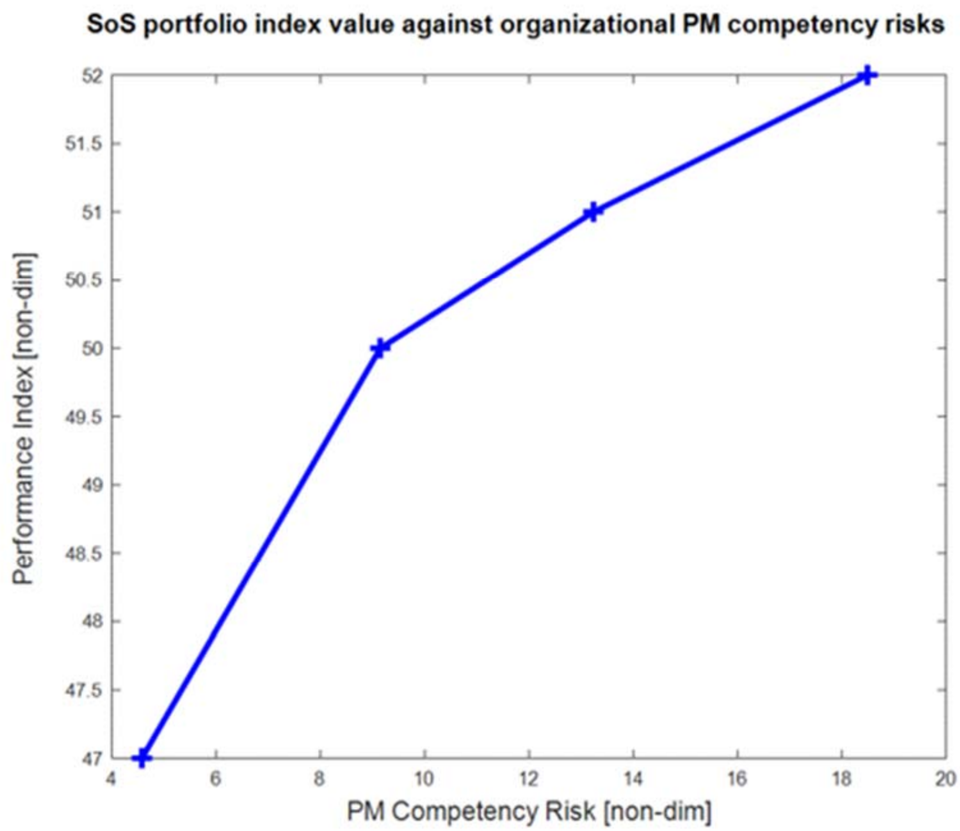

Figure 5. Efficiency Frontier of Performance Against PM Competency Risk (Risk Measured as Average Mean Squared Error)

Table 7. Portfolio of Systems and Program Manager Allocations

No.	Candidate Systems	Portfolio 1	Portfolio 2	Portfolio 3	Portfolio 4
1	Control Station 1	-	-	-	-
2	Control Station 2	-	-	-	-
3	Control Station 3	-	-	-	-
4	Control Station 4	-	-	-	-
5	Control Station 5	X	X	X	X
6	First Satellite 1	-	-	-	-
7	First Satellite 2	X	-	-	-
8	First Satellite 3	-	-	-	-
9	First Satellite 4	-	-	X	X
10	First Satellite 5	-	X	-	-
11	UAV-1	-	-	-	-
12	UAV-2	X	X	X	X
13	UAV-3	-	-	-	-
14	UAV-4	-	-	-	-
15	UAV-5	-	-	-	-
16	Carrier Ship -1	-	-	-	-
17	Carrier Ship -2	X	X	X	-
18	Carrier Ship -3	-	-	-	X
19	Second Satellite 1	-	-	-	-
20	Second Satellite 2	X	X	-	-
21	Second Satellite 3	-	-	-	X
22	Second Satellite 4	-	-	X	-
23	Second Satellite 5	X	X	X	X
Program Manager Type		**# of PMs (system # PM allocated to)**			
I		-	-	1 (9)	-
II		-	-	-	2(9,21)
III		1 (23)	2 (23,10)	2(22,23)	2(18,23)
IV		-	-	-	-

The results generated through solving the optimization problem of Equations 1–13 provide a way for decision-makers to assess potential tradeoffs between selecting different complex system architectures (here, portfolio of interconnected systems) and organizational architecture (here, program manager type allocations) by relegating some combinatorial aspects of the problem to the algorithm and delegating decision-making to the practitioner. The results show the progressive levels of complex system performance that can optimally be achieved, given each prescribed acceptable level of risk associated with the program manager performance, for each portfolio. As more capable systems are brought into the picture, to generate a higher performing complex system, program managers are additionally added in an optimized sense, in a manner that bounds risk the sequential increments enforced in Equation 13. The program manager allocation also adheres to the rulesets established (for example, the constraints established for allocation of program managers to systems with TRL<9 and TRL<5 as established in prior sections). While an initial instinct may be to first select program managers that are, on average, the least risky following Table 6, we see instead that the optimization selects program manager Type III in Portfolio 1 and Portfolio 2, due to the enabling effect that Type III manager has on developing low TRL systems with a higher potential to improve the complex system performance index. Another useful observation of the results presented, is that the solution generated by the optimization routine, reveals potential pathways for evolving an architecture; for example, when considering portfolios 3 and 4, we observe that a future upgrade from portfolio 3 to 4 will include retirement of Carrier-Ship 2 and a Second Satellite-2 unit, in favor of a Second Satellite-3 unit and a Carrier-Ship 3; this path of system addition and replacement is complemented by the need to replace a Type I program manager with two Type II program managers to facilitate the architectural transition. Early stage knowledge on such shifts can enable the correct requirements to be set on what type of program managers to look for or train for these future updates, thereby minimizing risks and organizational misalignments.

As the number of candidate systems increases and the dependencies increase as well, it becomes very difficult to objectively select systems that constitute a complex system and program managers that manage each of the constituent systems without the aid of quantitative means such the mathematical framework presented in this paper. The mixed integer programming formulation is efficient even for much larger instances of number of systems (and/or number of program manager types), assuming the same problem abstraction being used in this paper. Furthermore, the MIP perspective lends itself to further formulations of the problem at hand to better account for various forms of interdependencies between product and organization and data uncertainty.

Concluding Statements and Future Work

The approach presented in this paper represents a preliminary quantitative framework that facilitates the optimal selection of an organizational architecture and product architecture (in this case, a complex system architecture). The approach leverages theories from industrial organizational psychology and mathematical programming techniques from operations research to yield a unified approach that facilitates the selection of an optimal composition of systems and organizational structure (in this case, program managers) towards achieving a desired complex system performance.

We demonstrate the work for a concept problem based on a naval defense acquisition scenario and present the mathematical formulation and example solution of the problem. The concept problem utilizes a combination of qualitative and quantitative measures, driven by prior literature and *a priori* insights from program manager competency literature, to form the foundation of the organizational elements in our concept problem. The resulting mathematical programming problem is posed in a very flexible framework of a mixed integer linear programming problem—to which there are very well understood means of solution, even for large scale problems.

Potential future research may encompass extensions on the modeling techniques for capturing interdependency behaviors between interacting organizational elements (e.g., modeling interaction behaviors of program managers) and adapting the mathematical modeling to more explicitly include such interactions. Furthermore, additional elements of organizational structure, such as acquisition processes relevant to the acquisition of individual systems, can also be brought to bear within this framework.

References

Austin-Brenemann, J., Honda, T., & Yang, M. (2012). A study of student design team behaviors in complex system design. *Journal of Mechanical Design, 134*.

Bartram, D. (2005). *The great eight competencies: A criterion-centric approach to validation.*

Bartram, D., Robertson, I. T., & Callinan, M. (2002). Introduction: A framework for examining organizational effectiveness. In I. T. Robertson, M. Callinan, & D. Bartram (Eds.), *Organizational effectiveness: The role of psychology* (pp. 1–10). Chichester, UK: Wiley.

Davendralingam, N., Mane, M., & DeLaurentis, D. A. (2012). Capability and development risk management in system-of-systems architectures: A portfolio approach to decision making. In *Proceedings of the Ninth Annual Acquisition Research Symposium*. Monterey, CA: Naval Postgraduate School.

Defense Acquisition University. (2009). *Integrated defense acquisition, technology, and logistics life cycle management system*.

DeLaurentis, D. (2015). Optimal selection of organizational structuring for complex system development and acquisitions. *FY15 Acquisition Research Program*.

DoD. (2015). *Operation of the defense acquisition system* (DoD Instruction 5000.02). Washington, DC: Author.

Golob, M. P. (2002). *Implementing project management competencies in the workplace.* Unpublished doctoral dissertation, Capella University, Minneapolis, MN.

Honda, T., Ciucci, F., Lewis, K., & Yang, M. (2015). Comparison of information passing strategies in system-level modeling. *AIAA Journal, 53*(5), 1121–1133.

Kurz, R., & Bartram, D. (2002). Competency and individual performance: Modelling the world of work. In I. T. Robertson, M. Callinan, & D. Bartram (Eds.), Organizational effectiveness: *The role of psychology* (pp. 227–255). Chichester, UK: Wiley.

MacCormack, A., Baldwin, C., & Rausnak, J. (2012). Exploring the duality between product and organizational architectures: A test of the “mirroring” hypothesis. Elsevier, 41(8), 1309–1324.

Nijhuis, S. A., Vrijhoef, R., & Kessels, J. W. M. (2015). Towards a taxonomy for project management competences. *Procedia-Social and Behavioral Sciences, 194,* 181–191.

Wood, R. (2010). How well are PMs doing? Industry view of defense program manager counterparts. *Defense Acquisition University*, 206–218.

Acknowledgments

The authors would like to acknowledge Dr. Robert Kenley (Professor of Practice, School of Industrial Engineering, Purdue University) and Mr. Daniel Dumbacher (Professor of Practice, School of Industrial Engineering, Purdue University) for their invaluable insights into the defense acquisition life cycle and related processes. Further acknowledgments to Dr. Roy Wood for providing data from the survey paper on program manager competencies and deeply useful discussions on program manager assessments. The authors also wish to acknowledge the Naval Postgraduate School for funding this work under NPS Grant: N00244-16-1-0005.

Acquisition Research Program
Graduate School of Business & Public Policy
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net